

Integration eines interaktiven Whiteboards (Drawable) für SEE

-

Integration of an interactive whiteboard (drawable) for SEE
Masterarbeit

Michel Krause

Matrikelnummer: 4126108

26. März 2024



Fachbereich 3 — Mathematik und Informatik
Studiengang Informatik

1. Gutachter: Prof. Dr. Rainer Koschke
2. Gutachter: Prof. Dr. Udo Frese

ZUSAMMENFASSUNG

In dieser Masterarbeit wird die Integration eines interaktiven Whiteboards und Sticky-Notes als Drawables sowie deren Interaktionsmöglichkeiten für die Visualisierung SEE beschrieben.

Für die Interaktionsmöglichkeiten wird behandelt, wie frei gezeichnet werden, elf verschiedene geometrische Formen sowie Bilder hinzugefügt, Texte geschrieben und Mind Maps erstellt werden können. Des Weiteren wird darauf eingegangen, wie diese erstellten Objekte transformiert, gelöscht, gespeichert und geladen werden können. Zudem wird erklärt, wie Drawables in Form von Sticky-Notes während der Laufzeit hinzugefügt werden können.

Anschließend erfolgt die Erläuterung der Usability-Evaluierung dieser Integration. Hierzu wurde eine Studie mit vierzehn Probanden durchgeführt, die sieben verschiedene Aufgaben bewältigen mussten, um sämtliche Aktionen der Integration zu überprüfen.

Nach jeder Aufgabe füllten die Probanden einen After-Scenario Questionnaire (ASQ)-Fragebogen und am Ende der Studie einen System Usability Scale (SUS)-Fragebogen aus.

Die Usability wurde anhand dieser Fragebögen bewertet.

Des Weiteren wurde überprüft, ob Probanden, die aus dem Informatikbereich stammen, die Usability besser bewerteten als Nicht-Informatiker. Zudem erfolgte ein Vergleich der benötigten Zeit zwischen den beiden Gruppen.

Einige Probanden nahmen an einem zweiten Studiendurchlauf teil, um zu untersuchen, wie sich die Sicht auf die Usability bei erneuter Verwendung ändert und ob sich die benötigte Zeit für die Aufgabenbewältigung verkürzt.

Die Evaluation der Studie ergab, dass beide Gruppen die Integration zufriedenstellend bedienen und problemlos nutzen konnten. Zudem wurde mit dem SUS-Fragebogen eine exzellente Usability für die Integration festgestellt. Außerdem konnten die Informatiker die Aufgaben schneller und leichter bewältigen als Nicht-Informatiker, dabei wurde ein signifikanter Unterschied zwischen der empfundenen Komplexität und dem Aufwand zwischen den beiden Gruppen festgestellt.

Die Untersuchung des zweiten Durchlaufs stellte signifikante Unterschiede bei der benötigten Zeit, der SUS-Bewertung und den Daten des ASQs fest. Durch diese Untersuchung wurde deutlich, dass bei einer erneuten Verwendung die Benutzer noch besser mit der Integration umgehen konnten.

Außerdem wurden acht signifikante Korrelationen zwischen den ab-

hängigen und unabhängigen Variablen aufgedeckt.
Abschließend werden verschiedene Erweiterungen und potenzielle Verbesserungen für die Zukunft dieser Integration erläutert und ein Fazit gezogen.

ERKLÄRUNG

Ich versichere, diese Masterarbeit — sofern dies nicht explizit anders gekennzeichnet wurde — ohne fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, sind als solche kenntlich gemacht.

Bremen, den 26. März 2024

Michel Krause

DANKSAGUNG

Zuallererst möchte ich mich herzlich bei Prof. Dr. Rainer Koschke bedanken, der mir diese Masterarbeit ermöglicht hat und stets für meine Fragen zur Verfügung stand.

Ebenso möchte ich meinen Dank an Prof. Dr. Udo Frese richten, der die Zweitprüfung übernimmt.

Ein besonderer Dank geht an Falko, der mir hilfreiche Unterstützung bei Fragen zum Menü und der LaTeX-Vorlage geboten hat.

Abschließend möchte ich mich bei den Korrekturlesern sowie den Probanden der Studie für ihre wertvolle Mitarbeit und Unterstützung bedanken.

GENDER-HINWEIS

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers verzichtet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

INHALTSVERZEICHNIS

1	Einführung	1
1.1	Motivation	1
1.1.1	Unity	3
1.1.2	SEE	4
1.2	Forschungsfrage und Struktur	5
2	Konzept	7
2.1	Anforderungen und Ziele	7
2.2	Verwandtes Projekt	12
3	Implementierung	15
3.1	Funktionsweise	15
3.2	Drawable-Aufbau	17
3.2.1	Drawable-Varianten	18
3.2.1.1	Whiteboard	18
3.2.1.2	Sticky-Note	20
3.2.2	AttachedObject-Objekt	21
3.2.3	Drawable-Seitenränder-Funktionalität	22
3.2.4	Drawable-Konfiguration	23
3.3	ValueHolder	24
3.4	Drawable-Typen	24
3.4.1	Line-Drawable-Typ	26
3.4.2	Text-Drawable-Typ	30
3.4.3	Image-Drawable-Typ	34
3.4.4	Mind Map-Drawable-Typ	35
3.5	Objekte finden	38
3.6	Aktionen	40
3.6.1	HSV-Color-Picker	42
3.6.2	Hinzufüge-Aktionen	43
3.6.2.1	Draw Freehand-Aktion	44
3.6.2.2	Draw Shape-Aktion	48
3.6.2.3	Write Text-Aktion	57
3.6.2.4	Add an Image-Aktion	61
3.6.2.5	Mind Map-Aktion	64
3.6.3	Color Picker-Aktion	66
3.6.4	Transformations-Aktionen	68
3.6.4.1	Blink-Effekt	68
3.6.4.2	Kollisionserkennung	69
3.6.4.3	Nearest Point(s) - Berechnung für Linien	70
3.6.4.4	Edit-Aktion	71
3.6.4.5	Move or Rotate-Aktion	77
3.6.4.6	Scale-Aktion	80
3.6.4.7	Change The Sorting Layer-Aktion	82

	3.6.4.8	Cut, Copy, Paste-Aktion	83
	3.6.4.9	Move a Point-Aktion	85
	3.6.4.10	Line Split-Aktion	87
	3.6.5	Save-Aktion	89
	3.6.6	Load-Aktion	92
	3.6.7	Lösch-Aktionen	93
	3.6.7.1	Line Point Erase-Aktion	93
	3.6.7.2	Line Connection Erase-Aktion	96
	3.6.7.3	Erase-Aktion	98
	3.6.7.4	Cleaner-Aktion	99
	3.6.8	Sticky-Note-Aktion	100
	3.6.8.1	Spawn	101
	3.6.8.2	Move	104
	3.6.8.3	Edit	105
	3.6.8.4	Delete	106
	3.7	Drawable-Synchronizer	107
4		Evaluation	109
	4.1	Ziele und Forschungsfragen	110
	4.2	Versuchsaufbau	113
	4.2.1	Fragebögen	117
	4.2.1.1	Post-Study	117
	4.2.1.2	Post-Task	119
	4.2.2	Pilotstudie	121
	4.2.3	Tools	122
	4.3	Probanden und Objekt der Studie	123
	4.4	Gestellte Aufgaben	125
	4.4.1	Aufgabe 1	125
	4.4.2	Aufgabe 2	127
	4.4.3	Aufgabe 3	128
	4.4.4	Aufgabe 4	130
	4.4.5	Aufgabe 5	131
	4.4.6	Aufgabe 6	132
	4.4.7	Aufgabe 7	134
	4.5	Auswertung	134
	4.5.1	Demografische Daten	135
	4.5.2	RQ1: Insgesamte Benutzerfreundlichkeit	143
	4.5.3	RQ2: Vergleich Informatiker und Nicht-Informatiker	148
	4.5.4	RQ3: Zweiter Studiendurchlauf	159
	4.5.5	Vergleich Informatiker und zweiter Durchlauf der Nicht-Informatiker	171
	4.5.6	Korrelationen	172
	4.5.7	Kommentare der Probanden	176
	4.6	Threats to Validity	177
	4.6.1	Interne Validität	178
	4.6.2	Externe Validität	179
5		Ausblick und Fazit	181

5.1	Ausblick	181
5.1.1	Neue Drawable-Typen	181
5.1.1.1	Web	182
5.1.1.2	Video / GIF	182
5.1.1.3	PDF	183
5.1.1.4	Audio	183
5.1.1.5	UML-Diagramme	184
5.1.1.6	Zustandsautomat	188
5.1.1.7	Datenbankmodelle	189
5.1.2	Erweiterungen für den Text-Drawable-Typ . . .	191
5.1.2.1	Farbverlauf	191
5.1.2.2	Glow und Lighting	194
5.1.2.3	Unterschiedliche Schriftarten (Fonts) .	196
5.1.2.4	Text mit VR/AR	196
5.1.3	Verbesserungen für den Image-Drawable-Typ . .	197
5.1.4	Neue Aktionen	198
5.1.4.1	Add Point to Line - Linienpunkt hinzu- fügen	198
5.1.4.2	Mind Map Überlappungsproblemlöser	198
5.1.4.3	Mind Map Sortierer	199
5.1.4.4	Order Reducer - Ebenenreduzierer . .	200
5.1.5	Konfigurationsmenü	200
5.1.6	Erweiterungen für das Whiteboard	200
5.1.6.1	Seiten	200
5.1.6.2	Farbe ändern	201
5.1.7	Virtual Reality und Augmented Reality - Anbin- dung	201
5.1.8	Verbesserung für den Drawable-Synchronizer .	202
5.1.9	Weitere Erweiterungen sowie Verbesserungen .	202
5.2	Fazit	204
A	Glossar	207
B	Akronyme	213
C	Abbildungsverzeichnis	215
D	Tabellenverzeichnis	221
E	Literaturverzeichnis	223

EINFÜHRUNG

Diese Masterarbeit behandelt die Integration und Evaluation eines interaktiven *Drawable*-Systems.

Für das Schreiben dieser Arbeit wurde die von Falko Galperin erstellte LaTeX-Vorlage verwendet. Einige bereits vorhandene Glossar- und Akronymbegriffe wurden daraus übernommen.

Im ersten Kapitel werden die Motivation, einige Grundlagen, die zu erforschende Forschungsfrage sowie die Struktur dieser Arbeit erläutert.

1.1 MOTIVATION

Die Planung ist für einen wohldefinierten Softwareentwicklungsprozess unerlässlich.

Um die Bedeutung der Planung in einem Softwareprozess zu betonen, ziehen [Kecher u. a. \(2018\)](#) einen Vergleich mit dem Hausbau heran.

Sie beschreiben, dass die Bauphase eines Hauses auch erst nach einer gründlichen Planung des Architekten beginnen kann.

Der Softwareexperte [Brooks \(2002\)](#) unterstreicht die Notwendigkeit von Planung mit dem folgenden Satz:

“The bearing of a child takes nine months, no matter how many women are assigned.”

Damit möchte er verdeutlichen, dass es Aufgaben gibt, die selbst durch eine Erhöhung der Personenanzahl nicht beschleunigt werden können und daher eine sorgfältige Planung erfordern ([Keller, 2022](#)).

Besonders die Planung der Softwarearchitektur ist von entscheidender Bedeutung. Diese Schlussfolgerung lässt sich aus dem Beitrag von [Koschke \(2022\)](#) ableiten, in dem er die bedeutende Rolle der Architektur für die weiteren Prozesse der Softwareentwicklung beschreibt.

Eine Möglichkeit zur Unterstützung und Verbesserung des Planungsprozesses besteht darin, Notizen zu erstellen. Verschiedene Experten, wie beispielsweise David Allen¹, betonen die Wichtigkeit von Notizen für effektives Planen. Ein bekannter Grundsatz von Allen lautet:

“Your mind is for having ideas, not holding them.”

Dieser spiegelt die Idee wider, dass der Verstand zum Generieren von Ideen da ist und nicht um diese festzuhalten. Daher sollten Ideen

Drawable: Ist eine Ressource, die grafische Inhalte repräsentieren kann. In dieser Arbeit umfasst dies die Fläche des Whiteboards und des Sticky-Notes auf dem gezeichnet, geschrieben und Bilder hinzugefügt werden kann.

¹ Erfinder der “Getting Things Done“-Methode

in Notizen übertragen werden, um den Kopf frei für neue Ideen zu bekommen (Aßmann, 2021; Allen, 2012).

Mayer (2009) schrieb in seinem Buch "Multimedia Learning" über die kognitive Theorie des multimedialen Lernens. Darin beschreibt er, dass Menschen effektiver lernen können, wenn ihnen visuelle Darstellungen präsentiert werden, die eine Kombination aus verbaler und nonverbaler Form sind, im Vergleich zu rein verbalen Darstellungen.

Des Weiteren identifiziert er die zwei Hauptziele des Lernens als das Erinnern und das Verstehen.

Eine weitere Theorie, die auf diese visuelle Darstellungsform zurückgreift und als Grundlage der zuvor beschriebenen Theorie angesehen werden kann, ist die von Allan Paivio entwickelte Dual-Coding-Theorie. Diese beschreibt, dass Menschen dazu neigen, Informationen effektiver zu verarbeiten und zu erinnern, wenn sie in der zuvor beschriebenen visuellen Darstellungsform präsentiert werden. Paivio erklärt, dass dies auf die Verwendung von zwei kognitiven Kanälen für die Informationsverarbeitung zurückzuführen ist. Den verbalen und den nonverbalen Kanal. Die Aktivierung dieser beiden Kanäle trägt dazu bei, ein tieferes Verständnis für die dargestellte Information zu entwickeln (Themann, 2015; Cuofano, 2023).

Aus den beiden Theorien resultiert, dass Menschen Informationen effektiver verstehen und verarbeiten können, wenn diese sowohl in verbaler als auch in nonverbaler visueller Form dargestellt werden.

Des Weiteren ermöglicht diese Darstellungsform das visuelle Planen. Buzan und Buzan (2013) beschreiben, wie Visualisierungen, in ihrem Fall Mind Maps, die Kreativität fördern und komplexe Ideen strukturieren können. Dieser Aspekt kann besonders beim Planen von Projekten oder Aufgaben hilfreich sein.

Außerdem tragen visuelle Darstellungen zu einer verbesserten Kommunikation bei. In diesem Kontext wird die bekannte Metapher "Ein Bild sagt mehr als tausend Worte" relevant (Buzan und Buzan, 2013). Durch visuelle Elemente können Ideen effektiver kommuniziert werden, da sie Sprachunterschiede überbrücken und abstrakte Konzepte leichter verständlich darstellen können (Personal-Wissen, 2023).

Aufgrund der zuvor genannten positiven Aspekte dieser Darstellungsform ist es effektiver, eine Kombination aus verbalen und nonverbalen Elementen für Notizen zu verwenden. Eine solche Notizform wird auch als visuelle Notizen bezeichnet.

Sketchnotes sind visuelle Notizen, die sich aus verbalen und nonverbalen Elementen sowie einer Struktur zusammensetzen (Künzel und Schmitt, 2023).

Die Struktur beschreibt dabei, wie die Sketchnote aufgebaut ist. Rohde

(2014) stellt die fünf Grundelemente einer Sketchnote vor, mit denen alles gezeichnet werden kann, was man möchte: Kreise, Quadrate, Dreiecke, Linien und Punkte.

Er hebt hervor, dass es bei Sketchnotes um das Festhalten und Vermitteln von Ideen geht und nicht um Kunst. Eine einfache Zeichnung kann dieselbe Idee genauso gut übermitteln wie eine aufwendige Illustration. Dies verdeutlicht er anhand einer schlechten und einer guten Zeichnung eines Hundes mit dem Kommentar "Hund bleibt Hund".

Ein wichtiger Aspekt ist die Tatsache, dass eine Sketchnote sowohl analog als auch digital erstellt werden kann. Somit kann ein Drawable als große Sketchnote verwendet werden.

Des Weiteren sind Sketchnotes nicht auf ein bestimmtes Anwendungsgebiet beschränkt, sie können überall eingesetzt werden, wo ihre Anwendung als sinnvoll erachtet werden kann (Moritz, 2021).

Basierend auf den zuvor erläuterten Aspekten soll mittels dieser Masterarbeit eine Integration von visuellen Darstellungen für Drawables geschaffen werden. Dadurch sollen unter anderem Planungsprozesse und das Festhalten von Notizen in einem kollaborativen Meeting in *Software Engineering Experience (SEE)* unterstützt werden.

Im Folgenden werden die wesentlichen Grundlagen dieser Arbeit erläutert. Anschließend wird die Forschungsfrage und die Struktur dieser Arbeit vorgestellt.

1.1.1 Unity

Unity ist eine leistungsstarke und weitverbreitete Entwicklungs-Engine für 2D- und 3D-Anwendungen, insbesondere für Spiele, die von Unity Technologies im Jahr 2005 bereitgestellt wurde.

Die Engine bietet eine Multiplattform-Unterstützung, wodurch Spiele nicht nur für Windows, Linux und Mac, sondern auch für Konsolen, mobile Endgeräte und VR/AR bereitgestellt werden können.

Zum Entwickeln wird in Unity die Programmiersprache C# verwendet. Durch den Unity Asset Store werden diverse vorgefertigte *Assets* bereitgestellt, die eingebunden werden können, um die Entwicklungszeit zu verkürzen.

Unity Technologies veranstaltet seit 2008 regelmäßig sogenannte Unity Awards, bei denen die Leistungen und Innovationen der Community in verschiedenen Kategorien, wie z. B. "Bestes Mobiles" oder "Bestes AR/VR-Spiel", gewürdigt werden (Schuhmann, 2014; Pearson, 2014). Bekannte Spiele, die durch Unity entwickelt wurden, sind z. B. Hearthstone und Mario Run.

Hearthstone gewann bei den Unity Awards 2014 zwei Preise, einmal für das beste Gameplay und den golden Cube. Der golden Cube war 2014 der höchste zu gewinnende Preis (Pearson, 2014).

Asset: Ist eine digitale Ressource wie ein Skript, ein 3D-Modell, etc., das von Unity verwendet wird, um visuelle oder auch akustische Inhalte in das Spiel zu integrieren. (Nicolasreto, 2023; Seifert und Wislaug, 2017)

1.1.2 SEE

SEE ist eine kollaborative Softwarevisualisierung für verteilte Softwareentwicklerteams, die auf der [Unity](#)-Engine basiert.

Sie ermöglicht eine abstrakte und vielseitig konfigurierbare Visualisierung von Software. Zur Darstellung der Daten der auszuwertenden Software bedient sich SEE sogenannter *Code-Cities*-Metaphern. Auf Deutsch werden diese auch als Software-Städte bezeichnet ([Koschke, 2022](#); [Koschke und et al., 2024](#)). Die dreidimensionale Code-City einer Software wird in einem virtuellen Raum dargestellt. Diese virtuellen Räume können von den Benutzern mit verschiedenen Endgeräten wie Desktop-Computer, Tablets sowie Virtual- und Augmented Reality-Systemen betreten werden, worin die Benutzer als Avatare repräsentiert werden. Vor dieser Masterarbeit konnten die Benutzer lediglich verbal und mit Gesten miteinander interagieren.

Code-City: In der Code-City-Metapher werden Softwarekomponenten durch Gebäude in einer Stadt repräsentiert, wobei die Eigenschaften dieser Gebäude verschiedene Metriken der Software ausdrücken können — z. B. könnte die Höhe eines Gebäudes der Anzahl der Codezeilen entsprechen.

In dem Artikel “Modellierung, Visualisierung und Prüfung von Softwarearchitekturen in virtueller Realität“ beschrieb [Koschke \(2022\)](#) die vier grundlegenden Anwendungsszenarien von SEE wie folgt:

- **DEBUGGING:**

Dieses Anwendungsszenario ermöglicht es, dem zu überprüfenden System bei der Ausführung zuzuschauen. Dadurch kann das Verhalten der Software nachvollzogen oder gegebenenfalls Fehler gefunden werden.

SEE ermöglicht den Benutzern dabei, die Funktionsaufrufe sowohl in einer abstrakten visuellen Ebene als auch auf Quellcode-Ebene zu betrachten.

- **EVOLUTION:**

Mit diesem Anwendungsszenario können sich die Benutzer die Entwicklung der Software im Zeitraffer anschauen. Dabei wird mithilfe der Code-City aufgezeigt, wie sich die jeweiligen Codestellen der Software von einer Version zur nächsten verändern, welche hinzukommen oder wegfallen.

- **ARCHITEKTUR:**

Mithilfe dieses Anwendungsszenarios können die Benutzer die Architektur der Software prüfen oder rekonstruieren.

Dafür wird die statische Architektur in Form eines Graphen modelliert, wobei die Komponenten als Knoten und die Abhängigkeiten zwischen den Komponenten als Kanten dargestellt werden. Ebenso können die statische Struktur und die Abhängigkeiten des Quellcodes visualisiert werden.

Dies ermöglicht einen Vergleich der beiden Graphen, um Unterschiede beziehungsweise Verstöße gegen die Architekturvorgabe festzustellen.

- **QUALITÄT:**
Durch dieses Anwendungsszenario können die Benutzer Einblicke in die Qualität ihrer Software erhalten. Dabei können sie mithilfe verschiedener Qualitätsmetriken die visuellen Attribute der Stadt verändern. Visuelle Attribute beziehen sich hierbei auf Eigenschaften wie Höhe, Breite, Tiefe oder Farbe der Gebäude, Straßen oder Distrikte. Diese Darstellung wird durch interaktive klassische Charts erweitert, bei denen die Benutzer Punkte auswählen können. Die dazugehörigen Komponenten in der Code-City werden daraufhin hervorgehoben.

1.2 FORSCHUNGSFRAGE UND STRUKTUR

Als zentrale Forschungsfrage dieser Arbeit dient die Frage nach der Benutzerfreundlichkeit der Integration.

Um die Forschungsfrage zu beantworten, wird mit dem zweiten Kapitel damit begonnen, das zugrundeliegende **Konzept** inklusive der Zielsetzung der Arbeit sowie ein verwandtes Projekt zu beschreiben. Anschließend wird im dritten Kapitel die **Implementierung** der Integration erläutert.

Im vierten Kapitel folgt die **Evaluation** der durchgeführten Studie zur Beantwortung der Forschungsfrage. Die zentrale Forschungsfrage wird hierzu in dem Kapitel in drei Forschungsfragen unterteilt.

Abschließend wird im **fünften und letzten Kapitel** ein **Ausblick** auf mögliche Erweiterungen sowie Verbesserungen gegeben und ein zusammenfassendes **Fazit** gezogen.

KONZEPT

In diesem Kapitel werden die **Anforderungen und Ziele** dieser Masterarbeit vorgestellt und spezifiziert. Zudem werden die zugrundeliegenden Entwurfsentscheidungen zur Erfüllung der Anforderungen und Ziele erläutert.

Abschließend wird ein **verwandtes Projekt** präsentiert, wobei begründet wird, warum nicht auf diesem Projekt aufgebaut wurde.

2.1 ANFORDERUNGEN UND ZIELE

Das Hauptziel dieser Arbeit ist die Integration eines interaktiven Drawables für SEE. Interaktiv bedeutet dabei, dass verschiedene **Drawable-Typen** hinzugefügt und jeweils mittels **Transformations-Aktionen** transformiert werden können. Zusätzlich sollen Funktionen zum Speichern und Laden der Drawables sowie verschiedene **Lösch-Aktionen** für die Drawable-Typen bereitgestellt werden. Des Weiteren sollen zusätzliche Drawables in Form von **Sticky-Notes** erstellt werden können.

Ein besonderes Augenmerk liegt bei der Integration darauf, eine hohe Benutzerfreundlichkeit zu gewährleisten, sodass selbst Nicht-Informatiker die Integration als intuitiv ansehen und sie problemlos nutzen könnten.

Die Benutzerfreundlichkeit, auch als Usability bekannt, wird im vierten Kapitel, der **Evaluation**, ausgewertet. Das genaue Vorgehen sowie die Ziele dieser Evaluation werden in dem genannten Kapitel ebenfalls detailliert erläutert.

Die Drawable-Typen, die diese Arbeit bieten soll, umfassen den **Line-Drawable-Typ** für das Zeichnen, den **Text-Drawable-Typ** für das Schreiben von Texten, den **Image-Drawable-Typ** für das Hinzufügen von Bildern und den **Mind Map-Drawable-Typ**.

Dabei soll die Datenstruktur der Drawable-Typen so entworfen werden, dass sie leicht um weitere Drawable-Typen erweitert werden kann.

Für die Darstellung des **Line-Drawable-Typs** sollen dem Benutzer zwei unterschiedliche Hinzufüge-Aktionen zur Verfügung stehen. Mit der einen soll der Benutzer frei zeichnen können, während die andere es ihm ermöglicht, bestimmte geometrische Formen (Shapes) einzufügen. Die Formen sollen dabei anhand von Längeneingaben konfigurierbar sein. Die bereitgestellten Formen sollen Quadrate, Rechtecke, Rauten,

Drawable-Typ: Ist ein grafischer Inhalt, der auf einem Drawable repräsentiert werden kann. In dieser Arbeit sind dies Linien, Texte, Bilder und Mind Maps.

Sticky-Note: Ist ein Objekt, das dazu bestimmt ist, Notizen festzuhalten. Es wird auf Oberflächen von anderen Objekten angehaftet. Synonyme Bezeichnungen sind Haftnotiz oder Post-it.

Drachen, Dreiecke, Kreise, Ellipsen, Parallelogramme, Trapeze, Polygone und gerade Linien umfassen. Polygone sollen dabei so konfigurierbar sein, dass die Anzahl der Eckpunkte und die Länge der einzelnen Kanten angegeben werden kann. Für die geometrische Form "Linie" soll der Benutzer einzelne Punkte hinzufügen können, um eine gerade Linie zu zeichnen. Des Weiteren sollen verschiedene Einstellungen konfigurierbar sein:

Solid: Ist eine durchgezogene Linie, sie wird durchgängig und ohne Unterbrechungen gezeichnet.

Dashed: Ist eine gestrichelte Linie, sie wird mittels kurzen Segmentunterbrechungen gezeichnet.

Monochrome: Ist die Verwendung von einer gleichbleibenden Farbe.

Gradient: Stellt den Farbverlauf zwischen zwei Farben dar.

Two-Dashed: Stellt den Farbwechsel bei einer gestrichelten Linie bereit. Dabei werden die freigelassenen Stellen zwischen den Strichen eingefärbt.

Checkbox-Button: Ist ein Button, der zwischen ausgewählt und nicht ausgewählt umschalten kann.

Dropdown-Button: Ist ein Button, der ein Auswahlmenü in Form eines Dropdown-Menüs bereitstellt. Dieses Auswahlmenü öffnet sich erst, nachdem der Benutzer auf den Button geklickt hat.

- Die Auswahl der Linienart/Line-Kind (zwischen *Solid* und *Dashed*).
- Die Wahl der Farbart/Color-Kind (*Monochrome*, *Gradient*, *Two-Dashed*).
- Die Anpassung der Liniendicke.
- Die Wahl der primären sowie sekundären Farbe.
- Eine Loop-Funktion, die sicherstellt, dass das Linienende mit dem Linienanfang verbunden wird.
- Der Abstand der Segmente für eine gestrichelte (Dashed) Linienart.

Der **Text-Drawable-Typ** soll neben dem zuschreibenden Text auch konfigurierbare Einstellungen zur Anpassung der Textgröße, Textstyles und Textfarbe bieten. Zusätzlich dazu soll eine Option zur Aktivierung der Außenfarbe vorhanden sein, wobei sowohl die Farbe als auch die Dicke einstellbar sein sollen.

Für die Textstyles sollen, ähnlich wie bei bekannten Texteditoren wie z. B. Microsoft Word, *Checkbox-Button* zur Darstellung verwendet werden.

Die Alternative dazu wäre gewesen, ein Mehrfachauswahlmenü in Form eines *Dropdown-Buttons* zu entwerfen. Allerdings wurde diese Möglichkeit nicht gewählt, einerseits weil ein solches Dropdown-Menü für die Mehrfachauswahl derzeit noch nicht unterstützt wird und daher erst implementiert werden müsste und andererseits weil die Benutzer eher an die Benutzeroberfläche von herkömmlichen Texteditoren gewöhnt sind.

Die konfigurierbaren Einstellungen für den **Image-Drawable-Typ** sollen eine Spiegelungsoption sowie die Möglichkeit zur Farbüberlagerung bieten.

Des Weiteren soll zwischen zwei Optionen beim Hinzufügen unterschieden werden. Zum einen das Hinzufügen von einem Webbild und zum anderen das Hinzufügen von einem lokalen Bild.

Für die Auswahl eines lokalen Bildes soll ein *Filebrowser* zur Verfügung gestellt werden.

Als Alternative hätte ein Menü, ähnlich wie bei der "Load Board"-Aktion des Metric Boards, als ein *horizontales Selektormenü* bereitgestellt werden können (siehe [Abbildung 2.1](#)).

Der Nachteil bei dieser Variante ist jedoch, dass nur ein festgelegtes

Verzeichnis zur Dateiauswahl angeboten werden könnte, da der Selektor die Dateien eines gezielten Verzeichnisses anzeigt. Die Benutzer müssten dementsprechend die Bilder über das Dateiverwaltungssystem ihres Endgeräts in dieses Verzeichnis verschieben. Zudem kann die Auswahl des gewünschten Bildes mühsam werden, wenn viele Bilder in dem Verzeichnis existieren.

Daher wurde die Lösung bevorzugt, einen Filebrowser zur Dateiauswahl zur Verfügung zu stellen, um eine bessere Benutzerfreundlichkeit zu gewährleisten.

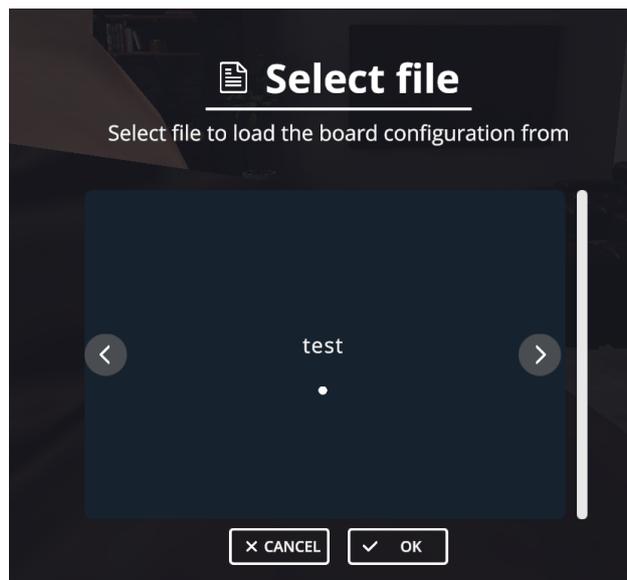


Abbildung 2.1: Demonstration der Dateiauswahl für das Metric Board.

Der **Mind Map-Drawable-Typ** soll die Bestandteile einer Mind Map repräsentieren. Es sollen Knotenarten für das zentrale Thema, Unterthemen und Blätter bereitgestellt werden. Eine Voraussetzung für die Mind Map ist, dass mindestens ein zentrales Thema vorhanden sein muss, damit ein Unterthema oder ein Blatt hinzugefügt werden kann. Für die Umrandung des Knotens sowie die Verbindungslinie zwischen den Knoten soll jeweils ein **Line-Drawable-Typ** und für den Text ein **Text-Drawable-Typ** verwendet werden.

Es wird bewusst auf eine Farbauswahl an dieser Stelle verzichtet, da zu viele Elemente und eine übermäßige Komplexität die Benutzerfreundlichkeit einschränken können (Whitenton, 2013).

Die Farbauswahl wird mittels der Edit-Aktion unterstützt, die im späteren Verlauf noch beschrieben wird.

Um die eben beschriebenen Einstellungsmöglichkeiten zu konfigurieren, sollen geeignete Menüs für die einzelnen Drawable-Typen bereitgestellt werden. Dabei ist zu beachten, dass die zu erstellenden Menüs die von SEE bereitgestellten UI-Elemente verwenden und sich an den bisherigen Design- und Entwurfsentscheidungen der UI-Elemente ori-

Filebrowser: Ist eine Komponente zur Darstellung des Dateisystems des jeweiligen Endgerätes. Dazu wird eine grafische Benutzeroberfläche bereitgestellt, worin der Benutzer Verzeichnisse nach gezielten Dateien durchsuchen und die gewünschte Datei auswählen kann.

horizontales Selektormenü: Ist ein Auswahlmenü, bei dem die Auswahlelemente nebeneinander angeordnet sind. Der Benutzer kann zwischen den Elementen durch Klicken auf einen Zurück- und Vorwärts-Buttons navigieren.

entieren sollen.

In den Menüs soll die Farbauswahl mittels des von SEE bereitgestellten HSV-Color-Pickers¹ ermöglicht werden.

Des Weiteren sollen, ähnlich wie bei üblichen Bildbearbeitungsprogrammen, wie beispielsweise Photoshop, verschiedene Ebenen zur Verfügung stehen, denen die Drawable-Typ-Objekte zugewiesen werden, um eine Reihenfolge zwischen den Objekten darzustellen.

Zudem sollen sich alle zu implementierenden Aktionen an den bisherigen SEE-Aktionen orientieren und somit folgenden Ablauf durchlaufen:

Awake -> Start -> Update* -> Stop²

Die Awake-Methode dient der Initialisierung der Aktion. Sie wird vor allen anderen Methoden aufgerufen. Danach folgt die Start-Methode, welche immer aufgerufen wird, wenn die Aktion startet. Anschließend wird die Update-Methode beliebig oft ausgeführt, um Benutzerinteraktionen für die Aktion zu behandeln. Abschließend wird die Stop-Methode aufgerufen, um die Aktion zu beenden. Dies tritt ein, sobald die Aktion entweder abgeschlossen oder verlassen wird.

Darüber hinaus soll die Funktionalität von *Undo* und *Redo* für die bereitgestellten Aktionen unterstützt werden. Außerdem sollen die ausgeführten Aktionen über das Netzwerk übertragen werden.

Im Folgenden werden die zuvor erwähnten **Transformations-Aktionen** beschrieben, die diese Integration unterstützen soll.

Undo: Ist eine Funktion, die es ermöglicht, die zuletzt durchgeführte Aktion des Benutzers rückgängig zu machen.

Redo: Ist eine Funktion, die es ermöglicht, eine zuvor rückgängig gemachte abgeschlossene Aktion zu wiederholen.

- **EDIT:**
Soll das Bearbeiten der zuvor beschriebenen Eigenschaften der Drawable-Typen ermöglichen.
- **MOVE:**
Soll das Bewegen der Drawable-Typen-Objekte ermöglichen.
- **ROTATE:**
Soll das Rotieren der Drawable-Typen-Objekte ermöglichen.
- **SCALE:**
Soll das Skalieren der Drawable-Typen-Objekte ermöglichen.
- **CHANGE LAYER:**
Soll das Ändern der Ebenenreihenfolge der Drawable-Typen-Objekte ermöglichen.
- **CUT, COPY, PASTE:**
Soll das Ausschneiden sowie Kopieren und anschließendes Hinzufügen der Drawable-Typen-Objekte ermöglichen.

¹ <https://github.com/judah4/HSV-Color-Picker-Unity>

² <https://github.com/uni-bremen-agst/SEE/blob/master/Assets/SEE/Utils/History/IRversibleAction.cs#L21>

- **MOVE A POINT:**
Soll das Bewegen eines Punktes einer Linie (**Line-Drawable-Typ**) ermöglichen.
- **LINE SPLIT:**
Soll das Teilen einer Linie (**Line-Drawable-Typ**) ermöglichen.

Des Weiteren sollen sowohl eine Primärfarbe als auch eine Sekundärfarbe als ausgewählte Farben im System festgehalten werden.

Zur Unterstützung dieser Farbauswahl soll neben den zuvor erwähnten HSV-Color-Picker noch ein Farbauswahlwerkzeug geboten werden. Im weiteren Verlauf dieser Arbeit wird dieses Farbauswahlwerkzeug als Color Picker bezeichnet. Mithilfe dieses Color Pickers soll es möglich sein, eine Farbe eines Drawable-Typen-Objekt als Primär- oder Sekundärfarbe festzuhalten.

Damit ein vorheriges Meeting fortgesetzt werden kann, muss die Integration eine Speicher- sowie Ladefunktionalität bereitstellen.

Hierbei soll für das Speichern die Möglichkeit bestehen, ein, mehrere oder alle Drawables auszuwählen, die gespeichert werden sollen.

Beim Laden sollen zwei Optionen zur Verfügung stehen. Zum einen soll die gewünschte Datei im Originalzustand geladen werden können. Zum anderen soll es möglich sein, die Datei auf ein einzelnes Drawable zu laden. Das bedeutet, dass, wenn die Datei beispielsweise zwei gespeicherte Drawables enthält, diese Option den Inhalt dieser beiden Drawables auf das ausgewählte Drawable laden soll.

Sowohl für das Speichern als auch für das Laden soll auch hier aus den oben genannten Gründen ein Filebrowser verwendet werden.

Außerdem müssen geeignete **Lösch-Aktionen** bereitgestellt werden, um die Drawable-Typen-Objekte wieder entfernen zu können.

Eine Option, die bereitgestellt werden soll, ist das vollständige Löschen eines oder mehrerer Drawable-Typ-Objekte.

Eine weitere Möglichkeit soll darin bestehen, das Drawable vollständig zu säubern. Vollständiges Säubern bedeutet, dass alle Objekte, die auf dem Drawable platziert wurden, entfernt werden sollen.

Zudem soll die Funktion integriert werden, Teile eines **Line-Drawable-Typ**-Objektes zu löschen.

Wie anfangs erwähnt, soll auch die Möglichkeit geschaffen werden, während der Laufzeit der Visualisierung weitere Drawables in Form von Sticky-Notes hinzuzufügen.

Zudem soll es möglich sein, die Eigenschaften des Sticky-Notes zu verändern. Diese Eigenschaften umfassen Farbe, Größe, Position und Rotation.

Natürlich soll es auch möglich sein, die Sticky-Notes wieder zu löschen.

Material: Ist eine Ressource, die die visuellen Eigenschaften eines Objektes definiert. Durch sie wird festgelegt, welche Texturen oder Farben auf der Oberfläche dargestellt werden und wie sich das Objekt visuell durch verschiedene Einflussfaktoren verändert. Die Veränderungen können Aspekte wie Beleuchtung, Glanz oder Schatten betreffen (Seifert und Wislaug, 2017).

Textur: Ist eine Grafik, die auf der Oberfläche eines Objektes aufgebracht wird. Sie dient dazu, das Objekt realistischer, detailgetreuer oder ansprechender zu gestalten (Seifert und Wislaug, 2017).

GameObject: Ist ein Objekt, das Entitäten repräsentiert und dem durch das Hinzufügen von Komponenten und Skripten bestimmte Eigenschaften, Funktionalitäten und Verhalten verliehen werden kann.

Renderer: Ist eine Komponente, die für die Darstellung der 3D- oder 2D-Grafiken verantwortlich ist. Er verarbeitet die visuelle Darstellung von Objekten und sorgt für ihre Anzeige (Seifert und Wislaug, 2017; Unity_Technologies, 2024a).

Abschließend soll es Benutzern ermöglicht werden, zu einem späteren Zeitpunkt der Visualisierung beizutreten und denselben Stand in Bezug auf die Drawables und Drawable-Typen zu haben. Damit ist gemeint, dass die Drawables synchronisiert werden sollen, sobald ein neuer Benutzer beitrifft.

Weitere Designentscheidungen werden im [Implementierungskapitel](#) näher erläutert, da diese eine spezielle Perspektive auf die Implementierung erfordern.

2.2 VERWANDTES PROJEKT

Ein Projekt, das die Möglichkeit zum Bemalen von 2D- und 3D-Objekten bietet, ist "2D/3D Paint", auch bekannt als "XDPaint", von Kostiantyn Saietskyi³. Diese Integration stellt eine Vielzahl von Optionen bereit. Beispielsweise können verschiedene Pinselarten und Farben oder *Materialien* zum Malen ausgewählt werden. Ein interessantes Feature ist das Füllwerkzeug, auch als Eimerwerkzeug (Bucket Tool) bekannt, das zum vollständigen Ausmalen eines Objektes verwendet wird. Die Integration kann in [Abbildung 2.2](#) betrachtet werden.

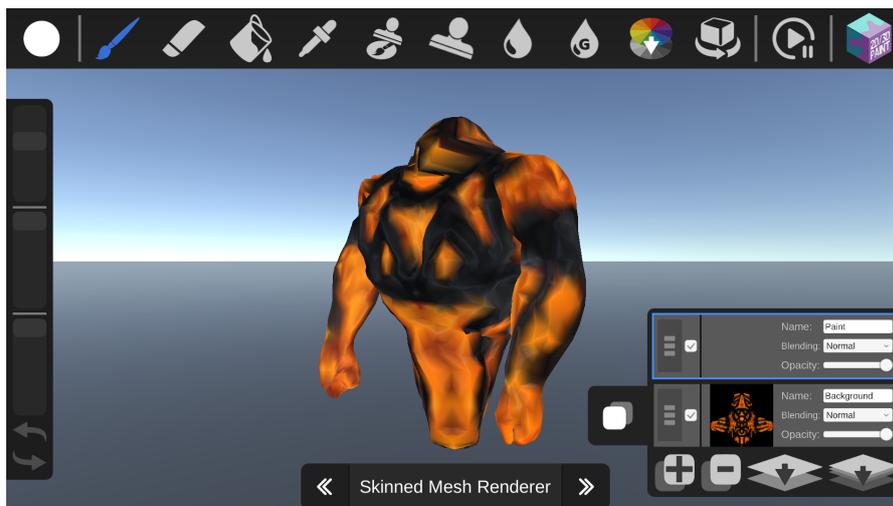


Abbildung 2.2: Beispielanwendung von XDPaint.

Aufgrund des Potenzials dieses Projektes stellt sich die Frage, warum nicht darauf aufgebaut wurde.

Das Problem hierbei ist, dass XDPaint seine Funktionalitäten mittels einer *Textur*-Manipulation durchführt (Saietskyi, n.d.).

Das setzt voraus, dass ein *GameObject* mit einem entsprechenden *Renderer* existieren muss.

Dies ist auch bei dem gegebenen Whiteboard dieser Arbeit der Fall.

³ <https://assetstore.unity.com/packages/tools/painting/2d-3d-paint-212475>

Aber entgegen der beschriebenen Zielsetzung im Abschnitt “[Anforderungen und Ziele](#)“ wären die gezeichneten Linien dann nicht mehr transformierbar, da sie keine eigenen GameObjects sind.

Zur Verdeutlichung: Texturen sind Bilder und die einzelnen Pixel innerhalb eines Bildes können beispielsweise nicht individuell rotiert werden. Es könnte lediglich das gesamte Bild rotiert werden, wodurch alle Zeichnungen rotieren würden. Dies widerspricht jedoch der zuvor genannten Zielsetzung.

Ob die Integration dieser Arbeit und die des Projektes kombiniert werden könnten, ist fraglich, da in der Dokumentation des XDPaint bestimmte Renderer vorausgesetzt werden. Jedoch hat nicht jeder der Drawable-Typen einen der vorausgesetzten Renderer. Zudem würde auch kein nennenswerter Mehrwert durch eine Kombination entstehen.

Bei dem gewählten Konzept dieser Arbeit ist leider kein Füllwerkzeug im eigentlichen Sinne möglich, bei dem beispielsweise weiße Flächen in einem Quadrat ausgefüllt werden können. Das liegt daran, dass das GameObject lediglich die Linien des Quadrates widerspiegelt und nicht die innere Fläche des Quadrates. Allerdings kann mittels der [Edit-Aktion](#) die vollständige Farbe des gezeichneten Quadrates verändert werden.

IMPLEMENTIERUNG

In diesem Kapitel wird die Implementation der im vorherigen Kapitel spezifizierten Integration beschrieben. Dazu wird zunächst auf die allgemeine **Funktionsweise** der Integration eingegangen. Anschließend wird der **Aufbau eines Drawables** verdeutlicht. Danach folgt die Beschreibung der **Konfigurationsklasse (ValueHolder)** für die Integration sowie die Erläuterung der verschiedenen **Drawable-Typen**. Abschließend werden die zahlreichen **Aktionen** und der **Drawable-Synchronizer** behandelt.

Aufgrund der Lesbarkeit wird an einigen Stellen der Begriff "Objekt" für GameObjects verwendet.

3.1 FUNKTIONSWEISE

Dieser Abschnitt behandelt die generelle Funktionsweise dieser Integration. Zur Erkennung der jeweiligen GameObjects wird das *Raycasting*-Verfahren verwendet. Bei dem Raycasting-Verfahren wird ein unsichtbarer Strahl in eine bestimmte Richtung gesendet und überprüft, ob der Strahl eine Kollision mit einem GameObject verursacht, das einen *Collider* besitzt (Seifert und Wislaug, 2017).

Ein Collider ist die Hauptkomponente zur Erkennung einer Kollision. Dadurch wird die Benutzerinteraktion mit dem jeweiligen Objekt ermöglicht.

SEE stellt für das Raycasting-Verfahren eine Klasse namens "Raycasting" mit verschiedenen Methoden bereit. In SEE wird der Strahl von einem Laser ummantelt, um so eine bessere Benutzbarkeit sowie Sichtbarkeit für die Selektierung zu bieten. Für diese Integration wird die Methode "RaycastAnything" der eben genannten Klasse benötigt. Mithilfe dieser Methode kann jedes 3D-Objekt mit einem Collider erkannt werden, sofern sich der Laser nicht auf einer *Graphical User Interface* (GUI)-Komponente befindet.

Von Unity werden vier verschiedene Collider bereitgestellt. Diese sind der *Box-Collider*, der *Sphere-Collider*, der *Capsule-Collider* und der *Mesh-Collider*. Die ersten drei werden als primitive Collider bezeichnet (Seifert und Wislaug, 2017). Diese verschiedenen Collider können in der nachfolgenden **Abbildung 3.1** von Seifert und Wislaug (2017) veranschaulicht werden. Die dargestellten Collider unterscheiden sich

GUI: Bezeichnet eine grafische Benutzeroberfläche, die eine Benutzerinteraktion ermöglicht.

alle in ihrer Form. Je komplexer die Form, desto mehr Kapazitäten werden für die Berechnung benötigt.

Aufgrund der Darstellung der Drawable-Typen kommen für diese Integration nur der Box-Collider und der Mesh-Collider in Frage. Mittels dem Box-Collider wird eine rechteckige Kollisionsform und mit dem Mesh-Collider eine *Mesh*-abhängige Kollisionsform hinzugefügt. Da diese Collider für 3D-Objekte gedacht sind und die Drawable-Typen zwar 3D-Objekte sind, aber als transformierte 2D-Objekte dargestellt werden, wird für die Tiefe des jeweiligen Colliders nur ein minimaler Wert gewählt. Welcher Collider für welchen Drawable-Typ gewählt und wie dieser transformiert wurde, wird in dem entsprechenden Drawable-Typ-Abschnitt beschrieben.

Mesh: Ist die Struktur eines Modells. Es verbindet die einzelnen Vektoren (Vertices) aus denen das Modell besteht, um eine Objektform zu bilden (Seifert und Wislaug, 2017).

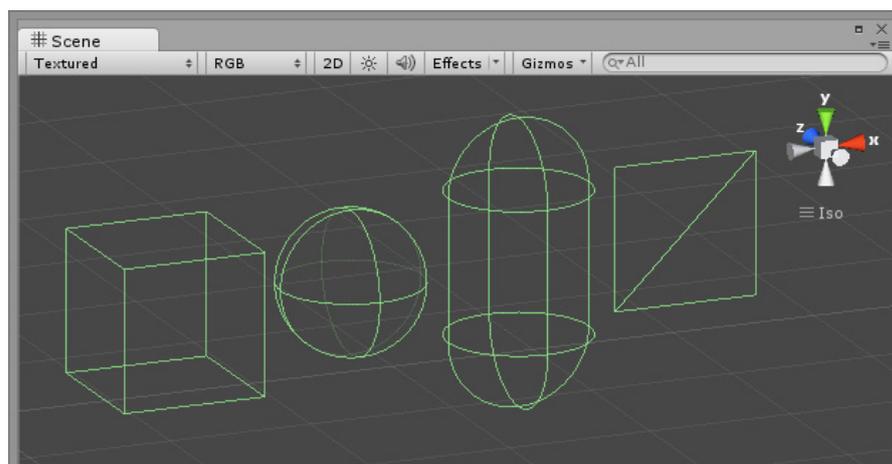


Abbildung 3.1: Überblick der Unity Collider von Seifert und Wislaug (2017).
Von links nach rechts: Box-, Sphere-, Capsule- und Mesh-Collider.

Zur Unterscheidung der jeweiligen zu erkennenden Objekte wird auf die von Unity bereitgestellte *Tag*-Funktionalität zurückgegriffen. Mittels des Tags wird dem jeweiligen GameObject ein Label in Form einer Beschriftung beziehungsweise Etiketts zugewiesen, das jederzeit abgefragt werden kann. Die Tags dienen somit zur Identifizierung der Art des Objektes während der Visualisierungslaufzeit (Seifert und Wislaug, 2017).

Zum Hinzufügen eines Tags muss im Unity-Editor ein GameObject ausgewählt und anschließend im Inspektor mittels des Tag-Dropdown-Buttons die Funktionalität "Add Tag..." ausgewählt werden. Anschließend wird mittels des "+"-Buttons ein neuer Tag mit dem gewünschten Namen angelegt. Zudem muss dieser auch in der "Tags"-Datei von SEE hinterlegt werden. In den nachfolgenden Abschnitten wird immer darauf hingewiesen, welche Tags hinzugefügt wurden und wofür sie benötigt werden.

3.2 DRAWABLE-AUFBAU

In diesem Abschnitt wird zunächst auf den Aufbau eines Drawables eingegangen. Anschließend werden die zwei [Varianten eines Drawables](#) vorgestellt. Abschließend wird der "Ablageort" für die Drawable-Typen auf einem Drawable, die [Funktionalität der Seitenränder](#) eines Drawables sowie die Drawable-[Konfigurationsklasse](#) beschrieben.

Damit ein Objekt als Drawable verwendet werden kann, benötigt es vier Konfigurationen, die in der folgenden nummerierten Liste vorgestellt werden.

1. Der "Drawable"- Tag muss dem Objekt zugewiesen werden. Dieser Tag wurde hinzugefügt, um Drawables zu identifizieren.
2. Das Objekt muss die richtige Rotation aufweisen. Die Rotation wird in Unity als 3D-Koordinatensystem verwendet ([Seifert und Wislaug, 2017](#)). Für das Objekt ist es wichtig, dass die Z-Achse die Tiefe widerspiegelt, sie muss folglich durch die Interaktionsfläche zeigen. Die X-Achse muss die Breite und die Y-Achse die Höhe des Objektes repräsentieren. In der folgenden [Abbildung 3.2](#) kann hierzu eine Demonstration begutachtet werden, wobei die Z-Achse durch den blauen Pfeil, die X-Achse durch den roten Pfeil und die Y-Achse durch den grünen Pfeil dargestellt wird.

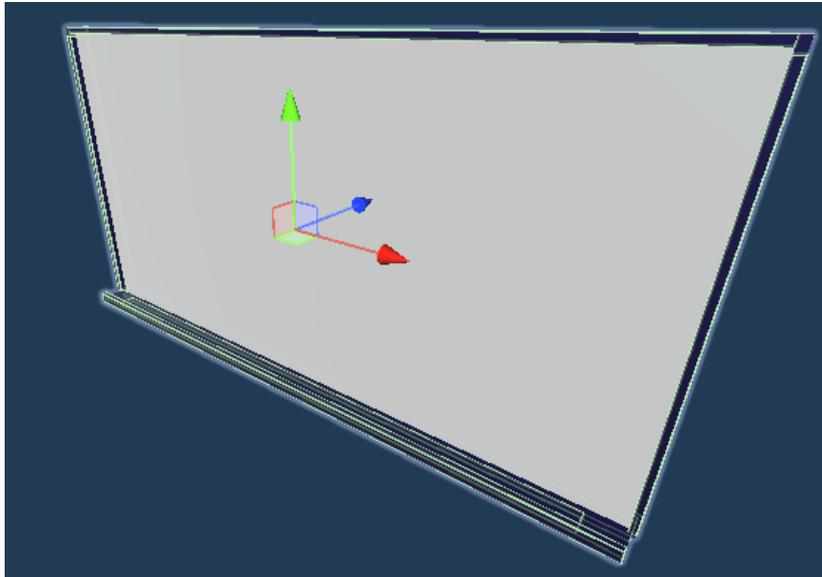


Abbildung 3.2: Demonstration der Rotation für ein Drawable.

3. Es wird ein eindeutiger Name für das Objekt benötigt, um eine eindeutige Zuordnung über das Netzwerk hinweg zu gewährleisten.

4. Mit dem Objekt muss interagiert werden können, dazu benötigt es einen entsprechenden Collider.

Falls das Drawable-Objekt aus mehreren Teilen besteht, wie es bei den Varianten dieser Integration der Fall ist, muss Punkt 2 und 3 auf dem obersten Objekt des *Prefabs* und Punkt 1 und 4 auf der Fläche, mit der interagiert werden soll, angewendet werden. Der Aufbau eines solchen mehrteiligen Drawables wird in den folgenden Unterabschnitten noch genauer erläutert.

Prefab: Ist eine Vorlage für GameObjects. Prefabs enthalten alle wichtigen Informationen für das Objekt, wie z. B. Materialien, Texturen, Collider, Kinderobjekte. Diese Prefabs können auch während der Laufzeit beliebig oft hinzugefügt werden (Seifert und Wislaug, 2017).

Wenn die zuvor beschriebenen vier Konfigurationen eingestellt worden sind, kann mit der frontalen Seite der Drawable-Fläche interagiert werden.

3.2.1 Drawable-Varianten

In diesem Abschnitt werden die zwei Drawable-Varianten die diese Integration bietet erläutert. Zum einen ist dies das **Whiteboard** und zum anderen das **Sticky-Note**. Beide Drawables haben jeweils ein übergeordnetes Elternobjekt, das Seitenränder bereitstellt.

3.2.1.1 Whiteboard

Das ursprüngliche Prefab für die Whiteboard-Variante wurde von einer früheren SEE-Version bereitgestellt. Die Objekthierarchie des modifizierten Whiteboards wird in der nachfolgenden **Abbildung 3.3** veranschaulicht.

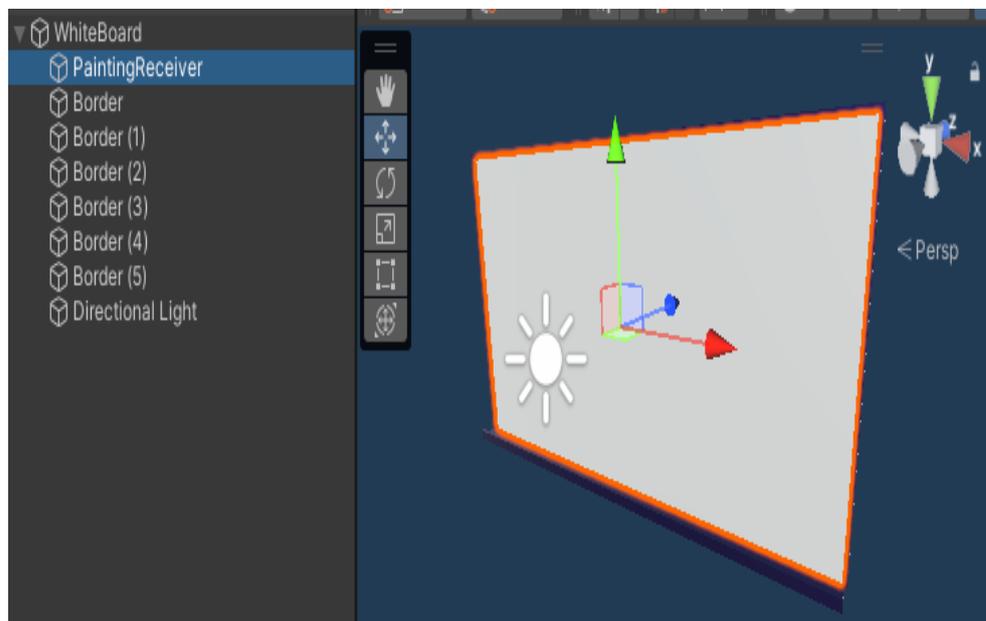


Abbildung 3.3: Die Objekthierarchie des Whiteboards.

Damit dieses Prefab als Drawable verwendet werden kann, wurde es wie zuvor beschrieben modifiziert. Dafür wurde dem Objekt "PaintingReceiver", das die weiße Fläche des Whiteboards repräsentiert, der "Drawable"-Tag zugewiesen (siehe [Abbildung 3.4](#)). Das "PaintingReceiver"-Objekt hatte bereits einen Mesh-Collider, weshalb kein anderer Collider hinzugefügt werden musste. Dadurch, dass nur dem "PaintingReceiver"-Objekt den "Drawable"-Tag zugewiesen wird, wird verhindert, dass auf den Seitenrändern des Whiteboards ein Drawable-Typ hinzugefügt werden kann.

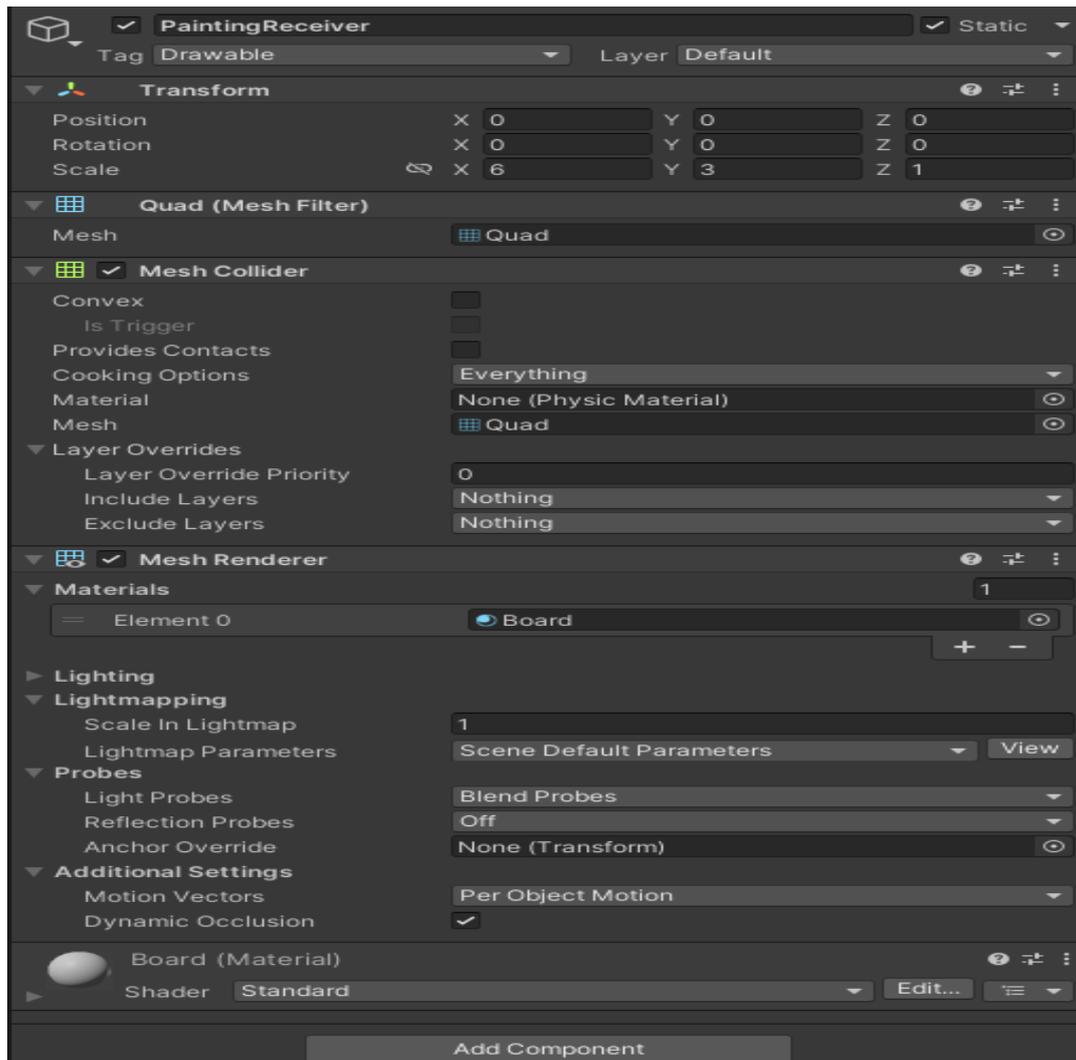


Abbildung 3.4: Inspektor des PaintingReceiver-Objektes.

Des Weiteren wurde das Prefab um eine direkte Lichtquelle erweitert, die auf das Whiteboard gerichtet ist. Dadurch wird sichergestellt, dass die Drawable-Typen gut beleuchtet und somit besser erkennbar werden.

Zudem wurden die Seitenränder um einen Controller zur Erkennung und Behandlung von Kollisionen mit einem Drawable-Typ erweitert.

Die dafür notwendigen Collider der Seitenränder waren bereits in dem Prefab integriert. Dies betrifft die Seitenränderobjekte Border, Border (1), Border (2) und Border (3) der in der [Abbildung 3.3](#) gezeigten Hierarchie. Border (4) und Border (5) sind die Elemente der Stiftablage des Whiteboards und werden daher für die [Seitenränder-Funktionalität](#) nicht benötigt. Der hinzugefügte Controller wird später im Abschnitt ["Seitenränder-Funktionalität"](#) näher erläutert.

3.2.1.2 Sticky-Note

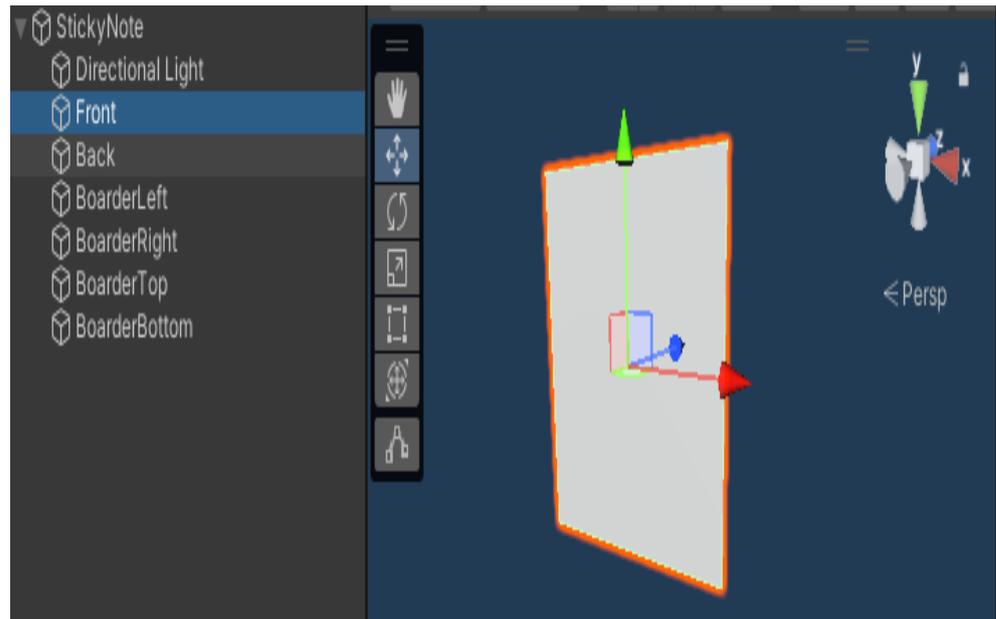


Abbildung 3.5: Die Objekthierarchie des Sticky-Notes.

Das Prefab für die Sticky-Notes wurde bewusst schlicht gehalten. Es ist ein Objekt mit zwei flachen Seiten, einer Vorder- und einer Rückseite. Beide Seiten verfügen über einen Box-Collider, jedoch können nur auf der Vorderseite Drawable-Typen hinzugefügt und mit ihnen interagiert werden. Aus diesem Grund wurde auch nur für die Vorderseite eine zusätzliche Lichtquelle zur Beleuchtung integriert. Der Collider für die Rückseite wird lediglich benötigt, damit der Sticky-Note zum Bewegen, Editieren oder Löschen auch von der Rückseite ausgewählt werden kann.

Das Objekt der Rückseite wird außerdem aufgrund der Vorgehensweise, wie ein Drawable-Typ hinzugefügt wird, benötigt. Diese Vorgehensweise wird in dem späteren Abschnitt ["Hinzufüge-Aktionen"](#) genauer behandelt.

Auf der Rückseite können keine Drawable-Typen hinzugefügt werden, da Sticky-Notes dafür konzipiert sind, mit der Rückseite auf einer Oberfläche zu haften. Darüber hinaus wäre es derzeit auch nicht möglich,

auf der Rückseite Drawable-Typen hinzuzufügen, da nur ein einzelner “Drawable“-Tag für das gesamte Objekt unterstützt wird.

Des Weiteren wurden vier Seitenränder hinzugefügt, die von der Vorderseite aus gesehen werden können, wie in der [Abbildung 3.5](#) zu sehen ist (orangene Linien). Diese sind als `BorderLeft`, `BorderRight`, `BorderTop` und `BorderBottom` bezeichnet. Wie auch bei dem [Whiteboard](#) wurden diese Seitenränder um die [Seitenränder-Funktionalität](#) erweitert.

3.2.2 AttachedObject-Objekt

In einer früheren Version der Integration wurden die Drawable-Typen als Kinderobjekte der Drawables-Fläche hinzugefügt. Für die Whiteboard-Variante war dies als Kinder des “PaintingReceiver“-Objektes ([Abbildung 3.3](#)).

Dadurch entstand das Problem, dass die Drawable-Typen verzerrt wurden, wenn die Drawable-Fläche skaliert ist. Dies ist für den “PaintingReceiver“, wie in der [Abbildung 3.4](#) zu sehen ist, der Fall.

[Seifert und Wislaug \(2017\)](#) beschreiben dieses Konzept als Parenting und erwähnen genau diesen Fall, dass beim Skalieren des Elternobjektes auch die Kinderobjekte mitskaliert werden. Beim Parenting gilt dasselbe auch für das Bewegen und Rotieren der Objekte dieser Eltern-Kind-Beziehung.

Aufgrund dieses Konzeptes und der Tatsache, dass das Elternobjekt “Whiteboard“ des “PaintingReceivers“ ebenfalls skaliert ist, mussten den vollständigen Drawable-Objekten jeweils ein neues oberstes Elternobjekt (Root-Objekt) zugewiesen werden, sobald ein Drawable-Typ auf ihnen hinzugefügt wird.

Im Rahmen dieser Integration wurde dieses Objekt als *Drawable-Holder* bezeichnet. Dem Drawable-Holder wird zum einen das Drawable sowie ein sogenanntes *AttachedObject-Objekt* als Kind zugewiesen. Dem AttachedObject-Objekt werden die einzelnen Drawable-Typen als Kinder hinzugefügt, es bildet somit den “Ablageort“ für diese Typen. Des Weiteren wird dem AttachedObject-Objekt der Tag “AttachedObjects“ zur Identifizierung zugewiesen.

Wichtig hierbei ist zu beachten, dass weder der Drawable-Holder noch das AttachedObject-Objekt skaliert sein dürfen, da sonst die Kinderobjekte wie oben beschrieben verzerrt werden würden. Nicht skaliert bedeutet in Unity, dass die Objekte den Standard-Skalierungsvektor von (1, 1, 1) aufweisen. Diese Vektorenschreibweise der Skalierung repräsentiert in Unity dabei die Skalierung entlang der x-, y- und z-Achse (x, y, z). 3D-Vektoren werden in Unity durch die Klasse “Vector3“ repräsentiert und werden neben der Skalierung unter anderem auch für die Positionierung und Rotierung benötigt.

Für die eben beschriebene Transformation eines Drawables wurde die Klasse “DrawableHolder” integriert. Diese erstellt das neue oberste Elternobjekt (Root-Objekt), im Code als “highestParent” bezeichnet, und das AttachedObject-Objekt für das Drawable, sobald ein Drawable-Typ hinzugefügt wird.

Jedes Mal, wenn ein Typ hinzugefügt wird, wird überprüft, ob der Drawable bereits transformiert wurde. Wenn dies nicht der Fall ist, wird die Transformation vollzogen, anderenfalls wird sie nicht erneut durchgeführt.

In der nachfolgenden [Abbildung 3.6](#) wird die Hierarchieveränderung einer solchen Transformation veranschaulicht.

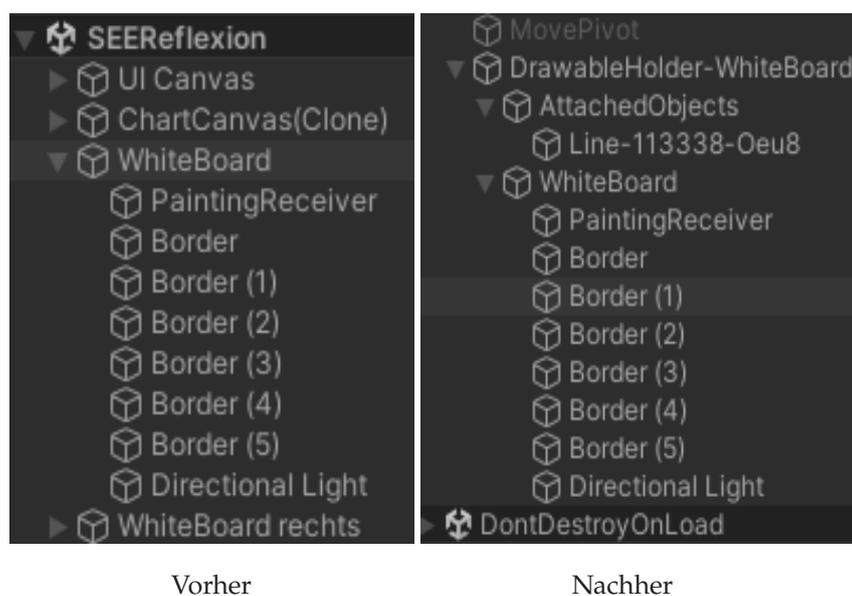


Abbildung 3.6: Transformation eines Drawables.

OnTriggerEvent: Ist ein Event, das ausgelöst wird, wenn ein anderes GameObject mit dem Collider-Bereich des jeweiligen Objektes interagiert. Diese Events können auftreten, wenn das GameObject in den Trigger-Bereich eintritt (OnTriggerEnter), ihn verlässt (OnTriggerExit) oder sich darin aufhält (OnTriggerStay) (Seifert und Wislaug, 2017).

3.2.3 Drawable-Seitenränder-Funktionalität

Die Seitenränder-Funktionalität (im Code “BorderTriggerController”) stellt sicher, dass die Drawable-Typen das Drawable nicht verlassen können. Für diese Funktionalität wurden die Tags “Top”, “Bottom”, “Left” und “Right” hinzugefügt und den entsprechenden Seitenränder zugewiesen. Zudem ist es für die Kollisionserkennung wichtig, dass jeder Box-Collider der Seitenränder die Eigenschaft “isTrigger” aktiviert hat, damit “OnTriggerEvents” verwendet werden können.

Wenn einer dieser Seitenränder eine Kollision mit einem Drawable-Typ-Objekt registriert, das zum selben Drawable gehört wie der Seitenrand, wird das Objekt des Drawable-Typen in die entgegengesetzte Richtung verschoben. Das bedeutet, dass beispielsweise bei einer Kollision mit dem linken Seitenrand das entsprechende Kollisionsobjekt nach rechts verschoben wird, um zu verhindern, dass es den Drawable-Bereich

verlässt.

Die Kollisionserkennung wird im späteren [gleichnamigen Abschnitt](#) näher erläutert. Eine Besonderheit für diese Funktionalität wird für den [Mind Map-Drawable-Typ](#) geboten, da auch die Kinderknoten für die Erkennung miteinbezogen werden können. Wie diese Besonderheit der Erkennung aktiviert werden kann, wird im Abschnitt [“Move or Rotate-Aktion“](#) beschrieben.

3.2.4 Drawable-Konfiguration

Für einige Aktionen werden verschiedene Daten eines Drawables benötigt. Um diese kompakt darstellen zu können, wurde eine Konfigurationsklasse namens `DrawableConfig` bereitgestellt. Damit Instanzen dieser Klasse über das Netzwerk versendet werden können, muss die Klasse das Schlüsselwort `Serializable` verwenden. Diese Klasse umfasst die Daten des Namens (ID) des Drawables, den Namen des Elternobjektes des Drawables, die Position, die Rotation, die Skalierung sowie verschiedene Listen der Konfigurationen der Drawable-Typen. Diese Konfigurationen werden später im Abschnitt der [Drawable-Typen](#) beschrieben.

Des Weiteren werden unter anderem die zwei folgenden Methoden bereitgestellt:

- [getDrawable\(\)](#): Sucht und liefert als Ergebnis das `GameObject` des Drawables der Konfiguration.
- [getAllDrawableTypes\(\)](#): Liefert als Liste alle Drawable-Typ-Konfigurationen des Drawables.

Weitere Methoden dieser Klasse werden im Abschnitt der [Save-Aktion](#) und [Load-Aktion](#) beschrieben.

Des Weiteren wird durch die Konfigurationsklasse `DrawableConfigs` eine Liste von `DrawableConfig`-Konfigurationen bereitgestellt. Auch diese muss serialisierbar sein, um über das Netzwerk verteilt werden zu können.

Mittels der statischen Klasse `DrawableConfigManager`, die für das Verwalten der Drawable-Konfiguration verantwortlich ist, wird sowohl eine `getDrawableConfig` als auch eine `getDrawableConfigs`-Methode bereitgestellt, mit der die Konfigurationen eines oder mehrerer Drawables erzeugt werden können. Das Erzeugen der Konfiguration umfasst das Sammeln der Drawable-Daten (Name, Elternname, Position, Rotation, Skalierung, Farbe und Ebenenreihenfolge) sowie das Erstellen von Listen der dazugehörigen Drawable-Typ-Objekte in Form von Konfigurationen. Diese Drawable-Typ-Konfigurationen werden später im Abschnitt [“Drawable-Typen“](#) beschrieben. Für das Erstellen der

Mind Map-Knoten Liste wird explizit darauf geachtet, dass die Knoten aufsteigend nach ihrer Mind Map-Ebene sortiert sind. Die Erklärung für diese Mind Map-Ebene erfolgt später im Kapitel [“Mind Map-Drawable-Typ”](#). Auf weitere Funktionalitäten dieser Klasse wird ebenfalls im Abschnitt der [Save-Aktion](#) sowie der [Load-Aktion](#) eingegangen.

3.3 VALUEHOLDER

Der *Value-Holder* (im Code `“ValueHolder”`) bildet die Konfigurationsinstanz dieser Integration. Darin werden alle notwendigen Konfigurationen für die Durchführung gespeichert. Hierfür wurde eine statische Klasse mit ausschließlich statischen Attributen verwendet.

Neben den durch ein Menü einstellbaren Werten wie Primärfarbe, Sekundärfarbe, Liniendicke, Textdicke etc. werden darin auch Konstanten, wie beispielsweise die Präfixe der Drawable-Typen, der Mindestabstand für die verschiedenen Ebenen oder die Geschwindigkeiten für die [Move-Aktion](#) gespeichert.

Die Instanz speichert über den eben genannten Daten noch viele weitere. Jedoch wird an dieser Stelle auf eine ausführliche Aufzählung verzichtet, da im späteren Verlauf dieses Kapitels immer wieder auf den Value-Holder in Bezug auf verwendete Daten verwiesen wird.

Der Vorteil dieser zentralen Speichervorgehensweise der Konfigurations- und Konstantenwerte besteht darin, dass hierdurch ein einfacheres Verwalten sowie Aktualisieren der Daten ermöglicht wird.

3.4 DRAWABLE-TYPEN

Im Folgenden werden die vier verschiedenen Drawable-Typen dieser Integration beschrieben.

Bei ihrer Erstellung erhalten alle dieser Typen eine *MonoBehaviour*-Komponente namens `“OrderInLayerValueHolder”`. Diese Komponente speichert die aktuelle Ebenenreihenfolge (Order in Layer) für das Objekt des Drawable-Typs.

Des Weiteren wird diese Komponente auch für Sticky-Notes genutzt, um deren Ebenenreihenfolge zu speichern. Zusätzlich bietet die Komponente für Sticky-Notes die Speicherung der originalen Position. Diese wird für das Wechseln der Ebenenreihenfolge benötigt und wird in der Erläuterung der [Change The Sorting Layer-Aktion](#) beschrieben.

Für die Drawable-Typen ist es wichtig, dass sie alle einen eindeutigen Namen besitzen, da sie sonst überschrieben werden könnten.

Dafür stellt der Value-Holder, wie erwähnt, Präfixe für die Drawable-Typen bereit. Diese sind `“Line”`, `“Text”`, `“Image”`, `“Theme”`, `“Subtheme”`, `“Leaf”` sowie `“BranchLine”`.

Um einen eindeutigen Namen für das jeweilige Drawable-Typ-Objekt

MonoBehaviour:
Dient als Basisklasse aller Skript-Klassen, sie wird als Komponente genutzt und kann einem `GameObject` angehängt werden. Sie stellt unter anderem Methoden wie `Awake`, `Start` und `Update` bereit. Diese funktionieren wie die im Kapitel [Anforderungen und Ziele](#) beschriebenen Methoden. Darüber hinaus können mit diesen Komponenten auch Physik-Methoden wie z. B. `“OnTriggerEnter”` bereitgestellt werden (Seifert und Wislaug, 2017).

zu generieren, wird der Präfix mit der Objekt-ID des Drawable-Typ-Objektes und einem zufällig generierten vierstelligen String durch Bindestriche verbunden, wie in der [Nachher-Abbildung 3.6](#) mit der Linie "Line-113338-Oeu8" zu sehen war. Die Zahl "113338" ist hierbei die Objekt-ID der Linie und "Oeu8" der generierte String.

Für die Generierung dieser zufälligen Strings stellt die Klasse des Drawable-Holder eine Methode namens "GetRandomString(int size)" bereit. Dabei kann der Parameter "size" die Anzahl der Zeichen festlegen. Der Generierungsalgorithmus bedient sich dabei der Zeichen "a-z" in Groß- und Kleinschreibung sowie der Zahlen "0-9" und einiger ausgewählter Sonderzeichen.

Des Weiteren haben alle vier Drawable-Typen neben den eindeutigen Namen (auch als ID bezeichnet) die folgenden gemeinsamen Eigenschaften:

Sie besitzen alle eine Position, eine Rotation in Form von "eulerAngles", eine Skalierung und eine Ebenenreihenfolge.

Die Position, Rotation und Skalierung werden dabei als Vektoren mit der zuvor beschriebenen Vektorenschreibweise (x, y, z) angegeben und werden von der *Transform*-Komponente bereitgestellt.

In der Integration werden diese Gemeinsamkeiten mittels der abstrakten Oberklasse "DrawableType" festgehalten. Von dieser Oberklassen erben die Konfigurationsklassen der vier Drawable-Typen. Diese Konfigurationsklassen sind erforderlich, um die *Polymorphie* in den [Aktionen](#) zu unterstützen.

Die Oberklasse "DrawableType" stellt darüber hinaus verschiedene Methoden bereit. Zwei davon werden im Folgenden kurz vorgestellt, die Übrigen werden später erläutert.

Zum einen ist dies die Methode "DrawableType Get(GameObject obj)", mit der für ein gegebenes Objekt die Konfiguration der Drawable-Typ-Spezialisierung ermittelt werden kann. Dafür bietet jeder Typ eine Methode, die eine Konfiguration des jeweiligen Drawable-Typs erstellt, sofern das Objekt von diesem Drawable-Typ ist.

