

Virtuelle-Realität-Interaktion in SEE

Bachelorarbeit

Lysander Gawenda

Matrikelnummer: 6117865

23. September 2024



Fachbereich 3 — Mathematik und Informatik
Studiengang Informatik

1. Gutachter: Prof. Dr. Rainer Koschke
2. Gutachter: Dr. Hui Shi

ZUSAMMENFASSUNG

In der vorliegenden Bachelorarbeit wird untersucht, wie sich die Usability der Interaktionen in der Code-Visualisierungs-Software *SEE* zwischen Desktop-Computern und VR-Hardware unterscheidet. Die Interaktionen für die Desktop-Hardware existierte dabei schon und die Interaktionen für VR-Hardware wurden im Rahmen dieser Arbeit implementiert. In der eigentlichen Studie haben Probanden ($n = 12$) jeweils Aufgaben mit der VR- und mit der Desktop-Hardware gelöst. Um die Usability der Interaktionen zu ermitteln, wurden zwei Fragebögen genutzt. Einmal der After-Scenario-Questionnaire (ASQ) nach jeder Aufgabe und die System Usability Scale (SUS) nach jedem Hardware-Abschnitt. Außerdem wurde die benötigte Zeit und die Korrektheit hinsichtlich der bearbeiteten Aufgaben gemessen. Die Evaluation zeigte dabei, dass die Desktop-Hardware in allen Punkten signifikant bessere Ergebnisse bekam, als die VR-Hardware. Es stellte sich jedoch auch heraus, dass beide Hardware-Alternativen noch deutliches Potenzial zur Verbesserung haben, da die VR-Hardware im inakzeptablen Bereich und die Desktop-Hardware im marginalen Akzeptanzbereich liegt. Abschließend wurden noch Erweiterungsmöglichkeiten für *SEE* aufgezeigt, die die Usability der Interaktionen bei beiden Hardware-Alternativen zukünftig verbessern könnten.

ERKLÄRUNG

Ich versichere, diese Arbeit — 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 23. September 2024

Lysander Gawenda

DANKSAGUNG

Ich möchte mich zuerst einmal bei Prof. Dr. Rainer Koschke bedanken, da er immer Zeit für meine Fragen hatte und mir die VR-Hardware für den Zeitraum meiner Bachelorarbeit zur Verfügung stellte. Ebenso möchte ich mich bei Falko bedanken, da er auch immer offen für Fragen war. Außerdem danke ich allen Probanden, ohne die diese Arbeit nicht möglich gewesen wäre und natürlich auch allen Korrekturlesern.

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	Einleitung	1
2	Grundlagen	3
2.1	SEE	3
2.2	XR Toolkit	3
2.2.1	Action-based behaviors	3
2.2.2	Interactors	4
2.2.3	Interactables	5
2.3	Verwandte Arbeiten	5
3	Ziele	7
3.1	Desktop-Steuerung	8
4	Implementierung	9
4.1	Umsetzung	9
4.2	Controller-Klasse	9
4.3	Context-Menü	10
4.4	Mode-Menü	10
5	Evaluation	15
5.1	Ziel und Hypothesen	15
5.1.1	Korrektheit	15
5.1.2	Bearbeitungszeit	15
5.1.3	Usability	15
5.2	Aufbau	16
5.3	Fragebogen	17
5.3.1	Demographischer Fragebogen	17
5.3.2	Post-Task-Fragebogen	17
5.3.3	Post-Study-Fragebogen	18
5.4	Aufgabenstellung	18
5.4.1	Aufgabe 1	18
5.4.2	Aufgabe 2	19
5.4.3	Aufgabe 3	20
5.5	Umsetzung	21
5.6	Auswertung	23
5.6.1	Demographische Angaben	23
5.6.2	Korrektheit	27
5.6.3	Bearbeitungszeit	32
5.6.4	Usability	37
5.6.5	Korrelationen	43
5.6.6	Kommentare der Teilnehmer	47
5.7	Threats to Validity	50
5.7.1	Interne Validität	50
5.7.2	Externe Validität	51
6	Fazit und Ausblick	53

6.1	Fazit	53
6.2	Ausblick	54
A	Glossar	55
B	Akronyme	57
C	Abbildungsverzeichnis	59
D	Tabellenverzeichnis	61
	Literaturverzeichnis	61

EINLEITUNG

Bisher wurde *Software Engineering Experience* (SEE) fast ausschließlich für die Desktop-Hardware entwickelt. Mit der *Virtual-Reality* (VR)-Hardware konnte sich nur in der virtuellen Welt von SEE bewegt, bzw. teleportiert werden. Alle anderen Interaktionen, die mit der Desktop-Hardware selbstverständlich sind, fehlten bis jetzt in VR. Dabei ist es ein Ziel von SEE, eine virtuelle Welt für Software-Entwickler zu schaffen, in der sie sich treffen und gemeinsam über Code austauschen können. Um diesen Ziel näher zukommen ist es Teil dieser Arbeit die Hauptinteraktionen von SEE für VR-Hardware zu realisieren. Im Anschluss soll dann die Usability der beiden Hardware-Alternativen verglichen werden, indem eine Usability-Studie durchgeführt wird, bei der Probanden beide Hardware-Alternativen benutzen und bewerten. Daraus ergibt sich auch die Forschungsfrage dieser Arbeit: „Wie unterscheidet sich die Usability der Interaktionen in SEE zwischen Desktop-Computern und VR-Hardware?“

SEE: Die Software Engineering Experience ist eine virtuelle Welt, in der sich

Softwareentwickler treffen können und Software in Form von Code-Cities dargestellt wird.

VR: Virtual-Reality ist eine computergenerierte Realität, mit der User durch eine VR-Brille interagieren können.

GRUNDLAGEN

2.1 SEE

SEE steht für Software-Engineering-Experience und ist die Software, die im Rahmen dieser Arbeit erweitert wird. SEE wurde mithilfe von *Unity* entwickelt, um eine virtuelle Welt zu schaffen, in der sich Softwareentwickler treffen und sich gemeinsam über Code austauschen können. Dabei wird der Code in Form von sogenannten *Code-Cities* dargestellt.

2.2 XR TOOLKIT

Da wir¹ für die Interaktionen auf die tatsächlichen Inputs der Hardware zugreifen müssen (in diesem Fall die Controller) und mit diesen auch mit der virtuellen Welt interagieren wollen, brauchen wir eine Schnittstelle um dies zu erreichen. Dafür gibt es von Unity eine eigene API (Application Programming Interface), welche einerseits eine Reihe von Interactors und Interactables bereitstellt, aber auch einen Interaction Manager der die beiden Komponenten verbindet. Außerdem verwendet XR Toolkit sogenannte Action-based behaviors, welche im Anschluss noch einmal genauer erklärt werden. Alternativ hätten auch das Asset „Auto Hand - VR Interaction“ genutzt werden können, welches auch für die Interaktionen in VR entwickelt wurde. Es hat den Vorteil, dass die Controller als Hände dargestellt werden und Interaktionen wie Greifen oder Zeigen darauf angepasst sind, wodurch dies eine immersivere Erfahrung bietet. Es wurde sich trotzdem für das XR Toolkit entschieden, da es die hauseigene Schnittstelle von Unity ist und somit langfristige Unterstützung und Kompatibilität mit der restlichen Funktionalität von Unity gewährleistet.

2.2.1 Action-based behaviors

Action-based behaviors werden genutzt, um den Input von ein oder mehreren Controls, anstatt direkt von einem spezifischen Controller zu lesen. Durch diese Herangehensweise wird von der eigentlichen Hardware abstrahiert und es muss nicht mehr im Code selber für jede Hardware spezifisch programmiert werden. Stattdessen gibt es eine Action-Map, in der Input Actions definiert werden können. Bspw.

¹ Mit „Wir“ wird in dieser Arbeit der Leser mit einbezogen. Die Arbeit wurde jedoch von einer einzelnen Person erstellt.

Unity: Unity ist eine Spiele-Engine, mit der es möglich ist, 2D- und 3D-Spiele aber auch interaktive Simulationen zu erstellen.

Code-City: Code-Cities sind dreidimensionale Tree-Maps, durch die eine Software und ihre einzelnen Bestandteile dargestellt werden können. Ihr Abbild erinnert dabei an eine Stadt, in der die Gebäude einzelne Softwarekomponenten darstellen. Die Größe, aber auch andere Attribute der Gebäude wird dabei durch Software-Metriken dargestellt.

könnte eine Input Action mit dem Namen „Greifen“ definiert werden, durch die es im Spiel möglich ist, Objekte zu greifen. In der Action-Map wird dazu dann festgelegt, durch welche Eingaben (z.B. A-Knopf auf dem rechten Controller, oder X-Knopf auf dem linken Controller) diese Aktion ausgelöst werden kann. Diese Eingaben können auch für spezifische Controller angepasst werden, sodass „Greifen“ für die Hardware einer Meta-Quest auf einem anderen Knopf gelegt werden kann als z.B. für die Hardware einer HTC-Vive. Außerdem kann die Zuordnung von Eingaben flexibel geändert werden, ohne dass der eigentliche Code verändert werden muss. Das ist auch praktisch, wenn der User selber die Steuerung nach seinen Wünschen anpassen möchte.

2.2.2 Interactors

Interactors sind Komponenten die *GameObjects* zugewiesen werden können. In unserem Fall werden sie den Controller-GameObjects zugewiesen. Sie sorgen dabei dafür, dass der User mithilfe seiner Controller mit der Spielwelt interagieren kann. Dafür werden sogenannte *Unity-Events* genutzt. Bspw. gibt es die Events „Hover Entered“ und „Hover Exited“, bei denen spezifiziert werden kann, was passieren soll, wenn der User anfängt über ein Interactable zu hovern, bzw. wenn er damit aufhört. Außerdem gibt es verschiedene Arten von Interactors, die jeweils unterschiedliche Funktionalitäten mit sich bringen. Im Folgenden werden wir eine Reihe von diesen vorstellen, die auch im Rahmen dieser Arbeit benutzt wurden, um die Liste nicht zu lang zu machen.

GameObject:
GameObjects sind die Grundbausteine in Unity. Sie werden mit Komponenten und Variablen zum Leben erweckt und können in der Spielwelt alles von Beleuchtung oder Wasser zu spielbaren Avataren darstellen.

Unity-Event:
Unity-Events sind ereignisgesteuert, d.h. dadurch ist es möglich auf Aktionen in der Spielwelt zu reagieren. Dabei ist die Besonderheit, dass keine festen Abhängigkeiten im Code entstehen müssen (außer es ist so gewollt) und Methoden dynamisch im Unity-Editor einem Event zugewiesen werden können.

- **Poke Interactor**

Mit diesem Interactor kann der User mit GameObjects interagieren, indem er diese berührt oder anstößt (= „Poke“). Dadurch können z.B. Knöpfe gedrückt werden, oder aber auch wie bei echten Touchscreens, Toucheingaben auf Benutzeroberflächen getätigt werden.

- **Ray Interactor**

Der Ray Interactor besitzt einen Laserstrahl, durch den der User auch auf Distanz mit interactables und Benutzeroberflächen interagieren kann.

- **Near-Far Interactor**

Dieser Interactor kann automatisch zwischen dem Direct Interactor (ermöglicht direkte Interaktionen mit Objekten in unmittelbarer Nähe, z.B. durch Berührungen oder Greifen.) und Ray Interactor wechseln, je nachdem wie weit ein Interactable entfernt ist.

2.2.3 Interactables

Interactables sind Komponenten, die GameObjects zugewiesen werden können. Dadurch wird es für den User möglich, mit diesen GameObjects mithilfe der Interactors zu interagieren. Wobei hier wieder durch den Entwickler, mithilfe der schon beschriebenen Unity-Events, individuelle Interaktionen erstellt werden können. Dabei gibt es unterschiedliche Arten von Interactables, wobei wir uns im Rahmen dieser Arbeit für zwei bestimmte Interactables interessieren.

- **Grab Interactable**
Diese Komponente ermöglicht es dem User, das jeweilige GameObject zu greifen und zu bewegen. Dafür muss außerdem ein Rigidbody (Komponente, durch die ein GameObject auf Physik reagiert) auf dem GameObject existieren.
- **Simple Interactable**
Diese Komponente ist, wie der Name schon sagt, recht simple und erkennt, wenn das entsprechende GameObject mit einem Interactor interagiert. Die restliche Logik, also was passiert, wenn ein Interactor über dieses GameObject hovers, oder es selektiert, ist dem Entwickler überlassen und kann individuell angepasst werden.

2.3 VERWANDTE ARBEITEN

Im Folgenden werden zwei Arbeiten vorgestellt, welche ähnliche Interaktionsformen untersuchen wie wir es in dieser Arbeit tun. Dabei wurden auch jeweils eine VR- mit einer Desktop-Alternative verglichen und die Ergebnisse präsentiert.

[Berkman u. a. \(2020\)](#) (S. 180) zeigten in ihrer Arbeit auf, dass bei dem Vergleich des Spielerlebnisses eines Puzzlespiels im Vergleich zwischen Desktop und VR, VR zwar keine höhere Spielerlebnis-Zufriedenheit unter den Probanden festgestellt werden konnte, aber eine höhere *Spatial presence* im Vergleich zu der Desktop-Alternative. Außerdem stellte sich heraus, dass Probanden beim Benutzen der VR-Alternative zwar eine höhere körperliche Arbeitsbelastung empfanden, es aber allgemein keinen Unterschied gab, was die Bearbeitungszeit und Erfolgsrate angeht.

In einer anderen Studie aus dem medizinischen Bereich ging es um das Entfernen von Gewebepflockaden in einer 3D-Umgebung, welche mit Unity erstellt wurde. Dabei wurde die Software hinsichtlich der Genauigkeit, Bearbeitungszeit und Zufriedenheit bei der Bearbeitung von Aufgaben mit VR- und Desktop-Alternative untersucht. Dabei stellte sich heraus, dass bei Aufgaben bei denen Objekte gedreht werden mussten, Probanden mit der VR-Alternative dreimal so schnell waren

Spatial presence:
Mit *Spatial presence* (= Räumliche Präsenz) ist das Gefühl gemeint, dass der User sich tatsächlich in der virtuellen Welt befindet und mit dieser interagieren kann [Caroux \(2023\)](#).

und ein Drittel mal genauere Ergebnisse bezogen auf eine Interaktion lieferten. Im Allgemeinen wurde jedoch die Desktop-Alternative als am besten geeignete Hardware für die eigentlichen User gewählt, da die Genauigkeit und Bearbeitungszeit den Ansprüchen der Experten entspricht. Außerdem sei die Hardware für die Desktop-Alternative günstiger, besser in die Arbeitsumgebung zu integrieren und die Bedienung einfacher zu erlernen [Bueckle u. a. \(2021\)](#) (S. 1).

ZIELE

In diesem Abschnitt sollen kurz die Hauptinteraktionen definiert werden, die im Rahmen dieser Bachelorarbeit implementiert werden. Dabei ist anzumerken, dass die Interaktionen schon für Desktop-Hardware existieren, bedeutet, die Logik hinter den Interaktionen existiert schon und muss nicht von Grund auf neu implementiert werden. Es wird also auf diese Logik zugegriffen und erweitert, sodass diese auch für VR-Hardware zu benutzen ist. Zum einen gibt es die direkten Interaktionen mit den Code-Cities. Hierzu wird einmal kurz aufgelistet, welche Interaktionen in dieser Arbeit implementiert werden:

- **Move-Aktion**
Mit dieser Aktion lassen sich *Nodes* in der Code-City verschieben.
- **Rotate-Aktion**
Mit dieser Aktion lassen sich Nodes in der Code-City rotieren.
- **Hide-Aktion**
Mit dieser Aktion lassen sich Kanten und die damit verbundenen Nodes ausblenden.
- **New *Edge*-Aktion**
Mit dieser Aktion können neue Kanten hinzugefügt werden.
- **New Node-Aktion**
Mit dieser Aktion können neue Nodes hinzugefügt werden.
- **Delete-Aktion**
Mit dieser Aktion können Kanten und Nodes gelöscht werden.
- **Show Code-Aktion**
Mit dieser Aktion kann der Quellcode für Nodes geöffnet werden.
- **Draw-Aktion**
Mit dieser Aktion kann in der dreidimensionalen Welt von SEE gemalt werden.
- **Accept Divergence-Aktion**
Mit dieser Aktion kann eine Kante zu einem *Reflexion-Graph* hinzugefügt werden, die eine derzeit abweichende Implementierungsabhängigkeit ermöglicht.

Zum anderen gibt es auch noch die Interaktionen mit Menüs. Hier soll es möglich sein, mit den folgenden Menüs zu interagieren:

Node: Ein Node ist ein einzelnes Element, innerhalb der Code-City, also z.B. eine Klasse oder eine Methode eines Softwareprojektes. Seine Erscheinungsform wird durch vorher festgelegte Metriken bestimmt. Bspw. könnte die Höhe die Anzahl an Codezeilen sein.

Edge: Eine Edge (=Kante) verbindet immer zwei Nodes miteinander und stellt dabei Abhängigkeiten zwischen diesen dar.

Reflexion-Graph: Mit dem Reflexion-Graph kann eine Implementierung gegen die eigentlich festgelegte Architektur verglichen werden.

- **Context-Menü**
Dieses Menü öffnet sich, wenn der User auf ein Node klickt. In dem Menü sind verschiedene Interaktionsmöglichkeiten in Bezug auf diese Node aufgelistet.
- **Tree View**
Die Tree View wird benutzt um die Code-City als Baumstruktur in Form von Ordnern und Dateien ähnlich wie bei Betriebssystemen anzuzeigen.
- **Code Window**
Dies hängt mit der Aktion „Show Code“ zusammen und zeigt wie schon erwähnt, den Quellcode zu einem Node an, ähnlich wie in herkömmlichen IDEs.
- **Mode-Selection-Menü**
In diesem Menü kann der User auswählen, in welchen Modus er sich befinden möchte. Bspw. könnte er Move auswählen um anschließend Nodes verschieben zu können.

Zudem soll es noch möglich sein, Aktionen rückgängig zu machen und wiederherstellen zu können.

