Visualisierungsinstitut der Universität Stuttgart (VISUS)

Universität Stuttgart
Allmandring 19
D–70569 Stuttgart

Bachelorarbeit

Einfluss der Kugelanzahl auf die visuelle Hervorhebungstechnik HiveFive

Jannik Schoger

Studiengang:	Softwaretechnik
Prüfer/in:	Prof. Dr. Michael Sedlmair
Betreuer/in:	Nina Dörr, M.Sc., Markus Wieland, M.Sc.
Beginn am:	2. Februar 2024
Beendet am:	2. August 2024

Kurzfassung

Um in der Virtuellen Realität (VR) Objekte mittels eines diegetischen Schwarms hervorzuheben, haben Lange et al. die HiveFive Methode entwickelt. Unser Ziel war es zu untersuchen, welche Auswirkungen eine Änderung der Anzahl der Kugeln aus denen der Schwarm besteht, nach sich ziehen würde. Dafür haben wir in Unity einen Versuchsaufbau entwickelt und anschließend eine Studie über die Crowdsourcing-Plattform Prolific durchgeführt. Wir haben dabei gefunden, dass die Zeit bis die Hervorhebung erkannt wird nur bis maximal zehn Kugeln signifikant sinkt und danach keine signifikanten Unterschiede aufweist, oder, im Fall von 13 Kugeln, wieder signifikant ansteigt. Die Anzahl an falsch erkannten Objekten ist von 3 auf 7 Kugeln signifikant gefallen, sonst haben wir keine signifikanten Unterschiede gefunden.

Inhaltsverzeichnis

1. Einleitung **11**

2. Implementierung **13**
2.1. Unity 13
2.2. Datenspeicherung 20
2.3. Webserver 20
2.4. BwCloud 20
2.5. Prolific 21

3. Studie **23**
3.1. Ziel 23
3.2. Studiendesign 23
3.3. Studienablauf 24
3.4. Prolific während der Studie 25
3.5. Teilnehmer 26

4. Ergebnisse und Diskussion **27**
4.1. Zeit zum Erkennen 27
4.2. Fehler 32

5. Zusammenfassung und Ausblick **37**
5.1. Zusammenfassung 37
5.2. Limitationen 37
5.3. Ausblick 37

Literaturverzeichnis **39**

A. Ishihara-Ergebnisse **41**

B. Weitere Daten der Auswertung **47**
B.1. Erkennzeit 47
B.2. Fehleranzahl 50
B.3. Likertfragebogen 51

Abbildungsverzeichnis

2.1. Ishihara Test mit den Tafeln 1, 9, 11, 19 und 23 15
2.2. Szene mit Likert-Fragebogen 16
2.3. Umgebung der Trainingsaufgabe 16
2.4. Hauptumgebung der Studie 17

4.1. Übersicht der Erkennzeit mit Scatter- und Boxplot. 28
4.2. Konfidenzintervalle von Tukey's HSD. 30
4.3. Konfidenzintervalle des robusten Post-Hoc Tests. 31
4.4. Übersicht der Fehleranzahl mit Scatterplot. 33
4.5. Tukey's HSD für Fehleranzahl pro Kugelanzahl. 34
4.6. Robuster Post-Hoc Test für Fehleranzahl pro Kugelanzahl. 35

A.1. Ishihara Tafel 1 (12) 41
A.2. Ishihara Tafel 9 (74) 42
A.3. Ishihara Tafel 11 (6) 43
A.4. Ishihara Tafel 19 (Leer/2/8) 44
A.5. Ishihara Tafel 23 (42) 45

B.1. Übersicht über alle Datenpunkte mit einer Erkennzeit über null, mittels Scatterplots und Boxplots. 47
B.2. Antwortsverteilung auf Likertfrage 1. 52
B.3. Antwortsverteilung auf Likertfrage 2. 52
B.4. Antwortsverteilung auf Likertfrage 3. 53
B.5. Antwortsverteilung auf Likertfrage 4. 53
B.6. Antwortsverteilung auf Likertfrage 5. 54
B.7. Antwortsverteilung auf Likertfrage 6. 54

Tabellenverzeichnis

4.1. Zusammenfassung für Erkennzeit pro Kugelanzahl. 27
4.2. Shapiro-Wilk Testergebnisse pro Kugelanzahl. 28
4.3. Paarweiser t-Test mit Bonferroni-Korrektur. 30
4.4. Zusammenfassung für Fehleranzahl pro Kugelanzahl. 32
4.5. Shapiro-Wilk Testergebnisse zu Fehleranzahl pro Kugelanzahl. 32
4.6. Paarweiser t-Test mit Bonferroni-Korrektur für Fehleranzahl. 34

B.1. Tukey's HSD mit Konfidenzintervallen. Signifikante Unterschiede sind markiert. 48
B.2. Robuster Post-Hoc Test mit Konfidenzintervallen. Signifikante Unterschiede sind markiert. 49
B.3. Tukey's HSD für Fehleranzahl pro Kugelanzahl. Signifikante Unterschiede sind markiert. 50
B.4. Robuster Post-Hoc Test für Fehleranzahl pro Kugelanzahl. Signifikante Unterschiede sind markiert. 51

1. Einleitung

Es existieren viele Methoden um in der Virtuellen Realität (VR) die Aufmerksamkeit der Nutzer auf ein bestimmtes Ziel zu lenken. Viele dieser von Woodworth und Borst [WB24] verglichenen Methoden verwenden dabei Reize die nicht in der dargestellten Umgebung existieren würden, sondern nur zum Lenken der Aufmerksamkeit in die Welt gesetzt wurden. Die Methode diegetische Reize zu verwenden, wie ein Bienenschwarm [LSGB20] oder ein Glühwürmchen [NMH+16], wird jedoch nur selten verwendet. Der dabei erwähnte Bienenschwarm ist die von Lange et al. [LSGB20] entwickelte HiveFive-Methode, die entwickelt wurde um in VR Objekte markieren und hervorheben zu können, ohne dabei den Nutzer aus dem Geschehen zu ziehen. Dabei wird ein Schwarm aus 10 gelben Kugeln verwendet, der um das hervorzuhebende Objekt herumkreist.

Unser Ziel war es zu ermitteln welche Auswirkungen eine Änderung der Anzahl der Kugeln im Schwarm nach sich ziehen würde. Dafür haben wir eine dreidimensionale Umgebung in der Spieleengine Unity entwickelt, in der wir die Kugelanzahl verändern, und somit eventuelle Auswirkungen erfassen konnten. Diese entwickelte Studie haben wir dann auf der Crowdsourcingplattform Prolific veröffentlicht um anhand einer großen Anzahl Nutzern möglichst genau Aussagen treffen zu können.

Diese Arbeit ist in folgender Weise gegliedert: Wir beschreiben unsere Implementierung der Software zum Durchführen der Studie in Kapitel 2. Das genaue Design und den Ablauf der Studie geben wir in Kapitel 3 an, die gefundenen Ergebnisse werten wir in Kapitel 4 aus. Schließlich fassen wir in Kapitel 5 die Ergebnisse unserer Arbeit zusammen, gehen auf einige Probleme ein und stellen mögliche weitere Forschungsansätze vor.

2. Implementierung

Da wir eine möglichst großen Anzahl von Teilnehmern angestrebt haben, haben wir uns entschieden die Studie über die Crowdsourcing-Platform Prolific zu verbreiten. Um unsere Studie überhaupt durchführen zu können, benötigten wir eine Möglichkeit eine eigene Szene im dreidimensionalen Raum darzustellen und die Interaktionen der Teilnehmer zu analysieren. Da wir nicht verlangen wollten, dass alle Teilnehmer ein Programm herunterladen und installieren müssen, kam nur eine vollständig webbasierte Lösung in Frage. Aufgrund dieser Anforderungen haben wir uns entschieden die Studie in Form einer Unity-Applikation für WebGL zu implementieren. In diesem Kapitel gehen wir auf die einzelnen Komponenten unserer Implementierung ein.

2.1. Unity

Unity ist eine Spiele-Engine die dafür ausgelegt ist eigene 2D oder 3D Spiele zu entwickeln. Hierbei kann man von anderen Nutzern bereitgestellte Komponenten aus dem Unity Asset Store verwenden, oder eigene Funktionalitäten via C# implementieren. Hierbei werden in sich abgeschlossene Teile des Unity Projekts meist in eigene Szenen zusammengefasst. Diesen beinhalten dann alles was für einen gewissen Abschnitt der resultierenden Unity-Applikation relevant ist, so wie etwa die darzustellenden 3D-Objekte, Bedienelemente welche Informationen anzeigen, und Skripte die die Interaktion des Nutzers innerhalb dieser jeweiligen Szene steuern.

2.1.1. Globale Verfügbarkeit für bestimmte Daten

In unserer Implementierung existieren mehrere Szenen die auf dem gleichen Informationsstand sein müssen und teilweise auch Daten an eine andere Szene geben sollen. Hierfür verwenden wir ein C# Skript welches eine statische Klasse implementiert und keinem `GameObject` einer Szene zugeordnet ist. Dadurch ist diese Klasse immer erreichbar und kann Informationen zum Durchlauf der Studie speichern. In dieser Klasse definieren wir Konstanten die die Funktion unserer Studie steuern, so wie die Rundenanzahl oder die Stufen unserer Unabhängigen Variable. Wir verwenden diese Klasse zum Bereitstellen von Hilfsmethoden welche wir in mehreren Szene benötigen oder um die ID des aktuellen Teilnehmers über Szenen hinweg zu speichern. Ebenso speichern wir hier die Listen der anfallenden Loggingdaten, damit diese aus einer anderen Szene zu einem späteren Zeitpunkt hochgeladen werden können.