Zum anderen ist dies die Methode "void Restore(DrawableType type, GameObject drawable)", die zum Wiederherstellen von Objekten verwendet wird. Mit ihr wird die entsprechende Wiederherstellungsmethode des jeweiligen Drawable-Typs für ein gegebenes Typ aufgerufen.

Damit die Instanzen der Konfigurationsklassen über das Netzwerk übertragen werden können, müssen die Klassen das Schlüsselwort "Serializable" aufweisen, um sie serialisierbar zu machen. Darüber hinaus werden Instanzen der Konfigurationsklassen für das verwendete *Memento*-Entwurfsmuster der Zustandsspeicherung sowie -wiederherstellung für die Undo- und Redo-Funktionalität benötigt.

Transform: Ist die grundlegende Komponente für *GameObjects*. Sie stellt die Position, die Rotation und die Skalierung für das Objekt bereit (Seifert und Wislaug, 2017).

Polymorphie: Sorgt dafür, dass Objekte, die eine gemeinsame Oberklasse haben, gleich behandelt werden können (Seifert und Wislaug, 2017).

Memento: Ist ein Entwurfsmuster zum Erfassen des Zustandes eines Objektes, um es zu speichern und wiederherstellen zu können (Pubudu, 2021).

Dafür enthält das Memento in SEE den vorherigen und den nachher Zustand einer Aktion. Im weiteren Verlauf des Kapitels wird auf diese Konfigurationsklassen näher eingegangen.

3.4.1 Line-Drawable-Typ

Die Grundkomponente für diesen Drawable-Typ ist der *LineRenderer*. Mithilfe dieser Komponente ist es möglich, Linien zu zeichnen. Der *LineRenderer* hat hierfür eine Liste von Punkten, die er miteinander verbindet. In dieser Arbeit werden diese Verbindungen zwischen den Punkten als Verbindungslinien bezeichnet.

Standardmäßig werden mit dem *LineRenderer* 3D-Linien gezeichnet, was für diese Integration nicht förderlich ist, da eine 2D-Darstellung benötigt wird. Daher wurde die Einstellung "Alignment", welche die Ausrichtung der Linie angibt, des *LineRenderers* auf "TransformZ" gesetzt. Dadurch wird die Z-Achse der Linie transformiert und flachgehalten.

Außerdem musste der *LineRenderer* auf lokale Positionen (*Local Space*) umgestellt werden. Standardmäßig arbeitet er mit globalen Positionen (*World Space*), dies führt jedoch dazu, dass die gezeichnete Linie weder verschoben noch rotiert werden kann. Um dieses Problem zu lösen, wurde die Option "Use World Space" deaktiviert.

Des Weiteren wurde für die Optik Linienenden mittels der Einstellung "End Cap Vertices" abgerundet.

Darüber hinaus wurde sichergestellt, dass die Linie eine durchgängig einheitliche Liniendicke (Thickness) aufweist, indem die Startdicke ("startWidth") und die Enddicke ("endWidth") gleich gesetzt wurden. Um eine gewählte Dicke einzustellen, wird diese der "startWidth"-Einstellung des *LineRenderers* zugewiesen.

Für die Loop-Funktion, welche den Start- und Endpunkt der Linie verbindet, bietet der *LineRenderer* eine Funktionalität, die über die Einstellung "Loop" aktiviert und deaktiviert werden kann. Standardmäßig ist diese deaktiviert.

Zudem wurden sechs Linienarten bereitgestellt, diese sind Solid, Dashed, Dashed25, Dashed50, Dashed75, Dashed100. Wenn "Dashed" gewählt wird, wird die Einstellung "Tiling" eingeblendet, mit der angegeben wird, wie oft das Liniensegment wiederholt werden soll. Je häufiger das Segment wiederholt wird, desto geringer ist der Abstand zwischen den Linienabschnitten.

Die genannten Linienarten können in zwei Grundarten unterteilt werden: Solid und Dashed. Jede dieser Arten erfordert ein eigenes Material, weshalb das "DrawableLineMaterial" für die Solid-Art und das "DrawableDashedLineMaterial" für die Dashed-Art angelegt wurden.

Der Hauptunterschied zwischen den beiden Materialien liegt im Rendering-Modus. Das Material für die Solid-Art nutzt den "Opaque"-

Local Space: Die Koordinaten sind abhängig von der Position und Rotation des Elternobjektes.

World Space: Die Koordinaten beziehen sich auf die globalen Koordinaten der 3D-Welt, unabhängig von der Position oder Rotation eines bestimmten Objektes.

Opaque: Ist ein Rendering Mode für undurchsichtige Objekte ohne transparente Bereiche (Unity_Technologies, 2024b).

Modus, während das Material für die Dashed-Art *“Cutout“*-Modus verwendet.

Zusätzlich zu den Rendering-Modi enthält das Material der Dashed-Art ein Bild des Brush, das wiederholt werden soll, sowie eine voreingestellte Tiling-Eigenschaft von 15. Die Verwendung dieser Eigenschaft wird im Folgenden erläutert.

Für die Dashed-Arten musste der *“Texture Scale“*-Wert des LineRenderers je nach Wiederholungsrate angepasst werden. Dieser Wert repräsentiert einen Multiplikator für die Tiling-Eigenschaft des Materials von Dashed-Linien, um die endgültige Tiling zu erhalten.

Im Falle der Linienart *“Dashed“* entspricht der Multiplikatorwert dem zuvor beschriebenen eingestellten *“Tiling“*-Wert.

Für *“Dashed25“* beträgt der Multiplikator $\frac{5}{3}$, der mit der Material-Tiling von 15 multipliziert wird. Daraus resultiert eine Tiling von 25. Entsprechend wird für die Arten *Dashed50*, *Dashed75* und *Dashed100* vorgegangen.

In der nachfolgenden [Abbildung 3.7](#) werden die sechs Linienarten veranschaulicht. Für die Dashed-Art sind dabei zwei Beispiele enthalten, wobei der Wert 10 den höchsten einstellbaren Wert für Dashed repräsentiert und 0,01 fast den kleinsten Wert darstellt. Auch der Wert 0 könnte gewählt werden, jedoch ist bei dieser Einstellung keine Kachelung zu sehen, wie in der [Abbildung 3.8](#) zu erkennen ist. Hierfür wurden Bilder aus der *“Scene“*- und der *“Game“*-View gemacht. Mit der *“Scene“*-View können z. B. die Collider betrachtet werden, während die *“Game“*-View die Visualisierung so darstellt, wie die Benutzer es sehen werden.

Da der Unterschied zwischen Dashed mit einer eingestellten Tiling von 10 und *Dashed100* in der [Abbildung 3.7](#) nicht so deutlich sichtbar ist, wird dieser Unterschied in der [Abbildung 3.9](#) dargestellt.

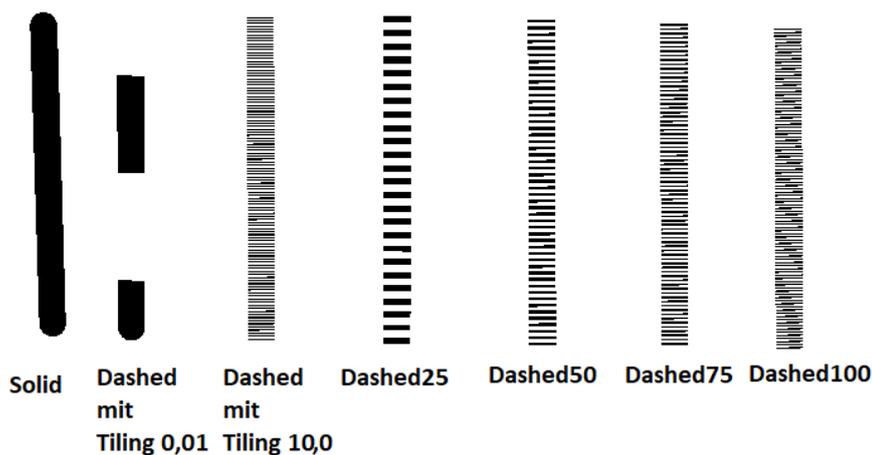


Abbildung 3.7: Die verschiedenen Linienarten.

Cutout: Ist ein Rendering Mode, der es ermöglicht, transparente Effekte mit klaren Kanten zwischen undurchsichtigen und transparenten Bereichen darzustellen (Unity_Technologies, 2024b).

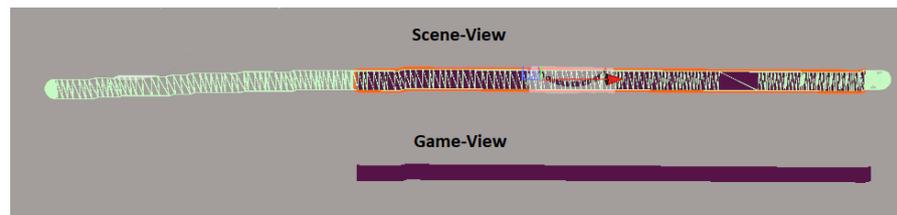


Abbildung 3.8: Dashed-Linienart mit einer eingestellten Tiling von 0.

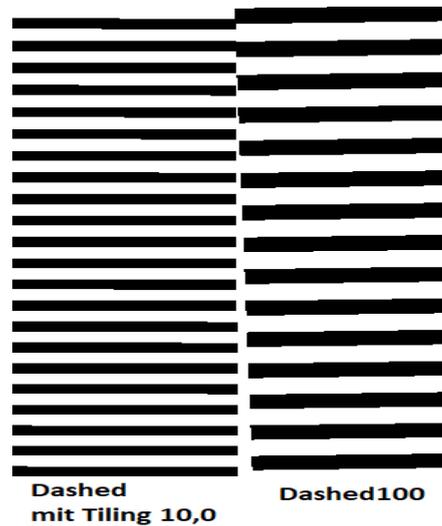


Abbildung 3.9: Vergleich der Dashed-Linienart mit einer eingestellten Tiling von 10 und der Dashed100-Linienart.

Des Weiteren wurden drei verschiedene Farbarten zur Verfügung gestellt: Monochrome, Gradient und TwoDashed.

Monochrome stellt eine einfarbige Linie dar, während Gradient den Farbverlauf zwischen zwei ausgewählten Farben darstellt. Für die Monochrome-Farbart wird die Farbe des Materials des Linienobjektes eingefärbt. Bei Gradient wird die Funktionalität des Farbverlaufs des LineRenderers verwendet und die Farbe des Materials weiß eingefärbt. TwoDashed funktioniert ausschließlich für die Dashed-Linienart und stellt die gestrichelte Linie abwechselnd in zwei Farben dar. Dabei wird der eigentlich transparente Bereich der Linie durch die zweite Farbe ausgefüllt. Dies geschieht über ein zweites Material, das der Linie zugewiesen wird.

In der nachfolgenden [Abbildung 3.10](#) werden die verschiedenen Farbvarianten anhand von Solid- und Dashed-Linien veranschaulicht.

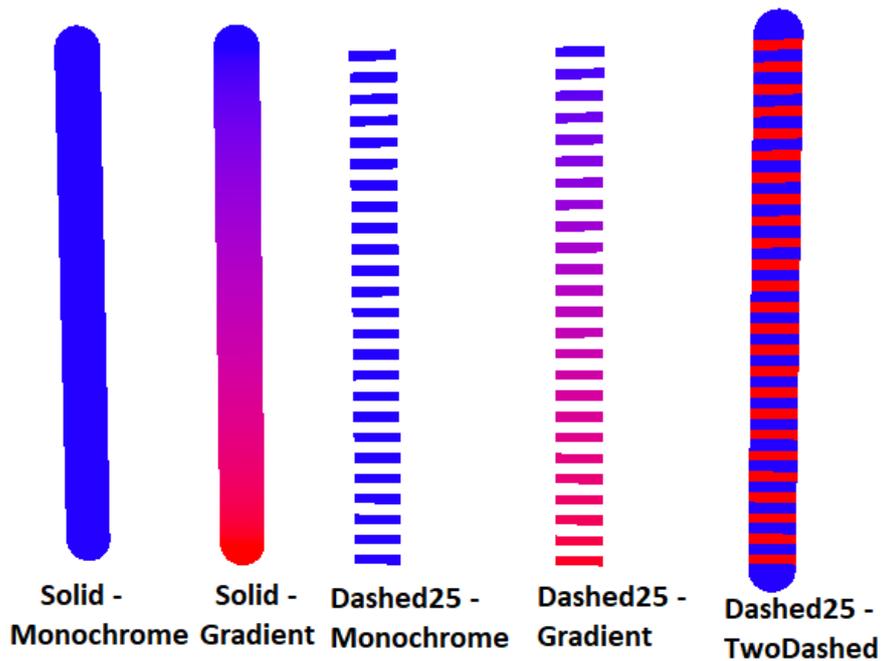


Abbildung 3.10: Die verschiedenen Farbvarianten anhand von Solid- und Dashed-Linien. Die gewählte Primärfarbe ist blau und die Sekundärfarbe rot.

Mittels der MonoBehaviour-Komponente "LinieValueHolder", die einer Linie bei ihrer Erstellung zugewiesen wird, wird die derzeit gewählte Linien- und Farbart der Linie gespeichert, um sie während der Laufzeit abfragen zu können.

Um ein Line-Drawable-Typ-Objekt während der Laufzeit identifizieren zu können, wurde der Tag "Line" hinzugefügt.

Aufgrund der Tatsache, dass Linien nicht immer gerade sind, passt keiner der primitiven Collider, weshalb der Mesh-Collider verwendet werden muss. In der nachfolgenden [Abbildung 3.11](#) kann der Mesh-Collider einer Spiralenlinie begutachtet werden.

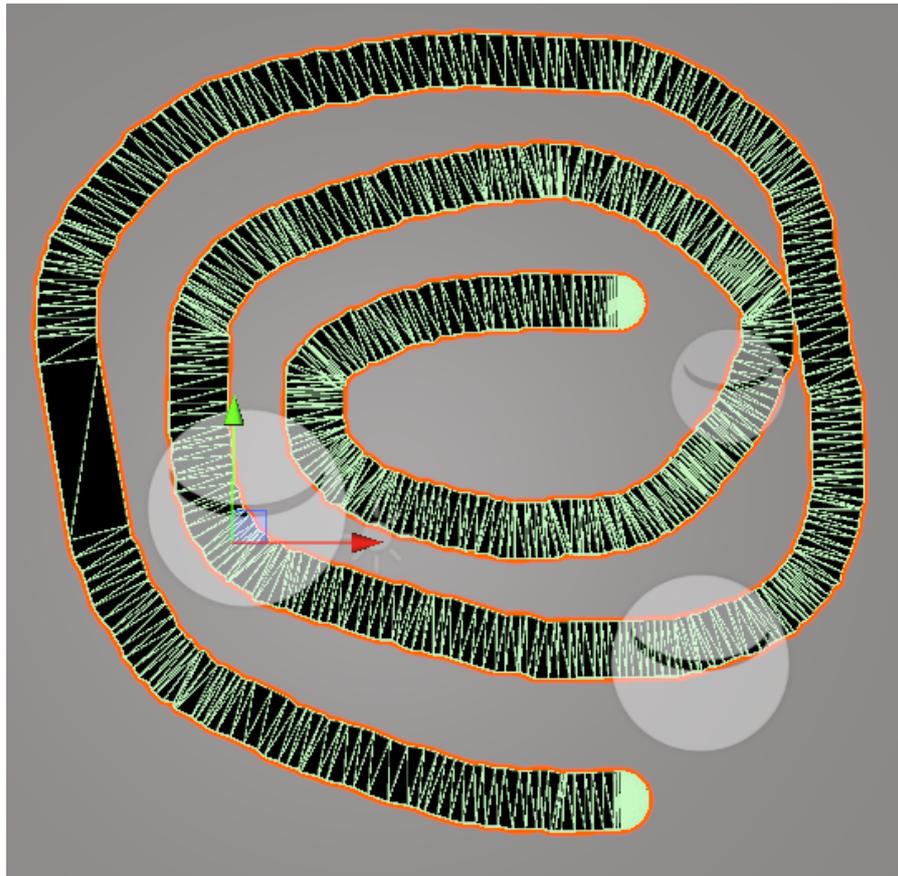


Abbildung 3.11: Der Mesh-Collider einer Spiralenlinie (Scene-View).

In dem späteren Abschnitt der [Draw Freehand-Aktion](#) wird erläutert, wie das freie Zeichnen bereitgestellt wurde, während im Abschnitt der [Draw Shape-Aktion](#) erklärt wird, wie und welche geometrische Formen gezeichnet werden können.

Zu diesem Drawable-Typ gehört die Konfigurationsklasse "LineConf". Diese Klasse umfasst sämtliche Eigenschaften der Linie. Zusammengefasst sind dies die Punkte der Linie, die verbunden wurden, die Loop-Eigenschaft, die Primär- sowie Sekundärfarbe, die Farbart, die Linienart, die Liniendicke und gegebenenfalls die Tiling-Rate.

3.4.2 Text-Drawable-Typ

Die Grundkomponente für diesen Drawable-Typ ist die [TextMeshPro \(TMP\)](#)-Komponente. Diese Komponente stellt einen Mesh-Renderer bereit, der die grafische Darstellung des Textes bereitstellt. Zudem wird durch die TMP-Komponente statt einer Transform-Komponente eine [RectTransform](#)-Komponente verwendet. Diese ist eine Erweiterung der Transform-Komponente und enthält zusätzliche Einstellungen für die Breite und Höhe des Objektes und legt somit die

Außenmaße fest (Seifert und Wislaug, 2017). Die Außenmaße bestimmen, wie breit der Text sein darf, bevor ein erzwungener Umbruch erfolgt.

Um die Außenmaße zu berechnen, wird die Breite des Textes anhand der Länge der ersten Zeile, der Schriftgröße, der Textstyles und der Schriftart approximiert. Dazu werden die *Rich-Text-Tags* aus der Zeile entfernt, wobei der Code dafür von Allen (2023) stammt.

Des Weiteren wurde untersucht, ob der Text die Textstyles Bold, Upper Case oder Small Caps aufweist, da diese die Breite des Textes beeinflussen. Die Idee und Teile der Berechnung stammen von Stephan_B (2019). Die Höhe der Außenmaße wird nun anhand der Anzahl der Zeilen, die durch die Breite entstehen, der Schriftgröße und einem kleinen Puffer berechnet. Die Schriftgröße wird über die TMP-Eigenschaft "fontSize" konfiguriert.

Für diesen Drawable-Typ wurde "DText" als Tag hinzugefügt, da der Tag "Text" bereits belegt war. Als Collider für diesen Drawable-Typ wurde der Mesh-Collider gewählt, um einen optimal passenden Collider bereitzustellen. Theoretisch hätte auch ein Box-Collider basierend auf den RectTransform-Außenmaßen gewählt werden können. Allerdings kann es hierbei zu dem Problem kommen, dass die Höhe der Außenmaße nicht der tatsächlichen Höhe des Textes entspricht. Dies kann der Fall sein, wenn der Text weitere Zeilenumbrüche "\n" enthält. Dadurch könnte ein Teil des Textes außerhalb des Box-Colliders stehen und wäre nicht interagierbar.

Eine Demonstration für diese beiden Collider-Arten kann in der nachfolgenden [Abbildung 3.12](#) begutachtet werden. Bei dem Box-Collider wurde mit der darunterstehenden Eingabe dafür gesorgt, dass die beschriebene Problematik entsteht. Die dunkelblaue Box wäre in diesem Beispiel die einzige interagierbare Fläche des Textes.

Aus diesem Grund wurde der Mesh-Collider verwendet, um dieses Problem zu umgehen.

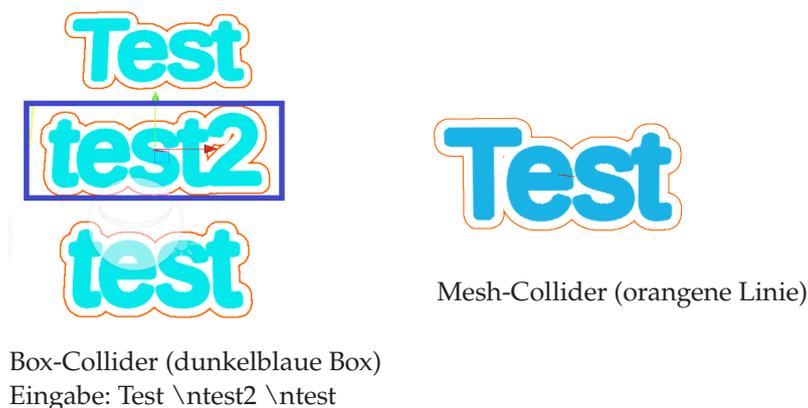


Abbildung 3.12: Demonstration der Collider für den Text-Drawable-Typ.

Shader: Ist ein Programm, das verwendet wird, um die Darstellung von 3D-Objekten oder 2D-Grafiken zu beeinflussen. Sie ermöglichen die Erzeugung von visuellen Effekten wie Licht und Schatten, Oberflächeneigenschaften, etc.

Als Font-Datei wird "DrawableTextFont" verwendet, die die Standard-TMP-Schriftart "LiberationSans" mit dem *Shader* "Distance Field" aus dem TextMeshPro-Mobile-Paket nutzt. Bei diesem Shader wird die Außenfarbe über das Schlüsselwort "OUTLINE_ON" ein- und ausgeschaltet. Wenn die Außenfarbe aktiviert wurde, kann mittels der TMP-Einstellung "outlineColor" die Außenfarbe gesetzt und zudem mittels "outlineWidth" die Dicke bestimmt werden.

Die Schriftfarbe wurde über zwei TMP-Einstellungen konfiguriert, zum einen über die "faceColor"-Einstellung und zum anderen über die "color"-Einstellung. Es wurden beide gewählt, um eine kräftige Farbe zu erhalten. Dadurch wird allerdings das Umfärben mit einem Rich-Text-Tag, wie beispielsweise "<color="green">...</color>", nicht unterstützt.

In der folgenden [Abbildung 3.13](#) kann ein Text mit einer schwarzen sowie einer mit einer blauen Schriftfarbe und ein Text mit einer roten Außenfarbe betrachtet werden.

The image displays three instances of the word "Test" in a bold, sans-serif font. The top-left instance is black with a thick red outline. The top-right instance is solid black. The bottom instance is solid blue.

Abbildung 3.13: Demonstration der verschiedenen Farbeinstellungen für Texte.

Außerdem werden mit TMP verschiedene Textstyles zur Verfügung gestellt. Diese umfassen Normal (keine Styles), Bold (fett), Italic (kursiv), Underline (unterstrichen), Strikethrough (durchgestrichen), Lower Case (Kleinbuchstaben), Upper Case (Großbuchstaben) und Small Caps (Darstellung als Großbuchstaben mit der Größe von Kleinbuchstaben). In der folgenden [Abbildung 3.14](#) werden die Textstyles dargestellt. Die Buchstaben in den eckigen Klammern entsprechen den Bezeichnungen der Buttons im später beschriebenen Text-Menü.

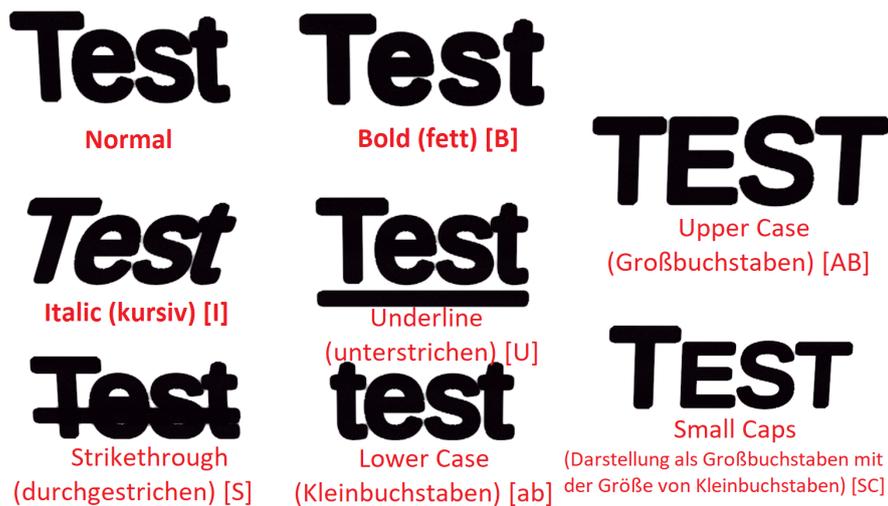


Abbildung 3.14: Demonstration der verschiedenen Textstyles.

Zusätzlich erlaubt TMP die Verwendung von Rich-Text-Tags. Das sind Tags ähnlich den HTML- oder XML-Tags, die das Aussehen und das Layout eines Textes verändern können (Unity_Technologies, 2023b). Die Verwendung eines solchen Rich-Text-Tags überschreibt die Eigenschaften des TMPs für den Bereich, den der Tag umfasst. [Abbildung 3.15](#) zeigt ein Beispiel für einen Text, der teilweise unterstrichen wird. Zusätzlich kann der eingegebene Text nachvollzogen werden.

Test Test Test

Eingabe: Test <u>Test</u> Test

Abbildung 3.15: Ein Beispiel für die Verwendung eines Rich-Text-Tag zum Unterstreichen.

Wie ein Text auf einem Drawable hinzugefügt werden kann, wird später im Abschnitt der [Write Text-Aktion](#) behandelt und anhand von Bildern verdeutlicht. Dabei wird auch auf das Konfigurieren der Eigenschaften eingegangen.

Die Klasse "TextConf" dient dem Text-Drawable-Typ als Konfigurationsklasse und umfasst sämtliche Eigenschaften für den Text-Drawable-Typ. Diese umfassen den geschriebenen Text, die Schriftfarbe, die Außenfarbe, den Status der Außenfarbe (aktiviert/deaktiviert), die Dicke der Außenfarbe, die Textstyles und die Schriftgröße.

3.4.3 Image-Drawable-Typ

Mit diesem Drawable-Typ wird ein Bild in der Visualisierung dargestellt. Dazu wurde zuerst der Tag "Image" zur Identifizierung der Objekte hinzugefügt.

Um ein Bild beziehungsweise eine Grafik in die Visualisierung einzufügen, wird die Image-Komponente von Unity benötigt. Mit ihrer Hilfe kann eine Grafik dargestellt werden, dafür muss die jeweilige Grafik jedoch als Sprite vorliegen. Ein Sprite ist eine 2D-Textur, die den Texture-Type "Sprite" besitzt (Seifert und Wislaug, 2017).

Beim Hinzufügen der Image-Komponente wird automatisch ein Canvas-Renderer hinzugefügt, der für die Darstellung des Bildes verantwortlich ist. Zudem wird eine RectTransform-Komponente hinzugefügt. Aufgrund der Tatsache, dass der Standardwert der Außenmaße mit dem Vektor von (100, 100) viel zu groß für die Visualisierung ist, wurden die Außenmaße auf den Vektor (1, 1) verkleinert.

Damit die Image-Komponente auf dem Drawable sichtbar wird, benötigt sie zusätzlich noch ein *Canvas*. Für das Canvas ist es laut Seifert und Wislaug (2017) wichtig, dass der Render Mode auf dem Standardwert "World Space" bleibt, damit das Canvas als ganz normales GameObject verwendet werden kann.

Aufgrund der rechteckigen Form des Canvas kann ein Box-Collider als Collider verwendet werden. Die Maße des Colliders entsprechen den Außenmaßen des RectTransforms für die Breite und Höhe, während für die Tiefe ein minimaler Wert von 0.01 gewählt wurde. Die [Abbildung 3.16](#) zeigt den Box-Collider für diesen Typ, wobei der Collider erneut orange eingefärbt ist.

Canvas: Ist ein rechteckiges grafisches Element, das als Container für verschiedene UI-Elemente dient.



Abbildung 3.16: Demonstration des Box-Colliders für den Image-Drawable-Typ. Rafay Ansari stellt dieses lizenzfreie Bild zur Verfügung¹.

¹ <https://unsplash.com/de/fotos/ein-marienkafer-sitzt-auf-einer-grunen-pflanze-prhMYojFJRY>

Die in der Anforderung gestellte Funktionalität der Farbüberlagerung wurde mittels der Eigenschaft "Color" der Image-Komponente umgesetzt. Mit dieser Eigenschaft kann das Bild eingefärbt werden. Ein Beispiel für diese Einfärbung ist in der folgenden [Abbildung 3.17](#) zu sehen. Hierzu wurde das von Rafay Ansari bereitgestellte Bild rötlich eingefärbt.

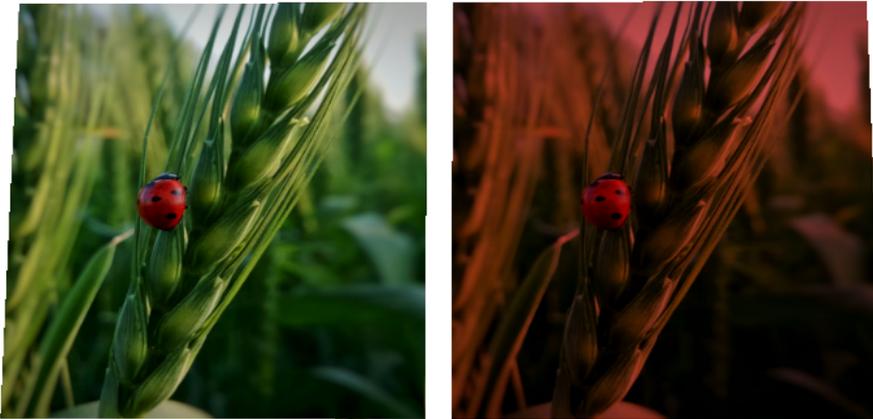


Abbildung 3.17: Demonstration der Einfärbung für den Image-Drawable-Typ.

Zu den bereits beschriebenen Komponenten wurde bei der Erstellung des Bildobjektes eine MonoBehaviour-Komponente namens "ImageValueHolder" hinzugefügt. Diese Komponente speichert die Werte für den Dateipfad und gegebenenfalls die Byte-Daten des jeweiligen Bildes.

Wie ein PNG- oder JPG-Bild hinzugefügt werden kann und welche Varianten dazu geboten werden, wird in dem Abschnitt "[Add an Image-Aktion](#)" erläutert. Zudem wird im Abschnitt der [Edit-Aktion](#) erklärt, wie ein Bild gespiegelt werden kann.

Die Konfigurationsklasse dieses Drawable-Typs ist die Klasse "ImageConf". Durch sie werden die Daten des Bildes festgehalten. Diese umfassen die Einfärbungsfarbe sowie den Dateipfad oder die Byte-Daten zu dem Bild.

3.4.4 Mind Map-Drawable-Typ

Dieser Drawable-Typ ist ein zusammengesetzter Typ, der aus dem [Line-Drawable-Typ](#) für den Knotenrand (Border) sowie der Verbindungslinie (BranchLine) zwischen zwei Knoten und den [Text-Drawable-Typ](#) für den Knotentext besteht.

Ein Mind Map-Knotenobjekt besteht aus einem Kinderobjekt für den Knotenrand und einem Kinderobjekt für den Text. Die dadurch entstehende Objekthierarchie ist in der folgenden [Abbildung 3.18](#) dargestellt.



Abbildung 3.18: Die Hierarchie eines Mind Map-Knotenobjektes.

Es stehen drei verschiedene Knotenarten zur Verfügung. Diese sind das zentrale Thema (Central Theme), das Unterthema (Subtheme) und das Blatt (Leaf). Jede dieser Knotenarten wird durch ein unterschiedliches Präfix gekennzeichnet. Darüber hinaus weisen alle drei Arten unterschiedliche Darstellungen auf, siehe [Abbildung 3.19](#). Es ist zu erkennen, dass verschiedene Knotenränder sowie Textstyles und Schriftgrößen verwendet werden. Die Arten des zentralen Themas und des Blattes verwenden einen ellipsenförmigen Knotenrand, wobei der des Blattes transparent und gestrichelt ist, wie aus der [Abbildung 3.20](#) ersichtlich ist. Lediglich die Unterthema-Art weist eine rechteckige Form auf. Dabei wird die Größe des jeweiligen Knotenrandes anhand der Rect-Transform-Außenmaße des Knotentextes unter Berücksichtigung eines kleinen Puffers berechnet. Zusätzlich stellt die Art des zentralen Themas den Text des Knotens fett und unterstrichen dar. Diese Darstellungsarten wurden gewählt, um die Knotenarten auf einen Blick unterscheiden zu können.



Abbildung 3.19: Die Knotenarten der Mind Map.



Abbildung 3.20: Die Blatt-Knotenart inklusive Box-Collider.

Um eine Mind Map zu erstellen, müssen die folgenden Regeln eingehalten werden:

1. Damit ein Unterthema- oder ein Blatt-Knoten hinzugefügt werden kann, muss mindestens ein zentrales Thema vorhanden sein.
2. Die Unterthema- und Blatt-Knoten haben immer einen Elternknoten.
3. Zentrale Themen und Unterthemen können beliebig viele Kinderknoten haben, während Blätter keine Kinderknoten haben können.

Die Eltern-Kind-Beziehung zwischen zwei Knoten wird durch eine Verbindungslinie dargestellt, welche den [Line-Drawable-Typ](#) verwendet. Aufgrund der Tatsache, dass diese Verbindungslinien immer zwei Knoten verbinden und keinem direkt untergeordnet sind, werden sie als eigenständige Objekt erstellt. Eine Besonderheit ist die Namensgebung einer solchen Linie. Sie erhält mit "BranchLine" ein anderes Präfix als herkömmliche Linien und erhält anstelle einer Objekt-ID und eines zufälligen Strings die IDs der beiden Knoten. Eine Beispielbenennung wäre demzufolge "BranchLine-111111-222222", wobei die erste ID für den Elternknoten und die zweite für den Kinderknoten steht, den die Linie verbindet.

Für diesen Typ wurde der Tag "MindMap" hinzugefügt, um die Objekte zu identifizieren. Als Collider wurde ein Box-Collider verwendet, der sich an der Breite und Höhe des Knotenrands orientiert und einen kleinen Puffer dazu rechnet. Die Verwendung dieses Colliders ist möglich, da die Ränder der Knotenarten eine vorhersehbare Form aufweisen und nicht davon abweichen. Zudem ermöglicht die Verwendung dieses Colliders, dass auch die leere Fläche zwischen dem Knotenrand und dem Text ausgewählt werden kann, um mit dem Knoten zu interagieren. Dies wäre bei einem Mesh-Collider nicht möglich.

Des Weiteren mussten die Mesh-Collider der Linien (Knotenrand und Verbindungslinien) sowie des Textes deaktiviert werden, um zu verhindern, dass mit diesen Teilkomponenten interagiert werden kann. Dies ist notwendig, um die Ausführung von Aktionen auf die Teilkomponenten zu verhindern, wie beispielsweise die [Line Point Erase-Aktion](#) auf den Knotenrand oder das Verschieben des Textes. Der Box-Collider eines Blattes konnte bereits in der [Abbildung 3.20](#) betrachtet werden, wobei der Collider orange dargestellt wurde. In der [Abbildung 3.21](#) können ebenfalls die Box-Collider des zentralen Themas sowie des Unterthemas in orange betrachtet werden.

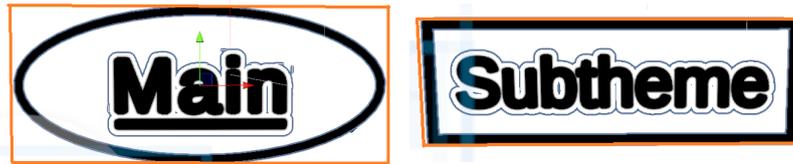


Abbildung 3.21: Box-Collider des zentralen Themas und des Unterthemas einer Mind Map.

Damit die Eigenschaften eines Mind Map-Knotens während der Laufzeit abgefragt werden können, wurde diesem bei seiner Erstellung eine `MonoBehaviour`-Komponente namens `MMNodeValueHolder` hinzugefügt. Diese Komponente speichert die für den Knoten gewählte Knotenart sowie gegebenenfalls den Eltern- und die Kinderknoten und eine Ebene (Layer), die nicht der zuvor beschriebenen Ebenenreihenfolge entspricht. Zu dem Elternknoten und den Kinderknoten werden die dazugehörigen Verbindungslinien gespeichert. Die Ebene dient der Reihenfolge, wie die Knoten hinzugefügt werden müssen, um sie wiederherstellen zu können. Zur Verdeutlichung: Es würde nicht funktionieren, zuerst den Kinderknoten und dann erst den Elternknoten zu erstellen, daher wird diese Ebene für die Sortierung der Knoten benötigt, die später in der [Save-Aktion](#) beschrieben wird.

Wie eine Mind Map erstellt werden kann, wird später im Abschnitt der [Mind Map-Aktion](#) beschrieben.

Die Konfigurationsklasse für diesen `Drawable`-Typ ist die Klasse `MindMapNodeConf`. Sie enthält alle Daten eines Mind Map-Knotens, darunter die Ebene zur Wiederherstellung, den Elternknoten und die Verbindungslinie zu diesem, die Knotenart, die Konfiguration der Knotenumrandung (als `LineConf`), die Konfiguration des Knotentextes (als `TextConf`), die Konfiguration der Verbindungslinie zum Elternknoten (`LineConf`) sowie ein *Dictionary* der Kinderknoten inklusive ihrer Verbindungslinien.

Dictionary: Ist eine Klasse, die Schlüssel-Wert-Paare speichert. Jeder Schlüssel darf dabei nur einmal vorkommen (Seifert und Wislaug, 2017). In anderen Programmiersprachen wird diese auch als Map oder HashMap bezeichnet.

3.5 OBJEKTE FINDEN

Um gezielt nach `Drawable`- und `Drawable`-Typ-Objekten während der Laufzeit zu suchen, wurden mit der statischen Klasse `GameFinder` verschiedene Methoden zur Objektsuche bereitgestellt.

Diese Methoden umfassen:

- **GameObject `FindDrawable`(string drawableID, string parentDrawableID):**
Durchsucht die Liste aller `Drawables` der Visualisierung nach einem `Drawable`, der mit den gegebenen Namen beziehungsweise IDs übereinstimmt. Es wurde auch die Möglichkeit integriert,

dass die Eltern-ID leer ist, jedoch wird dieser Fall mit den derzeitigen Drawables dieser Integration nicht genutzt.

Wenn kein passender Drawable gefunden wird, wird null zurückgeliefert.

- **GameObject FindChild(GameObject parent, string childName):**
Sucht nach einem Kindobjekt mit dem gegebenen Namen. Wenn das Elternobjekt ein Drawable ist, werden die Kinder des AttachedObject-Objekt durchsucht. In allen anderen Fällen werden alle Kinderobjekte des jeweiligen Elternobjektes durchsucht. Wenn kein übereinstimmendes Objekt gefunden wird, wird null zurückgeliefert.
- **GameObject GetDrawable(GameObject obj):**
Sucht das Drawable-Objekt zu einem gegebenen Objekt. Dazu wird vom obersten Elternobjekt nach einem Kind mit dem "Drawable"-Tag gesucht.
- **bool hasDrawable(GameObject child):**
Überprüft, ob das Objekt einem Drawable-Objekt zugeordnet werden kann. Diese Methode ist nur für Drawable-Typen gedacht, deshalb wird zur Überprüfung ein Elternobjekt mit einem "AttachedObjects"-Tag benötigt. Wenn kein solches Elternobjekt existiert, wird "false" zurückgeliefert.
- **bool IsPartOfADrawable(GameObject component):**
Ist eine boolesche Abfrage, ob das gegebene Objekt ein Teil eines Drawables ist. Hierfür wird kein Elternobjekt mit dem "AttachedObjects"-Tag benötigt.
- **GameObject FindChildWithTag(GameObject parent, string tag):**
Durchsucht die Kinderobjekte nach einem Objekt mit dem gegebenen Tag. Das zuerst gefundene Objekt mit diesem Tag wird dann zurückgeliefert. Wenn kein Kind mit dem Tag gefunden wird, wird null zurückgeliefert.
- **List<GameObject> FindAllChildrenWithTag(GameObject parent, string tag):**
Durchsucht die Kinderobjekte nach Objekten, deren Tag mit dem gesuchten übereinstimmt, und gibt alle gefundenen Objekte mittels einer Liste zurück.
- **List<GameObject> FindAllChildrenWithTagExceptParentHasTag(GameObject parent, string childTag, string parentTag):**
Durchsucht die Kinderobjekte nach Objekten, deren Tag mit dem gesuchten übereinstimmt, und gibt alle gefundenen Objekte mittels einer Liste zurück, es sei denn, ihr Elternobjekt weist den bestimmten Ausschluss-Tag auf.
Diese Methode wird nur von der Klasse "DrawableConfigMana-

ger“ benötigt, um bei der Erstellung der Listen für die Drawable-Typen die Knotenränder und Knotentexte der Mind Map-Knoten aus der Liste der Linien und Texte auszuschließen. Dies erfolgt, da Mind Map-Knoten in einer separaten Liste für Mind Map-Knoten aufgeführt werden und die Knotenränder beziehungsweise Knotentexte nicht alleinstehend existieren sollen, da diese Listen unter anderem für das Wiederherstellen der Drawable-Typ-Objekte von Drawables in der [Cleaner-Aktion](#) benötigt werden.

- **bool [hasChildWithTag](#)(GameObject parent, string tag):**
Überprüft, ob ein Kind mit dem gesuchten Tag existiert.
- **bool [hasAParent](#)(GameObject child):**
Überprüft, ob ein Elternobjekt existiert.
- **bool [hasAParentWithTag](#)(GameObject child, string tag):**
Überprüft, ob ein Elternobjekt mit dem gesuchten Tag existiert. Für diese Überprüfung werden alle Elternobjekte in der Objekthierarchie durchsucht, nicht nur das unmittelbar darüberliegende.
- **string [GetDrawableParentName](#)(GameObject drawable):**
Liefert den Namen (die ID) des Elternobjektes des Drawables, sofern ein Elternobjekt existiert. Anderenfalls wird ein leerer String zurückgeliefert. Im Fall des Whiteboards wäre dies “Whiteboard ...“, wobei “...” ein Platzhalter für eine ID ist.
- **GameObject [GetHighestParent](#)(GameObject child):**
Liefert das oberste Elternobjekt zu einem gegebenen Objekt. Bei Drawables und Drawable-Typen ist dies der Drawable-Holder.
- **GameObject [GetAttachedObjectsObject](#)(GameObject obj):**
Liefert das AttachedObject-Objekt zu einem gegebenen Objekt, sofern eins existiert, sonst null.

In den folgenden Abschnitten wird darauf eingegangen, wann diese Methoden verwendet werden.

3.6 AKTIONEN

In diesem Abschnitt werden die verschiedenen Aktionen erläutert, die durch diese Integration ermöglicht werden.

Dafür werden zunächst die Gemeinsamkeiten der angebotenen Aktionen erklärt. Darauf folgt eine Beschreibung der Verwendung sowie der Erweiterung des Color-Pickers im Abschnitt des [HSV-Color-Picker](#). Anschließend wird im Abschnitt der [Hinzufüge-Aktionen](#) beschrieben, wie die Objekte der Drawable-Typen dem Drawable hinzugefügt werden können. Daraufhin folgt die Erläuterung der [Color Picker-Aktion](#) sowie der [Transformations-Aktionen](#), gefolgt von den Abschnitten [Save-Aktion](#), [Load-Aktion](#) und [Lösch-Aktionen](#). Abschließend wird die

[Sticky-Note-Aktion](#) erklärt.

Um sicherzustellen, dass die bereitgestellten Aktionen den in der [Anforderungen und Ziele](#) geforderten Lebenszyklus (Ablauf) unterstützen, erben alle von der abstrakten Klasse "AbstractPlayerAction", die das Interface "IReversibleAction" implementiert. Neben dem Lebenszyklus wird den Aktionen dadurch zusätzlich die Schnittstellen für die Undo- und Redo-Funktionalität zur Verfügung gestellt. Außerdem ermöglicht diese Erbschaft die Angabe eines Aktionzustandes. Die möglichen Zustände sind dabei "NoEffect", "InProgress" und "Completed". Damit eine Aktion rückgängig gemacht und anschließend wiederholt werden kann, benötigt sie den Zustand "Completed".

Um zu gewährleisten, dass die getätigten Aktionen vollständig abgeschlossen sind, werden in den notwendigen Aktionen die getätigten Interaktionen rückgängig gemacht, sofern der Benutzer die Aktion nicht ordnungsgemäß beendet hat. In den folgenden Unterabschnitten wird das Vorgehen in den Aktionen beschrieben, in denen dies relevant ist.

Eine weitere Gemeinsamkeit der Aktionen ist, dass die getätigten Interaktionen über das Netzwerk verteilt werden müssen. Dazu bietet jede Aktion eine dazugehörige Netzwerkaktion, mit der Ausnahme von der [Color Picker-Aktion](#). Jeder Benutzer hat seinen eigenen Value-Holder und damit auch eine eigene zu wählende Primär- und Sekundärfarbe. Aus diesem Grund wird die Benutzung der [Color Picker-Aktion](#) nicht über das Netzwerk verteilt.

Abgesehen von einigen Funktionalitäten der [Sticky-Note-Aktion](#), benötigen alle anderen Aktionen in ihrer Netzwerkaktion die Methode "FindDrawable" der GameFinder-Klasse. Dadurch wird sichergestellt, dass die Aktion bei jedem Benutzer der Visualisierung auf demselben Drawable ausgeführt wird. Deshalb müssen den entsprechenden Netzwerkaktionen die Drawable-ID und die Eltern-ID des Drawables übergeben werden.

Die von den Aktionen bereitgestellten Menüs verwenden alle die von SEE zur Verfügung gestellten UI-Elemente, wodurch sie nach Belieben verschoben werden können und das gleiche Erscheinungsbild aufweisen.

Wenn ein Teil des Menüs ein- oder ausgeblendet wird, wird die Größe des Menüs mithilfe der statischen Methode "CalculateHeight" der Klasse "MenuHelper" neu berechnet und entsprechend angepasst. Nachdem das Menü angepasst wurde, wird es entsprechend verschoben, um die ursprüngliche Position zu erhalten. Dies geschieht, um Sprünge im Sinne von Verschiebungen zu verhindern.

Des Weiteren werden in einigen Aktionen zu den bereitgestellten Menüs noch zusätzliche Interaktionen durch Shortcuts geboten. Diese Shortcuts sind jedoch nur außerhalb des Menüs anwendbar. Das bedeutet, dass wenn sich der Benutzer mit dem Mauscursor beziehungsweise dem Raycasting-Laserstrahl auf einem GUI-Element befindet, stehen diese Shortcuts nicht zur Verfügung.

3.6.1 HSV-Color-Picker

Der im Abschnitt [“Anforderungen und Ziele“](#) beschriebene HSV-Color-Picker ist ein Hauptbestandteil der in den Aktionen bereitgestellten Menüs. Dieser kann in der nachfolgenden [Abbildung 3.22](#) betrachtet werden. Standardmäßig können lediglich elf Farbvoreinstellungen mittels eines Klicks auf den “+“-Button hinzugefügt werden, die nicht verändert werden können. Die hinzugefügte Farbe kann dann mittels eines linken Mausklicks auf der Farbbox gewählt werden.

Zudem können Farben über das Eingabefeld mittels eines Hexadezimalwertes und mittels den *Slidern* für die RGB-Werte eingestellt werden. Mittels des Sliders des A-Wertes kann die Transparenz der Farbe angegeben werden.

Slider: Ist eine Benutzeroberfläche mit einem Schieberegler, der zum Einstellen eines Wertes innerhalb eines bestimmten Wertebereichs benutzt wird. Zum Verändern des Wertes muss der Benutzer den Regler verschieben.

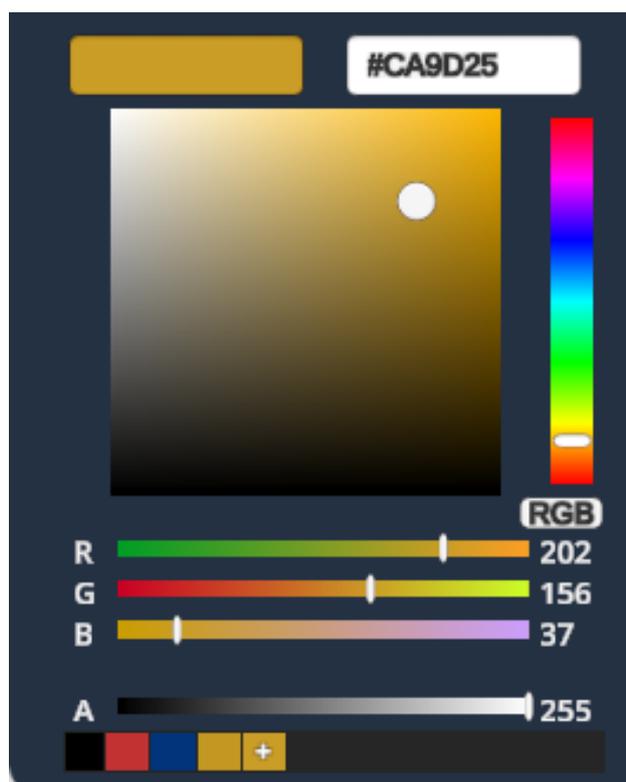


Abbildung 3.22: Darstellung des HSV-Color-Pickers.

Für diese Integration wurde der HSV-Color-Picker um die Funktionen des Änderns und Löschens von hinzugefügten Farben erweitert. Dazu

kann der Benutzer mittels eines mittleren Mausklicks (Mausradklicks) die Farbe ändern und mittels eines rechten Klicks die Farbe löschen. Außerdem wurde das Eingabefeld so verändert, dass die Shortcuts von SEE während der Eingabe deaktiviert werden. Zudem werden die konfigurierten Farben in einer Konfigurationsdatei namens "ColorPresets" gespeichert, sodass sie auch für die nächste Sitzung der Visualisierung zur Verfügung stehen. Hierzu wird eine Konfigurationsklasse namens "ColorsConfig" verwendet. Die Funktionsweise des Speicherns und Laden von Konfigurationsklassen wird in den Abschnitten der [Save-Aktion](#) und [Load-Aktion](#) behandelt. Die Speicherung dieser Konfiguration erfolgt nach jeder Änderung an den konfigurierten Farben sowie beim Anlegen einer neuen Farbe.

3.6.2 Hinzufüge-Aktionen

Dieser Abschnitt behandelt das Hinzufügen der Objekte der Drawable-Typen. Hierzu wird zunächst auf die Gemeinsamkeiten der Hinzufügeaktionen eingegangen. Anschließend wird das freie Zeichnen in der [Draw Freehand-Aktion](#) und das Zeichnen von geraden Linien sowie das Hinzufügen von geometrischen Formen in der [Draw Shape-Aktion](#) beschrieben. Daraufhin folgt die Erläuterung der [Write Text-Aktion](#), mit der Texte geschrieben werden können. Im Anschluss daran wird in der [Add an Image-Aktion](#) erklärt, wie ein Bild hinzugefügt werden kann und welche zwei Möglichkeiten dafür bereitgestellt werden. Abschließend wird in der [Mind Map-Aktion](#) erläutert, wie eine Mind Map erstellt werden kann.

Wenn ein Drawable-Typ-Objekt hinzugefügt werden soll, wird zuerst ein neues GameObject erstellt. Der Name dieses Objektes wird, wie im Abschnitt "[Drawable-Typen](#)" beschrieben, generiert. Im Anschluss daran wird geprüft, ob das Drawable wie im Abschnitt "[AttachedObject-Objekt](#)" beschrieben transformiert wurde, falls nicht, wird die Transformation durchgeführt. Anschließend wird dem Objekt der entsprechende Tag für den jeweiligen Drawable-Typ zugewiesen. Als nächstes wird das Objekt dem AttachedObject-Objekt als Kind zugewiesen, da dies wie beschrieben der "Ablageort" für die Drawable-Typ-Objekte ist. Danach werden dem jeweiligen Objekt die benötigten Komponenten, wie im Abschnitt "[Drawable-Typen](#)" und deren Unterabschnitten beschrieben, hinzugefügt. Dabei wird für jedes neu erstellte Objekt eine eigene Ebene für die Ebenenreihenfolge erstellt. Die Anzahl der Ebenen wird dabei um eins inkrementiert. Eine Ausnahme hierfür bildet der Mind Map-Knoten, diese wird im entsprechenden Abschnitt näher erläutert.

Abschließend wird die Rotation des AttachedObject-Objektes für das erstellte Objekt übernommen, damit dieses die gleiche Rotation aufweist wie das Drawable, und die Position des Objektes gesetzt. Beim

Einstellen der Position wird ein Wert abhängig von der Ebenenreihenfolge des Objektes und dem Mindestabstand zum Drawable von der Z-Achse abgezogen. Der Mindestabstand wird im Value-Holder durch den Vektor "distanceToDrawable" repräsentiert. Standardmäßig weist dieser einen Wert von (0, 0, 0.0001) auf. Durch das Abziehen dieses Wertes wird dafür gesorgt, dass das Objekt des Drawable-Typs vor der Drawable-Fläche dargestellt wird.

Das Hinzufügen von Drawable-Typ-Objekten ist deshalb auf der Rückseite des Sticky-Notes nicht möglich. Ein hinzugefügtes Objekt würde durch das Abziehen des Wertes von der Position wieder auf der Vorderseite dargestellt werden. Daher können Drawable-Typ-Objekte nur auf der Vorderseite eines Drawables hinzugefügt werden. Wenn versucht wird, ein Drawable-Typ-Objekt auf einem anderen Drawable-Typ-Objekt zu platzieren, ist dies möglich, indem das dazugehörige Drawable erfragt und verwendet wird.

Die Netzwerkaktionen der jeweiligen Hinzufüge-Aktion verwendet neben der Drawable-ID und der Eltern-ID des Drawables eine Instanz der jeweiligen Konfigurationsklasse, um das Drawable-Typ-Objekt zu erstellen.

Eine weitere Gemeinsamkeit der Hinzufüge-Aktionen besteht darin, dass abgeschlossene Aktionen mittels Undo rückgängig gemacht werden können. Dadurch wird das erstellte Objekt gelöscht. Durch Redo wird das Objekt wiederhergestellt.

3.6.2.1 Draw Freehand-Aktion

Durch diese Aktion wird das freie Zeichnen von Linien zur Verfügung gestellt.

Dazu wird mittels der Awake-Methode der Aktion das in der folgenden [Abbildung 3.23](#) dargestellte Line-Menü eingebunden.

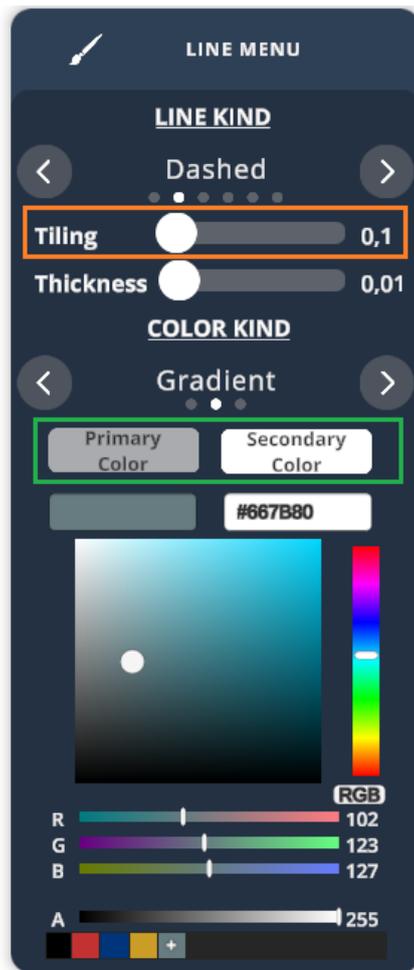


Abbildung 3.23: Das Line-Menü für die Draw Freehand-Aktion.

Mittels des Menüs kann über ein horizontales Selektormenü die Linienart eingestellt werden. Hierbei werden die sechs genannten Linienarten angeboten. Wenn der Benutzer eine andere Art als “Dashed” auswählt, wird die in der orangenen Box dargestellte Tiling-Ebene ausgeblendet. Der Slider für die “Tiling” weist einen Wertebereich von 0,0 bis 10,0 auf, wobei in 0,1 Schritten gewählt werden kann. Standardmäßig ist beim Starten des Menüs die Linienart “Solid” ausgewählt, jedoch wurde für die Abbildung zu Demonstrationszwecken der Tiling-Ebene “Dashed” gewählt.

Über den Slider für die “Thickness” kann die Liniendicke in 0,01 Schritten konfiguriert werden. Hierzu wird ein Wertebereich von 0,00 bis 0,30 bereitgestellt, wobei der Wert 0,00 eine sehr dünne Linie erzeugt, die nur aus der Nähe gut erkennbar ist. Der Standardwert hierfür ist eine Dicke von 0,01. In der nachfolgenden [Abbildung 3.24](#) wird der Vergleich der Liniendicke von dem Minimumwert, Standardwert und Maximumwert veranschaulicht.

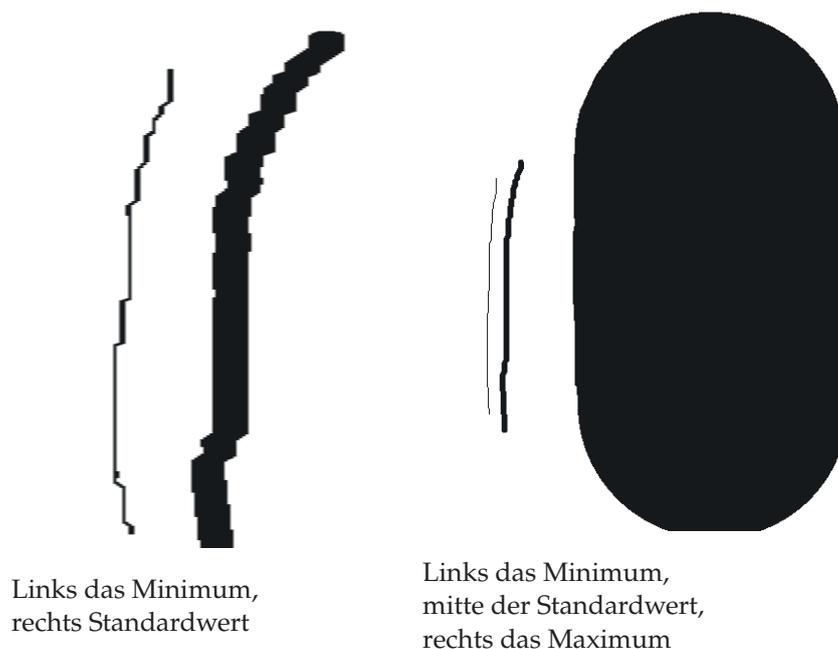


Abbildung 3.24: Vergleich der Liniendicke.

Des Weiteren kann über ein horizontales Selektormenü eine der drei Farbarten der Linie gewählt werden. Die in der [Abbildung 3.23](#) dargestellte grüne Box wird nur für die Farbarten "Gradient" und "Two-Dashed" eingeblendet. Standardmäßig ist beim Start des Menüs die Auswahl auf die Farbart "Monochrome" gesetzt.

Über die eingeblendeten Buttons "Primary Color" und "Secondary Color" kann entschieden werden, ob für die Primär- oder Sekundärfarbe eine Farbe gewählt werden soll. Mittels des HSV-Color-Pickers kann die gewünschte Farbe, wie im [entsprechenden Abschnitt](#) beschrieben, ausgewählt werden.

Ein Benutzer kann auf dem Drawable zeichnen, indem er die linke Maustaste gedrückt hält, solange er zeichnen möchte. Die Positionen für die Linie werden dabei durch das Raycasting-Verfahren gewonnen. Das bedeutet, dass während die linke Maustaste gedrückt wird, die Linie anhand der Punkte entlang des Raycasting-Strahls, der dem Mauszeiger folgt, gezeichnet wird. Während die linke Maustaste gedrückt wird, ist es nicht möglich, das Drawable zu wechseln. Dadurch wird verhindert, dass Linien willkürlich durch den virtuellen Raum gezogen werden.

Beim Starten des Zeichnens wird das Objekt, wie in dem Abschnitt des [Line-Drawable-Typs](#) beschrieben, erstellt. Dafür wird zunächst die globale Position als Startposition verwendet, die durch das Raycasting bereitgestellt wird. Sobald das Objekt erfolgreich erstellt wurde, wird die Startposition mithilfe der Methode "InverseTransformPoint" des Transforms des Objektes in eine lokale Position umgewandelt. Dies ist erforderlich, da der LineRenderer, wie im Abschnitt "[Line-Drawable-](#)

Typ“ erläutert, mit lokalen Positionen arbeiten muss.

Während die linke Maustaste gedrückt gehalten wird, wiederholt sich die Update-Methode der Aktion. Bei jedem Aufruf dieser Methode wird versucht, einen weiteren Punkt zur Linie hinzuzufügen. Bevor der Punkt dem LineRenderer hinzugefügt wird, wird er in eine lokale Position umgewandelt. Dabei wird geprüft, dass der neue Punkt nicht identisch mit dem vorherigen ist. Der LineRenderer händelt Überlappungen, indem er die Z-Achse der Positionen leicht verändert. Dieses Verhalten ist für diese Integration nicht akzeptabel, da die Z-Koordinaten die Ebenenreihenfolge steuern. Daher werden alle Z-Positionswerte der Punkte des LineRenderers auf 0 gesetzt.

Sobald die linke Maustaste losgelassen wird, wird das Beenden des Zeichnens eingeleitet. Zunächst wird geprüft, ob das Mesh der gezeichneten Linie konstruiert werden kann. Dafür benötigt das Mesh mindestens drei Eckpunkte. Wenn dies nicht der Fall ist, wird die gezeichnete Linie gelöscht. Dieser Fall tritt nur ein, wenn ein Benutzer nur einmal kurz die linke Maustaste drückt.

Anschließend wird der Pivot-Punkt, der den Mittelpunkt der Linie repräsentiert, neu festgelegt. Bei der Erstellung der Linie ist der Pivot-Punkt standardmäßig in der Mitte des Drawables. Wenn die Linie eine ungerade Anzahl an Punkten aufweist, wird der mittlere Punkt als Pivot-Punkt gewählt. Bei einer geraden Anzahl wird der Durchschnitt der beiden mittleren Punkte berechnet und als Pivot-Punkt verwendet. Nach dem Festlegen des Pivot-Punktes muss die Linie erneut gezeichnet werden, da sich die lokalen Positionen durch das Verschieben geändert haben. Das Verschieben des Pivot-Punktes ist für das spätere Rotieren erforderlich.

Danach wird das Mesh für die neu gezeichnete Linie berechnet und dem Mesh-Collider übergeben. Abschließend wird ein Memento der gezeichneten Linie erstellt und die Aktion abgeschlossen. Das Memento enthält dabei das Drawable, auf dem gezeichnet wurde, sowie eine Instanz der Konfigurationsklasse "LineConf" für die gezeichnete Linie.

Die Netzwerkaktion dieser Aktion ist die "DrawNetAction". Sie wird während des Zeichnens mit ausgeführt, um zu ermöglichen, dass jeder Teilnehmer des virtuellen Raums das Zeichnen in Echtzeit verfolgen kann.

Sobald die Aktion verlassen wird, wird das Line-Menü ausgeblendet. Falls die Aktion nicht vollständig abgeschlossen wurde, indem beispielsweise während des Zeichnens die Aktion gewechselt wird, wird die gezeichnete Linie gelöscht. Dies ist erforderlich, da mit einer nicht abgeschlossenen Linie nicht interagiert werden könnte. Zudem könnte die Aktion auch nicht per Undo rückgängig gemacht werden.

3.6.2.2 *Draw Shape-Aktion*

Diese Aktion ermöglicht das Zeichnen geometrischer Formen und bietet elf verschiedene Formen zur Auswahl. Zehn davon werden durch Wertangaben berechnet und dann dem Drawable hinzugefügt, während die elfte Form, die geraden Linien, einzeln durch das Hinzufügen von Punkten erstellt wird.

Im Folgenden wird zunächst das Menü dieser Aktion beschrieben, gefolgt von der Erläuterung des Hinzufügens gerader Linien. Danach wird das Hinzufügen der anderen zehn Formen behandelt.

Das Menü dieser Aktion besteht aus drei Teilen. Einem Switch-Menü, mit dem zwischen den anderen beiden Teilen des Menüs gewechselt werden kann. Diese umfassen das Shape-Menü und das Konfigurations-Menü. Eine Besonderheit besteht darin, dass sowohl das Shape-Menü als auch das Konfigurations-Menü an das Switch-Menü gebunden sind und nur über das Switch-Menü verschoben werden können. Das ausgewählte Menü wird im Switch-Menü dunkler dargestellt und der Button zum Auswählen des Menüs deaktiviert. In der nachfolgenden [Abbildung 3.25](#) werden die beiden Auswahlen demonstriert. Das Switch-Menü, bleibt bei jedem Wechsel an derselben Position. Dass dies in der gezeigten [Abbildung 3.25](#) anders aussieht, liegt an der Zentrierung der Abbildungen.



Das Shape-Menü

Das Konfigurations-Menü (Line-Menü)

Abbildung 3.25: Demonstration der Auswahl des Switch-Menüs der Shape-Aktion.

Mit dem Konfigurations-Menü kann die Linie konfiguriert werden. Es handelt sich um dasselbe Menü wie das aus der [Draw Freehand-Aktion](#). Durch das Shape-Menü kann über ein horizontales Selektormenü die gewünschte Form ausgewählt werden. Standardmäßig ist hier die gerade Linie ausgewählt. Wenn die gerade Linie ausgewählt ist, wird im Shape-Menü ein [Switch-Button](#) zum Konfigurieren der Loop-Funktionalität bereitgestellt. Bei aktivierter Einstellung wird der Endepunkt der Linie mit dem Startpunkt beim Beenden des Zeichnens verbunden. Zudem wird ein "Finish"-Button zum Beenden des Zeichnens der geraden Linien bereitgestellt. Diese Menüdarstellung konnte bereits im Shape-Menü der [Abbildung 3.25](#) betrachtet werden.

Switch-Button: Ist ein Button mit einem Schalter, der zum Ein- und Ausschalten von Funktionen verwendet werden kann.

Für die anderen Formen werden Slider mit einem ganzzahligen Wertebereich von 0 bis 100 für die verschiedenen Einstellungen bereitgestellt, mit der Ausnahme von zwei Einstellungen. Welche Einstellungen für welche Form verwendet werden und welche zwei Einstellungen einen anderen Wertebereich aufweisen, wird später in diesem Abschnitt erläutert. Zudem besteht die Möglichkeit, mittels eines Informations-Buttons ein Bild anzeigen zu lassen, das die Einstellungen für die jeweilige Form verdeutlicht. Mittels der [Abbildung 3.26](#) kann die Menüdarstellung für die Quadratform (Square) betrachtet werden. In dieser Abbildung wird ebenfalls das Anzeigen der Informationen demonstriert. Dabei ist in der linken Abbildung die Information ausgeblendet und in der rechten eingeblendet. Auch für die Demonstration dieser Abbildung gilt, dass das Switch-Menü beim Ein- und Ausschalten der Information an derselben Position bleibt. Das versetzte Darstellen der Abbildung liegt erneut an der Zentrierung der unterschiedlich großen Abbildungen.

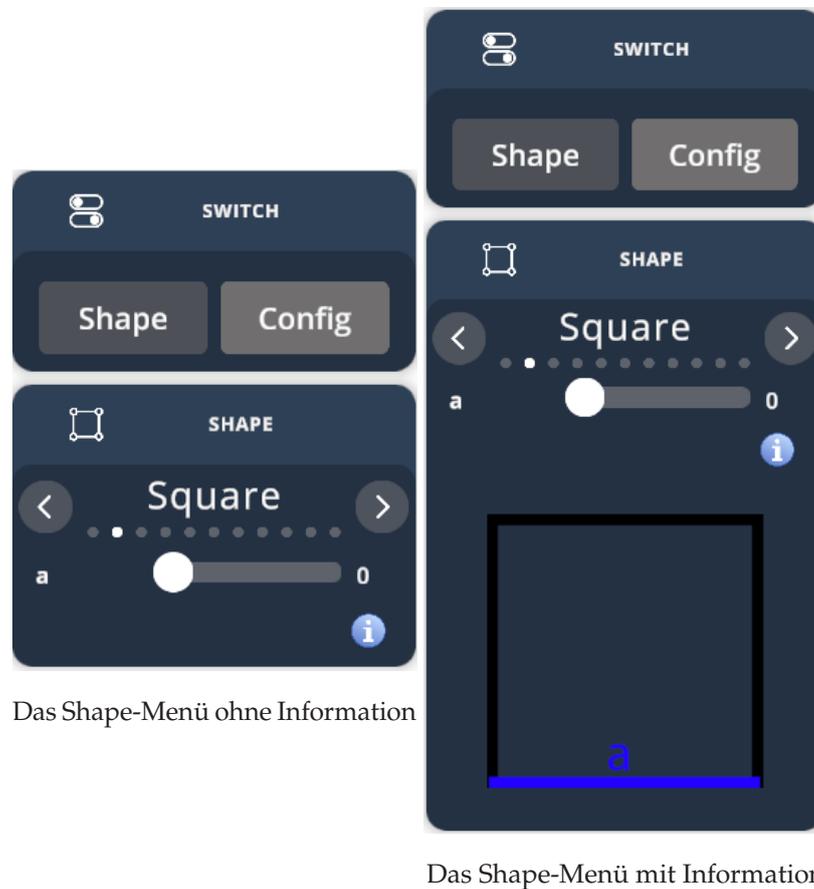


Abbildung 3.26: Demonstration des Shape-Menüs für hinzufügbare geometrische Formen.

Nachdem der Benutzer die entsprechende Form für die geraden Linien ausgewählt hat, kann er mittels eines linken Mausklicks einen Punkt zu der Linie hinzufügen. Sobald der erste Punkt der Linie platziert wurde, wird durch eine Vorschau die Linie zur aktuellen Raycasting-Position

gezeichnet. Mit einem erneuten linken Mausklick kann ein weiterer Punkt an der gewünschten Position der Linie hinzugefügt werden. Für jeden Punkt, der hinzugefügt werden soll, wird überprüft, ob das Drawable nicht gewechselt wurde, dies ist aus demselben Grund wie beim freien Zeichnen nicht erlaubt. Sowohl die Vorschau als auch das Hinzufügen eines Punktes werden direkt über das Netzwerk mittels der "DrawNetAction" gezeichnet.

Mittels der "Tab"-Taste kann der zuletzt hinzugefügte Punkt wieder entfernt werden. Dies kann für jeden hinzugefügten Punkt wiederholt werden. Sobald der Startpunkt durch das Rückgängigmachen entfernt wird, wird das Zeichnen abgebrochen. Sowohl beim Entfernen eines Punktes als auch beim Abbrechen des Zeichnens wird der Benutzer durch eine Benachrichtigung informiert. Diese Benachrichtigungen werden in den Abbildungen 3.27 und 3.28 veranschaulicht.

Zudem kann das Abbrechen des Zeichnens auch über die "ESC"-Taste erfolgen, wobei ebenfalls die Benachrichtigung über das Abbrechen angezeigt wird. Ebenfalls wird das Zeichnen abgebrochen, wenn der Benutzer während des Zeichnens die Aktion wechselt, jedoch wird hier nicht die Benachrichtigung angezeigt. Beim Abbrechen der Aktion über "ESC" oder durch den Aktionswechsel wird die bereits gezeichnete Linie gelöscht. Dies hat den gleichen Grund wie beim freien Zeichnen.

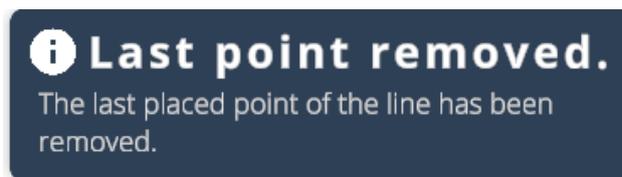


Abbildung 3.27: Demonstration der Information über das Entfernen des zuletzt hinzugefügten Punktes.

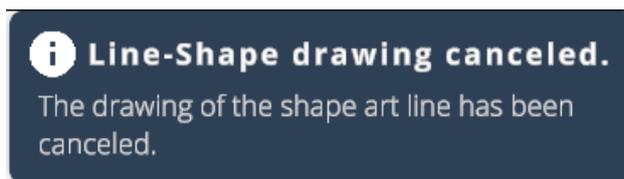
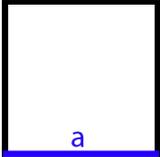
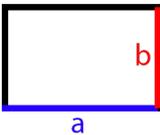
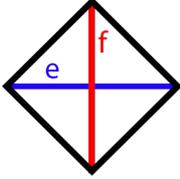


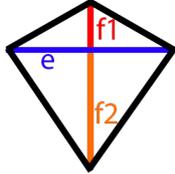
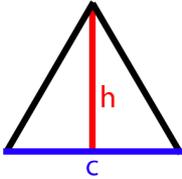
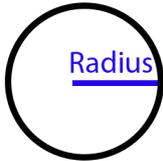
Abbildung 3.28: Demonstration der Information über das Abbrechen des Zeichnens gerader Linien.

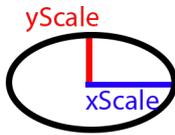
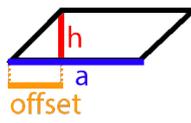
Zum Beenden des Zeichnens gerader Linien gibt es neben dem zuvor beschriebenen "Finish"-Button noch zwei weitere Möglichkeiten. Zum einen kann das Zeichnen beendet werden, indem beim Hinzufügen des letzten Punktes die "Strg"-Taste zusätzlich gedrückt wird. Zum anderen kann das Zeichnen durch einen Mauseklick beendet werden, wobei kein weiterer Punkt hinzugefügt wird. Beim Beenden des Zeichnens wird ein Memento über die gezeichnete Linie und das dazugehörige

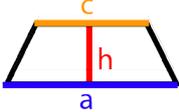
Drawable angelegt und die Aktion abgeschlossen.

Mithilfe der folgenden [Tabelle 3.1](#) werden die anderen zehn bereitgestellten Formen mit ihren Einstellungen und einem erklärenden Bild dargestellt. Das jeweils dargestellte Bild ist das Bild, das über den Informations-Button eingeblendet werden kann.

Form	Einstellungen	erklärendes Bild
Quadrat (Square)	Ein Quadrat besteht aus vier gleich langen Seiten. <ul style="list-style-type: none"> • Seitenlänge a: Gibt die Länge der Seiten für das Quadrat an. 	
Rechteck (Rectangle)	Bei einem Rechteck sind die gegenüberliegenden Seiten gleich lang. <ul style="list-style-type: none"> • Seitenlänge a: Gibt die Länge der Seite a an. • Seitenlänge b: Gibt die Länge der Seite b an. 	
Raute (Rhombus)	Eine Raute ist ein Viereck, bei dem alle Seiten gleich lang sind und bei dem gegenüberliegende Winkel gleich groß sind. Außerdem sind gegenüberliegende Seiten parallel zueinander (Studyflix, 2024). <ul style="list-style-type: none"> • Seitenlänge f: Gibt die Länge für die vertikale Diagonale an. • Seitenlänge e: Gibt die Länge für die horizontale Diagonale an. 	

<p>Drachen (Kite)</p>	<p>Ein Drachen ist ein Viereck, das symmetrisch zur Diagonalen f ist, wobei f aus den Diagonalen f_1 und f_2 besteht (Köller, 2004).</p> <ul style="list-style-type: none"> • Seitenlänge f_1: Gibt die Länge für die obere Hälfte der vertikalen Diagonale an. • Seitenlänge f_2: Gibt die Länge für die untere Hälfte der vertikalen Diagonale an. • Seitenlänge e: Gibt die Länge für die horizontale Diagonale an. 	 <p>The diagram shows a kite-shaped quadrilateral with two pairs of adjacent sides of equal length. A vertical diagonal is divided into two segments, f_1 (top) and f_2 (bottom), both labeled in red. A horizontal diagonal is labeled e in blue.</p>
<p>Dreieck (Triangle)</p>	<p>Bei dem hier bereitgestellten Dreieck handelt es sich um ein gleichschenkliges Dreieck, bei diesem sind mindestens zwei Seiten gleich lang. Dies sind in dem Bild die schwarzen Seiten. Wenn $c = h$ ist, dann handelt es sich um ein gleichseitiges Dreieck.</p> <ul style="list-style-type: none"> • Seitenlänge c: Gibt die Länge der Seite c an. • Seitenlänge h: Gibt die Höhe des Dreiecks an. Diese verläuft vertikal von der Mitte der Seite c aus. 	 <p>The diagram shows an isosceles triangle with two equal black sides and a blue base labeled c. A red vertical line from the top vertex to the base represents the height, labeled h.</p>
<p>Kreis (Circle)</p>	<p>Der Kreis ist eine runde Form, bei der alle Punkte den konstanten Abstand des Radius zum Mittelpunkt aufweisen.</p> <ul style="list-style-type: none"> • Seitenlänge Radius: Gibt den Radius zum Mittelpunkt an. 	 <p>The diagram shows a circle with a blue line segment from the center to the circumference, labeled "Radius" in blue.</p>

<p>Ellipse</p>	<p>Eine Ellipse ist eine ovalförmige geometrische Form, die durch die Angabe eines vertikalen und eines horizontalen Radius berechnet wird.</p> <ul style="list-style-type: none"> • Seitenlänge X-Scale: Gibt den horizontalen Radius zum Mittelpunkt an. • Seitenlänge Y-Scale: Gibt den vertikalen Radius zum Mittelpunkt an. 	
<p>Parallelogramm (Parallelogram)</p>	<p>Das Parallelogramm ist ein Viereck, bei dem gegenüberliegende Seiten gleich lang und parallel zueinander sind. Die Besonderheit dieser Form besteht darin, dass es versetzt sein kann, wodurch die Eckpunkte nicht auf einer vertikalen Linie liegen.</p> <ul style="list-style-type: none"> • Seitenlänge a: Gibt die Länge der Seite a an. • Seitenlänge h: Gibt die Höhe des Parallelogrammes an. • Shift (offset): Gibt die Versetzung des Parallelogrammes an. Hierfür wird ein anderer ganzzahliger Slider bereitgestellt, der einen Wertebereich von -50 bis 50 aufweist. Der Wertebereich von -50 bis 0 repräsentiert dabei, die Verschiebung nach links und die von 0 bis 50 nach rechts. Der Schieberegler des Sliders befindet sich standardmäßig in der Mitte und somit auf dem Wert 0. 	

<p>Trapez (Trapezoid)</p>	<p>Das Trapez ist eine geometrische Form, bei dem ein Paar der gegenüberliegenden Seiten (a und c) parallel zueinander ist. Das hier bereitgestellte Trapez stellt ein gleichschenkliges Trapez dar. Bei einem gleichschenkligen Trapez sind die beiden nicht parallelen Seiten gleich lang und zudem weist das Trapez eine Symmetrie in Bezug auf die mittlere vertikalen Linie auf (im Bild ist dies h) (Wikipedia, 2024c).</p> <ul style="list-style-type: none">• Seitenlänge a: Gibt die Länge der Seite a an.• Seitenlänge c: Gibt die Länge der Seite c an.• Seitenlänge h: Gibt die Höhe des Trapez an.	 <p>Das Diagramm zeigt ein gleichschenkliges Trapez. Die obere Basis ist orange markiert und mit 'c' beschriftet. Die untere Basis ist blau markiert und mit 'a' beschriftet. Die Höhe des Trapezes ist durch eine vertikale rote Linie in der Mitte dargestellt und mit 'h' beschriftet.</p>
-------------------------------	---	---

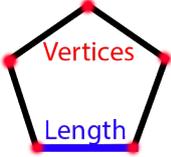
<p>Polygon</p>	<p>Ein Polygon ist eine geschlossene Form, die aus einer endlichen Anzahl von Eckpunkten besteht. Dabei werden benachbarte Eckpunkte durch gerade Linien verbunden (Pötzelberger, 2006). Das Polygon, das hier gezeichnet werden kann, ist konvex. Bei einem konvexen Polygon sind alle Innenwinkel kleiner als 180°, wodurch keine Einbuchtungen entstehen und keine innenliegenden Ecken vorhanden sind (Matheretter, n.d.). Aufgrund der Tatsache, dass bereits Dreiecke und verschiedene Vierecke bereitgestellt werden, beginnt der Wertebereich der Anzahl der Eckpunkte (Vertices) erst ab 5 und reicht bis 50.</p> <ul style="list-style-type: none"> • Seitenlänge Length: Gibt die Länge des Abstandes zwischen den Punkten an. • Eckpunkte Vertices: Gibt die Anzahl an Eckpunkten an. 	
----------------	---	---

Tabelle 3.1: Die bereitgestellten geometrischen Formen, ihre Einstellungen und jeweils ein erklärendes Bild.