3.1 DESKTOP-STEUERUNG

In der Desktop-Version von SEE kann der Avatar über die Tasten „W“, „A“, „S“, „D“ bewegt werden. Mit „Q“ und „E“ kann der Avatar außerdem nach oben und unten bewegt werden. Mit der „Leertaste“ kann das Mode-Menü geöffnet werden, in welchem Aktionen wie „Move-Aktion“ oder „Hide-Aktion“ ausgewählt werden können. Diese Aktionen können, wenn alle Bedingungen erfüllt sind, mit der linken Maustaste ausgeführt werden. Mit der rechten Maustaste kann zudem einerseits, wenn gleichzeitig die Maus bewegt wird, die Kamera bewegt werden und andererseits kann beim Klicken auf ein Objekt in der Code-City das Context-Menü geöffnet werden. Wenn auf eine Code-City gezeigt wird und dabei das Mausrad betätigt wird, kann in die Code-City rein und rausgezoomt werden. Mit „Tab“ kann zudem separat die TreeView geöffnet werden. Es gibt noch weitere Tastenbelegungen, die aber speziellerer Natur sind und an dieser Stelle weggelassen wurden. Dies hat den Grund, dass sie nicht in VR umgesetzt, aber auch in der Studie nicht benutzt wurden und an dieser Stelle eher verwirren könnten.

IMPLEMENTIERUNG

4.1 UMSETZUNG

Da die Interaktionen schon für die Desktop-Hardware vorhanden sind, muss der bestehende Code für jede Interaktion jeweils angepasst werden. Darüber hinaus bekommen die Objekte in der Code-City die Simple Interactable-Komponente, damit der User mit ihnen interagieren kann. Ein großer Teil der Interaktionen findet außerdem über Menüs statt, wie die TreeView, das CodeWindow oder auch das MetricWindow. Um diese Funktionalitäten der Menüs in VR nutzen zu können, aber das VR-Erlebnis nicht zu sehr Desktop-Like zu gestalten, wurde ein Tablet entwickelt [4.2a](#), auf dem die Menüs sichtbar sind. Dieses Menü kann der User greifen und wie in der echten Welt mit sich herumtragen, was durch die Grab Interactable-Komponente möglich ist. Im Fall des MetricWindow und der TreeView erscheint eine virtuelle Tastatur, wenn die Suchleiste angeklickt wird. Diese Tastatur hat die allernötigsten Funktionen, um in den beiden Menüs nach Einträgen zu suchen. Es gibt eine „Hochstell-Taste“, um Großbuchstaben und Zeichen zu schreiben. Zudem gibt es eine „Leer-Taste“, eine „Löschen-Taste“ und eine „Enter-Taste“ mit den bekannten Funktionen. Wenn die Enter-Taste gedrückt wird, verschwindet außerdem die Tastatur.

4.2 CONTROLLER-KLASSE

Für den Input der Controller wurde eine zentrale Controller-Klasse entwickelt, welche die Inputs der einzelnen Controller-Knöpfe entgegennimmt und verarbeitet, sodass wenn alle Anforderungen erfüllt sind, die gewünschten Aktionen wie Move, oder New-Edge ausgeführt werden. Dies wurde mit den schon beschriebenen Konzepten der Action-Map, Unity-Events und Input Action References erreicht. Mit den Unity-Events „On Hover Entered“ und „On Hover Exited“ wird jeweils das aktuell anvisierte GameObject als Referenz zwischengespeichert und für andere Aktionen verfügbar gemacht, falls diese ausgelöst werden. Außerdem werden im Fall von Nodes in der Code-City, diese als „Hovered“ hervorgehoben und deren Label angezeigt. Wenn eine spezifische Aktion ausgeführt wird, werden Booleans benutzt um bspw. der Move-Aktion mitzuteilen, dass der User eine Eingabe getätigt hat und die Aktion ausgeführt werden kann. Außerdem gibt es noch zwei

etwas besondere Aktionen, die mehr Interaktion vom User abverlangen und ein immersiveres Erlebnis bieten sollen:

- **Rotate-Aktion:** Bei dieser Aktion erscheint vor dem User ein dreidimensionaler Drehknopf. Dieser kann vom User gegriffen und durch den Raum bewegt werden, was wieder durch die Grab Interactable-Komponente möglich wird. Wenn der User auf eine Node selektiert und danach den Knopf dreht, wird auch die Node rotiert.
- **Draw-Aktion:** Bei dieser Aktion erscheint ein dreidimensionaler Stift vor dem User. Dieser kann gegriffen und bewegt werden, da er eine Grab Interactable-Komponente besitzt. Wenn der User den Stift greift, erscheint am Ende des Stiftes Farbe, sodass damit gezeichnet werden kann.

4.3 CONTEXT-MENÜ

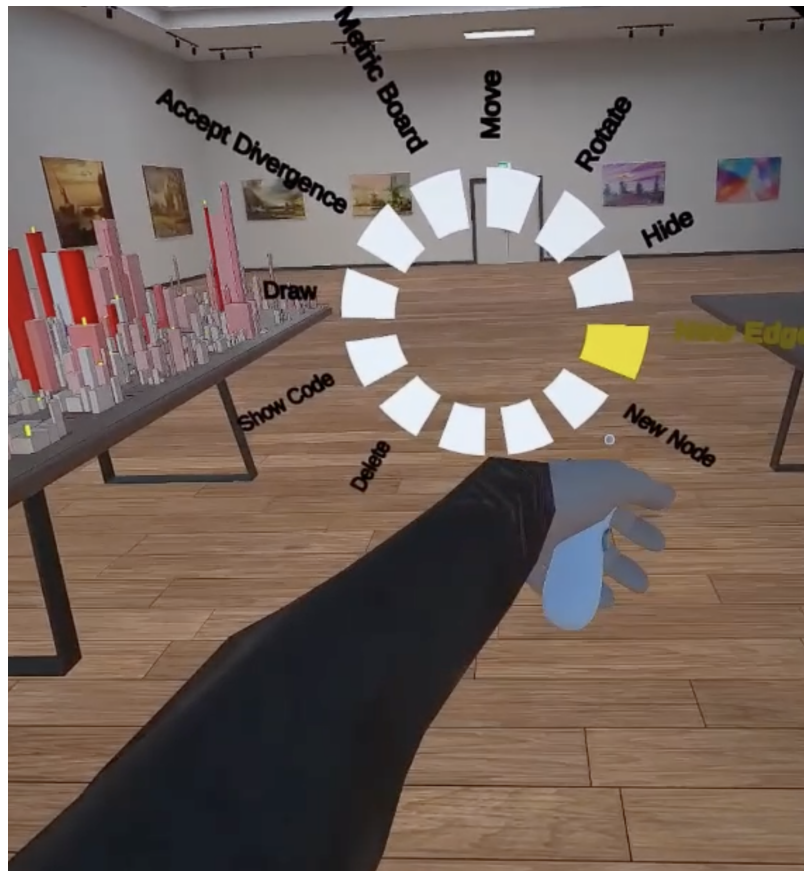
Da sich das Context-Menü in der Desktop-Version von SEE an der Stelle vom Maus-Zeiger öffnet, wenn die rechte Maus-Taste gedrückt wird, sollte diese Funktion in VR ähnlich sein. Da es in der VR-Version kein *Canvas* gibt, der den ganzen Bildschirm einnimmt (wie es bei der Desktop-Version der Fall ist), ist es schwierig, das Context-Menü an dem Ort erscheinen zu lassen, wo der Laserpointer des rechten Controllers hinzeigt. Stattdessen gibt es ein stationäres Context-Menü, welches sich direkt vor dem rechten Controller befindet [4.3a](#). Dies wird aktiviert, wenn der User den Thumb-Button des rechten Controllers betätigt, während er auf ein Objekt in der Code-City zielt. Indem der User nun mit dem linken Controller über dieses Context-Menü wischt, kann er hoch und runterscrollen. Wenn er einen Menü-Eintrag anstößt (= „Poke“), wählt er diesen aus.

Canvas: Ein Canvas ist in Unity ein Container, der verschiedene UI-Elemente aufnehmen kann.

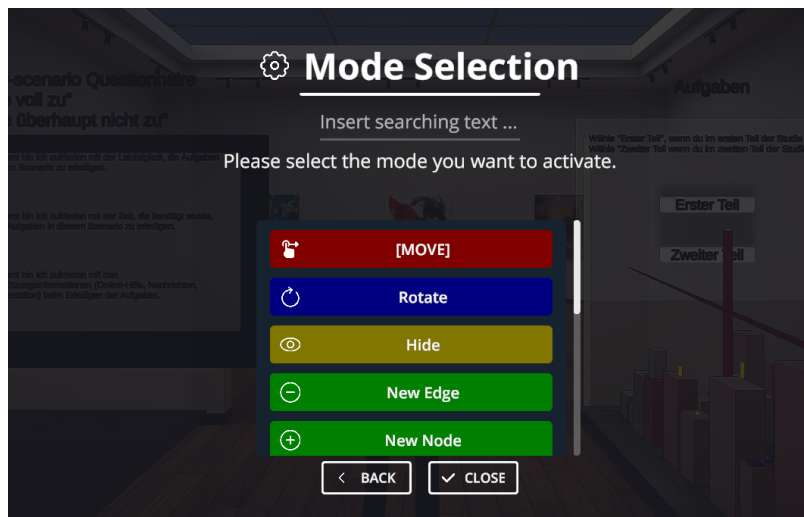
4.4 MODE-MENÜ

Das Mode-Menü ist ein zentrales Menü in SEE, da dort die Hauptaktionen für die Interaktionen mit der Code-City ausgewählt werden können. In der Desktop-Version öffnet es sich, wenn der User die Leertaste drückt. Hier gäbe es entweder die Option es genau so für VR zu lassen, mit der einzigen Änderung, dass es nicht die Leertaste wäre und es auf dem Tablet angezeigt wird. Die andere etwas immersivere Option ist, das Menü für VR neu zu gestalten und die Möglichkeiten an verschiedenen Interaktionsformen in VR zu nutzen. Die letztere Option wurde gewählt und es wurde ein rundes Menü entwickelt, welches sich vor dem linken Controller öffnet. Öffnen lässt es sich mit dem A-Knopf des rechten Controllers. Der User sollte dafür den linken Arm von sich weg strecken, um das Menü gut sehen zu können. Indem er den linken

Controller kreisförmig bewegt, kann er über verschiedene Einträge des Menüs hovern. Wenn er nun wieder den A-Knopf des rechten Controllers betätigt (wurde nach der Evaluation auf die Schultertaste gelegt), schließt sich das Menü und der User befindet sich in der ausgewählten Aktion.

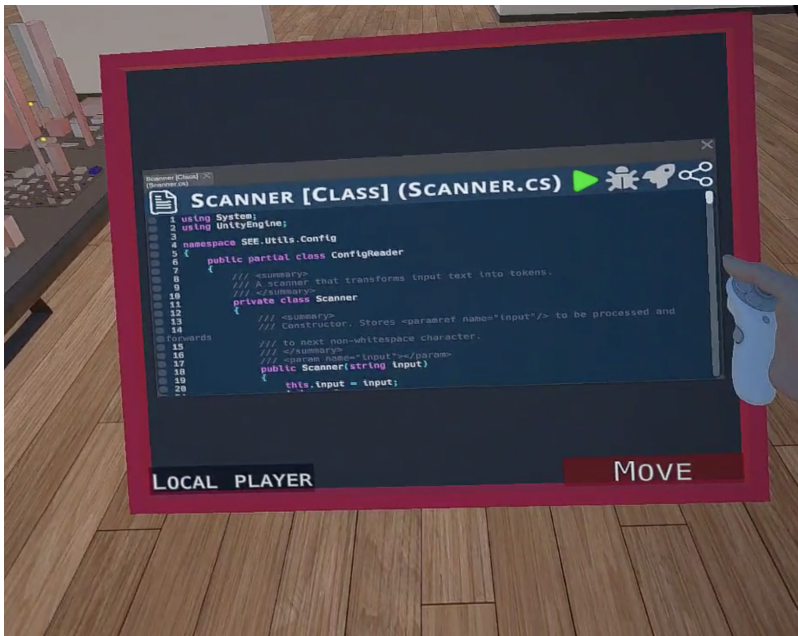


(a) Mode-Menü in VR



(b) Mode-Menü mit der Desktop-Hardware

Abbildung 4.1: Vergleich des Mode-Menüs zwischen VR und Desktop



(a) Tablet in VR mit dem CodeWindow



(b) CodeWindow mit der Desktop-Hardware

Abbildung 4.2: Vergleich des CodeWindows zwischen VR und Desktop



(a) Context-Menü in VR



(b) Context-Menü mit der Desktop-Hardware

Abbildung 4.3: Vergleich des Context-Menü zwischen VR und Desktop

EVALUATION

5.1 ZIEL UND HYPOTHESEN

In diesem Kapitel wird erläutert, welche Hypothesen aufgestellt und wie diese in der eigentlichen Studie getestet werden, um die grundlegende Forschungsfrage dieser Arbeit beantworten zu können. Da wir mit der Forschungsfrage: „Wie unterscheidet sich die Usability der Interaktionen in *SEE* zwischen Desktop-Computern und VR-Hardware?“ herausfinden wollen, in welchen Punkten genau sich die Usability unterscheidet, stellen wir im Folgenden gerichtete Hypothesen auf, da uns nicht nur interessiert **ob** sich die Usability der Interaktionen unterscheidet, sondern auch **wie** sie sich unterscheidet.

5.1.1 Korrektheit

Nullhypothese H_{c0} : „Die Korrektheit der Antworten bei der VR-Hardware ist kleiner/gleich der Korrektheit der Antworten bei der Desktop-Hardware“

Alternative Hypothese H_{c1} : „Die Korrektheit der Antworten bei der VR-Hardware ist größer der Korrektheit der Antworten bei der Desktop-Hardware“

5.1.2 Bearbeitungszeit

Nullhypothese H_{b0} : „Die Zeit für die Bearbeitung der Aufgabe ist mit der VR-Hardware länger/gleich als die Zeit mit der Desktop-Hardware“

Alternative Hypothese H_{b1} : „Die Zeit für die Bearbeitung der Aufgabe ist mit der VR-Hardware kürzer als die Zeit mit der Desktop-Hardware“

5.1.3 Usability

- **System Usability Scale (SUS)**

Nullhypothese H_{s0} : „Der SUS-Score¹ ist für die VR-Hardware kleiner/gleich dem SUS-Score der Desktop-Hardware“

SUS: Die System Usability Scale besteht aus 10 Fragen und wird in unserem Fall nach jedem Hardware-Abschnitt ausgefüllt.

¹ Je größer dieser ist, desto besser die Usability.

Alternative Hypothese H_{s1} : „Der SUS-Score ist für die VR-Hardware größer als der SUS-Score der Desktop-Hardware“

ASQ: Der After-Scenario-Questionnaire besteht aus drei Fragen und wird nach jeder Aufgabe ausgefüllt.

- **After-Scenario-Questionnaire (ASQ)**

- **ASQ-Aufwand**

Nullhypothese H_{a0} : „Der ASQ-Score² für die VR-Hardware in Bezug auf den Aufwand ist größer/gleich dem der Desktop-Hardware“

Alternative Hypothese H_{a1} : „Der ASQ-Score für die VR-Hardware in Bezug auf den Aufwand ist kleiner als der der Desktop-Hardware“

- **ASQ-Komplexität**

Nullhypothese H_{k0} : „Der ASQ-Score für die VR-Hardware in Bezug auf die Komplexität ist größer/gleich dem der Desktop-Hardware“

Alternative Hypothese H_{k1} : „Der ASQ-Score für die VR-Hardware in Bezug auf den Aufwand ist kleiner als der der Desktop-Hardware“

- **ASQ-Informationszufriedenheit**

Nullhypothese H_{i0} : „Der ASQ-Score für die VR-Hardware in Bezug auf die Informationszufriedenheit ist größer/gleich dem der Desktop-Hardware“

Alternative Hypothese H_{i1} : „Der ASQ-Score für die VR-Hardware in Bezug auf die Informationszufriedenheit ist kleiner als der der Desktop-Hardware“

5.2 AUFBAU

Convenience-Sampling: Convenience-Sampling ist eine Stichprobenmethode, bei der die Wahl der Probanden nicht zufällig ist, sondern die Wahl hauptsächlich von der Verfügbarkeit oder der geografischen Nähe der Probanden abhängt (Nikolopoulou (2023)).

Die Studie wurde so aufgebaut, dass die Probanden jeweils drei Aufgaben mit dem VR-Headset und drei Aufgaben mit Maus und Tastatur lösen mussten. Nach jeder Aufgabe soll ein Post-Task-Fragebogen und nach jedem Hardware-Abschnitt ein Post-Study-Fragebogen ausgefüllt werden, um später die beiden Hardware-Alternativen zu vergleichen. Um den Einfluss der Bearbeitungsreihenfolge mitzubeachten, sollten 50% der Probanden erst die VR-Hardware und danach die Desktop-Hardware nutzen und die anderen 50% erst die Desktop-Hardware und anschließend die VR-Hardware. Durch die Hardware-Voraussetzung waren wir auf ein *Convenience-Sampling* angewiesen.

² Je kleiner dieser ist, desto besser die Usability.

5.3 FRAGEBOGEN

In diesem Kapitel werden die verwendeten Fragebögen, die zur Durchführung der Studie benutzt werden kurz vorgestellt. Die Wahl der Fragebögen ist dabei durch die Arbeiten vorheriger Studenten der Arbeitsgruppe Softwaretechnik inspiriert, da es bei diesen Arbeiten auch um das Thema Usability, aber auch um den Faktor ging, die Studie nicht zu lang zu gestalten [Galperin \(2021\)](#) (S. 34-39), [Krause \(2024\)](#) (S. 117-121).