2.1.2. Nutzeroberflächen

In unserer Implementierung benutzen wir zum Darstellen von Text und ähnlichen 2D-Elementen generell die standardmäßige Methode für Unity. Hierbei definieren wir die gewünschten Bausteine als Kindelemente zu einem `Canvas`-Element und positionieren sie wie gewünscht. Diese Kombination aus Canvas, automatisch erstelltem, grauem Hintergrund und unseren UI-Elementen wird bei der Ausführung bildschirmfüllend angezeigt. Für Bilder verwenden wir die `RawImage`-Klasse, für andere Element-Bausteine verwenden wir das integrierte `TextMeshPro`-Paket. Diese vorgefertigten Elemente implementieren bereits standardmäßig die nötigen Funktionen zum Erkennen von Nutzereingaben und lassen sich leicht an den Anwendungszweck anpassen. Dies lässt sich erreichen indem man die gewünschten Skripte dem übergeordneten `Canvas` oder einem einzelnen Element zuordnet. Mehrfach war es nötig bestimmte Elemente ein- oder auszublenden. Dies haben wir erreicht, indem wir thematisch zusammenhängende Elemente einem leeren Objekt untergeordnet haben. Dieses konnten wir dann nach Belieben aktivieren und alles zeigen oder deaktivieren und somit verstecken.

2.1.3. Szenenübersicht

Die Implementierung unserer Studie besteht aus einem Unity-Projekt mit neun Szenen. Dabei haben wir jeden Schritt den die Teilnehmer durchführen sollten, in eine eigene Szene aufgeteilt. Dies sind zwei Szenen mit einer 3D-Umgebung, eine für die anfängliche Übungsaufgabe und eine für den Hauptteil der Studie. Die anderen Szenen bestehen nur aus 2D-Nutzeroberflächen, welche wir wie in Abschnitt 2.1.2 erklärt erstellt haben. In zwei dieser Szenen erheben wir zusätzliche Daten, die anderen Szene dienen zum Anzeigen von Informationen oder als Weiterleitung in eine nachfolgende Szene. Im Folgenden gehen wir auf Details in unserer Implementierung der verwendeten Szenen ein.

Farbsehtest

Während unserer Studie unterziehen wir unsere Teilnehmer einem kleinen Test um mittels fünf Ishihara-Tafeln [Ish85] ihre Farbsicht abschätzen zu können. Diese Tafeln bestehen aus vielen kleinen Punkten in ähnlichen Farben und haben meist ein bestimmtes Muster wie eine Zahl oder Linie in leicht abgegrenzter Farbe in der Mitte. Wenn jemand diesen minimalen Farbunterschied übersieht, und somit das enthaltene Muster falsch oder gar nicht erkennen kann, deutet dies auf eine Farbsehschwäche hin. Für unseren Test verwenden wir hochauflösende Versionen der Tafeln 1, 9, 11, 19 und 23 aus der großen Sammlung von 38 Tafeln [Ish85] aus der Public Domain[1].

Diesen Ishihara-Test haben wir als 2D Unity Szene implementiert. Dabei haben wir die Anweisungen zur Durchführung als Textblock an den oberen Rand platziert. Darunter haben wir unsere fünf verwendeten Tafeln, jeweils mit zwei Dropdownmenüs gruppiert, angeordnet. Diese Dropdownmenüs verwenden wir als Eingabemethode für die Teilnehmer. Dabei enthalten diese die Ziffern von 0

[1] Speziell haben wir die vom Wikipedia-Nutzer „Nicoguaro“ am 15. Mai 2008 hochgeladenen Versionen benutzt[Ish08a; Ish08b; Ish08c; Ish08d; Ish08e].

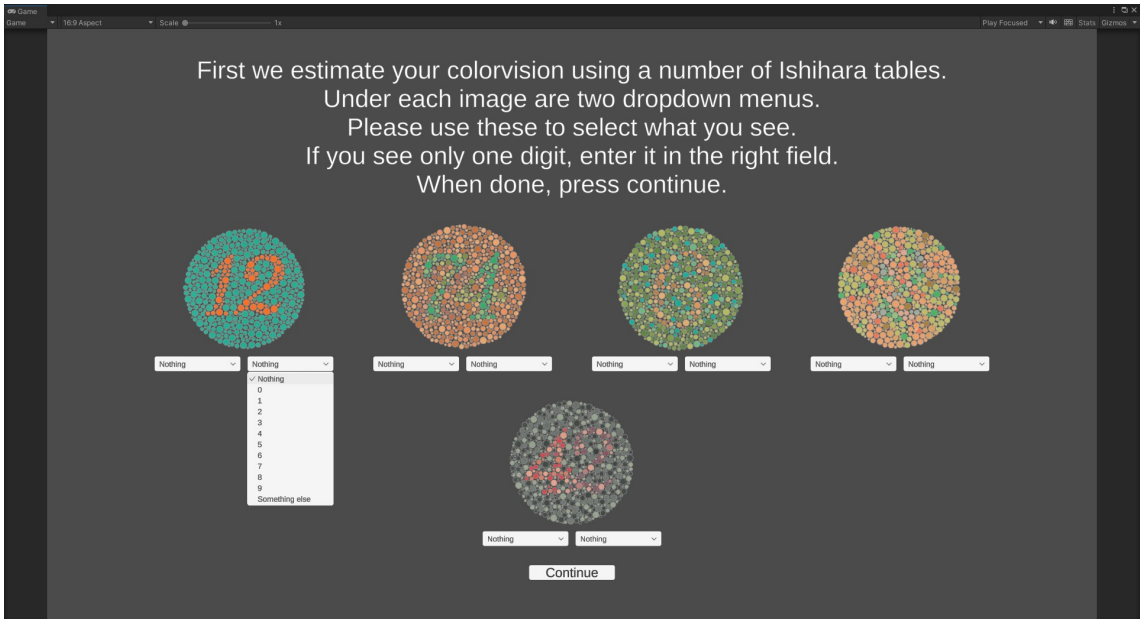


Abbildung 2.1.: Ishihara Test mit den Tafeln 1, 9, 11, 19 und 23

bis 9, sowie zwei zusätzliche Optionen um anzugeben dass man nichts, oder etwas ganz anderes gesehen hat. Diese Methode zur Eingabe haben wir gewählt um unsinnige Eingaben zu erschweren.

Likert-Fragebogen

Um die subjektiven Eindrücke der Teilnehmer ebenfalls zu erfassen, haben wir zum Schluss der Studie einen Likert-Fragebogen ausfüllen lassen. Dabei haben wir jeweils eine Aussage in einem Textfeld angezeigt und in einem danebenliegenden Dropdownmenü die entsprechende Antwort erfasst. Die Dropdownmenüs die wir für diesen Teil verwen haben, hatten zusätzlich zu den 5 Punkten unserer Likertskala ein „-“ als zuerst ausgewähltes Element. Falls ein Teilnehmer versucht diesen Teil zu überspringen ohne alle Antworten zu geben, ist das „-“ noch ausgewählt und wird erkannt. Der Teilnehmer wird dann darauf hingewiesen alle Aussagen zu bewerten.

Übungsumgebung

Für die Übungsaufgabe haben wir uns für eine möglichst einfache Umgebung und Hervorhebungstechnik entschieden um den Teilnehmern nur den Ablauf zu vermitteln, ohne zu große Hinweise auf die genaue Art der Hauptaufgabe zu geben. Die Umgebung der Szene der Übungsaufgabe besteht nur aus einem einfarbigem Boden und neun einfarbigen Würfeln im sonst leeren Raum. Die Würfel sind in einem 3x3 Gitter angeordnet, welches mittig auf dem Bildschirm angezeigt wird. Hier benutzen wir keine besondere Methode zum Hervorheben, sondern färben den Würfel rot ein.