In der nachfolgenden [Abbildung 3.29](#) können Polygone mit fünf bis zehn Eckpunkten betrachtet werden, die alle eine Seitenlänge von 10 aufweisen.

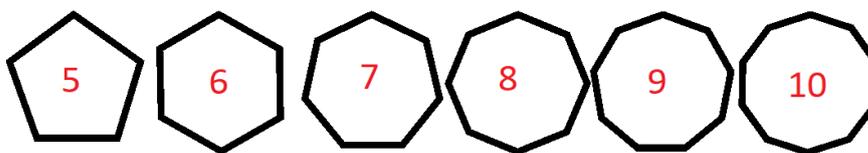


Abbildung 3.29: Polygone mit einer unterschiedlichen Anzahl von Eckpunkten.

Um eine der gezeigten geometrischen Formen hinzuzufügen, muss der Benutzer, nachdem er die gewünschte Form konfiguriert hat, mit einem linken Mausklick auf ein Drawable klicken. Mithilfe der statischen Klasse "ShapePointsCalculator" werden die Eckpunkte der Form unter Berücksichtigung der getätigten Einstellungen berechnet, sodass der

erkannte Punkt des Raycastings den Mittelpunkt der Form darstellt. Der Mittelpunkt wird auch als Pivot-Punkt des Objektes verwendet. Die Berechnung der Polygone wurde dabei durch ein Kommentar von Eric Balcon² inspiriert. Bei der Berechnung der Formen ist der zuletzt hinzugefügte Punkt der Startpunkt, damit die Form geschlossen wird. Hierfür wird nicht die Loop-Eigenschaft verwendet, um eine sinnvolle Interaktion mittels der [Line Split-Aktion](#), [Line Point Erase-Aktion](#) und der [Line Connection Erase-Aktion](#) zu ermöglichen. Wenn die Linie der Form die Loop-Eigenschaft aufweisen würde, würden beispielsweise bei einer Aufteilung der Linie durch die [Line Split-Aktion](#) beide Sublinien ebenfalls die Loop-Eigenschaft aufweisen und nicht das gewünschte Ergebnis des Teilens erzielen, da dadurch andere geometrische Formen erzeugt werden könnten.

Sobald die Form erfolgreich hinzugefügt wurde, wird sie über das Netzwerk hinweg gezeichnet. Dafür wird ebenfalls die "DrawNetAction" verwendet. Außerdem wird die Aktion abgeschlossen und ein Memento über die Linie der hinzugefügten Form und dem dazugehörigen Drawable angelegt.

3.6.2.3 *Write Text-Aktion*

Mit dieser Aktion wird das Schreiben von Texten auf einem Drawable ermöglicht. Beim Aufrufen dieser Aktion wird mittels der Awake-Methode das folgende Menü der [Abbildung 3.30](#) bereitgestellt. In der Abbildung wurde zu Demonstrationszwecken die Außenfarbe gewählt, wodurch der grüne Bereich eingeblendet wird. Außerdem wurde der Textstyle "Underline" mittels des Buttons "U" aktiviert.

² <https://www.youtube.com/watch?v=DdAfwHYNFOE>

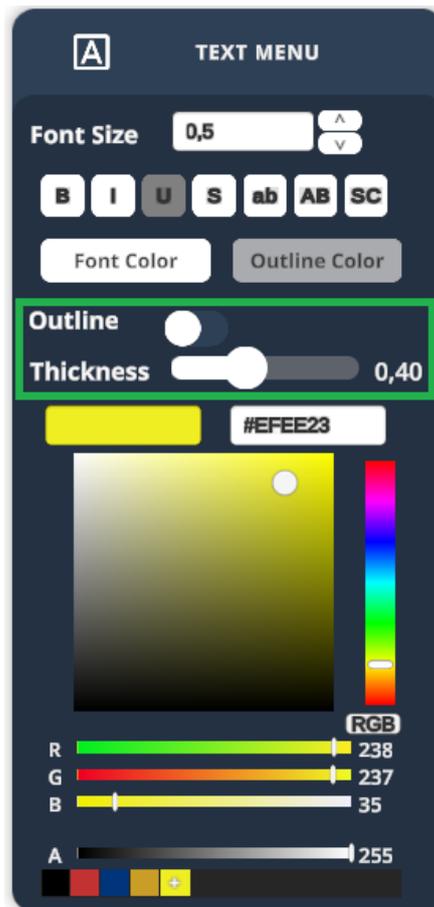


Abbildung 3.30: Demonstration des Text-Menüs.

Das Menü bietet ein Eingabefeld sowie ein Auf- und Ab-Button zur Einstellung der Schriftgröße. Wenn der Benutzer das Eingabefeld auswählt, werden die Shortcuts von SEE während des Eingabevorgangs deaktiviert. Die Auf- und Ab-Buttons ermöglichen es dem Benutzer, die Schriftgröße schrittweise um 0,01 in die entsprechende Richtung zu ändern, indem er mit der linken Maustaste klickt. Zudem wird mittels der MonoBehaviour-Komponente "ButtonHelded" das fortlaufende Verändern der Größe unterstützt, indem die rechte Maustaste gehalten wird. Diese Komponente wurde durch einen Beitrag von dem Benutzer Ahsanhabibrafy des Unity-Forums inspiriert³. Der Wertebereich für die Schriftgröße liegt zwischen 0,15 und 5,0. Die Standardgröße beträgt 0,5. In der folgenden [Abbildung 3.31](#) werden diese Unterschiede veranschaulicht. Die Darstellung der Schriftgröße variiert je nach Entfernung des Benutzers zum Drawable. Die Abbildung wurde mit einem größeren Abstand erstellt, um sicherzustellen, dass die größte Größe nicht zu riesig wirkt.

³ <https://discussions.unity.com/t/how-to-make-the-button-respond-to-touch-and-hold-feature/213417/6>



Abbildung 3.31: Demonstration der kleinsten, Standard- und größten Schriftgröße.

Des Weiteren stellt das Menü sieben verschiedene Textstyle-Buttons bereit, die mit einem einfachen linken Mausklick aktiviert oder deaktiviert werden können. Der aktivierte Zustand eines solchen Buttons ist in der [Abbildung 3.30](#) zu sehen. Die drei rechten Textstyle-Buttons sind so konzipiert, dass sie sich gegenseitig ausschließen, sodass jeweils nur einer zur gleichen Zeit aktiv sein kann. Wenn beispielsweise "ab" aktiviert ist und der Benutzer auf den Button "AB" klickt, wird "ab" automatisch deaktiviert. Die Abkürzungen der Buttons und ihre Effekte sind bereits in der [Abbildung 3.14](#) dargestellt worden.

Zudem bieten sowohl die Textstyle-Buttons als auch die Auf- und Ab-Buttons erklärende Tooltips durch die MonoBehaviour-Komponente "ButtonHoverTooltip". Um diese anzuzeigen, muss der Benutzer den Mauszeiger für eine gewisse Zeit über dem jeweiligen Button halten. Eine entsprechende Erklärung ist in der [Abbildung 3.32](#) zu sehen. Die Kombination aus Eingabefeld und Auf- und Ab-Buttons ist in verschiedenen bereitgestellten Menüs vertreten. Deshalb wurde eine eigene UI-Komponente namens "InputfieldWithButtons" für diese Kombination erstellt, wodurch bei jeder Verwendung dieser Komponente der Tooltip mit bereitgestellt wird.

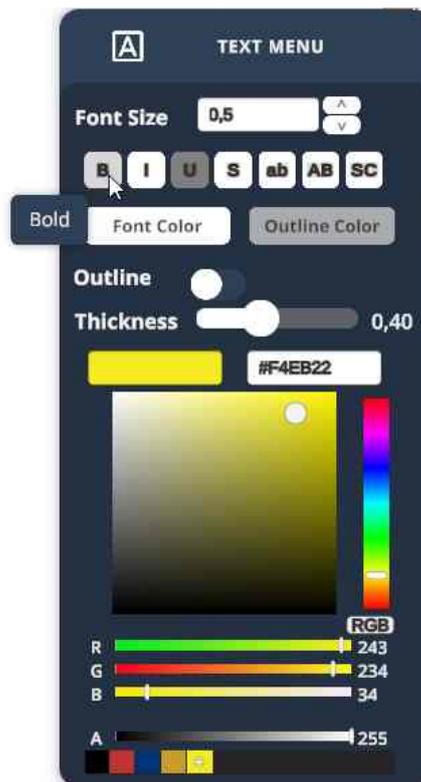


Abbildung 3.32: Demonstration des Tooltips für die Style-Buttons.

Außerdem bietet das Text-Menü die Möglichkeit zur Auswahl der Schriftfarbe oder der Außenfarbe. Wenn der Benutzer auf den Button "Outline Color" klickt, wird der Bereich, der in der [Abbildung 3.30](#) durch die grüne Box markiert ist, eingeblendet. Standardmäßig ist die Schriftfarbe aktiviert, wodurch die Einstellungen für die Außenfarbe ausgeblendet werden.

Mittels des Switch-Buttons für die Einstellung "Outline" kann die Außenfarbe aktiviert oder deaktiviert werden. Standardmäßig ist diese deaktiviert. Der Slider für die "Thickness" gibt die Dicke der Außenfarbe an.

Nachdem der Benutzer den Text nach Belieben konfiguriert hat, kann er durch einen linken Mausklick auf dem Drawable eine geeignete Stelle für den Text auswählen. Anschließend wird der Eingabedialog des Textes der [Abbildung 3.33](#) geöffnet, wodurch der gewünschte Text eingetragen werden kann. Durch die Bestätigung des Buttons "Ok" wird der Text an der ausgewählten Stelle positioniert. Wenn der Text erfolgreich erstellt wurde, wird er über das Netzwerk mittels der Methode "WriteTextNetAction" erstellt. Zusätzlich wird ein Memento über die Text-Konfiguration sowie dem zum Text gehörenden Drawable angelegt und die Aktion abgeschlossen. Sollte der eingegebene Text leer sein, wird eine Warnung ausgegeben und das Hinzufügen beginnt von vorn.

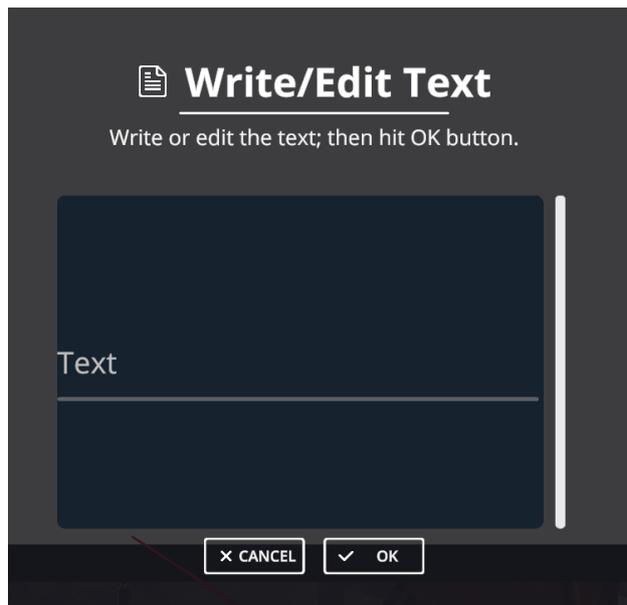


Abbildung 3.33: Demonstration des Dialogs, in dem der Text eingegeben wird.

3.6.2.4 Add an Image-Aktion

Mittels dieser Aktion wird das Hinzufügen von Bildern bereitgestellt. Der Benutzer muss als erstes eine Position für das Bild wählen, indem er an einer geeigneten Stelle auf dem Drawable einen linken Mausklick ausführt. Dadurch wird das in der [Abbildung 3.34](#) dargestellte Auswahlmenü bereitgestellt. In diesem muss der Benutzer auswählen, ob er ein lokales Bild oder ein Webbild, also ein Bild aus dem Internet, hinzuzufügen möchte. Der Benutzer kann die Auswahl mittels des bereitgestellten "X"-Buttons abbrechen. Sollte der Benutzer bei geöffnetem Menü die Aktion wechseln, wird dieses zerstört.

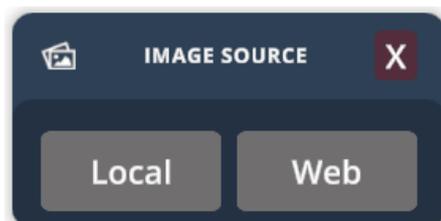


Abbildung 3.34: Demonstration der Quellenauswahl für Bilder.

Wenn der Benutzer "Local" wählt, wird der im Abschnitt "[Anforderungen und Ziele](#)" beschriebene Filebrowser geöffnet. Dieser wird durch die MonoBehaviour-Komponente "DrawableFileBrowser" bereitgestellt, die dem obersten UI-Objekt "UI-Canvas" hinzugefügt wird. Während der Filebrowser geöffnet ist, werden die SEE-Shortcuts sowie die Drawable-Interaktion deaktiviert. Mittels des Filebrowsers kann eine PNG- oder JPG-Datei aus einem beliebigen Verzeichnis gewählt werden. Standardmäßig öffnet der "DrawableFileBrowser" das von

SEE bereitgestellte Verzeichnis für die Drawable-Bilder. Dieser ist in der AppData unter "Drawable/Image/" zu finden. In der folgenden [Abbildung 3.35](#) kann dieser Filebrowser veranschaulicht werden. Nachdem der Benutzer ein Bild ausgewählt hat und das Laden über den "Load"-Button ausführt, wird das Bild als **Image-Drawable-Typ**-Objekt erstellt und hinzugefügt.

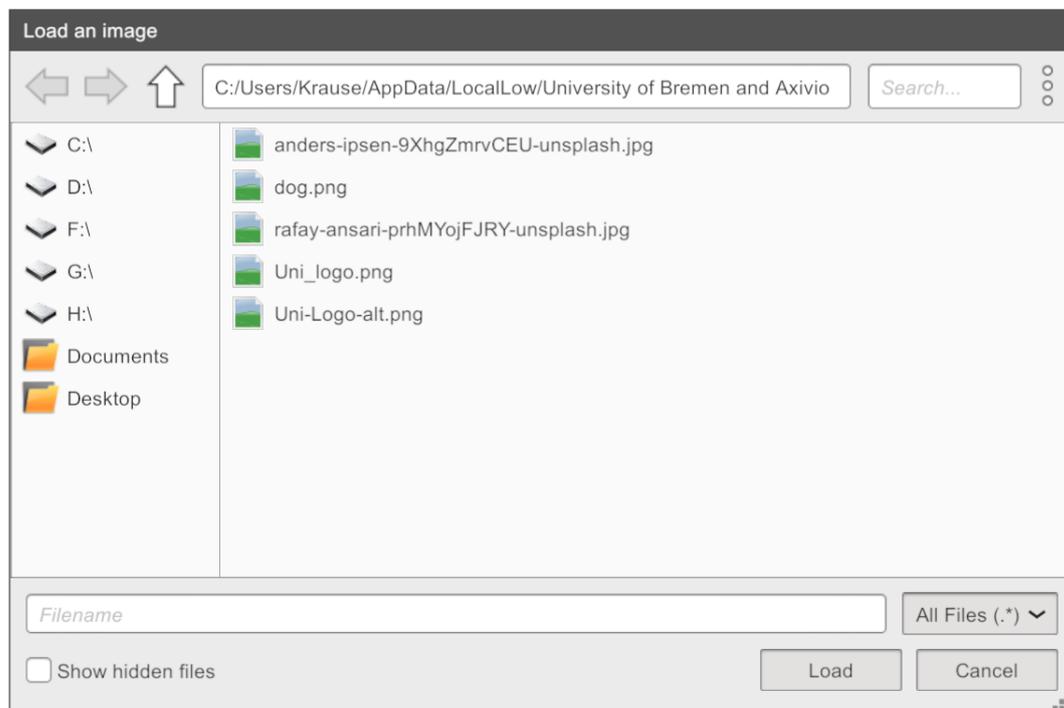


Abbildung 3.35: Demonstration des Filebrowsers für die Bilderauswahl.

Bei diesem Hinzufügevorgang wird mittels der Methode "CreateImageFile" überprüft, ob das Bild in dem eben genannten Drawable-Bilder-Verzeichnis vorhanden ist. Ist dies nicht der Fall, wird eine Sicherung des Bildes darin abgelegt. Falls bereits eine Datei mit dem gleichen Namen in dem Verzeichnis existiert, wird ein temporäres Verzeichnis mit einer temporären Bilddatei des anzulegenden Bildes erzeugt. Anschließend wird überprüft, ob die bereits existierende Datei und die temporäre Datei dieselbe Dateigröße aufweisen. Wenn dies der Fall ist, wird das Anlegen der Sicherung abgebrochen und die temporäre Datei sowie das temporäre Verzeichnis gelöscht. Falls die Dateien nicht dieselbe Größe aufweisen, wird dem zu erstellenden Bild eine Versionsnummer in runden Klammern hinzugefügt. Ein Beispiel hierfür wäre "Testdatei (1).png". Anschließend wird geprüft, ob eine Datei mit diesem neuen Dateinamen existiert. Ist dies nicht der Fall, wird die Datei mit dem neuen Namen angelegt. Anderenfalls wird erneut die Größe überprüft und bei Ungleichheit die nächsthöhere Versionsnummer gewählt. Es wird solange geprüft, bis entweder die gleiche Größe oder eine freie Versionsnummer gefunden wird. Am Ende beider Fälle werden die

temporären Dateien gelöscht. Die Sicherung des Bildes wird erstellt, um die Wiederherstellung des Bildes zu gewährleisten. Zudem werden der Pfad sowie die Byte-Daten dieser Sicherung in die "ImageViewHolder"-Komponente gespeichert und die DrawableFileChooser-Komponente des "UI-Canvas" zerstört.

Wenn der Benutzer im Auswahlménü "Web" wählt, wird der in der [Abbildung 3.36](#) gezeigte Eingabedialog eingeblendet. In dem Dialog muss der Benutzer in das URL-Eingabefeld die Webadresse des Bildes einfügen. Im Feld "Filename" kann er den gewünschten Dateinamen eingeben. Mittels des "Ok"-Buttons wird der Dialog geschlossen und dem Drawable, auf dem das Webbild hinzugefügt werden soll, eine MonoBehaviour-Komponente namens "DownloadImage" hinzugefügt. Die Idee zu dieser Komponente stammt von Marcus Ansley⁴. Mittels dieser Komponente wird versucht, das Bild zu downloaden. Wenn dies nicht möglich ist, wird der Benutzer darüber mittels einer Benachrichtigung informiert. Nach dem Download wird der Wunschname überprüft. Wenn kein Name eingetragen wurde, wird einer über die "GetRandomStringForFile" - Methode des Drawable-Holders generiert und die Dateierweiterung "PNG" verwendet. Sollte lediglich die Dateierweiterung vergessen werden, wird ebenfalls die Dateierweiterung "PNG" hinzugefügt. Anschließend wird eine Sicherung des Bildes, wie zuvor beschrieben, mit der "CreateImageFile"-Methode erstellt. Zudem wird eine Benachrichtigung über das erfolgreiche Downloaden angezeigt und die DownloadImage-Komponente zerstört. Abschließend wird das Bild auf dem Drawable platziert.



Abbildung 3.36: Demonstration des Hinzufügedialogs für Webbilder.

⁴ <https://m-ansley.medium.com/unity-web-requests-downloading-an-image-e88d7389dd5a>

Die hinzugefügten Bilder beider Varianten werden über das Netzwerk mittels der "AddImageNetAction" hinzugefügt. Dafür werden die Byte-Daten des jeweiligen Bildes verwendet. Zudem werden bei beiden Varianten am Ende des Hinzufügens ein Memento mit der Konfiguration des Bildes und dem dazugehörigen Drawables erstellt sowie die Aktion abgeschlossen.

3.6.2.5 Mind Map-Aktion

Diese Aktion stellt das Erstellen beziehungsweise Hinzufügen von Mind Map-Knoten bereit. Dazu wird in der Awake-Methode das in der [Abbildung 3.37](#) dargestellte Mind Map-Menü eingeblendet, um die Art des Knotens auszuwählen, der hinzugefügt werden soll.

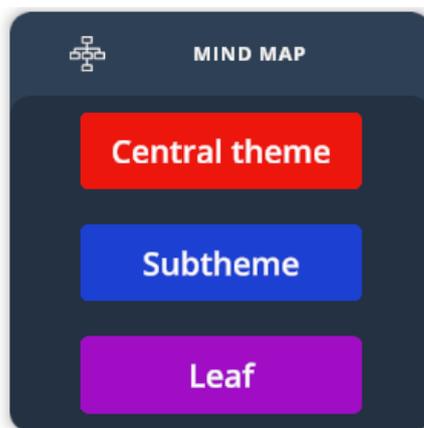


Abbildung 3.37: Demonstration des Mind Map-Menüs.

Wenn der Benutzer versehentlich die linke Maustaste betätigt, ohne eine Auswahl getroffen zu haben, wird er darüber informiert, dass er zuerst eine der bereitgestellten Operationen des Menüs auswählen muss.

Sollte der Benutzer versuchen, ein Unterthema oder Blatt hinzuzufügen, obwohl noch kein zentrales Thema auf dem Drawable existiert, wird eine entsprechende Warnung angezeigt, die darauf hinweist, dass zuerst ein zentrales Thema hinzugefügt werden muss.

Nachdem der Benutzer eine geeignete Knotenart ausgewählt hat und die gewünschte Position des Knotens mit einem linken Mausklick gewählt hat, wird der Eingabedialog für Texte (siehe [Abbildung 3.33](#)) geöffnet, um die Eingabe des Knotentextes zu ermöglichen. Durch das Klicken des "Ok"-Buttons in dem Dialog wird der Knoten hinzugefügt. Wenn es sich bei dem Knoten um einen Unterthema- oder Blatt-Knoten handelt, wird das in der [Abbildung 3.38](#) gezeigte Elternauswahl-Menü eingeblendet. Mit diesem kann der gewünschte Elternknoten ausgewählt werden. Hierfür stehen alle zentralen Themen und Unterthemen

des Drawables zur Verfügung. Durch das Klicken auf "Finish" kann die Auswahl bestätigt werden und die Aktion wird abgeschlossen.

Neben dem Elternauswahl-Menü kann das gewünschte Elternobjekt auch durch einen linken Mausklick auf das entsprechende Objekt ausgewählt werden. Dabei wird als Vorschau eine Verbindungslinie bereitgestellt. Nachdem die Auswahl des Elternknotens abgeschlossen ist, wird die Aktion abgeschlossen. Währenddessen wird der Knoten mittels der "MindMapCreateNodeNetAction" über das Netzwerk hinweg erstellt und das entsprechende Memento angelegt. Dieses Memento umfasst die Konfiguration des Mind Map-Knotens sowie die des Drawables und welche Art von Knoten hinzugefügt wurde.

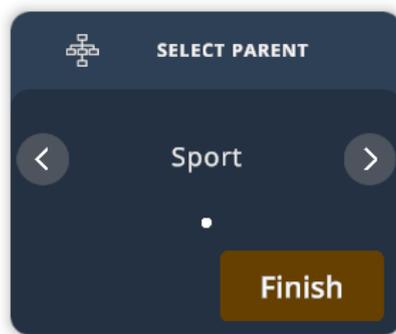


Abbildung 3.38: Demonstration der Auswahl des Elternknotens.

Während der Auswahl des Elternknotens kann die Aktion über die "ESC"-Taste abgebrochen werden. Dabei werden der erstellte Knoten sowie dessen Verbindungslinie für die Vorschau und das geöffnete Elternauswahl-Menü zerstört. Dasselbe geschieht auch, wenn der Benutzer die Aktion wechselt, während sie noch nicht abgeschlossen war.

Bei dieser Aktion wurde bewusst auf eine individuelle Konfiguration der Mind Map-Knoten verzichtet, um das Hinzufügen so einfach und schnell wie möglich zu gestalten. Deshalb werden die Komponenten, aus dem der Mind Map-Knoten dargestellt wird, schwarz eingefärbt. Die Linien und der Text, die zu einem Mind Map-Knoten gehören, können durch die [Edit-Aktion](#) individualisiert werden.

Mit der Undo-Funktionalität wird der hinzugefügte Knoten wieder entfernt und wenn dieser einem Elternknoten zugeordnet war, wird er aus der Liste der Kinderknoten dieses Elternknotens entfernt. Zudem wird dann ebenfalls die Verbindungslinie zu dem Elternknoten entfernt. Bei Redo wird der entfernte Knoten wieder hinzugefügt.

3.6.3 Color Picker-Aktion

Diese Aktion ermöglicht es, die Farbe eines Drawable-Typ-Objektes oder eines Sticky-Notes als aktuell ausgewählte Primär- oder Sekundärfarbe im Value-Holder festzulegen. Dazu wird mit der Awake-Methode der Aktion das Menü der [Abbildung 3.39](#) eingebunden. Über das Menü kann mittels des Switch-Buttons ausgewählt werden, ob die gewählte Farbe für die Primär- oder Sekundärfarbe festgelegt werden soll. Zusätzlich zeigt das Menü die aktuell im Value-Holder ausgewählte Primär- und Sekundärfarbe an. Die üblichen Funktionalitäten wie das Auswählen der Farbe über das Farbfeld oder die Eingabe eines Hex-Wertes mittels des HSV-Color-Pickers wurden für dieses Menü deaktiviert.

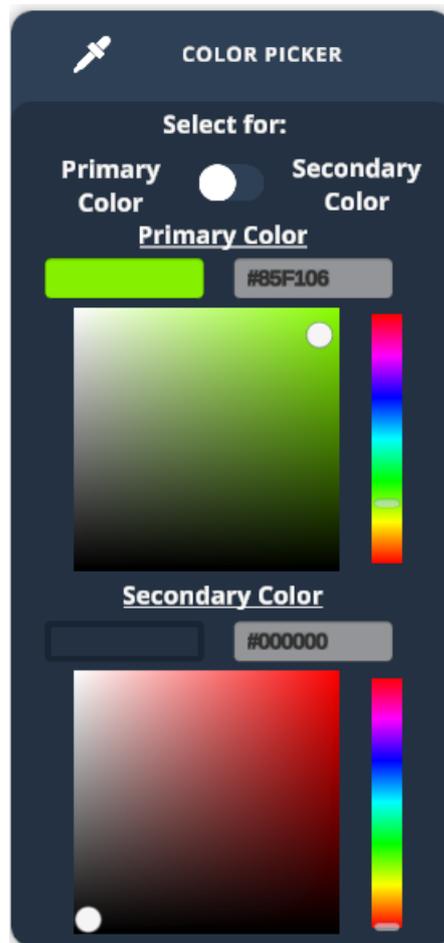


Abbildung 3.39: Demonstration des bereitgestellten Color-Picker Menüs.

Wenn der Benutzer mit einem linken Mausklick auf ein Drawable-Typ-Objekt klickt, wird die Primärfarbe des Objektes in den gewünschten Speicher für die Primär- oder Sekundärfarbe des Value-Holder geladen. Mit einem rechten Mausklick passiert dasselbe für die Sekundärfarbe

des gewählten Objektes.

Für Linien mit einer Monochrome-Farbart liefert sowohl der linke als auch der rechte Mausklick dieselbe Farbe. Gleiches gilt für ein Bild. Bei einem Text, bei dem die Außenfarbe seit seiner Erstellung nicht gewählt wurde, ist die Sekundärfarbe transparent. Wie eine transparente Farbe in dem Menü dargestellt wird, konnte aus der [Abbildung 3.39](#) in der Farbbox der Sekundärfarbe entnommen werden. Dort ist der Hex-Wert #000000 zu sehen, was für Schwarz steht. Die Farbe ist jedoch transparent, sodass die Hintergrundfarbe des Menüs zu sehen ist.

Eine Besonderheit stellen hierbei die Mind Map-Knoten dar, da sie aus mehreren Komponenten bestehen. Nach einem linken oder rechten Mausklick wird das Menü der folgenden [Abbildung 3.40](#) eingebunden.

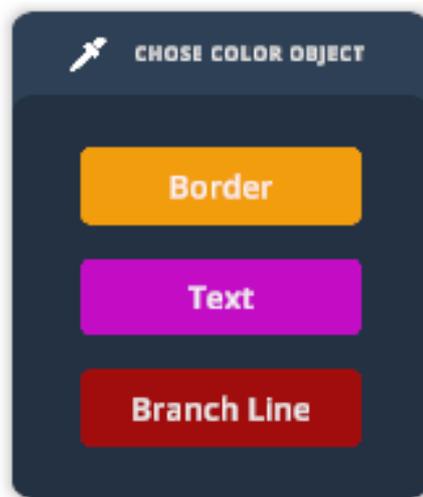


Abbildung 3.40: Demonstration der Objektauswahl eines Mind Map-Knotens für den Color-Picker.

Nachdem der Benutzer eines der verfügbaren Objekte ausgewählt hat, wird die entsprechende Farbe in den Speicher geladen. Sollte es sich bei dem Knoten um ein zentrales Thema handeln, wird der Button für die Verbindungslinie deaktiviert, da ein zentrales Thema keinen Elternknoten haben kann.

Nachdem der Benutzer die Farbe eines Drawable-Typs mittels linken oder rechten Mausklicks gewählt hat, wird die Aktion abgeschlossen. Zusätzlich wird ein Memento erstellt, das die Werte der vorherigen Primär- und Sekundärfarbe, die neu gewählte Farbe sowie den Status, ob für die Primär- oder Sekundärfarbe gewählt wurde, enthält.

Sobald die Aktion abgeschlossen ist, kann sie mittels Undo rückgängig gemacht werden, wodurch die ursprünglichen Farben in den Value-Holder geladen werden. Nach dem Undo kann die Aktion mittels Redo

wiederhergestellt werden, wodurch die neu gewählte Farbe erneut geladen wird.

Wenn der Benutzer die Aktion wechselt, werden die bereitgestellten Menüs zerstört.

3.6.4 Transformations-Aktionen

Dieser Abschnitt behandelt die Transformations-Aktionen. Eine Transformation beschreibt dabei die Wandlung von Form, Struktur oder Gestalt von einem Ausgangszustand in einen Zielzustand (Alt, 2020). Bevor die Aktionen dieses Abschnittes erläutert werden, wird zunächst die Komponente zum [Anzeigen des ausgewählten Objektes](#) sowie die [Kollisionserkennung](#) und die [Berechnung des nächstgelegenen Punktes für Linien](#) erklärt. Danach folgen die Erläuterungen zu den Transformations-Aktionen. Diese umfassen die [Edit-Aktion](#), die [Move or Rotate-Aktion](#), die [Scale-Aktion](#), die [Change The Sorting Layer-Aktion](#) und die [Cut, Copy, Paste-Aktion](#) für alle Drawable-Typ-Objekte. Des Weiteren werden die zwei Transformations-Aktionen beschrieben, die nur für die Objekte des [Line-Drawable-Typs](#) konzipiert sind. Diese sind die [Move a Point-Aktion](#) und die [Line Split-Aktion](#).

Die Undo-Funktionalität der Transformations-Aktionen setzt das Objekt in den vorherigen Zustand zurück, wodurch alle vorgenommenen Änderungen rückgängig gemacht werden. Die Redo-Funktionalität stellt die zuvor rückgängig gemachten Änderungen wieder her und aktualisiert das Objekt entsprechend.

3.6.4.1 Blink-Effekt

Der Blink-Effekt wird durch eine MonoBehaviour-Komponente dargestellt und dient dazu, ein ausgewähltes Objekt anzuzeigen. Mithilfe dieses Effektes wird für die Drawable-Typen-Objekte ein Blinken integriert. Dabei werden die Renderer bei Linien und Texten sowie der Canvas bei Bildern ein- und ausgeschaltet, um die Sichtbarkeit des Objektes zu steuern. Bei einem Mind Map-Knoten blinken sowohl die Knotenumrandung als auch der Knotentext.

Für Objekte, die keinen Renderer oder Canvas bieten, wird geprüft, ob eine [Highlight](#)-Komponente vorhanden ist. Falls nicht, wird eine mittels der Klasse "GameHighlighter" bereitgestellt. Auch für diese Highlight-Komponente wird, wie eben beschrieben, ein Blink-Effekt durch das An- und Ausschalten erzeugt. Der bereitgestellte Highlight-Effekt umfasst ein gelbes äußeres Rand-Highlighting mit zusätzlichem Leuchteffekt und wird in der folgenden [Abbildung 3.41](#) dargestellt.

Highlight: Ist das visuelle Hervorheben eines Objektes, um es deutlicher darzustellen.

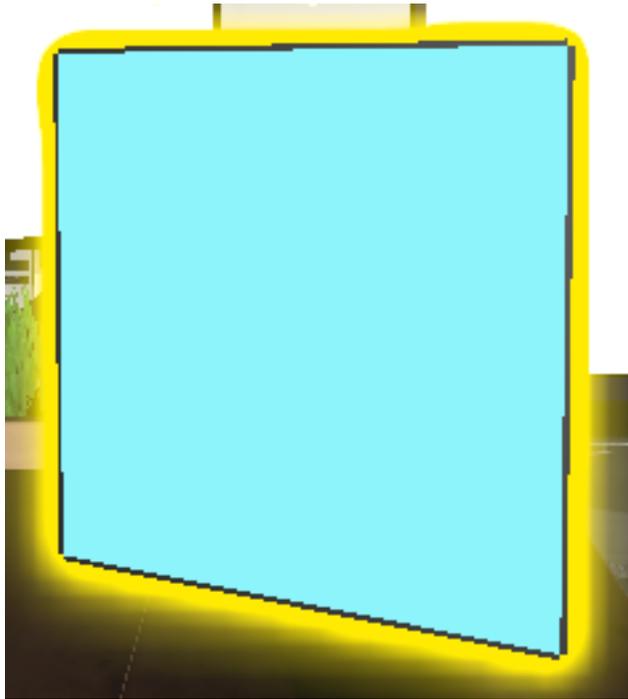


Abbildung 3.41: Demonstration des bereitgestellten Highlight-Effektes der Klasse "GameHighlighter".

Die Blink-Effekt-Komponente bietet eine Funktion zum Deaktivieren. Hierbei ist es wichtig, dass die jeweiligen Komponenten, die ein- und ausgeschaltet wurden, wieder eingeschaltet werden, abgesehen von der Highlight-Komponente. Anschließend wird die Blink-Effekt-Komponente sowie eine gegebenenfalls existierende Highlight-Komponente zerstört, um die Anzahl der nicht verwendeten Komponenten zu reduzieren.

Um Ressourcen zu schonen, wird der Blink-Effekt nur für den Benutzer angezeigt, der die Aktion ausführt. Eine Ausnahme bildet jedoch die [Line Split-Aktion](#), auf die später näher eingegangen wird.

3.6.4.2 Kollisionserkennung

Für die Kollisionserkennung benötigen die Drawable-Typ-Objekte neben den Collidern auch sogenannte *Rigidbody*s. Diese ermöglichen die Interaktion mit der Unity-Physik, wie beispielsweise die Kollisionserkennung [Seifert und Wislaug \(2017\)](#). Für die Rigidbody-Komponente, die dem Drawable-Typ-Objekt hinzugefügt wird, muss die Eigenschaft "isKinematic" aktiviert werden, um die OnTriggerEvents zu ermöglichen. Während der Kollisionserkennung wird den Drawable-Typ-Objekten eine MonoBehaviour-Komponente namens "CollisionController" hinzugefügt. Diese enthält eine Variable, um den Kollisionsstatus zu verfolgen. Die CollisionController-Komponente registriert eingehende Kollisionen mit den Objekten, die sich in derselben Drawable-Holder-

Objekthierarchie befinden, über die "OnTriggerEnter"-Methode. Dies sind ausschließlich die Drawable-Seitenränder.

Die Drawable-Typ-Objekte können untereinander keine Kollision erzeugen, da die Rigidbody-Komponente dem Objekt nur hinzugefügt wird, wenn während der auszuführenden Aktion eine Kollision mit einem Seitenrand erkannt werden soll. Nach Abschluss der entsprechenden Aktion wird die Rigidbody-Komponente wieder entfernt, um Ressourcen zu sparen. Es ergibt keinen sinnvollen Mehrwert, Kollisionen zwischen den Drawable-Typ-Objekten zu erkennen.

Sobald eine Kollision registriert wird, wird die Variable des Status entsprechend gesetzt. Der Status bleibt solange bestehen, wie die Kollision mit dem Seitenrand besteht. Erst wenn die Kollision aufgelöst wird, wird der Status mittels "OnTriggerExit" zurückgesetzt.

3.6.4.3 *Nearest Point(s) - Berechnung für Linien*

Um den nächstgelegenen Punkt einer Linie zu einer gewählten Raycasting-Position zu ermitteln, wurde die statische Klasse "NearestPoints" bereitgestellt. Diese stellt drei statische Methoden bereit. Die erste Methode führt die eigentliche Berechnung durch, während die zweite und dritte Methode diese Berechnung verwenden.

Die erste Methode "GetNearestIndicies" sucht den nächstgelegenen Punkt in einer Liste von globalen Positionen. Die Überprüfung erfolgt über die "Distance"-Methode der Vektor3-Klasse von Unity, mit der die Distanz zwischen zwei Punkten gemessen wird. Die Suche der Methode durchläuft die komplette Liste von Positionen und sucht nach der geringsten Distanz. Wenn mehrere Punkte die aktuell geringste Distanz aufweisen, werden sie zu den gefundenen Indizes hinzugefügt. Sobald eine noch geringere Distanz gefunden wird, werden die bereits ermittelten beziehungsweise gespeicherten Indizes gelöscht und die neue hinzugefügt.

Die zweite Methode namens "GetNearestPoint" wird für Mind Map-Knoten verwendet, um die kürzeste Verbindungslinie zwischen zwei Knoten zu ermitteln. Dazu wandelt die Methode die Positionen der Knotenumrandung zunächst in globale Positionen um. Dies ist notwendig, da die lokale Position abhängig von der Position des Elternobjektes ist. Nachdem die Positionen transformiert wurden, wird die Methode "GetNearestIndicies" mit dieser transformierten Liste und dem Punkt, zu dem der nächstgelegene Punkt ermittelt werden soll, aufgerufen. Der erste Index der gefundenen Indizes wird durch diese "GetNearestPoint"-Methode zurückgegeben.

Durch die dritte Methode "GetNearestPoints" werden die Positionen einer Linie in globale Positionen transformiert. Anschließend wird mittels der Methode "GetNearestPoints" die nächstgelegenen Punkte

zu einem gegebenen Punkt gesucht. Die Indizes aller gefundenen Punkte werden als Ergebnis bereitgestellt.

3.6.4.4 *Edit-Aktion*

Durch diese Aktion wird es ermöglicht, die Einstellungen der Drawable-Typ-Objekte zu editieren. Dazu muss der Benutzer ein geeignetes Objekt mittels eines linken Mausklicks auswählen. Dabei wird für das ausgewählte Objekt der Blink-Effekt aktiviert und zwei Drawable-Typ-Konfigurationen angelegt. Die eine enthält die originalen Einstellungen und die andere die Einstellungen nach der durchgeführten Änderung. Beide werden für das spätere Memento benötigt. Nachdem der Benutzer die linke Maustaste losgelassen hat, wird das entsprechende Edit-Menü für den ausgewählten Drawable-Typ geöffnet. Für jeden Typ werden aufgrund ihrer verschiedenen Einstellungen verschiedene Menüs benötigt. Änderungen, die durch das bereitgestellte Menü vorgenommen werden, werden in der entsprechenden Konfiguration gespeichert. Das Editieren kann mittels eines linken Mausklicks beendet werden. Dabei kann auch direkt ein neues Objekt zum Editieren ausgewählt werden. Zudem kann das Editieren mittels der "ESC"-Taste abgebrochen werden. Wenn der Benutzer die Aktion abbricht oder sie während der Ausführung verlässt, werden die durchgeführten Änderungen rückgängig gemacht. Zudem wird, wie auch bei dem korrekten Beenden der Aktion, der Blink-Effekt sowie das geöffnete Menü zerstört. Nachdem der Benutzer die Aktion korrekt beendet hat, wird anhand der zuvor beschriebenen Konfigurationen mittels der Methode "CheckEquals" geprüft, ob Änderungen vorgenommen wurden. Wenn dies der Fall ist, wird die Aktion abgeschlossen und ein Memento über das gewählte Objekt, dessen Drawable und die beiden Konfigurationen angelegt. Anderenfalls wird die Aktion zurückgesetzt.

Im Folgenden wird erläutert, wie die Edit-Menüs der einzelnen Drawable-Typen aufgebaut sind und welche Änderungen vorgenommen werden können.

- **Line:**
Für diesen Drawable-Typ wird das bereits bekannte Line-Menü verwendet. Zusätzlich zu den bereits vorgestellten einstellbaren Konfigurationen lässt sich mittels diesem noch die Loop-Funktionalität durch den Switch-Button aktivieren oder gegebenenfalls deaktivieren. Außerdem kann die Ebenenreihenfolge über einen Slider festgelegt werden. Hierbei ist 0 das Minimum und die aktuelle höchste Ebenenreihenfolge + 1 das Maximum. Das Edit-Menü für Linien wird in der folgenden [Abbildung 3.42](#) dargestellt.

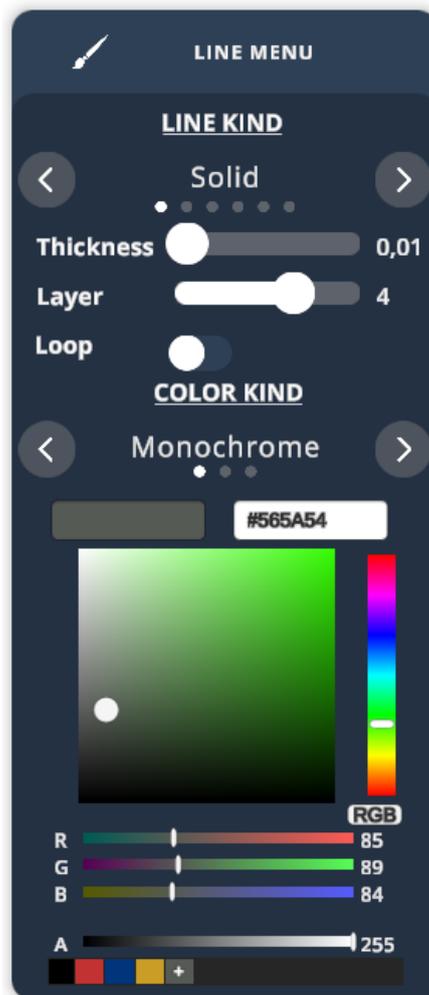


Abbildung 3.42: Demonstration des Edit-Menüs für Linien.

- **Text:**

Das Edit-Menü für Texte entspricht dem bereits bekannten Text-Menü, jedoch in erweiterter Form. In der [Abbildung 3.43](#) ist dieses dargestellt. Neben den bereits bekannten Konfigurationen wird hierbei das Editieren des Textes mittels des Buttons "Edit Text" bereitgestellt. Durch einen Klick darauf wird der Eingabedialog für Texte geöffnet und der bisherige Text in das Dialogfeld geladen. Außerdem ist die Ebenenreihenfolge, wie bei Line beschrieben, über ein Slider konfigurierbar.

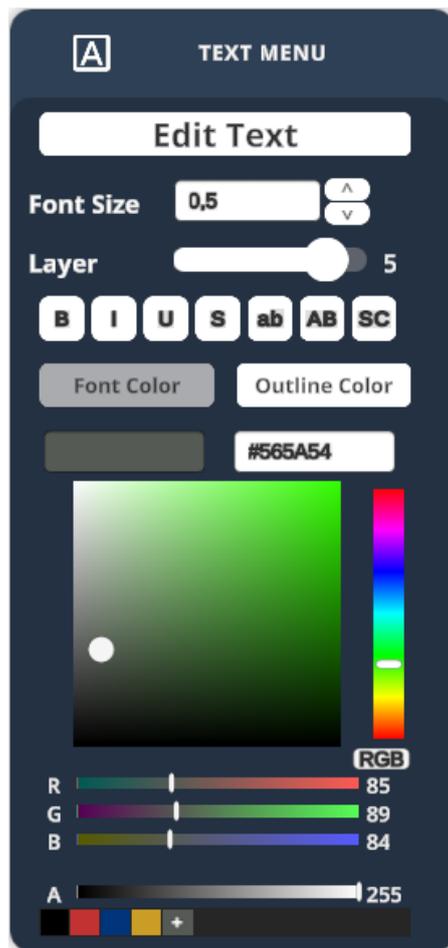


Abbildung 3.43: Demonstration des Edit-Menüs für Texte.

- **Image:**

Das Edit-Menü für Bilder ist in der [Abbildung 3.44](#) veranschaulicht. Dieses Menü stellt ein Miniaturbild des ausgewählten Bildes in seinem originalen unveränderten Zustand dar. Im HSV-Color-Picker-Feld ist zu erkennen, dass das ausgewählte Bild rötlich dargestellt wird, dieses Bild stellt das rechte Bild der [Abbildung 3.17](#) dar. Zusätzlich lässt sich die Ebenenreihenfolge über den Slider, wie beschrieben, konfigurieren. Zudem kann das Bild über den Switch-Button gespiegelt werden. Außerdem kann eine Farbüberlagerung über das Farbfeld gewählt werden.

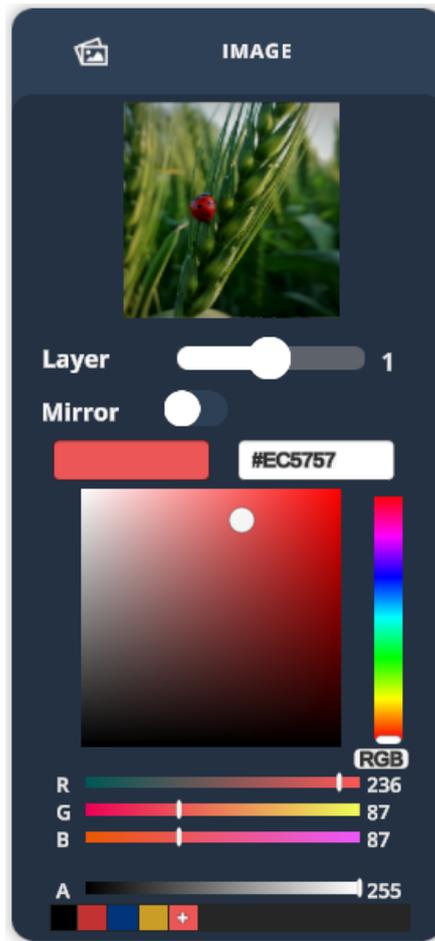


Abbildung 3.44: Demonstration des Edit-Menüs für Bilder.

- **Mind Map:**

Das bereitgestellte Menü für das Editieren von Mind Map-Knoten wird in der [Abbildung 3.45](#) dargestellt. Hierbei werden fünf verschiedene Optionen zum Editieren bereitgestellt. Der Button "Branch Line" für die Verbindungslinie zum Elternknoten steht nur zur Verfügung, wenn der ausgewählte Knoten einen Elternknoten besitzt. Anderenfalls ist dieser Button deaktiviert.

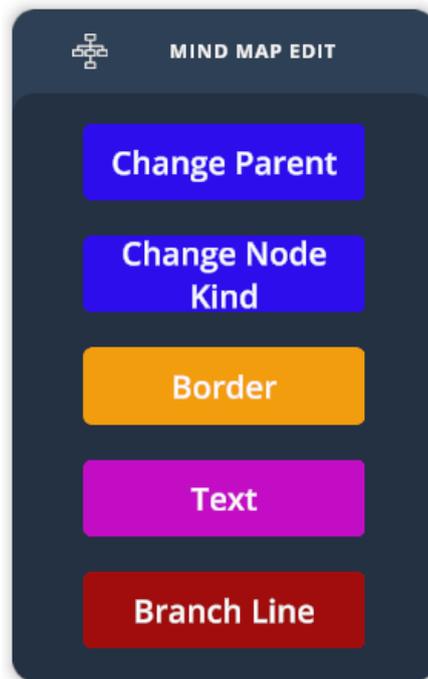


Abbildung 3.45: Demonstration des Edit-Menüs für Mind Map-Knoten.

Durch den Button “Change Node Kind“ wird das Menü, wie in der [Abbildung 3.46](#) dargestellt, eingeblendet, mit dem sich die Knotenart des gewählten Knotens ändern lässt. Hierbei kann ein Knoten nur zu einem Blatt umgewandelt werden, wenn dieser keine Kinderknoten aufweist. Zudem kann ein zentrales Thema nur umgewandelt werden, wenn ein anderes zentrales Thema auf dem Drawable existiert, das als neues Hauptthema fungieren kann. Wenn durch das Wechseln kein Elternknoten ausgewählt ist, wird das Elternauswahl-Menü, wie in der [Abbildung 3.38](#) dargestellt, für Mind Map-Knoten bereitgestellt. Bei der Auswahl des Elternknotens wird verhindert, dass Zyklen entstehen können. Zudem bleibt das Aussehen der Verbindungslinie beim Wechsel erhalten. Mit dem in der Abbildung ersichtlichen Button “Return to Mind Map Edit Menu“ kann zu dem Edit-Menü für Mind Maps zurückgekehrt werden, um weitere Änderungen vorzunehmen. Dieser Button wird auch für alle anderen bereitgestellten Menüs dieses Edit-Menüs mit eingebunden.

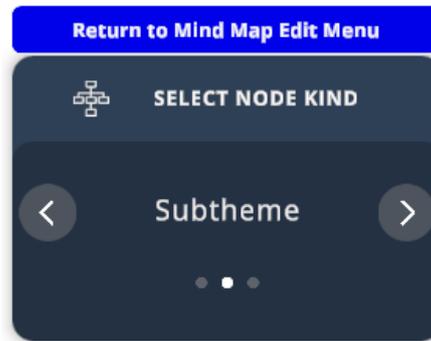


Abbildung 3.46: Demonstration des Edit-Menüs für die Knotenart eines Mind Map-Knotens.

Mit einem Klick auf den "Change Parent"-Button wird das Elternauswahl-Menü der [Abbildung 3.38](#) eingebunden. Hiermit kann, wie beschrieben, der Elternknoten ausgewählt werden, wobei nur Knoten angeboten werden, mit denen kein Zyklus entsteht. Wenn dieser Button gedrückt wird, während ein zentrales Thema ausgewählt ist, wird eine Warnung angezeigt, dass ein zentrales Thema keinen Elternknoten haben kann.

Mit dem "Border"-Button wird das Edit-Menü für Linien der [Abbildung 3.42](#) eingebunden, mit dem der Knotenrand verändert werden kann. Durch den "Text"-Button wird das Edit-Menü für Texte der [Abbildung 3.43](#) bereitgestellt, mit dem der Knotentext konfiguriert werden kann. Mittels des "Branch Line"-Buttons wird das Editieren der Verbindungslinie zum Elternknoten unterstützt. Hierzu wird das Edit-Menü für Linien eingebunden.

Für die drei zuletzt genannten Editiermöglichkeiten wird die Änderung der Ebenenreihenfolge nicht unterstützt, da die einzelnen Teilkomponenten keine unterschiedliche Ebenenreihenfolge aufweisen sollen. Zudem berechnet sich die Ebenenreihenfolge der Verbindungslinie anhand der Reihenfolge der beiden Knoten, die sie verbindet, und soll deshalb auch nicht durch den Benutzer veränderbar sein.

Für alle vier Drawable-Typen kann mittels der Undo-Funktion der originale Zustand des editierten Objektes wiederhergestellt werden. Dafür wird die von der DrawableType-Klasse bereitgestellte Methode "Edit" verwendet. Diese Methode prüft, um welchen Drawable-Typ es sich bei dem Objekt handelt und führt die entsprechenden Edit-Funktionen für diesen Drawable-Typ aus. Durch Redo werden die rückgängig gemachten Änderungen mittels derselben Methode wiederhergestellt.

3.6.4.5 Move or Rotate-Aktion

Diese Aktion besteht aus zwei Teilaktionen. Zum einen ist dies "Move", um die Drawable-Typ-Objekte zu bewegen, und zum anderen "Rotate", um diese Objekte zu rotieren. Sie werden zusammen in einer Aktion bereitgestellt, um ein schnelles Bewegen des Objektes zu ermöglichen, nachdem es gegebenenfalls durch die [Seitenränder-Funktionalität](#) beim Rotieren verschoben wurde. Eine Besonderheit für die Aktion stellen die Mind Map-Knoten dar. Für diese wird in beiden Menüs eine Funktionalität "Include Children" angeboten. In den beiden entsprechenden Abbildungen der Menüs wird diese durch eine grüne Box dargestellt. Diese Funktionalität wird nur eingeblendet, wenn das ausgewählte Objekt ein Mind Map-Knoten ist. Wenn diese Einstellung aktiviert wird, werden die Kinderknoten dem ausgewählten Knoten in der Hierarchie des GameObjects untergeordnet, um das zuvor beschriebene Parenting zu verwenden. Zusätzlich wird für die Kinderknoten die Kollisionserkennung aktiviert. Beim Deaktivieren werden die Kinderknoten wieder dem AttachedObject-Objekt untergeordnet und die Kollisionserkennung entfernt.

Als erstes muss der Benutzer ein geeignetes Drawable-Typ-Objekt mit einem linken Mausklick auswählen. Dadurch wird das in der folgenden [Abbildung 3.47](#) dargestellte Menü zur Wahl der gewünschten Teilaktion bereitgestellt. Zudem wird der Blink-Effekt für das ausgewählte Objekt aktiviert. Nachdem der Benutzer eine Auswahl getroffen hat, wird die gewünschte Teilaktion aufgerufen.

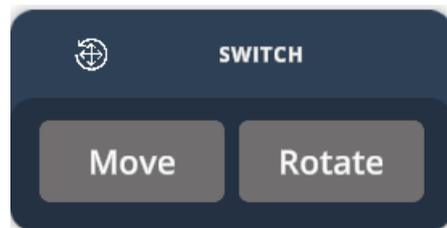


Abbildung 3.47: Demonstration des Auswahl-Menüs für Move oder Rotate.

Die ausgeführte Teilaktion kann mit einem linken Mausklick abgeschlossen werden. Dadurch wird ein Memento über das Objekt, das dazugehörige Drawable, die alte sowie neue Position und Rotation, die ausgeführte Teilaktion und ob Kinderknoten inkludiert wurden, angelegt. Des Weiteren wird das jeweilige Menü und der Blink-Effekt des Objektes zerstört.

Die ausführende Aktion kann durch die "ESC"-Taste abgebrochen werden, wodurch die bisherigen Änderungen rückgängig gemacht und die Menüs zerstört werden. Dasselbe geschieht, wenn der Benutzer die

Aktion wechselt, obwohl sie noch nicht abgeschlossen war.

Mit Undo wird die vorherige Position oder Rotation, je nach ausgeführter Teilaktion, wiederhergestellt. Mit Redo wird die rückgängig gemachte Änderung wiederhergestellt, sodass die neue Position oder Rotation erneut verwendet wird.

Move:

Wenn der Benutzer diese Teilaktion ausgewählt hat, wird das in der [Abbildung 3.48](#) dargestellte Move-Menü eingebunden. Dieses Menü ermöglicht das Bewegen über die vier Richtungstasten. Zudem kann die Geschwindigkeit mittels des Switch-Buttons erhöht werden. Durch den Value-Holder werden hierzu zwei verschiedene Geschwindigkeiten bereitgestellt. Zum einen die normale Geschwindigkeit mit der Einstellung "move" und zum anderen die schnelle Geschwindigkeit über "moveFast". Außerdem kann über den Switch-Button das Bewegen mittels der Maus beziehungsweise anhand des Raycasting-Strahls aktiviert werden. Abschließend bietet das Menü ein Switch-Button für das Aktivieren der bereits beschriebenen Funktion, um die Kinderknoten miteinzubeziehen.

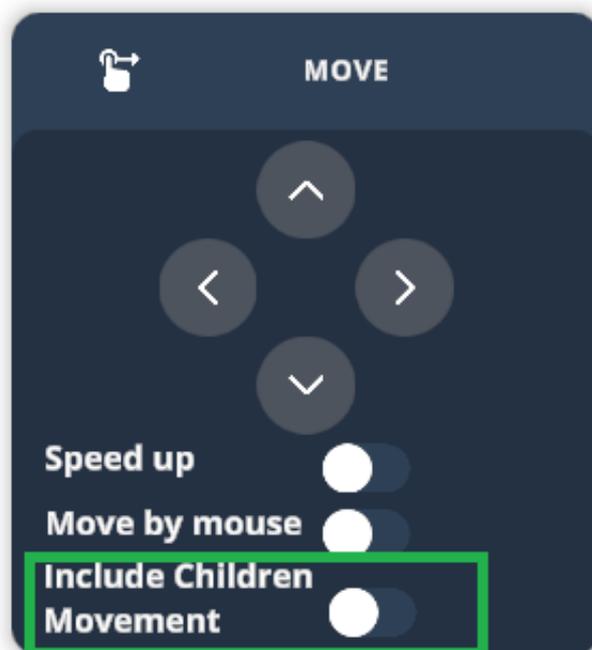


Abbildung 3.48: Demonstration des Move-Menüs.

Neben dem Menü wird das Bewegen auch über die Pfeiltasten bereitgestellt. Außerdem kann mittels eines Mauseklicks die Einstellung "Move by mouse" aktiviert oder deaktiviert werden und mit einem "Strg"-Klick die Einstellung "Speed up" ein- oder ausgeschaltet werden.

Mit der Option des Bewegens über die Richtungs- und Pfeiltasten wird das gewählte Objekt in die entsprechende Richtung über die eingestellte Geschwindigkeit verschoben. Außerdem wird beim Drücken einer der vier Tasten die Einstellung "Move by mouse" deaktiviert. Bei der Option der Mausbewegung wird die Position auf die umgewandelte lokale Position der Mausposition (Raycasting) gesetzt. Sobald eine Kollision erkannt wird, wird das Bewegen solange unterbrochen, bis die Kollision beseitigt wurde.

Rotate:

Nach der Auswahl dieser Teilaktion wird das Rotate-Menü der [Abbildung 3.49](#) bereitgestellt. Über einen Slider kann das ausgewählte Objekt im Bereich von 0 bis 359,9 Grad in 0,1 Schritten rotiert werden. Außerdem wird ein Eingabefeld zur Verfügung gestellt, über welches das Objekt während der Eingabe direkt rotiert werden kann. Durch die bereits beschriebene Funktion "Include Children" werden bei Mind Map-Knoten auch die Kinderknoten mit rotiert. Die Besonderheit hierbei ist, dass die Kinderknoten bei Aktivierung dieser Einstellung an die bereits eingestellte Rotation des ausgewählten Knotens angepasst werden.

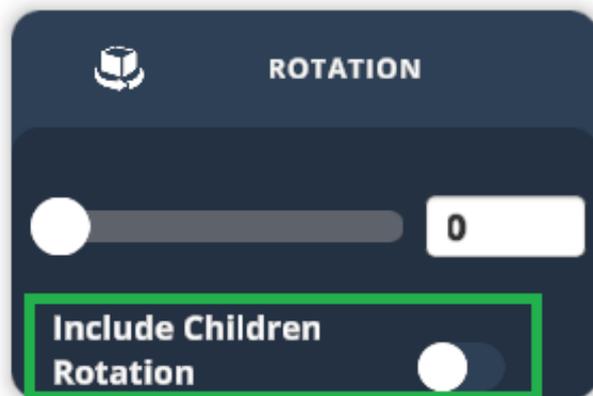


Abbildung 3.49: Demonstration des Rotate-Menüs.

Neben dem Menü wird auch die Möglichkeit des Rotierens über das Mousrad angeboten. Hierbei werden erneut zwei verschiedene Geschwindigkeiten mittels des Value-Holders zur Verfügung gestellt. Zum einen "rotate" für eine Rotation um einen Grad und "rotateFast" für eine Rotation um zehn Grad. Das schnelle Rotieren wird ermöglicht, indem während des Drehens des Mousrads in die entsprechende Richtung zusätzlich die "Strg"-Taste gedrückt gehalten wird.

Bei allen verfügbaren Möglichkeiten zum Rotieren wird das Objekt um den entsprechenden Gradwert rotiert. Das bedeutet, dass wenn das Objekt bereits um 50 Grad rotiert war und nun auf den Gradwert von 80 Grad rotiert wird, wird das Objekt um 30 Grad rotiert. Dabei

erfolgt die Rotation um den Pivot-Punkt (Mittelpunkt) des Objektes. Zusätzlich wird bei der Rotation für das Memento auch die Position des Objektes gespeichert. Dies ist notwendig, da durch die [Seitenränder-Funktionalität](#) das Objekt möglicherweise verschoben wurde.

3.6.4.6 Scale-Aktion

Diese Aktion ermöglicht das Skalieren eines Drawable-Typ-Objektes. Dabei können die X-Achse für die Breite und der Y-Achse für die Höhe skaliert werden. Aufgrund der Tatsache, dass die Drawable-Typ-Objekte als 2D-Objekte dargestellt werden, wird das Skalieren entlang der Z-Achse nicht unterstützt.

Zunächst wird mittels des Raycasting-Verfahrens ein entsprechendes Objekt zum Skalieren ausgewählt. Dabei wird mittels "hasDrawable" überprüft, ob das Objekt einem Drawable zugeordnet ist. Wenn dies der Fall ist, wird die Kollisionserkennung sowie der Blink-Effekt für das ausgewählte Objekt aktiviert und das in der [Abbildung 3.50](#) dargestellte Scale-Menü eingeblendet.

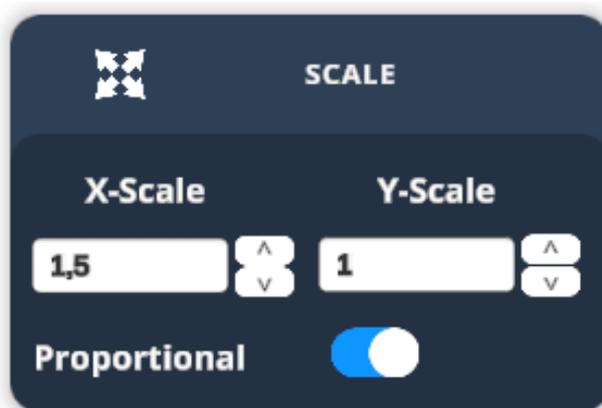


Abbildung 3.50: Demonstration des Scale-Menüs.

Im Scale-Menü kann der Benutzer mittels eines Switch-Buttons die Einstellung festlegen, ob die Skalierung des Objektes proportional erfolgen soll. Wenn diese Einstellung nicht aktiviert ist, wird unproportional skaliert, was bedeutet, dass der X-Achsen-Wert und der Y-Achsen-Wert unabhängig voneinander skaliert werden können. Bei proportionaler Skalierung werden sowohl der X-Achsen-Wert als auch der Y-Achsen-Wert um denselben Wert verändert. Wenn beispielsweise der X-Achsen-Wert 1,5 und der Y-Achsen-Wert 1 ist und nun der X-Achsen-Wert auf 3 gesetzt wird, also um 1,5 erhöht, wird der Y-Achsen-Wert ebenfalls um 1,5 erhöht und entsprechend auf 2,5 gesetzt. Die nachfolgende [Abbildung 3.51](#) zeigt zwei Skalierungsbeispiele anhand von drei gleichen Quadrat-Objekten. Das linke Objekt wurde nicht skaliert und weist demnach eine Skalierung von (1, 1, 1) auf. Das mittlere Objekt wurde

proportional skaliert und hat eine Skalierung von (1.5, 1.5, 1). Das rechte Objekt wurde unproportional über die Breite skaliert und weist den Wert (1.5, 1, 1) auf.

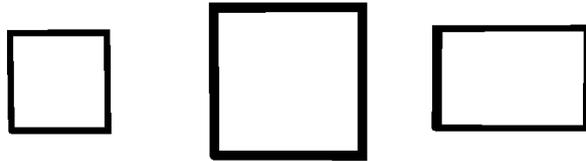


Abbildung 3.51: Demonstrationsbeispiel für proportionales und unproportionales Skalieren.

Die Skalierung kann über die jeweiligen Eingabefelder oder über die Auf- und Ab-Buttons erfolgen. Mittels der Auf- und Ab-Buttons kann der Wert mit einem linken Mausklick der Wert um 0,01 in die gewählte Richtung verändert werden, während bei gedrückter rechter Maustaste der Wert fortlaufend um 0,01 in die gewählte Richtung verändert wird, solange die Maustaste gedrückt gehalten wird.

Des Weiteren kann mittels des Mousrads ebenfalls proportional skaliert werden. Wenn dabei die "Strg"-Taste gedrückt gehalten wird, wird statt mit einem Skalierungswert von 0,01 mit einem von 0,1 zum Skalieren verwendet.

Jede ausgeführte Skalierung wird in Echtzeit mittels der "ScaleNetAction" über das Netzwerk ausgeführt. Zudem kann der auszuführende Benutzer den Skalierungsvorgang mittels der "ESC"-Taste abbrechen.

Eine Besonderheit beim Skalieren eines Mind Map-Knotens ist, dass die Verbindungslinien zu anderen Knoten neu gezeichnet werden, wenn der Knoten skaliert wird. Dies geschieht, da die Verbindungslinien sonst nicht exakt den Knotenrand tangieren.

Das Skalieren wird mit einem linken Mausklick außerhalb des Scale-Menüs beendet. Dabei kann der Benutzer auch direkt ein neues Objekt zum Skalieren auswählen, wobei das neue Objekt nicht das vorherige sein darf, um zu verhindern, dass versehentlich dasselbe Objekt ausgewählt wird, obwohl die Aktion eigentlich beendet werden sollte. Beim Beenden der Aktion wird ein Memento angelegt, welches das skalierte Objekt, die Werte der alten sowie neuen Skalierung und das Drawable, auf dem das Objekt platziert ist, festhält. Zusätzlich wird die Aktion abgeschlossen und das Scale-Menü sowie die Kollisionserkennung und der Blink-Effekt zerstört.

Wenn der Benutzer während des Skalierens die Aktion wechselt, werden jegliche Änderungen an der Skalierung des Objektes zurückgesetzt und das Scale-Menü zerstört.

3.6.4.7 *Change The Sorting Layer-Aktion*

Durch diese Aktion wird das Ändern der Ebenenreihenfolge eines Drawable-Typ-Objektes ermöglicht. Dafür kann der Benutzer mittels eines linken Mausklicks die Ebenenreihenfolge eines Drawable-Typ-Objektes um 1 erhöhen und mit einem rechten Mausklick um 1 verringern. Dabei wird das zu ändernde Objekt mittels des Raycasting-Verfahrens erkannt, und zusätzlich wird mittels "hasDrawable" überprüft, ob das Objekt einem Drawable zugeordnet werden kann. Darüber hinaus wird überprüft, ob das erkannte Objekt die Komponente "OrderInLayerValueHolder" aufweist. Sollte die Komponente nicht vorhanden sein oder kein Drawable vorhanden sein, kann die Aktion nicht ausgeführt werden.

In der nachfolgenden [Abbildung 3.52](#) kann eine Demonstration der Änderung der Ebenenreihenfolge nachvollzogen werden. Im ersten Schritt weist die schwarze Linie eine niedrigere Ebenenreihenfolge als die rote Linie auf. Zum Übergang zum zweiten Schritt wird die Ebenenreihenfolge der schwarzen Linie um 1 erhöht. Dadurch weisen beide Linien dieselbe Ebenenreihenfolge auf und überlappen sich deshalb gegenseitig. Für den dritten Schritt wurde die Reihenfolge der schwarzen Linie erneut erhöht, wodurch sie eine höhere Reihenfolge als die rote Linie aufweist.

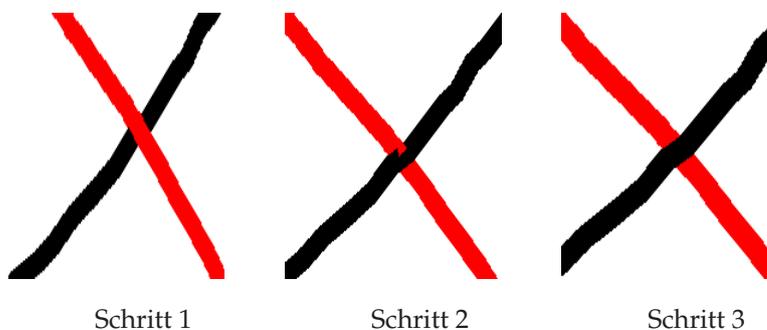


Abbildung 3.52: Demonstration der Änderung der Ebenenreihenfolge.

Schritt 1: Die schwarze Linie hat eine niedrigere Ebenenreihenfolge als die rote Linie.

Schritt 2: Beide Linien haben dieselbe Ebenenreihenfolge und überlagern sich deshalb gegenseitig im Schnittpunkt.

Schritt 3: Die schwarze Linie hat eine höhere Ebenenreihenfolge als die rote Linie.

Der ausführende Benutzer wird bei jeder Änderung mittels einer Benachrichtigung über die aktuelle Ebenenreihenfolge des Objektes informiert, abgesehen von der Undo- und Redo-Funktionalität. Eine

Demonstration hierfür ist in der [Abbildung 3.53](#) dargestellt, dabei wurde die Ebenenreihenfolge des Objektes inkrementiert.

Wenn das Objekt beim Dekrementieren bereits einen Ebenenreihenfolge von 0 aufweist und versucht wird, diese zu verringern, wird die in der [Abbildung 3.54](#) gezeigte Warnung angezeigt. Des Weiteren wird die Reihenfolge dann nicht verringert, da die Ebene 0 das Minimum darstellt.

Entsprechendes gilt auch für das Maximum der Ebenenreihenfolge. Das Maximum hierbei ist die Ebenenreihenfolge des zuletzt erstellten Objektes einer [Hinzufüge-Aktion](#) plus eins. Die [Abbildung 3.55](#) zeigt die entsprechende Warnung.



Abbildung 3.53: Demonstration der Benachrichtigung bei einer Änderung der Ebenenreihenfolge.

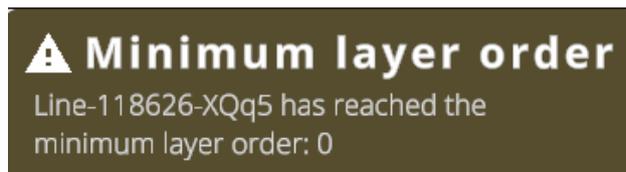


Abbildung 3.54: Demonstration der Warnung des Minimums.

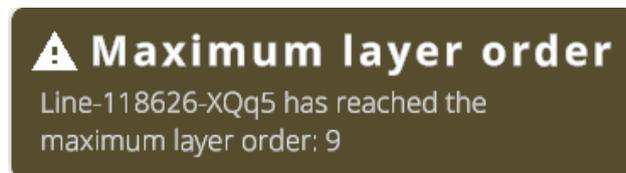


Abbildung 3.55: Demonstration der Warnung des Maximums.

Jede einzelne Änderung der Ebenenreihenfolge stellt dabei eine abgeschlossene Aktion dar, bei der ein Memento über die aktuelle und vorherige Ebenenreihenfolge, das Objekt und das dazugehörige Drawable sowie den Status, ob inkrementiert oder dekrementiert wurde, angelegt wird. Jede Änderung wird über die "LayerChangerNetAction" über das Netzwerk verteilt.

3.6.4.8 *Cut, Copy, Paste-Aktion*

Mittels dieser Aktion wird das Ausschneiden oder Kopieren und das anschließende Hinzufügen von Drawable-Typ-Objekten ermöglicht.

Das Hinzufügen kann auf einem beliebigen Drawable erfolgen. Zunächst muss der Benutzer ein Drawable-Typ-Objekt mit einem linken Mausklick auswählen. Dadurch wird das Cut, Copy, Paste-Menü der [Abbildung 3.56](#) geöffnet und der Blink-Effekt für das Objekt aktiviert. Mittels der "ESC"-Taste kann der Benutzer die Aktion abbrechen. Dabei wird der Blink-Effekt zerstört und das Menü geschlossen.

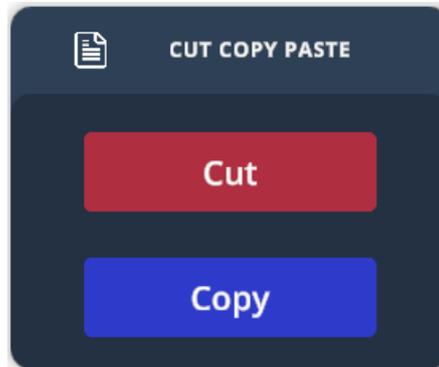


Abbildung 3.56: Demonstration des Cut, Copy, Paste-Menüs.

Im Menü hat der Benutzer die Auswahl zwischen "Copy" für das Kopieren und "Cut" für das Ausschneiden. Nachdem er eine Auswahl getroffen hat und eine geeignete Position auf einem beliebigen Drawable gefunden hat, kann er mittels eines linken Mausklicks das Objekt einfügen.

Sowohl bei dem Kopieren als auch dem Ausschneiden wird eine Kopie der Drawable-Typ-Konfiguration verwendet. Bei dieser wird der Name des Objektes entfernt und die Position durch die neu gewählte ersetzt. Durch das Entfernen des Namens wird dafür gesorgt, dass für dieses Objekt ein neuer Name vergeben wird. Somit wird ermöglicht, dass das Objekt mehrfach auf einem Drawable existieren kann, da jedes Objekt einen eindeutigen Namen aufweist. Eine Ausnahme bilden hierbei die Mind Map-Knoten. Diese werden nur umbenannt, wenn dies erforderlich ist.

Wenn die "Cut"-Option gewählt wurde, wird das ursprüngliche Objekt gelöscht. Bei Mind Map-Knoten wird das Objekt aus der Liste der Kinderknoten des Elternknotens entfernt und die Verbindungslinie zu diesem gelöscht, vorausgesetzt der Knoten besitzt einen Elternknoten. Bevor jedoch die Verbindungslinie gelöscht wird, wird eine Sicherung der Konfiguration angelegt, um die Optik dieser Linie später wiederherzustellen. Außerdem wird für den eingefügten Knoten das Mind Map Elternauswahl-Menü eingeblendet, wenn er vorher einen Elternknoten besaß. Hierbei wird die Optik mittels der eben erstellten Konfiguration wiederhergestellt. Nach den Vorgaben für Knoten aus dem [Mind Map-Drawable-Typ](#)-Abschnitt kann ein solcher Knoten nur auf einem

Drawable eingefügt werden, der ein Knoten mit der Knotenart "Theme" hat.

Zudem wird bei einem Mind Map-Knoten dessen Kinderknoten mit kopiert beziehungsweise ausgeschnitten, wenn er welche besitzt. Das Umbenennen der einzufügenden Knoten erfolgt dabei über die "RenameMindMap"-Methode der GameMindMap-Klasse. Sie sorgt dafür, dass alle umbenannten Knoten so verbunden bleiben, wie sie es vorher auch waren.

Damit ausschließlich die kopierten beziehungsweise ausgeschnittenen Knoten und deren Verbindungslinien in einer Konfiguration zusammengefasst werden können, wurde mittels der Methode "Summarize-SelectedNodeIncChildren" eine Drawable-Konfiguration erstellt, die nur diese Drawable-Typen-Konfigurationen enthält. Diese werden sowohl zum Erstellen der Knoten und Verbindungslinien als auch zum Rückgängigmachen und Wiederholen der Aktion benötigt.

Des Weiteren wird bei einem Mind Map-Knoten die Position der Konfiguration vor dem Erstellen nicht geändert. Der Knoten inklusive Kinderknoten wird nach dem Erstellen zu der ausgewählten Position verschoben. Dadurch wird ermöglicht, dass die Kinderknoten mit verschoben werden und das ursprüngliche Mind Map Erscheinungsbild um den ausgewählten Knoten erhalten bleibt.

Nach dem erfolgreichen Einfügen und gegebenenfalls Löschen des ausgewählten Objektes wird ein Memento über die Konfigurationen des alten sowie neuen Drawables, der alten und neuen Drawable-Typ-Konfiguration, der ausgeführten Operation (Cut oder Copy) sowie gegebenenfalls die alte und neue Drawable-Konfiguration, die nur die Mind Map-Daten enthält und die der alten Verbindungslinie zum ehemaligen Elternknoten, angelegt.

Undo löscht das neue Objekt und stellt bei der "Cut"-Operation das originale Objekt wieder her, bei Mind Map-Knoten wird die Optik der originalen Verbindungslinie ebenfalls wiederhergestellt. Redo stellt das neue Objekt wieder her und löscht gegebenenfalls das Originale wieder.

3.6.4.9 *Move a Point-Aktion*

Diese Aktion ermöglicht das Verschieben einzelner oder mehrerer Punkte einer Linie. Das Verschieben von mehreren Punkten wird nur unterstützt, wenn sie sich auf derselben Position befinden. Um einen Punkt zum Verschieben auszuwählen, muss der Benutzer mit einem linken Mausklick auf den gewünschten Punkt der Linie klicken. Mittels der "GetNearestPoints"-Methode werden der oder die nächstgelegenen

Punkte vom gewünschten Punkt zum Verschieben ermittelt. Außerdem wird der Blink-Effekt für die ausgewählte Linie aktiviert. Anschließend wird die Funktionalität zum Bewegen des Punktes aktiviert. Der Punkt folgt hierbei dem Raycasting-Strahl, solange sich dieser auf einem Drawable oder Drawable-Typ-Objekt befindet. Wenn der Zielpunkt erreicht wurde, kann das Verschieben mit einem erneuten linken Mausklick beendet werden. In der [Abbildung 3.57](#) wird ein verschobener Punkt dargestellt. Dabei stellt das linke Objekt der Abbildung den ursprünglichen Zustand dar und das rechte Objekt den verschobenen Zustand.

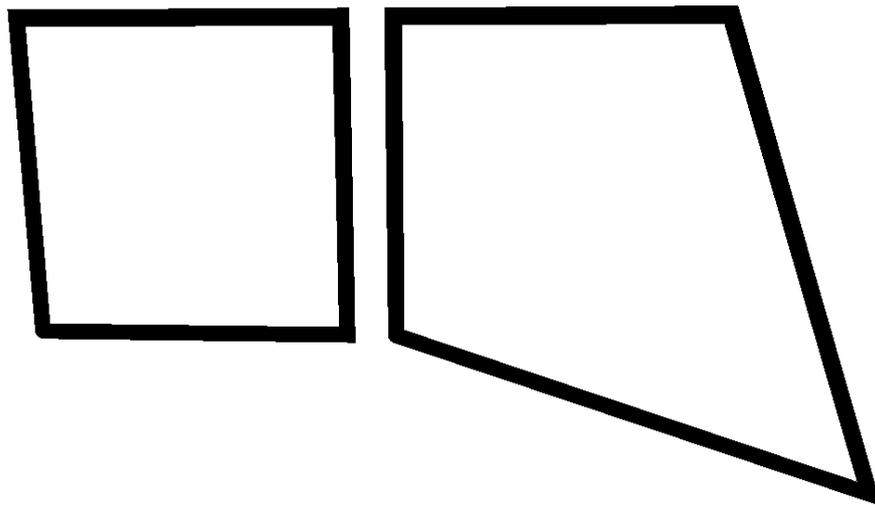


Abbildung 3.57: Demonstration des Verschiebens eines Punktes einer Linie.