5.3.1 *Demographischer Fragebogen*

Dieser Fragebogen wird genutzt, um die unabhängigen Variablen festzulegen. Dabei werden folgende Angaben abgefragt:

- **Hast du einen Hintergrund im Bereich Informatik?**
- **Alter**
- **Geschlecht**
- **Höchster Abschluss**
- **SEE-Erfahrung**
- **VR-Erfahrung**
- **Videospiel-Erfahrung**

5.3.2 *Post-Task-Fragebogen*

Da VR an sich schon anstrengend für Probanden sein kann und diese sich in „einer anderen Welt“ befinden, war es besonders wichtig, die Fragen nach den Aufgaben so kurz wie möglich zu halten, um den VR-Aufenthalt nicht unnötig in die Länge zu ziehen, andererseits aber auch um die Probanden zwischen den Aufgaben nicht zu sehr aus dem Geschehen zu reißen. Deshalb wurde sich für den After-Scenario-Questionnaire entschieden, da dieser aus nur drei Fragen besteht, welche jeweils von 1 „Stimme voll zu“ bis 7 „Stimme überhaupt nicht zu“ bewertet werden. Bei der ersten Frage geht es dabei um die Zufriedenheit mit der Leichtigkeit/Komplexität der Aufgabe, bei der zweiten Aufgabe um die Zufriedenheit mit dem Aufwand, bzw. benötigten Zeit um die Aufgabe zu lösen und bei der dritten Frage um die Zufriedenheit mit den unterstützenden Informationen [Lewis \(1991\)](#) (S.78-79).

5.3.3 *Post-Study-Fragebogen*

Als Post-Study-Fragebogen wurde die System Usability Scale benutzt, welche ein etablierter und schon seit 1986 verwendeter Fragebogen ist, mit dem versucht wird, die Gesamtzufriedenheit (aufgeteilt in Benutzerfreundlichkeit und Lernfähigkeit) mit einem System zu quantifizieren [Sauro \(2011\)](#). Dieser besteht aus 10 Fragen, welche immer abwechselnd positiv und negativ formuliert sind.

Der SUS-Score wird dabei wie folgt berechnet:

- Alle ungerade Fragen: Antwort des Probanden - 1.
- Alle geraden Fragen: 5 - Antwort des Probanden.
- Addiere die Ergebnisse (Wertebereich: [0,40]).
- Multipliziere das Ergebnis mit 2.5 um den Wertebereich [0,100] zu erhalten [Sauro \(2011\)](#).

5.4 AUFGABENSTELLUNG

Da immer noch die Usability der Interaktionen zwischen den beiden Hardware-Alternativen untersucht werden soll, müssen wir unterschiedliche Interaktionsarten abdecken. Grob lassen sich diese im Fall von *SEE* in drei Gruppen einteilen. Einmal die „Navigationsinteraktion“, damit ist die Bewegungs- und Kamera-Steuerung gemeint. Zudem gibt es noch die „City-Interaktionen“, also alle Interaktionen, die direkt mit der Code-City zu tun haben und diese ändern können. Dazu gehört unter anderem „Move“, womit Blöcke verschoben werden können, oder „New-Edge“, womit eine neue Kante hinzugefügt werden kann. Eine weitere Gruppe an Interaktionen kann unter dem Begriff „Userinterface-Interaktionen“ zusammengefasst werden. Wie der Name schon vermuten lässt, geht es hierbei um die Interaktion mit den in *SEE* vorhandenen Userinterfaces, wie das „CodeWindow“ oder die „TreeView“. Natürlich hängen alle diese Gruppen zusammen und es soll auch nur eine grobe Einteilung sein und als Hilfestellung dienen, um in den Aufgaben möglichst viele verschiedene Interaktionen in *SEE* abzudecken.

5.4.1 *Aufgabe 1*

Die erste Aufgabe soll den Fokus auf die „Move-Aktion“ legen. Da diese alleine nur schwer interpretierbare Ergebnisse liefern kann und auch nicht zu einfach sein soll, wurde beschlossen, diese mit Interaktionen aus der Gruppe „Userinterface-Interaktionen“ zu verbinden. Genauer gesagt, mit dem „ShowMetrics“ Menü, in welchem alle Metriken für einen Block angezeigt werden. Automatisch werden auch Interaktionen aus der Gruppe „Navigationsinteraktion“ mit einbezogen, da die

Probanden sich in der virtuellen Welt, aber auch in der Code-City, zurechtfinden müssen. Daraus entsteht die Aufgabe: „Verschiebe den Block X und den Block Y in den Ordner Z. Wie viel Lines-of-Code haben die 5 größten Blöcke im Ordner Z. Nutze dafür das Menü „Show Metrics“ und suche nach „Metric.Lines.Class.LOC“. Hinweis: Klicke auf einen Block um das Popupmenu zu öffnen, dort findest du die Aktion „Show Metrics“.“

Da die Aufgabe einmal mit VR- und Desktop-Hardware durchgeführt wird, repräsentiert diese Aufgabe die Aufgaben 1 und 4 in der Studiendurchführung (5.1). Für Aufgabe 1 wurde als Ordner „UI“ und für Aufgabe 4 der Ordner „Layout“ ausgesucht, da diese eine ähnliche Struktur aufweisen und somit gleichermaßen in der Code-City auffindbar sein sollten. Als Blöcke wurden für Aufgabe 1 „AnimationInteraction“ und „GraphRenderer“ ausgesucht und für Aufgabe 4 „HideAction“ und „GraphReader“. Hierbei wurde darauf geachtet, dass die Blöcke eine ungefähr ähnliche Größe haben und sich in der Nähe der Ordner befinden.

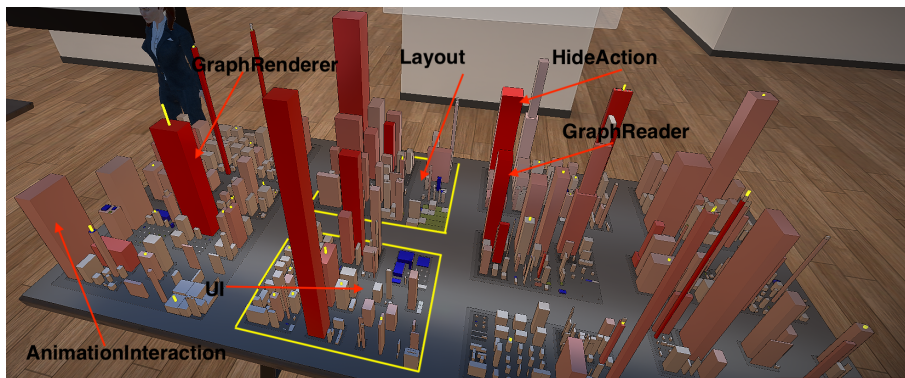


Abbildung 5.1: Die Ordner und Blöcke für Aufgabe 1 und Aufgabe 4.

5.4.2 Aufgabe 2

Die zweite Aufgabe soll den Fokus auf die „New-Edge-Aktion“ legen. Diese Aufgabe soll anregen, dass sich die Probanden genauer mit der Struktur der Code-City auseinandersetzen und sich in ihr zurechtfinden, aber im Hinblick auf die „New-Edge-Aktion“ auch eine gewisse Genauigkeit abverlangt. Um die Aufgabe ein wenig leichter zu gestalten, wird hier nur abgefragt, wie viele Kanten erstellt wurden. Daraus entsteht folgende Aufgabe: „Erstelle jeweils eine neue Kante vom Ordner X zu allen benachbarten Ordnern, ohne dass eine Kante über einen anderen Ordner hinweg verläuft. Zähle anschließend die Kanten. Wie viele Kanten wurden erstellt?“

Wie bei der ersten Aufgabe schon erwähnt, gibt es auch diese Aufgabe in der Studiendurchführung in zweifacher Ausführung. Dabei wurde für Aufgabe 2 der Ordner „DataModel“ und für die Aufgabe 5 der

Ordner „Utils“ ausgewählt, da beide eine ähnliche Größe besitzen, also gleich gut zu finden sein sollten und gleich viele benachbarte Ordner haben (5.2).

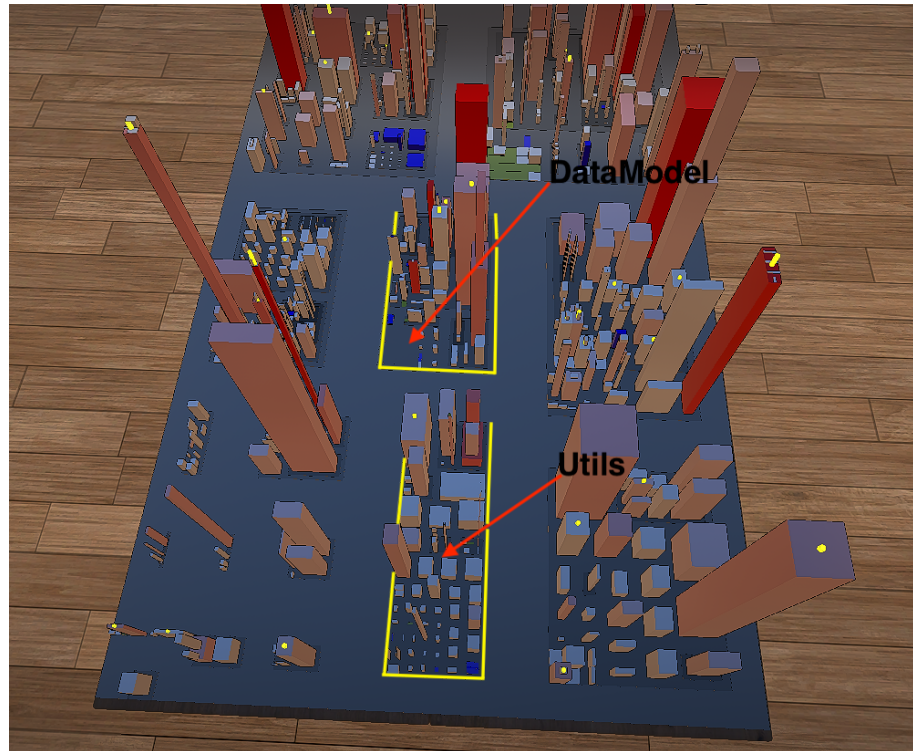


Abbildung 5.2: Die Ordner für Aufgabe 2 und Aufgabe 5.

5.4.3 Aufgabe 3

Die dritte Aufgabe soll nochmal den Fokus auf die „Userinterface-Interaktionen“ legen. Einerseits soll dafür das „CodeWindow“ verwendet werden, welches eins der zentralen Userinterfaces in SEE ist, da dies ermöglicht den Quelltext eines Blocks zu lesen. Andererseits soll die „TreeView“ genutzt werden, da dieses auch ein zentrales Userinterface von SEE ist, mit dem eine Code-City, wie in herkömmlichen Betriebssystemen als Baumstruktur in Form von Ordnern und Dateien dargestellt werden kann und vor allem für die Suche in der Code-City zwingend notwendig ist. Daraus ergibt sich die folgende Aufgabe: „Finde den Block X und öffne seinen Quelltext. Welchen Namen hat das Attribut auf Zeile Y? Hinweis: Nutze für die Suche die TreeView und öffne den Quelltext mit „Show Code“.“

Wie auch bei den letzten beiden Aufgaben, gibt es auch diese Aufgabe in der Studiendurchführung in zweifacher Ausführung. Für Aufgabe 3 (5.3a) wurde dabei der Block „ShowEdges“ und für die Aufgabe 6 (5.3b) der Block „SettingsMenu“ ausgewählt. Außerdem wird bei Aufgabe 3 nach dem Attribut auf Zeile 25 und bei Aufgabe 6 nach dem Attribut

auf Zeile 37 gefragt. Es wurde dabei darauf geachtet, dass beide Blöcke relativ klein und eher schwer zu finden sind, da für die Suche nicht das bloße Auge, sondern die TreeView genutzt werden soll.

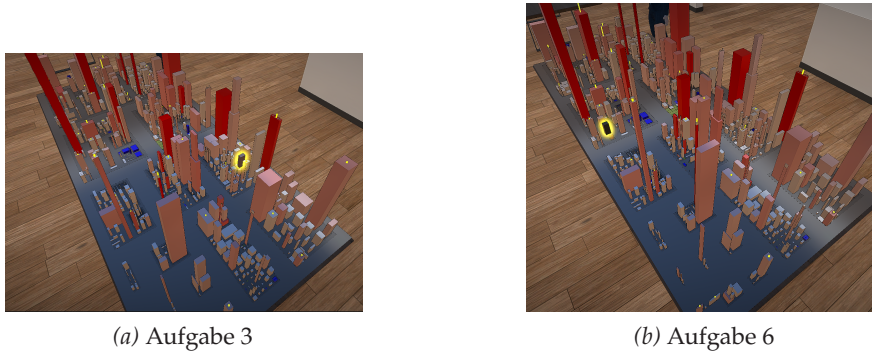


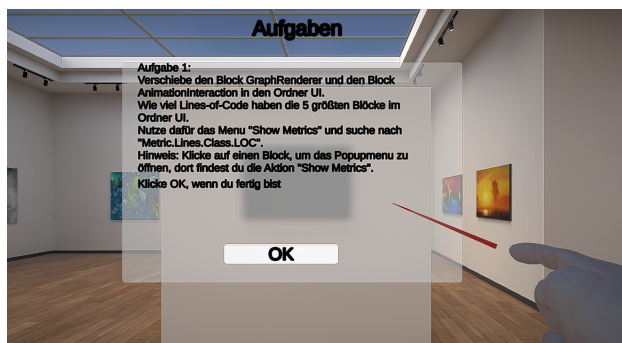
Abbildung 5.3: Die Blöcke für Aufgabe 3 und 6.

5.5 UMSETZUNG

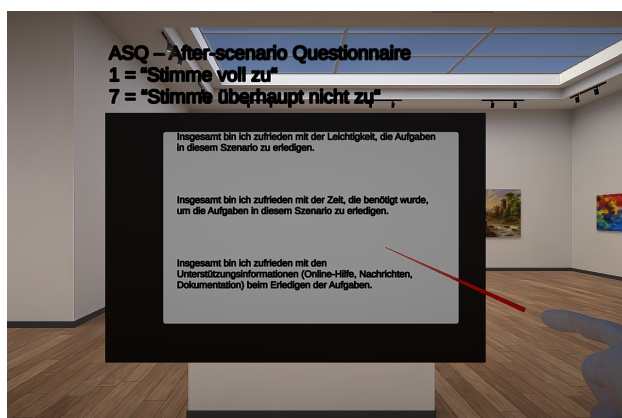
Aufgrund der Hardware-Voraussetzung, wurde entschieden, die Studie ausschließlich in Person durchzuführen. Als Ort wurde das „MZH“ an der Universität Bremen ausgewählt, da dort genug Freiraum für das Setup vorhanden ist (5.4) und auch die Chance höher ist, auf potenzielle Probanden zu treffen. Zweiteres stellte sich jedoch als schwieriger raus als gedacht, da zu der Zeit der Studie vorlesungsfreie Zeit war und nur sehr wenig Personen tatsächlich vor Ort in der Universität waren. Um sicher zu sein, dass die Software und Hardware tatsächlich funktioniert, wurde sich dazu entschieden, die eigene Hardware mitzunehmen, die auch für die Implementierung verwendet wurde. Vor Ort wurden dann alle Personen angesprochen, die dort den Tag über waren, bzw. vorbeigekommen sind. Dies spiegelt das Convenience-Sampling sehr gut wider. Für den Fragebogen wurde KoboToolbox verwendet, mit zwei Ausführungen des Fragebogens. Einmal VR→Desktop und einmal Desktop→VR. Die Zuweisung des Fragebogens wurde dann immer abgewechselt. Die ASQ-Fragen und die Aufgaben wurden auch in *SEE* in einem Canvas sichtbar gemacht, damit die Probanden nicht aus dem Erlebnis herausgezogen werden und in der virtuellen Welt bleiben können (5.5). Die Antworten wurden dann mündlich übermittelt und von mir in den Fragebogen eingetragen. Vor allem in VR ist dies sehr wichtig, da die Probanden sonst nach jeder Frage die VR-Brille hätten absetzen müssen, was schnell anstrengend geworden wäre.



Abbildung 5.4: Aufbau in der Universität



(a) Sichtbare Fragestellung in SEE



(b) Sichtbare ASQ-Fragen in SEE

Abbildung 5.5: Darstellung von Fragen innerhalb von SEE

5.6 AUSWERTUNG

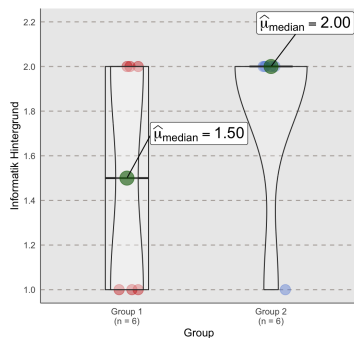
In dem folgenden Kapitel werden wir die erhobenen Daten der Studie auswerten. Dafür wurde ein bestehendes R-Skript von [Galperin u. a. \(2022\)](#) als Grundlage genommen und unseren Bedürfnissen angepasst. Dies ist auch noch mal auf dem USB-Stick zu dieser Arbeit zu finden. In fast allen Fällen nutzen wir dafür den Mann-Whitney-U-Test, da wir diesen mit ordinalskalierten Daten verwenden können, was ideal für unsere Zwecke ist.

5.6.1 Demographische Angaben

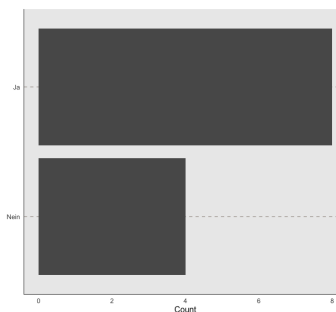
In diesem Abschnitt schauen wir uns die demographischen Angaben der Probanden an und untersuchen außerdem, ob es Unterschiede zwischen der Gruppe „VR-Desktop“ und der Gruppe „Desktop-VR“ gibt. Für jede Demographische Angabe nutzen wir den Mann-Whitney-U-Test mit einem Signifikanzniveau von 0,025 ($\alpha/2$) bei einem $n=m=6$ und einem kritischen Wert von 5.

- **Informatik Hintergrund**

In der Gruppe „VR-Desktop“ hatten 5 von 6 Probanden einen Hintergrund in Informatik (83,33%). Im Vergleich hatte diesen in der Gruppe „Desktop-VR“ nur 3 von 6 Probanden (50%). Der Median \hat{X} über beide Gruppen hinweg auf die Frage „Hast du einen Hintergrund in Informatik?“ ist dabei „Ja“. Der Mann-Whitney-U-Test ergab mit $U = 12$ und $p = 0,28$ keinen signifikanten Unterschied.