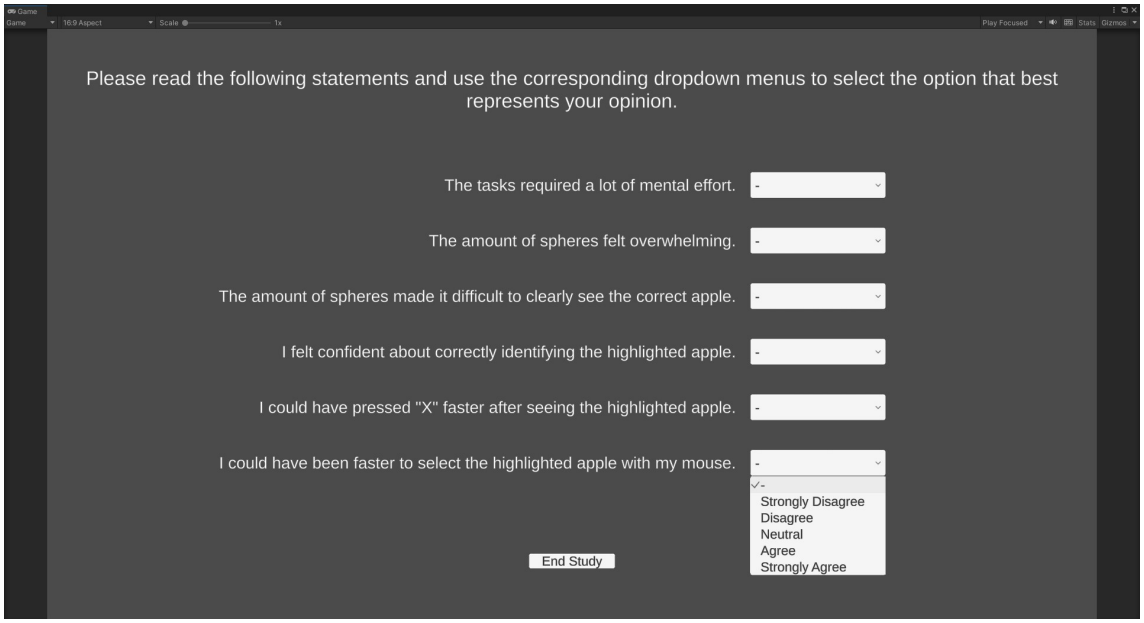


Abbildung 2.2.: Szene mit Likert-Fragebogen

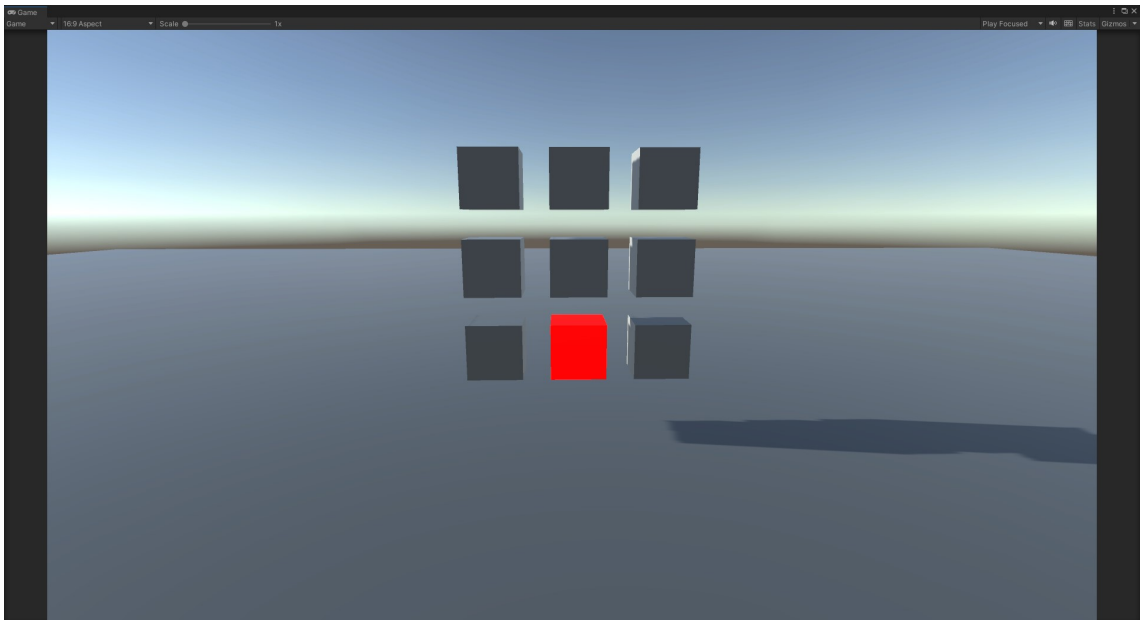


Abbildung 2.3.: Umgebung der Trainingsaufgabe

Abbildung 2.4.: Hauptumgebung der Studie

Studienumgebung

Lange et al. benutzten eine Umgebung aus dem „Windridge City“ Asset Pack als Kulisse für ihre Studie [LSGB20]. Da diese Asset Pack nicht mehr verfügbar ist, mussten wir eine eigene Umgebung erstellen und verwenden. Diese Szene, in der der Haupteil der Studie abläuft, besteht aus einem dreidimensionalen Gelände auf dem fünf Bäume mit jeweils etwa 18 Äpfeln stehen. Das Gelände haben wir mit dem Terraingenerator „World Machine“ [Wor22] manuell erstellt, die zugehörige Textur haben wir mit dem dazugehörigen Makro Coastal Overlay [nik10] generiert. Das resultierende 3D-Modell haben wir in Blender nachbearbeitet um die Anzahl der Polygone, mit möglichst wenig sichtbarem Qualitätsverlust, drastisch zu reduzieren. Für alle Bäume benutzen wir das gleiche Modell, speziell `URP_Tree_1_LOD_0.fbx` aus dem Asset Pack „URP Tree Models“ [ALI24]. Die Bäume haben wir manuell so positioniert, dass alle Bäume etwa gleich groß und äquidistant zueinander und zu der Kamera sind. Die Äpfel haben wir dem Asset Pack „Survival Kit Lite“ [Row17] entnommen und manuell in die Bäume gesetzt. Dabei haben wir darauf geachtet, dass aus Sicht der Kamera kein Apfel verdeckt wird oder im Schatten liegt. Außerdem haben wir die Äpfel so positioniert, dass sie möglichst gleichmäßig in den Bäumen verteilt sind. Zum Hervorheben benutzen wir in dieser Szene die HiveFive-Methode [LSGB20] mit einem nicht-biologischem Schwarm. Außerdem haben wir einen durchsichtigen `Canvas` mit einem versteckten Schriftzug als Hintergrund gesetzt. Damit ist es uns möglich diesen Schriftzug anzuzeigen und somit während einem Durchlauf eine Nachricht einzublenden.

2.1.4. Auswählen, Hervorheben und Anklicken

In der Übungsaufgabe auch als während dem Hauptteil der Studie selber, müssen wir ein Objekt in der jeweiligen Umgebung zufällig auswählen und hervorheben. Ebenso müssen die Klicks auf Objekte registriert und ausgewertet werden. Die grundelegende Methode für diese Schritte ist dabei in beiden Aufgaben gleich.

Auswählen des hervorzuhebenden Objekts

Sowohl in der Übungsaufgabe auch als in der tatsächlichen Studie gruppieren wir die möglichen Objekte als Kind-Elemente eines leeren Objekts. Dieses Objekt hat keine tatsächliche Präsenz im Raum, enthält jedoch ein Skript welches ausgeführt werden kann. Beim Starten der jeweiligen Szene kennt dieses Skript automatisch alle Kind-Objekte des leeren Objekts dem dieses Skript gehört.

Im Falle der Übungsaufgabe sind dies die neun Würfel. Da die Kinder, und somit auch deren Anzahl, dem Skript bekannt sind, kann nun eine zufällige Zahl zwischen 0 und der Anzahl an Kindern gezogen werden, welche wir dann als Index des ausgewählten Würfels interpretieren. Damit können wir nun auf diesen Würfel zugreifen und ihn uns in einer Variable merken.

Da wir in der Studienumgebung die Möglichkeit haben wollten die fünf Gruppen von Äpfeln unterscheiden zu können, haben wir diese in Kategorien eingeteilt und dementsprechend hierarchisch angeordnet. Als Wurzel haben wir ein Objekt mit dem Skript und drei Kind-Objekten, repräsentativ für die drei Kategorien in die die Bäume aufgeteilt werden können: In der Mitte, seitlich der Mitte und ganz am Rand. Diese drei Kategorien haben dann die jeweils ein leeres Objekt für die entsprechenden Bäume der Kategorie als Kind, welches dann die Äpfel am jeweiligen Baum als Kinder enthält. Hier wollen wir darauf achten, dass alle 3 Kategorien gleich oft ausgewählt werden. Dazu wählen wir zufällig eine der Kategorien aus und überprüfen wie oft diese schon ausgewählt wurde. Falls die maxmiale Anzahl, ein Drittel der gesamten Durchläufe, bereits erreicht wurde, wird solange eine zufällige Kategorie gewählt bis die Gewählte noch nicht oft genug durchgeführt wurde. Nun kombinieren wir alle Äpfel aller Bäume dieser gewählten Kategorie in eine einzige Liste und wählen ein zufälligen Apfel daraus aus. Dies machen wir um innerhalb einer Kategorie alle Äpfel gleich zu gewichten. Den nun ausgewählten Apfel speichern wir in einer Variable zur späteren Nutzung.

Hervorheben

Im vorherigen Schritt haben wir ein Objekt ausgewählt, nun müssen wir es hervorheben. In der Übungsaufgabe erreichen wir das, indem wir über die Variable auf den Würfel zugreifen und dessen Farbe auf Rot setzen, bevor wir dessen zuvorige Farbe ebenfalls in einer Variable speichern. Da diese Studie der HiveFive Methode gilt, verwenden wir diese um den ausgewählten Apfel in der Hauptszene hervorzuheben. Wir haben unsere Implementierung auf der Implementierung von Reynolds' Boids Algorithmus [Rey87] in Unity durch Lange et al. [LSGB22] basiert. Hierbei benutzen wir das Wurzelobjekt aller Äpfel als `_swarmManager` und den ausgewählten Apfel als `swarmHolder`. Die Größe des Schwarms setzen wir hier mit dem Parameter `size` auf die zu Beginn für diesen Teilnehmer festgelegte Größe.

In beiden Aufgaben sollen die Teilnehmer „X“ auf ihrer Tastatur drücken bevor sie das Objekt anklicken. Dadurch wird die Hervorhebung deaktiviert damit nicht zuerst „X“ gedrückt und dann das Objekt gesucht werden kann. In der Übungsaufgabe entfernen wir die Hervorhebung indem wir die Farbe auf die anfangs gemerkte vorherige Farbe setzen. Für die Studie setzen wir die Größe des Schwarms auf null, wodurch im nächsten angezeigten Frame alle Kugel verschwinden.