Die nachfolgende [Abbildung 3.58](#) besteht aus einer Darstellung vor und nach der Aktion. Das dargestellte Objekt besitzt für die verschobene Position zwei Punkte. Diese verschobene Position ist in der Darstellung der ursprünglichen Objektform durch die Punkte **B** und **I** gekennzeichnet. Diese ursprüngliche Objektform wird in verschiedenen Aktionen zur Erläuterung der Anwendung auf mehreren Punkten verwendet. Die gezeichneten Linien umfasst dabei folgende Verbindungslinien (Strecken) zwischen den Punkten: \overline{AB} , \overline{BC} , \overline{CD} , \overline{DE} , \overline{EF} , \overline{FG} , \overline{GH} und \overline{HI} . Durch das Verschieben der zuvor beschriebenen Position werden die Strecken \overline{AB} , \overline{BC} , \overline{EF} , \overline{FG} verändert.

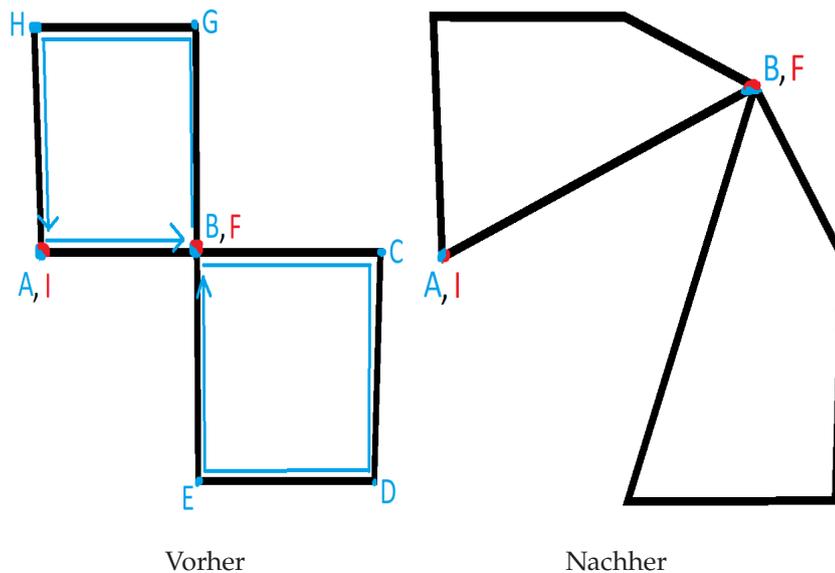


Abbildung 3.58: Anwendung der Move a Point-Aktion auf mehrere Punkte.

Die Aktion kann mittels der "ESC"-Taste während des Verschiebens einer Position abgebrochen werden. Wenn die Aktion abgebrochen wird oder der Benutzer während des Verschiebens die Aktion wechselt, werden die bisherigen Änderungen rückgängig gemacht. Nachdem der Benutzer das Verschieben beendet hat, wird die Aktion abgeschlossen, ein Memento erstellt und der Blink-Effekt zerstört. Das Memento umfasst die Linie, die Indizes der Punkte die verschoben wurden, die alte sowie neue Position der Punkte und das dazugehörige Drawable.

Mittels Undo werden der oder die Punkte in die ursprüngliche Position zurückverschoben. Mit Redo werden diese erneut auf die neue Position versetzt.

3.6.4.10 Line Split-Aktion

Durch diese Aktion ist es dem Benutzer möglich, eine Linie zu teilen. Dazu muss der Benutzer mit einem linken Mausklick auf die gewünschte Teilungsposition der Linie klicken. Mittels der "GetNearestPoints"-Methode werden anschließend der oder die nächstgelegenen Punkte (mit der gleichen Position) ermittelt. Durch die Methode "Split" der Klasse `GameLineSplit` wird die Linie an den ermittelten Punkten geteilt. Diese Methode stellt einen Parameter bereit, mit dem angegeben werden kann, ob die Teilungspunkte behalten oder verworfen werden sollen. Für diese Aktion wird behalten und für [Line Point Erase-Aktion](#) verworfen gewählt.

Durch das Teilen wird die originale Linie gelöscht und Teillinien erstellt. Die Teilungsposition wird, wie in der [Abbildung 3.59](#) dargestellt, mit einem Kreismarker markiert und mittels des Blink-Effekts kenntlich

gemacht. Der Marker verwendet die Komplementärfarbe der Linie, um deutlich erkannt zu werden. Die Komplementärfarbe ist dabei auch abhängig von der Farbart der Linie. Bei "Gradient" und "TwoDashed" wird jeweils der Mittelwert der Primär- und Sekundärfarbe für die Berechnung der Komplementärfarbe verwendet. Für Farben die keine Komplementärfarbe besitzen, wird die in der Value-Holder in der Einstellung "lineSplitDefaultMarkerColor" gespeicherte Farbe verwendet. Derzeit wurde hierfür Magenta gewählt. Nach einer gewissen Zeit, derzeit drei Sekunden, wird der Marker wieder gelöscht. Die Zeit kann mit der Value-Holder-Einstellung "lineSplitTime" konfiguriert werden. Damit dieser Marker auch über das Netzwerk hinweg blinkend angezeigt und anschließend gelöscht wird, wurden die Netzwerkaktionen "AddBlinkEffectNetAction" und "EraseAfterTimeNetAction" angelegt und verwendet.

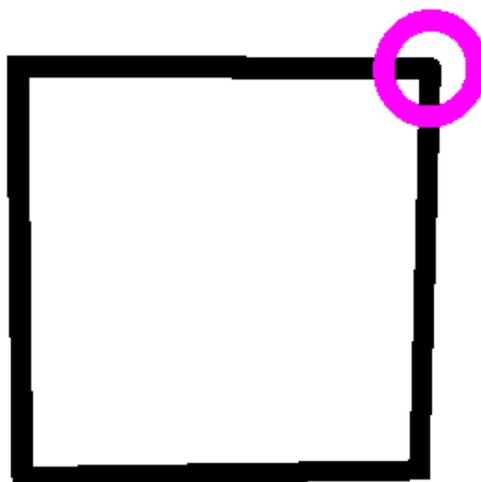


Abbildung 3.59: Demonstration des Markierens des Spaltens einer Linie.

Der Start- und Endpunkt einer Linie kann logischerweise nicht geteilt werden. Für diese beiden Fälle werden entsprechende Informationen angezeigt. Nachdem die Linie erfolgreich geteilt wurde, wird der Benutzer darüber benachrichtigt. Zudem enthält diese Benachrichtigung die Information in wie vielen Teillinien die Linie aufgeteilt wurde.

Die folgende [Abbildung 3.60](#) zeigt das Aufteilen der in der [Move a Point-Aktion](#) vorgestellten Objektform, die an der Teilungsposition zwei Punkte enthält. Nach dem erfolgreichen Teilen entstehen drei Teillinien. Die erste Linie enthält die Strecke \overline{AB} , die zweite die Strecken von \overline{BC} bis einschließlich \overline{EF} und die dritte enthält die Strecken von \overline{FG} bis einschließlich \overline{HI} .

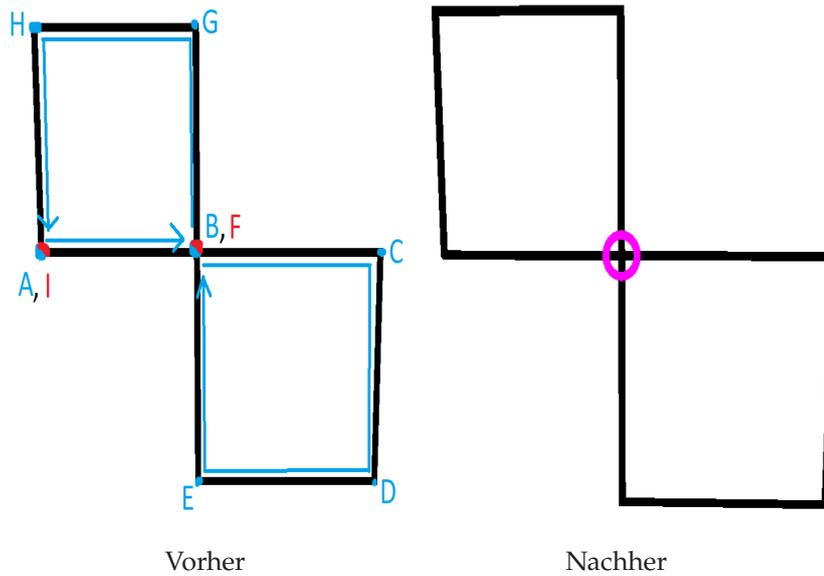


Abbildung 3.60: Anwendung der Line Split-Aktion auf mehrere Punkte.

Die Aktion wird abgeschlossen, sobald der Benutzer nach dem Teilen die linke Maustaste losgelassen hat. Zudem wird ein Memento über die Konfigurationen der originalen Linie, der entstandenen Teillinien sowie des dazugehörigen Drawables erstellt.

Mit Undo wird die originale Linie wiederhergestellt und die Teillinien gelöscht. Redo stellt die Teillinien wieder her und löscht die Originale.

3.6.5 Save-Aktion

Mit dieser Aktion wird das Speichern von Drawables ermöglicht. Dazu wird das Save-Menü mittels der Awake-Methode bereitgestellt. Eine Darstellung dieses Menüs ist in der nachfolgenden [Abbildung 3.61](#) zu sehen.

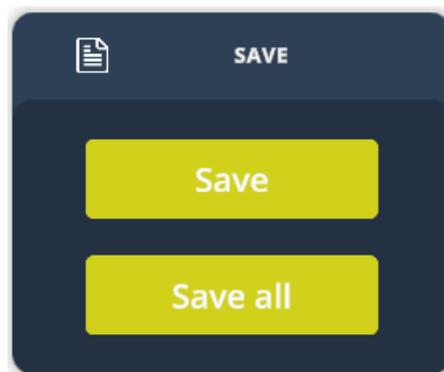


Abbildung 3.61: Demonstration des Save-Menüs.

Das Menü bietet dem Benutzer zwei Möglichkeiten zum Speichern. Zum einen kann er mit "Save all" alle Drawables des virtuellen Raumes speichern. Zum anderen kann er mit "Save" die zuvor mit einem linken Mausklick ausgewählten Drawables speichern. Falls kein Drawable ausgewählt wurde, wird eine entsprechende Information ausgegeben. Die zum Speichern ausgewählten Drawables werden mit einem Highlight-Effekt hervorgehoben, der sich optisch von den bisher verwendeten unterscheidet. Ein Beispiel für diesen Highlight-Effekt ist in der [Abbildung 3.62](#) dargestellt. Ein ausgewählter Drawable kann durch erneutes Anklicken mit einem linken Mausklick auch wieder abgewählt werden.

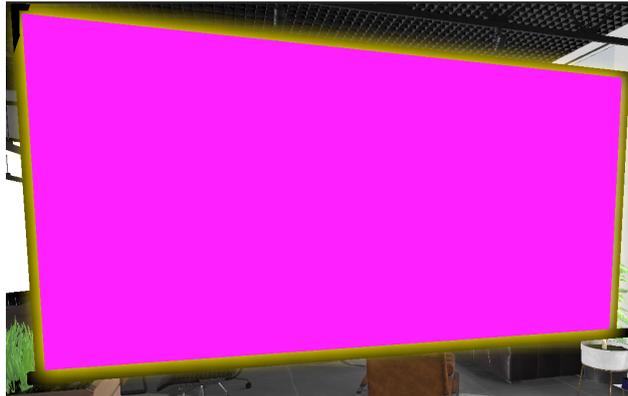


Abbildung 3.62: Demonstration des Highlight-Effektes für das Speichern und Laden.

Nachdem der Benutzer einen der beiden Buttons im Save-Menü gedrückt hat, öffnet sich der Filebrowser zum Speichern. Dabei wird zwischen zwei Verzeichnissen zum Speichern unterschieden. Ein Verzeichnis wird gewählt, wenn nur ein Drawable gespeichert werden soll, und das andere für den Fall, dass mehrere gespeichert werden sollen. Diese Verzeichnisse dienen lediglich als Vorschläge, der Benutzer kann über den Filebrowser auch zu einem anderen Verzeichnis wechseln und die Konfiguration dort speichern. Der geöffnete Filebrowser ist in der [Abbildung 3.63](#) dargestellt.

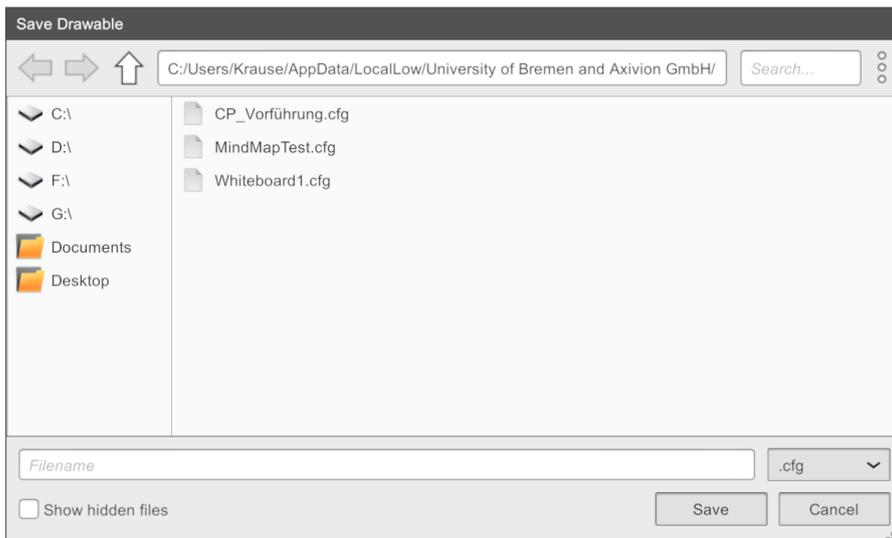


Abbildung 3.63: Demonstration des Filebrowsers für das Speichern.

Nachdem der Benutzer seine Eingabe mittels des "Save"-Buttons des Filebrowsers bestätigt hat, wird die Konfiguration des Drawables gespeichert. Hierzu wird die "Save"-Methode verwendet, die jede Konfigurationsklasse bereitstellt. Diese Methoden verwenden den von SEE bereitgestellten "ConfigWriter", um die Konfigurationsdatei zu erstellen und zu schreiben. Dabei werden die zu speichernden Daten in Schlüssel-Wert-Paare abgelegt. Die zuvor beschriebene Unterscheidung, ob ein oder mehrere Drawables gespeichert werden sollen, ist hierbei notwendig. Denn bei einem einzelnen Drawable wird die Methode "SaveDrawable" der DrawableConfigManager-Klasse verwendet, die wiederum zum Speichern die Klasse "DrawableConfig" verwendet. Wohingegen für mehrere Drawables die Methode "SaveDrawables" zum Speichern verwendet wird, die die Klasse "DrawableConfigs" verwendet. Die Save-Methode von DrawableConfigs iteriert über jeden zu speichernden Drawable und verwendet dessen DrawableConfig Save-Methode zur Erstellung. Die Save-Methode von DrawableConfig ruft für jedes auf dem Drawable vorhandene Drawable-Typ-Objekt dessen Save-Methode auf. Dadurch entsteht für die Save-Methode von DrawableConfigs eine Konfigurationsdatei, die zu jedem Drawable die Daten seiner Drawable-Typ-Objekte enthält. Bei einem einzelnen Drawable wäre dies folglich eine Konfigurationsdatei, die die Daten des Drawable sowie die Daten seiner Drawable-Typ-Objekte enthält.

Nachdem die Datei gespeichert wurde, wird die Aktion abgeschlossen, die Highlight-Effekte zerstört und ein Memento angelegt. Das Memento umfasst die Konfiguration der gespeicherten Drawables, den Dateipfad sowie den Status, ob ein, mehrere oder alle Drawables gespeichert wurden.

Für den Fall, dass der Benutzer Drawables ausgewählt hat und er die Aktion abbrechen möchte, kann er dies durch Drücken der “ESC“-Taste tun. Dadurch werden alle ausgewählten Drawables abgewählt und somit ihre Highlight-Effekte zerstört.

Durch die Undo-Funktionalität wird die angelegte Datei wieder gelöscht und mittels der Redo-Funktionalität wird sie wiederhergestellt.

3.6.6 Load-Aktion

Mit dieser Aktion wird es ermöglicht, die durch die [Save-Aktion](#) gespeicherte Konfigurationsdateien zu laden. Hierzu wird in der Awake-Methode dieser Aktion das in der [Abbildung 3.64](#) dargestellte Load-Menü bereitgestellt.

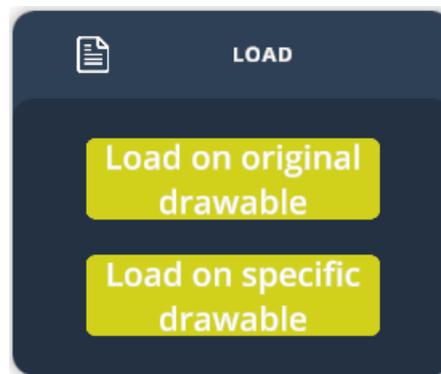


Abbildung 3.64: Demonstration des Load-Menüs.

Das Menü stellt zwei verschiedene Lademöglichkeiten zur Verfügung. Zum einen kann die Datei so geladen werden, wie sie gespeichert wurde. Hierbei werden die nicht vorhandenen Drawables als Sticky-Notes erstellt. Zum anderen kann mittels eines linken Mausklicks ein Drawable ausgewählt werden, um die Datei vollständig auf dem ausgewählten Drawable zu laden. Es kann immer nur ein Drawable ausgewählt sein. Wenn bereits eins ausgewählt ist und ein anderes ausgewählt wird, wird das bereits ausgewählte abgewählt. Sollte kein Drawable ausgewählt sein und auf den “Load on specific drawable“-Button geklickt werden, wird eine entsprechende Information angezeigt.

Nachdem einer der beiden Buttons gewählt wurde, wird das Memento für diese Aktion angelegt und ein Filebrowser verwendet, um eine Konfigurationsdatei zum Laden auszuwählen. Dieser Filebrowser ist ähnlich dem in der [Abbildung 3.63](#). Die gewählte Datei wird mittels der “LoadDrawables“-Methode des DrawableConfigManagers geladen und gelesen. Als nächstes müssen die Datensätze der geladenen Datei in Drawable-Konfigurationen (DrawableConfigs) umgewandelt werden. Die verschiedenen Konfigurationsklassen stellen hierfür die Methode

“bool Restore“ bereit. Diese sucht in dem eingelesenen Datensatz, der aus Schlüssel-Wert-Paaren besteht, nach den Schlüsseln der jeweiligen Konfiguration und sichert ihre dazugehörigen Werte, um die Konfiguration wiederherzustellen.

Nachdem die Konfigurationen der Drawable-Typen und der Drawables wiederhergestellt wurden, werden die entsprechenden Objekte zu den Drawable-Typ-Konfigurationen wiederhergestellt. Je nach gewählter Lademöglichkeit werden die Objekte entweder auf den bisherigen Drawables oder auf dem ausgewählten Drawable wiederhergestellt. Vor der Wiederherstellung wird geprüft, ob bereits ein gleichnamiges Objekt auf dem entsprechenden Drawable existiert. Wenn dies der Fall ist, wird das Objekt umbenannt. Für Mind Map-Knoten wird die in der [Cut, Copy, Paste-Aktion](#) beschriebene Methode “RenameMindMap“ der GameMindMap Klasse verwendet. Durch das Umbenennen wird ermöglicht, dass die geladene Datei erneut auf dem Drawable geladen werden könnte. Zum Wiederherstellen der Objekte wird die Methode “void Restore“ der DrawableType Klasse verwendet.

Nachdem die Objekte der Konfigurationen wiederhergestellt wurden, wird die Aktion abgeschlossen und fehlende Informationen in das Memento eingetragen. Das Memento dieser Aktion umfasst die Art, wie die Datei geladen wurde, die geladenen Konfigurationen sowie gegebenenfalls die Konfiguration des ausgewählten Drawables, auf dem die Datei wiederhergestellt wurde, und eine Liste der Drawables, die für das Laden erstellt wurden.

Mit Undo werden die geladenen Daten inklusive der gegebenenfalls erstellten Sticky-Notes wieder gelöscht. Durch Redo werden die durch Undo gelöschten Drawable-Typ-Objekte und Drawables wiederhergestellt. Die Aktion kann mit der “ESC“-Taste abgebrochen werden können, wodurch das ausgewählte Drawable abgewählt wird.

3.6.7 Lösch-Aktionen

In diesem Abschnitt werden die vier Lösch-Aktionen erläutert. Dazu wird zunächst auf die beiden Aktionen eingegangen, die nur für den [Line-Drawable-Typ](#) zur Verfügung stehen. Diese sind die [Line Point Erase-Aktion](#) und die [Line Connection Erase-Aktion](#). Abschließend werden die beiden Lösch-Aktionen für alle Drawable-Typen beschrieben. Diese umfassen die [Erase-Aktion](#) und die [Cleaner-Aktion](#).

3.6.7.1 Line Point Erase-Aktion

Mittels dieser Aktion können Punkte einer oder mehrere Linien während eines Aktionsdurchlaufes gelöscht werden. Um einen Punkt zu löschen, muss der Benutzer mittels der linken Maustaste auf den

gewünschten Punkt der Linie klicken. Mithilfe der Methode "GetNearestPoints" wird dann der oder die nächstgelegenen Punkte ermittelt. Anschließend wird die Linie an den ermittelten Punkten durch die Methode "Split" der Klasse GameLineSplit geteilt und gelöscht, indem der Teilungspunkt verworfen wird. Dadurch entstehen Teillinien, die den gelöschten Punkt nicht enthalten. Der Benutzer kann die linke Maustaste gedrückt halten und währenddessen weitere zu löschende Punkte auswählen. Für jeden gelöschten Punkt wird ein Memento über die Konfigurationen der Linie vor der Löschung, die Linien, die durch die Löschung entstanden sind, und dem dazugehörigen Drawable erstellt. Jedes dieser Mementos wird in einer Mementoliste, die diese Aktion bereitstellt, gespeichert. Zum Abschließen der Aktion muss der Benutzer die linke Maustaste loslassen. Ein Beispiel für diese Aktion kann in der nachfolgenden [Abbildung 3.65](#) betrachtet werden. Das linke Objekt stellt dabei den originalen Zustand dar und das rechte den Zustand nach der Löschung. Der zu löschende Punkt war der unten rechts.

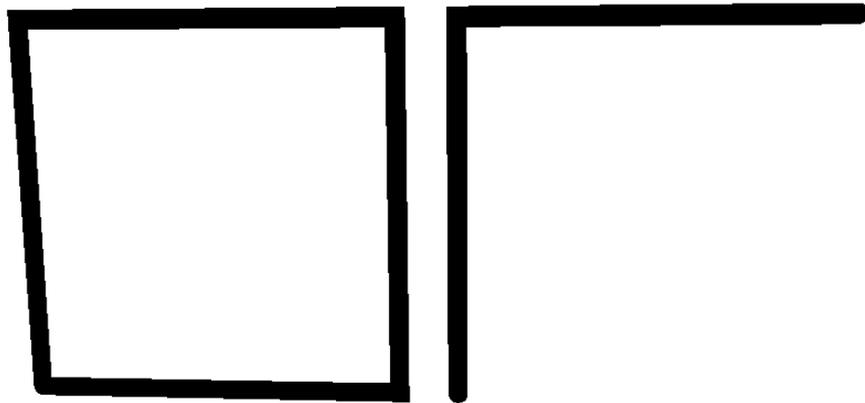


Abbildung 3.65: Demonstration des Löschens eines Punktes einer Linie.

Diese Aktion ist eher für frei gezeichnete Linien gedacht. Ein Beispiel dafür kann in der nachfolgenden [Abbildung 3.66](#) betrachtet werden. Hierbei wurden mehrere verschiedene Punkte mittels eines Durchlaufes entfernt.

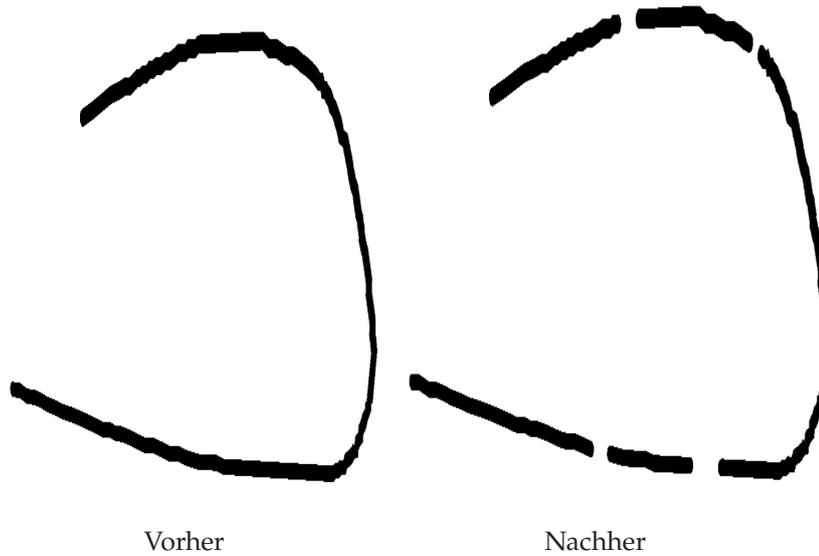


Abbildung 3.66: Anwendung der Line Point Erase-Aktion auf einer frei gezeichneten Linie.

In der folgenden [Abbildung 3.67](#) wird das Löschen mehrerer Punkte auf einer Position demonstriert. Es ist deutlich zu erkennen, dass jede Verbindungslinie zu der ausgewählten Position gelöscht wird. Dies umfasst die Strecken \overline{AB} , \overline{BC} , \overline{EF} und \overline{FG} .

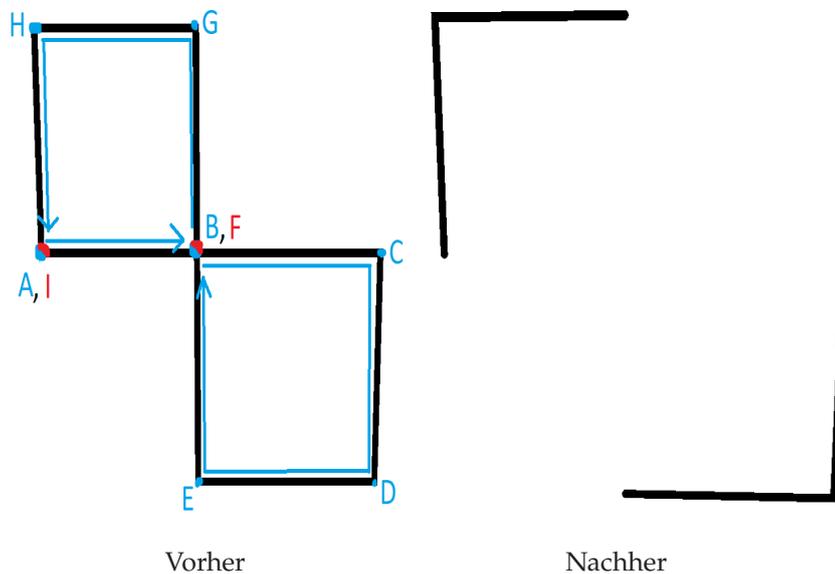


Abbildung 3.67: Anwendung der Line Point Erase-Aktion auf mehrere Punkte.

Mittels Undo können die originalen Linien wiederhergestellt werden. Zudem werden die durch die Löschung entstandenen Teillinien gelöscht. Dafür ist es notwendig, dass die Mementoliste in einer separaten Liste umgedreht und diese verwendet wird, bevor die Wiederherstellung startet. Es kann nämlich sein, dass der Benutzer weitere Punkte

der Linie und somit von einer der Teillinien gelöscht hat. Wenn die Liste nicht umgedreht werden würde, würden die zu löschenden Teillinien nicht gefunden werden und würden somit nicht gelöscht werden. Zur Erinnerung: Beim Teilen einer Linie wird die originale Linie gelöscht und Teillinien mit neuen Namen erstellt.

Bei der Redo-Funktionalität werden die Mementos von der ursprünglichen Mementoliste nach ihrer Erstellung abgearbeitet. Dabei werden die originale Linie gelöscht und die Teillinien erstellt.

3.6.7.2 *Line Connection Erase-Aktion*

Mit dieser Aktion wird das Löschen von einer Verbindungslinie zwischen zwei Punkten einer Linie ermöglicht. Ein Beispiel hierfür kann in der folgenden [Abbildung 3.68](#) betrachtet werden. Das linke Quadrat stellt dabei den originalen Zustand dar und das rechte den Zustand nach dieser Aktion.

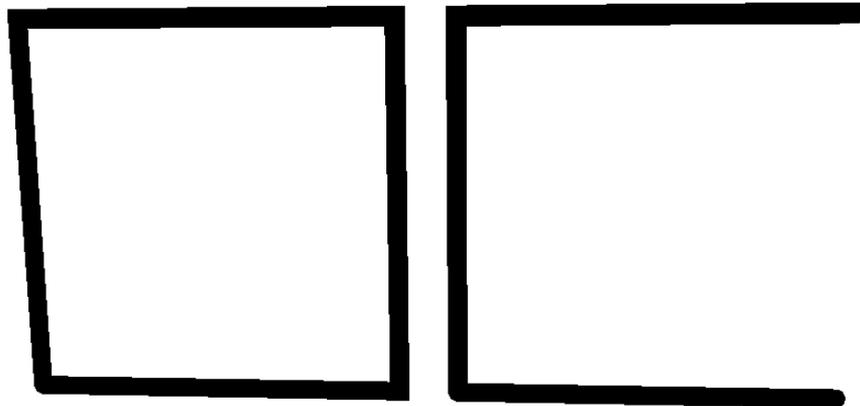


Abbildung 3.68: Demonstration des Löschens einer Verbindungslinie einer Linie.

Um dieses Resultat zu erzielen, muss der Benutzer mit einem linken Mausklick auf den gewünschten Punkt der Linie klicken, von dem die folgende Verbindungslinie entfernt werden soll. Für das Beispiel bedeutet dies, dass der Benutzer auf den Punkt unten rechts geklickt hat. Nachdem der Benutzer den Punkt ausgewählt hat, wird mittels "GetNearestPoints" die Punkte mit der nächstgelegenen Position ermittelt. Anschließend wird die Linie an der gefundenen Position mittels der Methode "EraseLinePointConnection" der statischen Klasse GameLineSplit aufgespalten und der Punkt bei allen Teillinien außer der ersten entfernt.

Logischerweise kann hierbei der Endpunkt der Linie nicht gewählt werden, da dieser keinen Folgepunkt aufweist und somit keine Verbindungslinie von ihm ausgeht. Sollte der Benutzer versehentlich den

Endpunkt wählen, wird eine entsprechende Warnung angezeigt.

Nachdem die Verbindungslinie, gegebenenfalls Verbindungslinien, entfernt wurden und der Benutzer die Maustaste losgelassen hat, wird ein Memento über die Konfigurationen der originalen Linie, der Teillinien und des dazugehörigen Drawables angelegt. Anschließend wird die originale Linie gelöscht und die Aktion abgeschlossen.

In der nachfolgenden [Abbildung 3.69](#) wird die Demonstration für das bekannte Beispiel vollzogen. Es wurde zur Ausführung der Aktion als Position erneut die des Punktes **B** und **F** ausgewählt. Der Abbildung kann entnommen werden, dass sowohl die Verbindungslinie \overline{BC} , die von dem Punkt **B** ausgeht, als auch die \overline{FG} , die vom Punkt **F** ausgeht, gelöscht wurden.

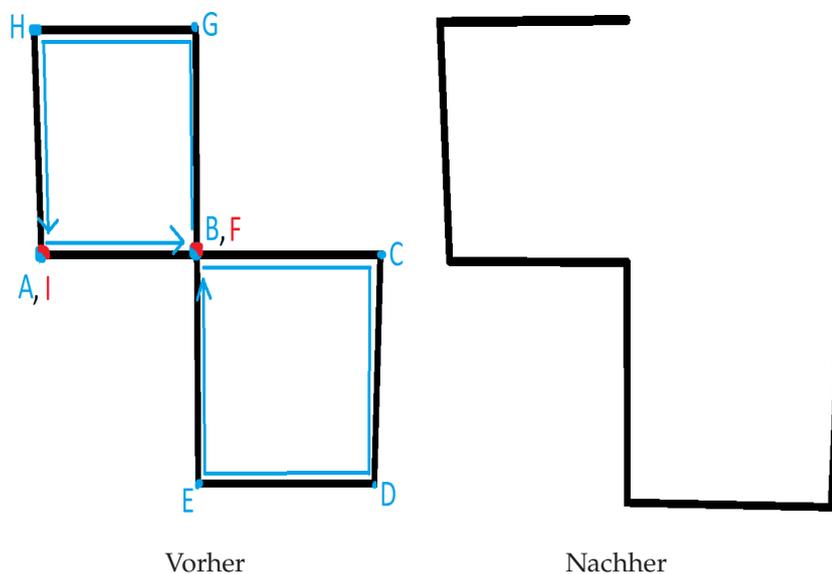


Abbildung 3.69: Anwendung der Line Connection Erase-Aktion auf mehrere Punkte.

Diese Aktion ist eher für die geometrischen Formen als für das freie Zeichnen gedacht. Beim freien Zeichnen würde lediglich eine Verbindung zwischen zwei Punkten gelöscht werden, was unter Umständen kaum sichtbar ist, da eine frei gezeichnete Linie aus vielen Punkten besteht.

Der Undo dieser Aktion stellt die originale Linie wiederher und löscht die Teillinien. Der Redo löscht die originale Linie wieder und stellt die Teillinien wieder her.

3.6.7.3 Erase-Aktion

Diese Lösch-Aktion ermöglicht das Löschen von einem oder mehreren Drawable-Typ-Objekten während eines Durchlaufes der Aktion. Zum Löschen muss der Benutzer ein Objekt auswählen und die linke Maustaste drücken, wenn er diese gedrückt hält können weitere Drawable-Typ-Objekte gelöscht werden. Diese Objekte können sich auf unterschiedlichen Drawables befinden. Deshalb wird für jedes gelöschte Objekt ein Memento über dessen Drawable-Typ- und Drawable-Konfiguration angelegt. Jedes dieser Mementos wird in einer Mementoliste gespeichert, die diese Aktion bereitstellt. Zum Beenden der Aktion muss der Benutzer die linke Maustaste loslassen.

Eine Besonderheit beim Löschen stellen Mind Map-Knoten dar. Beim Löschen werden die Kinderknoten des Knotens sowie die Verbindungslinien mitgelöscht, da diese ohne den Elternknoten nicht existieren können, laut den gestellten Bedingungen im Abschnitt [Mind Map-Drawable-Typ](#). Dieser Vorgang wird rekursiv für die zu löschenden Knoten wiederholt. Zudem wird der ausgewählte zu löschende Knoten aus der Kinderliste der "MMNodeValueHolder"-Komponente des Elternknotens entfernt, vorausgesetzt der ausgewählte Knoten hat einen Elternknoten. In der nachfolgenden [Abbildung 3.70](#) kann das Löschen des Mind Map-Knotens "Duathlon" betrachtet werden. Wie beschrieben wurde, wurden die Kinderknoten "Radfahren" und "Laufen" ebenfalls gelöscht.

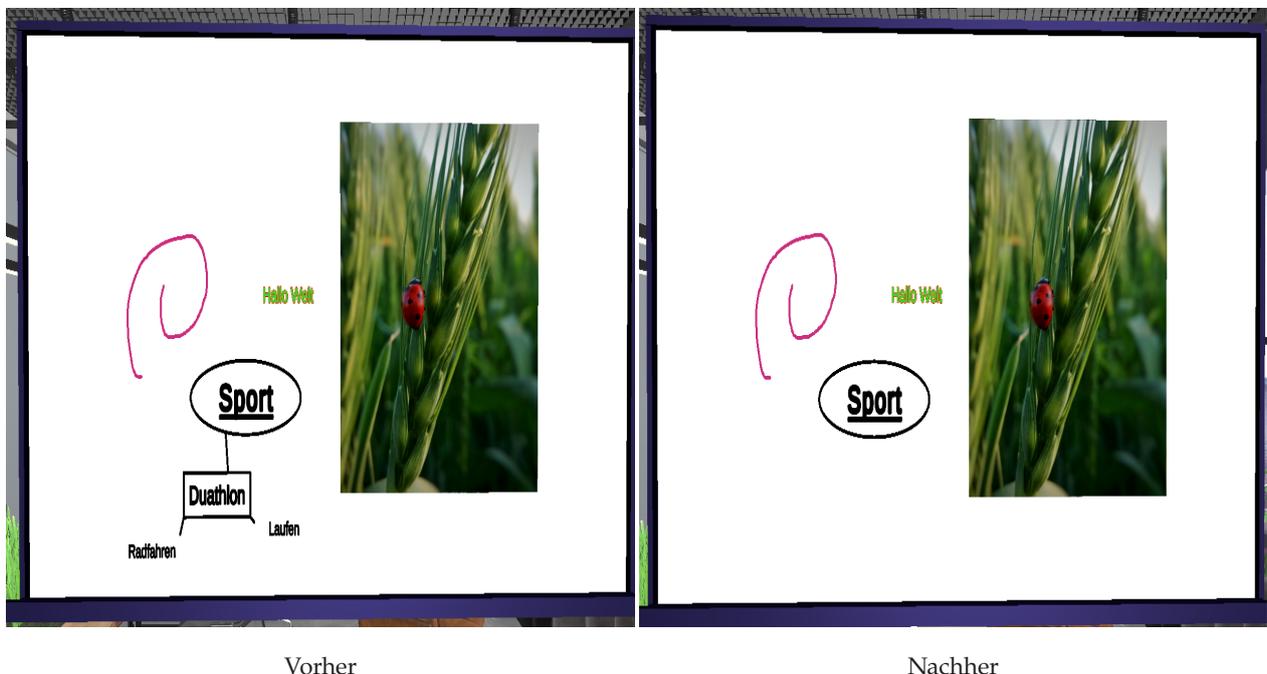


Abbildung 3.70: Anwendung der Erase-Aktion auf den Knoten "Duathlon".

Mit der Undo-Funktionalität werden die gelöschten Drawable-Typ-Objekte auf ihren dazugehörigen Drawables wiederhergestellt. Hierzu wird die "void Restore"-Methode der DrawableType-Klasse verwendet. Vorher müssen die wiederherzustellenden Objekte jedoch sortiert werden. Dies hat den bereits genannten Grund, dass ein Mind Map-Kinderknoten nicht vor dessen Elternknoten wiederhergestellt werden kann. Es könnte nämlich sein, dass der Benutzer zuerst einen Kinderknoten und anschließend den Elternknoten gelöscht hat. Dabei werden zuerst die Drawable-Typen mittels der von der DrawableType-Klasse bereitgestellten Methode "OrderOnType" sortiert. Dafür bekommen Linien eine Wertigkeit von 1, Texte eine von 2, Bilder eine von 3 und Mind Map-Knoten eine von 4. Dadurch wird sichergestellt, dass MindMap-Knoten zuletzt wiederhergestellt werden. Anschließend müssen die Mind Map-Knoten mittels der "OrderMindMap"-Methode sortiert werden, die ebenfalls von der DrawableType-Klasse bereitgestellt wird. Diese Methode sortiert die Knoten nach ihrer in der "MMNodeValueHolder"-Komponente enthaltene Ebene.

Die Redo-Funktionalität löscht die wiederhergestellten Objekte wieder.

3.6.7.4 *Cleaner-Aktion*

Durch diese Aktion kann ein Drawable vollständig von Drawable-Typ-Objekten gesäubert werden. Durch einen linken Mausklick kann die Aktion auf einem Drawable ausgeführt werden. Die Aktion wird ebenfalls ausgeführt, wenn das Raycasting statt einem Drawable ein Drawable-Typ-Objekt erkennt. In diesem Fall wird das dazugehörige Drawable mittels der GameFinder-Methode "getDrawable" erfragt.

Anschließend wird die Konfiguration des Drawables abgerufen und durch die im Abschnitt "[Drawable-Konfiguration](#)" beschriebene Methode "getAllDrawableTypes" alle Konfigurationen der dazugehörigen Drawable-Typ-Objekte erfragt. Danach wird für jede Konfiguration mittels der GameFinder-Methode "findChild" das dazugehörige Objekt ermittelt und gelöscht. In der nachfolgenden [Abbildung 3.71](#) kann das Säubern eines Drawables betrachtet werden.



Vorher

Nachher

Abbildung 3.71: Anwendung der Cleaner-Aktion auf das Whiteboard.

Sollte der Drawable bereits leer sein, wird dem Benutzer eine entsprechende Information angezeigt, die in der folgenden [Abbildung 3.72](#) veranschaulicht wird.



Abbildung 3.72: Information über ein bereits gesäubertes Drawable.

Nachdem das Drawable gesäubert wurde, wird ein Memento mit der zuvor erfragten Drawable-Konfiguration erstellt, das die gelöschten Drawable-Typ-Objekte enthält, und die Aktion abgeschlossen.

Um die gelöschten Objekte mit Undo wiederherzustellen, verwendet diese Aktion die zuvor beschriebene "void Restore"-Methode der DrawableType-Konfigurationsklasse. Mit Redo werden erneut die Objekte gelöscht.

3.6.8 Sticky-Note-Aktion

Diese Aktion stellt als Teilaktionen die Erstellung, das Bewegen sowie das Editieren und Löschen von Sticky-Notes bereit. Mit der Awake-Methode wird das Sticky-Note-Menü der [Abbildung 3.73](#) bereitgestellt.

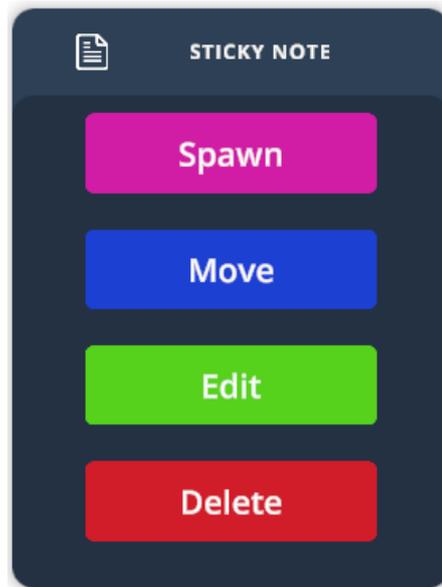


Abbildung 3.73: Darstellung des Sticky-Note-Menüs.

Durch das Menü wird die Auswahl der gewünschten Teilaktion ermöglicht. Die darin enthaltenen Teilaktionen werden in den Unterabschnitten dieser Aktion erläutert. Wenn der Benutzer ohne eine Teilaktion des Menüs zu wählen einen linken Mausklick durchführt, wird er darüber benachrichtigt, dass er keine Aktion ausgewählt hat.

Das Memento dieser Aktion besteht aus der originalen Sticky-Note-Konfiguration, der ausgeführten Teilaktion sowie gegebenenfalls einer abgeänderten Sticky-Note-Konfiguration. Die abgeänderte Konfiguration enthält Änderungen, die durch die Teilaktion **Move** oder **Edit** getätigt wurden. Da diese abgeänderte Konfiguration nicht für jede Teilaktion benötigt wird, wurde sie so ins Memento integriert, dass sie für die benötigten Zwecke zusätzlich konfiguriert werden kann.

Sofern eine Teilaktion nicht abgeschlossen ist und die Aktion gewechselt wird, werden die durchgeführten Änderungen rückgängig gemacht sowie die geöffneten Menüs und ein eventuell hinzugefügter Highlight-Effekt zerstört.

Zudem können die Teilaktionen **Spawn**, **Move** und **Edit** mittels der Taste "ESC" abgebrochen werden. Durch das Abbrechen werden ebenfalls getätigte Aktionen rückgängig gemacht und die Menüs sowie ein eventuell vorhandener Highlight-Effekt zerstört.

3.6.8.1 *Spawn*

Mittels dieser Teilaktion können neue Drawables dem virtuellen Raum der Visualisierung in Form von Sticky-Notes hinzugefügt werden. Zum Starten dieser Funktionalität muss der Benutzer im Sticky-Note-Menü

die Option "Spawn" auswählen.

Ein Sticky-Note kann durch einen linken Mausklick und die Erkennung eines Objektes durch das Raycasting erstellt werden. Hierfür werden zwei verschiedene Varianten bereitgestellt.

1. Wenn das erkannte Objekt ein Drawable, ein Drawable-Typ-Objekt, eine Komponente eines Drawable-Objektes oder ein geeignetes ("suitable") Objekt ist, wird die Rotation von dem erkannten Objekt übernommen. Mit einer Komponente des Drawable-Objektes sind z. B. die Rückseite eines Sticky-Notes oder die Seitenränder gemeint. Für die geeigneten Objekte stellt der Value-Holder eine Liste mit Objektnamen und Tags bereit, die diese Objekte repräsentieren. Mittels der von dem Value-Holder bereitgestellten Methode "IsASuitableObjectForStickyNote" kann überprüft werden, ob das gegebene Objekt als geeignet betrachtet werden kann.
2. Ist das Objekt keines der eben genannten, dann werden die Sticky-Note-Move- sowie Sticky-Note-Rotate-Menüs eingeblendet. Die Menüs können in der [Abbildung 3.74](#) und [Abbildung 3.75](#) betrachtet werden. Mit dem Move-Menü kann die Position des Sticky-Notes angepasst werden. Dabei können nicht nur die Positionen der X-Achse und Y-Achse angepasst werden, sondern auch die der Z-Achse. Eine Information für die Verwendung der durch das Menü bereitgestellten Buttons kann über den "i"-Button angezeigt werden. Diese Information kann in der [Abbildung 3.76](#) betrachtet werden. Zusätzlich kann das Objekt über die Pfeiltasten auf der X- und Y-Achse bewegt werden. Zudem kann mit den Tasten "Bild auf" und "Bild ab" das Objekt auf der Z-Achse bewegt werden. Durch das Rotate-Menü kann die Rotation der X- und Y-Achse bestimmt werden. Für die X-Rotation stehen die Optionen "liegend (laying)" und "hängend (hanging)" zur Verfügung. Über den "Next"-Button kann zur Rotation der Y-Achse gewechselt werden. Hierbei steht ein Slider für die Rotation sowie vier Buttons für 0°, 90°, 180° und 270° zur Verfügung. Zusätzlich kann während der Rotation der Y-Achse das Mousrad zum Rotieren verwendet werden. Auch hier ermöglicht die "Strg"-Taste eine schnellere Rotierung. Die verwendeten Geschwindigkeiten für die Rotation über das Mousrad sind hierbei die gleichen, wie bei der [Move or Rotate-Aktion](#). Dasselbe gilt auch für die Geschwindigkeiten der Bewegung. Mittels der "Finish"-Buttons, die von beiden Menüs bereitgestellt werden, kann das Positionieren beendet werden.

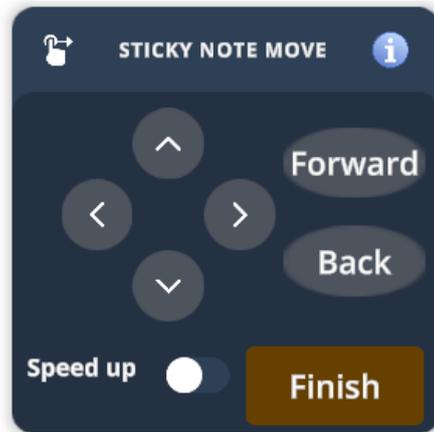
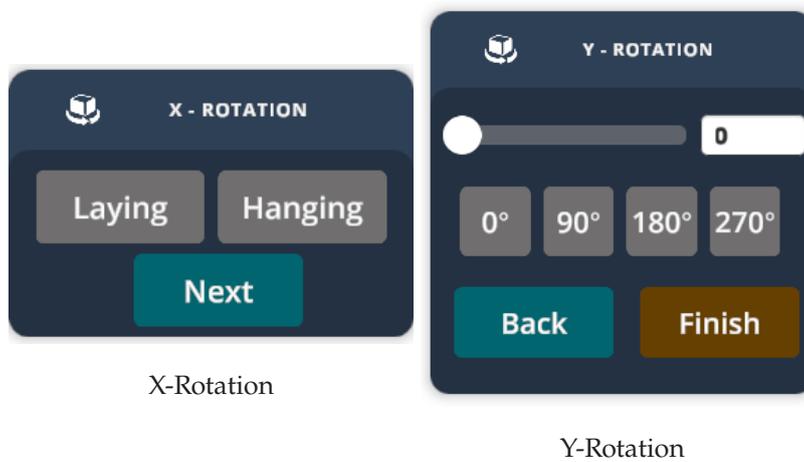


Abbildung 3.74: Darstellung des Sticky-Note-Move-Menüs.



X-Rotation

Y-Rotation

Abbildung 3.75: Demonstration der Rotate-Menüs für Sticky-Notes.

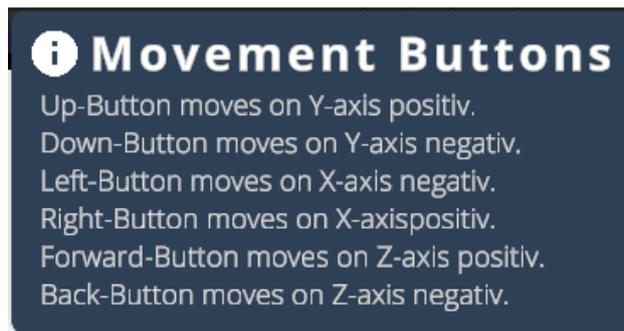


Abbildung 3.76: Informationen, die über den "i"-Button des Sticky-Note-Move-Menüs angezeigt werden können.

Nachdem das Sticky-Note vollständig platziert wurde, wird die Netzwerkaktion "StickyNoteSpawnNetAction" ausgeführt, um bei allen anderen Benutzern diese Sticky-Note zu erstellen. Anschließend wird ein Memento mit der Sticky-Note-Konfiguration sowie der ausführende

Operation "Spawn" erstellt und die Aktion abgeschlossen.

Die Undo-Funktionalität dieser Teilaktion löscht das erstellte Sticky-Note wieder. Durch Redo wird das Sticky-Note anhand der gespeicherten Drawable-Konfiguration wiederhergestellt.

3.6.8.2 *Move*

Für diese Teilaktion muss zunächst im Sticky-Note-Menü die Option "Move" gewählt werden. Anschließend muss ein Sticky-Note zum Bewegen ausgewählt werden. Dazu muss der Benutzer mit einem linken Mausklick das gewünschte Objekt auswählen. Zur Erkennung des Sticky-Notes stehen wieder die Drawable-Fläche, ein Drawable-Typ-Objekt, das sich auf dem Sticky-Note befindet, oder eine Teilkomponente des Sticky-Notes zur Verfügung. Sollte keines dieser Komponenten ausgewählt worden sein, wird ein entsprechender Hinweis angezeigt.

Nach der Auswahl wird damit begonnen, das Sticky-Note per Mauszeigerposition zu bewegen. Hierzu folgt das Sticky-Note dem Raycasting-Laser. Um dies zu ermöglichen, mussten die Collider des Sticky-Notes deaktiviert werden, da sonst der eigene Collider erkannt worden wäre. Wenn der Raycasting-Laser während des Bewegens des Sticky-Notes ein Drawable, ein Drawable-Typ-Objekt, eine Teilkomponente oder ein geeignetes Objekt erfasst, wird deren Rotation übernommen. Für die Drawable-Typ-Objekte oder eine Teilkomponente wird die Rotation des jeweiligen Drawables verwendet. Das Bewegen mit der Maus wird durch einen linken Mausklick beendet. Anschließend wird das Move- sowie Rotate-Menü für Sticky-Notes eingeblendet, um eine präzise Positionierung und Rotierung zu ermöglichen. Zusätzlich wird neben den Menüs die Tastatur- sowie Mausbelegung angeboten, die im Abschnitt "Spawn" beschrieben wurde. Durch einen der "Finish"-Buttons der Menüs kann diese Teilaktion abgeschlossen werden.

Das Memento wird bei dieser Teilaktion während der Auswahl des zu bewegendes Sticky-Notes erstellt und umfasst die Konfigurationen des Sticky-Notes vor und nach der Veränderung sowie die ausführende Operation "Move". Beim Abschließen wird die Position und Rotation in der Konfiguration, die die Änderungen enthält, abgeändert und die Collider wieder angeschaltet.

Die Undo-Funktionalität ermöglicht das Herstellen der vorherigen Position und Rotation des Sticky-Notes. Die Redo-Funktionalität stellt die neu konfigurierte Position und Rotation wieder her.

3.6.8.3 Edit

Diese Teilaktion ermöglicht das Editieren eines Sticky-Notes. Das Editieren umfasst hierbei die Anpassung der Rotation, der Skalierung und der Farbe der Vorderseite des Sticky-Notes.

Zunächst muss der Benutzer im Sticky-Note-Menü die Option "Edit" wählen. Anschließend muss er das Sticky-Note, das er editieren möchte, mit einem linken Mausklick auswählen. Hierfür gelten dieselben Bedingungen, die bereits in der [Move](#)-Teilaktion beschrieben wurden.

Nachdem das Sticky-Note ausgewählt wurde, wird ihm mittels des GameHighlighters ein Highlight-Effekt zugewiesen. Zudem wird das in der [Abbildung 3.77](#) gezeigte Sticky-Note-Edit-Menü eingebunden.

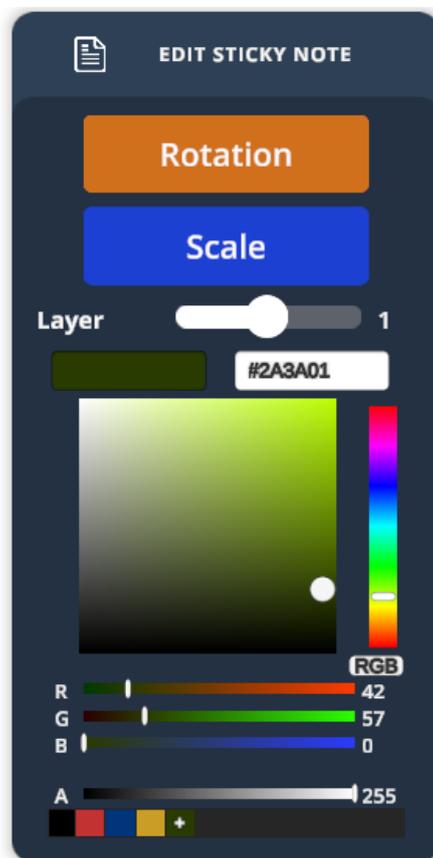


Abbildung 3.77: Darstellung des Sticky-Note-Edit-Menüs.

Durch die Option "Rotate" wird das Sticky-Note-Rotate-Menü geöffnet und das Edit-Menü geschlossen. Zudem wird ein Button zum Zurückkehren zum Edit-Menü eingeblendet. Das Rotieren funktioniert hierbei genauso wie in den Teilaktionen [Spawn](#) und [Move](#).

Mittels der Option "Scale" wird das in der [Abbildung 3.50](#) gezeigte Scale-Menü eingebunden. Hierbei werden jedoch zwei zusätzliche

Buttons eingebunden. Einer zum Zurückkehren zum Edit-Menü und der andere zum Beenden des Skalierens. Das Skalieren funktioniert hierbei genau wie im Abschnitt der [Scale-Aktion](#) beschrieben.

Im Edit-Menü selbst kann über einen Slider die Ebenenreihenfolge und über den Color-Picker eine Farbe für die Vorderseite des Sticky-Notes gewählt werden.

Vom Edit-Menü aus kann die Teilaktion durch einen linken Mausklick auf dem Drawable oder ein Drawable-Typ-Objekt beendet werden. Sollten keine Änderungen vorgenommen worden sein, wird die Teilaktion abgebrochen. Anderenfalls wird ein Memento mit den Änderungen, ähnlich wie bei der [Move](#)-Teilaktion, erstellt. Mit dem Unterschied das hier die ausführende Operation "Edit" ist. Zudem wird die Aktion abgeschlossen und der Highlight-Effekt sowie die Menüs zerstört.

3.6.8.4 *Delete*

Mit dieser Teilaktion kann ein Sticky-Note gelöscht werden, wodurch ebenfalls alle sich darauf befindenden Drawable-Typ-Objekte gelöscht werden. Um diese Aktion zu aktivieren, muss der Benutzer im Sticky-Note-Menü die Option "Delete" wählen. Anschließend kann er mit einem linken Mausklick das gewünschte Sticky-Note löschen. Dazu kann er entweder auf die Drawable-Fläche des Sticky-Notes klicken, auf einem Drawable-Typ-Objekt, das sich auf diesem befindet, oder auf eine Komponente des Sticky-Notes. Mit den Komponenten sind die Rückseite sowie die Seitenränder gemeint. Vor dem Löschen wird überprüft, ob das ausgewählte Objekt zu einem Sticky-Note gehört. Dazu wird geprüft, ob das Elternobjekt der Drawable-Fläche den Sticky-Note-Präfix aufweist. Wenn eine falsche Auswahl getroffen wird, wird eine entsprechende Benachrichtigung ausgegeben.

Nach erfolgreicher Auswahl wird vor dem Löschen ein Memento über die Sticky-Note-Konfiguration sowie der ausführenden Aktion "Delete" erstellt. Die Netzwerkaktion, die zum Löschen ausgeführt wird, ist die "StickyNoteDeleterNetAction". Das Löschen eines Sticky-Notes wird in der nachfolgenden [Abbildung 3.78](#) veranschaulicht. Anschließend wird die Aktion abgeschlossen.

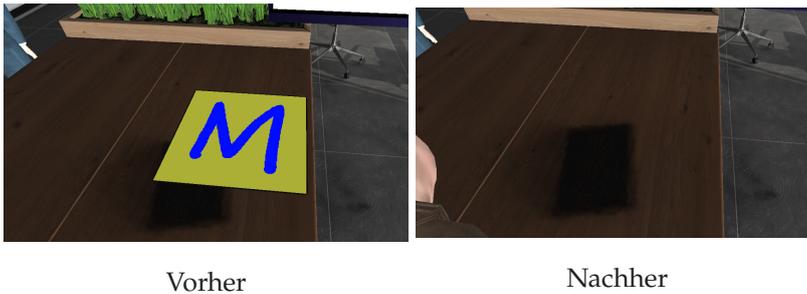


Abbildung 3.78: Demonstration des Löschens eines Sticky-Notes.

Die Funktionalität des Undos stellt das Sticky-Note wieder her und lässt mittels der “void Restore“-Methode der DrawableType-Klasse sämtliche Drawable-Typ-Objekte, die sich auf dem Sticky-Note befanden, wiederherstellen. Redo löscht das Sticky-Note erneut.

3.7 DRAWABLE-SYNCHRONIZER

Der Drawable-Synchronizer ist eine statische Klasse (“DrawableSynchronizer”), die sicherstellt, dass neu beigetretene Benutzer dieselben Drawable-Typen auf den jeweiligen Drawables angezeigt bekommen. Zudem wird die aktuelle Ebenenreihenfolge über das Netzwerk synchronisiert. Um die Synchronisation zu realisieren, wurde die “Spawn“-Methode der “PlayerSpawner“-Klasse, die jedes Mal von dem Host der Visualisierung ausgeführt wird, wenn ein neuer Benutzer beitrifft, so modifiziert, dass der Synchronizer aufgerufen wird, sofern mehr als ein Benutzer in dem virtuellen Raum der Visualisierung ist. Dazu werden zunächst alle Drawables der Visualisierung abgerufen. Anschließend wird für jeden Sticky-Note-Drawable die Netzwerkfunktionalität zum Erstellen ausgeführt. Zudem werden für alle Drawable-Typen ebenfalls dessen jeweilige Netzwerkfunktionalität zum Erstellen durchgeführt, um die Objekte zu synchronisieren.

Diese Vorgehensweise des Synchronizers ist jedoch nicht effizient, aufgrund der Tatsache, dass auch bei Benutzern die Drawable-Typen synchronisiert werden, bei denen es nicht erforderlich wäre, da sie bereits synchronisiert waren.

Daher sollte diese Vorgehensweise, wie später im Abschnitt [“Verbesserung für den Drawable-Synchronizer“](#) beschrieben, modifiziert werden. Diese Ineffizienz wird jedoch erst bemerkbar, wenn besonders viele [Image-Drawable-Typen](#) in dem Meeting vorhanden sind, die synchronisiert werden müssen.

EVALUATION

In diesem Kapitel wird die Evaluierung der Integration behandelt. Dazu werden im ersten Abschnitt die [Ziele der Untersuchung und die daraus resultierenden Forschungsfragen](#) vorgestellt. Im zweiten Abschnitt wird der zugrundeliegende [Versuchsaufbau](#) erläutert. Anschließend erfolgt die Beschreibung der [Auswahl geeigneter Probanden](#) sowie der [gestellten Aufgaben](#) der Studie. Abschließend erfolgt die [Auswertung](#) der Ergebnisse inklusive einer Diskussion darüber und eine Beschreibung der [Threats to Validity](#).

Zunächst wird geklärt, was die Benutzerfreundlichkeit ist und wie sie definiert ist. Die Benutzerfreundlichkeit, auch Gebrauchstauglichkeit oder Usability genannt, wird durch die ISO 9241-11 Norm definiert. Diese Norm stammt aus dem Jahr 1999 und wurde im Jahr 2018 erneuert ([Tesch, 2019](#)).

Die Benutzerfreundlichkeit setzt sich aus den drei Kriterien "Effektivität", "Effizienz" und "Zufriedenstellung" zusammen. Die Effektivität beschreibt dabei die Genauigkeit und Vollständigkeit, mit der ein Benutzer ein bestimmtes Ziel erreichen kann. Mittels der Effizienz wird das Verhältnis von Effektivität zum eingesetzten Aufwand bezeichnet, mit dem der Benutzer das Ziel erreichen kann. Mit dem Kriterium "Zufriedenstellung" wird die Zufriedenheit des Benutzers während der Benutzung des Systems beschrieben ([Dörflinger, n.d.b](#)).

[Nielsen \(2012\)](#) definiert die Benutzerfreundlichkeit durch fünf Kriterien. Diese umfassen die Erlernbarkeit, die Effizienz, die Erinnerung, die Fehler und die Zufriedenheit. Die Erlernbarkeit beschreibt, wie einfach es für den Benutzer ist, das System bei der Erstbenutzung zu erlernen. Die Effizienz beschreibt, wie schnell die Benutzer Aufgaben ausführen können, nachdem sie die Benutzung erlernt haben. Die Erinnerung bezieht sich darauf, wie leicht die Benutzer wieder in die Benutzung des Systems finden, nach längerer Zeit der Nichtbenutzung. Das Kriterium Fehler beschreibt, wie viele Fehler der Benutzer während der Benutzung macht, wie schwerwiegend diese sind und wie leicht diese rückgängig gemacht werden können. Abschließend beschreibt die Zufriedenheit, wie angenehm es für den Benutzer ist, das System zu benutzen.

4.1 ZIELE UND FORSCHUNGSFRAGEN

Wie im Abschnitt *“Anforderungen und Ziele“* beschrieben wurde, liegt ein besonderes Augenmerk dieser Integration auf einer guten Benutzerfreundlichkeit. Dies soll auch für Nicht-Informatiker (Laien) gelten, das bedeutet, dass selbst Personen ohne Informatikkenntnisse problemlos diese Integration verwenden können sollten. Daher reicht die im Abschnitt *“Forschungsfrage und Struktur“* genannte zentrale Forschungsfrage nach der Benutzerfreundlichkeit der Integration nicht aus, da diese sehr allgemein formuliert ist. Deshalb wird sie zur Beantwortung des wissenschaftlichen Aspektes dieser Arbeit in die folgenden drei Forschungsfragen unterteilt.

1. Welche Benutzerfreundlichkeit bietet die Integration insgesamt?
2. Bewerten Informatiker die Benutzerfreundlichkeit besser und führen sie gestellte Aufgaben schneller und mit geringerer Anstrengung durch als Nicht-Informatiker?
3. Wie verändert sich die Wahrnehmung der Benutzerfreundlichkeit, die benötigte Zeit und wahrgenommene Anstrengung bei erneuter Verwendung der Integration?

Mittels der ersten Forschungsfragen werden Aussagen über die allgemeine Benutzerfreundlichkeit getroffen. Die zweite Forschungsfrage dient dabei der Untersuchung, ob die Integration so benutzerfreundlich ist, dass sie auch für Personen ohne Informatikkenntnisse geeignet wäre. Durch die dritte Forschungsfrage werden Rückschlüsse auf das erinnerte Verständnis sowie die Wiedererkennung der Integration gezogen.

SUS: Ist ein standardisierter Fragebogen von John Brooke aus dem Jahr 1986, der am Ende der Studie zur Auswertung der Gesamtbenutzerfreundlichkeit des Systems verwendet wird und zehn verschiedene Fragen enthält.

ASQ: Ist ein standardisierter Fragebogen, der nach jeder erledigten Aufgabe ausgefüllt wird. Dabei bewertet er die Leichtigkeit der Aufgabenerledigung sowie die Zufriedenheit mit der benötigten Zeit für die Aufgabe und die Zufriedenheit mit den bereitgestellten Informationen (Dörflinger, n.d.a).

Diese drei Forschungsfragen wurden anhand der folgenden Hypothesen überprüft. Dafür ist es wichtig zu wissen, dass die gemessenen Daten für die Benutzerfreundlichkeit aus den Ergebnissen des *System Usability Scale (SUS)* und des *After Scenario Questionnaire (ASQ)* stammen. Der SUS liefert einen sogenannten SUS-Score, mit dem die Benutzerfreundlichkeit abgeleitet werden kann, während der ASQ aus der Zufriedenheit mit der Komplexität, dem Aufwand und den bereitgestellten Informationen der Integration besteht. Die Komplexität beschreibt dabei die Leichtigkeit, mit der eine Aufgabe bewältigt werden konnte. Beide Fragebögen werden später näher erläutert. Sauro (2018) stellt eine Tabelle bereit, aus der entnommen werden kann, dass eine gute Benutzerfreundlichkeit ab einem SUS-Score von mindestens 71,1 Punkten beginnt. Diese Information ist wichtig für die nachfolgende Hypothesenbeschreibung.

1. Hypothesen der ersten Forschungsfrage:

- *Nullhypothese \mathcal{H}_{a_0}* : Die Integration weist keine gute Benutzerfreundlichkeit auf: SUS-Score < 71,1

- *Alternative Hypothese* \mathcal{H}_{a_1} : Die Integration weist mindestens eine gute Benutzerfreundlichkeit auf: $\text{SUS-Score} \geq 71,1$

2. Hypothesen der zweiten Forschungsfrage:

Bei dieser Forschungsfrage wird zwischen Informatikern (α) und Nicht-Informatikern (β) unterschieden. Die Bezeichnung α und β werden später ebenfalls in den gezeigten Diagrammen verwendet.

Zunächst werden die Hypothesen des Unterschieds zwischen den Bewertungen für die Gesamtbenutzerfreundlichkeit aufgestellt.

- *Nullhypothese* \mathcal{H}_{b_0} : Die Nicht-Informatiker bewerten die Benutzerfreundlichkeit gleich gut oder besser als die Informatiker: $\text{SUS-Score}_\beta \geq \text{SUS-Score}_\alpha$
- *Alternative Hypothese* \mathcal{H}_{b_1} : Die Nicht-Informatiker bewerten die Benutzerfreundlichkeit schlechter als die Informatiker: $\text{SUS-Score}_\beta < \text{SUS-Score}_\alpha$

Als nächstes werden die Hypothesen des Unterschieds zwischen der benötigten Zeit formuliert.

- *Nullhypothese* \mathcal{H}_{c_0} : Die Nicht-Informatiker benötigen genauso viel oder weniger Zeit als die Informatiker: $\text{Time}_\beta \leq \text{Time}_\alpha$
- *Alternative Hypothese* \mathcal{H}_{c_1} : Die Nicht-Informatiker benötigen mehr Zeit als die Informatiker: $\text{Time}_\beta > \text{Time}_\alpha$

Abschließend werden die Hypothesen der ASQ-Daten präsentiert. Eine niedrigere Punktzahl stellt dabei eine höhere Zufriedenheit dar.

- **Komplexität:**

- *Nullhypothese* \mathcal{H}_{d_0} : Die Nicht-Informatiker empfinden die Komplexität als gleich oder leichter als die Informatiker: $\text{Ease}_\beta \leq \text{Ease}_\alpha$
- *Alternative Hypothese* \mathcal{H}_{d_1} : Die Nicht-Informatiker empfinden die Komplexität als schwerer als die Informatiker: $\text{Ease}_\beta > \text{Ease}_\alpha$

- **Aufwand:**

- *Nullhypothese* \mathcal{H}_{e_0} : Die Nicht-Informatiker empfinden den aufgetragenen Aufwand als gleich oder leichter als die Informatiker: $\text{Effort}_\beta \leq \text{Effort}_\alpha$
- *Alternative Hypothese* \mathcal{H}_{e_1} : Die Nicht-Informatiker empfinden den aufgetragenen Aufwand als höher als die Informatiker: $\text{Effort}_\beta > \text{Effort}_\alpha$

- **Informationszufriedenheit:**

Bedenke: Eine niedrigere Punktzahl bedeutet eine höhere Zufriedenheit.

- *Nullhypothese* \mathcal{H}_{f_0} : Die Nicht-Informatiker empfinden die Zufriedenheit mit den Informationen als gleich oder besser als die Informatiker: $\text{Info}_\beta \leq \text{Info}_\alpha$
- *Alternative Hypothese* \mathcal{H}_{f_1} : Die Nicht-Informatiker empfinden die Zufriedenheit mit den Informationen als schlechter als die Informatiker: $\text{Info}_\beta > \text{Info}_\alpha$

3. Hypothesen der dritten Forschungsfrage:

Hierbei wird zwischen dem ersten Durchlauf (\mathcal{F}) und dem zweiten Durchlauf (\mathcal{S}) unterschieden. Die Abkürzung \mathcal{F} steht dabei für “First” und \mathcal{S} für “Second”. Diese Bezeichnungen werden später in den gezeigten Diagramme verwendet.

Zunächst werden die Hypothesen des Unterschieds zwischen den Bewertungen für die Gesamtbenutzerfreundlichkeit aufgestellt.

- *Nullhypothese* \mathcal{H}_{g_0} : Das Ergebnis des SUS-Scores vom ersten Durchlauf ist gleich oder besser als das vom zweiten Durchlauf: $\text{SUS-Score}_{\mathcal{F}} \geq \text{SUS-Score}_{\mathcal{S}}$
- *Alternative Hypothese* \mathcal{H}_{g_1} : Das Ergebnis des SUS-Scores vom ersten Durchlauf ist schlechter als das vom zweiten Durchlauf: $\text{SUS-Score}_{\mathcal{F}} < \text{SUS-Score}_{\mathcal{S}}$

Als nächstes werden die Hypothesen des Unterschieds zwischen der benötigten Zeit formuliert.

- *Nullhypothese* \mathcal{H}_{h_0} : Die benötigte Zeit im ersten Durchlauf ist gleich oder geringer als die im zweiten Durchlauf: $\text{Time}_{\mathcal{F}} \leq \text{Time}_{\mathcal{S}}$
- *Alternative Hypothese* \mathcal{H}_{h_1} : Die benötigte Zeit vom ersten Durchlauf ist höher als die vom zweiten Durchlauf: $\text{Time}_{\mathcal{F}} > \text{Time}_{\mathcal{S}}$

Abschließend werden die Hypothesen der ASQ-Daten vorgestellt. Auch hier gilt, dass eine niedrigere Punktzahl eine höhere Zufriedenheit darstellt.

- **Komplexität:**
 - *Nullhypothese* \mathcal{H}_{i_0} : Die empfundene Komplexität im ersten Durchlauf ist gleich oder niedriger als im zweiten Durchlauf: $\text{Ease}_{\mathcal{F}} \leq \text{Ease}_{\mathcal{S}}$
 - *Alternative Hypothese* \mathcal{H}_{i_1} : Die empfundene Komplexität vom ersten Durchlauf ist höher als im zweiten Durchlauf: $\text{Ease}_{\mathcal{F}} > \text{Ease}_{\mathcal{S}}$
- **Aufwand:**
 - *Nullhypothese* \mathcal{H}_{j_0} : Der empfundene aufgebrauchte Aufwand ist im ersten Durchlauf gleich oder niedriger als im zweiten Durchlauf: $\text{Effort}_{\mathcal{F}} \leq \text{Effort}_{\mathcal{S}}$

- *Alternative Hypothese* \mathcal{H}_{j_1} : Der empfundene aufgebrauchte Aufwand ist im ersten Durchlauf höher als im zweiten Durchlauf: $\text{Effort}_{\mathcal{F}} > \text{Effort}_{\mathcal{S}}$
- **Informationszufriedenheit:**
Bedenke: Eine niedrigere Punktzahl bedeutet eine höhere Zufriedenheit.
 - *Nullhypothese* \mathcal{H}_{k_0} : Die Zufriedenheit mit den bereitgestellten Informationen ist im ersten Durchlauf gleich oder besser als im zweiten Durchlauf: $\text{Info}_{\mathcal{F}} \leq \text{Info}_{\mathcal{S}}$
 - *Alternative Hypothese* \mathcal{H}_{k_1} : Die Zufriedenheit mit den bereitgestellten Informationen ist im ersten Durchlauf schlechter als im zweiten Durchlauf: $\text{Info}_{\mathcal{F}} > \text{Info}_{\mathcal{S}}$

4.2 VERSUCHSAUFBAU

Zur Untersuchung der zuvor beschriebenen Forschungsfragen und zur Aufdeckung vorhandener Fehler der Integration wurde eine empirische Studie in Form einer Usability-Studie durchgeführt. Für die Aufdeckung vorhandener Fehler war es wichtig, dass durch die Aufgaben, die während der Studie bewältigt werden mussten, jede Aktion der Integration überprüft wurde. Aufgrund der Tatsache, dass diese Integration eine Vielzahl von Aktionen bietet, wurde entsprechend viel Zeit zur Bearbeitung der Aufgaben benötigt. Aus diesem Grund wurden die verwendeten Fragebögen so gewählt, dass sie nicht zu viel Zeit zum Ausfüllen benötigen. Dadurch sollte vermieden werden, dass die Probanden zu lange mit der Studie beschäftigt sind, um Ermüdungseffekten vorzubeugen. Daher wurde darauf geachtet, dass die Fragebögen so kurz wie möglich sind und dabei die bestmögliche Aussagekraft bieten.

Zunächst erfolgte in der Studie eine Einleitung, in der die Probanden darüber aufgeklärt wurden, weshalb diese Studie durchgeführt wird und wie ihr Ablauf ist. Zudem wurde beschrieben, worauf die Probanden achten müssen und wie viel Zeit sie ungefähr für die Studie einplanen sollten. Die angegebene Zeitspanne umfasste eine bis anderthalb Stunden. Besonders wichtig bei der Aufklärung war es, deutlich zu machen, dass die Probanden während der Aufgabenausführung auf Pausen verzichten sollen, da dies die benötigte Zeit negativ beeinflussen würde.

Darauf folgte eine typische Datenschutzerklärung, die von den Probanden bestätigt werden musste. In dieser Erklärung wurde beschrieben, wer für die Datenerhebung verantwortlich ist und wie diese Person kontaktiert werden kann. Darüber hinaus wurde erklärt, welche personenbezogenen Daten erhoben werden und zu welchem Zweck sie

erhoben werden. Es wurde betont, dass die Teilnahme freiwillig ist und die Daten anonymisiert und vertraulich behandelt werden. Außerdem wurde die Datenspeicherung sowie ihre Dauer erläutert, wobei die Weitergabe der Daten an Dritten ohne Zustimmung ausgeschlossen wurde. Abschließend wurden die Probanden über ihre Rechte aufgeklärt.

Anschließend wurde ein Fragebogen verwendet, um die demografischen Fragen der Probanden zu erheben. Diese Fragen dienen zur Ermittlung der unabhängigen Variablen dieser Studie. Sie umfassen das Alter, das Geschlecht, den bisher höchsten Bildungsabschluss, ob der Proband aus dem Informatikbereich stammt und dessen Erfahrung mit SEE, Bildbearbeitungsprogrammen und Videospiele. Bei der Geschlechtsangabe standen die Optionen "Männlich", "Weiblich" und "Divers" zur Verfügung. Für den Bildungsabschluss wurden die gängigen deutschen Bildungsabschlüsse verwendet, wobei "ohne Abschluss" auch den Fall "noch in der Schule" repräsentiert. Für die Erfahrungen mit SEE wurden die Auswahlmöglichkeiten "Keine", "Bereits von gehört", "Bereits genutzt" und "Daran entwickelt" bereitgestellt. Wohingegen für Bildbearbeitungsprogrammen und Videospiele die Optionen "Keine", "Kaum genutzt", "Etwas genutzt" und "Viel genutzt" zur Auswahl angeboten wurden.

Nach dem demografischen Fragebogen folgt die Erläuterung für die Einführung in die Integration. Zunächst wurden hierfür Begrifflichkeiten sowie die Steuerung geklärt. Anschließend folgte eine kurze Einführung, bei der beschrieben wurde, wie die Integration ausgeführt werden kann. Zudem sollte hier vermittelt werden, wie die Aktionen anzuwenden sind. Dazu sollte der Proband eine einfache Linie mittels der [Draw Freehand-Aktion](#) zeichnen und anschließend den Drawable durch die [Erase-Aktion](#) oder die [Cleaner-Aktion](#) säubern. Dabei war es wichtig, dass die Probanden lediglich die genannten Aktionen verwenden, um Lerneffekte zu vermeiden. Am Ende der Einführung wurde eine weitere unabhängige Variable erfragt, die die Art der Studiendurchführung betrifft. Dabei ging es darum, ob die Studie allein oder mit dem Studieninitiator durchgeführt wurde. Hierfür standen die Optionen "Ja, befindet sich physisch im selben Raum", "Ja, per Zoom" und "Nein" zur Verfügung. Der Option "Nein" wurde zur Verfügung gestellt, da es theoretisch möglich gewesen wäre, dass ein Proband die Studie alleine durchführen hätte können. Dies war jedoch nicht der Fall.

Anschließend folgten die zu erledigenden Aufgaben. Die konkreten Aufgabenstellungen werden im Abschnitt "[Gestellte Aufgaben](#)" beschrieben. Vor jeder Aufgabe wurde ein Hinweis angezeigt, dass im Folgenden die jeweilige Aufgabe startet und die zu verwendenden Aktionen für die Aufgabe erwähnt. Dieser Hinweis musste über ein

“Ok“-Button bestätigt werden. Ab der Bestätigung wurde die Zeit für die Aufgabe erhoben. Anschließend wurde die Aufgabe in verschiedenen Schritten dargestellt. Zu jedem Schritt wurde genau beschrieben, was der Proband zu erledigen hat und welche Aktion er dafür verwenden soll. Zudem wurden gelegentlich auch spezielle Steuerungshinweise gegeben. Zu jeder Aufgabe wurde auch ein Lösungsbild präsentiert, mit dem der Proband vergleichen konnte, ob seine Lösung passend ist oder um nachzuschauen, falls er etwas nicht verstanden hat. Am Ende der Aufgabe musste die Zeit mit einem erneuten “Ok“-Button gestoppt werden. Nach jeder erledigten Aufgabe folgt ein ASQ-Fragebogen zur Auswertung der Benutzerfreundlichkeit der jeweiligen Aufgabe sowie die Möglichkeit der Eingabe einer Anmerkung. Im späteren Verlauf des Unterabschnittes “Post-Task“ wird beschrieben, welche weiteren Fragebögen hierfür zur Verfügung gestanden hätten und warum der ASQ-Fragebogen verwendet wird.