(a) Vergleich der Informatikzugehörigkeit Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)

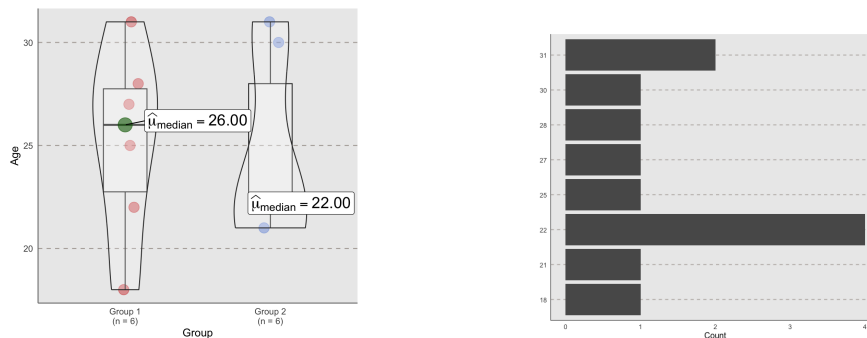


(b) Verteilung der Informatikzugehörigkeit aller Probanden

Abbildung 5.6: Vergleich der Informatikzugehörigkeit in den Gruppen und insgesamt

• **Alter**

Insgesamt wurde ein Durchschnitts-Alter von $\bar{X} \approx 24,92$ errechnet, wobei der Median bei $\tilde{X} = 23,5$ und die Standardabweichung bei $S \approx 4,38$ lag. Wir verwenden den Mann-Whitney-U-Test um zu schauen, ob es einen signifikanten Unterschied zwischen den beiden Gruppen gibt. Dieser ergab mit $U = 20$ und $p \approx 0,8$ keinen signifikanten Unterschied.



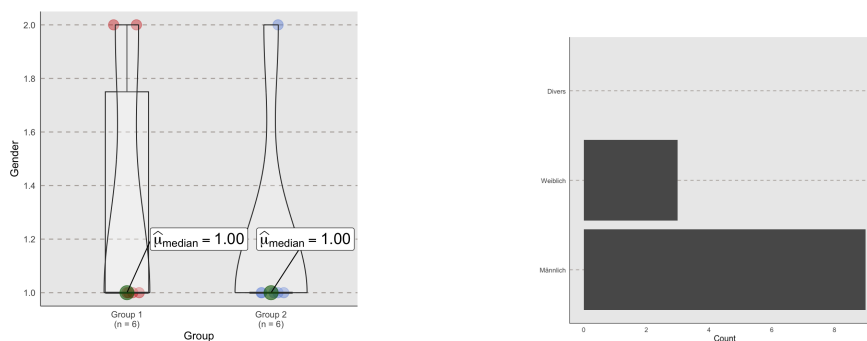
(a) Vergleich des Alters zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)

(b) Verteilung des Alters aller Probanden

Abbildung 5.7: Vergleich des Alters in den Gruppen und insgesamt

• **Geschlecht**

In der Gruppe „VR-Desktop“ waren fünf männliche Probanden und eine weibliche. In der Gruppe „Desktop-VR“ waren vier männliche und zwei weibliche Probanden. Der Mann-Whitney-U-Test ergab hierbei mit $U = 21$ und $p \approx 0,59$ keinen signifikanten Unterschied.



(a) Vergleich des Geschlechts zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)

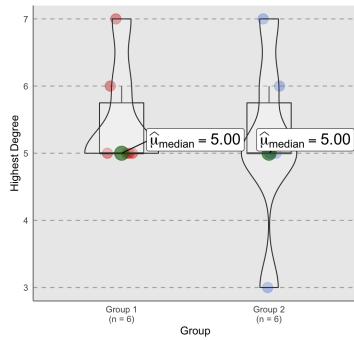
(b) Verteilung des Geschlechts aller Probanden

Abbildung 5.8: Vergleich des Geschlechts in den Gruppen und insgesamt

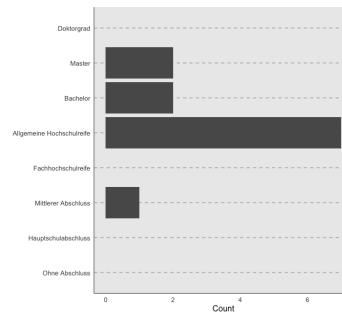
• **Höchster Abschluss**

In beiden Gruppen liegt hier der Median \tilde{X} bei „Allgemeine

Hochschulreife“, was darauf zurückzuführen ist, dass die Studie ausschließlich in der Universität durchgeführt wurde und demnach auch vermehrt Studierende teilgenommen haben. Der Mann-Whitney-U-Test ergab hier wieder mit $U = 20$ und $p \approx 0,79$ keinen signifikanten Unterschied.



(a) Vergleich des höchsten Abschlusses zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)

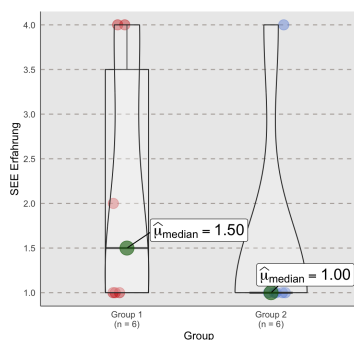


(b) Verteilung des höchsten Abschlusses aller Probanden

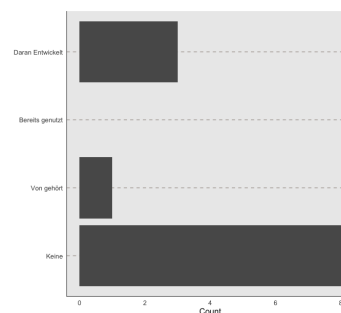
Abbildung 5.9: Vergleich des höchsten Abschlusses in den Gruppen und insgesamt

- **SEE Erfahrung**

In der Gruppe „VR-Desktop“ gab nur ein Proband an, bereits an SEE entwickelt zu haben, die Übrigen hatte noch keine Erfahrung. In der Gruppe „Desktop-VR“ gaben 2 Probanden an, an SEE entwickelt zu haben, ein Proband gab an, davon gehört zu haben und 2 weitere hatten noch keine Erfahrung. Insgesamt liegt der Median \tilde{X} bei „Keine“. Der Mann-Whitney-U-Test ergab jedoch, dass es keinen signifikanten Unterschied gibt, mit $U = 23,5$ und $p \approx 0,34$.



(a) Vergleich der SEE-Erfahrung zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)

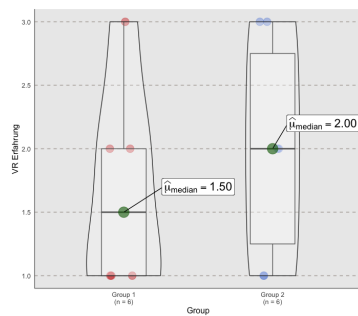


(b) Verteilung der SEE-Erfahrung aller Probanden

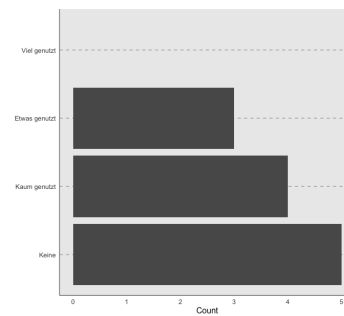
Abbildung 5.10: Vergleich und Verteilung der SEE-Erfahrung in den Gruppen und insgesamt

- **VR Erfahrung**

Hier lag der Median \tilde{X} bei „Kaum genutzt“, was nicht verwunderlich ist, da VR eine vergleichsweise neue Technologie ist, welche lange Zeit nicht für die breite Masse verfügbar war und im Vergleich zu anderen Videospieleplattformen immer noch ein Nischenprodukt ist [Armstrong \(2023\)](#). Der Mann-Whitney-U-Test ergab bei einem $U = 14$ und $p \approx 0,55$ wieder keinen signifikanten Unterschied.



(a) Vergleich der VR-Erfahrung zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)



(b) Verteilung der VR-Erfahrung aller Probanden

Abbildung 5.11: Vergleich und Verteilung der VR-Erfahrung in den Gruppen und insgesamt

- **Videospiel Erfahrung**

Hier lag der Median \tilde{X} bei „Viel genutzt“, was alleine durch den großen Anteil an Probanden mit Informatik-Hintergrund nicht ungewöhnlich ist. Auch hier ergab der Mann-Whitney-U-Test bei einem $U = 21$ und $p \approx 0,65$ keinen signifikanten Unterschied zwischen beiden Gruppen.

5.6.2 Korrektheit

Nun vergleichen wir die Korrektheit der Antworten für jede Aufgabe. Dabei vergleichen wir die Korrektheit auf zwei Weisen. Einmal wollen wir wissen, wie sich die Korrektheit jeder Aufgabe zwischen VR und Desktop unterscheidet. Darüber hinaus wollen wir uns aber auch noch damit befassen, wie sich die Korrektheit zwischen den beiden Gruppe verhält. Die korrekte Durchführung der Aufgaben ist noch einmal separat, in Form von Videos auf dem USB-Stick zu finden (VR_Aufgabenteil_1.mp4, VR_Aufgabenteil_2.mp4, Desktop_Aufgabenteil_1.mp4, Desktop_Aufgabenteil_2.mp4), um ein besseres Verständnis zu bekommen.

Wir starten mit der Korrektheit zwischen beiden Gruppen. Die Werte geben dabei an, wie viel Prozent die Probanden zusammengerechnet

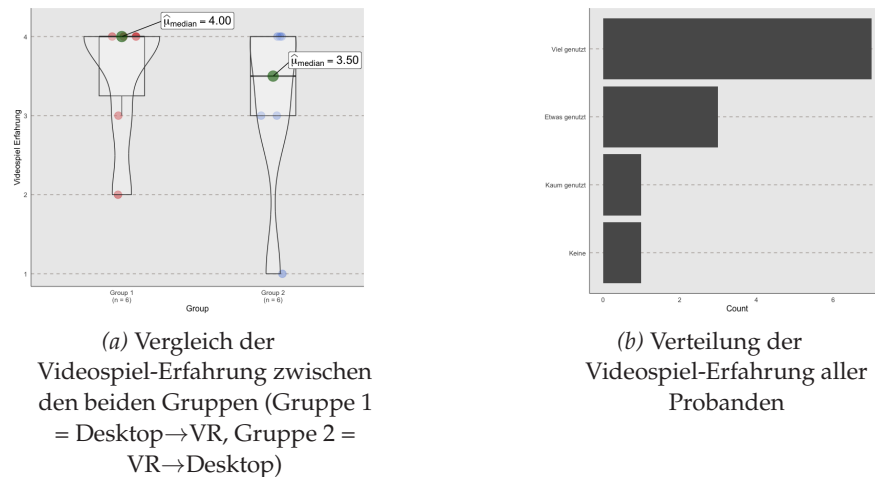


Abbildung 5.12: Vergleich und Verteilung der Videospiele-Erfahrung in den Gruppen und insgesamt

in allen Aufgaben korrekt gelöst hatten. Hier gab es über beide Gruppen hinweg einen Durchschnitt von $\bar{X} \approx 75,83\%$ und einen Median von $\tilde{X} = 75\%$ bei einer Standardabweichung von $S \approx 17,82\%$. Bei Gruppe 1 (Desktop→VR) gab es einen Durchschnitt von $\bar{X} \approx 78,33\%$ und einen Median von $\tilde{X} \approx 81,66\%$ bei einer Standardabweichung von $S \approx 18,47\%$. Bei der Gruppe 2 (VR-Desktop) gab es einen Durchschnitt von $\bar{X} \approx 73,33\%$ und einen Median von $\tilde{X} = 70\%$ bei einer Standardabweichung von $S \approx 18,5\%$. Wenig verwunderlich sagt auch der Mann-Whitney-U-Test in diesem Fall, mit $U = 21$ und $p \approx 0,69$, dass es keinen signifikanten Unterschied zwischen den beiden Gruppen gibt.

Kommen wir nun zu den einzelnen Aufgaben und wie sich ihre Korrektheit zwischen VR und Desktop verhält. Hierfür nutzen wir wieder den Mann-Whitney-U-Test, diesmal in gerichteter Form, mit einem kritischen Wert $U = 7$ für das $\alpha = 0,05$ bei $n = m = 6$.

- **Aufgabe 1**

Bei der ersten Aufgabe ergab sich ein Durchschnitt von $\bar{X} \approx 63,33\%$ und ein Median von $\tilde{X} = 70\%$ bei einer Standardabweichung von $S \approx 22,3\%$. Hier sehen wird durch den Mann-Whitney-U-Test mit $U = 5$ und $p \approx 0,014$, dass die Aufgabe 1 mit der Desktop-Hardware korrekter bearbeitet wurde. Dies spiegelt sich auch in der Abbildung wider.

- **Aufgabe 2**

Bei der zweiten Aufgabe ergab sich ein Durchschnitt von $\bar{X} \approx 66,67\%$ und ein Median von $\tilde{X} = 100\%$ bei einer Standardabweichung von $S \approx 49,23\%$. In beiden Gruppen gab es dabei zwei

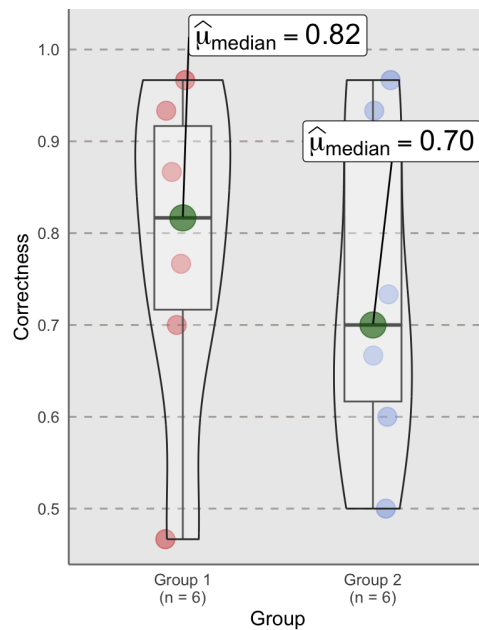


Abbildung 5.13: Vergleich der Korrektheit zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)

Probanden, die die Aufgabe falsch beantwortet haben und die Übrigen richtig. Der Mann-Whitney-U-Test gibt dabei ein $U = 18$ und $p \approx 0,54$ aus. Damit ist hier also keinen signifikanten Unterschied festzustellen. Dies zeigt sich auch in der Abbildung.

- **Aufgabe 3**

Bei der dritten Aufgabe gibt es einen Durchschnitt von $\bar{X} \approx 91,67\%$ und einen Median von $\tilde{X} = 100\%$ bei einer Standardabweichung von $S \approx 28,87\%$. In der Gruppe mit Desktop-Hardware gab es dabei keine falsche Antwort. Bei der Gruppe mit VR-Hardware gibt es jedoch eine falsche Antwort. Wahrscheinlich hat sich der jeweilige Proband in der Datei geirrt und die Aufgabe nicht richtig gelesen, dies ist jedoch nur eine Vermutung. Der Mann-Whitney-U-Test ergab hier bei einem $U = 15$ und $p \approx 0,2$ keinen signifikanten Unterschied.

- **Aufgabe 4**

Bei der vierten Aufgabe haben wir einen Durchschnitt von $\bar{X} = 75\%$ und einen Median von $\tilde{X} = 80\%$ bei einer Standardabweichung von $S \approx 29,7\%$. Der Mann-Whitney-U-Test gibt bei einem $U = 7,5$ und $p \approx 0,045$ einen signifikanten Unterschied, jedoch ist der U-Wert über unserem kritischen Wert, dies ist, wie Galperin (2021) (S. 58) schon in seiner Arbeit anmerkte, auf sogenannte „Ties“ zurückzuführen, da Rang basierte Tests mit diesen weniger gut umgehen können. Deshalb nutzen wir hier zur Überprüfung

Fisher-Pitman-
Randomisierungstest:
Der Fisher-Pitman-
Randomisierungstest
testet, ob zwei
unabhängige
Stichproben aus
derselben
Grundgesamtheit
stammen.

zusätzlich den *Fisher-Pitman-Randomisierungstest*. Dieser gibt bei einem $Z \approx -1,75$ und $p \approx 0,04$ einen signifikanten Unterschied zugunsten der Desktop-Hardware aus.

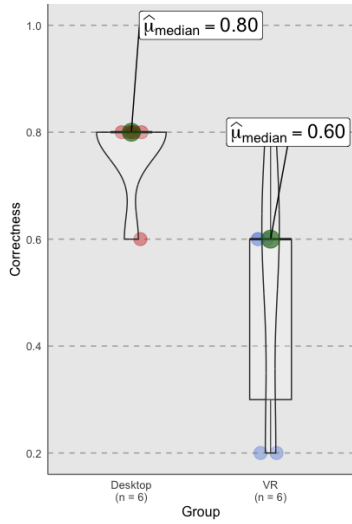
- **Aufgabe 5**

Hier haben wir einen Durchschnitt von $\bar{X} \approx 58,33\%$ und einen Median von $\tilde{X} = 100\%$ bei einer Standardabweichung von $S \approx 51,5\%$. Der Mann-Whitney-U-Test ergab in diesem Fall mit $U = 15$ und $p \approx 0,32$ keinen signifikanten Unterschied.

- **Aufgabe 6**

Bei der letzten Aufgabe gab es einen Durchschnitt von $\bar{X} = 100\%$ und einen Median von $\tilde{X} = 100\%$ bei einer Standardabweichung von $S = 0\%$. Der Mann-Whitney-U-Test ergab hier mit $U = 18$ und $p = 1$ auch keinen signifikanten Unterschied, da alle Probanden diese Aufgabe richtig gelöst haben.

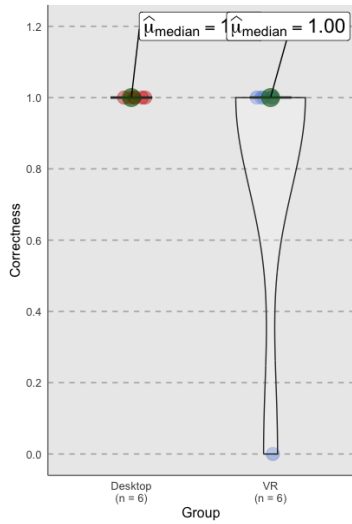
Insgesamt zeigte sich in zwei Fällen (Aufgabe 1 und Aufgabe 4) ein besseres Ergebnis für die Desktop-Hardware. In allen anderen Fällen gab es keinen signifikanten Unterschied. Wir können also unsere Nullhypothese H_{c0} nicht verwerfen und nehmen damit an, dass bei dem Bearbeiten der Aufgaben korrektere Ergebnisse mit der Desktop-Hardware erzielt werden.