Anklicken

Das Erfassen der Klicks funktioniert für beide Szenen identisch. Wir überprüfen für jeden Frame, ob die linke Maustaste gedrückt wurde und führen in diesem Fall einen Raycast durch. Dafür verwenden wir Unitys `Pyhsics.Raycast()` Funktion, welche das erste Objekt unter dem Mauszeiger zurückgibt. Hierbei haben wir die Szene so aufgebaut, dass dabei ausschließlich die Würfel in der Übungsaufgabe und die Äpfel im Haupteil erkannt werden können. Das zurückgegebene Objekt können wir nun mit dem in der Variablen gespeicherten Objekt vergleichen. Falls diese beiden Objekte identisch sind, zählt der Klick als ein Treffer und die aktuelle Runde wird erfolgreich beendet. Falls es zwei verschiedene Objekte sind, gilt der Klick als verfehlt und wird dementsprechend behandelt.

2.1.5. HiveFive

Da unser Ziel es war HiveFive zu untersuchen, verwenden wir diese Methode zum Hervorheben während dem Haupteil unserer Studie. Dabei verwenden wir die Implementierung von Reynolds' Boids Algorithmus [Rey87] in Unity durch Lange et al. [LSGB22] als Basis für unsere Implementierung. Wir benutzen dabei ein sonst leeres Objekt dem alle Äpfel untergeordnet sind als „Swarm Manager“, welche die grundlegenden Parameter beinhaltet und den die Funktion des Schwarm implementiert. Den hervorzuhebenden Apfel verwenden wir als „Swarm Holder“, welcher die als zentraler Punkt des Schwarms fungiert. Bedingt durch die Unterschiede zwischen der Darstellung in VR und 2D-Bildschirm, ebenso wie durch das notwendige Aufbauen einer neuen Umgebung, mussten wir die Parameter des Schwarms zur besseren Sichtbarkeit anpassen. Dabei haben wir die Größe der einzelnen Kugeln von 5 mm auf 25 mm erhöht und zwei weiter Parameter angepasst. Die Geschwindigkeit mit der die Kugeln sich bewegen haben wir mit dem Parameter `speed` von 0.35 auf 1.0 erhöht. Der Parameter `spread` beeinflusst in welchem Radius die Kugeln umherkreisen, diesen haben wir von 0.1 auf 0.2 erhöht. Durch diese Änderungen haben wir eine größtmöglichste Ähnlichkeit des visuellen Eindrucks unserer Implementierung, verglichen mit der Implementierung durch Lange et al., angestrebt.

2.1.6. Datenerfassung

Jeder Teilnehmer ruft unsere Studie mit deren jeweiliger Prolific-ID als URL-Parameter auf. Dadurch erfassen wir diese und können alle Daten dieses Teilnehmers zusammen zuordnen und speichern. Während der Studie zeichnen wir, mit Zeitstempeln versehen, in einigen Szenen kontinuierlich die Mausposition und wichtige Ereignisse in einer Liste auf. Diese Ereignisse sind etwa Knopf- und Tastendrücke oder die Ergebnisse des Raycasting. Dies ist der Fall bei dem Ishihara-Test und dem Likert-Fragebogen, ebenso wie bei den zwei Aufgaben in den dreidimensionalen Umgebungen. Wir erstellen ebenfalls jeweils eine kleine Zusammenfassung für den Farbsehtest

und den Fragebogen in der die Ergebnisse knapp dokumentiert werden. Ebenso fassen wir für alle Runden der Übungsaufgabe und Studie die erfassten Daten in einer gemeinsamen Liste zusammen. Dabei messen wir die Zeit zwischen dem Start der Szene mit bereits aktivierter Hervorhebung und dem Drücken von „X“. Ebenfalls messen wir die Zeit ab Szenenbeginn bis das korrekte Objekt angeklickt wird und die dafür benötigten Klicks. In dieser Zusammenfassung speichern wir auch welches Objekt hervorgehoben wurde, ob etwas für die Auswertung zu beachten ist und für welche Runde der Studie diese Werte gemessen wurden. Zum gegebenen Zeitpunkt wandeln wir die Textrepräsentation dieser Liste in ein Byte-Array um und laden dieses an unseren Webserver hoch.

2.2. Datenspeicherung

Um die Daten zu erfassen benutzen wir einen Webserver auf den die in Unity aufgezeichneten Daten hochgeladen werden. Hier benutzen wir das Flask Framework [Pal24] um in Python einen Webserver mit eigener Funktionalität zu implementieren. Dieser Server empfängt den von Unity initiierten POST-Request und extrahiert dabei die ID eines jeden Teilnehmers, sowie die hochgeladene Datei selber. Da während dem Durchführen der Studie mehrere Dateien entstehen, werden zur einfacheren Übersicht alle Dateien eines Teilnehmers in einem eigenen Ordner, bennant nach der jeweiligen ID, abgespeichert.

2.3. Webserver

Um Zugriff auf die Unity Applikation über das Internet möglich zu machen, benutzen wir nginx als zentralen Webserver der diese Dateien bereitstellt. Zusätzlich agiert nginx auch als Reverse-Proxy für den via Flask erstellten Webserver. Somit die beide Server über die selbe Domain erreichbar und die Einschränkungen der Same-Origin-Policy[2] werden umgangen.

2.4. BwCloud

Um die Daten der Unity-Applikation sowie die Webserver für die große Anzahl an Teilnehmern gleichzeitig über das Internet bereitzustellen benötigten wir einen leistungsfähigen Server der mit einer hohen Bandbreite an das Internet angebunden ist. Dieser wurde durch den Landesdienst bwCloud bereitgestellt. Bei diesem Dienst können sich Mitglieder von Universitäten in Baden-Württemberg registrieren und einen virtuellen Server beantragen. Wir haben uns für den Standort Mannheim, sowie die maximal kostenlosen möglichen Ressourcen mit einem CPU-Kern, 1GB RAM und 12GB Festplattenspeicher entschieden. Als Betriebssystem haben wir aufgrund bereits vorhandener Erfahrung Debian 11 gewählt.

[2] Diese verhindert, dass ein Script, welches von `https://example.com` geladen wurde, auf eine andere URL wie `https://other.example.com` ohne Weiteres zugreifen kann

Aufgrund der zurzeit hohen Anzahl an VMs an unserem Standort haben die Server standardmäßig keine externen IPv4 Adressen. Da dies bereits in den ersten Pilotstudien Probleme bereitete, haben wir manuell beim Support-Team der BwCloud eine Ausnahme erbeten. Dadurch haben wir eine IPv4 Adresse zugewiesen bekommen und konnten so die Erreichbarkeit auch für Anschlüsse ohne IPv6 garantieren.

2.5. Prolific

Prolific ist eine Crowdsourcing Plattform die Wissenschaftler mit Teilnehmern für Studien verbindet. Hierbei können die Forscher Studien anlegen und an die gewünschte Anzahl Teilnehmer verteilen. Dabei gibt man der Studie einen Titel, beschreibt was in der Studie zu tun ist und wie lange jemand voraussichtlich braucht um die Studie zu vollenden. Da die Teilnehmer für ihre Abreit fair bezahlt werden sollen, gibt man beim Erstellen zudem noch an wie viel Geld die Teilnehmer für die erfolgreiche Teilnahme erhalten, Richtwert ist dabei £9.00 pro Stunde. Außerdem kann man die Studienteilnahme auf eine bestimmte Art Gerät oder demographische Eigenschaften der Teilnehmer beschränken. Potentielle Teilnehmer können dann die Studie sehen und ihr, falls Interesse besteht, beitreten. Die Ersteller der Studie haben dann eine Liste der Teilnehmer und können die jeweiligen Einreichungen überprüfen und ablehnen oder akzeptieren und vergüten. Jeder Teilnehmer hat eine eigene ID mit welcher die Daten der Studie mit den demographischen Daten sowie dem Status der Teilnahme in Prolific zugeordnet werden kann. Um den Verlauf der Studie überprüfen zu können, stellt Prolific sogennante „Completion Codes“ bereit. Diese können beim Erstellen der Studie eingebaut und den Teilnehmern unter bestimmten Bedingungen gezeigt werden. Die Teilnehmer geben den erhaltenen Code dann in Prolific ein und registrieren somit ihren Status in der Studie. Während der laufenden Studie können die verfügbaren Plätze auch noch nachträglich erhöht werden.

2.5.1. Aufbau in Prolific

Da unsere Anweisungen und Erklärungen in der Studie auf Englisch geschrieben stehen, haben wir unsere Studie in Prolific so konfiguriert, dass nur Personen mit fließenden Englischkentnissen teilnehmen können. Da unsere Studie Maus und Tastatur benötigt, haben wir auch angegeben, dass die Teilnahme nur an einem Computer möglich ist. Da sonst keine weiteren Anforderungen an die Teilnehmer herrschen und die Resultate unserer Studie möglichst verallgemeinbar sein sollten, haben wir keine weiteren Einschränkungen vorgesehen. Die voraussichtliche Bearbeitungszeit der Studie haben wir mit zehn Minuten angegeben, die Bezahlung der Teilnehmer haben wir mit £1.50 angegeben. Die gewünschte Anzahl Teilnehmer haben wir zu Beginn auf 20 gesetzt und im Laufe unserer Studie schrittweise erhöht. Prolific hat die Möglichkeit die Studie über einen externen Service laufen zu lassen indem Teilnehmer an diesen Service weitergeleitet werden. Dafür stellt Prolific URL-Paramter bereit, die an den externen Service weitergeleitet und dort verwendet werden können. Dabei haben wir auf den Webserver mit unserer Unity-Applikation verlinkt und die ID mittels den gestellten URL-Parametern erfasst. Wir haben drei Completion Codes generiert und in unsere Studie in Unity eingebeut. Mit Hilfe dieser Completion Codes haben wir dann die Teilnehmer nach Beendigung der Studie in Gruppen aufgeteilt und an Prolific zurück geleitet. Dabei haben wir unterschieden zwischen den Teilnehmern ohne Zustimmung zur Datenerfassung, Teilnehmern

welche die Studie vollständig abgeschlossen haben und den Teilnehmern welche während der Studie ausgeschieden sind. Als letzte Gruppe haben wir die Teilnehmer, die mehrfach den Anweisungen der Übungsaufgabe nicht Folge geleistet haben, oder während einem Schritt mehr als fünf Minuten keine Aktivität gezeigt haben. Somit konnten wir abgeschlossene Einreichungen einfach zuordnen und akzeptieren. Teilnehmer in den anderen zwei Gruppen wurden dadurch automatisch gebeten sich von der Studie zurückzuziehen und somit Platz für neue Teilnehmer zu schaffen.