Am Ende der Studie erfolgt ein weiterer Fragebogen in Form eines SUS zur Bestimmung der Gesamtbenutzerfreundlichkeit der Integration. Im Abschnitt “Post-Study“ wird beschrieben, welche weiteren Fragebögen hierfür verwendet werden könnten und warum der SUS gewählt wurde. Abschließend hatte der Proband die Möglichkeit Verbesserungsvorschläge sowie Wunsch-Features und weitere Anmerkungen zu äußern.

Als nächstes werden die abhängigen Variablen, die durch die Studie ermittelt wurden, beschrieben.

- **Gesamtbenutzerfreundlichkeit der Integration (SUS-Score):**
Ist die errechnete Punktzahl aus dem SUS-Fragebogen. Anhand dieser Punktzahl kann, wie im Abschnitt “Ziele und Forschungsfragen“ beschrieben, die dazugehörige Benutzerfreundlichkeit anhand der von Sauro (2018) bereitgestellten Tabelle abgelesen werden. Wie sich dieser SUS-Score berechnet, wird später in der SUS-Beschreibung im Abschnitt “Post-Study“ beschrieben.
- **Benötigte Zeit zur Aufgabenbewältigung (Time):**
Die benötigte Zeit wird gemessen, indem die Zeit für das Abschließen jeder einzelnen gestellten Aufgabe, wie zuvor beschrieben, erhoben wird. Diese Zeiten werden anschließend addiert, um die Gesamtzeit für alle Aufgaben zu ermitteln. Die Zeit zum Ausfüllen der Fragebögen wird hierbei nicht mit einbezogen. Dies hat den Grund, dass nicht jeder Proband gleich schnell liest und vor allem, weil beim Ausfüllen kein Zeitdruck entstehen sollte. Dem Proband sollte hierbei so viel Zeit wie nötig geboten werden, damit er seine Antworten sorgfältig durchdenken kann, um Fehleinschätzungen oder versehentliche Falschauswahlen zu vermeiden.

Likert-Skala: Ist eine stufenweise Antwortskala, um Meinungen zu einem bestimmten Thema auszudrücken. Sie ist nach dem amerikanischen Forscher Rensis Likert benannt. In der Praxis haben sich fünf- oder siebenstufige Skalen durchgesetzt (Qualtrics.XM, 2024).

- **Komplexität (Ease):**
Mit dieser abhängigen Variable wird die empfundene Komplexität der einzelnen Aufgaben gemessen. Hierfür wird die vom ASQ bereitgestellten “Zufriedenheit mit der Leichtigkeit” verwendet. Der ASQ stellt zur Beantwortung seiner Zufriedenheiten jeweils eine siebenstufige *Likert-Skala* bereit, die ein Spektrum von 1 für starke Zustimmung bis 7 für starke Ablehnung bietet. Folglich gilt: Je niedriger der Wert, desto höher ist die Zustimmung. Um die Komplexität aller Aufgaben zu berechnen, werden die Komplexitäten jeder einzelnen Aufgabe addiert. Aufgrund der Tatsache, dass dieser Studie sieben gestellte Aufgaben zugrunde liegen, ergibt sich eine Komplexität für die Integration im Bereich von 7 bis 49 Punkten.
- **Aufwand (Effort):**
Durch diese abhängige Variable wird der aufgewendete Aufwand zur Aufgabenbewältigung gemessen. Hierfür wird die “Zufriedenheit der benötigten Zeit” des ASQs verwendet. Auch hier kommt die eben beschriebene Likert-Skala zum Einsatz. Außerdem wird dieselbe Berechnung wie bei der Komplexität, jedoch basierend auf dem Aufwand jeder einzelnen Aufgabe, durchgeführt.
- **Informationszufriedenheit (Information satisfaction):**
Mittels dieser Variable wird die Zufriedenheit mit den unterstützenden Informationen gemessen. Unterstützende Informationen beschreiben hierbei die Anzeigen oder Meldungen, die von der Integration angezeigt werden. Mit Anzeigen sind auch visuelle Veränderungen gemeint. Hierfür wird ebenfalls die Likert-Skala und die gleiche Berechnung, basierend auf der Informationszufriedenheit, verwendet.
- **Zeitersparnis:**
Diese abhängige Variable wird für die dritte Forschungsfrage erhoben. Sie wird durch die Zeitersparnis zwischen dem ersten und zweiten Durchlauf eines Probanden berechnet.

Für die dritte Forschungsfrage war ein erneuter Studiendurchlauf notwendig. Die Probanden führten dazu exakt dieselbe Studie durch, wobei der Zeitpunkt für die erneute Durchführung maximal eine Woche später lag. Die gestellten Aufgaben und das verwendete Programm blieben dabei unverändert. Eine Variation der Reihenfolge der gestellten Aufgabe wäre nur bedingt möglich gewesen, da diese, wie im Abschnitt “*Gestellte Aufgaben*” gezeigt wird, teilweise aufeinander aufbauen. Es wurden auch keine neuen Aufgaben verwendet, um deutliche Vergleiche über die benötigte Zeit ziehen zu können.

Im nachfolgenden Abschnitt wird die Wahl der verwendeten Fragebögen beschrieben. Anschließend folgt der Abschnitt über die Erkenntnisse, die durch die durchgeführte Pilotstudie vorgebracht wurden,

und abschließend werden die verwendeten Tools der durchgeführten Studie erklärt.

4.2.1 Fragebögen

Panagiotidi (2021b) beschreibt zwei Arten von Fragebögen zur Auswertung der Benutzerfreundlichkeit. Zum einen sind dies die **Post-Study**-Fragebögen, auch als Post-Test-Fragebögen bekannt, die am Ende der Studie verwendet werden, um den Gesamteindruck der Probanden von dem getesteten Objekt zu ermitteln. Dadurch wird die Gesamtbenutzerzufriedenheit gemessen. Zum anderen sind dies die **Post-Task**-Fragebögen, die nach Abschluss jeder einzelnen Aufgabe zum Einsatz kommen, um die Eindrücke zu dieser Aufgabe zu ermitteln.

Laubheimer (2018) erläutert, dass mit **Post-Task**-Fragebögen auch problematische Teile eines Systems erkannt werden können, was mit einem **Post-Study**-Fragebogen nicht möglich ist, da dieser den Gesamteindruck des Systems abfragt. Daher empfiehlt er, dass eine Kombination aus beiden Arten von Fragebögen für eine Studie verwendet werden sollte. Es sollte jedoch darauf geachtet werden, dass die Probanden nicht überanstrengt werden, um Ermüdungen zu vermeiden.

Bei der Auswahl der Fragebögen war neben der Länge auch entscheidend, dass eine deutsche Version des Fragebogens existieren muss, da nicht alle ausgewählten Probanden gut Englisch verstehen. Zudem sollte die Studie eine einheitliche Sprache aufweisen.

4.2.1.1 *Post-Study*

In diesen Abschnitt werden die Post-Study-Fragebögen zur Ermittlung der Gesamtbenutzerfreundlichkeit verglichen und die Wahl des SUS erläutert.

- *Questionnaire for User Interaction Satisfaction (QUIS)* wurde von Chin, Diehl und Norman im Jahr 1988 entwickelt. Für diesen Fragebogen existieren zwei verschiedene Varianten, die sich in der Länge unterscheiden. Die eine umfasst 41 Fragen und die andere 122. Dabei werden die Antwortmöglichkeiten der einzelnen Fragen mittels einer neunstufigen bipolaren Skala dargestellt. Außerdem gibt es eine deutsche Übersetzung (Panagiotidi, 2021b).
- *Software Usability Measurement Inventory (SUMI)* ist ein von der Human Factors Research Group im Jahr 1993 entwickelter Fragebogen mit 50 Fragen. Er bietet für die Antwortmöglichkeiten eine dreistufige Skala, mit der Zustimmung, Unentschlossenheit und Ablehnung ausgedrückt werden können. Dieser Fragebogen ist auf Deutsch verfügbar (Kirakowski und Corbett, 1993; Panagiotidi, 2021b).

Zustimmungsverzerrung: Beschreibt den Fall, dass der Proband den Fragen zustimmt, ohne dass dies seine persönliche Meinung widerspiegelt. Im Englischen ist dies als acquiescence bias/agreement bias bekannt (Toor, 2020).

- *Post-Study System Usability Questionnaire (PSSUQ)* ist ein Fragebogen, der von IBM entwickelt wurde und in seiner ursprünglichen Version 18 Fragen enthielt. Die aktuelle Version enthält 16 Fragen, die mit einer siebenstufigen Likert-Skala beantwortet werden können. Diese 16 Fragen können in vier Skalen unterteilt werden. Die erste stellt die Gesamtbenutzerzufriedenheit dar, die zweite eine sogenannte System-Qualitätsskala, die dritte die Informations-Qualitätsskala und die vierte die Interface-Qualitätsskala. Dieser Fragebogen ist jedoch anfällig für die sogenannte *Zustimmungsverzerrung* aufgrund der durchwegs positiv formulierten Fragen Panagiotidi (2021b).
- *System Usability Scale (SUS)* wurde von John Brooke im Jahr 1986 entwickelt und besteht aus zehn Fragen, die abwechselnd positiv und negativ formuliert sind. Dadurch wird die Zustimmungsverzerrung vermieden. Deshalb sind die ungeraden Fragen (1,3,5,7 und 9) positiv formuliert und die geraden (2, 4, 6, 8, 10) negativ. Brooke selbst bezeichnet seine Methode als "Quick and dirty" (Rauer, 2011). Sauro (2011) stellte jedoch fest, dass sie alles andere als "dirty" ist. Er prognostiziert, dass SUS auch noch in mindestens 25 Jahre verwendet wird. Außerdem beschreibt er, dass der SUS neben der Gesamtzufriedenheit mit dem System auch die Benutzerfreundlichkeit (Usability) und die Lernfähigkeit (Learnability) feststellen kann. Die Lernfähigkeit wird dabei über die Fragen 4 und 10 bewertet und die Benutzerfreundlichkeit über die restlichen acht Fragen. Die Beantwortung der Fragen des SUS erfolgt über eine fünfstufige Likert-Skala. Guerci (2020) zeigte, wie der SUS-Score berechnet werden kann. Aufgrund der abwechselnd positiven und negativen Formulierung der gestellten Fragen stellen die Extreme der Likert-Skala abwechselnd das best- und schlechtmögliche Ergebnis für die Auswahl dar. Deshalb müssen die Antworten des SUS-Fragebogens auf eine einheitliche Skala angeglichen werden. Dazu wird bei den ungeraden Fragen von dem Index der gewählten Auswahl 1 abgezogen. Das bedeutet, dass wenn beispielsweise für die erste Frage die Antwort "Stimme voll zu" und somit der Index 5 gewählt werden, wird $5 - 1$ berechnet, wodurch die Punktzahl 4 entsteht. Die Antworten von den geraden Fragen werden von 5 abgezogen. Als Beispiel "Stimme überhaupt nicht zu" mit dem Index 1: $5 - 1 = 4$. Beide Antworten stellen hierbei die bestmöglichen Antworten dar. Die Summe der zehn Fragen ergibt einen Wertebereich von 0 bis 40 Punkten. Um dies besser interpretieren zu können, wird das Ergebnis der Addition mit 2,5 multipliziert, um einen Wertebereich von 0 bis 100 zu erhalten. Rummel (2016) stellt eine deutsche Version des SUS zur Verfügung.

- *Standardized User Experience Percentile Rank Questionnaire (SUPR-Q)* ist ein von Jeff Sauro entwickelter Fragebogen mit lediglich acht Fragen. Allerdings ist dieser Fragebogen auf Webseiten ausgelegt und kann daher ausgeschlossen werden. Zudem ist für die Verwendung eine kostenpflichtige Lizenz erforderlich (Panagiotidi, 2021b; Jacobsen, 2019).

Aufgrund der Länge standen lediglich der SUS- und der PSSUQ-Fragebogen zur Auswahl. Allerdings wurde der PSSUQ aufgrund seiner Anfälligkeit für die Zustimmungsverzerrung nicht gewählt, daher fiel die Wahl auf den SUS-Fragebogen.

4.2.1.2 *Post-Task*

Mit diesem Abschnitt werden verschiedene Post-Task-Fragebögen vorgestellt und die Wahl des ASQs erläutert.

- *After Scenario Questionnaire (ASQ)* ist ein Fragebogen, der aus drei Fragen besteht, um die Zufriedenheit mit der Leichtigkeit der Aufgabenbewältigung, die als Komplexität oder kognitiver Anspruch wahrgenommen wird, die Zufriedenheit mit dem Aufwand, der für die Aufgabenbewältigung erforderlich war, sowie die Zufriedenheit mit den unterstützenden Informationen zu messen. Mit unterstützenden Informationen sind hier Benachrichtigung des Systems und visuelle Veränderungen gemeint. Die Antwortmöglichkeiten werden auf einer siebenstufigen Likert-Skala von 1 für "Stimme voll zu" bis 7 für "Stimme überhaupt nicht zu" bewertet (Lewis, 1991).
- *Subjective Mental Effort Questionnaire (SMEQ)* wurde im Jahr 1985 von Zijlstra und Doorn entwickelt und umfasst eine Frage zur Erfassung der mentalen Anstrengung. Die Skala zur Beantwortung reicht von 0 bis 150, wobei 0 für "absolut nicht aufwendig" und ab 110 für "extrem aufwendig" steht (Panagiotidi, 2021a).
- *Expectation ratings (ER)* ist ein Fragebogen, der aus einer Frage vor und einer nach der Aufgabe besteht. Die Frage vor der Aufgabe erfasst die zu erwartende Schwierigkeit zur Aufgabenbewältigung, während die Frage nach der Aufgabe die empfundene Schwierigkeit erfasst. Die Beantwortung erfolgt über eine fünf- oder siebenstufige Likert-Skala. Die durchschnittliche Bewertung der zu erwartenden und empfundenen Werte können in einem Streudiagramm dargestellt werden. Das Streudiagramm kann in vier Quadranten unterteilt werden. In der folgenden Erklärung der vier Quadranten wird mit "vorher" die zu erwartende und mit "nachher" die empfundene Schwierigkeit beschrieben.

1. Vorher einfach - nachher einfach: "Don't touch it" - Die Features dieses Quadranten benötigen keine zusätzliche Aufmerksamkeit.
2. Vorher einfach - nachher schwer: "Fix it fast" - In diesem Quadranten werden die Features dargestellt, die schnellstmöglich überarbeitet werden sollten, da sie die höchste Unzufriedenheit der Probanden darstellen.
3. Vorher schwer - nachher einfach: "Promote it" - In diesem Quadranten werden die Features dargestellt, mit denen sich von der Konkurrenz abheben kann, da die eigene Integration eine gute und leicht anwendbare Lösung für die gestellte Aufgabe bietet.
4. Vorher schwer - nachher schwer: "Big opportunity" - In diesem Quadranten könnten wichtige Chancen für Verbesserungen liegen.

Diese Quadranten können in der nachfolgenden [Abbildung 4.1](#) von [Albert \(2003\)](#) begutachtet werden. Durch dieses Streudiagramm ist gut erkennbar, was verbessert werden sollte und durch welche Lösung die eigene Integration hervorgehoben werden kann im Vergleich zu ähnlichen Integrationen ([Panagiotidi, 2021a](#); [Albert, 2003](#)).

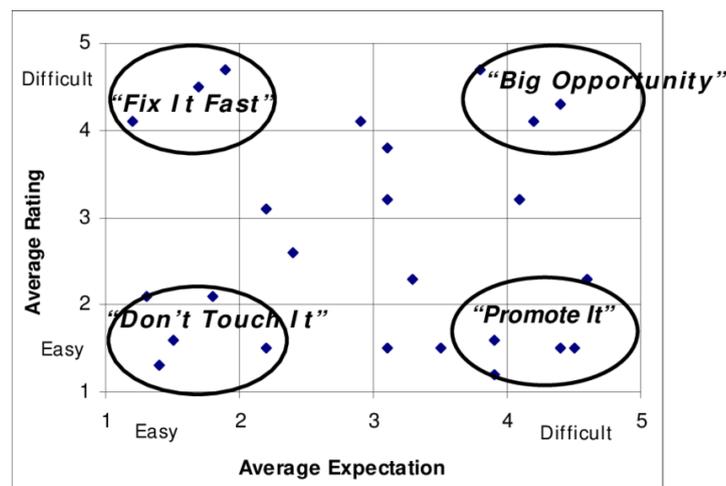


Abbildung 4.1: Das Streudiagramm des ER von [Albert \(2003\)](#).

- *Usability Magnitude Estimation (UME)* ist ein Fragebogen, der sich von den anderen Fragebögen deutlich unterscheidet. Anstatt vorgefertigte Antworten einer Likert-Skala zu wählen, muss der Proband mit diesem Fragebogen selbst eine Skala anlegen, indem er die empfundene Schwierigkeit einer Aufgabe mit den anderen Aufgaben vergleicht. Für die erste Aufgabe muss der Proband eine geeignete Schwierigkeit wählen und basierend darauf die folgenden Aufgabenbewertungen aufbauen. Die Berechnung hier-

für wird als recht komplex angesehen und ist folglich nicht für Anfänger zu empfehlen (Panagiotidi, 2021a).

- *NASA Task Load Index (NASA-TLX)* ist ein Fragebogen, der von der NASA entwickelt wurde und aus sechs Fragen besteht. Die Antwortmöglichkeiten werden auf einer 21-stufigen Skala von "sehr gering" zu "sehr hoch" repräsentiert. Mithilfe dieses Fragebogens wird die wahrgenommene Arbeitsbelastung bewertet, um die Wirksamkeit einer Aufgabe zu beurteilen. Über die sechs Fragen werden die mentale Anforderung, die körperliche Anforderung, die zeitliche Anforderung, die Leistung im Sinne des Arbeitserfolgs, der empfundene Aufwand sowie die Frustration gemessen (Birkett, 2021).
- *Single Ease Question (SEQ)* ist ein Fragebogen, der aus einer Frage besteht und die Antwort wird dabei auf einer siebenstufigen Likert-Skala dargestellt. Die Frage richtet sich dabei nach der empfundenen Schwierigkeit der Aufgabe. Die Antwortmöglichkeiten reichen dabei von "sehr schwer" bis "sehr leicht" (Laubheimer, 2018).

Die Wahl des Post-Tasks-Fragebogens wurde durch Falko Galperin inspiriert. In seiner Bachelorarbeit beschrieb Galperin (2021) den Fall, dass eine Aufgabe trotz geringer Komplexität, sehr aufwändig in Form von mühselig sein kann. Daher ist es ratsam, einen Fragebogen zu wählen, der beide Komponenten berücksichtigt. Deshlab und aufgrund der Länge wurde der ASQ gewählt.

Interessant wäre hier auch der NASA-TLX gewesen, aufgrund seiner enormen Vielfalt an Daten, die er erfassen kann. Jedoch wurde dieser nicht gewählt, um die Probanden nicht zu ermüden. Zudem wäre eine Kombination mit den ER interessant gewesen, um das Verhältnis zwischen erwarteter und tatsächliche Schwierigkeit zu erfassen. Auch dies wurde zur Vermeidung der Ermüdung nicht getan, da die Aufgabenbewältigung aufgrund der hohen Anzahl zu testender Aktionen entsprechend lang dauert.

4.2.2 Pilotstudie

Die Studie wurde vor dem offiziellen Beginn mit einem Probanden erprobt. Dadurch wurde die Bearbeitungszeit von ca. 1,25h ermittelt. Die dabei gesammelten Daten wurden nicht für die Studie verwendet.

Während der Pilotstudie fiel auf, dass eine Datenschutzerklärung für die Probanden fehlte. Außerdem wurden vereinzelt Rechtschreibfehler in Form von Tippfehlern entdeckt und Formulierungsverbesserungen geäußert. Es wurde empfohlen, weitere Hinweise für die Aufgaben

hinzuzufügen, um die Aufgabenstellung noch präziser zu gestalten.

Eine weitere wichtige Erkenntnis aus der Pilotstudie war, dass die Probanden in der Einführung lediglich die drei benannten Aktionen verwenden sollten, um Lerneffekte zu vermeiden und die Einführung kurz zu halten. Des Weiteren wurde festgestellt, dass die zur Verfügung gestellte Zip-Version in Wirklichkeit eine Rar-Version war.

Zudem wurde bemerkt, dass bei der erstmaligen Ausführung der Integration ein Firewall-Hinweis eingeblendet wird, der bestätigt werden muss. Des Weiteren wurde vorgeschlagen, das Lösungsbild zu jeder Aufgabe dauerhaft einzublenden, statt es ein- und ausblendbar zu gestalten.

Darüber hinaus wurde dazu geraten, die Aufgaben an einigen Stellen zu kürzen, um sie weniger aufwändig zu gestalten und um Zeit zu sparen. Dadurch sollte die Bearbeitungszeit auf ca. eine Stunde reduziert werden. Dafür wurde beispielsweise das Editieren der Mind Map-Umrandungen, -Texte und -Verbindungslinien entfernt und die zu erstellende Mind Map vereinfacht.

Des Weiteren wurde dazu geraten, dass die Probanden die Studie am besten im Beisein des Studieninitiators durchführen sollten. Deshalb wurde vorgeschlagen, eine Abfrage zu integrieren, um festzustellen, ob die Studie alleine, mit dem Initiator per Zoom, oder im selben Raum durchgeführt wurde.

Abschließend wurden einige Wünschänderungen geäußert, die anschließend umgesetzt wurden, wie beispielsweise die Tooltips für die Textstyle-Buttons oder die Möglichkeit, die Außenfarbe eines Textes über einen gesonderten Button zu aktivieren und deaktivieren.

4.2.3 Tools

Damit die Studie ortsunabhängig durchgeführt werden konnte, wurde sie über die Online-Plattform namens "KoboToolbox" bereitgestellt. Diese Plattform wurde von (Galperin, 2021) vorgeschlagen und ermöglicht das Erstellen von individuellen Formularen zur Datenerfassung. Dazu bietet die Plattform zwei Optionen. Zum einen kann das Formular über einen sogenannten Formulareditor direkt auf der Online-Plattform bearbeitet werden (siehe [Abbildung 4.2](#)). Mittels der Abbildung lässt sich erkennen, dass verschiedene Eingabemöglichkeiten über das Formular zur Verfügung gestellt werden. So kann für die Frage "Wie alt bist du?" nur eine Zahl eingetragen werden, während für "Wähle dein Geschlecht" die drei bereits genannten Optionen zur Verfügung gestellt werden. Die entsprechenden XML-Werte dienen später zur

Auswertung. Zum anderen können die Felder für das Formular über eine XLS-Datei konfiguriert werden.¹ Für die Erstellung dieser Studie wurden beide zur Verfügung gestellten Optionen verwendet, da einige Konfigurationen, wie beispielsweise das Darstellen von Bildern, nur über die Konfiguration der XLS-Datei möglich sind.

The screenshot displays the 'Demografische Fragen' (Demographic Questions) section of a KoboToolbox form editor. It contains three main question blocks:

- Question 123:** '* Wie alt bist du?' (How old are you?). It includes a 'Hinweis zur Frage' (Question Hint) and a settings icon.
- Question 124:** '* Wähle dein Geschlecht' (Choose your gender). It includes a 'Hinweis zur Frage' and a settings icon. The response options are:
 - Männlich (XML-Wert: maennlich)
 - Weiblich (XML-Wert: weiblich)
 - Divers (XML-Wert: divers)
 - + Klicken Sie um eine weitere Antwort hinzuzufügen... (XML-Wert: AUTOMATISCH)
- Question 125:** '* Wähle deinen bisher höchsten Schulabschluss' (Choose your highest level of education). It includes a 'Hinweis zur Frage' and a settings icon. The response options are:
 - Doktorgrad (XML-Wert: doktorgrad)
 - Master (XML-Wert: master)
 - Bachelor (XML-Wert: bachelor)
 - Allgemeine Hochschulreife (XML-Wert: allgemeine_hochschulreife)
 - Fachhochschulreife (XML-Wert: fachhochschulreife)
 - Mittlerer Abschluss (XML-Wert: mittlerer_abschluss)

Abbildung 4.2: Der Formulareditor des KoboToolbox-Tools.

Zusätzlich zu der Verwendung von KoboToolbox wurde Zoom für die Online-Treffen eingesetzt, um die Interaktion zwischen Proband und Studieninitiator zu ermöglichen. Alternativ konnte der Computer des Studieninitiators für die Durchführung verwendet werden.

4.3 PROBANDEN UND OBJEKT DER STUDIE

Für die Studiendurchführung konnten vierzehn Probanden gewonnen werden. Bei der Auswahl der Probanden wurde darauf geachtet, verschiedene Altersgruppen sowie einen bestimmten Frauenanteil abzudecken. Zudem wurden entsprechend der zweiten Forschungsfrage sowohl Informatiker als auch Nicht-Informatiker eingeladen.

¹ Das erstellte Formular ist als Datei "Formular_der_Studie.xlsx" auf dem mitgelieferten USB-Stick enthalten und ermöglicht die Betrachtung der Konfigurationsdatei des Formulars.

Der Frauenanteil in der Informatik entsprach laut [Friedrich u. a. \(2018\)](#) im Jahr 2011 17,1% und ist laut Berechnungen seit dem Jahr 1999 konstant bei 17%. Außerdem beschreiben [Friedrich u. a. \(2018\)](#), dass der Frauenanteil im Studienbereich Informatik 19,4% beträgt.

Für das Jahr 2021 wurde laut [Liebig \(n.d.\)](#) ein Anteil von 19% verzeichnet, wodurch Deutschland im europäischen Vergleich im hinteren Mittelfeld liegen soll. Spitzenreiter sind Bulgarien mit 28%, gefolgt von Rumänien mit 26% und Malta mit 25% Frauenanteil.

Daher wurde darauf geachtet, dass dieser Prozentsatz an Frauen mindestens vertreten ist. Die Probanden waren zu 28,57% weiblich, wodurch mindestens der prozentuale Frauenanteil in der Informatik europaweit abgedeckt wird.

Für die Unterteilung in Informatiker und Nicht-Informatiker konnten sechs Informatiker gewonnen werden, wobei einer der sechs Probanden weiblich war. Dies entspricht einem Frauenanteil von 16,67%, was leider etwas unter dem deutschen Frauenanteil in der Informatik liegt. Von den acht Nicht-Informatikern waren drei weiblich, was einem Frauenanteil von 37,5% entspricht und somit über dem deutschen Frauenanteil liegt.

Durch die Probanden waren die Altersgruppen von 15 bis 59 Jahren vertreten, jedoch waren die meisten Probanden zwischen Anfang zwanzig und Ende dreißig. Die Eigenschaften der Probanden, die für diese Studie relevant und deshalb erfragt werden müssen, wurden bereits im Abschnitt "[Versuchsaufbau](#)" als unabhängige Variablen beschrieben.

Sieben der vierzehn Probanden nahmen an einer zweiten Durchführung der Studie zur Beantwortung der dritten Forschungsfrage teil. In diesem zweiten Studiendurchlauf wurde exakt dieselbe Studie und dieselben Aufgaben durchgeführt und die gleiche Integration untersucht wie in dem ersten Durchlauf.

[Nielsen \(2000\)](#) bezeichnet großangelegte Benutzerfreundlichkeit-Tests als eine Verschwendung von Ressourcen. Er beschreibt, dass die besten Ergebnisse erzielt werden, wenn mit höchstens fünf Probanden getestet wird und so viele kleinere Tests durchführt, wie man sich leisten kann. Mit fünf Probanden sollen laut Nielsen 85% der Benutzerfreundlichkeitsprobleme erkannt werden. Fehler sollen mit einer Wahrscheinlichkeit von 31% von einem Probanden erkannt werden. Seiner Meinung nach werden nur mehr Probanden benötigt, wenn beispielsweise eine Webseite sehr unterschiedliche Benutzergruppen anspricht.

[Faulkner \(2003\)](#) zeigte, dass diese Ergebnisse stark von den gewählten Probanden abhängt. Sie führte eine Studie mit 60 Probanden durch, bei der zufällig ausgewählte Gruppen von fünf Probanden mindestens

zwischen 55% bis 99% der Probleme fanden. Außerdem zeigte sie, dass eine Gruppe von zehn Probanden mindestens 82% der Fehler finden und fünfzehn bereits mindestens 90%. Ab zwanzig Probanden finden die Gruppen mindestens 95% der Probleme. Sie führt die Auflistung bis zu einer Gruppengröße von 50 Probanden in Zehnerschritten fort. Jedoch war der prozentuale Zuwachs für diese Steigerungen gering. Aus diesem Grund wurde versucht, so viele Probanden wie möglich zu gewinnen, um die bestmöglichen Ergebnisse zu erzielen.

Das zu untersuchende Objekt der Studie ist die gesamte Drawable-Integration mit allen angebotenen Aktionen. Aufgrund der zeitlichen Begrenzung der Studie und zur Vereinfachung des Studienaufwands wird jedoch nicht jede spezifische Einzelheit der Aktionen überprüft, sondern nur die allgemeine Funktionalität. Beispiele hierfür sind das Editieren der Mind Map-Knotenumrandung sowie die anderen entfernten Teilaufgaben des Abschnittes "Pilotstudie".

4.4 GESTELLTE AUFGABEN

Der Studie lagen sieben verschiedene Aufgaben zugrunde, die von den Probanden absolviert werden mussten, um sämtliche Aktionen der Integration zu überprüfen. In den folgenden Unterabschnitten werden die einzelnen sieben Aufgaben vorgestellt. Die Erläuterungen für die einzelnen Aufgaben umfassen die verwendeten Aktionen, eine kurze Zusammenfassung, die einzelnen Schritte, die die Probanden durchführten, gegebenenfalls ein Hinweisbild und abschließend immer ein Lösungsbild. Zu den einzelnen Schritten wurden jeweils Hinweise angegeben, diese werden in grauer Farbe dargestellt.

4.4.1 Aufgabe 1

Verwendete Aktionen: Draw Shape-Aktion, Line Split-Aktion, Edit-Aktion, Draw Freehand-Aktion und Change The Sorting Layer-Aktion

Zusammenfassung: In dieser Aufgabe sollte der Proband ein dreistufiges schwarzes Podest mit einer Linie zeichnen und den Bereich des Erstplatzierten rot einfärben. Anschließend sollte das Podest mit einer blauen Linie übermalt werden und die Reihenfolge der überdeckenden Linie so verändert werden, dass sie einen blauen Hintergrund für das Podest bildet.

Einzelne Schritte:

1. Zeichne ein dreistufiges Podest (Siegertreppechen / Siegerpodest) in Schwarz mittels der Draw Shape-Aktion an einer geeigneten Stelle, zum Beispiel unten links. Verwende hierfür die Shape-Art

“Line“. Bitte schau dir im [Lösungsbild](#) die ungefähre Relation an, wie groß das Podest werden soll. Bevor du mit dem Zeichnen beginnst, stelle bitte die folgenden Standardwerte wieder her:

- Thickness: 0,01
- Line kind: Solid
- Color kind: Monochrome

Beginne nun mit dem Zeichnen mit der Bodenlinie.

Im Hinweisbild ist der geeignete Startpunkt **grün** markiert. Das Zeichnen kann mittels des “Finish“ - Buttons (im Shape-Menü-Tab) oder einem Linksklick mit gedrückter “Strg“-Taste beendet werden. Die Farbe kann in dem “Config“-Menü-Tab ausgewählt werden. Tipp: Wechsle nach dem Wählen der Farbe wieder in den “Shape“-Tab.

2. Splitte (Teile) das eben gezeichnete Podest an den im [Hinweisbild](#) gekennzeichneten Punkten. Durch das Teilen wird der erste Platz vom Podest abgespaltet.

Verwende dazu die [Line Split-Aktion](#). Im [Hinweisbild](#) sind die beiden Splitpunkte **blau** markiert.

3. Editiere die eben abgetrennte Linie (Platz 1 Podest) so, dass die Linie rot gefärbt ist.

Verwende dazu die [Edit-Aktion](#). Das Bearbeiten kann durch einen linken Mausklick auf dem Drawable beendet werden.

4. Übermale das dreistufige Podest vollständig mittels der [Draw Freehand-Aktion](#) mit einer Linie. Konfiguriere die Linie vorher so, dass sie hellblau ist und den maximalen “Thickness“ Wert hat. (Siehe [Lösungsbild](#))

5. Verändere die Objekt-Reihenfolge der eben gezeichneten hellblauen Linie so, dass das Podest im Vordergrund steht. Damit ist gemeint, dass das Podest wieder vollständig sichtbar ist. Schau im [Lösungsbild](#) nach, falls noch Unklarheiten bestehen.

Verwende dazu die rechte Maustaste in der [Change The Sorting Layer-Aktion](#). Alternativ könnte die Reihenfolge auch über die [Edit-Aktion](#) verändert werden.



Abbildung 4.3: Hinweisbild der Aufgabe 1.

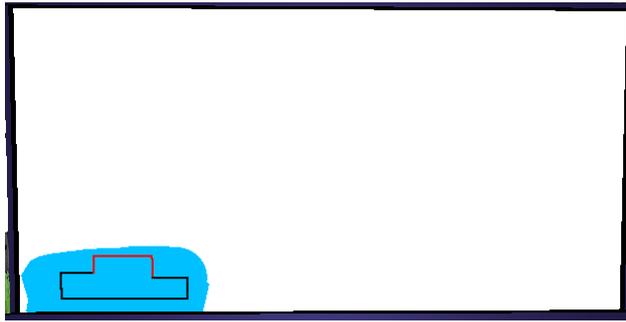
Lösungsbild:

Abbildung 4.4: Lösungsbild der Aufgabe 1.

4.4.2 Aufgabe 2

Verwendete Aktionen: [Write Text-Aktion](#), [Cut](#), [Copy](#), [Paste-Aktion](#), [Edit-Aktion](#), [Move or Rotate-Aktion](#), [Save-Aktion](#), [Cleaner-Aktion](#) und [Load-Aktion](#)

Zusammenfassung: In dieser Aufgabe sollte der Proband drei Texte mit den Inhalten "1", "2" und "3" erstellen, die jeweils unterschiedlich konfiguriert sind, und außerhalb des Podestes platzieren. Anschließend wurde ein Text editiert und alle an ihre entsprechenden Positionen im Podest verschoben. Danach sollte der Proband das Drawable speichern, säubern und abschließend erneut laden.

Einzelne Schritte:

1. Schreibe einen unterstrichenen Text mit dem Inhalt "1", wähle folgende Eigenschaften für den Text:
 - Unterstrichen (U-Button)
 - Schriftgröße: 1
 - Schriftfarbe (Font color): Gold
 - Außenfarbe (Outline color): Rot

Platziere den Text außerhalb des Podestes.

Verwende dazu die [Write Text-Aktion](#). Die Farbe "Gold" kann aus den Farbvoreinstellungen gewählt werden.

2. Schreibe einen weiteren Text mit dem Inhalt "2". Der Text soll dabei **nicht** unterstrichen sein, eine Schriftgröße von 0,75 haben, eine Schriftfarbe von Grau/Silber aufweisen und eine schwarze Außenfarbe besitzen. Platziere den Text außerhalb des Podestes.
3. Kopiere den zuletzt geschriebenen Text und füge ihn mittels eines linken Mausklicks ein. Editiere ihn nun wie folgt: Der Inhalt soll zu "3" geändert werden. Zusätzlich soll der Text kursiv (I-Button) sein, eine Schriftgröße von 0,5 und eine Schriftfarbe von Bronze (# 6E4D25) aufweisen. Die Außenfarbe soll deaktiviert werden.

Platziere den Text außerhalb des Podestes.

Verwende zum Kopieren die [Cut, Copy, Paste-Aktion](#) und zum Editieren die [Edit-Aktion](#). Den Farbcode für die Schriftfarbe kann im dazugehörigen Eingabefeld eingegeben werden.

4. Bewege die drei Texte ("1", "2" und "3") in das Podest, so dass die Plätze für den ersten, zweiten und dritten Platz richtig beschriftet sind. Siehe dazu im [Lösungsbild](#) nach, um die genauen Positionen zu erfahren.

Verwende zum Bewegen / Positionieren der Texte die "Move"-Funktionalität der [Move or Rotate-Aktion](#). Neben dem bereitgestellten Menü können auch die Pfeiltasten zur Bewegung des ausgewählten Objektes verwendet werden. Zusätzlich kann mittels der "Strg"-Taste das Geschwindigkeitslevel gewechselt werden. Außerdem lässt sich mit einem Mausradklick die Bewegung mittels der Maus aktivieren und deaktivieren. Zum Beenden der Aktion wird ein linker Mausklick auf dem Drawable benötigt.

5. Speichere den Drawable (Whiteboard). Verwende dazu die [Save-Aktion](#) und wähle den Drawable zum Speichern aus. Anschließend wird der ausgewählte Drawable mittels des "Save"-Buttons gespeichert.
6. Führe die [Cleaner-Aktion](#) auf dem Drawable durch. Verwende dazu die [Cleaner-Aktion](#).
7. Lade die eben gespeicherte Datei. Diese befindet sich standardmäßig im "1. Single Drawable"-Verzeichnis. Verwende dazu die Option "Load on original drawable" der [Load-Aktion](#).

Lösungsbild:

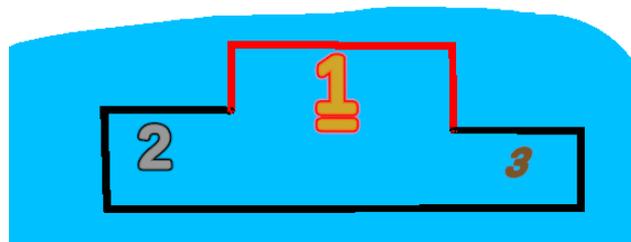


Abbildung 4.5: Lösungsbild der Aufgabe 2.

4.4.3 Aufgabe 3

Verwendete Aktionen: [Color Picker-Aktion](#), [Draw Shape-Aktion](#), [Move or Rotate-Aktion](#), [Move a Point-Aktion](#), [Edit-Aktion](#), [Cut, Copy, Paste-Aktion](#), [Line Point Erase-Aktion](#) und [Line Connection Erase-Aktion](#)

Zusammenfassung: In dieser Aufgabe sollte der Proband ein Quadrat platzieren und anschließend rotieren. Danach sollte ein Punkt des Quadrates verschoben werden und das dadurch entstandene Drachen-

viereck editiert werden. Anschließend sollte das Drachenviereck kopiert werden. Bei dem Original sollte der unterste Punkt entfernt werden und bei der Kopie die Verbindungslinie vom untersten Punkt zum rechts davon liegenden Punkt.

Einzelne Schritte:

1. Führe die [Color Picker-Aktion](#) auf der schwarzen Podestlinie für die Primärfarbe durch.
Dadurch wird die Farbe "Schwarz" als Primärfarbe ausgewählt.
2. Platziere ein Quadrat (Square) mit der Line-Thickness von 0,01 und mit der Kantenlänge "a" von 30 an einer geeigneten Stelle, zum Beispiel über dem Podest. Lasse hierbei jedoch genügend Platz zum Podest.
Verwende dazu die [Draw Shape-Aktion](#) und wähle die Shape-Art "Square" aus.
3. Rotiere das Quadrat auf 45°.
Verwende dazu die "Rotate"-Funktion der [Move or Rotate-Aktion](#). Neben dem Slider und dem Eingabefeld des Menüs lässt sich das ausgewählte Objekt auch mithilfe des Mausekzes in Ein-Grad-Schritten rotieren. Wenn dabei die "Strg"-Taste gedrückt wird, erfolgt die Rotation des Objektes in Zehn-Grad-Schritten. Zum Beenden der Aktion wird ein linker Mausklick auf dem Drawable benötigt.
4. Verschiebe den rechten Punkt des rotierten Quadrats so, dass die geometrische Form "Drachen/Drachenviereck" entsteht. Dafür muss der Punkt nach rechts verschoben werden. Schaulie hierzu gerne auf das [Hinweisbild](#).
Verwende dazu die [Move a Point-Aktion](#). Zum Beenden der Aktion wird ein linker Mausklick auf dem Drawable benötigt.

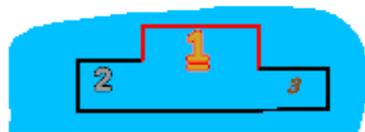
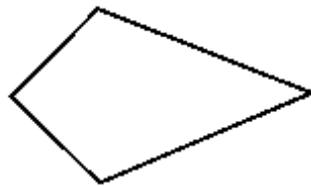


Abbildung 4.6: Hinweisbild der Aufgabe 3.

5. Editiere den Drachen so, dass
 - die Linienart (Line kind) "Dashed25"

- die Farbart (Color kind) "TwoDashed"
- die Primärfarbe rot
- und die Sekundärfarbe blau ist.

Verwende dazu die [Edit-Aktion](#). Zum Beenden der Aktion wird ein linker Mausklick auf dem Drawable benötigt.

6. Kopiere nun den Drachen und füge die Kopie rechts neben dem Original ein.

Verwende dazu die [Cut, Copy, Paste-Aktion](#).

7. Führe auf dem originalen Drachen einen "Line Point Erase" auf dem untersten Punkt durch.

Verwende dazu die [Line Point Erase-Aktion](#).

8. Führe auf der Kopie einen "Line Connection Erase" auf dem untersten Punkt durch.

Verwende dazu die [Line Connection Erase-Aktion](#).

Lösungsbild:

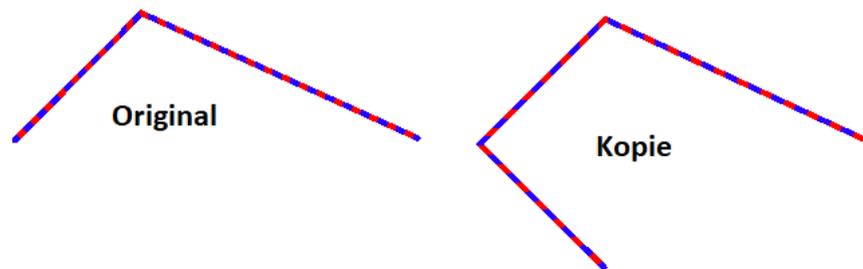


Abbildung 4.7: Lösungsbild der Aufgabe 3.

4.4.4 Aufgabe 4

Verwendete Aktionen: [Add an Image-Aktion](#), [Edit-Aktion](#) und [Scale-Aktion](#)

Zusammenfassung: In dieser Aufgabe sollte der Proband ein Webbild hinzufügen, es bearbeiten und abschließend skalieren.

Einzelne Schritte:

1. Füge das bereitgestellte Bild als Webbild hinzu. Suche dir eine geeignete Stelle auf dem Drawable, zum Beispiel oben links. Als Dateiname kann ein beliebiger Name gewählt werden. Es eignet sich zum Beispiel "Marienkaefer".

Verwende dazu die [Add an Image-Aktion](#).

2. Bearbeite das Bild so, dass das Bild rötlich eingefärbt und gespiegelt (Mirrored) wird.

Verwende dazu die [Edit-Aktion](#).

3. Das Bild wirkt derzeit viel zu groß, skaliere es deshalb auf die Maße (0,35 / 0,35).
Verwende dazu die [Scale-Aktion](#). Neben dem Menü kann erneut das Mausrad sowie die "Strg"-Taste zum Skalieren verwendet werden.

Lösungsbild:

(In der Studie wurde hier ein anderes Bild und eine andere Farbe verwendet, jedoch wurde es aufgrund rechtlicher Bedenken hier und in den nachfolgenden Abbildungen durch das bereits bekannte lizenzfreie Bild ausgetauscht. Für das ursprüngliche Bild wurde eine bläuliche Einfärbung verwendet.)



Abbildung 4.8: Lösungsbild der Aufgabe 4.

4.4.5 Aufgabe 5

Verwendete Aktionen: "Spawn" und "Move" der [Sticky-Note-Aktion](#) und [Cut, Copy, Paste-Aktion](#)

Zusammenfassung: In dieser Aufgabe sollte der Proband das zuvor hinzugefügte Bild ausschneiden und auf einem Sticky-Note platzieren. Anschließend sollte er das Sticky-Note auf einen Tisch legen.

Einzelne Schritte:

1. Erstelle ein Sticky-Note und platziere ihn auf der rechten Seite des Drawables.
Verwende dazu die "Spawn"-Funktion der [Sticky-Note-Aktion](#).
2. Schneide das eben hinzugefügte Bild aus der vorherigen Aufgabe aus und füge es mittig auf dem Sticky-Note ein.
Verwende dazu die "Cut"-Funktion der [Cut, Copy, Paste-Aktion](#).
3. Platziere das Sticky-Note auf einem Tisch und rotiere es so, dass es auf dem Tisch liegt (Laying-Button).
Verwende dazu die "Move"-Funktion der [Sticky-Note-Aktion](#). Die Steuerung erfolgt zunächst grob per Maus. Anschließend

werden zwei Menüs zur Verfeinerung bereitgestellt: eins für das Rotieren und das andere zum Bewegen. Zusätzlich zu den Menüs wird die Steuerung per Tastatur/Mausrad ermöglicht.

Lösungsbild:

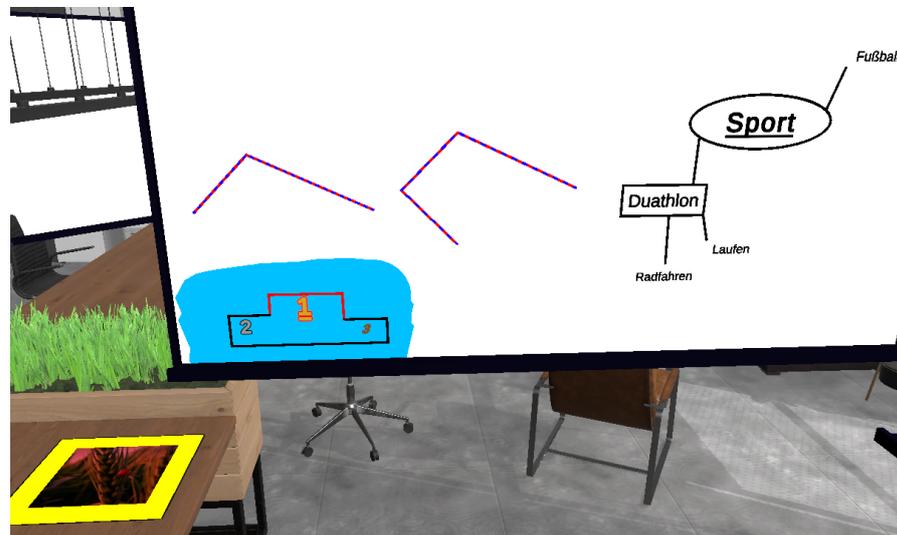


Abbildung 4.9: Lösungsbild der Aufgabe 5.

4.4.6 Aufgabe 6

Verwendete Aktionen: [Mind Map-Aktion](#), [Erase-Aktion](#) und [Edit-Aktion](#)

Zusammenfassung: In dieser Aufgabe sollte der Proband eine Mind Map zum Thema "Sport" erstellen und die Editiermöglichkeiten sowie die Besonderheit beim Löschen eines Knotens kennenlernen. Interessant an dieser Aufgabe war, dass fünf der vierzehn Probanden (35,71%) den Schritt 4 überlesen haben. Einem dieser Probanden fiel gegen Ende der Aufgabe selbstständig auf, dass dieser Schritt übersprungen wurde.

Einzelne Schritte:

1. Erstelle eine Mind Map zum Thema (Central Theme) "Sport".
Verwende dazu den "Central Theme"-Button in der [Mind Map-Aktion](#).
2. Füge die Blätter (Leaves) "Duathlon", "Laufen" und "Fußball" als Kinderknoten von Sport hinzu.
Verwende dazu den "Leaf"-Button in der [Mind Map-Aktion](#).
3. Füge das Unterthema (Subtheme) "Hilfsmittel" als Kinderknoten von Sport hinzu.
Verwende dazu den "Subtheme"-Button in der [Mind Map-Aktion](#).

4. Fügen den Leaf "Ball" als Kinderknoten von "Hilfsmittel" hinzu.
Verwende dazu den "Leaf"-Button in der [Mind Map-Aktion](#).
5. Verändere den Knoten "Duathlon" so, dass die Knotenart (Node kind) zu "Subtheme" gewechselt wird.
Verwende dazu die Funktionalität "Change Node Kind" der [Edit-Aktion](#).
6. Verändere den Knoten "Laufen", so dass er den neuen Elternknoten "Duathlon" zugewiesen bekommt.
Verwende dazu die Funktionalität "Change Parent" der [Edit-Aktion](#).
7. Fügen dem Unterthema "Duathlon" den Kinderknoten "Radfahren" hinzu.
Verwende dazu den "Leaf"-Button in der [Mind Map-Aktion](#).

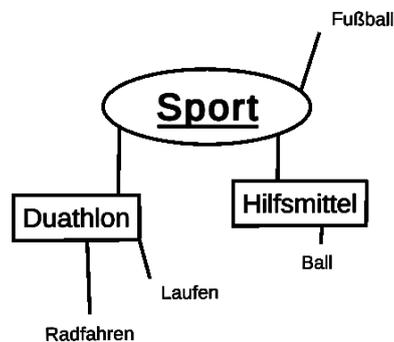


Abbildung 4.10: Zwischenbild der Aufgabe 6.

8. Ändere den Themen-Text von "Sport" zu "Sportarten".
Verwende dazu die Funktionalität "Text" der [Edit-Aktion](#).
9. Lösche den nicht passenden Unterthema-Knoten "Hilfsmittel".
Verwende dazu die [Erase-Aktion](#).

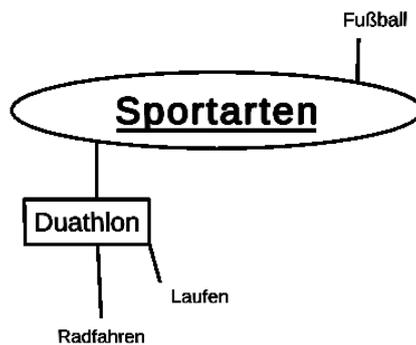
Lösungsbild:

Abbildung 4.11: Lösungsbild der Aufgabe 6.

4.4.7 Aufgabe 7

Verwendete Aktionen: [Save-Aktion](#) und [Load-Aktion](#)

Zusammenfassung: In dieser Aufgabe sollte der Proband alle Drawables speichern und die gespeicherte Datei auf einem bestimmten anderen Drawable laden.

Einzelne Schritte:

1. Speichere alle Drawables. Die Dateibenennung bleibt dir überlassen.
Verwende dazu den "Save all"-Button der [Save-Aktion](#).
2. Lade die eben gespeicherte Datei auf einem anderen Whiteboard im Raum mittels "Load on specific drawable".
Verwende dazu die [Load-Aktion](#). Wähle das Ziel Whiteboard mit einem linken Mausklick aus. Anschließend wähle die Datei zum Laden über den "Load on specific drawable"-Button aus.

Lösungsbild:

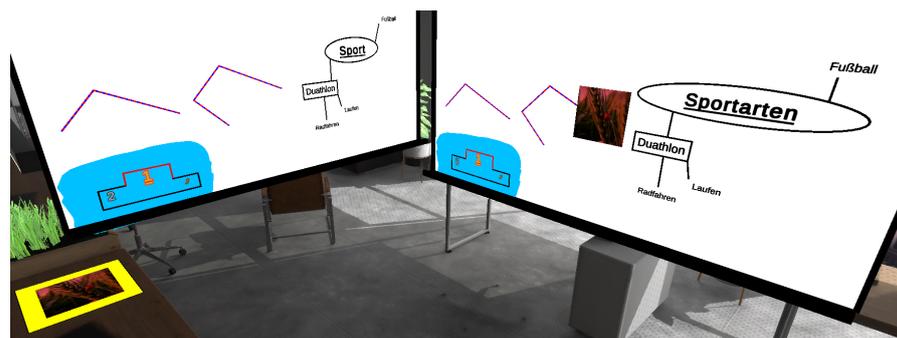


Abbildung 4.12: Lösungsbild der Aufgabe 7.

4.5 AUSWERTUNG

In diesem Abschnitt wird die Auswertung der Studie präsentiert. Alle gezeigten Ergebnisse und Diagramme können mithilfe der auf dem USB-Stick mitgelieferten Datei "calc.R" reproduziert werden. Die verwendeten Berechnungen dieser Datei basieren auf dem R-Skript² des Artikels "VISSOFT2022" von [Galperin u. a. \(2022\)](#).

Neben den im Folgenden vorgestellten Ergebnissen der Forschungsfragen deckte die Studiendurchführung einige kleinere Fehler in der Integration auf, die nicht weiter aufgeführt werden. Die Fehler waren größtenteils Anzeigefehler. Einige geäußerte Verbesserungswünsche

² https://github.com/uni-bremen-agst/VISSOFT2022/blob/main/calc_results.R

wurden ebenfalls direkt umgesetzt.

Zunächst werden die **demografischen Daten** ausgewertet, gefolgt von der Auswertung der **ersten Forschungsfrage**. Danach folgen die Auswertungen der **zweiten** sowie **dritten** Forschungsfrage, ein **Vergleich der Ergebnisse des zweiten Durchlaufs der Nicht-Informatiker und des ersten Durchlaufs der Informatiker** sowie die Beschreibung der gefundenen **Korrelationen** und abschließend die **Kommentare der Probanden**.

Zur Erinnerung, wie bereits in dem Abschnitt **“Ziele und Forschungsfragen”** beschrieben, wird die Gruppe der Informatiker in den folgenden Diagrammen als α und die der Nicht-Informatiker als β bezeichnet.

4.5.1 Demografische Daten

In diesem Abschnitt werden die demografischen Daten der Probanden ausgewertet. Hierfür werden sowohl die Daten aller Probanden als auch der beiden Gruppen der Informatiker und Nicht-Informatiker analysiert.

Die signifikanten Unterschiede der beiden Gruppen werden hierbei mittels des *Mann-Whitney-U-Tests* mit einem Signifikanzniveau von $p = 0,05$ überprüft. Dieser Test wurde gewählt, da er keine Normalverteilung der Daten erfordert und mit ordinalskalierten Daten verwendet werden kann.

- **Informatik Zugehörigkeit:**

Wie bereits im Abschnitt **“Probanden und Objekt der Studie”** erwähnt, haben sechs Informatiker und acht Nicht-Informatiker an der Studie teilgenommen. In dem folgenden Balkendiagramm der **Abbildung 4.13** können die Antworten zu der Frage **“Stammst du aus dem Informatikbereich”** ausgewertet werden. Dabei repräsentieren logischerweise die Nein-Antworten die Nicht-Informatiker und die Ja-Antworten die Informatiker.

Mann-Whitney-U-Test: Mit diesem Test wird überprüft, ob es einen signifikanten Unterschied zwischen zwei Gruppen gibt. Für diese Überprüfung wird die Rangsumme der Gruppen gebildet und für den Vergleich genutzt. Zur Durchführung des Tests werden lediglich zwei unabhängige Stichproben mit mindestens Ordinalskalenniveau benötigt. Es ist keine spezifische Verteilungskurve erforderlich (DATAtab_Team, 2024b). Dieser Test ist auch als Wilcoxon-Mann-Whitney-Test oder Wilcoxon-Rangsummentest bekannt (Hemmerich, 2024).

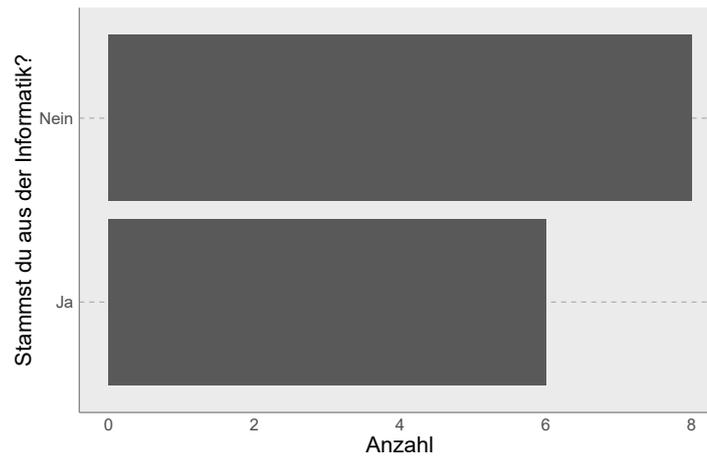


Abbildung 4.13: Stammst du aus dem Informatikbereich?

- **Geschlecht:**

Ebenfalls im Abschnitt [“Probanden und Objekt der Studie“](#) wurde bereits erwähnt, dass 28,57% der Probanden weiblich waren, das bedeutet, dass vier von den vierzehn Probanden weiblich waren. Die restlichen zehn Probanden waren männlich. Keiner war divers. Diese Ergebnisse können in der [Abbildung 4.14](#) betrachtet werden. Ein Proband der sechs Informatiker war weiblich und drei der acht Nicht-Informatiker. Das Violin-Plot-Diagramm hierfür kann in der [Abbildung 4.15](#) betrachtet werden. Der Unterschied zwischen den beiden Gruppen war nicht signifikant ($U = 19; p \approx 0,459$). Der Median für beide Auswertungen ist *“Männlich“*.

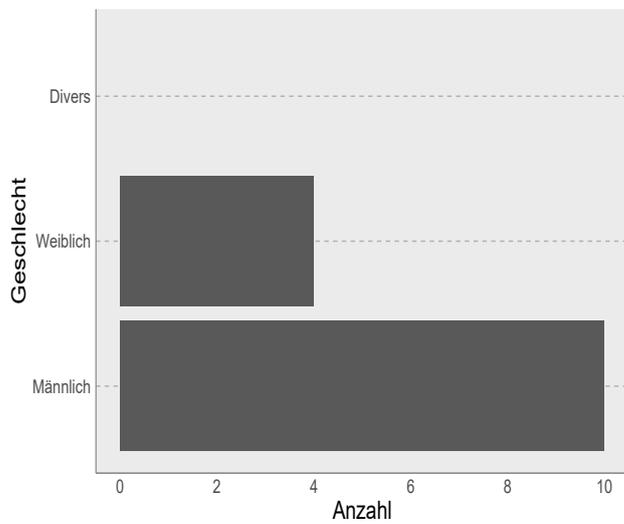


Abbildung 4.14: Verteilung des Geschlechts aller Probanden.

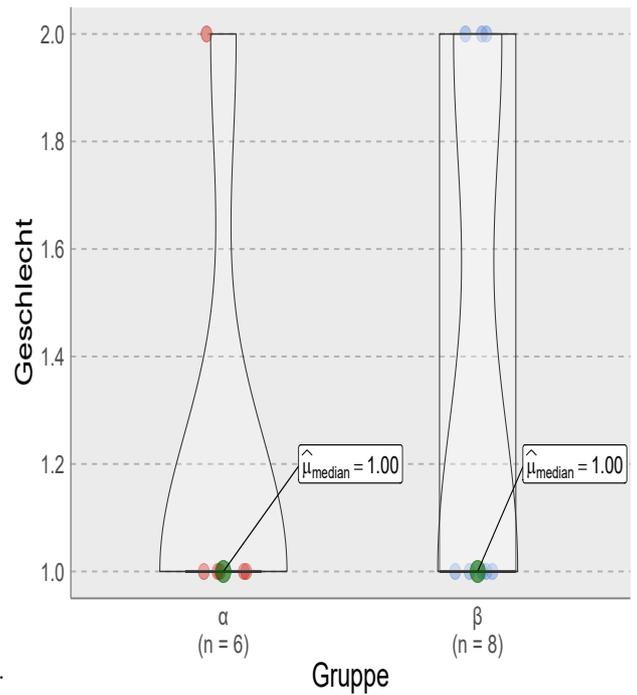


Abbildung 4.15: Verteilung des Geschlechts in beiden Gruppen.

- **Alter:**

Der Altersbereich der Probanden reichte von 15 Jahren und bis 59 Jahre. Das durchschnittliche Alter aller Probanden betrug $\bar{X} \approx 30,43$ Jahre, während der Median $\tilde{X} = 29$ Jahre betrug. Die Standardabweichung wies einen Wert von $S \approx 10,06$ Jahre. Die Verteilung kann in der [Abbildung 4.16](#) betrachtet werden. Die Informatiker hatten ein durchschnittliches Alter von $\bar{X} \approx 27,33$ Jahren und einen Median von $\tilde{X} = 25,5$ Jahren. Die Standardabweichung betrug $S \approx 5,57$ Jahre. Bei den Nicht-Informatikern betrug das durchschnittliche Alter $\bar{X} = 32,75$ Jahre und der Median $\tilde{X} = 31$ Jahre. Die Standardabweichung war hierbei $S \approx 12,3$ Jahre. In der [Abbildung 4.17](#) ist die Verteilung des Alters der beiden Gruppen zu sehen. Hierbei wurde kein signifikanter Unterschied festgestellt ($U = 13,5; p \approx 0,195$).

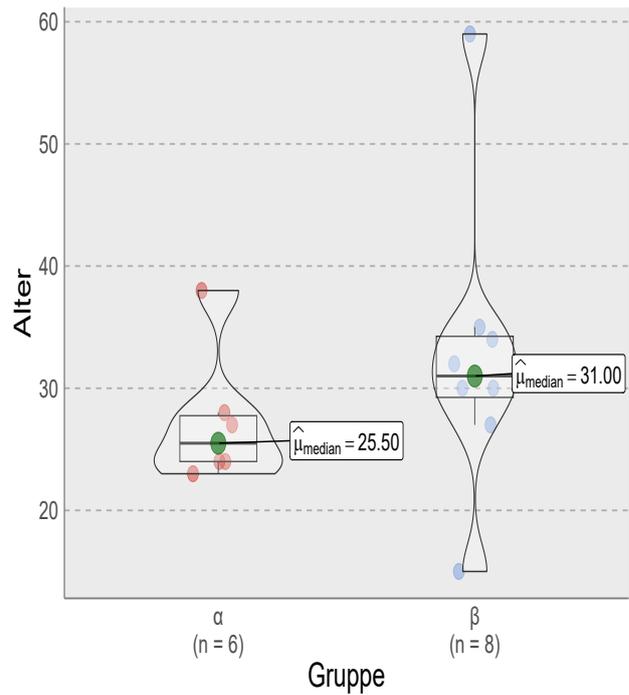
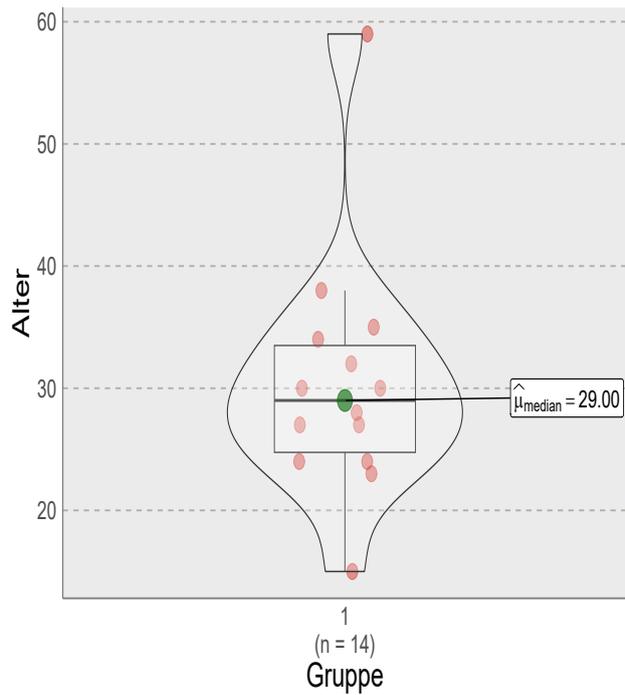


Abbildung 4.16: Alter aller Probanden.

Abbildung 4.17: Verteilung des Alters beider Gruppen.

- **Höchster Abschluss:**

In der [Abbildung 4.18](#) ist zu sehen, dass fast alle deutschen Bildungsabschlüsse abgedeckt wurden, lediglich der Hauptschulabschluss, die Fachhochschulreife und der Doktorgrad wurden nicht abgedeckt. "Ohne Abschluss" ist hierbei gleichzusetzen mit "noch in der Schule", da dieser Proband zum Zeitpunkt der Studie 15 Jahre alt war. Der Median für alle Probanden betrug den Abschluss "Allgemeine Hochschulreife". Der [Abbildung 4.19](#) kann entnommen werden, dass der Median der Nicht-Informatiker bei dem mittleren Schulabschluss liegt und der Median der Informatiker zwischen der Allgemeinen Hochschulreife und dem Bachelor. Zwischen den beiden Gruppen wurde kein signifikanter Unterschied festgestellt ($U = 36; p \approx 0,125$). Aufgrund der Tatsache, dass der höchste Bildungsabschluss ein ordinalskalierter Wert darstellt, ist es nicht möglich, ein Durchschnittswert anzugeben, da für diesen eine Intervallskala benötigt wird ([Berlin, 2024](#); [Wikipedia, 2024b](#)). Dasselbe gilt für die nachfolgenden demografischen Daten.

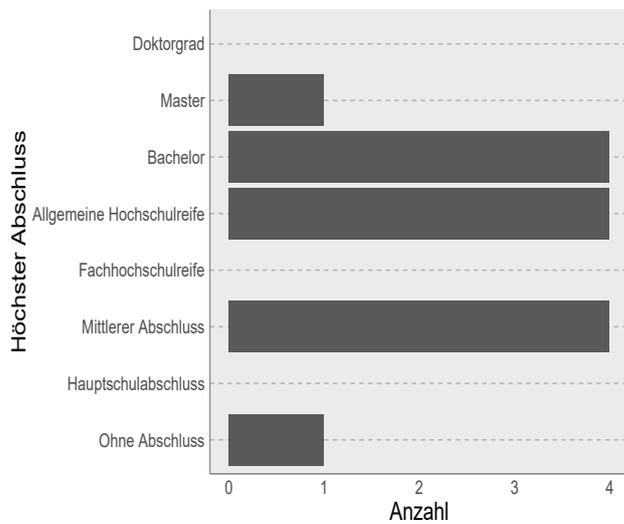


Abbildung 4.18: Verteilung des höchsten Abschlusses aller Probanden.

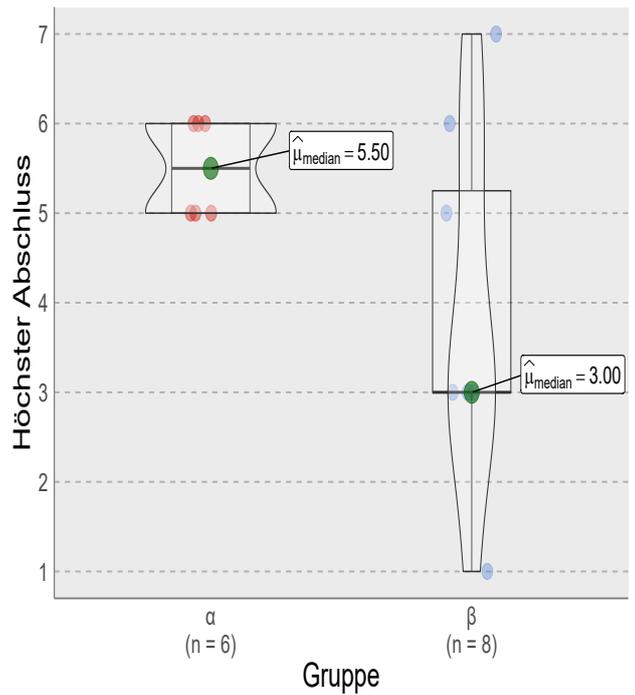


Abbildung 4.19: Verteilung des höchsten Abschlusses beider Gruppen.

- **SEE-Erfahrung:**

Für die Bewertung der Erfahrung mit SEE wurden die vier genannten Möglichkeiten "Keine", "Von gehört", "Bereits genutzt" und "Daran entwickelt" bereitgestellt. Die Antworten der Probanden kann der [Abbildung 4.20](#) entnommen werden. Der Median aller Probanden lag in diesem Fall bei der Option "Keine"-Erfahrung. Der Median der Informatiker lag bei der Option "Daran entwickelt", was auf die Tatsache zurückzuführen ist, dass ausschließlich SEE-Informatiker teilnahmen. Verständlicherweise war der Median der Nicht-Informatiker bei "Keine"-Erfahrung. Der Unterschied zwischen den beiden Gruppen war daher logischerweise signifikant ($U = 48; p = 0,000552$) und kann in der [Abbildung 4.21](#) betrachtet werden.

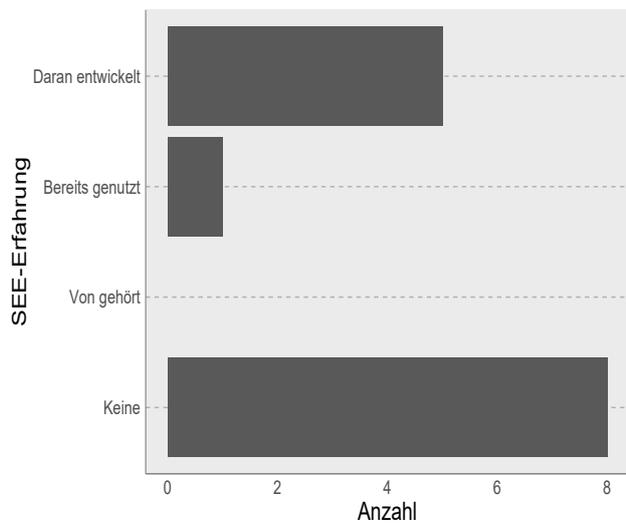


Abbildung 4.20: Verteilung der SEE-Erfahrung aller Probanden.

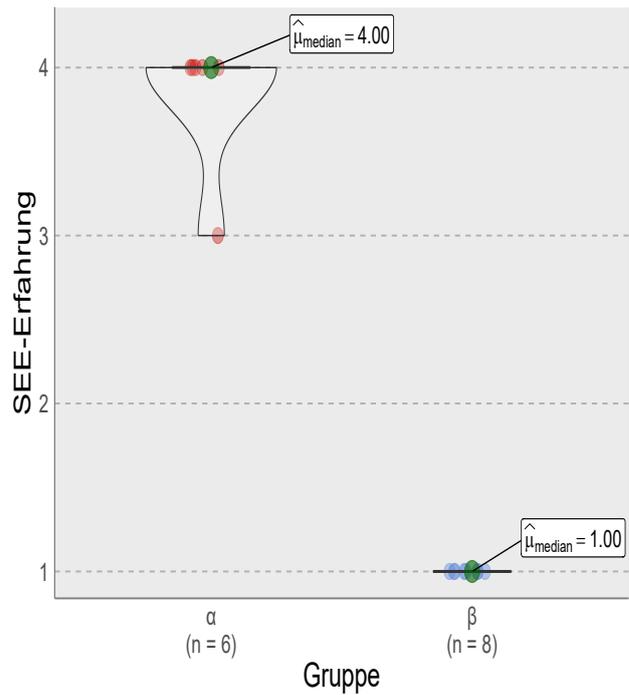


Abbildung 4.21: Verteilung der SEE-Erfahrung beider Gruppen.

- **Bildbearbeitungsprogramm-Erfahrung:**

Auch bei dieser Erfahrung wurden die bereits bekannten Optionen "Keine", "Kaum genutzt", "Etwas genutzt" und "Viel genutzt" zur Verfügung gestellt. Die Verteilung der Beantwortung aller Probanden kann in der [Abbildung 4.22](#) betrachtet werden. Der Median hierbei beträgt die Option "Etwas genutzt". Die Verteilung beider Gruppen kann in der [Abbildung 4.23](#) betrachtet werden. Hierbei ist ersichtlich, dass der Median der Informatiker ebenfalls bei "Etwas genutzt" liegt und bei den Nicht-Informatikern bei "Kaum genutzt". Leider ist die Option "Viel genutzt" nicht vertreten, wobei Probanden dieser Option sicherlich interessante Resultate liefern würden, da die zugrundeliegende Integration einem Bildbearbeitungsprogramm ähnelt. Zwischen den Informatikern und Nicht-Informatikern wurde ein signifikanter Unterschied bei der Beantwortung dieser Erfahrung festgestellt ($U = 38,5; p \approx 0,043$).

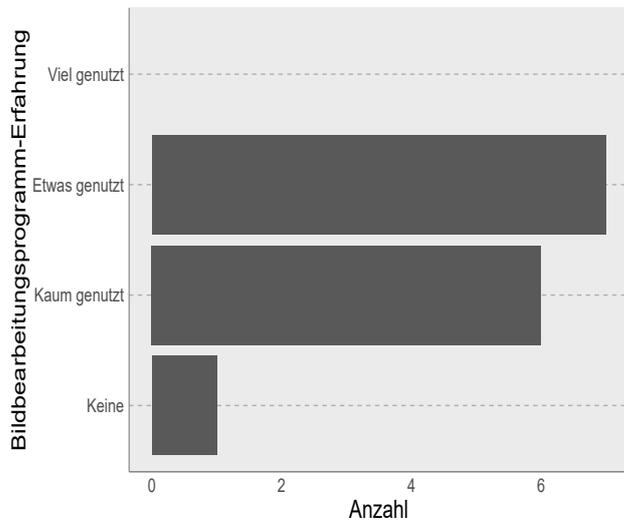


Abbildung 4.22: Verteilung der Bildbearbeitungsprogramm-Erfahrung aller Probanden.

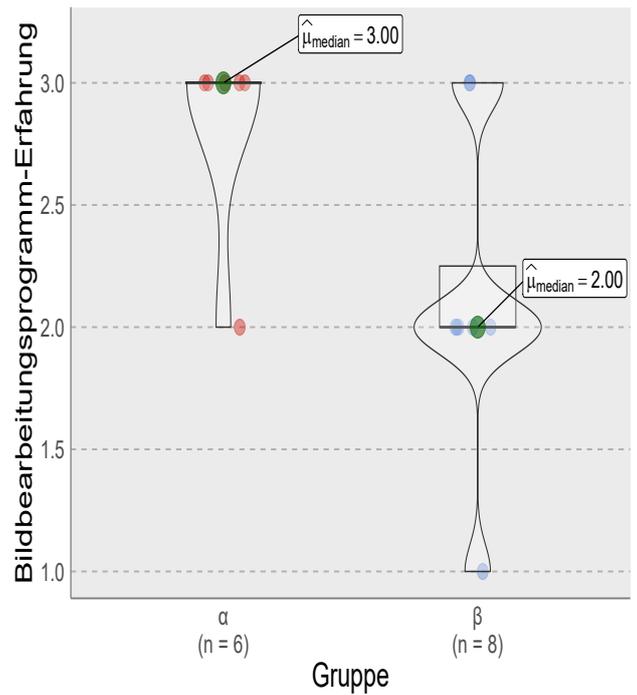


Abbildung 4.23: Verteilung der Bildbearbeitungsprogramm-Erfahrung beider Gruppen.

- **Videospiel-Erfahrung:**

Bei der Beantwortung dieser Erfahrung standen dieselben Optionen wie bei der Beantwortung der Bildbearbeitungserfahrung zur Verfügung. Die Antworten aller Probanden kann der [Abbildung 4.24](#) entnommen werden, wobei ersichtlich ist, dass die meisten Probanden über viel Videospielderfahrung verfügen. Der Median für alle Probanden sowie für beide Gruppen liegt bei dieser Option. Die Verteilung beider Gruppen ist in der [Abbildung 4.25](#) zu sehen. In beiden Abbildungen ist zu erkennen, dass kein Probanden angegeben hat, keine Videospielderfahrung gesammelt zu haben, was jedoch in der heutigen Zeit nicht ungewöhnlich ist. Für die beiden Gruppen wurde kein signifikanter Unterschied festgestellt ($U = 27; p \approx 0,705$).

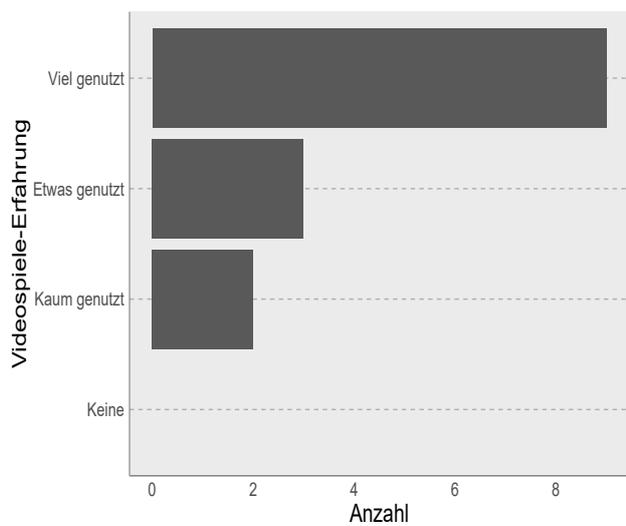


Abbildung 4.24: Verteilung der Videospiel-Erfahrung aller Probanden.

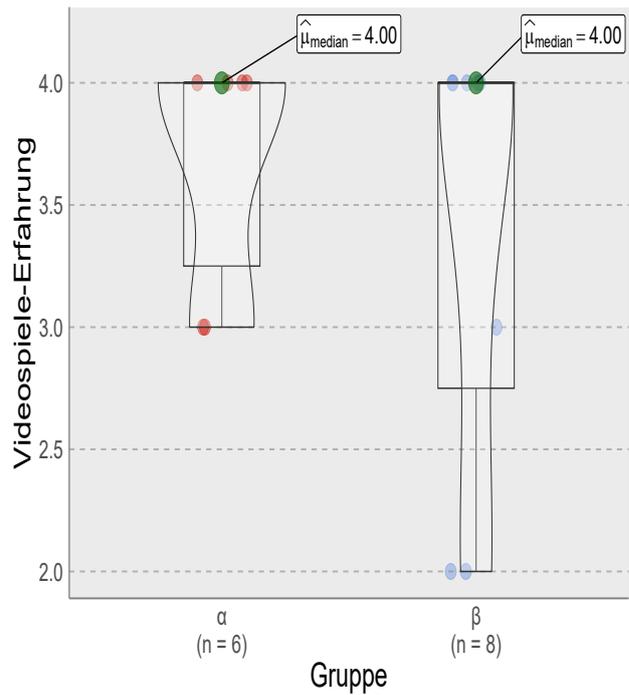


Abbildung 4.25: Verteilung der Videospiel-Erfahrung beider Gruppen.

- **Studiendurchführung:**

Durch diese Abfrage wurde überprüft, wie die Probanden die Studie durchgeführt haben. Durch die [Abbildung 4.26](#) ist ersichtlich, dass kein Proband die Studie alleine durchgeführt hat. Die anderen beiden Antworten waren zu gleichen Teilen vorhanden. Die Verteilung beider Gruppen kann in der [Abbildung 4.27](#) betrachtet werden. Dabei zeigt sich, dass die Studiendurchführung der Informatiker ausschließlich online erfolgte. Die Nicht-Informatiker nutzten bis auf einer alle den Computer des Studieninitiators, weshalb der Median logischerweise bei der Option "Ja, ist physisch im selben Raum" lag. Aufgrund dieser Tatsache wurde ein signifikanter Unterschied zwischen beiden Gruppen festgestellt ($U = 3; p \approx 0,002$).

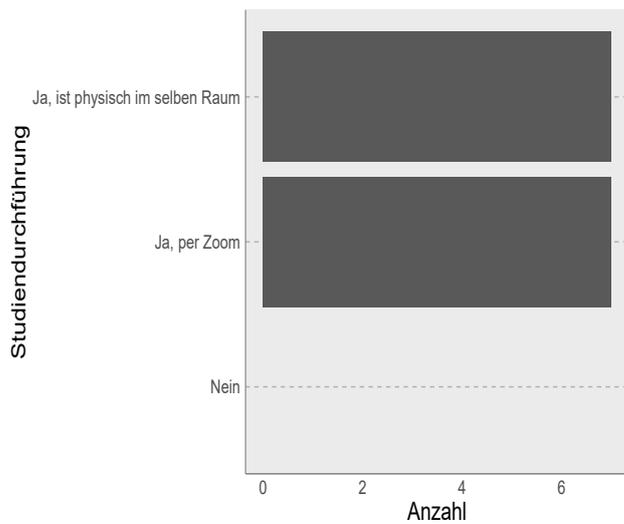


Abbildung 4.26: Verteilung der Art der Studiendurchführung aller Probanden.