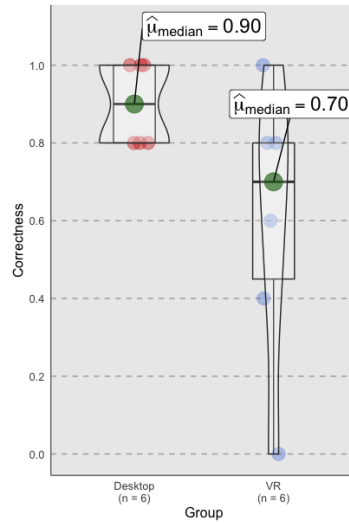
(a) Korrektheit Aufgabe 1



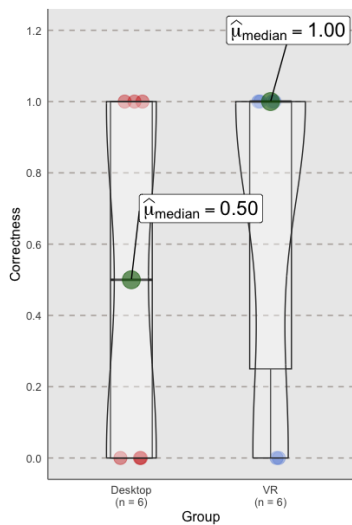
(b) Korrektheit Aufgabe 2



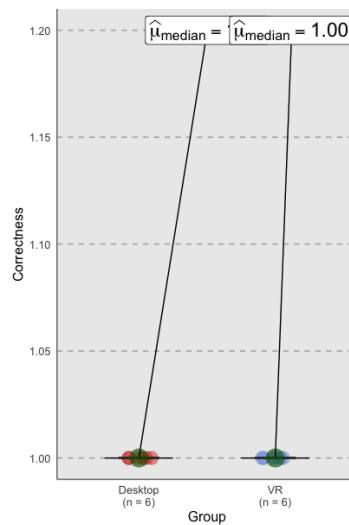
(c) Korrektheit Aufgabe 3



(d) Korrektheit Aufgabe 4



(e) Korrektheit Aufgabe 5



(f) Korrektheit Aufgabe 6

Abbildung 5.14: Die Korrektheit der Antworten für die einzelnen Aufgaben aller Probanden, angegeben in Prozent, im Vergleich zwischen Desktop und VR

5.6.3 Bearbeitungszeit

In diesem Abschnitt schauen wir uns die Bearbeitungszeit der einzelnen Aufgaben an. Diese wurde gemessen, indem vor jeder Aufgabe ein Knopf im Fragebogen erschien, mit „Klicke OK, wenn du bereit für die erste Aufgabe bist, ab dann wird die Zeit gemessen, die du für diese Aufgabe benötigst.“ und nach jeder Aufgabe ein Knopf „Klicke OK, wenn du fertig bist.“. Bei beiden Klicks wurde ein Zeitstempel erzeugt, wodurch wir die Differenz der beiden Zeitpunkte ausrechnen können, was die benötigte Zeit zur Bearbeitung der Aufgabe widerspiegelt.

Zunächst schauen wir uns die summierte Zeit für beide Gruppen an. Hier sehen wir, dass die erste Gruppe mit einem Durchschnitt von $\bar{X} \approx 36,16$ Minuten und Median von $\tilde{X} = 33,38$ Minuten bei einer Standardabweichung von $S \approx 14,21$ Minuten weniger Zeit brauchte als die zweite Gruppe mit einem Durchschnitt von $\bar{X} \approx 39,8$ Minuten und Median von $\tilde{X} = 34,12$ Minuten bei einer Standardabweichung von $S \approx 14,95$ Minuten. Der ungerichtete Mann-Whitney-U-Test zeigt aber schon, dass dieser Unterschied nicht signifikant ist, bei einem $U = 14$ und $p \approx 0,57$.

Als Nächstes kommen wir wieder zu den einzelnen Aufgaben und wie sich ihre Bearbeitungszeit zwischen VR und Desktop verhält. Hierfür nutzen wir den Mann-Whitney-U-Test wieder in gerichteter Form, mit einem kritischen Wert $U = 7$ für das $\alpha = 0,05$ bei $n = m = 6$.

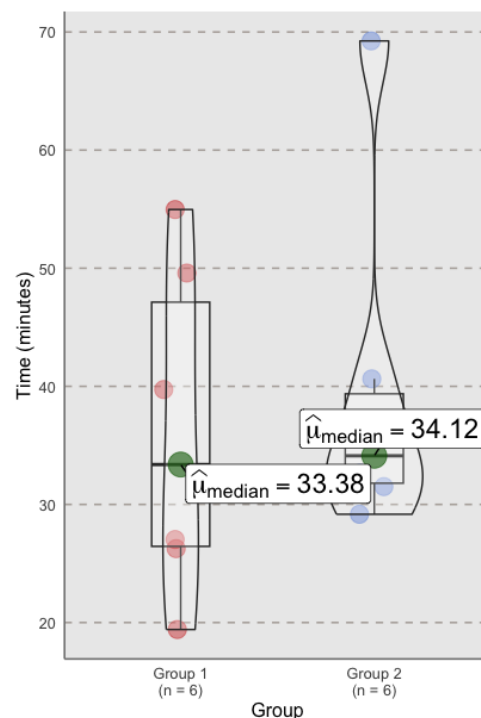


Abbildung 5.15: Vergleich der Bearbeitungszeit zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)

- **Aufgabe 1**

Bei der ersten Aufgabe ergibt sich insgesamt ein Durchschnitt von $\bar{X} = 10,86$ Minuten und ein Median von $\tilde{X} = 7,98$ Minuten bei einer Standardabweichung von $S \approx 6,93$ Minuten. Die Abbildung 5.16a lässt schon vermuten, dass es einen deutlichen Unterschied zwischen der VR- und Desktop-Hardware gibt. Der Mann-Whitney-U-Test bestätigt diesen Eindruck mit $U = 2$ und $p \approx 0,0043$, zeigt dieser uns einen deutlich signifikanten Unterschied. Mit der VR-Hardware benötigten Probanden also wesentlich mehr Zeit, mit einer durchschnittlichen Bearbeitungszeit von $\bar{X} \approx 15,52$ Minuten und einen Median von $\tilde{X} = 15,5$ Minuten bei einer Standardabweichung von $S \approx 7,06$ Minuten, als Probanden mit der Desktop-Hardware bei einer durchschnittlichen Bearbeitungszeit von $\bar{X} \approx 6,19$ Minuten und einen Median von $\tilde{X} = 5,85$ Minuten bei einer Standardabweichung von $S \approx 1,92$ Minuten.

- **Aufgabe 2**

Bei der zweiten Aufgabe ergibt sich insgesamt ein Durchschnitt von $\bar{X} \approx 5,93$ Minuten und ein Median von $\tilde{X} = 5,21$ Minuten bei einer Standardabweichung von $S \approx 3,27$ Minuten. Auch hier kann man wieder der Abbildung 5.16b entnehmen, dass es einen deutlichen Unterschied zwischen den beiden Gruppen gibt. Der Mann-Whitney-U-Test bestätigt diese Vermutung wieder bei einem $U = 5$ und $p \approx 0,020$ und zeigt uns dabei einen signifikanten Unterschied. Auch hier lässt sich also feststellen, dass die Aufgabe mit VR-Hardware eine längere Bearbeitungszeit mit einem Durchschnitt von $\bar{X} \approx 7,82$ Minuten und einem Median von $\tilde{X} \approx 7,07$ Minuten bei einer Standardabweichung von $S \approx 3,54$ Minuten benötigt, als mit der Desktop-Hardware bei einer durchschnittlichen Bearbeitungszeit von $\bar{X} \approx 4,03$ Minuten, einem Median von $\tilde{X} \approx 3,89$ Minuten und einer Standardabweichung von $S \approx 1,58$ Minuten.

- **Aufgabe 3**

Bei der dritten Aufgabe ergibt sich insgesamt ein Durchschnitt von $\bar{X} \approx 4,55$ Minuten und ein Median von $\tilde{X} \approx 2,59$ Minuten bei einer Standardabweichung von $S \approx 5,51$ Minuten. Der Mann-Whitney-U-Test ergibt wieder bei einem $U = 2$ und $p \approx 0,0043$ einen signifikanten Unterschied zugunsten der Desktop-Hardware. Diese hatte bei der Aufgabe eine durchschnittliche Bearbeitungszeit von $\bar{X} \approx 1,58$ Minuten und einen Median von $\tilde{X} \approx 1,64$ Minuten bei einer Standardabweichung von $S \approx 0,95$ Minuten im Vergleich zu der VR-Hardware mit einer durchschnittlichen Bearbeitungszeit von $\bar{X} \approx 7,52$ Minuten und einen Median von $\tilde{X} \approx 5,86$ Minuten

bei einer Standardabweichung von $S \approx 6,68$ Minuten.

- **Aufgabe 4**

Bei der vierten Aufgabe gab es insgesamt einen Durchschnitt von $\bar{X} \approx 10,28$ Minuten und einen Median von $\tilde{X} \approx 6,28$ Minuten bei einer Standardabweichung von $S \approx 7,94$ Minuten. Bei der Betrachtung der Abbildung 5.16d wird wieder ein deutlicher Unterschied bemerkbar. Auch der Mann-Whitney-U-Test ergibt wieder mit $U = 2$ und $p \approx 0,0043$ einen signifikanten Unterschied zugunsten der Desktop-Hardware. Dabei hatte diese eine durchschnittliche Bearbeitungszeit von $\bar{X} \approx 5,16$ Minuten und einen Median von $\tilde{X} = 4,89$ Minuten bei einer Standardabweichung von $S \approx 0,95$ Minuten im Vergleich zu der VR-Hardware mit einer durchschnittlichen Bearbeitungszeit von $\bar{X} = 15,41$ Minuten und einen Median von $\tilde{X} = 15,6$ Minuten bei einer Standardabweichung von $S \approx 8,63$ Minuten.

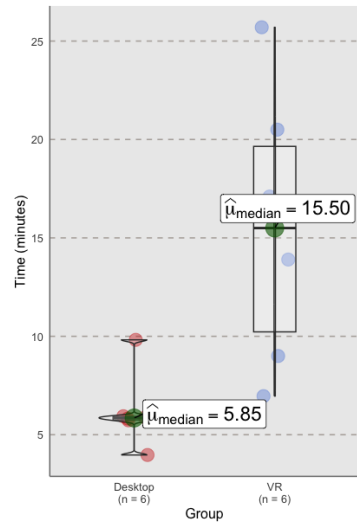
- **Aufgabe 5**

Bei der fünften Aufgabe gab es insgesamt eine durchschnittliche Bearbeitungszeit von $\bar{X} \approx 3,48$ Minuten und einen Median von $\tilde{X} \approx 3,47$ Minuten bei einer Standardabweichung von $S \approx 1,13$ Minuten. Der Mann-Whitney-U-Test zeigt bei einem $U = 4$ und $p \approx 0,013$ wieder einen signifikanten Unterschied zugunsten der Desktop-Hardware. Dabei hatte diese eine durchschnittliche Bearbeitungszeit von $\bar{X} \approx 2,82$ Minuten und einen Median von $\tilde{X} = 2,58$ Minuten bei einer Standardabweichung von $S \approx 0,66$ Minuten im Vergleich zu der VR-Hardware mit einer durchschnittlichen Bearbeitungszeit von $\bar{X} \approx 4,14$ Minuten und einen Median von $\tilde{X} \approx 3,93$ Minuten bei einer Standardabweichung von $S \approx 1,15$ Minuten.

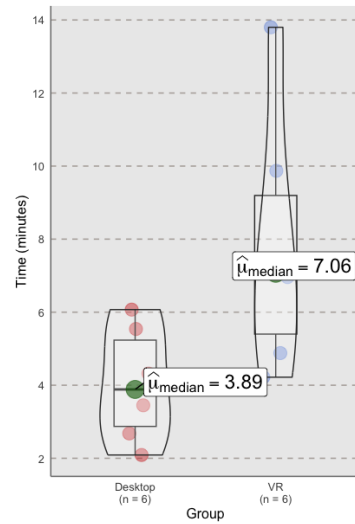
- **Aufgabe 6**

Bei der sechsten Aufgabe gab es insgesamt eine durchschnittliche Bearbeitungszeit von $\bar{X} \approx 2,89$ Minuten und einen Median von $\tilde{X} \approx 2,5$ Minuten bei einer Standardabweichung von $S \approx 2,99$ Minuten. Der Mann-Whitney-U-Test zeigt hier wieder mit $U = 0$ und $p \approx 0,001$ einen signifikanten Unterschied zugunsten der Desktop-Hardware. Diese hatte eine durchschnittliche Bearbeitungszeit von $\bar{X} \approx 0,95$ Minuten und einen Median von $\tilde{X} \approx 0,74$ Minuten bei einer Standardabweichung von $S \approx 0,78$ Minuten im Vergleich zu der VR-Hardware mit einer durchschnittlichen Bearbeitungszeit von $\bar{X} \approx 4,82$ Minuten und einen Median von $\tilde{X} \approx 3,94$ Minuten bei einer Standardabweichung von $S \approx 3,18$ Minuten.

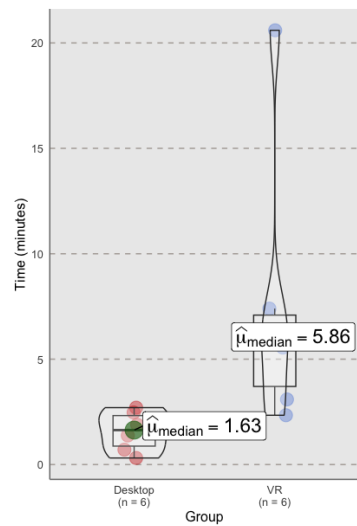
Über alle Aufgaben hinweg konnten wir also sehen, dass die Aufgaben mit der Desktop-Hardware deutlich schneller, als mit der VR-Hardware zu bearbeiten waren. Im Fall von Aufgabe 1 und Aufgabe 4 können wir sogar sagen, dass sie mit der Desktop-Hardware sowohl schneller als auch korrekter beantwortet werden konnten. Wir können unsere Nullhypothese H_{b_0} also nicht verwerfen und nehmen diese an.



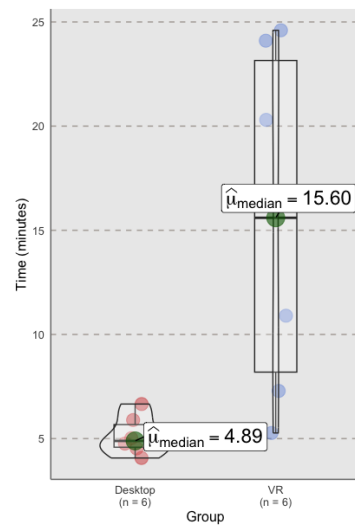
(a) Bearbeitungszeit Aufgabe 1



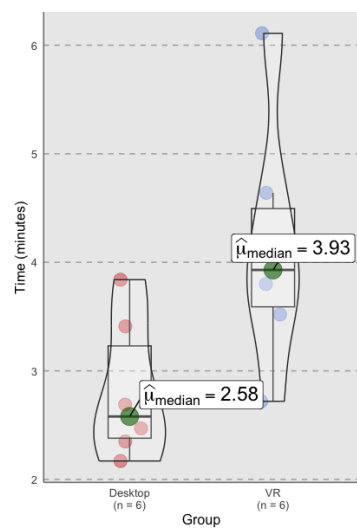
(b) Bearbeitungszeit Aufgabe 2



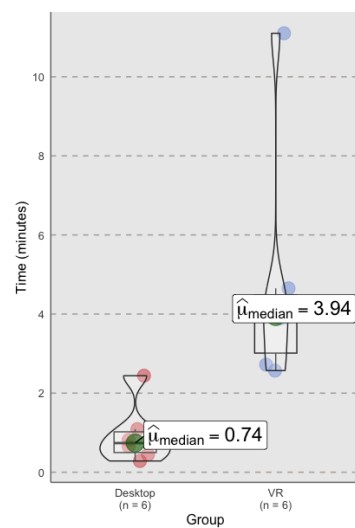
(c) Bearbeitungszeit Aufgabe 3



(d) Bearbeitungszeit Aufgabe 4



(e) Bearbeitungszeit Aufgabe 5



(f) Bearbeitungszeit Aufgabe 6

Abbildung 5.16: Die Bearbeitungszeit für die einzelnen Aufgaben aller Probanden, angegeben in Minuten, im Vergleich zwischen Desktop und VR

5.6.4 Usability

In diesem Abschnitt werten wir die Usability-Fragen des Fragebogens aus. Im ersten Teil werden wir uns dabei auf den SUS-Score fokussieren. Im Hinblick auf die VR-Hardware haben wir hier einen Durchschnitt von $\bar{X} \approx 45,41$ und einen Median von $\tilde{X} = 45$ bei einer Standardabweichung von $S \approx 17,77$. Im Vergleich dazu haben wir bei der Desktop-Hardware einen Durchschnitt von $\bar{X} \approx 63,54$ und einen Median von $\tilde{X} = 72,5$ bei einer Standardabweichung von $S \approx 22,01$. Bei der Anwendung des Mann-Whitney-U-Tests mit $\alpha = 0,05$ und $n = m = 12$ und einem kritischen Wert $U = 42$ bekommen wir $U = 40$ und $p \approx 0,034$ und haben damit einen signifikanten Unterschied. Wir müssen also unsere Nullhypothese H_{s0} annehmen und dürfen sie nicht verwerfen. Die Usability der Desktop-Hardware wird also laut dem SUS-Score als signifikant besser bewertet als die Usability der VR-Hardware.