3. Studie

3.1. Ziel

In der von Lange et al. [LSGB20] entwickelten HiveFive Methode wird ein Schwarm von 10 Kugeln um das Objekt herum zum Hervorheben verwendet. Lange et al. haben diese Kugelanzahl sowie weitere Parameter des Schwarms mithilfe einer Pilotstudie (N=3) bestimmt, haben jedoch keine Angabe gemacht was zu dieser Entscheidung geführt hat. Unser Ziel war es zu untersuchen ob diese gewählte Anzahl optimal ist, oder welche Auswirkungen eine Änderung nach sich ziehen würde. Dabei haben wir uns folgende Fragen gestellt:

- Wie groß ist der Einfluss der Kugelanzahl auf die Zeit bis der hervorgehobene Apfel gesehen wird?
- Wie genau beeinflusst die Kugelanzahl die gemachten Fehler?

Unsere ersten Vermutungen vor der Implementierung besagten, dass viele Kugeln schneller ins Auge fallen, diese hohe Anzahl jedoch auch den Radius des Schwarms erhöht und somit auch umliegende Äpfel umfässt. Schlussendlich haben wir folgende Hypothesen formuliert:

- **H1**: Je mehr Kugeln zum Hervorheben verwendet werden, desto schneller wird das hervorgehobene Objekt erkannt.
- **H2**: Je mehr Kugeln zum Hervorheben verwendet werden, desto mehr Fehler passieren beim Anklicken.

3.2. Studiendesign

Die finale Version unserer Studie entspricht einem Within-Subjects-Design. Hierbei durchlaufen alle Teilnehmer standardmäßig 30 Runden.

3.2.1. Unabhängige Variable

Da unser Ziel ist die Auswirkung verschiedener Kugelanzahlen zu untersuchen, nehmen wir dies als unsere Unabhängige Variable. Diese Stufen der Variable wurden unseren Teilnehmern zufällig zugeordnet, wobei jeder Teilnehmer während dem gesamten Durchlauf nur eine Stufen testet. Hierbei haben wir zusätzlich zu den 10 Kugel aus der originalen Studie 6 weitere Stufen betrachtet: 3, 5, 7, 13, 16 und 20 Kugeln.

Da bei einer Kugel diese die Hälfe der Zeit nicht zusehen und somit nichts hervorheben konnte, musste unsere untere Grenze darüber liegen. Bei zwei Kugel haben sich diese beinahe sofort in einen stabilen Orbit um den Apfel eingependelt und entsprachen somit nicht dem gewünschten Aussehen eines natürlichen Schwarms. Somit haben wir 3 Kugeln als untere Grenze gewählt, welche dieses Einpendeln komplett verhindert, oder längere Zeit hinauszögert.

Durch eine Pilotstudie im Familien- und Freundeskreis haben wir 20 als eine obere Grenze gewählt. Hier haben wir durch visuelle Auswertung der Daten über 20 keinen Unterschied in der Reaktionszeit feststellen können, jedoch mehrfach die Rückmeldung erhalten, dass es sehr viele Kugeln im Bild waren.

3.2.2. Abhängige Variablen

Wir haben mehrere unserer erfassten Daten als Abhängige Variablen betrachtet. Dabei haben wir die zwei folgenden Variablen quantitativ gemessen:

- Die Zeit bis zum Drücken von „X“, welche wir auch als „Erkennzeit“ bezeichnen.
- Die Anzahl falsch angeklickter Äpfel, auch als „Fehleranzahl“ bezeichnet.

Hierbei ist zu beachten, dass vor allem diese Erkennzeit auf einer unkontrollierbaren Eingabe der Teilnehmer beruht und somit ungenauer ist als eine in kontrollierter Umgebung mit spezieller Technik durchgeführte Studie.

3.3. Studienablauf

Die Teilnehmer treten in Prolific der Studie bei und werden auf unsere Seite mit der Unity-Implementierung geleitet. Diese Seite lädt die notwendigen Daten und beginnt automatisch die Studie. Zu Beginn sehen die Teilnehmer einen Text in dem wir erklären welche Daten wir während der Studie erheben und können diesen Bedingungen durch Knopfdruck zustimmen oder ablehnen. Bei Zustimmung wechselten wir zur nächsten Szene, anderenfalls wurde die Studie abgebrochen und die Teilnehmer an Prolific zurückgeleitet.

In dieser Szene führen wir unsere Farbsehtest mittels Ishihara-Tafeln durch. Zu Beginn sehen die Teilnehmer die Anleitung zur Durchführung des Tests und können zum Beginnen mittels Knopfdruck die Ishihara-Tafel einblenden. Dabei geben die Teilnehmer mittels je zwei Dropdownmenüs an, was sie in den fünf Tafeln sehen und können den Test mit einem beim Start erschienenen Knopf beenden.

Anschließend stellen wir in einer neuen Szene die Übungsaufgabe vor und geben vor wie die Aufgabe durchzuführen ist. Zum Starten haben wir ein rotes Kreuz in der Mitte platziert auf welches die Teilnehmer ihre Maus positionieren sollen. Sobald die Teilnehmer die Maus auf dem Kreuz haben startet ein Countdown zu dessen Ende die Szene wechselt. Die Übungsaufgabe startet dann direkt mit einem durch rote Einfärbung hervorgehobenem Würfel. Sobald die Teilnehmer diesen Würfel erkennen mussten sie die Taste „X“ auf ihrer Tastatur drücken, woraufhin dessen Farbe identisch zu den restlichen Würfeln wurde. Nun sollten die Teilnehmer mit der Maus auf den zuvor eingefärbten Würfel klicken. Wenn ein Teilnehmer während dieser Übungsaufgabe den

Würfel angeklickt hat ohne zuvor „X“ zu drücken, haben wir diese Runde direkt abgebrochen und die Teilnehmer darauf hingewiesen. Ebenso haben wir darauf geachtet dass die Teilnehmer maximal jeweils 30 s zum Erkennen der Hervorhebung sowie zum Auswählen des Würfels nach dem Drücken von „X“ benötigen. Alle Teilnehmer hatten 3 Versuche diese Übung erfolgreich zu beenden, Teilnehmer die es nicht schafften haben wir aus der Studie ausgeschlossen und an Prolific zurückgeleitet. Beim korrekten Abschließen der Übung blenden wir die Erklärung der haupsächlichen Aufgabe der Studie ein, woraufhin die Teilnehmer mit Knopfdruck die Szene wechseln können.

Dies ist eine kleine Szene mit nochmal einer minimalen Erklärung der Aufgabe und einem mittig platziertem roten Kreuz. Auch hier startet ein Countdown in die nächste Szene sobald die Maus im Kreuz ist. Nach jeder Runde der Studie wird der Nutzer hierher zurückgeleitet und beginnt so jede Runde mit der Maus mittig auf dem Bildschirm.

Beim Wechsel in die Studienszene wird mittels der HiveFive-Methode ein zufälliger Apfel[1] hervorgehoben welcher erkannt und angeklickt werden soll. Wie in der Übungsaufgabe warten wir hier auf das Drücken von „X“ und deaktivieren daraufhin den Schwarm. Nun sollen die Teilnehmer den korrekten Apfel anklicken. Falls Teilnehmer mehrere falsche Äpfel anklicken nehmen wir an, dass die Teilnehmer vergessen haben welcher Apfel der richtige war und blenden im Himmel der Szene einen Schriftzug ein. Mit diesem geben wir den Teilnehmern die Möglichkeit durch Drücken von „A“ die Runde abzubrechen. Damit wird diese Runde für unsere Auswertung markiert und wiederholt[2]. Falls ein Teilnehmer klickt ohne zuvor „X“ gedrückt zu haben, wird die Runde zwar fortgeführt, aber für unsere Auswertung als ungültig markiert.

Nach Abschluss einer Runde laden wir die Rundendaten hoch und leiten die Teilnehmer zurück in die Szene mit der kurzen Erklärung und dem Kreuz.

Nachdem die Teilnehmer alle Runden abgeschlossen haben, haben wir sie zu einem Likert-Fragebogen weitergeleitet. Hier sollten die Teilnehmer ihre subjektive Meinungen zu sechs Aussagen auf je einer 5-Punkt Likertskala ausdrücken. Dabei haben wir darauf geachtet, dass alle Aussagen bewertet werden mussten. Sobald dies geschehen war, konnten die Teilnehmer die Studie beenden und wurden zurück zu Prolific geleitet.

3.4. Prolific während der Studie

Während der Durchführung der Studie gaben manche Teilnehmer über die eingebaute Messaging-Funktion an, undefinierte technische Schwierigkeiten zu erleiden. Auf Nachfrage an die Teilnehmer haben wir keine genauen Informationen zu Problemen bekommen. Wir konnten jedoch feststellen, dass bei einer großen Anzahl zeitgleicher Teilnehmer das Wechseln zwischen Szenen in Unity manchmal nicht ordnungsgemäß funktioniert. Indem wir die Anzahl an offenen Plätzen nur in kleinen Schritten erhöht haben, haben wir dieses Problem behoben.