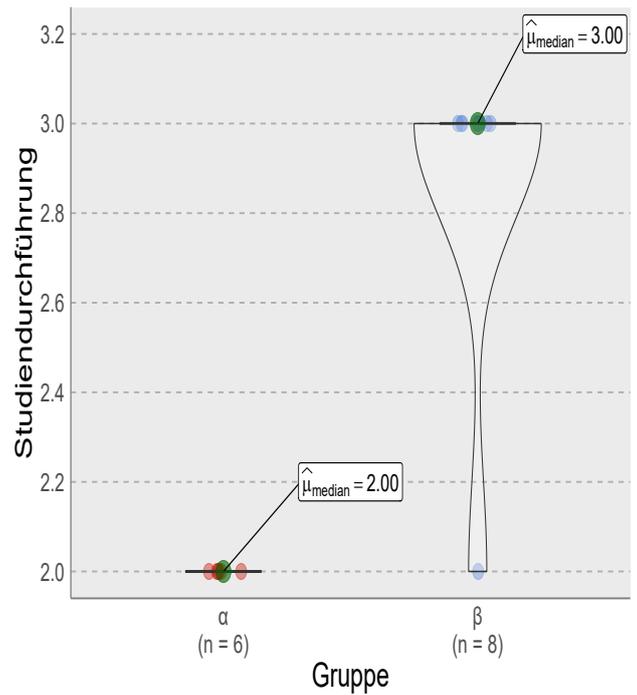


Abbildung 4.27: Verteilung der Art der Studiendurchführung beider Gruppen.

4.5.2 RQ1: Ingesamte Benutzerfreundlichkeit

In diesem Abschnitt werden die Ergebnisse für die erste Forschungsfrage ausgewertet. Zunächst wird der SUS-Score untersucht und anschließend, wie beschrieben, in seine Unterkategorien "Usability" und "Learnability" aufgeteilt und ausgewertet. Anschließend werden die zusammengefassten ASQ-Daten und die benötigte Zeit für die Bewältigung aller Aufgaben zusammengefasst ausgewertet. Die zusammengefassten ASQ-Daten liegen in einem Bereich von 7 bis 49 möglichen Punkten. Je niedriger der Wert ist, desto höher ist die Zufriedenheit der Probanden mit dem jeweiligen ASQ-Aspekt. Diese Wertebereiche entstehen, da ein ASQ zur Bewertung des jeweiligen Aspektes 1 bis 7 Punkte bereitstellt und die Probanden sieben Aufgaben bewerteten. Darüber hinaus wird die zu untersuchende Hypothese bewertet. Abschließend werden die Daten der einzelnen Aufgaben vorgestellt.

- **SUS:**

Der errechnete SUS-Score aller Probanden betrug durchschnittlich $\bar{x} \approx 82,32$ mit einer Standardabweichung von $S \approx 11,07$. Der Median war hierbei $\tilde{x} = 85$. Laut der Tabelle von Sauro (2018) deutet der Durchschnitt auf eine exzellente Benutzerfreundlichkeit hin, was einer amerikanischen Bewertung der Note "A" entspricht. Der Median würde sogar auf das bestmögliche Ergebnis mit einer Benotung von "A+" hinweisen. In der nachfolgenden [Abbildung](#)

4.28 kann diese Gesamtverteilung betrachtet werden. Dabei ist deutlich ein Ausreißer zu sehen, der die Benutzerfreundlichkeit unter 60 einstuft. Auf diesen Probanden wird im Abschnitt zur [Untersuchung der zweiten Forschungsfrage](#) näher eingegangen.



Abbildung 4.28: Auswertung der gesamten SUS-Scores.

– **SUS-Usability:**

Auch der SUS-Score für die Unterkategorie “Usability“ deutet mit einem Durchschnitt von $\bar{\mathcal{X}} \approx 83,26$, einem Median von $\tilde{\mathcal{X}} = 87,5$ und einer Standardabweichung von $S \approx 10,94$ auf eine exzellente bis bestmögliche Benutzerfreundlichkeit hin. In der [Abbildung 4.29](#) kann die Verteilung hierfür begutachtet werden. Ebenfalls fällt hier besonders der negative Ausreißer auf.

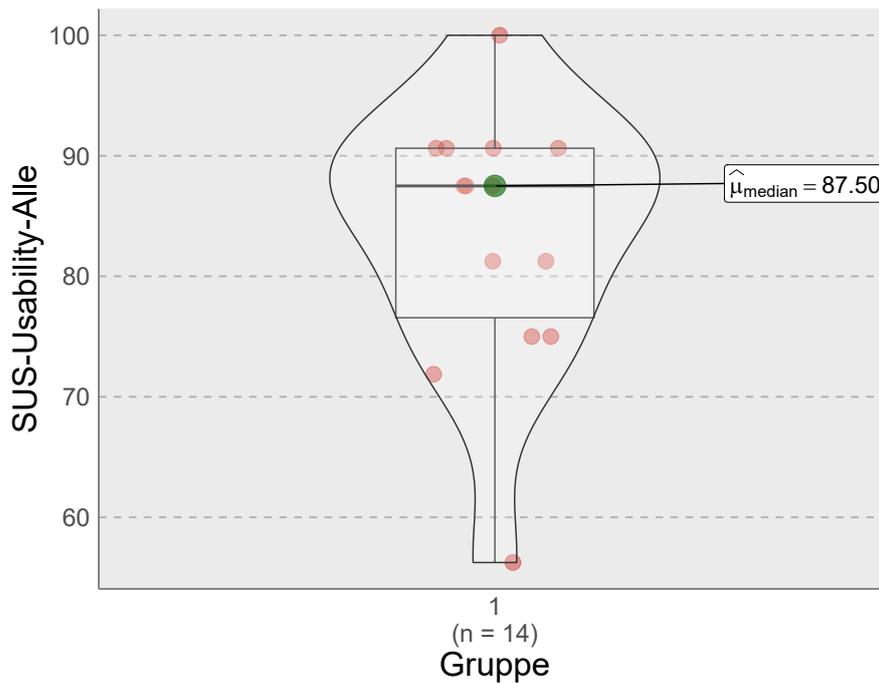


Abbildung 4.29: Auswertung der Usability-SUS-Scores.

– **SUS-Learnability:**

Die Lernfähigkeit der Integration wird mit einem Durchschnitt von $\bar{x} \approx 78,57$, einem Median von $\tilde{x} = 81,25$ und einer Standardabweichung von $s \approx 21,61$ bewertet. Der Durchschnitt deutet hierbei auf eine gute Lernfähigkeit hin, die im amerikanischen Benotungssystem ein "B+" erhalten würde. Der Median deutet allerdings auf eine exzellente Lernfähigkeit hin. Die Verteilung hierfür wird in der nachfolgenden [Abbildung 4.30](#) dargestellt. Es ist zu sehen, dass neben dem bereits bekannten Ausreißer noch drei weitere Probanden die Lernfähigkeit mit einem SUS-Score von unter 68 bewerteten. Diese drei Probanden bewerteten die Lernfähigkeit mit einem "Ok". Im anschließenden Abschnitt wird untersucht, wie sich diese Probanden auf die beiden Gruppen verteilen.



Abbildung 4.30: Auswertung der Learnability-SUS-Scores.

- ASQ-Leichtigkeit/Komplexität (Ease):**
 Die Komplexität beziehungsweise die Leichtigkeit aller Aufgaben zusammen wurde mit einem Durchschnittswert von $\bar{X} \approx 13,79$ und einem Median von $\tilde{X} = 13,5$ bewertet. Die Standardabweichung betrug dabei $S \approx 4,71$. Für die einzelnen Aufgaben entspricht dies einem Durchschnitt von $\bar{X} \approx 1,97$ Punkten. Die Ergebnisse zeigen, dass die Probanden eine sehr starke bis starke Zustimmung in Bezug der Zufriedenheit mit der Leichtigkeit angaben. Das bedeutet, dass die Probanden die Komplexität als gering empfanden.
- ASQ-Aufwand (Effort):**
 Der Aufwand aller Aufgaben wurde auf eine durchschnittliche Punktezahl von $\bar{X} \approx 14,07$ und einem Median von $\tilde{X} = 14$ Punkten bewertet. Mit einer Standardabweichung von $S \approx 4,21$ Punkten. Für die einzelnen Aufgaben entspricht dies einem Aufwand in Höhe von $\bar{X} \approx 2,01$ Punkten. Diese Ergebnisse deuten auf eine starke Zufriedenheit mit dem aufgewendeten Aufwand für die Aufgaben hin. Das bedeutet, dass die Probanden den aufgetragenen Aufwand als gering empfanden.
- ASQ-Informationszufriedenheit (Information satisfaction):**
 Die Zufriedenheit mit den angezeigten Informationen in der Visualisierung wurde mit einer durchschnittlichen Punktezahl von $\bar{X} \approx 10,86$ und einem Median von $\tilde{X} = 10$ Punkten bewertet. Die Standardabweichung betrug hierbei $S \approx 3,08$ Punkte. Für

die einzelnen Aufgaben bedeutet dies eine durchschnittliche Zufriedenheit in Höhe von $\bar{X} \approx 1,55$ Punkten. Dies repräsentiert eine sehr starke bis starke Zufriedenheit der Probanden mit den angezeigten Informationen.

- **Benötigte Zeit:**

Die benötigte Zeit für die Aufgabenbewältigung der Probanden betrug durchschnittlich $\bar{X} \approx 39,37$ Minuten. Der Median betrug $\tilde{X} \approx 37,56$ Minuten. Die Standardabweichung war hierbei $S \approx 9,39$ Minuten.

Mit den eben präsentierten Ergebnissen, insbesondere in Bezug auf den SUS-Scores, lässt sich die *Nullhypothese* \mathcal{H}_{a_0} , die besagt, dass die Integration keine gute Benutzerfreundlichkeit bietet, widerlegen. Die *Alternative Hypothese* \mathcal{H}_{a_1} kann angenommen werden, da der SUS-Score mit einem durchschnittlichen Wert von $\bar{X} \approx 82,32$ größer ist als die geforderten 71,1. Dasselbe gilt auch für den Median des SUS-Scores. Die erste Forschungsfrage kann demnach wie folgt beantwortet werden: Basierend auf den Ergebnissen der Probanden und gemäß der Tabelle von Sauro (2018) weist die Integration mindestens eine exzellente Benutzerfreundlichkeit auf.

Die folgende [Tabelle 4.1](#) enthält die Zeiten der einzelnen Aufgaben aller Probanden. Die Werte ohne ein Gleichheitszeichen sind gerundete Werte. Aufgrund der Übersichtlichkeit wurde auf ein “ \approx ”-Zeichen verzichtet. Es lässt sich bereits eine leichte Tendenz, gemessen anhand der benötigten Zeit für die Aufgaben, erkennen, dass die Probanden den Umgang zuerst erlernen müssen. Wenn beispielsweise die Aufgabenlänge im Sinne von den Aufgabenschritten der ersten Aufgabe mit denen der dritten oder sechsten Aufgabe verglichen wird, wird weniger Zeit für diese Aufgaben benötigt als für die kürzere erste Aufgabe. Dies deutet auf einen Lerneffekt in der grundlegenden Bedienung der Integration hin.

	A (Durchschnitt)	M (Median)	S (Standardabweichung)
A1	7,88	= 7,85	1,62
A2	9,15	8,69	2,15
A3	7,38	7,33	2,47
A4	2,96	2,82	0,81
A5	2,79	2,76	0,72
A6	7,75	6,78	2,54
A7	1,47	=1,45	0,36

Tabelle 4.1: Die Zeiten der einzelnen Aufgaben aller Probanden in Minuten.

Mit der nachfolgenden [Tabelle 4.2](#) werden die ASQ-Daten der einzelnen Aufgaben aller Probanden ausgewertet. Zuerst werden die Werte für

die Komplexität, dann für den Aufwand und abschließend für die Zufriedenheit mit den Informationen aufgeführt. Dabei enthalten die dazugehörigen Spalten zuerst den Durchschnitt, dann den Median und abschließend die dazugehörige Standardabweichung. Die Medianwerte und die durch ein Gleichheitszeichen gekennzeichneten Werte entsprechen exakten Werten, alle anderen Werte sind gerundet. Wie bereits bei der Zeitentabelle wurde auf das "≈"-Zeichen zur besseren Übersichtlichkeit verzichtet.

Die Auswertung der Tabelle ergibt, dass die Aufgabe 1 als schwierigste und aufwendigste Aufgabe angesehen wurde. Das liegt höchstwahrscheinlich daran, dass die Probanden zu diesem Zeitpunkt die grundlegende Bedienung noch nicht so gut beherrschten, wie im späteren Verlauf der Studie. Die nachfolgenden Aufgaben weisen höchstens eine leichte Komplexität auf und erforderten, abgesehen von der zweiten Aufgabe, einen geringen Aufwand zur Bewältigung der Aufgabe.

Aufg.	Komplexität			Aufwand			Information		
	A	M	S	A	M	S	A	M	S
A1	2,71	3	0,83	= 3	3	1,3	2,14	2	0,77
A2	2,36	2	1,01	2,57	2,5	1,22	1,86	2	0,77
A3	2,07	2	1	2,29	2	0,91	1,57	1	0,76
A4	1,43	1	0,65	1,36	1	0,5	1,21	1	0,43
A5	1,64	1,5	0,84	1,64	1,5	0,84	1,36	1	0,5
A6	2,07	2	1,07	1,93	2	0,92	= 1,5	1,5	0,52
A7	= 1,5	1	0,76	1,29	1	0,61	1,21	1	0,58

Tabelle 4.2: Die ASQ-Ergebnisse der einzelnen Aufgaben aller Probanden in Bewertungspunkten.

4.5.3 RQ2: Vergleich Informatiker und Nicht-Informatiker

Mit diesem Abschnitt wird die zweite Forschungsfrage, die dieser Arbeit zugrundeliegt, überprüft. Dazu werden die Ergebnisse der Informatiker und Nicht-Informatiker des Studiendurchlaufes miteinander verglichen. Zur Überprüfung der signifikanten Unterschiede wurde, wie auch schon im Abschnitt der [Demografische Daten](#), der Mann-Whitney-U-Test mit einem Signifikanzniveau von $p = 0,05$ verwendet. Dabei wird sowohl der Fall, dass die Informatiker bessere Ergebnisse erzielen, als auch der Fall, dass die Nicht-Informatiker bessere Ergebnisse erzielen, auf signifikante Unterschiede geprüft.

Zunächst werden die abhängigen Variablen für alle Aufgaben zusammen überprüft. Basierend auf diesen Ergebnissen wird die Forschungsfrage beantwortet. Abschließend wird auf die signifikanten Unterschiede der einzelnen Aufgaben eingegangen.

- **SUS:**

Die Informatiker bewerteten die Gesamtbenutzerfreundlichkeit durch den SUS-Score durchschnittlich mit $\bar{X} = 81,25$ und einem Median von $\tilde{X} = 85$. Die Standardabweichung betrug dabei $S \approx 14,03$ SUS-Punkte. Die Nicht-Informatiker bewerteten die Gesamtbenutzerfreundlichkeit mit durchschnittlich $\bar{X} \approx 83,13$ Punkten und mit einem Median von $\tilde{X} = 83,75$. Die Standardabweichung hierbei betrug $S \approx 9,23$ Punkte. Die Informatiker bewerten demnach die Benutzerfreundlichkeit basierend auf dem durchschnittlichen SUS-Score mit exzellent, wobei der Median auf das bestmögliche Ergebnis hinweist. Nicht-Informatiker bewerten sie als exzellent. Basierend auf der Tabelle von [Sauro \(2018\)](#).

Beide Testergebnisse der Überprüfung auf signifikante Unterschiede waren identisch und zeigten mit dem Ergebnis $U = 24; p \approx 0,53$ keine signifikanten Unterschiede. In der nachfolgenden [Abbildung 4.31](#) können die SUS-Scores der einzelnen Probanden gruppiert nach ihrer Informatik-Zugehörigkeit betrachtet werden. Wie im Abschnitt der Untersuchung der ersten Forschungsfrage beschrieben, ist hier deutlich ein negativer Ausreißer aus dem Bereich der Informatik ersichtlich. Basierend auf seinen Ergebnissen des ASQ-Fragebogens ist die Bewertung des Probanden leider nicht nachvollziehbar. Seine Zustimmungen zu den ASQ-Daten waren durchweg positiv. Der Proband hinterlies auch nur ein Kommentar, der nicht für Klarheit sorgt. Auf diesen Kommentar wird später im Abschnitt "[Kommentare der Probanden](#)" eingegangen.

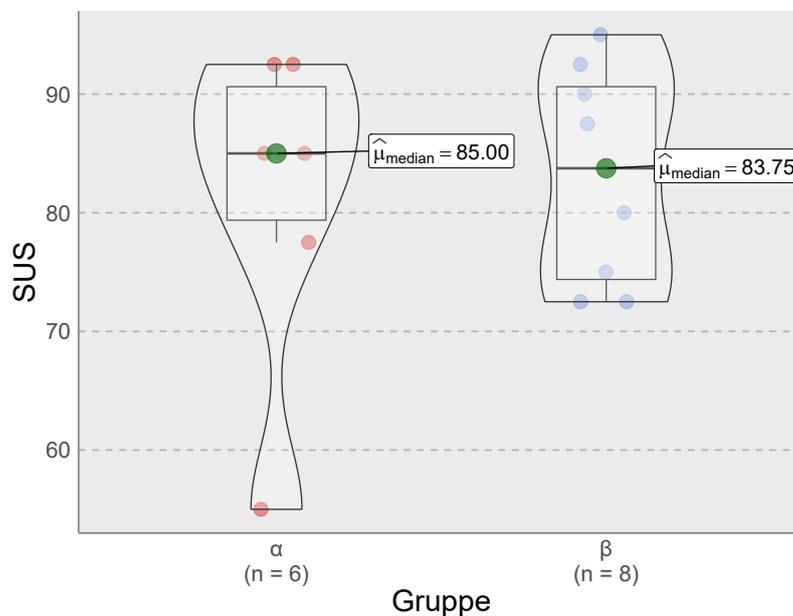


Abbildung 4.31: Auswertung der SUS-Scores der Informatiker (α) und Nicht-Informatiker (β).

– **SUS-Usability:**

Den Usability-Aspekt des SUS bewerteten die Informatiker durchschnittlich mit $\bar{X} \approx 80,21$ und mit einem Median von $\tilde{X} \approx 84,38$. Die Standardabweichung betrug dabei $S \approx 13,21$. Hingegen bewerteten Nicht-Informatiker ihn durchschnittlich mit $\bar{X} \approx 85,55$ und mit einem Median von $\tilde{X} = 87,5$. Die Standardabweichung lag bei $S \approx 9,14$. Die Bewertung der Informatiker deutet auf eine exzellente Usability hin, während die der Nicht-Informatiker auf das bestmögliche Ergebnis hinweist. Die Prüfung auf signifikante Unterschiede ergab für beide Überprüfungen keine signifikanten Unterschiede. Die Ergebnisse der Untersuchungen waren $U = 20; p \approx 0,72$ und $U = 20; p \approx 0,32$. In der nachfolgenden [Abbildung 4.32](#) können die Ergebnisse der Probanden gruppiert betrachtet werden, erneut fällt hier besonders der Informatik-Ausreißer ins Auge.

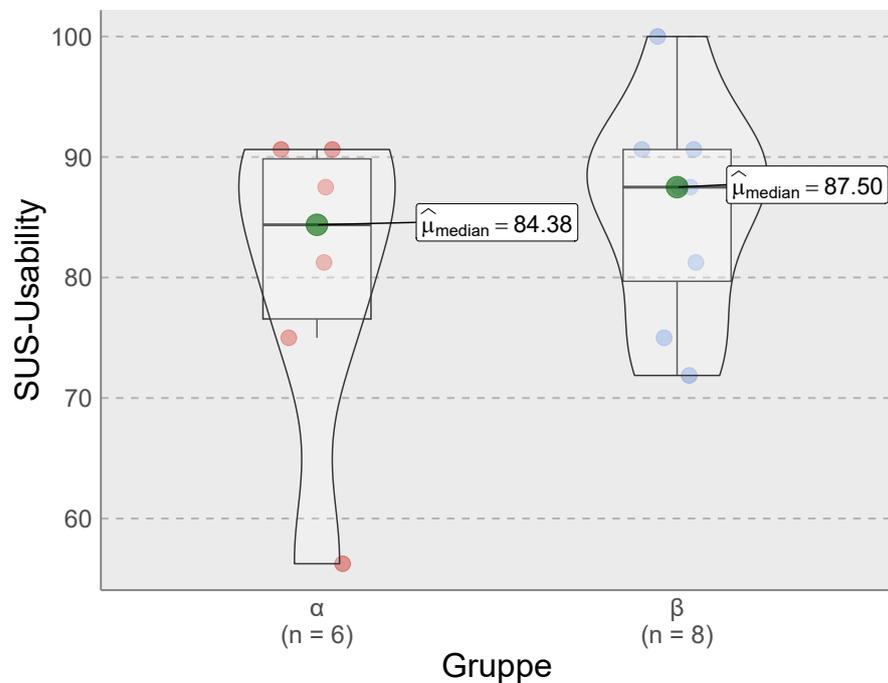


Abbildung 4.32: Auswertung der Usability-SUS-Scores der Informatiker (α) und Nicht-Informatiker (β).

– **SUS-Learnability:**

Der Learnability-SUS-Aspekt wurde von den Informatikern mit einem durchschnittlichen Punktwert von $\bar{X} \approx 85,42$ und einem Median von $\tilde{X} = 93,75$ bewertet. Die Standardabweichung betrug hier $S \approx 20,03$ Punkte. Sowohl der Durchschnitt als auch der Median weisen für die Lernfähigkeit die bestmögliche Bewertung auf. Hingegen bewerteten die Nicht-Informatiker diesen Aspekt mit durchschnittlich \bar{X}

$\approx 73,44$ Punkten und mit einem Median von $\tilde{x} = 75$ Punkten, was auf eine schlechtere Bewertung dieses Aspektes hinweist. Die Standardabweichung betrug $S \approx 22,6$ Punkte. Die Ergebnisse der Nicht-Informatiker weisen auf eine gute Lernfähigkeit hin. Es wurde kein signifikanter Unterschied durch die beiden Tests festgestellt. Die Ergebnisse hierfür waren $U = 32; p \approx 0.16$ und $U = 32; p \approx 0.87$. In der folgenden [Abbildung 4.33](#) werden die Bewertungen der Probanden dargestellt. Neben dem Informatik-Ausreißer sind auch zwei negativ Nicht-Informatiker-Ausreißer zu erkennen. Ein Wert von 50 stellt jedoch kein schlechtes Ergebnis dar, sondern ein neutrales Ergebnis. Dieses ergibt sich, wenn für die Fragen 4 und 10 der mittlere Wert und somit die neutrale Antwort gewählt wurde. Lediglich ein Proband lag unter dem Wert von 50. Dieser hatte für die zehnte Frage "Ich musste eine Menge lernen, bevor ich anfangen konnte das Drawable-System zu verwenden." vier Punkte vergeben. Es ist fraglich, ob der Proband wirklich der Ansicht war, dass er viel lernen musste, oder ob er die Frage nicht richtig gelesen hat. Bei einigen Probanden führte das nicht richtige Lesen der Aufgabenstellungen zu Problemen bei deren Bearbeitung.

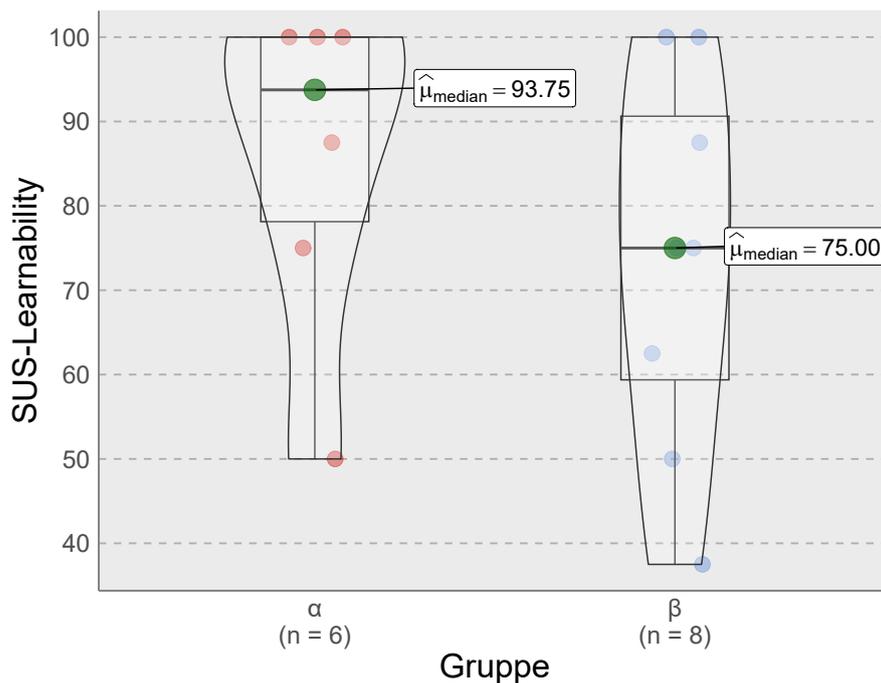


Abbildung 4.33: Auswertung der Learnability-SUS-Scores der Informatiker (α) und Nicht-Informatiker (β).

Basierend auf den SUS-Ergebnissen und der Tatsache, dass kein signifikanter Unterschied festgestellt wurde, kann die *Nullhypothese* \mathcal{H}_{b_0} nicht widerlegt werden. Somit kann angenommen werden,

dass die Benutzerfreundlichkeit von Informatikern sowie von Nicht-Informatikern ähnlich bewertet wurde. Daraus lässt sich schließen, dass Nicht-Informatiker die Benutzerfreundlichkeit genauso oder ähnlich empfinden wie Informatiker.

- **ASQ-Leichtigkeit/Komplexität (Ease):**

Informatiker bewerteten die Komplexität der Aufgaben gemessen an ihrer Leichtigkeit durchschnittlich mit $\bar{X} = 11$ Punkten und mit einem Median von $\tilde{X} = 9,5$ Punkten. Die Standardabweichung lag hierbei bei $S \approx 3,41$ Punkten. Umgerechnet auf die einzelnen Aufgaben bedeutet dies einen durchschnittlichen Punktwert von $\bar{X} \approx 1,57$ Punkte. Nicht-Informatiker bewerten die Komplexität mit einem durchschnittlichen Punktwert von $\bar{X} \approx 15,88$ Punkten und einem Median von $\tilde{X} = 16$ Punkten. Die Standardabweichung hierbei betrug $S \approx 4,61$ Punkte. Auf die einzelnen Aufgaben umgerechnet bedeutet dies einen durchschnittlichen Punktwert von $\bar{X} \approx 2,27$ Punkte. Wie beschrieben, liegt der Wertebereich des ASQs zwischen 7 und 49 Punkten. Daher ergibt sich für die Informatiker eine sehr starke bis starke Zustimmung für die Zufriedenheit mit der Komplexität der Aufgaben. Für die Nicht-Informatiker liegt hier eine starke bis mäßige Zufriedenheit vor. Es wurde ein signifikanter Unterschied für die empfundene Komplexität mit dem Testergebnis ($U = 8,5; p \approx 0.026$) festgestellt. Das deutet darauf hin, dass Informatiker die gestellten Aufgaben als kognitiv leichter empfanden als die Nicht-Informatiker. In der nachfolgenden [Abbildung 4.34](#) können die Ergebnisse, die zu dem signifikanten Unterschied führten, betrachtet werden. Hierbei fällt auf, dass es vereinzelt Nicht-Informatiker gab, die die Komplexität ähnlich wie die Informatiker bewerteten. Es sind allerdings auch zwei negativ Ausreißer bei den Nicht-Informatikern zu erkennen, welche die Komplexität über 20 Punkte einschätzen, was für eine mäßige Zustimmung bezüglich der Komplexität steht. Bezogen auf den festgestellten signifikanten Unterschied kann die *Nullhypothese* \mathcal{H}_{d_0} widerlegt werden und die *Alternative Hypothese* \mathcal{H}_{d_1} angenommen werden, da die empfundene Komplexität der Informatiker geringer ist als die der Nicht-Informatiker.

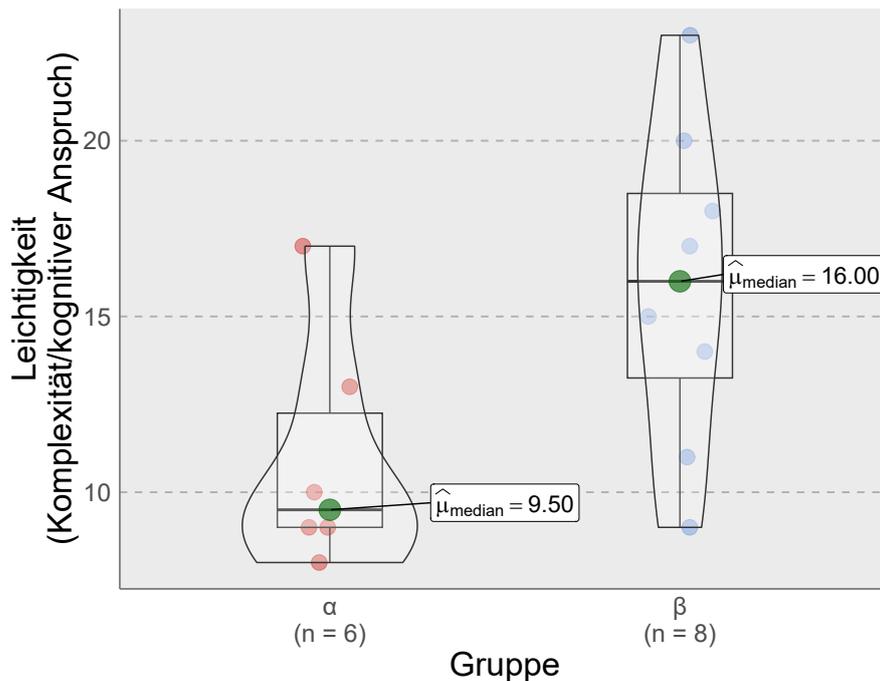


Abbildung 4.34: Auswertung der ASQ-Leichtigkeit/Komplexität der Informatiker (α) und Nicht-Informatiker (β).

- **ASQ-Aufwand (Effort):**

Den aufgewendeten Aufwand zur Bewältigung der Aufgaben bewerteten die Informatiker durchschnittlich mit einem Punktwert von $\bar{X} \approx 11,33$ und einen Median von $\tilde{X} = 12,5$. Die Standardabweichung betrug $S \approx 3,5$ Punkte. Umgerechnet auf den durchschnittlichen Aufwand pro Aufgabe ergibt sich ein Wert von $\bar{X} \approx 1,62$ Punkte. Hingegen bewerteten Nicht-Informatiker den Aufwand durchschnittlich mit $\bar{X} \approx 16,13$ Punkten, wobei der Median $\tilde{X} = 17,5$ Punkte ergab. Die Standardabweichung lag hierbei bei $S \approx 3,6$ Punkte. Auf die einzelnen Aufgaben umgerechnet ergibt dies einen durchschnittlichen Punktwert von $\bar{X} \approx 2,3$ Punkten. Wie auch schon bei der Komplexität, deuten diese Ergebnisse darauf hin, dass die Informatiker eine sehr starke bis starke Zustimmung hinsichtlich der Zufriedenheit mit dem aufgewendeten Aufwand aufweisen, während die Nicht-Informatiker eine starke bis mäßige Zustimmung aufweisen. Zwischen den beiden Gruppen wurde ein signifikanter Unterschied bezüglich des Aufwandes mit dem Untersuchungsergebnis $U = 7; p \approx 0,02$ festgestellt. Basierend auf diesem signifikanten Unterschied kann die *Nullhypothese* \mathcal{H}_{e_0} verworfen und die *Alternative Hypothese* \mathcal{H}_{e_1} angenommen werden, da der empfundene Aufwand bei den Informatikern signifikant geringer ist als bei den Nicht-Informatikern. In der [Abbildung 4.35](#) werden die Bewertungen, die zu diesem signifikanten Unterschied führten, dargestellt. Hierbei ist deutlich

zu erkennen, dass es einen Nicht-Informatiker-Probanden gab, der den Aufwand deutlich geringer einschätzte als der Großteil der Informatiker. Zwei weitere Nicht-Informatiker-Probanden schätzten den Aufwand ähnlich ein wie die Informatiker.

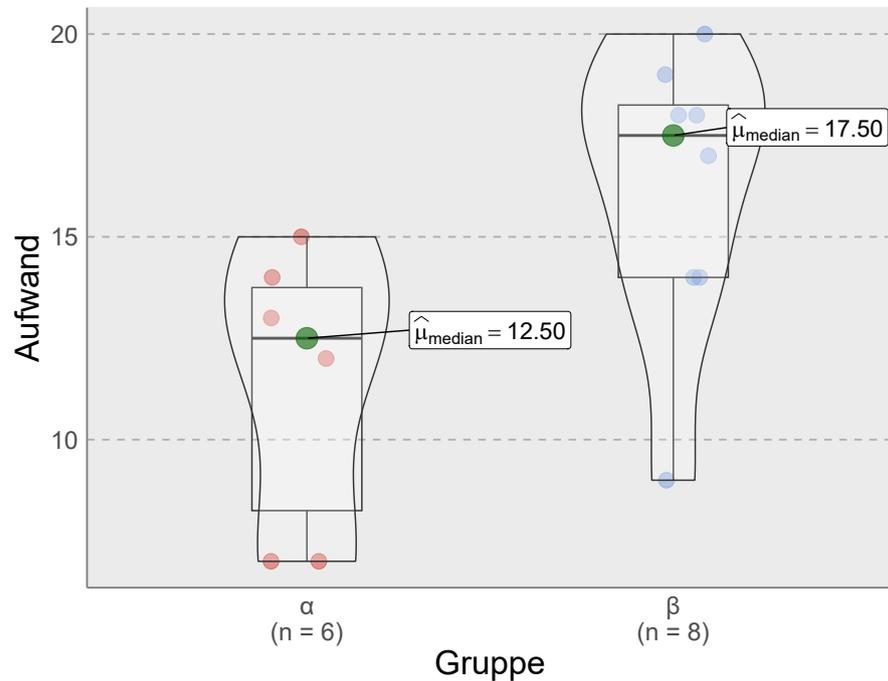


Abbildung 4.35: Auswertung des aufgebrauchten ASQ-Aufwands der Informatiker (α) und Nicht-Informatiker (β).

- ASQ-Informationszufriedenheit (Information satisfaction):**
 Die Zufriedenheit mit den angezeigten Information bewerteten Informatiker durchschnittlich mit einer Punktezahl von $\bar{X} = 10,5$ Punkten, der Median betrug hierbei $\tilde{X} = 8,5$ Punkte. Die Standardabweichung lag bei $S \approx 4,23$ Punkten. Auf die einzelnen Aufgaben umgerechnet entspricht dies einem durchschnittlichen Punktewert von $\bar{X} = 1,5$ Punkten. Dies deutet auf eine sehr starke bis starke Zustimmung mit der Zufriedenheit hin. Nicht-Informatiker bewerten die Zufriedenheit mit einem durchschnittlichen Wert von $\bar{X} \approx 11,23$ Punkten und einem Median von $\tilde{X} = 11$ Punkten. Die Standardabweichung betrug hierbei $S \approx 2,17$ Punkte. Umgerechnet auf die einzelnen Aufgaben betrug die Zufriedenheit demnach $\bar{X} \approx 1,59$ Punkte und entspricht ebenfalls einer sehr starken bis starken Zufriedenheit. Mittels der Signifikanzprüfung wurden keine signifikanten Unterschiede festgestellt ($U = 16; p \approx 0.16$ und $U = 32; p \approx 0.87$). Aufgrund der Tatsache, dass kein signifikanter Unterschied zwischen den beiden Gruppen festgestellt wurde, kann die Nullhypothese \mathcal{H}_{f_0} nicht verworfen werden und muss angenommen werden. Die [Abbildung 4.36](#) stellt die Antworten bezüglich der Informationszufriedenheit der

Probanden dar. Hierbei fällt ein negativer Informatik-Ausreißer ins Auge. Abgesehen von diesem und einen weiteren bewerten die Informatiker die Zufriedenheit größtenteils deutlich besser als die Nicht-Informatiker. Lediglich ein weiterer Informatiker bewertet die Zufriedenheit leicht schlechter als die bestbewerteten Antworten der Nicht-Informatiker.

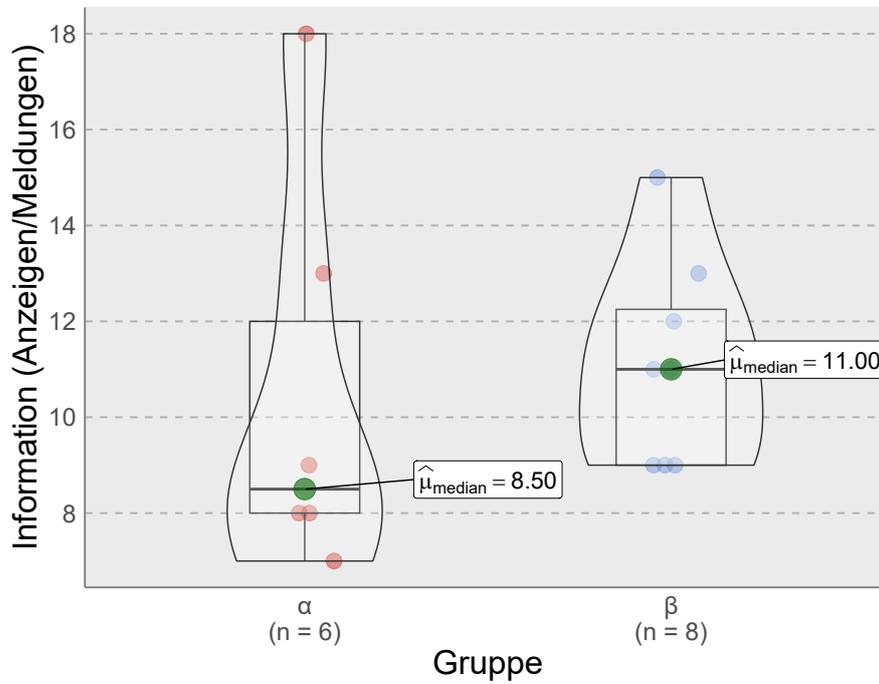


Abbildung 4.36: Auswertung der ASQ-Informationszufriedenheit der Informatiker (α) und Nicht-Informatiker (β).

- **Benötigte Zeit:**

Die Informatiker benötigten durchschnittlich $\bar{x} \approx 33,74$ Minuten zur Bewältigung der Aufgaben. Der Median betrug hierbei $\tilde{x} \approx 35,37$ Minuten und die Standardabweichung $s \approx 6,04$ Minuten. Nicht-Informatiker benötigten durchschnittlich $\bar{x} = 43,6$ Minuten. Der Median lag hierbei bei $\tilde{x} \approx 41,94$ Minuten und die Standardabweichung bei $s \approx 9,49$ Minuten. Daraus lässt sich errechnen, dass Informatiker die Aufgaben im Durchschnitt um 22,61% schneller absolvieren konnten als die Nicht-Informatiker. Der Mann-Whitney-U-Test stellte einen fast signifikanten Unterschied fest ($U = 11; p = 0,05395$). Aufgrund der Tatsache, dass kein eindeutiger signifikanter Unterschied besteht, kann die Nullhypothese \mathcal{H}_{c_0} nicht widerlegt werden und muss daher angenommen werden. In der [Abbildung 4.37](#) können die benötigten Zeiten zur Aufgabenbewältigung gruppiert betrachtet werden. Hierbei fällt auf, dass vier der Nicht-Informatiker ähnliche Zeiten wie der Großteil der Informatiker aufwiesen. Die restlichen vier Nicht-Informatiker hingegen benötigten deutlich mehr Zeit.

Von diesen vier Probanden hatten lediglich zwei eine geringe Erfahrung mit Videospiele. Die anderen beiden unterschieden sich nicht grundlegend in ihren Erfahrungen im Vergleich zu den anderen Nicht-Informatikern.

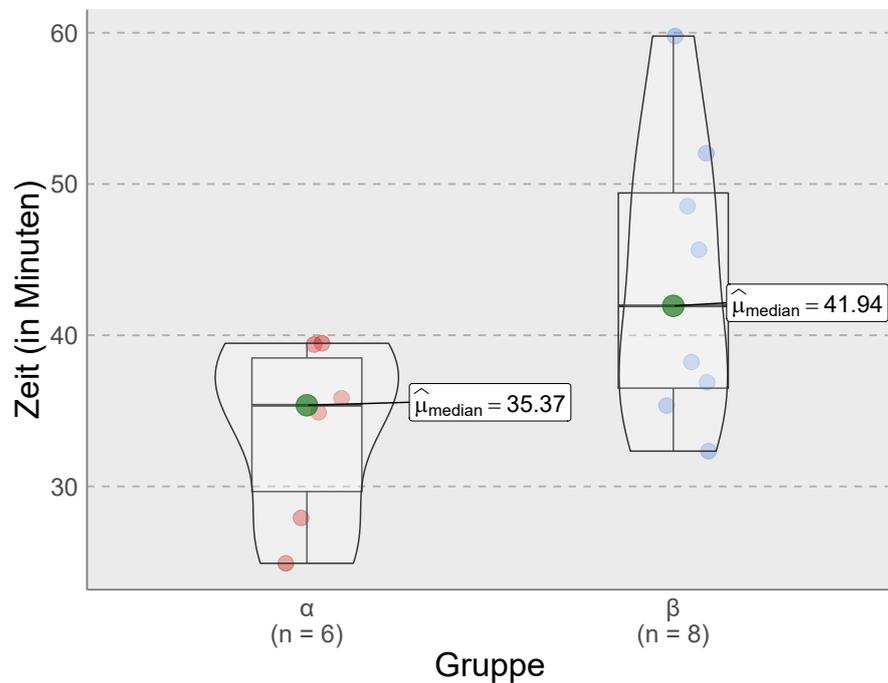


Abbildung 4.37: Auswertung der benötigten Zeit der Informatiker (α) und Nicht-Informatiker (β).

Basierend auf den vorliegenden Ergebnissen kann die zweite Forschungsfrage vorerst wie folgt beantwortet werden: Bei der ersten Verwendung der Integration zeigten die Informatiker im Vergleich zu den Nicht-Informatikern einen signifikanten Unterschied in der wahrgenommenen Komplexität und dem aufgewendeten Aufwand. Dies deutet darauf hin, dass Informatiker die Aufgaben als leichter empfanden und weniger Aufwand aufbringen müssen, um sie zu bewältigen. Darüber hinaus konnten die Informatiker die Aufgaben im Durchschnitt um 22,61% schneller absolvieren als die Nicht-Informatiker.

Im Folgenden werden die signifikanten Unterschiede der einzelnen Aufgaben erläutert. Für die einzelnen Aufgaben können die Diagramme, die den Unterschied beider Gruppen darstellen, mittels des zuvor erwähnten R-Skriptes erzeugt und anschließend betrachtet werden. Zusätzlich befinden sich diese Diagramme ebenfalls auf dem mitgelieferten USB-Stick. Diese Diagramme werden jedoch aufgrund ihrer hohen Anzahl in dieser Arbeit nicht eingebunden. Für die Untersuchung der Forschungsfragen war der Gesamtunterschied zwischen Informatikern und Nicht-Informatikern aller Aufgaben relevanter als jede einzelne Aufgabe. Zusätzlich würden diese Diagramme zu vie-

le Seiten in Anspruch nehmen. Der Mehrwert für diese Einbindung ist außerdem nicht gegeben, da die relevanten Informationen für die einzelnen Aufgaben aus den folgenden Tabellen 4.3 - 4.9 entnommen werden können. In Zahlen ausgedrückt, wären es für die benötigte Zeit sieben zusätzliche Diagramme. Jeweils ein Violin-Plot-Diagramm pro Aufgabe, welches den Zeitvergleich der beiden Gruppen darstellt. Für die ASQ-Daten wären dies 21 zusätzliche Diagramme, für jede Aufgabe ein Violin-Plot-Diagramm für jeden der drei ASQ-Aspekte.

In den eben genannten Tabellen sind die signifikanten Unterschiede grün dargestellt. Ein signifikanter Unterschied wurde in der Aufgabe 2 in Bezug auf die benötigte Zeit ($U = 8; p \approx 0,02$) sowie den Aufwand ($U = 9,5; p \approx 0,03$) zwischen beiden Gruppen festgestellt. In der Aufgabe 3 wurde ein signifikanter Unterschied in Bezug auf die Komplexität ($U = 10; p \approx 0,03$) und den Aufwand ($U = 5; p \approx 0,006$) zwischen den Gruppen gefunden. Ansonsten wurden keine weiteren signifikanten Unterschiede zwischen den beiden Gruppen festgestellt. Es zeigt sich jedoch, dass die Informatiker in jeder Aufgabe weniger Zeit benötigten als die Nicht-Informatiker. Mit der Ausnahme weniger Fälle (vier) zeigten die Informatiker eine höhere Zufriedenheit mit den ASQ-Daten als die Nicht-Informatiker.

Die erläuterten signifikanten Unterschiede für die Gesamtbetrachtung aller Aufgaben ergeben sich daraus, dass die Werte nicht gerundet wurden und die kleinen Unterschiede in den einzelnen Aufgaben sich zu einem signifikanten Unterschied summieren.

Data	Informatiker			Nicht-Informatiker			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	7,28	6,95	2,06	8,33	8,13	1,14	$U = 18; p \approx 0,25$ $U = 30; p \approx 0,79$
Komplexität	2,17	2	0,75	3,13	3	0,64	$U = 32; p \approx 0,88$ $U = 16; p \approx 0,15$
Aufwand	2,33	2	1,5	3,5	3,5	0,93	$U = 31; p \approx 0,84$ $U = 17; p \approx 0,2$
Information	2,17	2,5	0,98	2,13	2	0,64	$U = 14,5; p \approx 0,11$ $U = 33,5; p \approx 0,92$

Tabelle 4.3: Die Daten des Vergleiches der Aufgabe 1 zwischen Informatiker und Nicht-Informatiker.

Data	Informatiker			Nicht-Informatiker			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	7,75	7,68	1,24	10,2	10,4	2,13	$U = 8; p \approx 0,02$
Komplexität	1,67	1,5	0,82	2,86	3	0,83	$U = 30,5; p \approx 0,83$ $U = 17,5; p \approx 0,21$
Aufwand	1,67	1,5	0,82	3,25	3	1,04	$U = 9,5; p \approx 0,03$
Information	1,5	1	0,84	2,13	2	0,64	$U = 33,5; p \approx 0,92$ $U = 14,5; p \approx 0,11$

Tabelle 4.4: Die Daten des Vergleiches der Aufgabe 2 zwischen Informatiker und Nicht-Informatiker.

Data	Informatiker			Nicht-Informatiker			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	6,06	6,03	1,66	8,37	7,63	2,59	$U = 35; p \approx 0,93$ $U = 13; p \approx 0,09$
Komplexität	1,67	1,5	0,82	2,38	2,5	1,06	$U = 10; p \approx 0,03$
Aufwand	1,83	2	0,75	2,3	3	0,92	$U = 5; p \approx 0,006$
Information	1,33	1	0,82	1,75	2	0,71	$U = 33; p \approx 0,92$ $U = 15; p \approx 0,11$

Tabelle 4.5: Die Daten des Vergleiches der Aufgabe 3 zwischen Informatiker und Nicht-Informatiker.

Data	Informatiker			Nicht-Informatiker			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	2,67	2,48	0,86	3,17	3,08	0,76	$U = 32; p \approx 0,86$ $U = 16; p \approx 0,17$
Komplexität	1,17	1	0,41	1,63	1,5	0,74	$U = 31; p \approx 0,87$ $U = 17; p \approx 0,16$
Aufwand	1,17	1	0,41	1,5	1,5	0,53	$U = 30; p \approx 0,84$ $U = 18; p \approx 0,2$
Information	1,17	1	0,41	1,25	1	0,46	$U = 22; p \approx 0,39$ $U = 26; p \approx 0,67$

Tabelle 4.6: Die Daten des Vergleiches der Aufgabe 4 zwischen Informatiker und Nicht-Informatiker.

Data	Informatiker			Nicht-Informatiker			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	2,51	2,51	0,69	2,99	3,21	0,71	$U = 28; p \approx 0,71$ $U = 20; p \approx 0,33$
Komplexität	1,17	1	0,41	2	2	0,93	$U = 32,5; p \approx 0,9$ $U = 15,5; p \approx 0,12$
Aufwand	1,33	1	0,52	1,88	2	0,99	$U = 32,5; p \approx 0,9$ $U = 15,5; p \approx 0,12$
Information	1,33	1	0,52	1,38	1	0,52	$U = 30; p \approx 0,84$ $U = 18; p \approx 0,2$

Tabelle 4.7: Die Daten des Vergleiches der Aufgabe 5 zwischen Informatiker und Nicht-Informatiker.

Data	Informatiker			Nicht-Informatiker			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	6,24	6,48	0,95	8,89	8,11	2,82	$U = 27; p \approx 0,67$ $U = 21; p \approx 0,38$
Komplexität	1,67	1,5	0,82	2,38	2	1,19	$U = 33; p \approx 0,9$ $U = 15; p \approx 0,12$
Aufwand	1,5	1	0,84	2,25	2,5	0,89	$U = 34,5; p \approx 0,94$ $U = 13,5; p \approx 0,08$
Information	1,5	1,5	0,55	1,5	1,5	0,53	$U = 31; p \approx 0,87$ $U = 17; p \approx 0,17$

Tabelle 4.8: Die Daten des Vergleiches der Aufgabe 6 zwischen Informatiker und Nicht-Informatiker.

Data	Informatiker			Nicht-Informatiker			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	1,22	1,18	0,2	1,65	1,64	0,35	$U = 34; p \approx 0,91$ $U = 14; p \approx 0,11$
Komplexität	1,5	1	0,84	1,5	1	0,76	$U = 23,5; p = 0,5$ $U = 24,5; p \approx 0,56$
Aufwand	1,5	1	0,84	1,13	1	0,35	$U = 21,5; p \approx 0,36$ $U = 26,5; p \approx 0,71$
Information	1,5	1	0,84	1	1	0	$U = 18; p \approx 0,12$ $U = 30; p \approx 0,92$

Tabelle 4.9: Die Daten des Vergleiches der Aufgabe 7 zwischen Informatiker und Nicht-Informatiker.

4.5.4 RQ3: Zweiter Studiendurchlauf

Durch diesen Abschnitt wird die dritte Forschungsfrage überprüft. Hierzu wurden die Daten des ersten und des zweiten Durchlaufes der sieben Probanden miteinander verglichen. Zur Überprüfung der signifikanten Unterschiede konnte der *Wilcoxon-Vorzeichen-Rang-Test* mit einem Signifikanzniveau von $p = 0,05$ eingesetzt werden, da es sich in beiden Durchläufen um dieselben Probanden handelte. Es wurde

Wilcoxon-Vorzeichen-Rang-Test: Dieser Test prüft, ob zwei abhängige Stichproben signifikant unterschiedlich sind. Die Daten müssen dabei nicht normalverteilt sein. Damit der Test angewendet werden kann, müssen die Daten paarweise vorliegen (DATAtab_Team, 2024c).

untersucht, ob die Daten des zweiten Durchlaufes signifikant besser waren als die des ersten Durchlaufes.

Zunächst werden die abhängigen Variablen gesamt betrachtet überprüft. Basierend auf diesen Ergebnissen wird die dritte Forschungsfrage beantwortet. Abschließend wird auf die Zeitersparnisse und die signifikanten Unterschiede der einzelnen Aufgaben eingegangen. Die in diesem Abschnitt bereitgestellten Diagramme verwenden den Begriff "First" für den ersten Durchlauf und "Second" für den zweiten.

- **SUS:**

Im ersten Durchlauf bewerteten die sieben Probanden den SUS mit durchschnittlich $\bar{X} \approx 81,79$ Punkten, wobei der Median $\tilde{X} = 80$ Punkte betrug. Die Standardabweichung lag bei $S \approx 9,1$ Punkten. Im zweiten Durchlauf verbesserte sich die Punktzahl auf durchschnittlich $\bar{X} \approx 87,86$ Punkte und der Median stieg auf $\tilde{X} = 90$ Punkte. Die Standardabweichung betrug hierbei $S \approx 7,7$ Punkte. Folglich ist der SUS-Score im zweiten Durchlauf durchschnittlich um 6,07 Punkte gestiegen, was einen prozentualen Anstieg von 7,42% darstellt. Durchschnittlich bewerteten die Probanden die Gesamtbenutzerfreundlichkeit im ersten Durchlauf gemäß [Sauro \(2018\)](#) als exzellent und im zweiten als das bestmögliche Ergebnis. Die Signifikanzprüfung stellte mit $V = 2; p \approx 0,025$ einen signifikanten Unterschied zwischen der Bewertung des ersten und zweiten Durchlaufes fest. Aufgrund dieses signifikanten Unterschiedes kann die *Nullhypothese* \mathcal{H}_{g_0} verworfen und die *Alternative Hypothese* \mathcal{H}_{g_1} angenommen werden, da der SUS des zweiten Durchlaufes signifikant besser ist als der des ersten. In der folgenden [Abbildung 4.38](#) kann dieser signifikante Unterschied veranschaulicht werden. Dabei fällt die Bewertung eines Probanden im zweiten Durchlauf besonders ins Auge, der die Benutzerfreundlichkeit mit 75 Punkten einschätzt. Leider gab dieser Proband keine Kommentare in den bereitgestellten Feldern ab und äußerte auch keine im Gespräch.

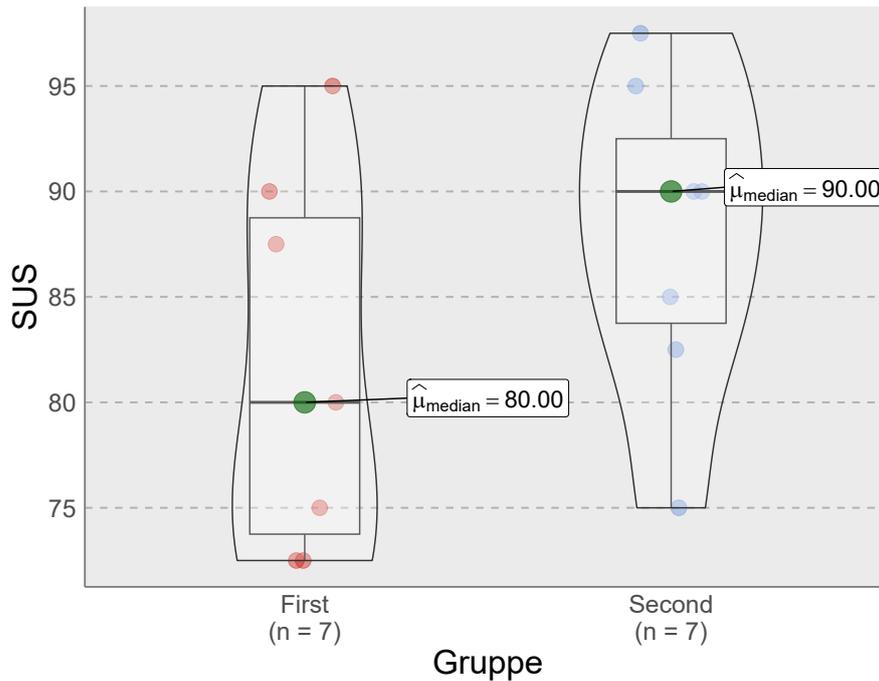


Abbildung 4.38: Auswertung der SUS-Scores des ersten und zweiten Durchlaufes (First / Second).

– **SUS-Usability:**

Der Usability-Aspekt des SUS wurde im ersten Durchlauf mit durchschnittlich $\bar{X} \approx 84,82$ Punkten und einem Median von $\tilde{X} = 87,5$ Punkten bewertet. Die Standardabweichung lag hierbei bei $S \approx 9,62$ Punkten. Im zweiten Durchlauf verbesserten sich diese Werte auf durchschnittlich $\bar{X} \approx 89,73$ Punkte, wobei der Median $\tilde{X} \approx 90,63$ Punkte betrug und die Standardabweichung $S \approx 5,62$ Punkte. Durchschnittlich stieg die Punktzahl folglich um 4,91 Punkte, was einen durchschnittlichen prozentualen Anstieg von 5,79% bedeutet. Gemäß Sauro (2018) stellt die durchschnittliche Bewertung des ersten Durchlaufes eine exzellente Benutzerfreundlichkeit dar, während der zweite Durchlauf auf das bestmögliche Ergebnis deutet. Es wurde hierbei kein signifikanter Unterschied festgestellt ($V = 5; p \approx 0,07$). In der nachfolgenden [Abbildung 4.39](#) können die Ergebnisse der Bewertungen für beide Durchläufe betrachtet werden. Hierbei fällt auf, dass die Probanden, die im ersten Durchlauf die Benutzerfreundlichkeit zuerst geringer einschätzen, im zweiten Durchlauf über 80 Punkte liegen und somit in einem exzellenten Bereich liegen. Zu erkennen ist auch ein Proband, der im ersten Durchlauf die Benutzerfreundlichkeit besser bewertete als im zweiten Durchlauf. Leider wurden auch von diesem Probanden keine Kommentare bereitgestellt, um die Ursache hierfür nachvollziehen zu können.

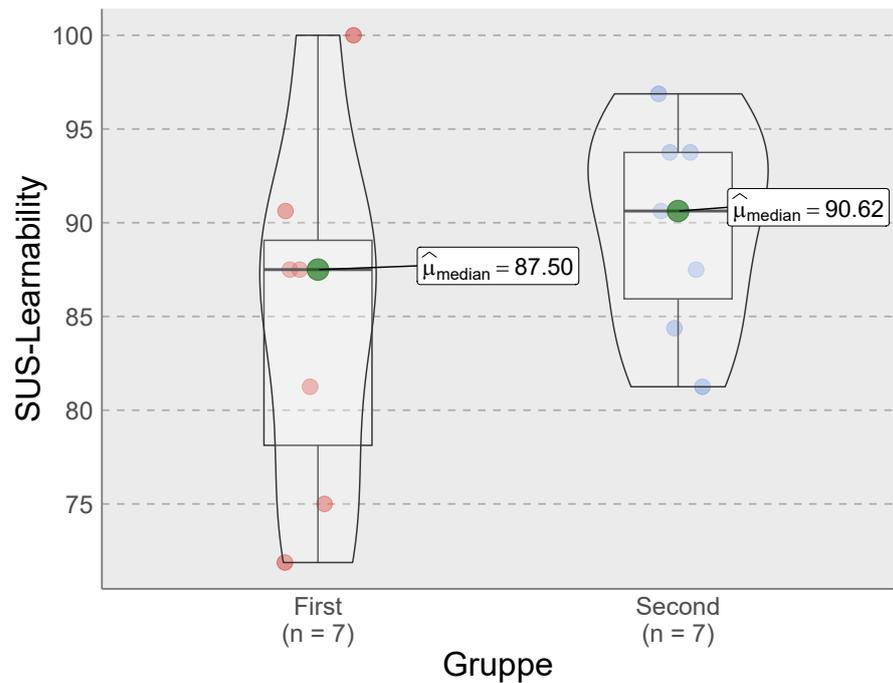


Abbildung 4.39: Auswertung der Usability-SUS-Scores des ersten und zweiten Durchlaufes (First / Second).

– **SUS-Learnability:**

Der Learnability-Aspekt des SUS wurde im ersten Durchlauf durchschnittlich mit $\bar{X} \approx 69,64$ Punkten bewertet, wobei der Median $\tilde{X} = 75$ Punkte betrug. Die Standardabweichung betrug $S \approx 21,48$ Punkte. Der zweite Durchlauf zeigte eine Verbesserung der Werte auf durchschnittlich $\bar{X} \approx 80,36$ Punkte, wobei der Median erneut $\tilde{X} = 75$ Punkte betrug, mit einer Standardabweichung von $S \approx 17,47$ Punkten. Bezogen auf den Median entstand hier kein Anstieg, jedoch stieg die Punktzahl durchschnittlich um 10,72 Punkte, was einen prozentualen Anstieg von 15,39% bedeutet. Gemäß Sauro (2018) zeigen die Ergebnisse des ersten Durchlaufes eine mäßige bis gute Lernfähigkeit und die des zweiten Durchlaufes eine gute bis exzellente Lernfähigkeit. Die Signifikanzprüfung stellte hierbei keinen signifikanten Unterschied fest ($V = 4,5; p \approx 0,12$). In der [Abbildung 4.40](#) werden die Bewertungen der Probanden veranschaulicht. Hierbei fällt auf, dass die Probanden die Lernfähigkeit im zweiten Durchlauf besser bewerteten.

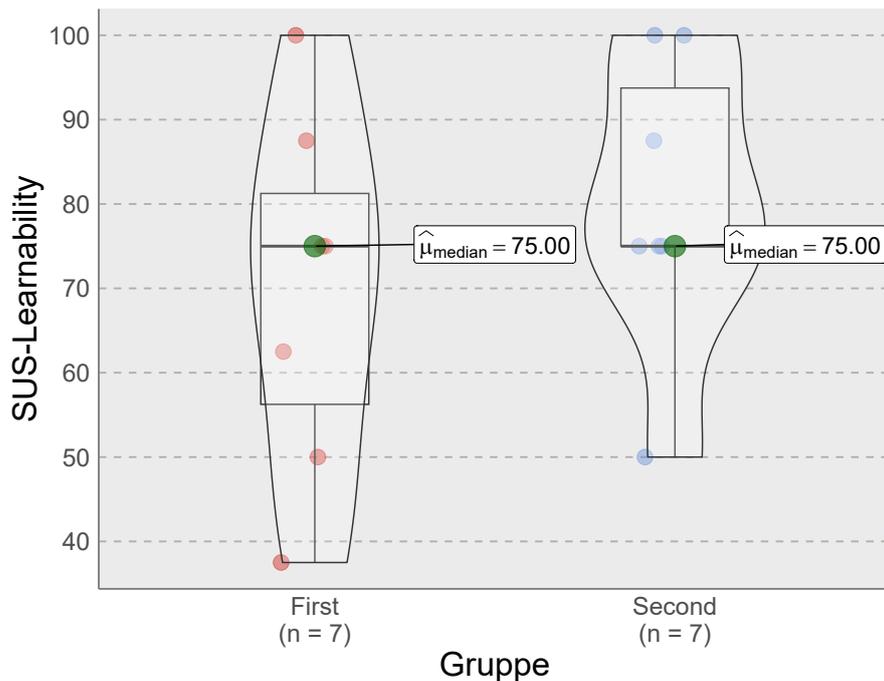


Abbildung 4.40: Auswertung der Learnability-SUS-Scores des ersten und zweiten Durchlaufes (First / Second).

- **ASQ-Leichtigkeit/Komplexität (Ease):**

Die Komplexität der Aufgaben bewerteten die Probanden im ersten Durchlauf durchschnittlich mit $\bar{X} \approx 16,86$ Punkten, wobei der Median $\tilde{X} = 17$ Punkte betrug. Die Standardabweichung war hierbei $S \approx 3,98$ Punkte. Umgerechnet auf die einzelnen Aufgaben bedeutet dies eine durchschnittliche Punktzahl von $\bar{X} \approx 2,41$ Punkten. Im zweiten Durchlauf sanken die Werte auf durchschnittlich $\bar{X} = 12$ Punkte, wobei der Median ebenfalls $\tilde{X} = 12$ Punkte betrug und die Standardabweichung $S = 3$ Punkte. Auf die einzelnen Aufgaben bedeutet dies eine durchschnittliche Punktzahl von $\bar{X} \approx 1,71$ Punkten. Das zeigt, dass die Zufriedenheit mit der Komplexität von einer starken bis mäßigen Zustimmung zu einer sehr starken bis starken Zustimmung anstieg. Durchschnittlich stieg die Zufriedenheit um 28,83%. Hierzu wurde ein signifikanter Unterschied zwischen dem ersten und zweiten Durchlauf festgestellt ($V = 0; p \approx 0,01$). Aufgrund dieses signifikanten Unterschiedes kann die *Nullhypothese* \mathcal{H}_{i_0} verworfen und die *Alternative Hypothese* \mathcal{H}_{i_1} angenommen werden, da die Zufriedenheit in Bezug auf die empfundene Komplexität im zweiten Durchlauf signifikant höher ist als im ersten Durchlauf. Die [Abbildung 4.41](#) veranschaulicht diesen signifikanten Unterschied, wobei die bessere Beurteilung der Probanden deutlich sichtbar wird.

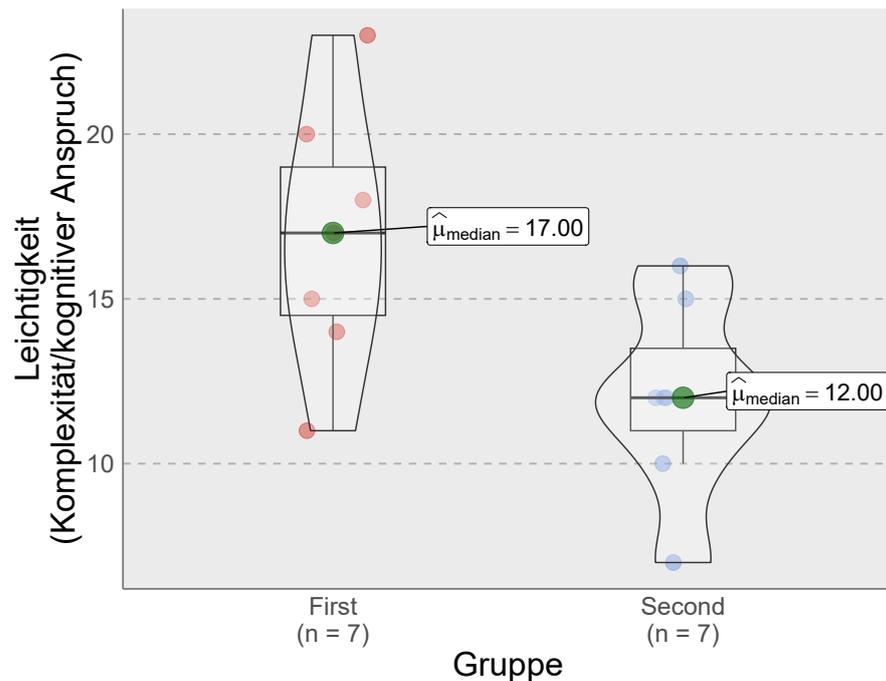


Abbildung 4.41: Auswertung der ASQ-Leichtigkeit/Komplexität des ersten und zweiten Durchlaufes (First / Second).

- **ASQ-Aufwand (Effort):**

Die Zufriedenheit mit dem aufgewendeten Aufwand zur Aufgabenbewältigung bewerteten die Probanden im ersten Durchlauf durchschnittlich mit $\bar{X} \approx 17,14$ Punkten, wobei der Median $\tilde{X} = 18$ Punkte und die Standardabweichung $S \approx 2,34$ Punkte betrug. Umgerechnet auf die einzelnen Aufgaben bedeutet dies eine durchschnittliche Bewertung von $\bar{X} \approx 2,44$ Punkten. Im zweiten Durchlauf wurde die Zufriedenheit durchschnittlich mit $\bar{X} \approx 11,29$ Punkten und einem Median von $\tilde{X} = 12$ Punkten bewertet. Die Standardabweichung betrug hierbei $S \approx 2,93$ Punkte. Durchschnittlich ergibt das für die einzelnen Aufgaben einen Wert von $\bar{X} \approx 1,61$ Punkten. Somit stieg auch hier die Zufriedenheit von einer starken bis mäßigen Zufriedenheit auf eine sehr starke bis starke Zufriedenheit an. Durchschnittlich betrug die Steigerung der Zufriedenheit 34,13%. Durch die Überprüfung der Signifikanz der Ergebnisse wurde ein signifikanter Unterschied zwischen beiden Durchläufen festgestellt ($V = 0; p \approx 0,00995$). Aufgrund dieses signifikanten Unterschieds kann die *Nullhypothese* \mathcal{H}_{j_0} widerlegt und die *Alternative Hypothese* \mathcal{H}_{j_1} angenommen werden. In der [Abbildung 4.42](#) können diese deutlichen Verbesserungen der Empfindung des aufgebrauchten Aufwandes veranschaulicht werden.

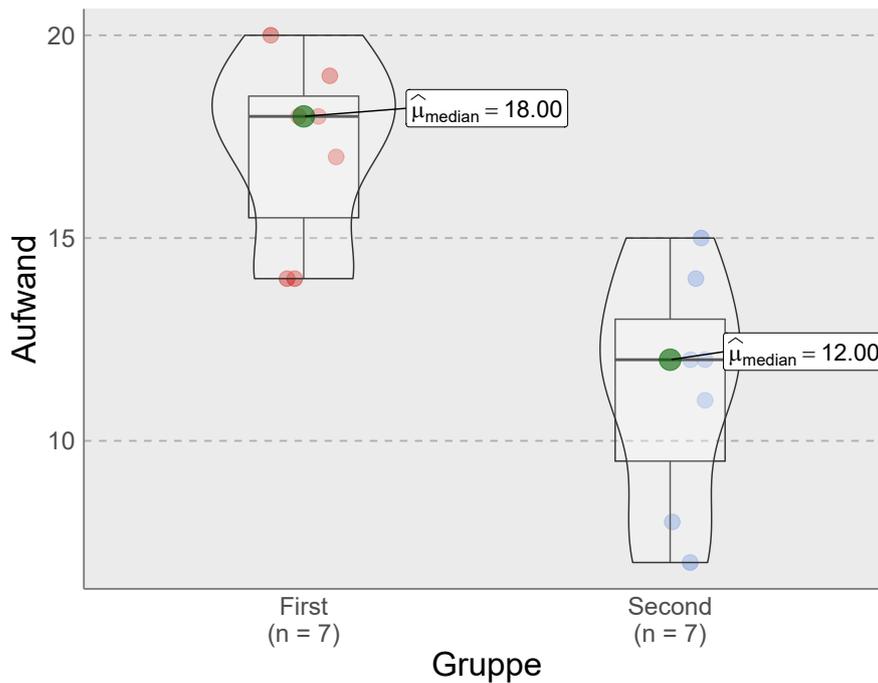


Abbildung 4.42: Auswertung des aufgebrauchten ASQ-Aufwandes des ersten und zweiten Durchlaufes (First / Second).

- **ASQ-Informationszufriedenheit (Information satisfaction):**

Die Zufriedenheit mit den angezeigten beziehungsweise von der Integration bereitgestellten Informationen bewerteten die Probanden im ersten Durchlauf durchschnittlich mit $\bar{X} \approx 11,43$ Punkten, wobei der Median $\tilde{X} = 11$ Punkte und die Standardabweichung $S \approx 2,15$ Punkte betrug. Auf die einzelnen Aufgaben umgerechnet, bedeutet dies einen durchschnittlichen Wert von $\bar{X} \approx 1,63$ Punkten. Im zweiten Durchlauf wurde die Zufriedenheit mit durchschnittlich $\bar{X} \approx 9,29$ Punkten und einem Median von $\tilde{X} = 8$ Punkten bewertet. Die Standardabweichung betrug dabei $S \approx 2,81$ Punkte. Umgerechnet auf die einzelnen Aufgaben ergibt sich hier durchschnittlich eine Punktzahl von $\bar{X} \approx 1,33$ Punkten. Die Zufriedenheit lag im ersten und im zweiten Durchlauf zwischen einer sehr starken und starken Zustimmung, jedoch liegt die Zustimmung des zweiten Durchlaufes deutlich näher an der sehr starken Zustimmung als die des ersten. Die Zufriedenheit stieg folglich um durchschnittlich 18,71%. Es wurde hierzu ein signifikanter Unterschied mit dem Ergebnis $V = 2; p \approx 0,03$ festgestellt. Dadurch kann die Nullhypothese \mathcal{H}_{k_0} verworfen und die Alternative Hypothese \mathcal{H}_{k_1} angenommen werden. In der [Abbildung 4.43](#) kann dieser signifikante Unterschied der Bewertung begutachtet werden.

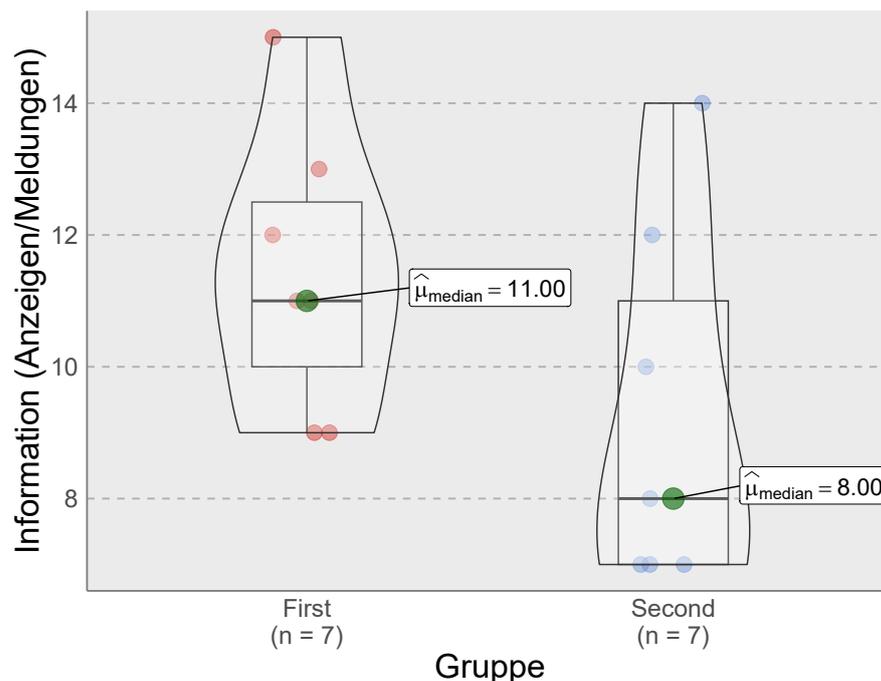


Abbildung 4.43: Auswertung der ASQ-Informationszufriedenheit des ersten und zweiten Durchlaufes (First / Second).

- **Benötigte Zeit:**

Durchschnittlich benötigten die Probanden im ersten Durchlauf $\bar{x} \approx 45,21$ Minuten. Der Median betrug hierbei $\tilde{x} = 45,65$ Minuten und die Standardabweichung $S \approx 8,99$ Minuten. Im zweiten Durchlauf benötigten sie durchschnittlich $\bar{x} \approx 32,69$ Minuten. Der Median lag hierbei bei $\tilde{x} \approx 28,33$ Minuten und die Standardabweichung bei $S \approx 8,75$ Minuten. Dadurch ergibt sich eine Zeitersparnis von durchschnittlich $\bar{x} \approx 12,52$ Minuten, während der Median $\tilde{x} \approx 10,93$ Minuten und die Standardabweichung $S \approx 6,52$ Minuten betrug. Das bedeutet, dass die Probanden während des zweiten Durchlaufes durchschnittlich 27,68% Zeit gespart haben. Der Unterschied zwischen der benötigten Zeit beider Durchläufe war mit $V = 28; p \approx 0.0078$ signifikant. Deshalb kann die Nullhypothese \mathcal{H}_{h_0} verworfen und die Alternative Hypothese \mathcal{H}_{h_1} angenommen werden. In der [Abbildung 4.44](#) können die Unterschiede der benötigten Zeit der Probanden gruppiert als erster und zweiter Durchlauf betrachtet werden und in der [Abbildung 4.45](#) wird die Zeitersparnis jedes einzelnen Probanden dargestellt.

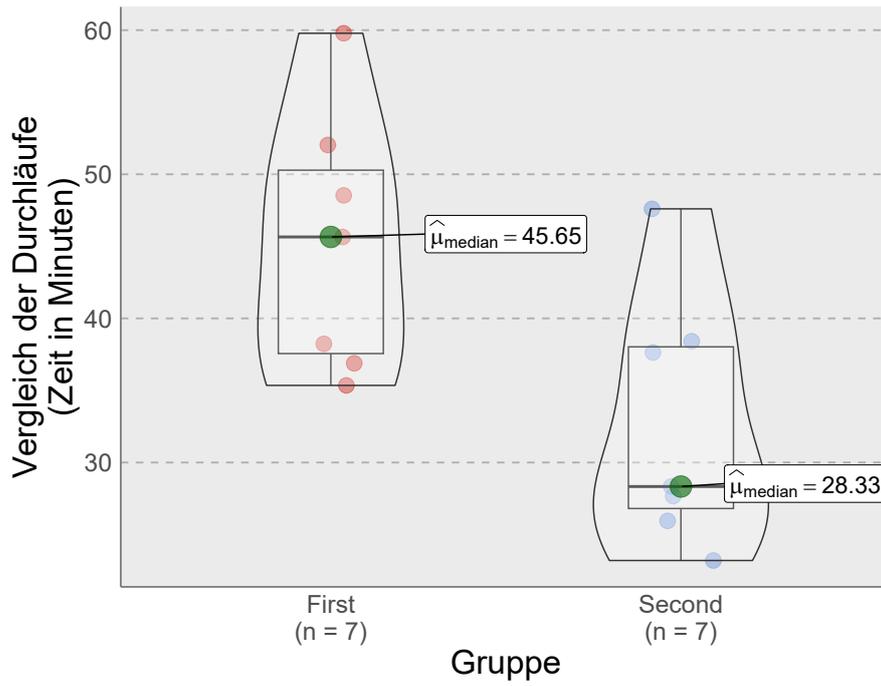


Abbildung 4.44: Auswertung der benötigten Zeit des ersten und zweiten Durchlaufes (First / Second).

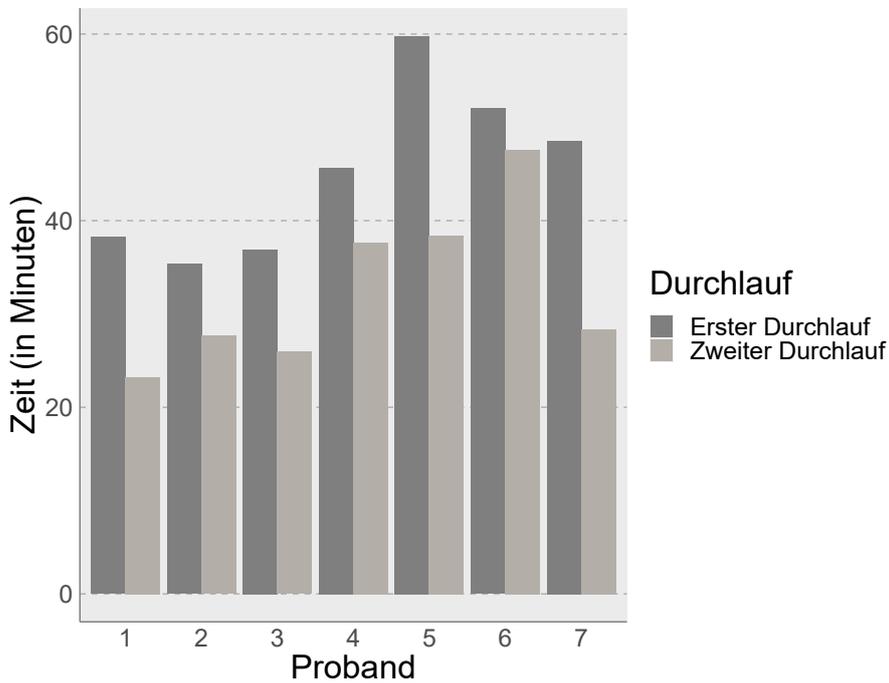


Abbildung 4.45: Auswertung der Zeitersparnis zwischen dem ersten und zweiten Durchlauf jedes Probanden.