- **SUS-Usability**

Die eigentliche Usability wurde bei der VR-Hardware im Durchschnitt mit einem Score von $\bar{X} \approx 44,53$ und einem Median von $\tilde{X} = 43,75$ bei einer Standardabweichung von $S \approx 20,90$ und bei der Desktop-Hardware im Durchschnitt mit $\bar{X} \approx 61,72$ und einem Median von $\tilde{X} \approx 71,88$ bei einer Standardabweichung von $S \approx 23,99$ bewertet. Der Mann-Whitney-U-Test ergab hier bei einem $U = 42,5$ und $p \approx 0,047$ einen signifikanten Unterschied zugunsten der Desktop-Hardware. Diesen müssen wir jedoch wieder mit dem Fisher-Pitman-Randomisierungstest überprüfen, da der U-Wert über unserem kritischen Wert ist. Der Test ergibt mit einem $Z \approx -1,78$ und $p \approx 0,038$ jedoch auch einen signifikanten Unterschied zugunsten der Desktop-Hardware.

- **SUS-Learnability**

Die Learnability wurde bei der VR-Hardware im Durchschnitt mit $\bar{X} \approx 48,96$ und einem Median von $\tilde{X} = 50$ bei einer Standardabweichung von $S \approx 19,55$ und bei der Desktop-Hardware im Durchschnitt mit $\bar{X} \approx 70,83$ und einem Median von $\tilde{X} = 75$ bei einer Standardabweichung von $S \approx 22,19$ bewertet. Der Mann-Whitney-U-Test ergibt dabei mit $U = 32,5$ und $p \approx 0,011$ einen signifikanten Unterschied zugunsten der Desktop-Hardware.

Nun kommen wir zu dem ASQ-Score. Wir schauen uns wieder für jede Aufgabe separat an, ob es einen signifikanten Unterschied zwischen den einzelnen Teilen des ASQ (Leichtigkeit/Komplexität, Aufwand, Informationszufriedenheit) in Bezug auf VR- und Desktop-Hardware gibt. Dabei werden im Folgenden nur die Fälle aufgezählt, bei denen es tatsächlich signifikante Unterschiede gibt. Wir verwenden dafür wieder den Mann-Whitney-U-Test mit $\alpha = 0,05$ und $n = m = 6$ und $U = 7$.

- **Aufgabe 4**

Leichtigkeit/Komplexität: Bei der Leichtigkeit/Komplexität ergab der Mann-Whitney-U-Test mit $U = 30,5$ und $p \approx 0,024$ laut p -Wert einen signifikanten Unterschied, jedoch ist der U-Wert nicht unter unserem kritischen U-Wert von 7. Zur Überprüfung wird noch mal der Fisher-Pitman-Randomisierungstest genutzt, welcher mit $Z \approx 2,11$ und $p \approx 0,017$ ein signifikantes Ergebnis liefert, was in diesem Fall bedeutet, dass die Aufgabe mit der VR-Hardware schwerer zu bewältigen waren (5.18d).

- **Aufgabe 6**

Leichtigkeit/Komplexität: Bei der Leichtigkeit/Komplexität ergab der Mann-Whitney-U-Test mit $U = 30$ und $p \approx 0,025$ einen signifikanten Unterschied. Wie aber auch bei der vierten Aufgabe müssen wir hier noch einmal gegenprüfen, da der U-Wert deutlich größer ist, als unser kritischer Wert von 7. Daher nutzen wir wieder den Fisher-Pitman-Randomisierungstest und bekommen mit $Z \approx 2,1$ und $p \approx 0,018$ ein signifikantes Ergebnis, was wieder bedeutet, dass die Aufgabe mit der VR-Hardware schwerer zu bewältigen war (5.18f).

In allen anderen Fällen gab es keine signifikanten Unterschiede. Beide Unterschiede beziehen sich auf die Leichtigkeit/Komplexität der Aufgaben, wir müssen also unsere Nullhypothese H_{k0} in diesem Fall annehmen und davon ausgehen, dass die Aufgaben mit der VR-Hardware als komplexer wahrgenommen werden. Auch die Nullhypothesen H_{a0} und H_{i0} in Bezug auf den Aufwand und die Informationszufriedenheit können wir nicht verwerfen.

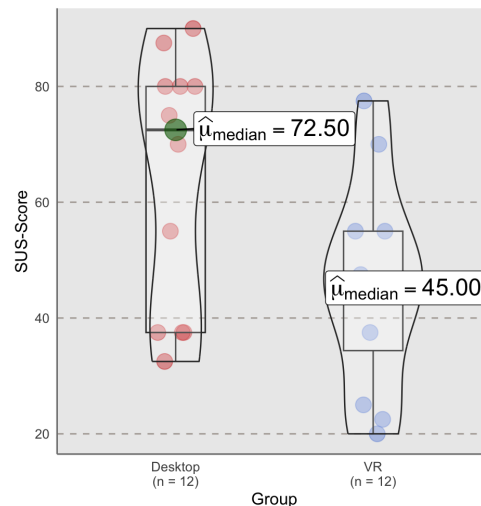
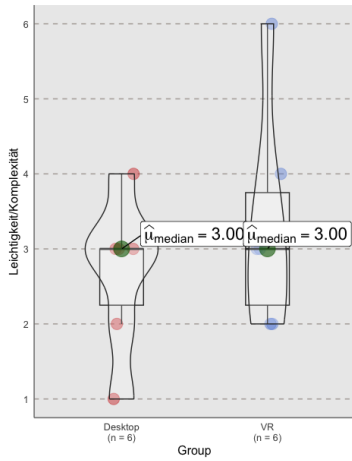
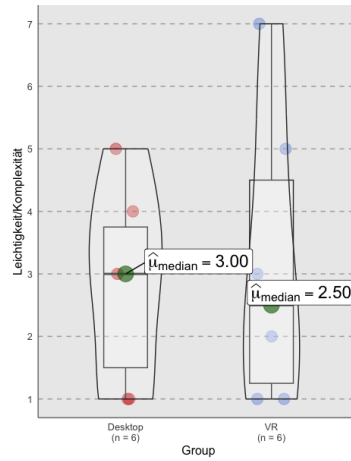


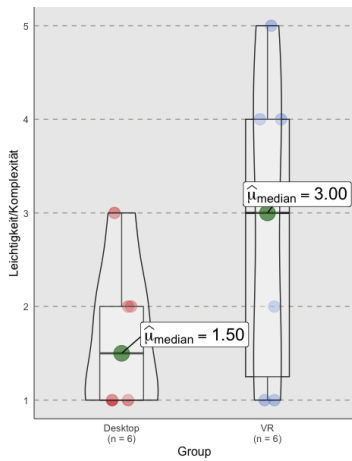
Abbildung 5.17: SUS-Score im Vergleich zwischen Desktop- und VR-Hardware



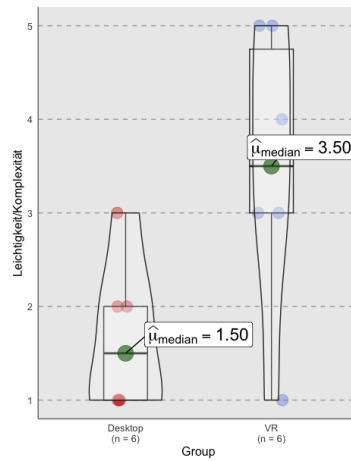
(a) Komplexität Aufgabe 1



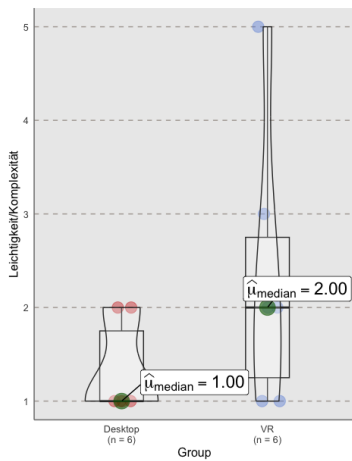
(b) Komplexität Aufgabe 2



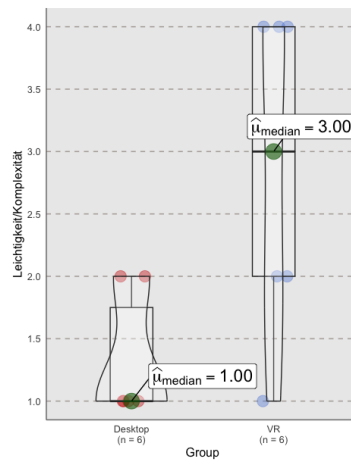
(c) Komplexität Aufgabe 3



(d) Komplexität Aufgabe 4

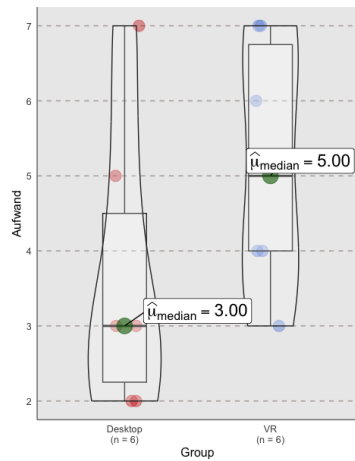


(e) Komplexität Aufgabe 5

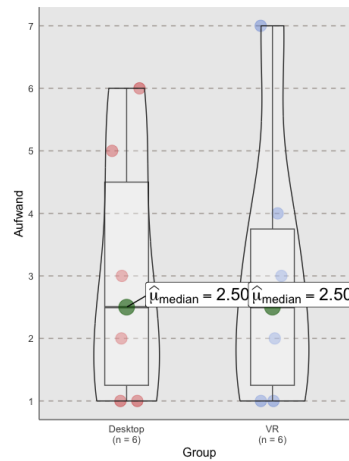


(f) Komplexität Aufgabe 6

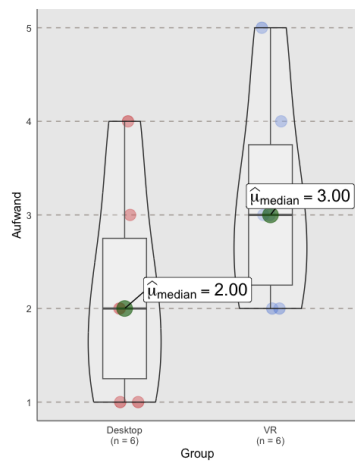
Abbildung 5.18: Die Komplexität für die einzelnen Aufgaben aller Probanden, im Vergleich zwischen Desktop und VR



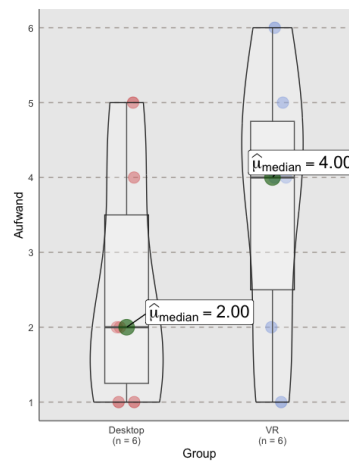
(a) Aufwand Aufgabe 1



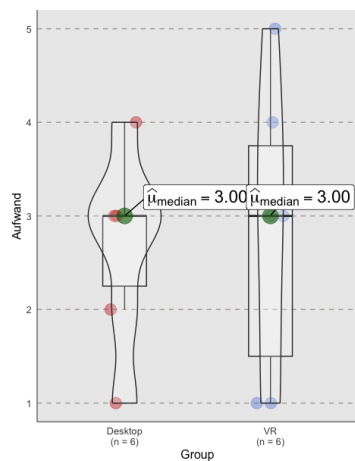
(b) Aufwand Aufgabe 2



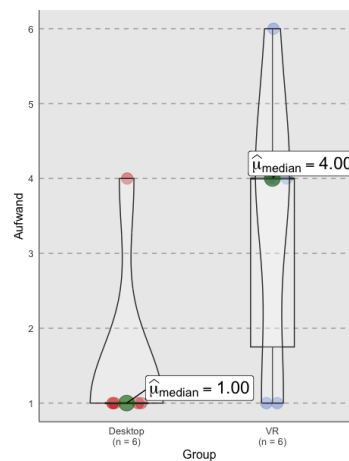
(c) Aufwand Aufgabe 3



(d) Aufwand Aufgabe 4

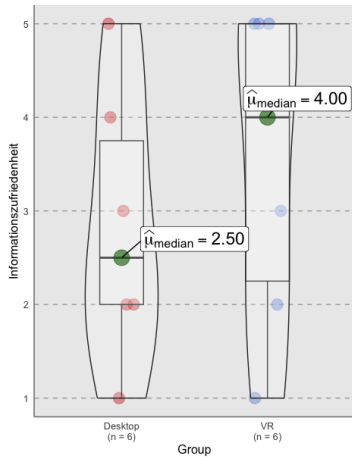


(e) Aufwand Aufgabe 5

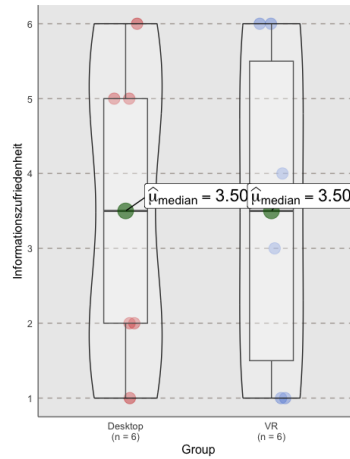


(f) Aufwand Aufgabe 6

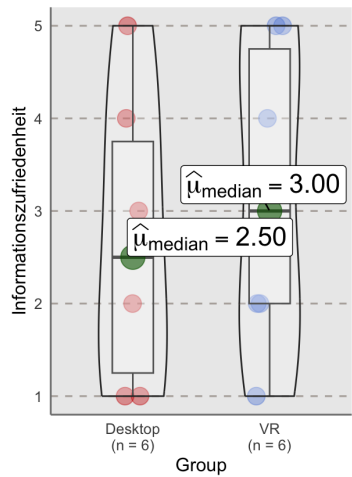
Abbildung 5.19: Der Aufwand für die einzelnen Aufgaben aller Probanden, im Vergleich zwischen Desktop und VR



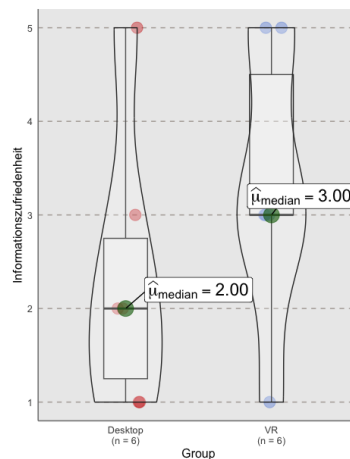
(a) Informationszufriedenheit Aufgabe 1



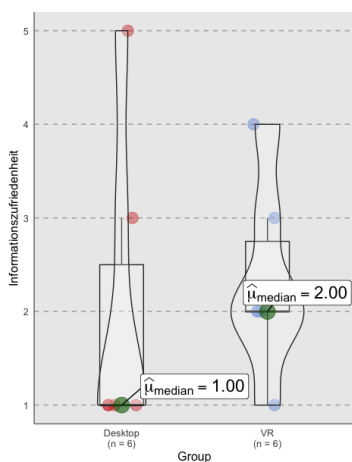
(b) Informationszufriedenheit Aufgabe 2



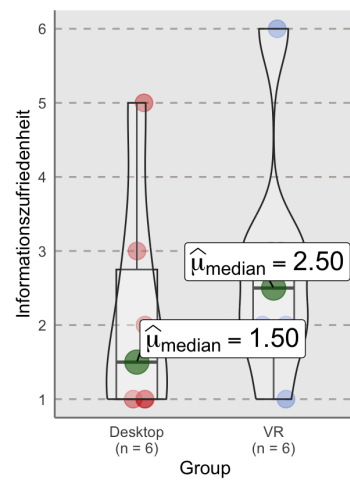
(c) Informationszufriedenheit Aufgabe 3



(d) Informationszufriedenheit Aufgabe 4



(e) Informationszufriedenheit Aufgabe 5



(f) Informationszufriedenheit Aufgabe 6

Abbildung 5.20: Die Informationszufriedenheit für die einzelnen Aufgaben aller Probanden, im Vergleich zwischen Desktop und VR

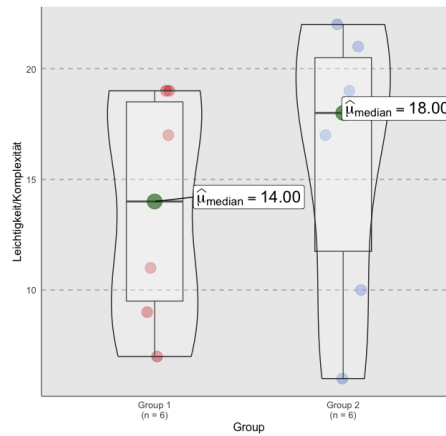


Abbildung 5.21: Vergleich der Komplexität zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)

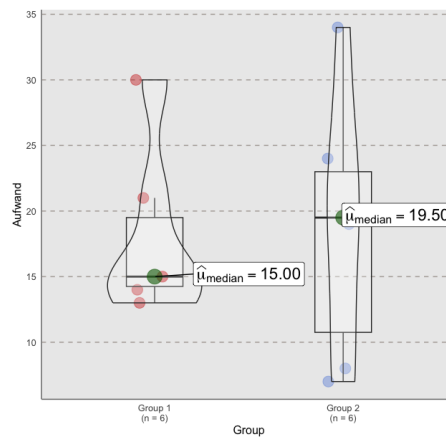


Abbildung 5.22: Vergleich des Aufwands zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)

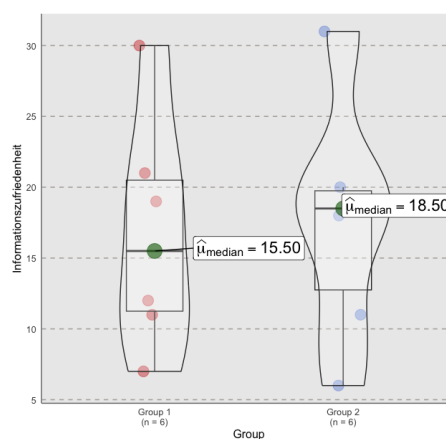


Abbildung 5.23: Vergleich der Informationszufriedenheit zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)

5.6.5 Korrelationen

Nun untersuchen wir noch mögliche Korrelationen zwischen den unabhängigen und abhängigen Variablen. Dafür nutzen wir den *Kendall-Tau-Korrelationskoeffizient* mit $p = 0,05$, da unsere Variablen Ordinalskalenniveau haben.