[1] Genaueres dazu in Abschnitt 2.1.4

[2] Dabei verringern wir die Anzahl der bereits durchgeführten Runden, mit Äpfeln aus der entsprechenden Kategorie, um eins. Dadurch finden weiterhin gleich viele Runden pro Kategorie statt

3.5. Teilnehmer

Da wir unsere Teilnehmer über die Crowdsourcing-Platform Prolific rekrutiert haben, war es uns möglich eine große Anzahl an Teilnehmern anzustreben. Unser Ziel war es für jede der 7 Stufen unserer unabhängigen Variable 50 Teilnehmer zu finden. Da mehrere Teilnehmer unsere Studie abgebrochen haben, sind insgesamt 411 Personen angetreten bevor wir unsere geplanten 350 Teilnehmer erreicht haben. Bei zwei Teilnehmern fehlten die Resultate des Ishihara-Tests, weshalb wir beide Teilnehmer für dessen Auswertung ignorieren. Bei einem weiteren Teilnehmer wurden die Antworten auf den Likert-Bogen nicht auf unseren Server hochgeladen, konnten jedoch manuell nachgereicht werden. Da die Daten zur eigentlichen Studie vorhanden sind, schließen wir diese drei Teilnehmer nicht aus der Auswertung aus. Im Folgenden konzentrieren wir uns ausschließlich auf diese Teilnehmer die die Studie vollständig durchgeführt haben.

Die hier aufgeführten demographischen Daten der erfolgreichen Teilnehmer unserer Studie wurde uns von Prolific zur Verfügung gestellt. Das Durchschnittsalter der Teilnehmer war 34 Jahre (M=34.04 SD=10.52), mit allen Teilnehmern zwischen 18 und 72 Jahren. Insgesamt waren von unseren Teilnehmern 133 weiblich und 215 männlich, zwei haben keine Angabe gemacht. Insgesamt haben nur 66 Teilnehmer die offiziell korrekten Zahlen auf den Ishihara-Tafeln gesehen. Wenn wir jedoch auch übliche Alternativantworten[3] erlauben, erhöht sich diese Zahl auf 247 Teilnehmer. Für keine der Tafeln haben alle Teilnehmer die korrekte Antwort exakt gegeben, bei der Referenztafel war der Anteil an korrekten Antworten mit 324 am höchsten. Die genau Verteilung der Antworten ist in Anhang A zu sehen.

[3] 71 statt 74 für Tafel 9, sowie 2 und 8 für Tafel 19

4. Ergebnisse und Diskussion

In diesem Kapitel analysieren wir unsere erhobenen Daten und besprechen gefundenen Ergebnisse. Die Analysen haben wir mit der eigens für statistische Analysen entwickeln Programmiersprache R durchgeführt[R C24]. Dabei haben wir die zugehörige IDE RStudio verwendet[Pos24]. Unsere Grafiken haben wir mit dem „ggplot2“-Paket erstellt [Wic16].

4.1. Zeit zum Erkennen

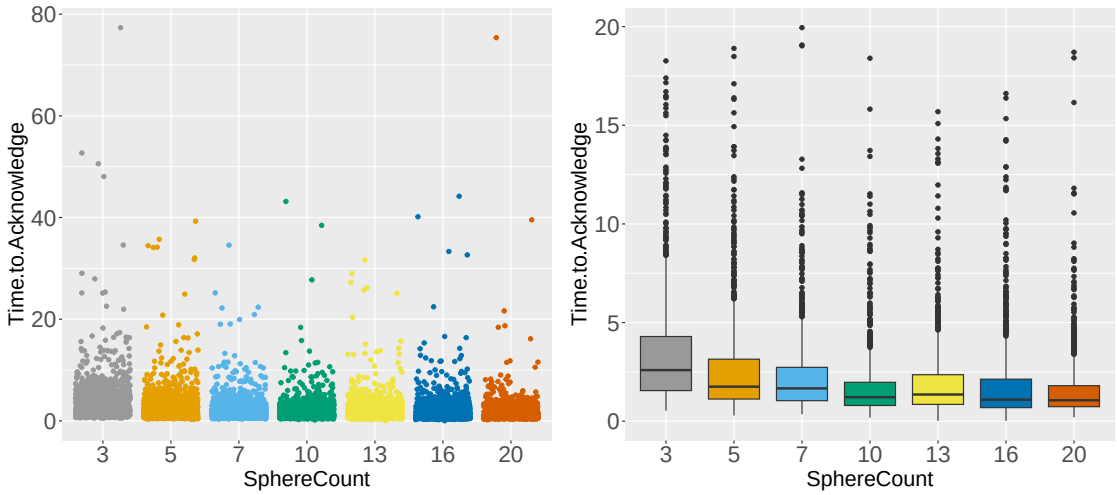
Hier geben wir unsere Auswertung der gemessenen Werte zur benötigten Erkennzeit an. Da Teilnehmer manchmal mit der Maus geklickt haben bevor sie „X“ gedrückt haben, haben wir für diese Runden keine Messung der Erkennzeit, weshalb wir diese Runden aus unserer Auswertung ausschließen. Außerdem wurde in drei Runden die Taste „X“ innerhalb des ersten Frames gedrückt und somit eine Erkennzeit von 0,0 s gemessen. Wir führen dies darauf zurück, dass hier die Taste „X“ bereits vor Beginn der Szene gedrückt gehalten wurde und schließen diese Runden ebenfalls aus. Somit stehen uns hier insgesamt 10366 Datenpunkte zur Verfügung, die wir im Folgendem alle für unsere Auswertung betrachten.

Eine grobe Übersicht über all diese Daten geben wir in Abbildung 4.1. Abbildung 4.1a zeigt alle Datenpunkt und lässt dabei einen leichten Unterschied zwischen den Gruppen erkennen. Indem wir die 44 Ausreißer über 20 s entfernen, können wir in Abbildung 4.1b diesen Unterschied mit Hilfe eines Boxplots klar aufzeigen[1]. Außerdem zeigen wir in Tabelle 4.1 pro Stufe der Kugelanzahl die Anzahl der Datenpunkte, die durchschnittliche Erkennzeit (mean) und deren Standardabweichung (SD).

Kugelanzahl	N	Erkennzeit mean	SD
3	1469	3.631662	4.256379
5	1470	2.665374	3.225207
7	1486	2.266318	2.312222
10	1498	1.778705	2.263819
13	1484	1.998167	2.405373
16	1504	1.856746	2.674857
20	1455	1.632900	2.690926

Tabelle 4.1.: Zusammenfassung für Erkennzeit pro Kugelanzahl.

[1] Ein Scatterplot ohne Ausreißer, sowie ein Boxplot mit allen Daten sind in Abbildung B.1 zu sehen.



(a) Alle validen Datenpunkte als Scatterplot.

(b) Nur Erkennzeit unter 20 s als Boxplot.

Abbildung 4.1.: Übersicht der Erkennzeit mit Scatter- und Boxplot.

Kugelanzahl	W	p-Wert
3	0.50112	<2.2e-16
5	0.52317	<2.2e-16
7	0.57745	<2.2e-16
10	0.43647	<2.2e-16
13	0.4979	<2.2e-16
16	0.43889	<2.2e-16
20	0.28843	<2.2e-16

Tabelle 4.2.: Shapiro-Wilk Testergebnisse pro Kugelanzahl.

Bereits eine visuelle Inspektion diese Messdaten deutet darauf hin, dass hier keine Normalverteilung innerhalb der, durch die Kugelanzahl gegebenen, Gruppen vorliegt. Dies haben wir durch einen Shapiro-Wilk Test pro Stufe der Kugelanzahl überprüft. Die Nullhypothese des Shapiro-Wilk Tests besagt, dass die überprüften Daten normalverteilt sind, falls der berechnete p-Wert darunter liegt, lehnen wir damit die Normalität dieser Daten ab. Deren Ergebnisse sind in Tabelle 4.2 aufgelistet und zeigen, dass alle p-Werte unter dem Signifikanzniveau von 0.05 liegen. Somit lehnen wir die Nullhypothese der Normalität für alle Gruppen ab.

Anschließend haben wir mittel Levene's Test die Homogenität der Varianzen zwischen unseren Gruppen überprüft. Die Nullhypothese dieses Tests besagt, dass die Varianzen aller Gruppen gleich sind. Mit diesem Test haben wir einen signifikanten Unterschied zwischen den Varianzen der Gruppen gefunden, $F_{Levene}(6, 10359) = 31.356$, $p < 2.2e - 16$, und lehnen somit die Hypothese der Varianzhomogenität ab.

Da wir, mit etwa 1500 Messwerten pro Gruppe, sehr große Stichproben haben, besteht die, wenn auch geringe, Möglichkeit, dass sowohl der Shapiro-Wilk Test, als auch Levene's Test, bei vergleichsweise geringen Abweichungen von der Normalität bzw. Varianzhomogenität, fälschlicherweise ein signifikantes Ergebnis finden. Aus diesem Grund führen wir mehrere unterschiedliche Tests durch

und stellen deren Ergebnisse gegenüber. Das Ziel all dieser Tests ist es zu bestimmen, ob Unterschiede zwischen den einzelnen Gruppen bestehen. Dabei trifft keiner der Tests eine Aussage zwischen welchen Gruppen Unterschiede existieren, sondern nur, dass welche existieren.