Basierend auf den zuvor erläuterten Ergebnissen, kann die dritte Forschungsfrage dahingehend beantwortet werden, dass sich die Wahrneh-

mung der Benutzerfreundlichkeit, die benötigte Zeit für die Aufgabenbewältigung, die empfundene Komplexität und der wahrgenommene Aufwand sowie die Zufriedenheit mit den angezeigten Informationen bei einer erneuten Verwendung signifikant verbessert. Zudem ergibt sich eine Zeitersparnis von durchschnittlich 27,68%.

Diese Erkenntnis zeigt, dass es sich generell für Benutzerfreundlichkeitsstudien lohnen würde, eine Messung zu wiederholen. Dadurch würde auch ein höherer Realismus dargestellt werden, da Menschen im realen Leben neue Werkzeuge auch erst erlernen müssen. Aus diesem Grund wäre es für Benutzerfreundlichkeitsstudien sinnvoll, mindestens eine zweite Messung durchzuführen und die Ergebnisse der zweiten Messung für die Auswertung zu verwenden, um den Realismus repräsentativer zu gestalten.

Im Folgenden werden die erhobenen Daten der einzelnen Aufgaben vorgestellt. Zunächst wird die Zeitersparnisse der einzelnen Aufgaben erläutert, gefolgt von den signifikanten Unterschieden in Bezug auf die benötigte Zeit sowie die ASQ-Daten der einzelnen Aufgaben.

Wie bereits bei der Untersuchung der zweiten Forschungsfrage können auch hier die Diagramme der einzelnen Aufgaben mithilfe des zuvor erwähnten R-Skriptes erzeugt und anschließend betrachtet werden. Auf dem mitgelieferten USB-Stick befinden sich zusätzlich auch diese Diagramme. Diese Diagramme werden aufgrund ihrer hohen Anzahl in dieser Arbeit nicht eingebunden. Die Gesamtbetrachtung ist für die Untersuchung der Forschungsfrage relevanter als jede einzelne Aufgabe. Außerdem würden diese Diagramme zu viele Seiten in Anspruch nehmen. Des Weiteren ist der Mehrwert für diese Einbindung nicht gegeben, da die relevanten Informationen für die einzelnen Aufgaben aus den folgenden Tabellen 4.11 - 4.17 entnommen werden können. In Zahlen ausgedrückt, wären es für die benötigte Zeit beziehungsweise Zeitersparnis vierzehn zusätzliche Diagramme. Jeweils zwei Diagramme pro Aufgabe, ein Violin-Plot-Diagramm, das den Vergleich als zwei Gruppen (1. Durchlauf / 2. Durchlauf) darstellt, und ein gruppiertes Balkendiagramm, das die erzielten Zeitunterschiede jedes Probanden aufzeigt. Für die ASQ-Daten wären dies 21 zusätzliche Diagramme, ein Violin-Plot-Diagramm für jeden der drei ASQ-Aspekte pro Aufgabe.

Zunächst werden die Zeitersparnisse der einzelnen Aufgaben betrachtet, die aus der folgenden [Tabelle 4.1](#) entnommen werden können. Es ist deutlich zu erkennen, dass die meiste Zeit während der ersten Aufgabe gespart werden konnte. Das erscheint auch logisch, da die Probanden im ersten Durchlauf noch nicht wirklich mit der Integration vertraut waren. Einige Probanden zeichneten in der Einführung lediglich zwei kurze Linien. Überraschend ist, dass mit der siebten Aufgabe, die sehr kurz und schnell auszuführen war, dennoch durchschnittlich 34,48%

der Zeit eingespart werden konnte. Es ist gut zu erkennen, dass bei den längeren Aufgaben (1, 2, 3 und 6) über 25% der Zeit gespart werden konnte, was die Notwendigkeit einer zweiten Messung für die Benutzerfreundlichkeitsstudien zusätzlich unterstreicht.

Aufg.	A	M	S	Prozentuale Zeitersparnis
A1	3,09	3,05	1,37	37,18%
A2	3,15	3,32	1,71	29,49%
A3	2,43	2,1	1,96	27,43%
A4	0,27	0,03	0,53	8,21%
A5	0,58	0,43	1,13	18,71%
A6	2,4	3,12	1,91	26,03%
A7	0,6	0,62	0,19	34,48%

Tabelle 4.10: Die Zeitersparnisse der einzelnen Aufgaben in Minuten. Die prozentuale Zeitersparnis wurde bemessen am Durchschnitt.

Abschließend wird nun auf die signifikanten Unterschiede der einzelnen Aufgaben eingegangen. Die signifikanten Unterschiede werden dabei erneut grün und annähernd signifikante Unterschiede orange dargestellt. In der ersten Aufgabe wurden für die benötigte Zeit, die empfundenen Komplexität und den wahrgenommenen Aufwand signifikante Unterschiede festgestellt. Die Informationszufriedenheit war mit $V = 18; p \approx 0,0599$ annähernd signifikant. Dasselbe zeigt sich auch für die zweite Aufgabe. In der dritten wurde ebenfalls ein signifikanter Unterschied bei der benötigten Zeit, der empfundenen Komplexität sowie des wahrgenommenen Aufwandes festgestellt. Die Überprüfungen der sechsten Aufgabe ergaben einen signifikanten Unterschied in Bezug auf die benötigte Zeit und den empfundenen Aufwand. Bei der siebten Aufgabe wurde abschließend ein signifikanter Unterschied in der benötigten Zeit ermittelt. Die zuvor beschriebenen signifikanten Unterschiede für die Gesamtbetrachtung aller Aufgaben ergeben sich auch hier daraus, dass die Werte nicht gerundet wurden und die kleinen Unterschiede in den einzelnen Aufgaben sich zu einem signifikanten Unterschied summieren.

Data	1. Durchlauf			2. Durchlauf			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	8,31	7,8	1,23	5,22	5,07	0,99	$V = 28; p \approx 0,008$
Komplexität	3,29	3	0,49	2,143	2	0,69	$V = 21; p \approx 0,02$
Aufwand	3,71	4	0,76	2	2	0,82	$V = 21; p \approx 0,02$
Information	2,14	2	0,69	1,43	1	0,53	$V = 18; p \approx 0,0599$

Tabelle 4.11: Die Daten des Vergleiches der Aufgabe 1 zwischen erstem und zweitem Durchlauf.

Data	1. Durchlauf			2. Durchlauf			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	10,68	1,92	1,77	7,53	6,57	2,38	$V = 28; p \approx 0,008$
Komplexität	3	3	0,82	2	2	0,82	$V = 25; p \approx 0,03$
Aufwand	3,43	3	0,98	2	2	1	$V = 19,5; p \approx 0,04$
Information	2,14	2	0,69	1,43	1	0,53	$V = 18; p \approx 0,0599$

Tabella 4.12: Die Daten des Vergleiches der Aufgabe 2 zwischen erstem und zweitem Durchlauf.

Data	1. Durchlauf			2. Durchlauf			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	8,86	7,7	2,38	6,43	5,92	1,77	$V = 27; p \approx 0,02$
Komplexität	2,57	3	0,98	1,86	2	0,68	$V = 10; p \approx 0,04$
Aufwand	2,86	3	0,69	1,71	2	0,49	$V = 28; p \approx 0,007$
Information	1,86	2	0,69	1,29	1	0,49	$V = 6; p \approx 0,09$

Tabella 4.13: Die Daten des Vergleiches der Aufgabe 3 zwischen erstem und zweitem Durchlauf.

Data	1. Durchlauf			2. Durchlauf			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	3,29	3,23	0,73	3,02	3,2	0,83	$V = 19; p \approx 0,23$
Komplexität	1,71	2	0,76	1,43	1	0,53	$V = 4,5; p \approx 0,29$
Aufwand	1,57	2	0,53	1,43	1	0,53	$V = 4; p \approx 0,39$
Information	1,29	1	0,49	1,29	1	0,49	$V = 1,5; p \approx 0,68$

Tabella 4.14: Die Daten des Vergleiches der Aufgabe 4 zwischen erstem und zweitem Durchlauf.

Data	1. Durchlauf			2. Durchlauf			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	3,1	3,33	0,68	2,52	2,17	1,09	$V = 21; p \approx 0,15$
Komplexität	2,14	2	0,9	1,71	2	0,49	$V = 8; p \approx 0,17$
Aufwand	2	2	1	1,43	1	0,53	$V = 6; p \approx 0,09$
Information	1,43	1	0,53	1,43	1	0,53	$V = 0; p \approx 1$

Tabella 4.15: Die Daten des Vergleiches der Aufgabe 5 zwischen erstem und zweitem Durchlauf.

Data	1. Durchlauf			2. Durchlauf			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	9,22	9,47	2,86	6,82	6,58	2,71	$V = 27; p \approx 0,02$
Komplexität	2,57	2	1,13	1,71	2	0,76	$V = 6; p \approx 0,09$
Aufwand	2,43	3	0,79	1,57	2	0,53	$V = 10; p \approx 0,047$
Information	1,57	2	0,53	1,29	1	0,49	$V = 7,5; p \approx 0,21$

Tabella 4.16: Die Daten des Vergleiches der Aufgabe 6 zwischen erstem und zweitem Durchlauf.

Data	1. Durchlauf			2. Durchlauf			Unterschied
	A	M	S	A	M	S	
benötigte Zeit	1,74	1,75	0,28	1,14	1,03	0,26	$V = 28; p \approx 0,008$
Komplexität	1,57	1	0,79	1,14	1	0,38	$V = 8; p \approx 0,17$
Aufwand	1,14	1	0,38	1,14	1	0,38	$V = 1,5; p \approx 0,68$
Information	1	1	0	1,14	1	0,38	$V = 0; p \approx 0,98$

Tabelle 4.17: Die Daten des Vergleiches der Aufgabe 7 zwischen erstem und zweitem Durchlauf.

4.5.5 Vergleich Informatiker und zweiter Durchlauf der Nicht-Informatiker

Aufgrund der signifikanten Verbesserung der Ergebnisse bei den Nicht-Informatikern im zweiten Durchlauf und der Feststellung signifikanter Unterschiede zwischen den Informatikern und Nicht-Informatikern in der Untersuchung zur zweiten Forschungsfrage, sollten diese Unterschiede erneut überprüft werden. Dazu werden die erhobenen Daten der Informatiker aus dem ersten Studiendurchlauf mit den erhobenen Daten der Nicht-Informatiker aus dem zweiten Durchlauf verglichen. Obwohl es wahrscheinlich ist, dass auch Informatiker sich im zweiten Durchlauf verbessern würden, sind Informatiker erfahrener im Umgang mit IT-Systemen. Daher ist es aussagekräftiger für die Beantwortung der Forschungsfrage, die Daten der Nicht-Informatiker aus dem zweiten Durchlauf zu verwenden. Auf diese Weise kann überprüft werden, wie sich dieser Unterschied darstellt, wenn die Nicht-Informatiker die Integration zunächst kennenlernen konnten. Wie bereits beschrieben, führt das vorherige Kennenlernen eines Systems vor der eigentlichen Studie zu realistischeren Ergebnissen.

Es gab einen signifikanten Unterschied zwischen den Informatikern und Nicht-Informatikern bezogen auf die Komplexität sowie den benötigten Aufwand zur Aufgabenbewältigung. Die fast Signifikanz wurde in der benötigten Zeit der Aufgabenbewältigung festgestellt.

Die Überprüfung des Unterschiedes zwischen den Informatikern und Nicht-Informatikern hinsichtlich der Komplexität ergab mit den neuen Daten keinen signifikanten Unterschied mehr. Hierzu wurden erneut beide möglichen Fälle (Informatiker > Nicht-Informatiker und Informatiker < Nicht-Informatiker) überprüft. Der Mann-Whitney-U-Test lieferte folgende Ergebnisse: $U = 16,5; p \approx 0,282$ und $U = 25,5; p \approx 0,764$. Damit kann die *Nullhypothese* \mathcal{H}_{d_0} doch nicht widerlegt werden. Aufgrund der Untersuchung beider Seiten kann ausgeschlossen werden, dass es einen signifikanten Unterschied bei der empfundenen Komplexität der Aufgaben für eine der beiden Gruppen besteht.

Bei der Überprüfung des Aufwandes ergab der Test die folgenden Ergebnisse: $U = 22; p \approx 0,586$ und $U = 20; p \approx 0,471$. Demnach besteht auch hier kein signifikanter Unterschied mehr. Deshalb kann die *Nullhypothese* \mathcal{H}_{e_0} nicht widerlegt werden.

Abschließend wurde der fast signifikante Unterschied der benötigten Zeit überprüft. Der Test ergab keinen fast signifikanten Unterschied mehr ($U = 24; p \approx 0,686$ und $U = 18; p \approx 0,365$). Die Nicht-Informatiker waren in ihrem zweiten Durchlauf durchschnittlich 1,05 Minuten schneller als die Informatiker im ersten Durchlauf, was einem prozentualen Wert von 3,11% entspricht. Es ist zu beachten, dass Informatiker höchstwahrscheinlich im zweiten Durchlauf ebenfalls schneller wären, jedoch nahm leider kein Informatiker an dem zweiten Durchlauf teil.

Aufgrund des Ausschlusses der signifikanten Unterschiede kann angenommen werden, dass sowohl Informatiker als auch Nicht-Informatiker die Integration zufriedenstellend bedienen und problemlos nutzen können. Folglich kann die zweite Forschungsfrage durch den folgenden Absatz ergänzt werden: Bei einer erneuten Verwendung der Integration konnten die Nicht-Informatiker die signifikanten Unterschiede zu den Informatikern des ersten Durchlaufes ausgleichen, sodass keine signifikanten Unterschiede mehr bestehen. Dadurch kann gesagt werden, dass bei einer erneuten Verwendung der Integration durch die Nicht-Informatiker, die Informatiker nicht mehr schneller und mit geringerem Aufwand die Aufgaben ausführen können.

4.5.6 Korrelationen

Der folgende Abschnitt befasst sich mit Korrelationen zwischen den abhängigen und unabhängigen Variablen. Hierzu wird der *Kendall-Tau-Korrelationskoeffizient* mit einem Signifikanzniveau von $p = 0,05$ verwendet. Positive Zusammenhänge werden dabei in **grün**, negative in **rot** und keine Zusammenhänge in schwarz dargestellt. Zusammenhänge, die fast signifikant sind, werden in **orange** dargestellt.

Die folgende **Tabelle 4.18** enthält die Ergebnisse des Kendall-Tau-Korrelationskoeffizienten für die unabhängigen Variablen mit der abhängigen Variable der benötigten Zeit der Aufgabenbewältigung. Hier zeigt sich eine signifikante Korrelation zwischen der benötigten Zeit und der Art der Studiendurchführung zu erkennen. Diese Korrelation legt nahe, dass Probanden die sich mit dem Studieninitiator im selben Raum befanden, länger für die Aufgaben benötigten als die, die nur per Zoom mit ihm im Kontakt standen. Es könnte vermutet werden, dass der Studieninitiator die Probanden im selben Raum abgelenkt hätte, jedoch ist dies nicht der Fall. Vielmehr liegt es daran, dass die Probanden

Kendall-Tau-Korrelationskoeffizient:

Mit diesem Korrelationstest wird geprüft, ob zwei Variablen einen ungerichteten Zusammenhang besitzen. Dabei ist ein positiver, negativer oder kein Zusammenhang möglich (Walther, 2022). Beide Variablen müssen nicht normalverteilt sein und benötigen lediglich ein Ordinalskalenniveau (DATAtab_Team, 2024a).

im selben Raum Nicht-Informatiker waren und diese länger für die Durchführung benötigten.

	benötigte Zeit		
	τ	p	z
Informatik Zugehörigkeit	0,39	0,09	1,68
Alter	0,18	0,38	0,88
Geschlecht	0,23	0,32	0,99
höchster Bildungsabschluss	-0,11	0,61	-0,51
SEE-Erfahrung	-0,39	0,09	-1,7
Bildbearbeitungsprogramm-Erfahrung	-0,35	0,12	-1,55
Videospiel-Erfahrung	-0,28	0,22	-1,23
Art der Studiendurchführung	0,52	0,03	2,24

Tabelle 4.18: Ergebnisse der Korrelationsprüfung der abhängigen Variable "benötigte Zeit".

Die Ergebnisse der Korrelationsprüfung für die unabhängigen Variablen und der abhängigen Variable "SUS" sind in der folgenden [Tabelle 4.19](#) dargestellt. Hierbei wurde keine signifikante Korrelation festgestellt.

	SUS		
	τ	p	z
Informatik Zugehörigkeit	0	1	0
Alter	0,26	0,2	1,28
Geschlecht	-0,05	0,83	-0,21
höchster Bildungsabschluss	-0,19	0,39	-0,86
SEE-Erfahrung	-0,06	0,8	-0,25
Bildbearbeitungsprogramm-Erfahrung	0,07	0,76	0,31
Videospiel-Erfahrung	-0,02	0,95	-0,06
Art der Studiendurchführung	-0,14	0,56	-0,58

Tabelle 4.19: Ergebnisse der Korrelationsprüfung der abhängigen Variable "SUS".

Die Korrelationsuntersuchung mit der abhängigen Variable "ASQ-Komplexität" ist in der [Tabelle 4.20](#) dargestellt und ergab drei signifikante Korrelationen sowie eine fast signifikante Korrelation. Von den drei signifikanten Korrelationen sind zwei positiv und eine negativ. Die erste positive Korrelation deutet darauf hin, dass Informatiker die gestellten Aufgaben im Vergleich zu Nicht-Informatiker als leichter empfanden. Die zweite positive Korrelation zeigte, dass Probanden, die sich physisch im selben Raum wie der Studieninitiator befanden, die Aufgaben als schwieriger empfanden als diejenigen, die nur per Zoom mit ihm in Kontakt standen. Wie bereits bei der Untersuchung der

Korrelation der benötigten Zeit erwähnt, sind diejenigen, die mit ihm im selben Raum befanden, die Nicht-Informatiker. Diese empfanden nach der zuerst festgestellten Korrelation die Aufgaben als schwieriger. Die negative Korrelation besagt, dass Probanden mit mehr Videospieelerfahrung die Aufgaben als leichter empfanden als diejenigen mit weniger Spielerfahrung. Die fast Korrelation zeigt eine negative Beziehung zur Erfahrung mit Bildbearbeitungsprogrammen auf. Auch bei dieser empfanden Probanden mit mehr Erfahrung die Aufgaben als leichter.

	ASQ-Komplexität		
	τ	p	z
Informatik Zugehörigkeit	0,48	0,04	2,01
Alter	0,19	0,35	0,94
Geschlecht	0,19	0,43	0,78
höchster Bildungsabschluss	0,13	0,57	0,57
SEE-Erfahrung	-0,43	0,065	-1,84
Bildbearbeitungsprogramm-Erfahrung	-0,45	0,053	-1,93
Videospiel-Erfahrung	-0,48	0,04	-2,09
Art der Studiendurchführung	0,613	0,01	2,57

Tabelle 4.20: Ergebnisse der Korrelationsprüfung der abhängigen Variable "ASQ-Komplexität" (Leichtigkeit).

Die nachfolgende [Tabelle 4.21](#) zeigt die Ergebnisse der Korrelationsuntersuchung mit der abhängigen Variable "ASQ-Aufwand". Hierbei wurden erneut zwei positive, eine negative und eine fast Korrelation festgestellt. Die erste positive zeigt, dass Informatiker weniger Aufwand zur Aufgabenbewältigung benötigen als Nicht-Informatiker. Die zweite positive besagt erneut, dass Probanden, die sich mit dem Studieninitiator im selben Raum befanden, mehr Aufwand benötigten. Diese Probanden waren, wie bereits bekannt, die Nicht-Informatiker und diese benötigen nach der erstgenannten Korrelation mehr Aufwand. Die negative Korrelation besagt, dass Probanden mit mehr Videospieelerfahrung weniger Aufwand zur Aufgabenbewältigung benötigen als diejenigen mit weniger Spielerfahrung. Die fast Korrelation zeigt eine negative Beziehung zur Erfahrung mit SEE. Sie zeigt, dass Probanden mit mehr SEE-Erfahrung weniger Aufwand benötigen als diejenigen mit weniger Erfahrung.

	ASQ-Aufwand		
	τ	p	z
Informatik Zugehörigkeit	0,53	0,027	2,2
Alter	0,2	0,35	0,94
Geschlecht	0,2	0,39	0,85
höchster Bildungsabschluss	0,05	0,82	0,23
SEE-Erfahrung	-0,44	0,057	-1,9
Bildbearbeitungsprogramm-Erfahrung	-0,44	0,06	-1,87
Videospiel-Erfahrung	-0,51	0,026	-2,22
Art der Studiendurchführung	0,66	0,006	2,77

Tabella 4.21: Ergebnisse der Korrelationsprüfung der abhängigen Variable "ASQ-Aufwand".

Die [Tabella 4.22](#) enthält die Ergebnisse der Korrelationsuntersuchung mit der abhängigen Variable "ASQ-Informationszufriedenheit". Hierzu konnte keine signifikante Korrelation mit den unabhängigen Variablen festgestellt werden.

	ASQ-Info.		
	τ	p	z
Informatik Zugehörigkeit	0,26	0,29	1,05
Alter	0,05	0,82	0,22
Geschlecht	-0,12	0,61	-0,5
höchster Bildungsabschluss	-0,03	0,91	-0,12
SEE-Erfahrung	-0,17	0,48	-0,7
Bildbearbeitungsprogramm-Erfahrung	-0,19	0,41	-0,82
Videospiel-Erfahrung	-0,29	0,21	-1,25
Art der Studiendurchführung	0,32	0,19	1,3

Tabella 4.22: Ergebnisse der Korrelationsprüfung der abhängigen Variable "ASQ-Informationszufriedenheit".

In der letzten [Tabella 4.23](#) werden die Korrelationsuntersuchungen für die abhängige Variable der Zeitersparnis der dritten Forschungsfrage dargestellt. Einige der Untersuchungen zeigen "NA" als Ergebnis, da dies aufgrund der Tatsache, dass die Probanden des zweiten Durchlaufes nur Nicht-Informatiker waren und die Antworten der Probanden sich für die jeweilige unabhängige Variable nicht unterschieden. Die Untersuchung ergab eine negative Beziehung zum Alter der Probanden. Diese Korrelation besagt, dass jüngere Probanden im zweiten Durchlauf mehr Zeit sparen können als ältere.

	Zeitersparnis		
	τ	p	z
Informatik Zugehörigkeit	NA	NA	NA
Alter	-0,68	0,03	-2,13
Geschlecht	0,25	0,48	0,71
höchster Bildungsabschluss	-0,31	0,35	-0,94
SEE-Erfahrung	NA	NA	NA
Bildbearbeitungsprogramm-Erfahrung	0,07	0,85	0,19
Videospiel-Erfahrung	0,58	0,09	1,7
Art der Studiendurchführung	NA	NA	NA

Tabelle 4.23: Ergebnisse der Korrelationsprüfung der abhängigen Variable "Zeitersparnis".

4.5.7 Kommentare der Probanden

In diesem Abschnitt wird über die geäußerten Kommentare der Probanden während der Studie und die geschriebenen Anmerkungen in den Kommentarfeldern ausgewertet. Leider nutzte nicht jeder Proband die Möglichkeit, Kommentare schriftlich zu äußern.

Lediglich ein Proband nutzte ein Kommentarfeld der einzelnen Aufgaben. Er äußerte die Präferenz, dass nach dem Aktivieren der "Move"-Funktionalität der [Move or Rotate-Aktion](#) bei der Desktop-Variante automatisch die Option des Verschiebens mittels der Maus aktiviert werden sollte.

Die geäußerten Verbesserungsvorschläge werden im nachfolgenden Abschnitt "[Ausblick](#)" näher erläutert. Die Probanden äußerten als Vorschläge:

- dass die Hand von SEE störend ist und deshalb ausblendbar sein sollte,
- dass das Aktions-Menü in weitere Untermenüs unterteilt werden sollte,
- oder dass der Benutzer die Menüpunkte (alphabetisch oder nach Relevanz) sortieren lassen kann,
- dass der Benutzer mehr Benachrichtigungen in den Aktionen erhält
- sowie die Bereitstellung einer deutschen Übersetzung.

Das Aktionsauswahl-Menü ist derzeit nach den Aktionskategorien sortiert. Das bedeutet, dass zuerst die Aktionen, mit denen Drawable-Typ-Objekte hinzufügen kann, aufgeführt werden. Danach folgt die [Color Picker-Aktion](#) und die [Transformations-Aktionen](#), gefolgt von

den Aktionen zum Speichern und Laden sowie die [Lösch-Aktionen](#). Zum Schluss wird die [Sticky-Note-Aktion](#) aufgeführt.

Die geäußerten Zukunftsfeaturewünsche umfassten lediglich die Einbindung von Video- und Audiodateien.

Als weitere Anmerkungen wurden die folgenden geäußert:

- “Anfangs sehr ungewohnt, aber man kommt sehr schnell in die Funktionen rein“
- “Die Menü Verschachtelung erscheint mir etwas tief.“
Dieser Kommentar stammt von dem im Abschnitt “[RQ2: Vergleich Informatiker und Nicht-Informatiker](#)“ beschriebenen SUS-Ausreißer der Informatiker. Der Kommentar steht im Widerspruch zu dem, was die anderen Probanden äußerten, mit dem Wunsch, die Verschachtelung zu erhöhen, indem die Aktionen in weiteren Untermenüs unterteilt werden sollten. Sollte der Proband mit der Menüverschachtelung die bereitgestellten Menüs der Aktionen meinen, ist leider auch nicht nachvollziehbar, auf welches Menü sich der Proband bezieht.
- “Sehr angenehm zu verwenden. Positiv ist, dass Funktionen wie Edit verschiedene Aktionen abhängig vom ausgewählten Objekt ausführen.“
- “Mega gut!“

In der folgenden Aufzählung wird von den geäußerten gesprochenen Kommentaren der Probanden während der Studiendurchführung berichtet:

- “Nächstes Mal wäre ich schneller.“
- “Ich wünschte, wir hätten in der Schule so etwas.“
- “Ich fand es echt cool, hat mir viel Spaß bereitet.“
- “Erinnert mich an meine Schulzeit.“ Diese Aussage bezog sich auf das Erstellen von Mind Maps.
- “Die Studie hat mir viel Spaß gemacht“
- “Dein Programm ist sehr intuitiv.“

4.6 THREATS TO VALIDITY

Dieser Abschnitt beschreibt die Validität der Studie. Validität bezieht sich auf die Gültigkeit einer Messung beziehungsweise eines Forschungsergebnisses bezogen auf ihre Zielsetzung. Eine Studie ist valide, wenn sie das misst, was beabsichtigt ist. Eine valide Studie beziehungsweise Forschung liefert somit verwertbare und überzeugende

Ergebnisse für die Wissenschaft. Hierbei wird zwischen der **internen** und der **externen** Validität unterschieden (Genau, 2023).

Im Folgenden werden die relevanten Validitäten der Studie beschrieben, die Koschke (2021) in seiner Softwaretechnik-Vorlesung vorgestellt hat.

4.6.1 Interne Validität

Mit dieser Validität wird bewertet, ob die Ergebnisse intern valide sind, das bedeutet, dass tatsächlich das gemessen wird, was beabsichtigt wurde, um so glaubwürdige und zuverlässige Ergebnisse zu liefern (Pfeiffer, 2021). Dies ist notwendig, um sicherzustellen, dass die beobachteten Resultate nicht auf beispielsweise Störvariablen zurückzuführen sind.

- **Historie/Zeit:** Die Durchführungen der Studie fanden in einem Zeitraum zwischen 10 Uhr und 20 Uhr abends statt. Die Auswahl der Uhrzeit und des Tages für die Durchführung richtete sich individuell nach den Probanden, je nachdem, wann sie Zeit für die Teilnahme hatten. Daher kann ein Einfluss auf den Tag und die Uhrzeit ausgeschlossen werden. Es ist jedoch möglich, dass die gewählte Zeit der Probanden nicht ihrem natürlichen Bio-Rhythmus entspricht, da einige von ihnen erst nach oder vor ihrer regulären Arbeit Zeit hatten. Dies könnte potenziell einen negativen Einfluss auf die benötigte Zeit für die Aufgabenbewältigung haben, da sie möglicherweise nicht ihr maximales Leistungspotenzial aufbringen können. Mau und Haase (2021) beschrieben in ihrem Beitrag, dass jeder Mensch täglich Phasen hat, in denen er besonders fit, kreativ und leistungsfähig ist. Diese Phasen richten sich nach der inneren Uhr des jeweiligen Menschen und sind bei jedem Menschen unterschiedlich.
- **Störvariable:** Als Störvariable werden Variablen bezeichnet, die neben den erhobenen unabhängigen Variablen auch die abhängigen Variablen beeinflussen können (Ortmann, 2021). Eine solche Störvariable ist, dass nicht jeder Proband dieselbe Computerkonfiguration für die Studiendurchführung verwendet hat. Dies kann die benötigte Zeit zur Aufgabenbewältigung leicht beeinflussen. Der Großteil der Nicht-Informatiker verwendete den Computer des Studieninitiators, jedoch keiner der Informatiker. Hinzu kommt, dass nicht alle Probanden einen zweiten Bildschirm besaßen und somit zwischen der Visualisierung und dem Webbrowser hin- und herwechseln mussten, um die Studie durchzuführen, was sich natürlich negativ auf die Zeit auswirkt. Jedoch war diese Zeitdifferenz in der späteren Auswertung nicht sichtbar. Dies liegt wahrscheinlich daran, dass dieser Proband bereits an das Arbeiten mit einem Bildschirm gewöhnt ist. Es ist fraglich, ob alle Probanden ihre Erfahrungen mit den Video-

spielen und Bildbearbeitungsprogrammen richtig eingeschätzt haben. Bogner und Landrock (2015) beschrieben das Problem der sozialen Erwünschtheit, bei dem Probanden eine positive Beschreibung der eigenen Person abgeben möchten und daher unter Umständen die bisherige Erfahrung fehlerhaft einschätzen, da diese als unerwünscht angesehen werden kann. Dies könnte beispielsweise bei einer zu hohen Videospieldauer eintreten, um nicht als "Videospielesüchtig" eingestuft zu werden. Hinzu kommt noch die mögliche Selbsttäuschung, die beschreibt, dass eine Person bewusst oder unbewusst ihre Selbstwahrnehmung beziehungsweise Selbstbild verfälscht und sich selbst belügt, um sich selbst etwas vorzumachen (Mai, 2024).

- **Sättigung:** Durchaus könnten bei einer Studiendurchführung von etwa einer Stunde Ermüdungserscheinungen auftreten. Jedoch wurde bei keinem Probanden eine solche Erscheinung beobachtet. Jeder Proband hat die Aufgaben erfolgreich absolviert. Zudem wurde die Studienlänge so angepasst, dass Ermüdungseffekte vermieden werden konnten.
- **Wiederholtes testen:** Der Effekt des wiederholten Testens in Form von Lerneffekten tritt während der ersten Studiendurchführung nur insofern auf, als dass die Aktionsdurchführungen ähnlich sind. Allerdings liegt der Fokus der einzelnen Aufgaben auf verschiedene Aktionen, weshalb dieser Effekt ausgeschlossen werden kann. Bei der zweiten Studiendurchführung tritt der Effekt logischerweise ein und ist Teil der Untersuchung.
- **Experimentoreinfluss:** Ungefähr die Hälfte der Probanden kennt den Studieninitiator persönlich und könnten daher ihre Bewertungen möglicherweise an sein gewünschtes Ergebnis angepasst haben. Dies erscheint jedoch unwahrscheinlich, da die andere Hälfte der Probanden die Integration ähnlich bewertet hat.
- **Selektion:** Die Nicht-Informatiker stammen aus den unterschiedlichsten Berufen, während die Informatiker ausschließlich aus dem SEE-Projekt stammen. Ursprünglich waren auch Informatiker vorgesehen, die nicht Teil des SEE-Projekts sind, jedoch haben diese leider abgesagt, da sie keine Zeit für die Durchführung fanden. Allerdings dürfte die Erfahrung mit SEE keinen Einfluss auf die Ergebnisse haben, da die Integration unabhängig von den bereitgestellten Funktionalitäten von SEE funktioniert.

4.6.2 Externe Validität

Diese Validität bewertet, inwieweit die Ergebnisse auch auf andere Situationen, Gruppen oder Personen beziehungsweise der gesamten

Bevölkerung verallgemeinert werden können. Voraussetzung für diese Validität ist die [interne Validität](#). (Pfeiffer, 2023).

- **Repräsentativ:** Zwar wurde der bekannte Frauenanteil in der Informatik abgedeckt, jedoch ist über die genaue Gesamtpopulation der Informatiker nichts bekannt. Zudem besteht, wie zuvor in der Selektion beschrieben, die Gruppe der Informatiker ausschließlich aus SEE-Projektteilnehmern und besteht ausschließlich aus Informatik-Studenten. Einige von ihnen haben bereits ihren Bachelorabschluss in Informatik und arbeiten nebenbei als Entwickler. Es könnte jedoch einen deutlichen Erfahrungsunterschied zu langjährigen Informatikern geben. Daher dürfte die Gruppe der Informatiker nicht repräsentativ für die Gesamtpopulation der Informatiker sein.
- **Situation:** Die Tageszeiten sollten repräsentativ sein, da der Zeitraum zwischen 10 Uhr und 20 Uhr größtenteils abgedeckt wurde. Die bereitgestellte Version der Visualisierung war zwar auf Windows-Computer ausgelegt, jedoch verwenden nicht alle potenziellen Benutzer einen Windows-Computer. Daher könnte dies einen nicht repräsentativen Aspekt darstellen, allerdings sollten die unterschiedlichen Betriebssysteme keine Auswirkung auf die Integration haben. Zwei der Probanden hatten keinen Windows-Computer und mussten daher auf eine virtuelle Maschine oder einen Remote-Computer zurückgreifen, was jedoch keinerlei Probleme verursachte.
- **Reaktivität:** Personen verändern ihr Verhalten, wenn sie merken, dass sie beobachtet werden. Dies wird als der Hawthorne-Effekt bezeichnet. Eine solche Verhaltensänderung ist möglicherweise nicht repräsentativ für das Verhalten der potenziellen Benutzer ohne Beobachtung (Hemmerich, 2022). Die Probanden hatten zwar die Möglichkeit, diese Studie komplett alleine durchzuführen, jedoch wurde ihnen empfohlen, die Studie im Beisein des Studieninitiators durchzuführen, wodurch ein solcher Effekt nicht ausgeschlossen werden kann. Allerdings konnten die Probanden aufgrund der Online-Durchführung der Studie in ihrer gewohnten Umgebung arbeiten. Ein solcher Effekt könnte lediglich Einfluss auf die benötigte Zeit haben, da nur die Aufgabenbearbeitung beobachtet wurde, um die Anonymität der Beantwortung der Fragebögen zu gewährleisten.

AUSBLICK UND FAZIT

Im Folgenden werden im Unterkapitel "[Ausblick](#)" einige Ideen für die Zukunft vorgestellt. Abschließend folgt ein zusammenfassendes [Fazit](#).

5.1 AUSBLICK

Im Nachfolgenden werden einige Ideen zur Erweiterung und Verbesserung der entwickelten sowie neue Features für die Zukunft beschrieben.

Zuerst werden [neue Drawable-Typen](#) vorgestellt, gefolgt von [Erweiterungen für den Text - Drawable-Typ](#) sowie [Verbesserungen für den Image - Drawable-Typ](#). Anschließend werden [neue Aktionen](#), ein [Konfigurationsmenü](#), [Erweiterungen für das Whiteboard](#), Ideen für die [Virtual Reality und Augmented Reality - Anbindung](#), [Verbesserung für den Drawable-Synchronizer](#) und [weitere Erweiterungen sowie Verbesserungen](#) vorgestellt.

5.1.1 Neue Drawable-Typen

In den folgenden Abschnitten werden einige Vorschläge für neue Drawable-Typen präsentiert, die implementiert werden könnten.

Die Drawable-Typen [UML-Diagramme](#), [Zustandsautomat](#) und die [Datenbankmodelle](#) würden, wie im Fall des [Mind Map-Drawable-Typ](#), eine Kombination aus den Drawable-Typen [Line](#) und [Text](#) bilden.

Für die Drawable-Typen [Web](#), [Video / GIF](#), [PDF](#) und [Audio](#) wäre es sinnvoll, die entsprechenden Drawable-Typ-Objekte, ähnlich wie den [Image-Drawable-Typ](#), auf ein Canvas mit Box-Collider zu platzieren. Durch das Canvas wird sichergestellt, dass der [Blink-Effekt](#) angewendet werden kann, während der Box-Collider für die [Transformations-Aktionen](#) benötigt wird.

Web-View: Durch eine Web-View wird der Zugriff auf einen Ingame-Webbrowser ermöglicht.

5.1.1.1 Web

Um das Evaluieren oder gemeinsame Anschauen einer Webseite zu unterstützen, könnte eine *Web-View* als neuer Drawable-Typ eingeführt werden.

Es gibt bereits einige Projekte, die eine Web-View für Unity bereitstellen. „3D WebView: the ultimate cross-platform web browser for Unity“ von Vuplex¹ ist z. B. ein solches Projekt.

Die Web-View von Vuplex soll neben das Anzeigen von Webseiten auch das Anschauen von Videos (insbesondere YouTube) und PDFs unterstützen.

Sollten die Lizenzen für dieses Asset jedoch zu kostspielig sein, gibt es durchaus günstigere Alternativen, die unter Umständen weniger Features bieten. Eine weitere Möglichkeit wäre die Entwicklung einer eigenen Web-View.

5.1.1.2 Video / GIF

Für kollaboratives Arbeiten und Verständnis kann es in einigen Situationen von Vorteil sein, gemeinsam ein Video oder ein animiertes Bild in Form eines GIFs anzuschauen.

Mit einem GIF können unter anderem z. B. Zustandsübergänge simuliert werden.

Um dieses Feature zu unterstützen, wäre es notwendig, einen Video-Player in SEE zu integrieren.

Dafür gibt es zahlreiche Assets im [Unity Asset Store](#)².

Alternativ kann ein solcher Video-Player auch selbst entwickelt werden. Im Internet finden sich zahlreiche Anleitungen dafür.

Aus Gründen der Performanz sollte darauf geachtet werden, dass die Videos und GIFs im Web gehostet und von den Usern von dort heruntergeladen beziehungsweise gestreamt werden. Dieses Vorgehen reduziert die Netzwerkdatenlast von SEE, da zu große Datenpakete, die über das Netzwerk von SEE verschickt werden, bei dem Sender zu kurzen bis längeren Verzögerungen führen können, abhängig von der Größe des Datenpakets.

Des Weiteren wäre es für GIFs ratsam, eine automatische Abspielung inklusiver Dauerschleifen-Funktion zu aktivieren, um die ursprüngliche Idee der animierten Bilder zu unterstützen.

¹ <https://developer.vuplex.com/webview/overview>

² <https://assetstore.unity.com/?q=video%20player&orderBy=1>

5.1.1.3 PDF

Für viele Bereiche, in denen Zusammenarbeit und Diskussionen eine Rolle spielen, wäre es sinnvoll, einen PDF-Viewer zu integrieren. Auch hierfür gibt es zahlreiche Assets im Unity Asset Store. Es wäre ratsam, auf eine Integration zu setzen, die das PDF tatsächlich rendert und nicht nur als Folge von Bildern bereitstellt. Ein weiteres nützliches Feature für den Viewer wäre die Möglichkeit, wichtige Textpassagen zu markieren und Notizen in das PDF zu schreiben. Zudem sollte dann die Möglichkeit bestehen, das bearbeitete PDF speichern zu können.

5.1.1.4 Audio

Ein weiteres nützliches Feature könnte das Aufnehmen sowie das Bereitstellen von Audiodateien sein. Dabei sollte der von SEE bereitgestellte Sprachkanal aufgenommen und in einer entsprechenden Audiodatei abgespeichert werden können. Zudem sollte es die Möglichkeit geben, Audiodateien mittels eines Audio-Players bereitzustellen. Der Audio-Player sollte den Audio-Titel, die aktuelle sowie maximale Laufzeit anzeigen und die gängigen Operationen (Play, Pause, Stop und Spulen) bereitstellen. Unter Umständen wäre auch die Option, die Geschwindigkeit der Audio-Datei zu ändern, hilfreich.

5.1.1.5 UML-Diagramme

OMG: Ist ein internationales, gemeinnütziges Konsortium, das sich auf die Entwicklung und Pflege von Technologiestandards konzentriert.

Unified Modeling Language (UML) ist eine visuelle Modellierungssprache, die von der *Object Management Group* (OMG) entwickelt und standardisiert wurde. Sie stellt verschiedene Diagramme und Notationen bereit, um die statischen und dynamischen Aspekte eines Projektes zu modellieren (Kecher u. a., 2018).

Die statischen Aspekte werden durch die Strukturdiagramm-Gruppe erfasst, während die dynamischen Aspekte durch die Verhaltensdiagramm-Gruppe dargestellt werden. Beide Gruppen umfassen derzeit jeweils sieben Diagramme, wobei die Verhaltensdiagramme noch die Untergruppe der Interaktionsdiagramme enthält (siehe [Abbildung 5.1](#)).

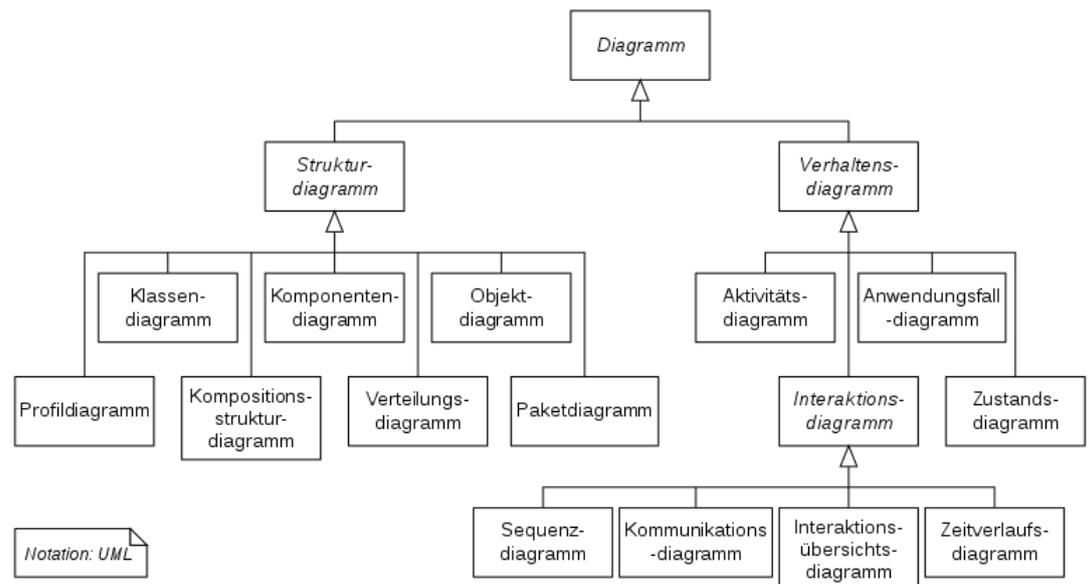


Abbildung 5.1: UML-Diagrammhierarchie von [Wikipedia](#)³.

³ <https://de.wikipedia.org/wiki/Datei:UML-Diagrammhierarchie.svg>

Gogolla u. a. (2005) entwickelten ein Tool namens *A UML-based Specification Environment* (USE). Mithilfe von USE ist es dem Anwender möglich, UML-Diagramme mittels einer textuellen Beschreibung, die als USE-Spezifikation bezeichnet wird, zu erstellen (siehe [Listing 5.1](#)). Diese USE-Spezifikation kann durch die Verwendung von *Object Constraint Language* (OCL) um Bedingungen und Einschränkungen erweitert werden (siehe [Listing 5.2](#)).

Darüber hinaus besteht die Möglichkeit, eine Simulation der Modellierung durchzuführen, um die Spezifikation zu validieren (Gogolla u. a., 2007) (siehe [Abbildung 5.2](#)).

OCL: Ist ein Bestandteil von UML, mit dem notwendige Randbedingungen festgelegt werden können, wie beispielsweise Invarianten für Klassendiagramme, Bedingungen in Sequenzdiagrammen oder Vor- und Nachbedingungen für Methoden.

```

model Company
-- classes
class Employee
attributes
  name : String
  salary : Integer
end

class Department
attributes
  name : String
  location : String
  budget : Integer
end

class Project
attributes
  name : String
  budget : Integer
end

-- associations

association WorksIn between
  Employee[*]
  Department[1..*]
end

association WorksOn between
  Employee[*]
  Project[*]
end

association Controls between
  Department[1]
  Project[*]
end

```

Listing 5.1: Ein Beispiel für eine USE-Spezifikation von [USE Quick Tour](#)⁴.

```

-- OCL constraints

constraints

context Department
  -- the number of employees working in a
  -- department must be greater or equal to
  -- the number of projects controlled
  -- by the department
  inv MoreEmployeesThanProjects :
    self.employee->size >= self.project->size

context Employee
  -- employees get a higher salary when they
  -- work on more projects
  inv MoreProjectsHigherSalary :
    Employee.allInstances ->forAll(e1, e2 |
      e1.project->size > e2.project->size
      implies e1.salary > e2.salary)

context Project
  -- the budget of a project must not exceed the
  -- budget of the controlling department
  inv BudgetWithinDepartmentBudget :
    self.budget <= self.department.budget

  -- employees working on a project must also
  -- work in the controlling department
  inv EmployeesInControllingDepartment :
    self.department.employee->
      includesAll(self.employee)

```

Listing 5.2: Ein Beispiel für eine USE-OCL Verwendung von [USE Quick Tour](https://www.db.informatik.uni-bremen.de/projects/USE-2.2.0/qt.html)⁴.

⁴ <https://www.db.informatik.uni-bremen.de/projects/USE-2.2.0/qt.html>

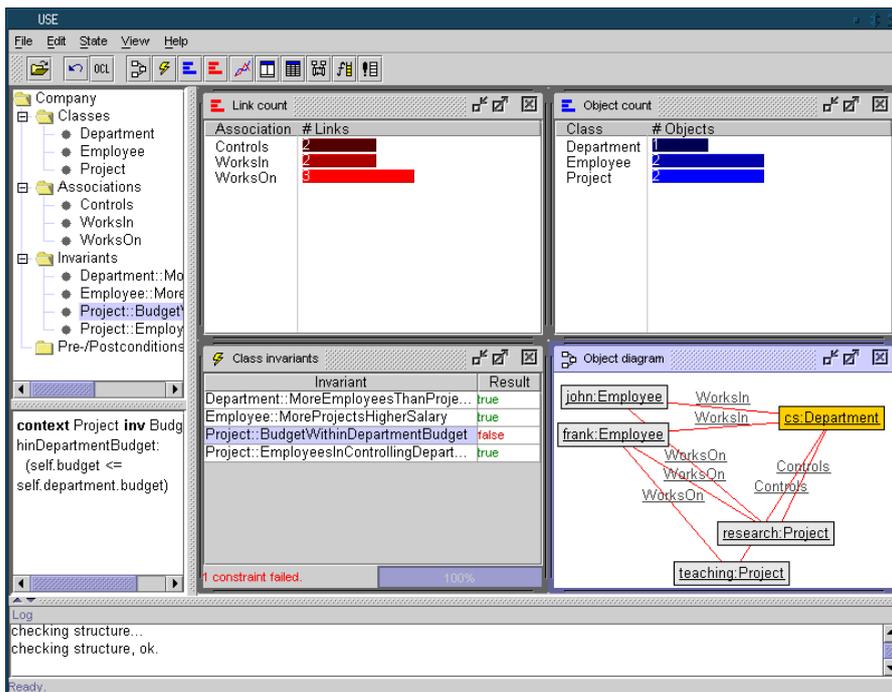


Abbildung 5.2: USE - Analyse von USE Quick Tour⁵.

Für die Integration in SEE würde sich neben einer grafischen Oberfläche, ähnlich wie bei dem [Mind Map-Drawable-Typ](#), auch eine textuelle Beschreibung wie die USE-Spezifikation anbieten.

Diese Spezifikationseingabe könnte über das *Code-Window* von SEE erfolgen. Ebenso könnte dieses Code-Window für die Integration von OCL verwendet werden. Wenn eine solche OCL-Integration eingebunden würde, böte sich ebenfalls eine Analyse ähnlich der in [Abbildung 5.2](#) gezeigten an.

Da die Integration aller vierzehn Diagramme viel Zeit in Anspruch nehmen würde, wäre es ratsam, den einzelnen Diagramme eine Relevanz zuzuordnen und sie dann in dieser Reihenfolge zu implementieren.

Code-Window: Ist ein Fenster, in dem Programmcode angezeigt, eingegeben und bearbeitet werden kann.

⁵ <https://www.db.informatik.uni-bremen.de/projects/USE-2.2.0/qt7.png>

5.1.1.6 Zustandsautomat

Eine weitere mögliche Integration wäre die des Zustandsautomaten, auch endlicher Automat genannt.

Als Darstellungsvariante bietet sich ein Übergangsdiagramm an.

In der [Abbildung 5.3](#) kann ein solches Übergangsdiagramm für einen deterministischen Automaten in Form eines Getränkeautomat-Beispiels betrachtet werden.

Da es *deterministische* und *nichtdeterministische* Automaten gibt, wäre es sinnvoll, zusätzlich einen Überprüfer zu implementieren, der überprüft, ob es sich bei dem dargestellten Automaten um einen deterministischen Automaten handelt. Wenn dies nicht der Fall ist, sollten nichtdeterministische Stellen markiert werden.

Eine Erweiterung dieser Integration könnte durch die Bereitstellung von *Turingmaschinen* erfolgen, wobei dann eine Eingabe für das Bandalphabet angeboten werden sollte.

Um die Integration abzurunden, könnte außerdem die Visualisierung eines Automatenlaufes implementiert werden.

deterministisch:

Wenn für jede Eingabe eine eindeutige Ausgabe vorhersehbar ist.

nichtdeterministisch:

Wenn für eine Eingabe verschiedene Ausgaben möglich sind. Die Ausgabe ist folglich nicht eindeutig vorhersehbar.

Turingmaschine: Ist ein Automatenmodell mit einem unendlich langen Band und einem Lese-/Schreibkopf als Speicher. Eine Eingabe für das Band muss jedoch endlich sein.

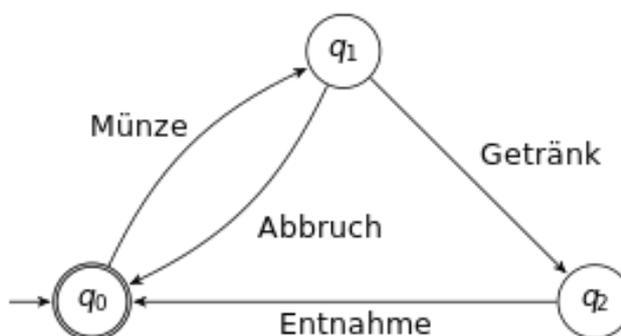


Abbildung 5.3: Getränkeautomat Beispiel für einen deterministischen Zustandsautomaten von [Wikipedia](#)⁶.

⁶ https://de.wikipedia.org/wiki/Datei:DFA_M%C3%BCnzen.svg

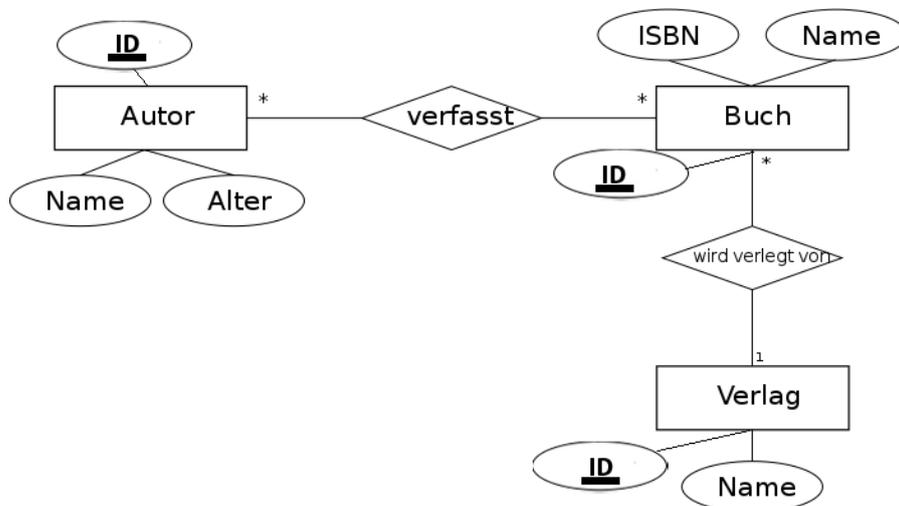
5.1.1.7 Datenbankmodelle

Als letzter möglicher Drawable-Typ werden zwei Modelle für die Visualisierung von Datenbanken vorgestellt. Zum einen das **Entity - Relationship - Modell** und zum anderen das **relationale Datenbankmodell**.

Das **Entity - Relationship - Modell** dient dazu, die Struktur einer Datenbank auf einer abstrakten Ebene zu modellieren.

Das Modell basiert auf drei Arten von Objekten, die *Entitäten*, die *Beziehungen* und die *Attribute* (Wikipedia, 2024a).

Die **Abbildung 5.4** liefert ein beispielhaftes, jedoch stark vereinfachtes, Modell für den Bereich des Buches.



Entität: Objekt der realen Welt, die in der Datenbank gespeichert werden sollen.

Beziehung: Beschreiben die Verbindung zwischen verschiedenen Entitäten.

Attribut: Ist eine Eigenschaft oder ein Merkmal, das einer Entität oder einer Beziehung zugeordnet ist.

Abbildung 5.4: Ausschnitt eines Beispiels für das ER-Modell von Wikipedia⁷.

⁷ <https://de.wikipedia.org/wiki/Datei:Er-diagramm.svg>

Relation: Ist eine Tabelle, die eine bestimmte Entität oder Beziehung repräsentiert.

Schlüssel: Es gibt zwei Arten von Schlüsseln in relationalen Datenbanken: den Primärschlüssel und den Fremdschlüssel. Jede Relation hat einen Primärschlüssel, der aus einem oder mehreren Attributen bestehen kann und jeden Tupel der Relation eindeutig identifiziert. Der Fremdschlüssel wird verwendet, um Beziehungen zwischen Relationen herzustellen, indem er auf den Primärschlüssel einer anderen Relation verweist.

Das **relationale Datenbankmodell** dient ebenfalls dazu, die Struktur einer Datenbank zu modellieren. Diese Modellart basiert auf *Relationen*, deren Attribute sowie deren *Schlüssel*. Für die Darstellung dieser Modellart wird auf das UML - Klassendiagramm zurückgegriffen (Schulz, n.d.).

Ein Beispiel kann in der [Abbildung 5.5](#) betrachtet werden.

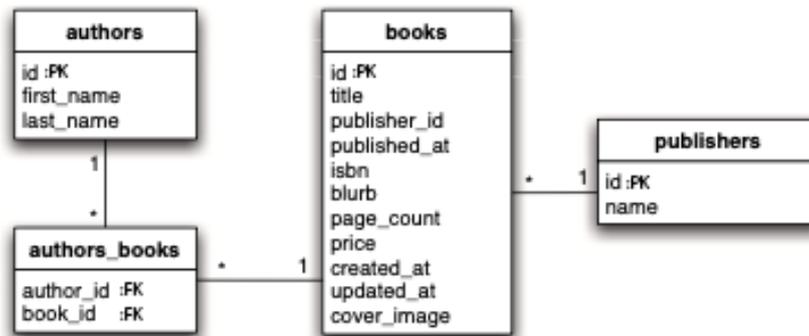


Abbildung 5.5: Ausschnitt eines Beispiels für das relationale Datenbankmodell von [Stackoverflow](#)⁸.

Es ist zu beachten, dass die originalen Bilder der beiden Abbildungen 5.4 und 5.5 angepasst wurden, um die Schlüssel mit einzuschließen.

⁸ <https://stackoverflow.com/questions/71301161/how-can-i-create-new-book-and-assign-two-authors-with-it-using-rails-minitest>

5.1.2 Erweiterungen für den Text-Drawable-Typ

In diesem Abschnitt werden einige Erweiterungen für den [Text-Drawable-Typ](#) vorgestellt.

Die Erweiterungen umfassen dabei einen [Farbverlauf](#), einen [Glow und Lightning - Effekt](#) sowie eine [Anbindung für den VR/AR - Bereich](#).

Für die Erweiterungen des Farbverlaufes sowie der Glow- und Lighting-Effekte sollte die [Color Picker-Aktion](#) so angepasst werden, dass auch die neu eingeführten Farben ausgewählt werden können. Hierfür könnte ein Auswahlmü ähnlich dem für die [Mind Map-Drawable-Typen](#) verwendet werden.

5.1.2.1 Farbverlauf

Ebenso wie für den [Line-Drawable-Typ](#) existiert auch für den [Text-Drawable-Typ](#) die Möglichkeit des Farbverlaufes.

Zur Erinnerung: Bei einem [Line-Drawable-Typ](#) wird der Farbverlauf auf die gesamte Linie angewendet, während beim Text der ausgewählte Farbverlauf auf jedes einzelne Zeichen angewendet wird.

Im weiteren Verlauf dieses Kapitels wird dies noch veranschaulicht.

Ab der TMP-Version 4.0.0-pre.2 werden Farbverläufe auch über den Rich-Text-Tag "`<gradient>`" bereitgestellt. Die aktuell verwendete Version in dieser Integration ist jedoch 3.0.6. Folglich müsst die Version aktualisiert werden, um die Bereitstellung zu unterstützen.

In der folgenden [Abbildung 5.6](#) wird eine solche Verwendung demonstriert.



Abbildung 5.6: Ein Beispiel für die Verwendung des Rich-Text-Tags "`<gradient>`" von der [TextMeshPro Unity Paket Dokumentation](#)⁹.

Da diese Verwendung nicht besonders benutzerfreundlich ist, wäre es ratsam, den Farbverlauf, ähnlich wie bei den [Line-Drawable-Typ](#), über ein Menü bereitzustellen.

⁹ <https://docs.unity3d.com/Packages/com.unity.textmeshpro@4.0/manual/RichTextGradient.html>

Für den Farbverlauf ist es wichtig zu wissen, dass eine weiße Textfarbe (Vertex- und Face-Color) die reine Farbverlaufsfarbe widerspiegelt (Unity_Technologies, 2023a).

Dabei gilt: Je heller die Farbe, desto deutlicher kommt die Farbverlaufsfarbe zur Geltung. Für die schwarze Textfarbe ist der Farbverlauf deshalb nicht sichtbar. Bei allen anderen Textfarben wird die Farbe mit der Farbverlaufsfarbe gemischt.

Um dieses Mischen der Farben zu verdeutlichen, wurde in dem Beispiel in der [Abbildung 5.7](#) eine blaue Textfarbe und eine rote Farbverlaufsfarbe gewählt, wodurch ein lila Farbton entsteht.



Abbildung 5.7: Ein Beispiel für die Vermischung von Farben mittels eines Farbverlaufs mit der Textfarbe blau und der Farbverlaufsfarbe rot.

Nach der Aktivierung der Funktionalität “Color Gradient“ stehen für den [Text-Drawable-Typ](#) vier verschiedene Farbverläufe zur Verfügung, die im Folgenden näher erläutert werden:

- **SINGLE:**

Dieser Farbverlauf überlagert den Text vollständig mit der gewählten Farbverlaufsfarbe.

In der oben gezeigten [Abbildung 5.7](#) wurde bereits der Single-Farbverlauf demonstriert.

Wie dieser Farbverlauf auf einem hellgrauen Text aussehen würde, kann in der nachfolgenden [Abbildung 5.8](#) betrachtet werden.



Abbildung 5.8: Anwendung des Single-Farbverlaufes mit der Verlaufsfarbe rot auf einem hellgrauen Text.

- **VERTIKAL:**

Bei dieser Art des Farbverlaufs können zwei Farben gewählt werden, die sich vertikal mit der Textfarbe vermischen.

In der folgenden [Abbildung 5.9](#) kann ein Beispiel hierfür betrachtet werden. Für das Beispiel wurde als obere Farbverlaufsfarbe gelb und als untere rot gewählt.



Abbildung 5.9: Anwendung des Vertikal-Farbverlaufes mit der Verlaufsfarbe gelb für oben und rot für unten auf einem hellgrauen Text.

- **HORIZONTAL:**

Auch bei dieser Art des Farbverlaufs können zwei Farben gewählt werden, die sich horizontal mit der Textfarbe vermischen.

Die Demonstration hierfür kann in der [Abbildung 5.10](#) begutachtet werden. Hierfür wurde als linke Farbe gelb und als rechte rot gewählt.



Abbildung 5.10: Anwendung des Horizontal-Farbverlaufes mit der Verlaufsfarbe gelb für links und rot für rechts auf einem hellgrauen Text.

- **VIER ECKEN (FOUR CORNERS):**

Mit dieser Art des Farbverlaufs können vier Farben gewählt werden, die sich dann von den jeweiligen Ecken mit der Textfarbe vermischen. [Abbildung 5.11](#) zeigt ein Beispiel hierfür, mit den gewählten Farben:

oben links: gelb	oben rechts: magenta
unten links: hellblau/türkis	unten rechts: rot



Abbildung 5.11: Anwendung des Vier-Ecken-Farbverlaufes mit der Verlaufsfarbe gelb für oben links, magenta für oben rechts, hellblau/türkis für unten links und für unten rechts rot auf einem hellgrauen Text.

5.1.2.2 *Glow und Lighting*

Um die Effekte Glow und Lighting zu unterstützen, müsste der Shader des *Drawable-Fonts* von "TextMeshPro/Mobile/Distance Field" auf "TextMeshPro/Distance Field" geändert werden.

Dies hat den Vorteil, dass die Texte durch den neuen Shader besser aussehen, wie in der nachfolgenden [Abbildung 5.12](#) veranschaulicht wird.



Abbildung 5.12: Vergleich der beiden Shader-Varianten. Links ist der alte und rechts der neue Shader.

Allerdings entsteht dabei das Problem, dass sich die *Outline-Color* mit dem neu gewählten Shader nicht mehr deaktivieren lässt, da dieser die Option zum Ein- und Ausschalten nicht unterstützt.

Dafür gibt es zwei Lösungsansätze:

Entweder wird das Deaktivieren der Outline-Color so umgebaut, dass der *Alpha-Wert* der Farbe auf 0 gesetzt wird. Das hätte zur Folge, dass der alte Alpha-Wert gespeichert werden müsste.

Alternativ könnte ein eigener Shader implementiert werden, der den neugewählten Shader um die Funktionalität des Ein- und Ausschaltens der Outline-Color erweitert. Es könnte natürlich auch der bisherige Shader um die Glow- und Lighting-Option erweitert werden.

Durch die Abbildungen [5.13](#) und [5.14](#) wird der Unterschied zwischen Alpha-Wert von 0 oder Outline-Color-Thickness von 0 für den Outline-Color-Effekt veranschaulicht.

Es ist zu erkennen, dass ein Alpha-Wert von 0 denselben Effekt aufweist wie das Ausschalten des Outline-Color-Effekts. Wohingegen das Setzen des Thickness-Wertes auf 0 den Text dicker erscheinen lässt.

Drawable-Font: Ist das Font Asset, das nicht nur die Schriftart selbst, sondern auch verschiedene Einstellungen wie beispielsweise die Schriftfarbe enthält. Je nachdem, welcher Shader angegeben wird, werden zusätzliche Eigenschaften hinzugefügt.

Outline-Color: Beschreibt in diesem Kontext die Außenfarbe beziehungsweise Umrandungsfarbe des Textes.

Alpha-Wert: Gibt die Transparenz einer Farbe an.



Abbildung 5.13: Vergleich zwischen Outline-Color ausgeschaltet und Alpha-Wert auf 0.

Resultat: Gleiches Ergebnis.



Abbildung 5.14: Vergleich zwischen Outline-Color ausgeschaltet und Outline-Color-Thickness(Dicke) auf 0.

Resultat: Thickness auf 0 lässt den Text dicker erscheinen.

Der Glow-Effekt lässt die Zeichen des Textes nach innen hin leuchten. Ähnlich wie bei dem Farbverlauf wird der Effekt umso deutlicher, je heller die Textfarbe ist. Eine schwarze Textfarbe leuchtet nicht. In der [Abbildung 5.15](#) kann die Anwendung eines solchen Glow-Effekts begutachtet werden.



Abbildung 5.15: Demonstrationsbeispiel für den Glow-Effekt.

Der Lighting-Effekt bietet eine Vielzahl an Möglichkeiten, um den Text anders darzustellen. So kann dem Text beispielsweise ein metallisches Aussehen verliehen werden.

Weitere Konfigurationsmöglichkeiten können in der [Unity Dokumentation zu dem Distance Field - Shader](#)¹⁰ nachgeschlagen werden.

Eine Demonstration für ein einfaches Beispiel kann in der [Abbildung 5.16](#) betrachtet werden.

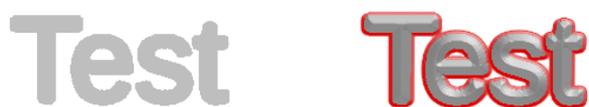


Abbildung 5.16: Demonstrationsbeispiel für den Lighting-Effekt.

Abschließend zeigt die [Abbildung 5.17](#) ein Beispiel für eine Kombination aus dem Glow- und Lighting-Effekt.

¹⁰ <https://docs.unity3d.com/Packages/com.unity.textmeshpro@3.2/manual/ShadersDistanceField.html#Lighting>



Abbildung 5.17: Demonstrationsbeispiel für eine Kombination aus dem Glow- und Lighting-Effekt.

5.1.2.3 *Unterschiedliche Schriftarten (Fonts)*

Es wäre interessant, auch die Möglichkeit anzubieten, verschiedene Schriftarten (Fonts) zur Auswahl zu stellen. Derzeit wird lediglich die standard Schriftart "Liberation Sans" des TMP unterstützt. Eine Darstellung verschiedener Schriftarten ist in der [Abbildung 5.18](#) zu sehen. Für dieses Beispiel wurde das Font-Asset "freshmaker SDF" von SEE verwendet.



Abbildung 5.18: Demonstrationsbeispiel für unterschiedliche Schriftarten.

5.1.2.4 *Text mit VR/AR*

Da der [Text-Drawable-Typ](#) für die Eingabe eine Tastatur erfordert und nicht jeder Benutzer bei der Nutzung eines Virtual Reality- oder Augmented Reality-Systems eine physische Tastatur zur Hand hat, wäre es sinnvoll, eine virtuelle Tastatur oder eine Spracheingabe anzubieten.

5.1.3 Verbesserungen für den Image-Drawable-Typ

Nachfolgend werden zwei Verbesserungen für den [Image-Drawable-Typ](#) beschrieben:

- **SENDEN DER BILDER ÜBER DAS NETZWERK:**

Momentan können kurzzeitige Verzögerungen beim Hinzufügen von [Image-Drawable-Typ](#)-Objekten auftreten. Dies liegt daran, dass die Bilddateien beim Hinzufügen des Bildes über das Netzwerk verschickt werden. Dies geschieht sowohl für den Fall eines Webbildes als auch für ein lokales Bild.

Eine verbesserte Variante für lokale Bilder wäre das Hochladen der Bilder auf einen separaten Server, von dem sie dann verteilt werden. Dadurch könnte auch ein weiteres aktuelles Problem gelöst werden. Wenn der Benutzer A ein Bild überträgt, jedoch Benutzer B bereits ein Bild mit diesem Namen im entsprechenden Ordner besitzt, welches nicht dasselbe ist, wird das Bild umbenannt. Wenn Benutzer B den Stand des Drawable speichert und später wieder lädt, könnte das umbenannte Bild gegebenenfalls bei Benutzer A als nicht vorhanden angesehen und somit erstellt werden. Dadurch hätte Benutzer A zweimal das gleiche Bild mit unterschiedlichen Namen.

Im Falle von Webbildern wäre es sinnvoller, die URL zu übertragen, damit die *Clients* die Bilddatei direkt von dem ursprünglichen Server herunterladen können.

Alternativ müsste überprüft werden, ob eine asynchrone Netzwerkübermittlung das Problem lösen könnte.

Dieses Verzögerungsproblem betrifft insbesondere den [Drawable-Synchronizer](#). Die jeweiligen Verzögerungen verhalten sich proportional zur Anzahl an [Image-Drawable-Typ](#)-Objekten in der Visualisierung.

Daher sollte in Erwägung gezogen werden, die [Image-Drawable-Typ](#)-Objekte vorübergehend aus dem [Drawable-Synchronizer](#) zu entfernen, falls keine Optimierung möglich ist. Dies wäre jedoch nur der Fall, falls intensiv mit [Image-Drawable-Typ](#) gearbeitet wird.

- **KOLLISIONSERKENNUNG BEIM HINZUFÜGEN:**

Derzeit besteht die Möglichkeit, dass ein [Image-Drawable-Typ](#)-Objekt beim Hinzufügen über den Rand des Drawable hinausragt. Daher wäre es empfehlenswert, beim Hinzufügen die [Kollisionserkennung](#) hinzuzufügen. Diese muss nach dem erfolgreichen Hinzufügen und der Auflösung einer eventuellen Kollision entfernt werden, da sie sonst unnötig Ressourcen verbraucht.

Es muss beachtet werden, dass auch im Redo-Fall die [Kollisionserkennung](#) aktiviert werden muss.

Client: Beschreibt in diesem Fall die ausgeführte eigene SEE-Instanz.

5.1.4 Neue Aktionen

Im Folgenden werden einige neue Aktionen vorgestellt, die zukünftig implementiert werden können, um die Drawable-Integration abzurunden.

5.1.4.1 Add Point to Line - Linienpunkt hinzufügen

In einigen Situationen kann es nützlich sein, einen neuen Punkt in einer Linie hinzuzufügen. Aus diesem Grund wäre es sinnvoll, eine Aktion zu implementieren, die das Hinzufügen von Punkten unterstützt. Die nachfolgende [Abbildung 5.19](#) demonstriert diese Aktion, indem an der rot markierten Stelle ein Punkt hinzugefügt wird. In diesem Beispiel wurde der hinzugefügte Punkt anschließend mit der [Move a Point-Aktion](#) verschoben.

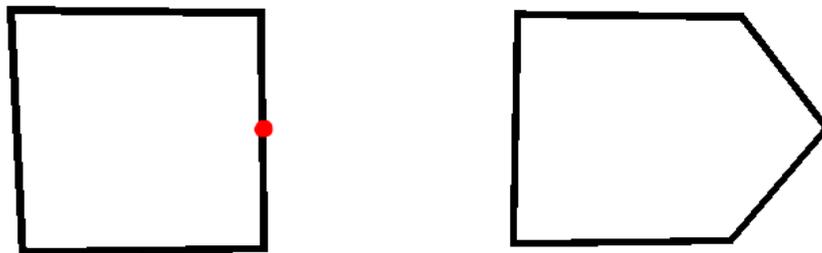


Abbildung 5.19: Demonstrationsbeispiel für Add Point to Line.

5.1.4.2 Mind Map Überlappungsproblemlöser

Das unachtsame Hinzufügen von neuen Knoten in einer Mind Map kann schnell zu Überlappungen von Knoten oder Verbindungslinien führen.

Um dies zu vermeiden, wäre es hilfreich, eine Funktionalität zu implementieren, die solche Überlappungen mithilfe von [Kollisionserkennung](#) erkennt und beseitigt.

In der nachfolgenden [Abbildung 5.20](#) wird veranschaulicht, wie diese Funktionalität zur Vermeidung von Überlappungen beispielsweise arbeiten könnte.

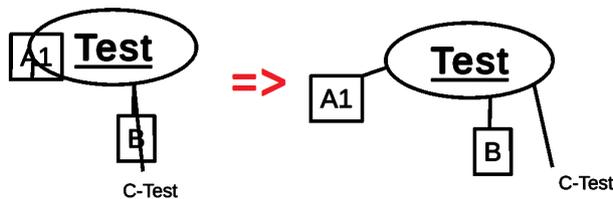


Abbildung 5.20: Funktionalität des Überlappungsproblemlösers.

5.1.4.3 Mind Map Sortierer

Da Mind Maps bei schlechter Sortierung schnell unübersichtlich oder unanschaulich werden können, wäre es sinnvoll, eine Funktionalität zu implementieren, die das Sortieren von Mind Maps bereitstellt.

Neben der vollständigen Sortierung sollte es auch möglich sein, nur ein Teil der Mind Map zu sortieren. Dazu soll der Benutzer einen Subthemen-Knoten auswählen können, dessen Kinder dann entsprechend sortiert werden sollen.

Für die Sortierung können zwei verschiedene Varianten in Betracht gezogen werden:

(Die Resultate der beiden Varianten beziehen sich auf die unsortierte Mind Map der [Abbildung 5.21](#))

- **RUND-UM-SORTIERUNG:**
Die Sortierung erfolgt rund um den ausgewählten Knoten. Ein mögliches Resultat dieser Variante kann in der [Abbildung 5.22](#)) betrachtet werden.
- **BAUM-SORTIERUNG:**
Die Sortierung erfolgt hierarchisch baumförmig von dem gewählten Knoten aus. Ein mögliches Resultat dieser Variante kann in der [Abbildung 5.23](#)) betrachtet werden.

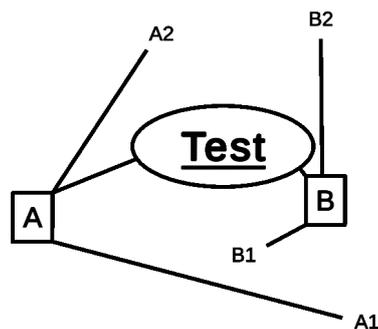


Abbildung 5.21: unsortiert

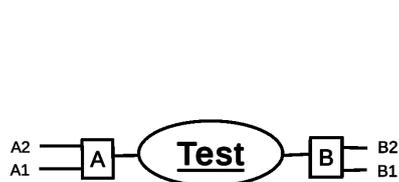


Abbildung 5.22: Rund-um-Sortierung

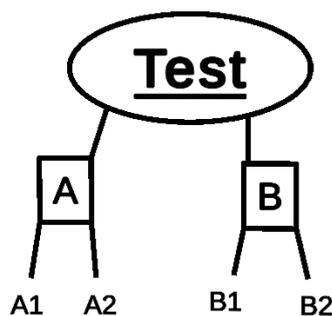


Abbildung 5.23: Baum-Sortierung

5.1.4.4 *Order Reducer - Ebenenreduzierer*

Aufgrund der Tatsache, dass jeder hinzugefügte Drawable-Typ die Ebenenreihenfolge ("Order in Layer") erhöht, kann dies schnell zu hohen Ebenenzahlen führen.

Um den Überblick über die Ebenen zu bewahren und die Benutzerfreundlichkeit zu gewährleisten, wäre es sinnvoll, einen Reduzierer zu implementieren. Dieser Reduzierer soll nicht verwendete Reihenfolgezahlen identifizieren und die Drawable-Typen entsprechend auf diese freien Reihenfolgezahlen anpassen. Angenommen es gibt drei Drawable-Typen mit den Reihenfolgen 6, 7 sowie 9 und der Reduzierer identifiziert die Reihenfolgen 2, 3 und 4 als frei, dann sollen die Drawable-Typen auf diese identifizierten freien Plätze angepasst werden. Zudem wäre es ratsam, den Reduzierer jedes Mal beim Speichern aufzurufen, um sicherzustellen, dass die gespeicherten Dateien konsistent die minimale Reihenfolge enthalten.

5.1.5 Konfigurationsmenü

Es wäre ratsam, ein Menü zur Konfiguration der verschiedenen Einstellungen des [ValueHolder](#) bereitzustellen.

Dadurch kann der Benutzer beispielsweise selbst die Geschwindigkeiten für normales oder schnelles Bewegen der Objekte wählen. Gleiches gilt für die restlichen Werte, wie für das Rotieren, Skalieren, die Anzeigedauer des Splitpunktes etc.

Zudem wäre es hilfreich, über das Menü gezielt einzelne Benachrichtigungen deaktivieren zu können, wie beispielsweise die Benachrichtigung der geänderten Ebenenreihenfolge bei der [Change The Sorting Layer-Aktion](#).

5.1.6 Erweiterungen für das Whiteboard

In diesem Abschnitt werden zwei Erweiterungen für das Whiteboard erläutert. Zum einen die Erweiterung um [Seiten](#) und zum anderen die Funktionalität des [Farbwechsels](#).

5.1.6.1 *Seiten*

Eine mögliche Erweiterung für das Whiteboard bietet die Funktionalität des Seitenwechsels. Dabei wird das Whiteboard, wie in der [Abbildung 5.24](#) zu sehen, um zwei Buttons und ein Eingabefeld erweitert. Die Buttons dienen der Navigation für die vorherige beziehungsweise nächste Seite. Über das Eingabefeld soll die gewünschte Seite direkt angesteuert werden können.

Um dieses Feature zu realisieren, gibt es zwei Herangehensweisen:

- Den jeweiligen Drawable-Typ-Objekten könnte eine Seitenzahl zugewiesen werden und dürften nur sichtbar sein, wenn die entsprechende Seite ausgewählt ist. Anderenfalls müssen sie ausgeblendet werden. Dafür würde jedes Drawable-Typ-Objekt eine Komponente mit der entsprechenden Seitenzahl zugewiesen werden. Bei einem Seitenwechsel würde dann jedes Objekt einzeln ein- oder ausgeblendet werden.
- Es könnten verschiedene [AttachedObject-Objekte](#) eingebunden werden, die jeweils die Objekte für eine bestimmte Seite enthalten. Die

AttachedObject-Objekte der nicht ausgewählten Seite würden dann ausgeblendet werden.



Abbildung 5.24: Seiten-Funktionalität für das Whiteboard.

5.1.6.2 Farbe ändern

In bestimmten Situationen kann es nützlich sein, anstelle eines weiß-grauen Hintergrundes eine andere Farbe für das Whiteboard zu wählen, wenn z. B. ein helles Bild eingefügt oder helle Linien gezeichnet werden sollen. Dafür wäre es hilfreich, wenn die Hintergrundfarbe ähnlich wie bei den [Sticky-Notes](#) geändert werden könnte.

5.1.7 Virtual Reality und Augmented Reality - Anbindung

Ein interessantes Feature könnte darin bestehen, Objekte wie Stifte zum Zeichnen in die Hand zu nehmen und dann damit die [Draw Freehand-Aktion](#) auszuführen.

Zudem könnte ein Radierer verwendet werden, um die [Erase-Aktion](#) durchzuführen.

Für diese Funktion wäre es sinnvoll, wenn der Benutzer die entsprechenden Objekte selbst spawnen kann. Bei einem Stift könnte eine zusätzliche Farbauswahl und Einstellung der Stiftstärke angeboten werden.

Ein potenzielles Problem bei dieser Funktion könnte eine zu realistische Implementierung sein. Wenn beispielsweise erst gezeichnet werden soll, wenn die Stiftspitze das Drawable berührt, also durch eine Kollision, dann muss der Benutzer zu nah vor dem Drawable stehen und verliert schnell den Überblick. Hinzu kommt, dass ein zusätzliches Objekt mitten im Bild, wie der Stift, die auszuführende Aktion behindern kann.

5.1.8 Verbesserung für den Drawable-Synchronizer

Wie im Abschnitt des [Drawable-Synchronizer](#) beschrieben, ist das aktuelle Vorgehen der Synchronisation nicht effizient. Die optimale Lösung für das Synchronisieren wäre, dass die auszuführenden Netzwerkaktionen zum Erstellen der Sticky-Notes und der Drawable-Typen nur bei den neubeigetretenen Benutzern ausgeführt werden. Dazu müsste eine Möglichkeit geschaffen werden, Netzwerkaktionen gezielt bei einem Benutzer auszuführen. Zudem sollten für den Synchronizer die im Abschnitt "[Verbesserungen für den Image-Drawable-Typ](#)" beschriebenen Änderung für das Senden der Bilder über das Netzwerk vorgenommen werden.