- **Korrelation zur Zeit**
Hier wurde eine positive Korrelation zwischen dem Geschlecht und der benötigten Zeit bei Aufgaben mit der VR-Hardware festgestellt. Weibliche Probanden benötigten also mehr Zeit für die Aufgaben mit der VR-Hardware (5.1).
- **Korrelation zur Korrektheit**
Hier wurde eine positive Korrelation zwischen der SEE-Erfahrung und der Korrektheit bei den Aufgaben mit der Desktop-Hardware festgestellt. Dies könnte daran liegen, dass Probanden, die mit SEE schon vorher Erfahrung hatten und an die Art der Aufgaben gewöhnt sind (5.2).
- **Korrelation SUS**
Hier wurde keine Korrelation festgestellt.
- **Korrelation ASQ-Komplexität**
Hier wurde keine Korrelation festgestellt.
- **Korrelation ASQ-Aufwand**
Hier wurde keine Korrelation festgestellt.
- **Korrelation ASQ-Informationszufriedenheit**
Hier wurde keine Korrelation festgestellt.

Kendall-Tau-Korrelationskoeffizient: Der Kendall-Tau-Korrelationskoeffizient testet, ob ein Zusammenhang zwischen zwei Variablen besteht, die mindestens ein ordinales Skalenniveau besitzen (DATAtab-Team (2024), Walther (2022)). 1 ist dabei eine stark positive Korrelation, 0 keine Korrelation und -1 eine stark negative Korrelation.

Unabhängige Variablen	benötigte Zeit Desktop		benötigte Zeit VR	
	τ	p	τ	p
Informatik Zugehörigkeit	-0,44	0,09	-0,17	0,5
Alter	-0,11	0,62	-0,08	0,73
Geschlecht	0,36	0,17	0,59	0,02
höchster Bildungsabschluss	0,02	0,94	-0,09	0,7
SEE-Erfahrung	-0,15	0,56	-0,27	0,28
VR-Erfahrung	0,05	0,82	0,2	0,42
Videospiel-Erfahrung	0,15	0,53	-0,11	0,64

Table 5.1: Korrelation zu der benötigten Zeit für Desktop und VR (Grün = positive Korrelation, Rot = negative Korrelation, Schwarz = keine Korrelation)

Unabhängige Variablen	Korrektheit Desktop		Korrektheit VR	
	τ	p	τ	p
Informatik Zugehörigkeit	0,02	0,93	0,38	0,15
Alter	0,16	0,51	-0,23	0,32
Geschlecht	-0,19	0,5	-0,22	0,4
höchster Bildungsabschluss	0,06	0,81	-0,38	0,12
SEE-Erfahrung	0,53	0,05	0,06	0,8
VR-Erfahrung	-0,16	0,54	-0,15	0,55
Videospiel-Erfahrung	-0,06	0,81	0,19	0,43

Table 5.2: Korrelation der Korrektheit für Desktop und VR (Grün = positive Korrelation, Rot = negative Korrelation, Schwarz = keine Korrelation)

Unabhängige Variablen	SUS Desktop		SUS VR	
	τ	p	τ	p
Informatik Zugehörigkeit	-0,23	0,39	0	1
Alter	-0,27	0,25	0,41	0,08
Geschlecht	0,32	0,22	0	1
höchster Bildungsabschluss	-0,06	0,81	-0,04	0,88
SEE-Erfahrung	0,02	0,93	0,21	0,4
VR-Erfahrung	0,09	0,71	0,22	0,37
Videospiel-Erfahrung	-0,24	0,34	0,16	0,53

Tabelle 5.3: Korrelation des SUS-Scores für Desktop und VR (**Grün** = positive Korrelation, **Rot** = negative Korrelation, Schwarz = keine Korrelation)

Unabhängige Variablen	ASQ-Komplexität Desktop		ASQ-Komplexität VR	
	τ	p	τ	p
Informatik Zugehörigkeit	-0,39	0,14	0,04	0,86
Alter	-0,29	0,23	-0,1	0,67
Geschlecht	0,47	0,08	-0,32	0,23
höchster Bildungsabschluss	-0,12	0,64	0,1	0,7
SEE-Erfahrung	-0,17	0,5	-0,13	0,61
VR-Erfahrung	0,08	0,76	-0,26	0,3
Videospiel-Erfahrung	0,14	0,58	0,16	0,53

Tabelle 5.4: Korrelation der ASQ-Komplexität für Desktop und VR (**Grün** = positive Korrelation, **Rot** = negative Korrelation, Schwarz = keine Korrelation)

Unabhängige Variablen	ASQ-Aufwand Desktop		ASQ-Aufwand VR	
	τ	p	τ	p
Informatik Zugehörigkeit	-0,3	0,26	-0,09	0,73
Alter	-0,41	0,08	-0,22	0,36
Geschlecht	0,45	0,09	0,12	0,64
höchster Bildungsabschluss	-0,06	0,81	0	1
SEE-Erfahrung	-0,27	0,31	-0,32	0,21
VR-Erfahrung	0,15	0,55	-0,15	0,55
Videospiel-Erfahrung	-0,2	0,43	0	1

Tabelle 5.5: Korrelation des ASQ-Aufwand für Desktop und VR (**Grün** = positive Korrelation, **Rot** = negative Korrelation, Schwarz = keine Korrelation)

Unabhängige Variablen	ASQ-Info.-Zufr. Desktop		ASQ-Info.-Zufr. VR	
	τ	p	τ	p
Informatik Zugehörigkeit	-0,46	0,08	-0,48	0,07
Alter	-0,34	0,15	-0,1	0,67
Geschlecht	0,2	0,45	-0,22	0,4
höchster Bildungsabschluss	-0,32	0,2	-0,18	0,48
SEE-Erfahrung	-0,36	0,17	-0,24	0,35
VR-Erfahrung	-0,1	0,71	-0,46	0,06
Videospiel-Erfahrung	0,24	0,34	0,24	0,34

Tabelle 5.6: Korrelation der ASQ-Informationszufriedenheit für Desktop und VR (**Grün** = positive Korrelation, **Rot** = negative Korrelation, Schwarz = keine Korrelation)

5.6.6 *Kommentare der Teilnehmer*

In diesem Kapitel gehen wir kurz auf die Kommentare der Teilnehmer ein. Dazu gab es am Ende der Studie ein Textfeld, in dem bei Bedarf ein Kommentar abgegeben werden konnte. Die Kommentare werden dabei in ihrer unveränderten Form vorgestellt.

- **„1. Feedback für Aktionen, Auswahl, etc. 2. Größeren Tisch, um Unterordner besser anwählen zu können. 3. Funktionen in einem Drop-down Menü auflisten.“**

Mit Feedback für Aktionen war hier vor allem die New-Edge-Aktion gemeint. Bei dieser hatten Probanden öfter anfangs Probleme, da sie nicht wussten, ob sie schon einen Block als Startpunkt ausgewählt haben, oder ob einfach nichts passiert ist. Hier wäre also tatsächlich eine mögliche Erweiterung, ähnlich wie beim Verschieben von Blöcken, zu simulieren, wie eine Kante vom Startpunkt aus aussehen würde, zu dem Punkt, zu dem der User grade zeigt. Den Effekt von größeren Tischen ist indirekt schon durch Zoomen zu erreichen. Evtl. wurde im Abschnitt „Steuerung“ übersehen, dass dies möglich ist. Jedoch wäre es auch eine Möglichkeit, dass man den Usern auch als Option bereitstellt, die Tischgröße flexibel zu ändern, sodass es den individuellen Ansprüchen entspricht. In Bezug auf den letzten Punkt „Funktionen in einem Drop-down-Menü auflisten“, lässt sich sagen, dass es schon ein Einstellungs-Menü gibt, in welchem die einzelnen Funktionen und ihre Tastenbelegungen aufgelistet zu sehen sind. Dies wussten die Probanden jedoch nicht, aber wäre durchaus sinnvoll gewesen, als Tipp im Abschnitt „Steuerung“ des Fragebogens zu hinterlegen. Dort fehlen allerdings auch noch ausführliche Beschreibungen zu den einzelnen Aktionen, was jedoch relativ schnell zu lösen ist. So ein Menü fehlt für VR komplett, wäre aber auch hier sehr sinnvoll, da sich der User dort komplett in der virtuellen Welt befindet und diese nicht verlassen möchte, um die Steuerung nachzugucken.

- **„standard tree view Unterstützung, zoom für tablet“**

Mit Standard Tree View Unterstützung ist gemeint, dass die TreeView zu jeder Zeit auf einem separaten Tablet offen ist und nicht erst geöffnet werden muss. Dies wäre eine gute Erweiterung, da es ein sehr zentrales Menü ist und wenn dies schon am Anfang offen ist, weiß der User auch direkt, dass es existiert und erkennt möglicherweise schneller, welche Vorteile es hat. In Bezug auf den zweiten Punkt „Zoom für Tablet“ lässt sich wieder sagen, dass es im Prinzip schon indirekt möglich ist, indem der User das Tablet näher an sich ran holt. Anstatt zoomen zu realisieren, wäre es wahrscheinlich effektiver, das Tablet skalieren zu können, da dann nicht jedes Menü einzeln angepasst werden muss. Der Aufwand der Implementierung wäre dabei relativ gering.

- **„Insgesamt fand ich das VR System besser zu Benutzen als die Computersteuerung. Bei beiden ist aber Einarbeitungszeit notwendig.“**

- **„Visuelle Wahrnehmung in VR merkbar besser (höhen erkennen).“**

- **„Die Taste zum Auswählen muss trotz des Menus dasselbe sein.“**

Das Feedback gaben auch andere Probanden, die es nicht als Kommentar geschrieben haben, daher wurde es nach Abschluss der Studie umgesetzt, in der Hoffnung die Benutzerfreundlichkeit in der Hinsicht zu verbessern.

- **„Einfacher, wenn man die Steuerung erst mit Maus und Tastatur lernt. Die Mausbewegung sollte beim bewegen der Kamera gespiegelt sein.“**

Die Bemerkung zu der Mausbewegung kam öfter auf. Hier lässt sich die Vermutung äußern, dass die jeweiligen Probanden dies aus anderen Videospiele als natürlicher wahrnehmen. Dieser Punkt wäre also auch eine mögliche Verbesserung.

- *** Auflösung zu gering für längere Arbeitsdauer. Kunsthallen-umgebung eher störend als hilfreich. UI-Bedienung in VR-Ansicht inkonsistent und unintuitiv. Tastaturbedienung umständlich (aber oft notwendig). Kein ersichtlicher Vorteil von Tischübersicht ggü. Auflistung. Eigener Körper stört in Vogelperspektive, außerdem ist der Kontrast zu gering.“**

Die erwähnte geringe Auflösung ist wahrscheinlich der Hardware geschuldet, bzw. der Grafikkarte, welche genutzt wurde, um die Studie durchzuführen, aber dennoch ein valider Punkt und wohl eine der größten Hürden, die VR-Variante massentauglich zu gestalten. Als Alternative zu der Kunsthallen-umgebung könnten noch weitere, evtl. auch schlichtere Umgebungen bereitgestellt werden, sodass der User sich aussuchen kann, in welchem Umfeld die Code-Cities dargestellt werden sollen. Wie oben schon erwähnt, wurde die UI-Bedienung nach der Studie ein Stück weit einheitlicher gemacht, sodass dieser Punkt hoffentlich abgehakt werden kann. Zu der Tastaturbedienung lässt sich nur spekulieren, was genau gemeint war, evtl. waren nur die Tastenbelegungen nicht ideal für denjenigen Probanden. Da in dieser Studie der Fokus auf die Interaktionen und nicht auf dem tatsächlichen Kern von SEE, also den Metriken lag, kam, auch wenn es zu Anfang der Studie erklärt wurde, die Vermittlung des Nutzens der Code-Cities evtl. zu kurz. Auch wäre denkbar, dass die Aufgaben den Sinn von SEE nicht optimal rübergebracht

haben. Zu dem vorletzten Punkt wäre eine schnelle Lösung, einen Toggle einzubauen, der den Körper des Avatars bei Bedarf transparent macht. Gleiches existiert schon für den rechten Arm. Der Kommentar zu dem Kontrast könnte auch auf die Hardware (Bildschirm) zurückzuführen sein, könnte andernfalls aber auch als zusätzliche Einstellung, wie auch bei anderen Videospielen im Einstellungsmenü zur Verfügung gestellt werden.

- **„Konsistentere Tastenbelegungen, bessere Ordnerstruktur (nicht so klein)“**
Wahrscheinlich war der erste Teil auch auf die Menüs-Interaktionen in VR bezogen und sollte damit abgehakt sein. Zu dem zweiten Teil wurde auch schon bei einem vorherigen Kommentar Bezug genommen.
- **„Menu-Controlls für VR nur mit den linken Controller und Steuerung etc. rechts.“**
Dies wäre auch eine denkbare Alternative zu der aktuellen Controller-Belegung. Theoretisch kann diese nach Belieben den eigenen Wünschen angepasst werden.
- **„Scroll umdrehen bitte! Margins von den Directories erhöhen, damit man besser auswählen kann. fliegen in vr bitte. delete Button rausnehmen. Information über Buttons von vr Controllern anzeigen auf Knopfdruck oder permanent für Orientierung“**
Das Scrollen umzukehren, könnte eine weitere Einstellung im Einstellungsmenü sein, sodass dies jeder User selber nach Belieben einstellen kann. Die Margins der Directories zu erhöhen sollte keine Schwierigkeit sein und wäre für die Koordination in der Code-City wahrscheinlich eine gute Erweiterung. Fliegen in VR wurde bisher noch nicht implementiert, um den User nicht zu sehr zu überfordern, wäre aber eine mögliche Erweiterung. Mit Delete-Button ist der Knopf im Context-Menü gemeint, mit dem Blöcke direkt gelöscht werden können. Da die Menü-Steuerung für Personen ohne bzw. mit wenig VR-Erfahrung frustrierend sein kann, wäre es tatsächlich eine Überlegung wert, diesen Knopf zwar nicht komplett zu entfernen, diesen aber als letzten Eintrag im Context-Menü anzuzeigen, damit dieser nicht so oft aus Versehen geklickt wird. Eine Anzeige über die Controller-Belegung wäre eine gute Erweiterung, da, wie oben schon erwähnt, dann User nicht die virtuelle Welt verlassen müssen, um sich mit der Steuerung vertraut zu machen.

5.7 THREATS TO VALIDITY

In diesem Kapitel schauen wir uns noch einmal die „Threats to Validity“ an, also mögliche Gefahren für die Validität der gesammelten Daten, die im Rahmen dieser Studie erfasst und ausgewertet wurden. Wir orientieren uns dabei an den internen und externen Validitäten, die Rainer Koschke in der Vorlesung „Softwaretechnik“ an der Universität Bremen vorgestellt hat [Koschke \(2023/2024\)](#). Bei der internen Validität liegt dabei der Fokus auf der Kausalität, also ob wirklich davon ausgegangen werden kann, dass es kausale Zusammenhänge zwischen den von uns beobachteten abhängigen und unabhängigen Variablen in Bezug auf die Hardware Desktop und VR gibt. Bei der externen Validität hingegen geht es um die Anwendbarkeit unserer Ergebnisse auf andere Kontexte.

5.7.1 Interne Validität

- **Störvariable**

Eine mögliche Störvariable ist, dass manche Probanden Brillenträger waren und somit evtl. einen Nachteil bei der VR-Hardware hatten. Dies wurde im Fragebogen leider nicht protokolliert. Auch gab es mit der VR-Hardware gelegentlich leichte Verzögerungen, was mit der Hardware des benutzten Computers zusammenhängt. Bei den ersten beiden Probanden gab es außerdem einen kleinen technischen Fehler im Fragebogen, dort musste bei einer Frage mehrere Zahlen eingefügt werden, die durch ein Komma getrennt sind, was bei den ersten beiden Probanden jedoch nicht richtig funktionierte. Dies wurde im Anschluss behoben. Die Daten konnten trotzdem verwendet werden, da der Studieninitiator die Daten der Probanden noch einmal separat notiert hat. Des Weiteren ist fraglich, ob alle Probanden ihre Erfahrung mit Videospiele und VR richtig eingeschätzt haben.

- **Historie/Zeit**

Die einzelnen Studiendurchführungen begannen alle im Zeitraum zwischen 10 Uhr und 16 Uhr, wobei sich immer danach gerichtet wurde, wann die jeweiligen Probanden Zeit hatten. Der Einfluss der Uhrzeit kann also ausgeschlossen werden. Jedoch wurde die Studie in der vorlesungsfreien Zeit vor Ort in der Universität durchgeführt, weshalb vermutet werden kann, dass ein Großteil der Probanden sich gerade in der Prüfungsphase befanden und dadurch evtl. weniger Konzentration für diese Studie hatten.

- **Sättigung**

Da die Studie im Schnitt eine Stunde gedauert hat, ist es nicht auszuschließen, dass Ermüdungseffekte aufgetreten sind. Jedoch wurde die letzte Aufgabe von allen Probanden korrekt beant-

wortet, was dagegen spricht. Auch bei den Aufgabenteilen mit VR-Hardware ist es denkbar, dass dieser die Probanden mehr Energie kosteten als Aufgabenteile mit der Desktop-Hardware, da die Probanden erstens stehen müssen, um die VR-Hardware zu nutzen und außerdem ihren Körper mehr einsetzen müssen, als bei der Desktop-Hardware. Es ist also nicht auszuschließen, dass Probanden, die erst die VR-Hardware benutzen mussten, danach schon weniger Energie für den zweiten Teil der Studie hatten und sich das auf diesen negativ auswirkt, in Form von bspw. mehr Fehlern oder benötigte Zeit bei Aufgaben. Diesbezüglich konnte zwar kein signifikanter Unterschied zwischen den Gruppen gefunden werden, ist aber dennoch nicht auszuschließen.