ANOVA Da die Annahmen der ANOVA durch die Nicht-Normalverteilung und Varianzheterogenität nicht erfüllt sind, lässt sich keine klare Aussage zur Korrektheit ihrer Ergebnisse machen. In manchen Fällen kann eine ANOVA dennoch als robust gelten[2], deshalb haben wir diese zur Vollständigkeit auch betrachten. Dabei haben wir einen signifikanten Effekt der Kugelanzahl auf die Erkennzeit gefunden, $F(6, 10359) = 84.42$, p <2e-16, $\omega = 0.046$.

Welch's F Dieser Test beruht auf einer Variante des bei ANOVAs verwendeten F-Ratios, welche als robust gegen Unterschiede zwischen den Varianzen einzelner Gruppen gilt [Wel51]. Diesen Test verwenden wir, da laut dem durchgeführten Levene's Test keine Homogenität der Varianzen besteht. Auch hierbei haben wir einen signifikanten Effekt der Kugelanzahl auf die Erkennzeit gefunden, $F_W(6, 4585.7) = 55.302$, p <2.2e-16.

Robuste ANOVA mittels „trimmed means“ Anders als beim berechnen einer regulären ANOVA, wird bei dieser Methode jeweils ein bestimmter Anteil der kleinsten und größten Werte einer Gruppe verworfen, bevor der Mittelwert dieser Gruppe bestimmt wird. Dies hat zur Folge, dass Ausreißer weniger Einfluss haben und somit genauere Aussage zu den Gruppen getroffen werden können. Zur Berechnung dieses Tests haben wir das „WRS2“-Paket von Mair und Wilcox [MW20] verwendet. Auch dieser Test hat einen signifikanten Unterschied zwischen den Gruppen gefunden, $F_{TM}(6, 2746.25) = 159.5225$, $p = 0$, $\omega = 0.42$.

Kruskal-Wallis Dieser Test verwendet eine andere Herangehensweise als die vorherigen Varianten der ANOVA, und ist robust gegen Verletzen der Annahmen von ANOVA [KW52]. Hier werden alle Datenpunkte, über Gruppen hinweg, aufsteigend geordnet und bekommen ihren jeweiligen Rang in dieser Sortierung zugeordet. Der Vergleich zwischen den Gruppen wird nun anhand dieses Ranges, statt des tatsächlichen Messwerts, gezogen. Hier haben wir ebenfalls einen signifikanten Unterschied zwischen den Gruppen gefunden, $H(6) = 1258.9$, p <2.2e-16.

Da all diese Tests einen Unterschied finden, müssen wir auch bestimmen zwischen welchen Gruppen genau diese Unterschiede liegen. Auch hier gibt es wieder mehrere Post-Hoc Tests, die auf Verletzen der Annahmen unterschiedlich reagieren.

Paarweiser t-Test Hierbei wird jede Kombination von zwei Gruppen einem t-Test unterzogen, um diese Gruppen auf einen Unterschied zwischen ihnen zu prüfen. Dabei besagt die Nullhypothese, das zwischen den zwei verglichenen Gruppen keine Unterschiede herrschen. Da somit sehr viele Tests anfallen, steigt die Wahrscheinlichkeit mindestens eine Nullhypothese fälschlicherweise abzulehnen stark an. Ein Mittel dagegen ist die Bonferroni-Korrektur, bei der alle berechneten p-Werte mit der Gesamtanzahl der Tests multipliziert wird. Dadurch muss der originale p-Wert

[2] Kapitel 10.3 in *Discovering Statistics Using R*[And12]

	3	5	7	10	13	16
5	<2e-16	-	-	-	-	-
7	<2e-16	0.0040	-	-	-	-
10	<2e-16	2.1e-15	9.6e-05	-	-	-
13	<2e-16	9.3e-09	0.2494	0.8209	-	-
16	<2e-16	7.2e-13	0.0024	1.0000	1.0000	-
20	<2e-16	<2e-16	7.2e-08	1.0000	0.0137	0.7576

Tabelle 4.3.: Paarweiser t-Test mit Bonferroni-Korrektur.

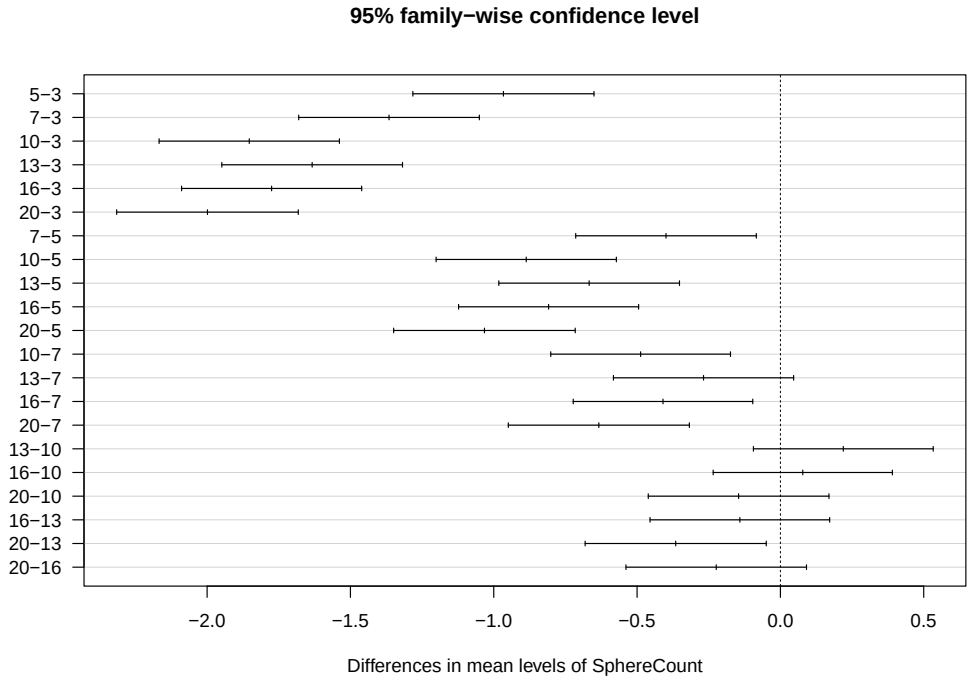


Abbildung 4.2.: Konfidenzintervalle von Tukey's HSD.

jedes einzelnen Tests viel geringer sein, um am Ende unter dem Signifikanzniveau zu liegen und somit einen Unterschied zwischen den zwei Gruppen des Tests zu finden. Die Ergebnisse unserer Auswertung sind in Tabelle 4.3 zu sehen. Hierbei sind die Vergleiche mit signifikanten Unterschieden hervorgehoben.

Tukey's HSD Dieser Test gibt die Konfidenzintervalle der paarweisen Vergleiche der Gruppen aus. Wenn dieses Intervall dabei die Null nicht kreuzt, wird der Unterschied zwischen diesen Gruppen als signifikant angenommen. In Abbildung 4.2 ist deutlich zu sehen, dass in allen außer sechs Vergleichen ein signifikanter Unterschied gefundenwurde. Dieses Ergebnis stimmt mit dem der paarweisen t-Tests überein. Die genauen Werte der Konfidenzintervalle sind in Tabelle B.1 zu sehen.

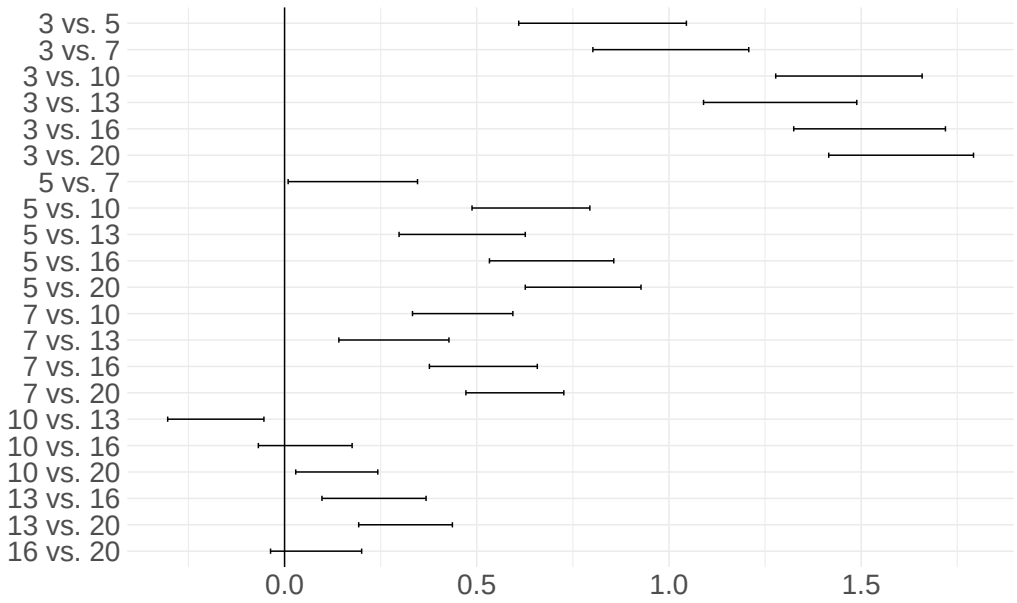


Abbildung 4.3.: Konfidenzintervalle des robusten Post-Hoc Tests.

Robuster Post-Hoc Test Dieser robuste Post-Hoc Test ist ebenso wie die robuste ANOVA aus dem „WRS2“-Paket von Mair und Wilcox [MW20]. Dabei wird hier ebenfalls wie bei der robusten ANOVA mit trimmed means gearbeitet. Wie bei Tukey's HSD werden auch hier Konfidenzintervalle berechnet, wobei ein Kreuzen der Null hier auch bedeutet, dass der Vergleich dieser Gruppen keinen signifikanten Unterschied gefunden hat. In Abbildung 4.3 ist dabei zu sehen dass alle außer zwei Vergleiche einen signifikanten Unterschied zwischen den jeweiligen Gruppen gefunden haben. Die genauen Werte geben wir auch hier wieder an, sie sind in Tabelle B.2 zu finden.