5.1.9 Weitere Erweiterungen sowie Verbesserungen

In diesem abschließenden Abschnitt werden weitere Erweiterungen und Verbesserungen vorgestellt, die in keine der vorherigen Kategorien fallen.

- **BILD-/VIDEOAUFNAHME:**

Es sollte eine Möglichkeit geschaffen werden, um Bilder oder sogar Videos von den Drawables aufzunehmen. Hierbei könnte eine VR-Anbindung in Form einer Kamera bereitgestellt werden, die von dem Benutzer in die Hand genommen werden kann, um dann mit ihr Fotos oder Videos aufzunehmen.

Zudem wäre es sinnvoll, eine Screenshottaste zu integrieren, wie es bei anderen Plattformen üblich ist. Ein Beispiel hierfür die Steam-Plattform, bei der die Taste "F12" zum Screenshot-Machen verwendet wird. Dies würde den Benutzern eine schnelle und zusätzliche Möglichkeit bieten, ihre Darstellungen festzuhalten und zu teilen.

- **AKTION-VERLASSEN-DIALOG:**

Derzeit ist es so, dass beim Verlassen einer Aktion die nicht abgeschlossenen Änderungen, die in der jeweiligen Aktion vorgenommen wurden, verworfen werden.

Um die Benutzerfreundlichkeit in dieser Hinsicht weiter zu verbessern, wäre es sinnvoll, vor dem Verlassen einen Dialog einzubinden, der abfragt, ob der Benutzer die Aktion wirklich verlassen möchte und auf die nicht abgeschlossenen Änderungen hinweist.

Alternativ könnte stattdessen erfragt werden, ob die Aktion abgeschlossen werden soll.

Die Machbarkeit müsste für beide Varianten überprüft werden. Die erste Variante wäre ohne weiteres nicht möglich, da die Aktionen, wie im Kapitel "[Forschungsfrage und Struktur](#)" beschrieben, einem bestimmten Ablauf folgen und nicht zurückspringen können. Für die zweite Variante müsste überprüft werden, ob in der "Stop"-Methode einer Aktion auf eine Benutzereingabe gewartet werden kann.

- **HAND AUSBLENDEN:**

Sowohl in der Pilotstudie als auch in der regulären Studie ist aufgefallen, dass die Anwesenheit der Hand die Ausführung von Aktionen stören kann.

Daher wäre es sinnvoll, einen Shortcut für das Ein- und Ausblenden der Hand einzuführen. Beim Ausblenden ist jedoch zu beachten, dass

der Laserpointer weiterhin sichtbar bleiben muss, da er essenziell für die Interaktion mit dem Drawable ist.

- **BENACHRICHTIGUNGEN (NOTIFICATIONS):**

Derzeit ist es so, dass Benachrichtigungen als Stack verarbeitet werden. Das bedeutet, dass die neusten Benachrichtigungen zuerst verarbeitet werden, wodurch sie als Erstes auslaufen und die älteren pausiert werden.

Dieses Vorgehen erweist sich für die benötigten Benachrichtigungen in dieser Integration als nicht förderlich und kann unter Umständen sogar störend sein. Ein Beispiel hierfür ist das Ändern der Ebenenreihenfolge über [Change The Sorting Layer-Aktion](#), bei dem viele Benachrichtigungen auftreten können.

Eine hilfreiche Verbesserung wäre, dass die Benachrichtigungen unabhängig voneinander verarbeitet und auslaufen können, ohne dass sie pausiert werden.

- **HILFSLINIEN:**

Ein weiteres nützliches Feature könnte das Einblenden von Hilfslinien sein, ähnlich einem Koordinatensystem, wie es beispielsweise in Photoshop verwendet wird. Ein Beispiel für solche Hilfslinien ist in der [Abbildung 5.25](#) dargestellt.

Diese Hilfslinien könnten den Benutzer dabei unterstützen, die Objekte beispielsweise mittig, auf derselben Höhe oder Breite wie ein anderes Objekt zu positionieren, insbesondere bei Aktionen wie der Move-Aktion.

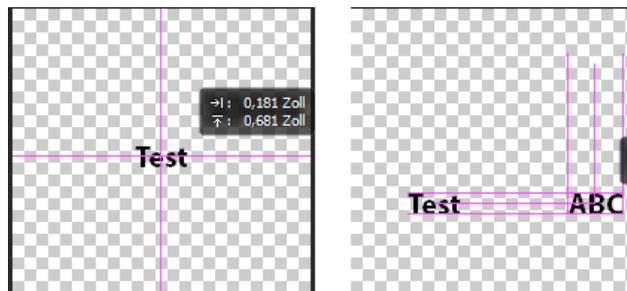


Abbildung 5.25: Beispiel für Hilfslinien.

- **ALTE POSITIONEN WIEDERHERSTELLEN:**

Die Aktionen [Scale](#) und [Rotate](#) können dazu führen, dass das zu transformierende Drawable-Typ-Objekt durch eine Kollision mit dem Rand eines Drawables verschoben wird.

Wenn nun auf die abgeschlossene Aktion ein Undo ausgeführt wird, wird derzeit die alte Position nicht wiederhergestellt. Dies liegt daran, dass ein darauf folgendes Redo die [Kollisionserkennung](#) für das ausgewählte Objekt aktivieren müsste, um eine Kollision zu vermeiden.

Eine mögliche Verbesserung wäre, dass die alte Position wie gerade beschrieben wiederhergestellt wird und dann für ein Redo kurzzeitig die [Kollisionserkennung](#) aktiviert wird.

- **MEHRERE DRAWABLE-TYPEN SELEKTIEREN:**

Ein weiteres nützliches Feature für die [Transformations-Aktionen](#) wäre die Möglichkeit, mehrere Drawable-Typ-Objekte für die ausgewählte Aktion zu selektieren.

Um dies zu ermöglichen, könnten die ausgewählten Objekte, ähnlich wie bei der “include Children“-Option der Move-Aktion für Mind Map Knoten, einem übergeordneten Objekt zugewiesen werden. Dieses übergeordnete Objekt würde dann für die jeweilige Aktion verwendet werden. Beim Trennen in die ursprüngliche Objekthierarchie müsste darauf geachtet werden, dass die Änderungen beibehalten werden.

- **DRAWABLE-MENÜ-UNTERTEILUNG:**
Das aktuelle Drawable-Menü enthält eine Vielzahl von Aktionen und ist daher recht voll.
Es wäre hilfreich, eine Möglichkeit zu schaffen, das Menü in zusätzliche Untermenüs wie “Hinzufüge-Aktionen“, “Transformations-Aktionen“, “Lösch-Aktionen“, etc. zu unterteilen.
- **KREIS-LADEBALKEN (LOADINGBAR):**
In der **Move-Aktion** kann es zu kurzen Wartezeiten beim Abschließen der Aktion kommen, wenn noch eine Kollision verarbeitet werden muss. In solchen Fällen wäre es hilfreich, diesen Zustand durch einen Kreis-Ladebalken anzuzeigen.
- **SPRACHPAKETE:**
Ein Proband äußerte den Wunsch nach Sprachpaketen für SEE.
Eine mögliche Umsetzung hierfür könnte der Umbau auf *Lokalisierung* sein. Dabei würden sämtliche englischen Texte im Quellcode durch *Lokalisierungs-Schlüssel* ersetzt. Diese Lokalisierungs-Schlüssel würden dann auf ein angelegtes Wörterbuch der jeweils gewählten Sprache verweisen.
Zudem muss der Benutzer durch ein bereitgestelltes Menü die Sprache wählen können.

Lokalisierung: Ist die Möglichkeit, die Software in Bezug auf die Sprache an verschiedene Regionen oder Länder anzupassen.

Lokalisierungs-Schlüssel: Ist der Begriff, der verwendet wird, um den gewünschten Text in einem Wörterbuch oder einer Ressourcendatei zu identifizieren.

5.2 FAZIT

Die Arbeit wurde mit dem **ersten Kapitel** eingeleitet, welches die Motivation, die benötigten Grundlagen und die grundlegende Forschungsfrage zur Benutzerfreundlichkeit der Integration behandelt. Im **zweiten Kapitel** wurden die Anforderungen und Ziele für die Integration definiert. Außerdem wurde ein verwandtes Projekt vorgestellt und begründet, warum dies nicht als Grundlage dienen konnte. Anschließend wurden diese Anforderungen und Ziele im **dritten Kapitel** umgesetzt. Dafür wurden zwei Arten von Drawables bereitgestellt. Dies sind zum einen das Whiteboard und zum anderen die Sticky-Notes. Des Weiteren wurden Drawable-Typen zum Zeichnen von Linien, zum Schreiben von Texten, zum Darstellen von Bildern und zum Erstellen von Mind-Maps bereitgestellt. All diese Darstellungen sind hoch konfigurierbar und können über die vielen bereitgestellten **Transformations-Aktionen** transformiert werden. Zusätzlich können sie durch die **Lösch-Aktionen** gelöscht, über die **Save-Aktion** gespeichert und mithilfe der **Load-Aktion** wiederhergestellt werden. Außerdem können mittels der **Sticky-Note-Aktion** weitere Drawables während der Laufzeit bereitgestellt werden. Zudem können diese Sticky-Notes editiert, bewegt und gelöscht werden. Im **vierten Kapitel** wurde die Evaluation der Integration beschrieben, wobei die grundlegende Forschungsfrage in die drei folgenden Forschungsfragen unterteilt wurde:

1. Welche Benutzerfreundlichkeit bietet die Integration insgesamt?

2. Bewerten Informatiker die Benutzerfreundlichkeit besser und führen sie gestellte Aufgaben schneller und mit geringerer Anstrengung durch als Nicht-Informatiker?
3. Wie verändert sich die Wahrnehmung der Benutzerfreundlichkeit, die benötigte Zeit und wahrgenommene Anstrengung bei erneuter Verwendung der Integration?

Um diese Forschungsfragen zu untersuchen, wurde eine empirische Studie mit vierzehn Probanden durchgeführt. Diese absolvierten sieben verschiedene Aufgaben, um alle Aktionen der Integration abzudecken. Während der Studie wurden diverse unabhängige Variablen zur Person des Probanden erfragt und abhängige Variablen in Form der Beantwortung mehrerer ASQ-Fragebögen, eines SUS-Fragebogens und die benötigte Zeit für die Aufgabenbewältigung erhoben.

Für die erste Forschungsfrage wurde festgestellt, dass die Integration mindestens eine exzellente Gesamtbenutzerfreundlichkeit anhand des ermittelten SUS aufweist. Des Weiteren wurde ermittelt, dass die Probanden durchschnittlich $\bar{X} \approx 39,37$ Minuten zum Bearbeiten der Aufgaben benötigen und dass die Zustimmung mit der Gesamtkomplexität auf eine sehr starke bis starke Zufriedenheit hindeutet. Zudem wurde eine starke Zufriedenheit mit dem empfundenen Aufwand für die Aufgabenbewältigung und eine sehr starke bis starke Zufriedenheit mit den angezeigten Informationen festgestellt. Daher konnte die erste Forschungsfrage beantwortet werden mit: Die Integration weist laut den gemessenen Ergebnissen der Probanden und gemäß der Tabelle von [Sauro \(2018\)](#) mindestens eine exzellente Benutzerfreundlichkeit auf.

Für die zweite Forschungsfrage wurden die Informatiker und die Nicht-Informatiker miteinander verglichen. Dieser Vergleich wurde auf signifikante Unterschiede mittels des Mann-Whitney-U-Tests geprüft. Zunächst wurden signifikante Unterschiede in der empfundenen Komplexität sowie des Aufwandes festgestellt. Der Unterschied der benötigten Zeit war dabei fast signifikant. Informatiker konnten die gestellten Aufgaben im Durchschnitt 22,61% schneller absolvieren als Nicht-Informatiker. Daher kann die zweite Forschungsfrage beantwortet werden mit: Während des ersten Durchlaufes bewerten Informatiker die empfundene Komplexität und den aufzubringenden Aufwand signifikant geringer als Nicht-Informatiker. Zudem sind sie fast signifikant schneller in der Aufgabenbewältigung. Die Bewertung der Benutzerfreundlichkeit unterscheidet sich dabei nicht signifikant.

Zur Beantwortung der dritten Forschungsfrage wurde ein zweiter Studierendurchlauf mit sieben Probanden durchgeführt. Alle sieben Probanden waren Nicht-Informatiker. Die Probanden führten exakt dieselbe Studie mit denselben Aufgaben durch. Die verwendeten Tools und das zu untersuchende Objekt blieben dabei gleich. Es wurden zwischen dem ersten und zweiten Durchlauf signifikante Unterschiede für die abhängigen Variablen SUS, die drei ASQ-Daten und die benötigte Zeit festgestellt. Für diese Überprüfung wurde der Wilcoxon-Vorzeichen-Rang-Test verwendet. Diese deutliche Verbesserung zwischen dem ersten und zweiten Durchlauf zeigt, dass es generell für Benutzerfreundlichkeitstudien lohnenswert wäre, Messungen zu wiederholen. Dies würde auch einen höheren Realismus widerspiegeln, da Menschen neue Werkzeuge zuerst kennenlernen müssen, bevor sie sicher damit umgehen können. Aus diesem Grund konnte die dritte Forschungsfrage beantwortet werden mit: Bei einer erneuten Verwendung der Integration werden die Wahrnehmung der Benutzerfreundlichkeit, die benötigte Zeit sowie die empfundene Anstrengung in Form der Komplexität der Aufgaben und des empfundenen

Aufwands signifikant besser eingeschätzt. Ebenfalls steigt die Zufriedenheit mit den angezeigten Informationen signifikant.

Anschließend wurden die Daten des zweiten Durchlaufes mit den Daten der Informatiker des ersten Durchlaufes überprüft, ob die zuvor genannten signifikanten Unterschiede bestehen, wenn die Nicht-Informatiker die Integration zuvor bereits verwenden konnten. Sowohl die zwei signifikanten Unterschiede als auch der fast signifikante Unterschied wurden dabei ausgeglichen, was darauf hindeutet, dass sowohl Informatiker als auch Nicht-Informatiker die Integration als intuitiv betrachten und zufriedenstellend bedienen sowie problemlos nutzen können. Deshalb kann die Beantwortung der zweiten Forschungsfrage um Folgendes ergänzt werden: Bei einer erneuten Verwendung der Integration konnten jedoch die Nicht-Informatiker die signifikanten Unterschiede zu den Informatikern des ersten Durchlaufes ausgleichen, sodass keine signifikanten Unterschiede mehr bestehen.

Abschließend wurde in diesem Kapitel mittels des Kendall-Tau-Korrelationskoeffizienten überprüft, ob Korrelationen zwischen den abhängigen und unabhängigen Variablen dieser Studie existieren. Dabei wurden insgesamt acht Korrelationen festgestellt, wobei zwei Korrelationen das Gleiche aufzeigen wie zwei andere Korrelationen. Das liegt daran, dass als unabhängige Variable die Art der Studiendurchführung mit geprüft wurde. Die Informatiker führten die Studie online mit einem Zoom-Meeting mit dem Studieninitiator durch, während der Großteil der Nicht-Informatiker (7 von 8) mit ihm im selben Raum war. Daher repräsentieren die beiden Varianten ebenfalls die Informatiker- und Nicht-Informatiker-Gruppe. Die festgestellten Korrelationen sind:

- Informatiker empfanden die Aufgaben tendenziell leichter als Nicht-Informatiker.
- Informatiker benötigten tendenziell weniger Aufwand zur Aufgabenbewältigung.
- Probanden mit mehr Videospiele-Erfahrung empfanden die Aufgaben als leichter als diejenigen mit weniger Erfahrung.
- Probanden mit mehr Videospiele-Erfahrung benötigten weniger Aufwand zur Aufgabenbewältigung als diejenigen mit weniger Erfahrung.
- Jüngere Probanden konnten im zweiten Durchlauf mehr Zeit einsparen als ältere.
- Die Probanden, die die Studie online und mit der Zoom-Option wählten, waren in der Aufgabenbewältigung schneller als diejenigen, die mit dem Studieninitiator im selben Raum waren.
- Die Probanden, die die Studie online und mit der Zoom-Option wählten, empfanden die Komplexität der Aufgaben als leichter als diejenigen, die mit dem Studieninitiator im selben Raum waren.
- Die Probanden, die die Studie online und mit der Zoom-Option wählten, empfanden den aufzubringenden Aufwand zur Aufgabenbewältigung als leichter als diejenigen, die mit dem Studieninitiator im selben Raum waren.

Im abschließenden [fünften Kapitel](#) wurden zahlreiche potenzielle Erweiterungen und Verbesserungen für diese Integration vorgestellt, die in der Zukunft implementiert werden könnten.



GLOSSAR

- Alpha-Wert** Gibt die Transparenz einer Farbe an. [194](#), [195](#), [219](#)
- Asset** Ist eine digitale Ressource wie ein Skript, ein 3D-Modell, etc., das von Unity verwendet wird, um visuelle oder auch akustische Inhalte in das Spiel zu integrieren. (Nicolasreto, 2023; Seifert und Wislaug, 2017) [3](#), [182](#), [183](#), [196](#)
- AttachedObject-Objekt** Ist das Elternobjekt von den Drawable-Typen eines Drawables und repräsentiert den "Ablageort" für diese Typen. Dieses Objekt darf nicht skaliert sein. [21](#), [22](#), [39](#), [40](#), [43](#), [77](#), [201](#)
- Attribut** Ist eine Eigenschaft oder ein Merkmal, das einer Entität oder einer Beziehung zugeordnet ist. [189](#), [190](#)
- Beziehung** Beschreiben die Verbindung zwischen verschiedenen Entitäten. [189](#)
- Box-Collider** Ist ein Komponententyp, der einem GameObject hinzugefügt werden kann, um diesen mit einer rechteckigen Kollisionsform auszustatten. Diese Kollisionsform ist erforderlich, um mit dem Objekt zu interagieren. [15](#), [16](#), [20](#), [22](#), [31](#), [34](#), [36–38](#), [181](#), [215](#)
- Canvas** Ist ein rechteckiges grafisches Element, das als Container für verschiedene UI-Elemente dient. [34](#), [68](#), [181](#)
- Checkbox-Button** Ist ein Button, der zwischen ausgewählt und nicht ausgewählt umschalten kann. [8](#)
- Client** Beschreibt in diesem Fall die ausgeführte eigene *SEE*-Instanz. [197](#)
- Code-City** In der Code-City-Metapher werden Softwarekomponenten durch Gebäude in einer Stadt repräsentiert, wobei die Eigenschaften dieser Gebäude verschiedene Metriken der Software ausdrücken können — z. B. könnte die Höhe eines Gebäudes der Anzahl der Codezeilen entsprechen. [4](#), [5](#)
- Code-Window** Ist ein Fenster, in dem Programmcode angezeigt, eingegeben und bearbeitet werden kann. [187](#)
- Collider** Ist eine wichtige Komponente zur Kollisionserkennung. Durch Collider-Komponenten wird außerdem die Benutzerinteraktion mit dem jeweiligen Objekt ermöglicht. [15](#), [16](#), [18–20](#), [27](#), [29](#), [31](#), [34](#), [37](#), [69](#), [104](#), [215](#)
- Cutout** Ist ein Rendering Mode, der es ermöglicht, transparente Effekte mit klaren Kanten zwischen undurchsichtigen und transparenten Bereichen darzustellen (Unity_Technologies, 2024b). [27](#)
- Dashed** Ist eine gestrichelte Linie, sie wird mittels kurzen Segmentunterbrechungen gezeichnet. [8](#)
- deterministisch** Wenn für jede Eingabe eine eindeutige Ausgabe vorhersehbar ist. [188](#)
- Dictionary** Ist eine Klasse, die Schlüssel-Wert-Paare speichert. Jeder Schlüssel darf dabei nur einmal vorkommen (Seifert und Wislaug, 2017). In anderen Programmiersprachen wird diese auch als Map oder HashMap bezeichnet. [38](#)

- Drawable** Ist eine Ressource, die grafische Inhalte repräsentieren kann. In dieser Arbeit umfasst dies die Fläche des Whiteboards und des Sticky-Notes auf dem gezeichnet, geschrieben und Bilder hinzugefügt werden kann. [1](#), [3](#), [7](#), [11](#), [12](#), [17–19](#), [21–23](#), [33](#), [34](#), [38–41](#), [43](#), [44](#), [46–48](#), [51](#), [52](#), [56–58](#), [60–65](#), [70](#), [75](#), [77](#), [80–86](#), [89–94](#), [97–102](#), [104](#), [106](#), [107](#), [114](#), [125–128](#), [130](#), [131](#), [134](#), [197](#), [198](#), [201–204](#), [215](#), [217](#)
- Drawable-Font** Ist das Font Asset, das nicht nur die Schriftart selbst, sondern auch verschiedene Einstellungen wie beispielsweise die Schriftfarbe enthält. Je nachdem, welcher Shader angegeben wird, werden zusätzliche Eigenschaften hinzugefügt. [194](#)
- Drawable-Holder** Ist das oberste Elternobjekt (Root-Objekt) eines Drawables. Dieses Objekt darf nicht skaliert sein. [21](#), [25](#), [40](#), [63](#), [69](#)
- Drawable-Typ** Ist ein grafischer Inhalt, der auf einem Drawable repräsentiert werden kann. In dieser Arbeit sind dies Linien, Texte, Bilder und Mind Maps. [7](#), [9–13](#), [16](#), [19–26](#), [29–31](#), [33–35](#), [38–40](#), [43](#), [44](#), [66–71](#), [76](#), [77](#), [80](#), [82–86](#), [91](#), [93](#), [98–100](#), [102](#), [104](#), [106](#), [107](#), [176](#), [181](#), [182](#), [189](#), [200](#), [202–204](#), [215](#)
- Dropdown-Button** Ist ein Button, der ein Auswahlmü in Form eines Dropdown-Menüs bereitstellt. Dieses Auswahlmü öffnet sich erst, nachdem der Benutzer auf den Button geklickt hat. [8](#), [16](#)
- Entität** Objekt der realen Welt, die in der Datenbank gespeichert werden sollen. [189](#)
- Filebrowser** Ist eine Komponente zur Darstellung des Dateisystems des jeweiligen Endgerätes. Dazu wird eine grafische Benutzeroberfläche bereitgestellt, worin der Benutzer Verzeichnisse nach gezielten Dateien durchsuchen und die gewünschte Datei auswählen kann. [8](#), [9](#), [11](#), [61](#), [62](#), [90–92](#), [216](#)
- GameObject** Ist ein Objekt, das Entitäten repräsentiert und dem durch das Hinzufügen von Komponenten und Skripten bestimmte Eigenschaften, Funktionalitäten und Verhalten verliehen werden kann. [12](#), [13](#), [15](#), [16](#), [23](#), [34](#), [38–40](#), [43](#), [77](#)
- Gradient** Stellt den Farbverlauf zwischen zwei Farben dar. [8](#)
- Highlight** Ist das visuelle Hervorheben eines Objektes, um es deutlicher darzustellen. [68](#), [69](#), [90–92](#), [216](#)
- horizontales Selektormenü** Ist ein Auswahlmü, bei dem die Auswahllemente nebeneinander angeordnet sind. Der Benutzer kann zwischen den Elementen durch Klicken auf einen Zurück- und Vorwärts-Buttons navigieren. [8](#), [45](#), [46](#), [49](#)
- Kendall-Tau-Korrelationskoeffizient** Mit diesen Korrelationstest wird geprüft, ob zwei Variablen einen ungerichteten Zusammenhang besitzen. Dabei ist ein positiver, negativer oder kein Zusammenhang möglich (Walther, 2022). Beide Variablen müssen nicht normalverteilt sein und benötigen lediglich ein Ordinalskalenniveau (DATAtab_Team, 2024a). [172](#), [206](#)
- Likert-Skala** Ist eine stufenweise Antwortskala, um Meinungen zu einem bestimmten Thema auszudrücken. Sie ist nach dem amerikanischen Forscher Rensis Likert benannt. In der Praxis haben sich fünf- oder siebenstufige Skalen durchgesetzt (Qualtrics.XM, 2024). [116](#), [118–121](#)

- LineRenderer** Ist eine Komponente, die das Zeichnen von Linien ermöglicht. Der LineRenderer verwendet dazu eine Liste von Punkten, die miteinander verbunden werden. [26–28](#), [46](#), [47](#)
- Local Space** Die Koordinaten sind abhängig von der Position und Rotation des Elternobjektes. [26](#)
- Lokalisierung** Ist die Möglichkeit, die Software in Bezug auf die Sprache an verschiedene Regionen oder Länder anzupassen. [204](#)
- Lokalisierungs-Schlüssel** Ist der Begriff, der verwendet wird, um den gewünschten Text in einem Wörterbuch oder einer Ressourcendatei zu identifizieren. [204](#)
- Mann-Whitney-U-Test** Mit diesem Test wird überprüft, ob es einen signifikanten Unterschied zwischen zwei Gruppen gibt. Für diese Überprüfung wird die Rangsumme der Gruppen gebildet und für den Vergleich genutzt. Zur Durchführung des Tests werden lediglich zwei unabhängige Stichproben mit mindestens Ordinalskalenniveau benötigt. Es ist keine spezifische Verteilungskurve erforderlich ([DATAtab_Team, 2024b](#)). Dieser Test ist auch als Wilcoxon-Mann-Whitney-Test oder Wilcoxon-Rangsummentest bekannt ([Hemmerich, 2024](#)). [135](#), [148](#), [155](#), [171](#), [205](#)
- Material** Ist eine Ressource, die die visuellen Eigenschaften eines Objektes definiert. Durch sie wird festgelegt, welche Texturen oder Farben auf der Oberfläche dargestellt werden und wie sich das Objekt visuell durch verschiedene Einflussfaktoren verändert. Die Veränderungen können Aspekte wie Beleuchtung, Glanz oder Schatten betreffen ([Seifert und Wislaug, 2017](#)). [12](#), [26–28](#)
- Memento** Ist ein Entwurfsmuster zum Erfassen des Zustandes eines Objektes, um es zu speichern und wiederherstellen zu können ([Pubudu, 2021](#)). [25](#), [26](#), [47](#), [51](#), [57](#), [60](#), [64](#), [65](#), [67](#), [71](#), [77](#), [80](#), [81](#), [83](#), [85](#), [87](#), [89](#), [91–94](#), [96–98](#), [100](#), [101](#), [103](#), [104](#), [106](#)
- Mesh** Ist die Struktur eines Modells. Es verbindet die einzelnen Vektoren (Vertices) aus denen das Modell besteht, um eine Objektform zu bilden ([Seifert und Wislaug, 2017](#)). [16](#), [47](#)
- Mesh-Collider** Ist ein Komponententyp, der einem GameObject zur Kollisionserkennung hinzugefügt werden kann. Die Kollisionsform passt sich an das Mesh des Objektes an ([Seifert und Wislaug, 2017](#)). [15](#), [16](#), [19](#), [29–31](#), [37](#), [47](#), [215](#)
- MonoBehaviour** Dient als Basisklasse aller Skript-Klassen, sie wird als Komponente genutzt und kann einem GameObject angehängt werden. Sie stellt unter anderem Methoden wie Awake, Start und Update bereit. Diese funktionieren wie die im Kapitel [Anforderungen und Ziele](#) beschriebenen Methoden. Darüber hinaus können mit diesen Komponenten auch Physik-Methoden wie z. B. "OnTriggerEnter" bereitgestellt werden ([Seifert und Wislaug, 2017](#)). [24](#), [29](#), [35](#), [38](#), [58](#), [59](#), [61](#), [63](#), [68](#), [69](#)
- Monochrome** Ist die Verwendung von einer gleichbleibenden Farbe. [8](#)
- nichtdeterministisch** Wenn für eine Eingabe verschiedene Ausgaben möglich sind. Die Ausgabe ist folglich nicht eindeutig vorhersehbar. [188](#)
- OnTriggerEvent** Ist ein Event, das ausgelöst wird, wenn ein anderes GameObject mit dem Collider-Bereich des jeweiligen Objektes interagiert. Diese Events können auftreten, wenn das GameObject in den Trigger-

- Bereich eintritt (OnTriggerEnter), ihn verlässt (OnTriggerExit) oder sich darin aufhält (OnTriggerStay) (Seifert und Wislaug, 2017). 22, 69
- Opaque** Ist ein Rendering Mode für undurchsichtige Objekte ohne transparente Bereiche (Unity_Technologies, 2024b). 26
- Outline-Color** Beschreibt in diesem Kontext die Außenfarbe beziehungsweise Umrandungsfarbe des Textes. 194, 195, 219
- Polymorphie** Sorgt dafür, dass Objekte, die eine gemeinsame Oberklasse haben, gleich behandelt werden können (Seifert und Wislaug, 2017). 25
- Prefab** Ist eine Vorlage für GameObjects. Prefabs enthalten alle wichtigen Informationen für das Objekt, wie z. B. Materialien, Texturen, Collider, Kinderobjekte. Diese Prefabs können auch während der Laufzeit beliebig oft hinzugefügt werden (Seifert und Wislaug, 2017). 18–20
- Raycasting** Ist ein Verfahren, bei dem ein Strahl in eine bestimmte Richtung gesendet wird und überprüft, ob eine Kollision mit einem Objekt mit einem Collider entsteht (Seifert und Wislaug, 2017). 15, 42, 46, 50, 57, 70, 78–80, 82, 86, 99, 102, 104
- RectTransform** Ist eine Spezialisierung der Transform-Komponente, die für UI-Elemente verwendet wird. Sie besitzt zusätzlich noch eine Außenmaße für das Objekt. 30, 31, 34, 36
- Redo** Ist eine Funktion, die es ermöglicht, eine zuvor rückgängig gemachte abgeschlossene Aktion zu wiederholen. 10, 25, 41, 44, 65, 67, 68, 76, 78, 82, 85, 87, 89, 92, 93, 96, 97, 99, 100, 104, 107, 197, 203
- Relation** Ist eine Tabelle, die eine bestimmte Entität oder Beziehung repräsentiert. 190
- Renderer** Ist eine Komponente, die für die Darstellung der 3D- oder 2D-Grafiken verantwortlich ist. Er verarbeitet die visuelle Darstellung von Objekten und sorgt für ihre Anzeige (Seifert und Wislaug, 2017; Unity_Technologies, 2024a). 12, 13
- Rich-Text-Tag** Verändern das Aussehen und das Layout von Texten. Der Tag ist ähnlich zu einem HTML- oder XML-Tag (Unity_Technologies, 2023b). Ein solcher Tag kann die Eigenschaften einer TextMeshPro-Komponente überschreiben. 31–33, 191, 215
- Rigidbody** Ist die zentrale Komponente, die die Interaktion mit der Unity-Physik ermöglicht. Dies umfasst neben der Bewegung und Rotation des Objektes auch die Kollisionserkennung (Seifert und Wislaug (2017). 69, 70
- Schlüssel** Es gibt zwei Arten von Schlüsseln in relationalen Datenbanken: den Primärschlüssel und den Fremdschlüssel. Jede Relation hat einen Primärschlüssel, der aus einem oder mehreren Attributen bestehen kann und jeden Tupel der Relation eindeutig identifiziert. Der Fremdschlüssel wird verwendet, um Beziehungen zwischen Relationen herzustellen, indem er auf den Primärschlüssel einer anderen Relation verweist. 190
- Shader** Ist ein Programm, das verwendet wird, um die Darstellung von 3D-Objekten oder 2D-Grafiken zu beeinflussen. Sie ermöglichen die Erzeugung von visuellen Effekten wie Licht und Schatten, Oberflächeneigenschaften, etc. 32, 194, 219
- Slider** Ist eine Benutzeroberfläche mit einem Schieberegler, der zum Einstellen eines Wertes innerhalb eines bestimmten Wertebereichs benutzt

- wird. Zum Verändern des Wertes muss der Benutzer den Regler verschieben. [42](#), [45](#), [50](#), [54](#), [60](#), [71–73](#), [79](#), [102](#), [106](#)
- Solid** Ist eine durchgezogene Linie, sie wird durchgängig und ohne Unterbrechungen gezeichnet. [8](#)
- Sticky-Note** Ist ein Objekt, das dazu bestimmt ist, Notizen festzuhalten. Es wird auf Oberflächen von anderen Objekten angehaftet. Synonyme Bezeichnungen sind Haftnotiz oder Post-it. [7](#), [11](#), [20](#), [24](#), [44](#), [66](#), [92](#), [93](#), [100–107](#), [131](#), [202](#), [204](#), [215](#), [217](#)
- Switch-Button** Ist ein Button mit einem Schalter, der zum Ein- und Ausschalten von Funktionen verwendet werden kann. [49](#), [60](#), [66](#), [71](#), [73](#), [78](#), [80](#)
- Tag** Ist ein Etikett für GameObjects, das während der Visualisierungslaufzeit zur Identifizierung dient (Seifert und Wislaug, 2017). [16](#), [17](#), [19](#), [21](#), [22](#), [29](#), [31](#), [34](#), [37](#), [39](#), [40](#), [43](#), [102](#)
- Textur** Ist eine Grafik, die auf der Oberfläche eines Objektes aufgebracht wird. Sie dient dazu, das Objekt realistischer, detailgetreuer oder ansprechender zu gestalten (Seifert und Wislaug, 2017). [12](#), [13](#), [34](#)
- Transform** Ist die grundlegende Komponente für GameObjects. Sie stellt die Position, die Rotation und die Skalierung für das Objekt bereit (Seifert und Wislaug, 2017). [25](#), [30](#), [46](#)
- Turingmaschine** Ist ein Automatenmodell mit einem unendlich langen Band und einem Lese-/Schreibkopf als Speicher. Eine Eingabe für das Band muss jedoch endlich sein. [188](#)
- Two-Dashed** Stellt den Farbwechsel bei einer gestrichelten Linie bereit. Dabei werden die freigelassenen Stellen zwischen den Strichen eingefärbt. [8](#)
- Undo** Ist eine Funktion, die es ermöglicht, die zuletzt durchgeführte Aktion des Benutzers rückgängig zu machen. [10](#), [25](#), [41](#), [44](#), [47](#), [65](#), [67](#), [68](#), [76](#), [78](#), [82](#), [85](#), [87](#), [89](#), [92](#), [93](#), [95](#), [97](#), [99](#), [100](#), [104](#), [107](#), [203](#)
- Unity** Eine Entwicklungs-Engine für 2D- und 3D-Anwendungen, insbesondere für Spiele. [3](#), [15–17](#), [21](#), [34](#), [58](#), [69](#), [70](#)
- Value-Holder** Ist die Konfigurationsinstanz dieser Integration, sie speichert die derzeitigen gewählten Werte wie Primär- und Sekundärfarbe oder Liniendicke. Außerdem werden auch Konstanten bereitgestellt. [24](#), [41](#), [44](#), [66](#), [67](#), [78](#), [79](#), [88](#), [102](#)
- Web-View** Durch eine Web-View wird der Zugriff auf einen Ingame-Webbrowser ermöglicht. [182](#)
- Wilcoxon-Vorzeichen-Rang-Test** Dieser Test prüft, ob zwei abhängige Stichproben signifikant unterschiedlich sind. Die Daten müssen dabei nicht normalverteilt sein. Damit der Test angewendet werden kann, müssen die Daten paarweise vorliegen (DATAtab_Team, 2024c). [159](#), [205](#)
- World Space** Die Koordinaten beziehen sich auf die globalen Koordinaten der 3D-Welt, unabhängig von der Position oder Rotation eines bestimmten Objektes. [26](#)
- Zustimmungsverzerrung** Beschreibt den Fall, dass der Proband den Fragen zustimmt, ohne dass dies seine persönliche Meinung widerspiegelt. Im Englischen ist dies als acquiescence bias/agreement bias bekannt (Toor, 2020). [118](#), [119](#)

AKRONYME

- ASQ** Ist ein standardisierter Fragebogen, der nach jeder erledigten Aufgabe ausgefüllt wird. Dabei bewertet er die Leichtigkeit der Aufgabenerledigung sowie die Zufriedenheit mit der benötigten Zeit für die Aufgabe und die Zufriedenheit mit den bereitgestellten Informationen (Dörflinger, n.d.a). [110–112](#), [115](#), [116](#), [119](#), [121](#), [143](#), [146–149](#), [152–155](#), [157](#), [163–166](#), [168](#), [173–175](#), [205](#), [218](#), [221](#), [222](#)
- ER** Ist ein Post-Task-Fragebogen, bestehend aus zwei Fragen. Die erste wird vor der Aufgabe gestellt und erfasst die zu erwartende Schwierigkeit der Aufgabe. Die zweite wird nach Abschluss der Aufgabe gestellt und erfasst die empfundene Schwierigkeit. [119–121](#), [217](#)
- GUI** Bezeichnet eine grafische Benutzeroberfläche, die eine Benutzerinteraktion ermöglicht. [15](#), [42](#)
- NASA-TLX** Ist ein Post-Task-Fragebogen mit sechs Fragen, die die Themen mentale, körperliche, zeitliche Anstrengung sowie Leistung im Sinne von dem Arbeitserfolg, empfundenen Aufwand und Frustration umfasst. [121](#)
- OCL** Ist ein Bestandteil von UML, mit dem notwendige Randbedingungen festgelegt werden können, wie beispielsweise Invarianten für Klassendiagramme, Bedingungen in Sequenzdiagrammen oder Vor- und Nachbedingungen für Methoden. [185](#), [187](#)
- OMG** Ist ein internationales, gemeinnütziges Konsortium, das sich auf die Entwicklung und Pflege von Technologiestandards konzentriert. [184](#)
- PSSUQ** Ist ein Post-Study-Fragebogen, der ursprünglich aus 18 Fragen bestand und in der aktuellen Version 16 Fragen umfasst. [118](#), [119](#)
- QUIS** Ist ein Post-Study-Fragebogen aus dem Jahr 1988, der zwei verschiedene Längen bietet. Entweder 41 oder 122 Fragen. [117](#)
- SEE** Eine interaktive Visualisierung von Software, welche die *Code-City*-Metapher verwendet und einen kollaborativen Multiplayer über verschiedene Plattformen¹ hinweg ermöglicht. [3](#), [4](#), [7](#), [9](#), [10](#), [15](#), [16](#), [18](#), [26](#), [41](#), [43](#), [58](#), [61](#), [91](#), [114](#), [139](#), [140](#), [173–176](#), [179](#), [180](#), [182](#), [183](#), [187](#), [196](#), [204](#), [217](#)
- SEQ** Ist ein Post-Task-Fragebogen mit einer Frage, die die empfundene Schwierigkeit der Aufgabe erfragt. [121](#)
- SMEQ** Ist ein Post-Task-Fragebogen mit einer Frage, die die mentale Anstrengung erfasst. Die Antwort wird mittels einer Skala von 0 bis 150 erfasst. [119](#)

¹ Neben Desktop- und Touch-Umgebungen noch Virtual Reality (z. B. *Valve Index*) und Augmented Reality (z. B. *Microsoft HoloLens*)

SUMI Ist ein Post-Study-Fragebogen aus dem Jahr 1993, der aus 50 Fragen besteht. [117](#)

SUPR-Q Ist ein Post-Study-Fragebogen bestehend aus acht Fragen, der auf Webseiten ausgelegt ist. [119](#)

SUS Ist ein standardisierter Fragebogen von John Brooke aus dem Jahr 1986, der am Ende der Studie zur Auswertung der Gesamtbenutzerfreundlichkeit des Systems verwendet wird und zehn verschiedene Fragen enthält. [110–112](#), [115](#), [117–119](#), [143–147](#), [149–151](#), [160–163](#), [173](#), [177](#), [205](#), [217](#), [218](#), [221](#)

TMP Ist eine Komponente zur Darstellung von Texten. Es bietet gegenüber der standardmäßigen Textkomponente von Unity eine erweiterte Funktionalität und eine bessere Kontrolle über die Darstellung. [30–33](#), [191](#), [196](#)

UME Ist ein Post-Task-Fragebogen, der sich mit der empfundenen Schwierigkeit der Aufgaben befasst. Dazu muss der Proband selbst eine Skala definieren und die Aufgaben entsprechend bewerten. [120](#)

UML Definiert eine visuelle Modellierungssprache. [184](#)

USE Ist eine UML-basierte Spezifikationsumgebung, mit der UML-Diagramme erstellt, ausgewertet und mittels *OCL* verfeinert werden können. [185](#)

ABBILDUNGSVERZEICHNIS

Abbildung 2.1	Demonstration der Dateiauswahl für das Metric Board.	9
Abbildung 2.2	Beispielanwendung von XDPaint.	12
Abbildung 3.1	Überblick der Unity Collider von Seifert und Wislaug (2017) . Von links nach rechts: Box-, Sphere-, Capsule- und Mesh-Collider.	16
Abbildung 3.2	Demonstration der Rotation für ein Drawable.	17
Abbildung 3.3	Die Objekthierarchie des Whiteboards.	18
Abbildung 3.4	Inspektor des PaintingReceiver-Objektes.	19
Abbildung 3.5	Die Objekthierarchie des Sticky-Notes.	20
Abbildung 3.6	Transformation eines Drawables.	22
Abbildung 3.7	Die verschiedenen Linienarten.	27
Abbildung 3.8	Dashed-Linienart mit einer eingestellten Tiling von 0.	28
Abbildung 3.9	Vergleich der Dashed-Linienart mit einer eingestellten Tiling von 10 und der Dashed100-Linienart.	28
Abbildung 3.10	Die verschiedenen Farbvarianten anhand von Solid- und Dashed-Linien. Die gewählte Primärfarbe ist blau und die Sekundärfarbe rot.	29
Abbildung 3.11	Der Mesh-Collider einer Spiralenlinie (Scene-View).	30
Abbildung 3.12	Demonstration der Collider für den Text-Drawable-Typ.	31
Abbildung 3.13	Demonstration der verschiedenen Farbeinstellungen für Texte.	32
Abbildung 3.14	Demonstration der verschiedenen Textstyles.	33
Abbildung 3.15	Ein Beispiel für die Verwendung eines Rich-Text-Tag zum Unterstreichen.	33
Abbildung 3.16	Demonstration des Box-Collider für den Image-Drawable-Typ.	34
Abbildung 3.17	Demonstration der Einfärbung für den Image-Drawable-Typ.	35
Abbildung 3.18	Die Hierarchie eines Mind Map-Knotenobjektes.	36
Abbildung 3.19	Die Knotenarten der Mind Map.	36
Abbildung 3.20	Die Blatt-Knotenart inklusive Box-Collider.	36
Abbildung 3.21	Box-Collider des zentralen Themas und des Untertemas einer Mind Map.	38
Abbildung 3.22	Darstellung des HSV-Color-Pickers.	42
Abbildung 3.23	Das Line-Menü für die Draw Freehand-Aktion.	45
Abbildung 3.24	Vergleich der Liniendicke.	46
Abbildung 3.25	Demonstration der Auswahl des Switch-Menüs der Shape-Aktion.	49
Abbildung 3.26	Demonstration des Shape-Menüs für hinzufügbare geometrische Formen.	50
Abbildung 3.27	Demonstration der Information über das Entfernen des zuletzt hinzugefügten Punktes.	51
Abbildung 3.28	Demonstration der Information über das Abbrechen des Zeichnens gerader Linien.	51

Abbildung 3.29	Polygone mit einer unterschiedlichen Anzahl von Eckpunkten.	56
Abbildung 3.30	Demonstration des Text-Menüs.	58
Abbildung 3.31	Demonstration der kleinsten, Standard- und größten Schriftgröße.	59
Abbildung 3.32	Demonstration des Tooltips für die Style-Buttons. . .	60
Abbildung 3.33	Demonstration des Dialogs, in dem der Text eingegeben wird.	61
Abbildung 3.34	Demonstration der Quellenauswahl für Bilder. . . .	61
Abbildung 3.35	Demonstration des Filebrowsers für die Bilderauswahl.	62
Abbildung 3.36	Demonstration des Hinzufügedialogs für Webbilder.	63
Abbildung 3.37	Demonstration des Mind Map-Menüs.	64
Abbildung 3.38	Demonstration der Auswahl des Elternknotens. . . .	65
Abbildung 3.39	Demonstration des bereitgestellten Color-Picker Menüs.	66
Abbildung 3.40	Demonstration der Objektauswahl eines Mind Map-Knotens für den Color-Picker.	67
Abbildung 3.41	Demonstration des bereitgestellten Highlight-Effektes der Klasse "GameHighlighter".	69
Abbildung 3.42	Demonstration des Edit-Menüs für Linien.	72
Abbildung 3.43	Demonstration des Edit-Menüs für Texte.	73
Abbildung 3.44	Demonstration des Edit-Menüs für Bilder.	74
Abbildung 3.45	Demonstration des Edit-Menüs für Mind Map-Knoten.	75
Abbildung 3.46	Demonstration des Edit-Menüs für die Knotenart eines Mind Map-Knotens.	76
Abbildung 3.47	Demonstration des Auswahl-Menüs für Move oder Rotate.	77
Abbildung 3.48	Demonstration des Move-Menüs.	78
Abbildung 3.49	Demonstration des Rotate-Menüs.	79
Abbildung 3.50	Demonstration des Scale-Menüs.	80
Abbildung 3.51	Demonstrationsbeispiel für proportionales und unproportionales Skalieren.	81
Abbildung 3.52	Demonstration der Änderung der Ebenenreihenfolge.	82
Abbildung 3.53	Demonstration der Benachrichtigung bei einer Änderung der Ebenenreihenfolge.	83
Abbildung 3.54	Demonstration der Warnung des Minimums.	83
Abbildung 3.55	Demonstration der Warnung des Maximums.	83
Abbildung 3.56	Demonstration des Cut, Copy, Paste-Menüs.	84
Abbildung 3.57	Demonstration des Verschiebens eines Punktes einer Linie.	86
Abbildung 3.58	Anwendung der Move a Point-Aktion auf mehrere Punkte.	87
Abbildung 3.59	Demonstration des Markierens des Spaltens einer Linie.	88
Abbildung 3.60	Anwendung der Line Split-Aktion auf mehrere Punkte.	89
Abbildung 3.61	Demonstration des Save-Menüs.	89
Abbildung 3.62	Demonstration des Highlight-Effektes für das Speichern und Laden.	90
Abbildung 3.63	Demonstration des Filebrowsers für das Speichern. .	91
Abbildung 3.64	Demonstration des Load-Menüs.	92
Abbildung 3.65	Demonstration des Löschens eines Punktes einer Linie.	94
Abbildung 3.66	Anwendung der Line Point Erase-Aktion auf einer frei gezeichneten Linie.	95

Abbildung 3.67	Anwendung der Line Point Erase-Aktion auf mehrere Punkte.	95
Abbildung 3.68	Demonstration des Löschens einer Verbindungslinie einer Linie.	96
Abbildung 3.69	Anwendung der Line Connection Erase-Aktion auf mehrere Punkte.	97
Abbildung 3.70	Anwendung der Erase-Aktion auf den Knoten "Duathlon".	98
Abbildung 3.71	Anwendung der Cleaner-Aktion auf das Whiteboard.	100
Abbildung 3.72	Information über ein bereits gesäubertes Drawable.	100
Abbildung 3.73	Darstellung des Sticky-Note-Menüs.	101
Abbildung 3.74	Darstellung des Sticky-Note-Move-Menüs.	103
Abbildung 3.75	Demonstration der Rotate-Menüs für Sticky-Notes. .	103
Abbildung 3.76	Informationen, die über den "i"-Button des Sticky-Note-Move-Menüs angezeigt werden können.	103
Abbildung 3.77	Darstellung des Sticky-Note-Edit-Menüs.	105
Abbildung 3.78	Demonstration des Löschens eines Sticky-Notes. . .	107
Abbildung 4.1	Das Streudiagramm des ER von Albert (2003)	120
Abbildung 4.2	Der Formulareditor des KoboToolbox-Tools.	123
Abbildung 4.3	Hinweisbild der Aufgabe 1.	126
Abbildung 4.4	Lösungsbild der Aufgabe 1.	127
Abbildung 4.5	Lösungsbild der Aufgabe 2.	128
Abbildung 4.6	Hinweisbild der Aufgabe 3.	129
Abbildung 4.7	Lösungsbild der Aufgabe 3.	130
Abbildung 4.8	Lösungsbild der Aufgabe 4.	131
Abbildung 4.9	Lösungsbild der Aufgabe 5.	132
Abbildung 4.10	Zwischenbild der Aufgabe 6.	133
Abbildung 4.11	Lösungsbild der Aufgabe 6.	133
Abbildung 4.12	Lösungsbild der Aufgabe 7.	134
Abbildung 4.13	Stammst du aus dem Informatikbereich?	136
Abbildung 4.14	Verteilung des Geschlechts aller Probanden.	137
Abbildung 4.15	Verteilung des Geschlechts in beiden Gruppen. . . .	137
Abbildung 4.16	Alter aller Probanden.	138
Abbildung 4.17	Verteilung des Alters beider Gruppen.	138
Abbildung 4.18	Verteilung des höchsten Abschlusses aller Probanden.	139
Abbildung 4.19	Verteilung des höchsten Abschlusses beider Gruppen.	139
Abbildung 4.20	Verteilung der SEE-Erfahrung aller Probanden. . . .	140
Abbildung 4.21	Verteilung der SEE-Erfahrung beider Gruppen. . . .	140
Abbildung 4.22	Verteilung der Bildbearbeitungsprogramm-Erfahrung aller Probanden.	141
Abbildung 4.23	Verteilung der Bildbearbeitungsprogramm-Erfahrung beider Gruppen.	141
Abbildung 4.24	Verteilung der Videospiele-Erfahrung aller Probanden.	142
Abbildung 4.25	Verteilung der Videospiele-Erfahrung beider Gruppen.	142
Abbildung 4.26	Verteilung der Art der Studiendurchführung aller Probanden.	143
Abbildung 4.27	Verteilung der Art der Studiendurchführung beider Gruppen.	143
Abbildung 4.28	Auswertung der gesamten SUS-Scores.	144
Abbildung 4.29	Auswertung der Usability-SUS-Scores.	145
Abbildung 4.30	Auswertung der Learnability-SUS-Scores.	146

Abbildung 4.31	Auswertung der SUS-Scores der Informatiker (α) und Nicht-Informatiker (β).	149
Abbildung 4.32	Auswertung der Usability-SUS-Scores der Informatiker (α) und Nicht-Informatiker (β).	150
Abbildung 4.33	Auswertung der Learnability-SUS-Scores der Informatiker (α) und Nicht-Informatiker (β).	151
Abbildung 4.34	Auswertung der ASQ-Leichtigkeit/Komplexität der Informatiker (α) und Nicht-Informatiker (β).	153
Abbildung 4.35	Auswertung des aufgebrauchten ASQ-Aufwands der Informatiker (α) und Nicht-Informatiker (β).	154
Abbildung 4.36	Auswertung der ASQ-Informationszufriedenheit der Informatiker (α) und Nicht-Informatiker (β).	155
Abbildung 4.37	Auswertung der benötigten Zeit der Informatiker (α) und Nicht-Informatiker (β).	156
Abbildung 4.38	Auswertung der SUS-Scores des ersten und zweiten Durchlaufes (First / Second).	161
Abbildung 4.39	Auswertung der Usability-SUS-Scores des ersten und zweiten Durchlaufes (First / Second).	162
Abbildung 4.40	Auswertung der Learnability-SUS-Scores des ersten und zweiten Durchlaufes (First / Second).	163
Abbildung 4.41	Auswertung der ASQ-Leichtigkeit/Komplexität des ersten und zweiten Durchlaufes (First / Second).	164
Abbildung 4.42	Auswertung des aufgebrauchten ASQ-Aufwandes des ersten und zweiten Durchlaufes (First / Second).	165
Abbildung 4.43	Auswertung der ASQ-Informationszufriedenheit des ersten und zweiten Durchlaufes (First / Second).	166
Abbildung 4.44	Auswertung der benötigten Zeit des ersten und zweiten Durchlaufes (First / Second).	167
Abbildung 4.45	Auswertung der Zeitersparnis zwischen dem ersten und zweiten Durchlauf jedes Probanden.	167
Abbildung 5.1	UML-Diagrammhierarchie	184
Abbildung 5.2	USE - Analyse	187
Abbildung 5.3	Getränkeautomat Beispiel für einen deterministischen Zustandsautomaten	188
Abbildung 5.4	Ausschnitt eines Beispiels für das ER-Modell	189
Abbildung 5.5	Ausschnitt eines Beispiels für das relationale Datenbankmodell	190
Abbildung 5.6	Beispiel für die Verwendung des Rich-Text-Tags “<gradient>“	191
Abbildung 5.7	Ein Beispiel für die Vermischung von Farben mittels eines Farbverlaufs mit der Textfarbe blau und der Farbverlaufsfarbe rot.	192
Abbildung 5.8	Anwendung des Single-Farbverlaufes mit der Verlaufsfarbe rot auf einem hellgrauen Text.	192
Abbildung 5.9	Anwendung des Vertikal-Farbverlaufes mit der Verlaufsfarbe gelb für oben und rot für unten auf einem hellgrauen Text.	193
Abbildung 5.10	Anwendung des Horizontal-Farbverlaufes mit der Verlaufsfarbe gelb für links und rot für rechts auf einem hellgrauen Text.	193

Abbildung 5.11	Anwendung des Vier-Ecken-Farbverlaufes mit der Verlaufsfarbe gelb für oben links, magenta für oben rechts, hellblau/türkis für unten links und für unten rechts rot auf einem hellgrauen Text.	193
Abbildung 5.12	Vergleich der beiden Shader-Varianten. Links ist der alte und rechts der neue Shader.	194
Abbildung 5.13	Vergleich zwischen Outline-Color ausgeschaltet und Alpha-Wert auf 0. Resultat: Gleiches Ergebnis. . . .	195
Abbildung 5.14	Vergleich zwischen Outline-Color ausgeschaltet und Outline-Color-Thickness(Dicke) auf 0. Resultat: Thickness auf 0 lässt den Text dicker erscheinen.	195
Abbildung 5.15	Demonstrationsbeispiel für den Glow-Effekt.	195
Abbildung 5.16	Demonstrationsbeispiel für den Lighting-Effekt. . . .	195
Abbildung 5.17	Demonstrationsbeispiel für eine Kombination aus dem Glow- und Lighting-Effekt.	196
Abbildung 5.18	Demonstrationsbeispiel für unterschiedliche Schriftarten.	196
Abbildung 5.19	Demonstrationsbeispiel für Add Point to Line. . . .	198
Abbildung 5.20	Funktionalität des Überlappungsproblemlösers. . . .	198
Abbildung 5.21	unsortiert	199
Abbildung 5.22	Rund-um-Sortierung	199
Abbildung 5.23	Baum-Sortierung	199
Abbildung 5.24	Seiten-Funktionalität für das Whiteboard.	201
Abbildung 5.25	Beispiel für Hilfslinien.	203

TABELLENVERZEICHNIS

Tabelle 3.1	Die bereitgestellten geometrischen Formen, ihre Einstellungen und jeweils ein erklärendes Bild.	56
Tabelle 4.1	Die Zeiten der einzelnen Aufgaben aller Probanden in Minuten.	147
Tabelle 4.2	Die ASQ-Ergebnisse der einzelnen Aufgaben aller Probanden in Bewertungspunkten.	148
Tabelle 4.3	Die Daten des Vergleiches der Aufgabe 1 zwischen Informatiker und Nicht-Informatiker.	157
Tabelle 4.4	Die Daten des Vergleiches der Aufgabe 2 zwischen Informatiker und Nicht-Informatiker.	158
Tabelle 4.5	Die Daten des Vergleiches der Aufgabe 3 zwischen Informatiker und Nicht-Informatiker.	158
Tabelle 4.6	Die Daten des Vergleiches der Aufgabe 4 zwischen Informatiker und Nicht-Informatiker.	158
Tabelle 4.7	Die Daten des Vergleiches der Aufgabe 5 zwischen Informatiker und Nicht-Informatiker.	159
Tabelle 4.8	Die Daten des Vergleiches der Aufgabe 6 zwischen Informatiker und Nicht-Informatiker.	159
Tabelle 4.9	Die Daten des Vergleiches der Aufgabe 7 zwischen Informatiker und Nicht-Informatiker.	159
Tabelle 4.10	Die Zeitersparnisse der einzelnen Aufgaben in Minuten. Die prozentuale Zeitersparnis wurde bemessen am Durchschnitt.	169
Tabelle 4.11	Die Daten des Vergleiches der Aufgabe 1 zwischen erstem und zweitem Durchlauf.	169
Tabelle 4.12	Die Daten des Vergleiches der Aufgabe 2 zwischen erstem und zweitem Durchlauf.	170
Tabelle 4.13	Die Daten des Vergleiches der Aufgabe 3 zwischen erstem und zweitem Durchlauf.	170
Tabelle 4.14	Die Daten des Vergleiches der Aufgabe 4 zwischen erstem und zweitem Durchlauf.	170
Tabelle 4.15	Die Daten des Vergleiches der Aufgabe 5 zwischen erstem und zweitem Durchlauf.	170
Tabelle 4.16	Die Daten des Vergleiches der Aufgabe 6 zwischen erstem und zweitem Durchlauf.	170
Tabelle 4.17	Die Daten des Vergleiches der Aufgabe 7 zwischen erstem und zweitem Durchlauf.	171
Tabelle 4.18	Ergebnisse der Korrelationsprüfung der abhängigen Variable "benötigte Zeit".	173
Tabelle 4.19	Ergebnisse der Korrelationsprüfung der abhängigen Variable "SUS".	173
Tabelle 4.20	Ergebnisse der Korrelationsprüfung der abhängigen Variable "ASQ-Komplexität" (Leichtigkeit).	174
Tabelle 4.21	Ergebnisse der Korrelationsprüfung der abhängigen Variable "ASQ-Aufwand".	175

Tabelle 4.22	Ergebnisse der Korrelationsprüfung der abhängigen Variable "ASQ-Informationszufriedenheit".	175
Tabelle 4.23	Ergebnisse der Korrelationsprüfung der abhängigen Variable "Zeitersparnis".	176

LITERATURVERZEICHNIS

- [Albert 2003] ALBERT, William: *Is this what you expected? The use of expectation measures in usability testing*. URL: <https://www.researchgate.net/publication/230786752-Is-this-what-you-expected-The-use-of-expectation-measures-in-usability-testing>. 2003. – Zuletzt abgerufen am: 15.03.2024
- [Allen 2012] ALLEN, David: *Wie ich die Dinge geregelt kriege: Selbstmanagement für den Alltag*, PIPER, 2012. – Epub: Kapitel "Warum Ihnen etwas nicht aus dem Kopf geht"
- [Allen 2023] ALLEN, Sam: *Remove HTML tags*. URL: <https://www.dotnetperls.com/remove-html-tags>. 2023. – Zuletzt abgerufen am: 24.02.2024
- [Alt 2020] ALT, Prof. Dr. R.: *Transformation*. URL: <https://www.gabler-banklexikon.de/definition/transformation-70694>. 2020. – Zuletzt abgerufen am: 28.02.2024
- [Aßmann 2021] AßMANN, Jens: *Brain Dump: Dein Gehirn ist dafür da, Ideen zu haben, nicht sie sich zu merken*. URL: <https://www.der-generalist.de/braindump/>. 2021. – Zuletzt abgerufen am: 29.01.2024
- [Berlin 2024] BERLIN, Freie U.: *Skalenniveaus (Fortsetzung)*. URL: https://www.ewi-psy.fu-berlin.de/erziehungswissenschaft/arbeitsbereiche/lernpsych/Paedagogische_Diagnostik/BL_Statistische_Auswertung/300/310/. 2024. – Zuletzt abgerufen am: 18.03.2024
- [Birkett 2021] BIRKETT, Alex: *Measuring User Satisfaction: 8 Ways to Improve Experience*. URL: <https://cxl.com/blog/8-ways-to-measure-ux-satisfaction/#h-2-nasa-tlx>. 2021. – Zuletzt abgerufen am: 15.03.2024
- [Bogner und Landrock 2015] BOGNER, Kathrin ; LANDROCK, Uta: *GESIS Survey Guidelines - Antworttendenzen in standardisierten Umfragen*, gesis Leibniz-Institut für Sozialwissenschaften, 2015, S. 2. – URL: https://www.gesis.org/fileadmin/upload/SDMwiki/Antworttendenzen_Bogner_Landrock_08102015_1.1.pdf, Zuletzt abgerufen am: 17.03.2024
- [Brooks 2002] BROOKS, Frederick P.: *The Mythical Man-Month*, ADDISON-WESLEY, 2002, S. 16–17
- [Buzan und Buzan 2013] BUZAN, Tony ; BUZAN, Barry: *Das Mind-Map Buch: Die beste Methode zur Steigerung Ihres geistigen Potenzials*, mvgverlag, 2013, S. 61–66,73–77,83–86,137–156
- [Cuofano 2023] CUOFANO, Gennaro: *Duale Codierung*. URL: <https://fourweekmba.com/de/doppelte-Codierung/>. 2023. – Zuletzt abgerufen am: 29.01.2024

- [DATAtab_Team 2024a] DATATAB_TEAM: *Kendalls Tau*. URL: <https://datatab.de/tutorial/kendalls-tau>. 2024. – Zuletzt abgerufen am: 18.03.2024
- [DATAtab_Team 2024b] DATATAB_TEAM: *Mann-Whitney U-Test*. URL: <https://datatab.de/tutorial/mann-whitney-u-test>. 2024. – Zuletzt abgerufen am: 18.03.2024
- [DATAtab_Team 2024c] DATATAB_TEAM: *Wilcoxon-Test*. URL: <https://datatab.de/tutorial/wilcoxon-test>. 2024. – Zuletzt abgerufen am: 18.03.2024
- [Dörflinger n.d.a] DÖRFLINGER, Michael: *ASQ – AFTER-SCENARIO QUESTIONNAIRE*. URL: <https://www.digital-management.at/asq-after-scenario-questionnaire/>. n.d.. – Zuletzt abgerufen am: 08.03.2024
- [Dörflinger n.d.b] DÖRFLINGER, Michael: *DIN EN ISO 9241-11 – Anforderungen an die Gebrauchstauglichkeit – Leitsätze*. URL: <https://www.digital-management.at/din-en-iso-9241-11/>. n.d.. – Zuletzt abgerufen am: 08.03.2024
- [Faulkner 2003] FAULKNER, Laura: *Beyond the five-user assumption: Benefits of increased sample sizes in usability testing*. URL: https://www.researchgate.net/publication/9032126_Beyond_the_five-user_assumption_Benefits_of_increased_sample_sizes_in_usability_testing. 2003. – Zuletzt abgerufen am: 15.03.2024
- [Friedrich u. a. 2018] FRIEDRICH, Julius-David ; HACHMEISTER, Cort-Denis ; NICKEL, Sigrun ; PEKSEN, Sude ; ROESSLER, Isabel ; ULRICH, Saskia: *Frauen in Informatik: Welchen Einfluss haben inhaltliche Gestaltung, Flexibilisierung und Anwendungsbezug der Studiengänge auf den Frauenanteil?*, Centrum für Hochschulentwicklung, 2018, S. 5, 14
- [Galperin 2021] GALPERIN, Falko: *Visualisierung von Code-Smells in Code-Cities*, Bachelorarbeit an der Universität Bremen, 2021, S. 37, 43
- [Galperin u. a. 2022] GALPERIN, Falko ; KOSCHKE, Prof. Dr. R. ; STEINBECK, Marcel: *Visualizing Code Smells: Tables or Code Cities? A Controlled Experiment*. URL: https://github.com/uni-bremen-agst/VISSOFT2022/blob/main/_Vissoft2022CodeSmellsCodeCities.pdf. 2022. – Zuletzt abgerufen am: 18.03.2024
- [Genau 2023] GENAU, Lea: *Die 5 Arten von Validität mit Formulierungshilfen mit Beispielen*. URL: <https://www.scribbr.de/methodik/validitaet/>. 2023. – Zuletzt abgerufen am: 17.03.2024
- [Gogolla u. a. 2005] GOGOLLA, M. ; HILKEN, F. ; DOAN, K.-H. ; DESAI, N. ; KÄSTNER, A.: *A UML-based Specification Environment*. URL: <https://www.db.informatik.uni-bremen.de/projects/USE-2.2.0/>. 2005. – Zuletzt abgerufen am: 22.01.2024
- [Gogolla u. a. 2007] GOGOLLA, M. ; HILKEN, F. ; DOAN, K.-H. ; DESAI, N. ; KÄSTNER, A.: *USE - A UML based Specification Environment*. URL: <https://www.db.informatik.uni-bremen.de/projects/USE/use-documentation.pdf>. 2007. – Zuletzt abgerufen am: 22.01.2024

- [Guerci 2020] GUERCI, Jonatas: *Easily calculate SUS Score*. URL: <https://uxplanet.org/easily-calculate-sus-score-a464d753e5aa>. 2020. – Zuletzt abgerufen am: 14.03.2024
- [Hemmerich 2022] HEMMERICH, W.: *Hawthorne-Effekt*. URL: <https://statistikguru.de/lexikon/hawthorne-effekt.html>. 2022. – Zuletzt abgerufen am: 17.03.2024
- [Hemmerich 2024] HEMMERICH, W.: *Einführung in den Mann-Whitney-U-Test*. URL: <https://statistikguru.de/spss/mann-whitney-u-test/einleitung-8.html>. 2024. – Zuletzt abgerufen am: 18.03.2024
- [Jacobsen 2019] JACOBSEN, Jens: *UX messbar machen & die UX-Kultur im Unternehmen stärken*. URL: <https://www.testingtime.com/blog/ux-messbar-machen-kultur-staerken/>. 2019. – Zuletzt abgerufen am: 14.03.2024
- [Kecher u. a. 2018] KECHER, C. ; SALVANOS, A. ; HOFFMANN-ELBERN, R.: UML 2.5 - Das umfassende Handbuch, Rheinwerk Verlag, 2018, S. 19–20, 22–23, 26–33
- [Keller 2022] KELLER, Dominik: *The Mythical Man Month: 5 Lessons About Software Development*. URL: <https://five.co/blog/5-lessons-on-software-development-the-mythical-man-month/>. 2022. – Zuletzt abgerufen am: 07.02.2024
- [Kirakowski und Corbett 1993] KIRAKOWSKI, Jurek ; CORBETT, Mary: *SUMI: the Software Usability Measurement Inventory*. URL: <https://bera-journals.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8535.1993.tb00076.x>. 1993. – Zuletzt abgerufen am: 14.03.2024
- [Koschke 2021] KOSCHKE, Prof. Dr. R.: *Softwaretechnik - Empirische Softwaretechnik - Threats To Validity*. URL: https://seafiler.zfn.uni-bremen.de/d/6fb8c90886d64a9c9ed1/files/?p=/12_empirie-experimente-threats-to-validity.pdf. 2021. – S. 9–13, Zuletzt abgerufen am: 17.03.2024
- [Koschke 2022] KOSCHKE, Prof. Dr. R.: *Auge in Auge mit Ihrer Softwarearchitektur – Modellierung, Visualisierung und Prüfung von Softwarearchitekturen in virtueller Realität*. URL: <https://www.embedded-software-engineering.de/modellierung-visualisierung-und-pruefung-von-softwarearchitekturen-in-virtueller-realitaet-a-1d5a6c7a2caaa9b0a9a8544cd58b7088/>. 2022. – Zuletzt abgerufen am: 26.01.2024
- [Koschke und et al. 2024] KOSCHKE, Prof. Dr. R. ; AL. et: *SOFTWARE ENGINEERING EXPERIENCE*. URL: <https://see.uni-bremen.de/>. 2024. – Zuletzt abgerufen am: 26.01.2024
- [Köller 2004] KÖLLER, Jürgen: *Drachenviereck*. URL: <https://www.mathematische-basteleien.de/drachen.htm>. 2004. – Zuletzt abgerufen am: 01.03.2024
- [Künzel und Schmitt 2023] KÜNZEL, Heidrun ; SCHMITT, Peter: *Leichter lernen mit Sketchnotes & Co.: Visuelle Lernmethoden für Schule, Uni und Weiterbildung*, mitp, 2023, S. 19–21. – Seitenzahlen könnten abweichen, daher Abschnitt: "Was verstehen wir unter Sketchnotes?"

- [Laubheimer 2018] LAUBHEIMER, Page: *Beyond the NPS: Measuring Perceived Usability with the SUS, NASA-TLX, and the Single Ease Question After Tasks and Usability Tests*. URL: <https://www.nngroup.com/articles/measuring-perceived-usability/>. 2018. – Zuletzt abgerufen am: 14.03.2024
- [Lewis 1991] LEWIS, James R.: *Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the ASQ*. URL: <https://dl.acm.org/doi/abs/10.1145/122672.122692>. 1991. – Zuletzt abgerufen am: 15.03.2024
- [Liebig n.d.] LIEBIG, Deborah: *WO SIND DIE FRAUEN IN DER IT?* URL: <https://www.get-in-it.de/magazin/arbeitswelt/it-arbeitsmarkt/wo-sind-die-frauen-in-der-it>. n.d.. – Zuletzt abgerufen am: 15.03.2024
- [Mai 2024] MAI, Jochen: *Selbsttäuschung: Unsere Wahrnehmung ist ein Trugbild*. URL: <https://karrierebibel.de/selbsttaeuschung/>. 2024. – Zuletzt abgerufen am: 17.03.2024
- [Matheretter n.d.] MATHERETTER: *Konkave und konvexe Polygone*. URL: <https://www.matheretter.de/wiki/konkave-konvexe-polygone>. n.d.. – Zuletzt abgerufen am: 01.03.2024
- [Mau und Haase 2021] MAU, Katharina ; HAASE, Martin: *Im Bio-Rhythmus arbeiten: 5 Tipps für mehr Produktivität*. URL: <https://focusbusiness.de/magazin/biorhythmus-arbeiten-innere-uhr>. 2021. – Zuletzt abgerufen am: 17.03.2024
- [Mayer 2009] MAYER, Richard E.: *Multimedia Learning: Second Edition*, Cambridge University Press, 2009, S. 1–3, 19
- [Moritz 2021] MORITZ, Lisa M.: *Sketchnotes in der IT: Abstrakte Themen mit Leichtigkeit visualisieren*, dpunkt.verlag, 2021, S. 18
- [Nicolasreto 2023] NICOLASRETO: *What are game assets in 2023?* URL: <https://tips.clip-studio.com/de-de/articles/8608?org=1>. 2023. – Zuletzt abgerufen am: 22.01.2024
- [Nielsen 2000] NIELSEN, Jakob: *Why You Only Need to Test with 5 Users*. URL: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>. 2000. – Zuletzt abgerufen am: 15.03.2024
- [Nielsen 2012] NIELSEN, Jakob: *Usability 101: Introduction to Usability*. URL: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>. 2012. – Zuletzt abgerufen am: 08.03.2024
- [Ortmann 2021] ORTMANN, Dr. M.: *Konfundierende Variablen*. URL: <https://ortmann-statistik.de/glossar/konfundierende-variablen/>. 2021. – Zuletzt abgerufen am: 17.03.2024
- [Panagiotidi 2021a] PANAGIOTIDI, Dr. M.: *Standardized usability questionnaires: post-task measures*. URL: <https://uxpsychology.substack.com/p/standardized-usability-questionnaires-ccb>. 2021. – Zuletzt abgerufen am: 14.03.2024
- [Panagiotidi 2021b] PANAGIOTIDI, Dr. M.: *Standardized usability questionnaires: which one to use?* URL: https://uxpsychology.substack.com/p/standardized-usability-questionnaires?utm_campaign=post&utm_source=direct. 2021. – Zuletzt abgerufen am: 14.03.2024

- [Pearson 2014] PEARSON, Dan: *Monument Valley and Hearthstone win at Unity awards*. URL: <https://www.gamesindustry.biz/monument-valley-and-hearthstone-win-at-unity-awards>. 2014. – Zuletzt abgerufen am: 22.01.2024
- [Personal-Wissen 2023] PERSONAL-WISSEN, Redaktion: *So kannst du Ideen einfach visualisieren*. URL: <https://www.personal-wissen.net/wissen/kreative-visualisierung-einer-strategie-1638/>. 2023. – Zuletzt abgerufen am: 29.01.2024
- [Pfeiffer 2021] PFEIFFER, Franziska: *Interne Validität - Einfache Definition mit mehreren Beispielen*. URL: <https://www.scribbr.de/methodik/interne-validitaet/>. 2021. – Zuletzt abgerufen am: 17.03.2024
- [Pfeiffer 2023] PFEIFFER, Franziska: *Externe Validität - Einfache Definition mit mehreren Beispielen*. URL: <https://www.scribbr.de/methodik/externe-validitaet/>. 2023. – Zuletzt abgerufen am: 17.03.2024
- [Pubudu 2021] PUBUDU, Nisal: *Understanding The Memento Design Pattern*. URL: <https://medium.com/nerd-for-tech/understanding-memento-design-pattern-5c4f09be639>. 2021. – Zuletzt abgerufen am: 28.02.2024
- [Pötzelberger 2006] PÖTZELBERGER, Werner: *Das Polygon - Definition 2d*. URL: <http://www.dma.ufg.ac.at/app/link/Allgemein%3AModule/module/13152/sub/13193>. 2006. – Zuletzt abgerufen am: 01.03.2024
- [Qualtrics.XM 2024] QUALTRICS.XM: *Likert-Skala: Definition, Beispiel und Vorteile*. URL: <https://www.qualtrics.com/de/erlebnismanagement/marktforschung/likert-skala/>. 2024. – Zuletzt abgerufen am: 11.03.2024
- [Rauer 2011] RAUER, Matthias: *Quantitative Usability-Analysen mit der System Usability Scale (SUS)*. URL: <https://seibert.group/blog/2011/04/11/usability-analysen-system-usability-scale-sus/>. 2011. – Zuletzt abgerufen am: 14.03.2024
- [Rohde 2014] ROHDE, Mike: *Das Sketchnote Handbuch: Der illustrierte Leitfaden zum Erstellen visueller Notizen*, mitp, 2014, S. 13, 15–18, 149–151, 153
- [Rummel 2016] RUMMEL, Bernhard: *System Usability Scale – jetzt auch auf Deutsch*. URL: <https://community.sap.com/t5/additional-blogs-by-sap/system-usability-scale-jetzt-auch-auf-deutsch/ba-p/13487686>. 2016. – Zuletzt abgerufen am: 14.03.2024
- [Saietskyi n.d.] SAIETSKYI, Kostiantyn: *2D/3D Paint - XDPaint - Dokumentation*. URL: <https://unitymedved.bitbucket.io/XDPaint/3.1/Docs/index.html>. n.d.. – Zuletzt abgerufen am: 01.02.2024
- [Sauro 2011] SAURO, Jeff: *Measuring Usability with the System Usability Scale (SUS)*. URL: <https://measuringu.com/sus/>. 2011. – Zuletzt abgerufen am: 14.03.2024
- [Sauro 2018] SAURO, Jeff: *5 Ways to Interpret a SUS Score*. URL: <https://measuringu.com/interpret-sus-score/>. 2018. – Zuletzt abgerufen am: 08.03.2024

- [Schuhmann 2014] SCHUHMANN: *Hearthstone: Preisgekrönt von den Engine-Machern*. URL: <https://mein-mmo.de/hearthstone-preisgekraent-von-den-engine-machern258/>. 2014. – Zuletzt abgerufen am: 22.01.2024
- [Schulz n.d.] SCHULZ, Dr. C.: *Das Datenmodell – die Strukturen eines Anwendungsbereiches eindeutig erfassen*. URL: <https://www.palladio-consulting.de/datenmodell/>. n.d.. – Zuletzt abgerufen am: 23.01.2024
- [Seifert und Wislaug 2017] SEIFERT, Carsten ; WISLAUG, Jan: *Spiele entwickeln mit Unity 5: 2D- und 3D-Games mit Unity und C# für Desktop, Web & Mobile.*, HANSER, 2017, S. 29, 31, 64, 70, 78, 109–110, 113, 117–121, 138, 149, 226–227, 232–234, 236–238, 256, 327–328, 372–377, 383, 403
- [Stephan_B 2019] STEPHAN_B: *Beitrag im Unity Forum zum Thema: "Calculate Width of a Text before/without assigning it to a TMP object"*. URL: <https://forum.unity.com/threads/calculate-width-of-a-text-before-without-assigning-it-to-a-tmp-object.758867/#post-5057900>. 2019. – Zuletzt abgerufen am: 24.02.2024
- [Studyflix 2024] STUDYFLIX: *Raute*. URL: <https://studyflix.de/mathematik/raute-3156>. 2024. – Zuletzt abgerufen am: 01.03.2024
- [Tesch 2019] TESCH, Guido: *Die neue ISO 9241-11 (2018) "Usability: Definitions and concepts"*. URL: <https://www.procontext.de/aktuelles/2019/12/die-neue-iso-9241-11-usability-definitions-and-concepts.html>. 2019. – Zuletzt abgerufen am: 08.03.2024
- [Themann 2015] THEMANN, Tim: *Text und Bild - ein tolles Paar!* URL: <https://die-computermaler.de/text-und-bild-ein-tolles-paar/>. 2015. – Zuletzt abgerufen am: 29.01.2024
- [Toor 2020] TOOR, Meena: *What is acquiescence bias and how can you stop it?* URL: <https://www.qualtrics.com/blog/avoiding-the-yes-bias/>. 2020. – Zuletzt abgerufen am: 14.03.2024
- [Unity_Technologies 2023a] UNITY_TECHNOLOGIES: *Color Gradients - TextMeshPro*. URL: <https://docs.unity3d.com/Packages/com.unity.textmeshpro@4.0/manual/ColorGradients.html>. 2023. – Zuletzt abgerufen am: 25.01.2024
- [Unity_Technologies 2023b] UNITY_TECHNOLOGIES: *Rich Text*. URL: <https://docs.unity3d.com/Packages/com.unity.textmeshpro@2.2/manual/RichText.html>. 2023. – Zuletzt abgerufen am: 24.02.2024
- [Unity_Technologies 2024a] UNITY_TECHNOLOGIES: *Renderer*. URL: <https://docs.unity3d.com/ScriptReference/Renderer.html>. 2024. – Zuletzt abgerufen am: 01.02.2024
- [Unity_Technologies 2024b] UNITY_TECHNOLOGIES: *Rendering Mode*. URL: <https://docs.unity3d.com/Manual/StandardShaderMaterialParameterRenderingMode.html>. 2024. – Zuletzt abgerufen am: 23.02.2024
- [Walther 2022] WALTHER, Björn: *Kendall-Tau-Korrelationskoeffizient in SPSS berechnen*. URL: <https://bjoernwalther.com/kendall-tau-korrelationskoeffizient-in-spss-berechnen/>. 2022. – Zuletzt abgerufen am: 18.03.2024

- [Whitenton 2013] WHITENTON, Kathryn: *Minimize Cognitive Load to Maximize Usability*. URL: <https://www.nngroup.com/articles/minimize-cognitive-load/>. 2013. – Zuletzt abgerufen am: 31.01.2024
- [Wikipedia 2024a] WIKIPEDIA: *Entity-Relationship-Modell*. URL: <https://de.wikipedia.org/wiki/Entity-Relationship-Modell>. 2024. – Zuletzt abgerufen am: 23.01.2024
- [Wikipedia 2024b] WIKIPEDIA: *Skalenniveau*. URL: <https://de.wikipedia.org/wiki/Skalenniveau>. 2024. – Zuletzt abgerufen am: 18.03.2024
- [Wikipedia 2024c] WIKIPEDIA: *Trapez (Geometrie)*. URL: [https://de.wikipedia.org/wiki/Trapez_\(Geometrie\)](https://de.wikipedia.org/wiki/Trapez_(Geometrie)). 2024. – Zuletzt abgerufen am: 01.03.2024