- **Wiederholtes Testen**

Es wurde zwar versucht, Lerneffekte größtmöglich durch die zufällige Zuweisung der Hardware-Reihenfolge zu vermeiden, da jedoch die Art der Aufgaben mit den beiden Hardware-Alternativen gleich ist, bis auf die Wahl der Ordner/Dateien, ist es nicht auszuschließen, dass diesbezüglich Lerneffekte aufgetreten sind. Dazu konnte auch kein signifikanter Unterschied in den Gruppen festgestellt werden, kann jedoch trotzdem nicht komplett ausgeschlossen werden.

- **Selektion**

In unseren vorherigen Untersuchungen zeigte sich, dass in Bezug auf die demographischen Angaben keine signifikanten Unterschiede festzustellen waren, daher können wir ausschließen, dass es dadurch einen Einfluss auf die danach erhobenen Daten gab.

- **Experimentatoreinfluss**

Da nur zwei der Probanden den Studieninitiator flüchtig kannten, ist auszuschließen, dass dies Einfluss auf die erhobenen Daten hat.

5.7.2 Externe Validität

- **Repräsentanz**

Hier lässt sich positiv festhalten, dass der Frauenanteil in der Informatik, der laut Tiesler (n.d.) in 2021 bei 19% lag, abgedeckt wurde und die meisten Probanden noch keine Erfahrung mit SEE hatten. Jedoch bestand der Großteil der Probanden aus Studierenden, was nicht sehr repräsentativ für die Allgemeinheit sein dürfte.

- **Situation**

Da von 10 Uhr bis 16 Uhr alle Zeiten abgedeckt wurden, sollte es in dieser Hinsicht repräsentativ sein. Der Ort war jedoch an manchen Uhrzeiten stärker besucht als an anderen. Hier wäre es

noch interessant gewesen, die Studie in einem separaten Raum ohne zusätzliche Personen durchzuführen, da dies wohl ein realistischerer Ort wäre um SEE zu benutzen.

- **Reaktivität**

Der sogenannte Hawthorne-Effekt, bei dem sich in unserem Fall, Probanden anders verhalten, wenn sie unter Beobachtung stehen [Hemmerich \(2022\)](#), ist bei dieser Studie nicht auszuschließen, da der Studieninitiator sich direkt neben den Probanden befand. Das Verhalten der Probanden ist in diesem Fall also möglicherweise nicht repräsentativ für das tatsächliche Verhalten von SEE-Nutzern.

FAZIT UND AUSBLICK

6.1 FAZIT

Mit dieser Studie sollte die Forschungsfrage: „Wie unterscheidet sich die Usability der Interaktionen in SEE zwischen Desktop-Computern und VR-Hardware?“ beantwortet werden. Dabei zeigte sich, dass die Desktop-Hardware in allen Punkten besser abgeschnitten hat, da Aufgaben sowohl schneller und zum Teil auch korrekter beantwortet wurden, als mit der VR-Hardware. Außerdem wurde auch durch den Post-Task und Post-Study Fragebogen gezeigt, dass die Desktop-Hardware in Bezug auf Usability besser abschneidet, als die VR-Hardware, beide jedoch noch viel Verbesserungspotenzial haben, da einerseits die VR-Hardware nach [Bangor u. a. \(2009\)](#) (S. 114-123) im inakzeptablen Bereich und andererseits die Desktop-Hardware im marginalen Akzeptanzbereich liegt (6.1). Als Feedback gab es jedoch auch die Rückmeldung, dass bspw. die visuelle Wahrnehmung mit der VR-Hardware merkbar besser sei, als mit der Desktop-Hardware und diese somit auch ihre Vorteile hat.

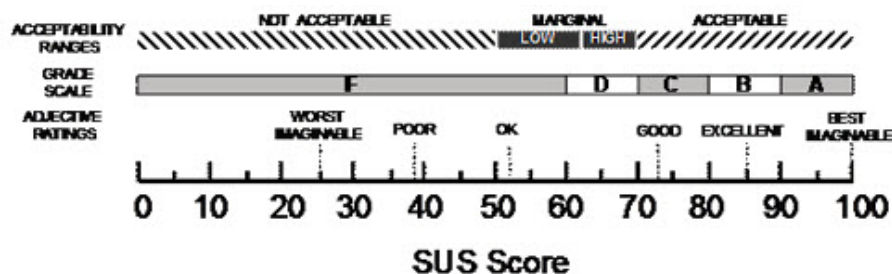


Abbildung 6.1: Akzeptanzbereiche nach [Bangor u. a. \(2009\)](#) (S. 114-123)

6.2 AUSBLICK

Wie Abschnitt schon erwähnt gibt es einige kleinere und größere Erweiterungen die zukünftig in SEE umgesetzt werden könnten. Diese werden hier noch einmal kurz aufgelistet:

- **Simulierung von neuen Kanten**
- **Vergrößerbare Tische**
- **Erweiterung der Einstellungen, sowohl allgemein, als auch für Controlls und Integration für VR**
- **Seperates Tablet für die TreeView**
- **Skalierfunktion für das Tablet**
- **Toggle um den Avatar transparent zu machen**
- **Erhöhung der Margins von Directories**
- **Fliegen in VR**
- **Verschieben des Delete-Buttons im Context-Menü**
- **Anzeige für die Controller-Belegung in VR**



GLOSSAR

- Canvas** Ein Canvas ist in Unity ein Container, der verschiedene UI-Elemente aufnehmen kann. [10](#), [21](#)
- Code-City** Code-Cities sind dreidimensionale Tree-Maps, durch die eine Software und ihre einzelnen Bestandteile dargestellt werden können. Ihr Abbild erinnert dabei an eine Stadt, in der die Gebäude einzelne Softwarekomponenten darstellen. Die Größe, aber auch andere Attribute der Gebäude wird dabei durch Software-Metriken dargestellt. [3](#), [7–10](#), [18–20](#), [48](#), [49](#)
- Convenience-Sampling** Convenience-Sampling ist eine Stichprobenmethode, bei der die Wahl der Probanden nicht zufällig ist, sondern die Wahl hauptsächlich von der Verfügbarkeit oder der geografischen Nähe der Probanden abhängt [Nikolopoulou \(2023\)](#). [16](#)
- Edge** Eine Edge (=Kante) verbindet immer zwei Nodes miteinander und stellt dabei Abhängigkeiten zwischen diesen dar. [7](#), [9](#), [18](#), [19](#), [47](#)
- Fisher-Pitman-Randomisierungstest** Der Fisher-Pitman-Randomisierungstest testet, ob zwei unabhängige Stichproben aus derselben Grundgesamtheit stammen. [30](#), [37](#), [38](#)
- GameObject** GameObjects sind die Grundbausteine in Unity. Sie werden mit Komponenten und Variablen zum Leben erweckt und können in der Spielwelt alles von Beleuchtung oder Wasser zu spielbaren Avataren darstellen. [4](#), [5](#), [9](#)
- Kendall-Tau-Korrelationskoeffizient** Der Kendall-Tau-Korrelationskoeffizient testet, ob ein Zusammenhang zwischen zwei Variablen besteht, die mindestens ein ordinales Skalenniveau besitzen [DATAtab-Team \(2024\)](#), [Walther \(2022\)](#). 1 ist dabei eine stark positive Korrelation, 0 keine Korrelation und -1 eine stark negative Korrelation. [43](#)
- Node** Ein Node ist ein einzelnes Element, innerhalb der Code-City, also z.B. eine Klasse oder eine Methode eines Softwareprojektes. Seine Erscheinungsform wird durch vorher festgelegte Metriken bestimmt. Bspw. könnte die Höhe die Anzahl an Codezeilen sein. [7–10](#)

Reflexion-Graph Mit dem Reflexion-Graph kann eine Implementierung gegen die eigentlich festgelegte Architektur verglichen werden.

7

Spatial presence Mit Spatial presence (= Räumliche Präsenz) ist das Gefühl gemeint, dass der User sich tatsächlich in der virtuellen Welt befindet und mit dieser interagieren kann [Caroux \(2023\)](#).

5

Unity Unity ist eine Spiele-Engine, mit der es möglich ist, 2D- und 3D-Spiele aber auch interaktive Simulationen zu erstellen. [3](#), [5](#)

Unity-Event Unity-Events sind ereignisgesteuert, d.h. dadurch ist es möglich auf Aktionen in der Spielwelt zu reagieren. Dabei ist die Besonderheit, dass keine festen Abhängigkeiten im Code entstehen müssen (außer es ist so gewollt) und Methoden dynamisch im Unity-Editor einem Event zugewiesen werden können. [4](#), [5](#), [9](#)

AKRONYME

ASQ Der After-Scenario-Questionnaire besteht aus drei Fragen und wird nach jeder Aufgabe ausgefüllt. [16](#), [21](#), [22](#), [37](#), [43](#), [45](#), [46](#), [61](#)

SEE Die Software Engineering Experience ist eine virtuelle Welt, in der sich Softwareentwickler treffen können und Software in Form von Code-Cities dargestellt wird. [1](#), [3](#), [7](#), [8](#), [10](#), [15](#), [17](#), [18](#), [20–22](#), [26](#), [43](#), [48](#), [51–54](#), [59](#)

SUS Die System Usability Scale besteht aus 10 Fragen und wird in unserem Fall nach jedem Hardware-Abschnitt ausgefüllt. [15](#), [16](#), [18](#), [37](#), [43](#), [45](#), [61](#)

VR Virtual-Reality ist eine computergenerierte Realität, mit der User durch eine VR-Brille interagieren können. [1](#), [3](#), [5](#), [7–10](#), [12–17](#), [19](#), [21](#), [23–29](#), [31–51](#), [53](#), [54](#), [59–61](#)

ABBILDUNGSVERZEICHNIS

Abbildung 4.1	Vergleich des Mode-Menüs zwischen VR und Desktop	12
Abbildung 4.2	Vergleich des CodeWindows zwischen VR und Desktop	13
Abbildung 4.3	Vergleich des Context-Menü zwischen VR und Desktop	14
Abbildung 5.1	Die Ordner und Blöcke für Aufgabe 1 und Aufgabe 4.	19
Abbildung 5.2	Die Ordner für Aufgabe 2 und Aufgabe 5.	20
Abbildung 5.3	Die Blöcke für Aufgabe 3 und 6.	21
Abbildung 5.4	Aufbau in der Universität	22
Abbildung 5.5	Darstellung von Fragen innerhalb von SEE	22
Abbildung 5.6	Vergleich der Informatikzugehörigkeit in den Gruppen und insgesamt	23
Abbildung 5.7	Vergleich des Alters in den Gruppen und insgesamt	24
Abbildung 5.8	Vergleich des Geschlechts in den Gruppen und insgesamt	24
Abbildung 5.9	Vergleich des höchsten Abschlusses in den Gruppen und insgesamt	25
Abbildung 5.10	Vergleich und Verteilung der SEE-Erfahrung in den Gruppen und insgesamt	26
Abbildung 5.11	Vergleich und Verteilung der VR-Erfahrung in den Gruppen und insgesamt	27
Abbildung 5.12	Vergleich und Verteilung der Videospiele-Erfahrung in den Gruppen und insgesamt	28
Abbildung 5.13	Vergleich der Korrektheit zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)	29
Abbildung 5.14	Die Korrektheit der Antworten für die einzelnen Aufgaben aller Probanden, angegeben in Prozent, im Vergleich zwischen Desktop und VR	31
Abbildung 5.15	Vergleich der Bearbeitungszeit zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)	32
Abbildung 5.16	Die Bearbeitungszeit für die einzelnen Aufgaben aller Probanden, angegeben in Minuten, im Vergleich zwischen Desktop und VR	36

Abbildung 5.17	SUS-Score im Vergleich zwischen Desktop- und VR-Hardware	38
Abbildung 5.18	Die Komplexität für die einzelnen Aufgaben aller Probanden, im Vergleich zwischen Desktop und VR	39
Abbildung 5.19	Der Aufwand für die einzelnen Aufgaben aller Probanden, im Vergleich zwischen Desktop und VR	40
Abbildung 5.20	Die Informationszufriedenheit für die einzelnen Aufgaben aller Probanden, im Vergleich zwischen Desktop und VR	41
Abbildung 5.21	Vergleich der Komplexität zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)	42
Abbildung 5.22	Vergleich des Aufwands zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)	42
Abbildung 5.23	Vergleich der Informationszufriedenheit zwischen den beiden Gruppen (Gruppe 1 = Desktop→VR, Gruppe 2 = VR→Desktop)	42
Abbildung 6.1	Akzeptanzbereiche nach Bangor u. a. (2009) (S. 114-123)	53

TABELLENVERZEICHNIS

Tabelle 5.1	Korrelation zu der benötigten Zeit für Desktop und VR (Grün = positive Korrelation, Rot = negative Korrelation, Schwarz = keine Korrelation)	44
Tabelle 5.2	Korrelation der Korrektheit für Desktop und VR (Grün = positive Korrelation, Rot = negative Korrelation, Schwarz = keine Korrelation) . . .	44
Tabelle 5.3	Korrelation des SUS-Scors für Desktop und VR (Grün = positive Korrelation, Rot = negative Korrelation, Schwarz = keine Korrelation) . . .	45
Tabelle 5.4	Korrelation der ASQ-Komplexität für Desktop und VR (Grün = positive Korrelation, Rot = negative Korrelation, Schwarz = keine Korrelation)	45
Tabelle 5.5	Korrelation des ASQ-Aufwand für Desktop und VR (Grün = positive Korrelation, Rot = negative Korrelation, Schwarz = keine Korrelation) . . .	46
Tabelle 5.6	Korrelation der ASQ-Informationszufriedenheit für Desktop und VR (Grün = positive Korrelation, Rot = negative Korrelation, Schwarz = keine Korrelation)	46

LITERATURVERZEICHNIS

- [Armstrong 2023] ARMSTRONG, M.: *VR Gaming Destined to Remain Niche?* statista. Februar 2023. – URL <https://www.statista.com/chart/29401/vr-gaming-penetration-rate-vs-video-game-penetration-global/>. – Abgerufen am 19. September 2024
- [Bangor u. a. 2009] BANGOR, Aaron ; KORTUM, Phil ; MILLER, James: Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. In: *J. Usability Stud.* 4 (2009), April, S. 114–123. – URL <https://uxpajournal.org/de/determining-what-individual-sus-scores-mean-adding-an-adjective-rating-scale/>
- [Berkman u. a. 2020] BERKMAN, Mehmet ; CATAK, Guven ; EREMEKTAR, Mistik: Comparison of VR and Desktop Game User Experience in a Puzzle Game: "Keep Talking and Nobody Explodes". In: *AJIT-e Online Academic Journal of Information Technology* 11 (2020), Oktober, S. 180. – URL https://www.researchgate.net/publication/344899877_Comparison_of_VR_and_Desktop_Game_User_Experience_in_a_Puzzle_Game_Keep_Talking_and_Nobody_Explodes
- [Bueckle u. a. 2021] BUECKLE, Andreas ; BUEHLING, Kilian ; SHIH, Patrick C. ; BÖRNER, Katy: 3D virtual reality vs. 2D desktop registration user interface comparison. In: *PLOS ONE* 16 (2021), Oktober, S. 1. – URL <https://doi.org/10.1371/journal.pone.0258103>. – Abgerufen am 19. September 2024
- [Caroux 2023] CAROUX, Loïc: Presence in video games: A systematic review and meta-analysis of the effects of game design choices. In: *Applied Ergonomics* 107 (2023), S. 1. – URL <https://www.sciencedirect.com/science/article/pii/S0003687022002599>. – Abgerufen am 19. September 2024
- [DATAtab-Team 2024] DATATAB-TEAM: *Kendalls Tau*. DATAtab. 2024. – URL <https://datatab.de/tutorial/kendalls-tau>. – Abgerufen am 19. September 2024
- [Galperin 2021] GALPERIN, Falko: *Visualisierung von Code-Smells in Code-Cities*, Universität Bremen, Bachelorarbeit, 2021. – S. 34–39, 58.
- [Galperin u. a. 2022] GALPERIN, Falko ; KOSCHKE, Prof. Dr. R. ; STEINBECK, Marcel: Visualizing Code Smells: Tables or Code Cities? A Controlled Experiment. In: *VISSOFT 2022* (2022). – URL <https://github.com/uni-bremen-agst/VISSOFT2022/blob/>

- [main/_Vissoft2022CodeSmellsCodeCities.pdf](#). – Abgerufen am 19. September 2024
- [Hemmerich 2022] HEMMERICH, Wanja A.: *Hawthorne-Effekt*. StatistikGuru. August 2022. – URL <https://statistikguru.de/lexikon/hawthorne-effekt.html>. – Abgerufen am 19. September 2024
- [Koschke 2023/2024] KOSCHKE, Prof. Dr. R.: *Softwaretechnik*. 2023/2024. – URL <https://www.uni-bremen.de/life/life-themenfelder/informatik-digitale-medien-digitalisierung/life-flex-moduluebersicht/softwaretechnik>. – Abgerufen am 19. September 2024
- [Krause 2024] KRAUSE, Michel: *Integration of an interactive whiteboard (drawable) for SEE*, Universität Bremen, Masterarbeit, 2024. – S. 117–121.
- [Lewis 1991] LEWIS, James R.: Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the ASQ. In: *SIGCHI Bull.* 23 (1991), S. 78–79. – URL <https://doi.org/10.1145/122672.122692>. – Abgerufen am 19. September 2024
- [Nikolopoulou 2023] NIKOLOPOULOU, K.: *Was ist Convenience Sampling? | Definition und Beispiele*. Scribbr. Juni 2023. – URL <https://www.scribbr.com/methodology/convenience-sampling/>. – Abgerufen am 19. September 2024
- [Sauro 2011] SAURO, J.: *Measuring Usability with the System Usability Scale (SUS)*. measuringu.com. Februar 2011. – URL <https://measuringu.com/sus/>. – Abgerufen am 19. September 2024
- [Tiesler n.d.] TIESLER, J.: *Wo sind die Frauen in der IT?* get-in-it. n.d.. – URL <https://www.get-in-it.de/magazin/arbeitswelt/it-arbeitsmarkt/wo-sind-die-frauen-in-der-it#>. – Abgerufen am 19. September 2024
- [Walther 2022] WALTHER, B.: *Kendall-Tau-Korrelationskoeffizient in R berechnen*. bjoernwalther.com. Dezember 2022. – URL <https://bjoernwalther.com/kendall-tau-korrelationskoeffizient-in-r-berechnen/>. – Abgerufen am 19. September 2024