Bereits in Tabelle 4.1 und Abbildung 4.1b konnte man erkennen, dass zwischen den Gruppen Unterschiede existieren. Wir haben zuerst gezeigt, dass mindestens einer dieser Unterschiede signifikant ist und können nun die durchgeführten Post-Hoc Tests auswerten um zu zeigen, welche Unterschiede genau als signifikant gelten.

Alle drei Post-Hoc Tests finden einen signifikanten Unterschied zwischen der Erkennzeit bei drei Kugeln und jedem anderen Level. Ebenso besteht ein signifikanter Unterschied bei der Erkennzeit bei fünf Kugeln verglichen mit allen anderen Stufen. Dabei ist anzumerken, dass der Post-Hoc Test mittels trimmed means der Unterschied zwischen fünf und sieben Kugeln nur sehr knapp als signifikant einstuft. Indem diese als signifikant gefundenen Unterschiede untersucht werden, wie durch Abgleichen mit den in Tabelle 4.1 gegebenen Durchschnitten, können Aussagen über die Richtung des Unterschieds getroffen werden. Somit wird eine stetige, signifikante Verkleinerung der Erkennzeit von drei bis zehn Kugel gefunden. Die Ergebnisse von sowohl t-Tests mit Bonferroni-Korrektur, als auch Tukey's HSD werden generell als sehr konservativ angesehen[3] und lehnen signifikante Unterschiede eventuell fälschlicherweise ab. Unsere Hypothese **H1** besagt, dass die Erkennzeit mit zunehmender Kugelanzahl monoton fällt. Indem wir nur die Ergebnisse der paarweisen t-Tests und Tukey's HSD betrachten, scheint dies der Fall zu sein, da nur der Schritt von

[3] Kapitel 10.5.1 in *Discovering Statistics Using R* [And12]

Kugelanzahl	N	Fehleranzahl mean	SD
3	1469	0.15316542	1.1760953
5	1470	0.11904762	0.9727977
7	1486	0.04441454	0.6013837
10	1498	0.05206943	0.6309426
13	1484	0.06132075	0.6818225
16	1504	0.08377660	0.7872251
20	1455	0.06323024	0.7646698

Tabelle 4.4.: Zusammenfassung für Fehleranzahl pro Kugelanzahl.

SphereCount	W	p-Wert
3	0.10992	< 2.2e-16
5	0.098122	< 2.2e-16
7	0.04656	< 2.2e-16
10	0.055614	< 2.2e-16
13	0.062437	< 2.2e-16
16	0.08119	< 2.2e-16
20	0.055411	< 2.2e-16

Tabelle 4.5.: Shapiro-Wilk Testergebnisse zu Fehleranzahl pro Kugelanzahl.

Stufe vier (10 Kugeln) auf Stufe fünf (13 Kugeln) zu einer höheren durchschnittlichen Erkennzeit führt, dieser Unterschied jedoch nicht als signifikant eingestuft wird. Da unsere Daten die Annahmen der Normalverteilung und Varianzhomogenität jedoch verletzen, müssen wir die Ergebnisse des robusten Post-Hoc Tests in Betracht ziehen. Dieser findet für den Schritt von 10 zu 13 Kugeln einen signifikanten Unterschied, weswegen wir die Erkennzeit hier nicht als monoton fallend betrachten können. Somit können wir unsere Hypothese **H1**: „Je mehr Kugeln zum Hervorheben verwendet werden, desto schneller wird das hervorgehobene Objekt erkannt.“ nur eingeschränkt bestätigen.

4.2. Fehler

Hier werten wir die Anzahl der falsch angeklickten Äpfel, die „Fehleranzahl“ pro Kugelanzahl aus. Dabei verwenden wir die selben 10366 Datenpunkte wie zur Auswertung der Erkennzeit, eine Zusammenfassung mit der durchschnittlichen Fehleranzahl (mean) und Standardabweichung (SD) pro Kugelanzahl geben wir in Tabelle 4.4. Eine visuelle Übersicht über alle Datenpunkte zur Fehleranzahl geben wir in Abbildung 4.4. Die Anhäufung bei 8 Fehlern pro Runde liegt daran, dass wir nach so vielen Fehlern die Möglichkeit zu Abbruch der Runde angeboten haben.

Auch hier deutet eine visuelle Inspektion diese Messdaten darauf hin, dass keine Normalverteilung innerhalb der Gruppen vorliegt. Dies haben wir wieder durch einen Shapiro-Wilk Test pro Stufe der Kugelanzahl überprüft. Deren Ergebnisse sind in Tabelle 4.5 aufgelistet und zeigen, dass alle p-Werte unter dem Signifikanzniveau von 0.05 liegen. Somit lehnen wir die Nullhypothese der Normalität für alle Gruppen ab.

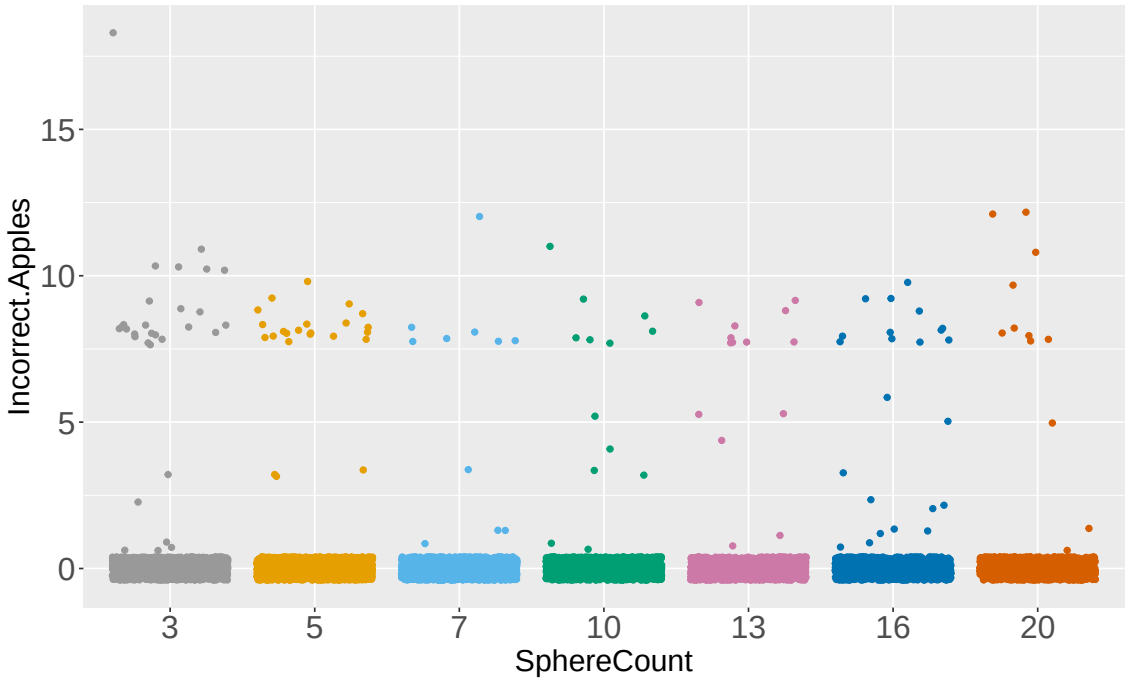


Abbildung 4.4.: Übersicht der Fehleranzahl mit Scatterplot.

Mittels Levene's Test haben wir die Varianzhomogenität untersucht und dabei einen signifikanten Unterschied gefunden, $F_{Levene}(6, 10359) = 3.453, p = 0.0021$. Deshalb lehnen wir auch hier die Nullhypothese der Homogenität der Varianzen zwischen dern Gruppen ab.

Eine durchgeführte ANOVA hat einen signifikanten Effekt der Kugelanzahl auf die Fehleranzahl gefunden, $F(6, 10359) = 3.453, p = 0.0021, \omega = 0.002075$, ebenso wie eine ANOVA mit Welch's F, $F_W(6, 4581) = 2.6276, p = 0.01519$, die wir aufgrund der Varianzheterogenität zwischen den Gruppen durchgeführt haben. Da unsere Daten auch die Annahmen der Normalverteilung verletzen haben wir ebenfalls den Kruskal-Wallis Test durchgeführt, welcher ebenfalls einen signifikanten Unterschied gefunden hat, $H(6) = 18.221, p = 0.0057$. Die robuste ANOVA lies sich nicht durchführen, ohne den Anteil der verworfen Datenpunkte je Ende von 0.2 auf 0.007 zu verringern, was dies somit wenig aussagekräftig macht, dennoch geben wir hier zur Vollständigkeit ihr Ergebnis an. Dabei wurde ebenfalls ein signifikanter Unterschied gefunden, $F_{TM}(6, 4313.18) = 2.8763, p = 0.0085, \omega = 0.05$.

Mehrere paarweise durchgeführte t-Tests mit Bonferroni-Korrektur haben nur zwischen 3 und 7 Kugeln, sowie zwischen 3 und 10 Kugeln einen signifikanten Unterschied gefunden, dies ist in Tabelle 4.6 zu sehen.

Der Tukey's HSD Test findet mehrere Intervalle die nicht die Null kreuzen und somit signifikante Unterschiede darstellen. Die Ergebnisse sind in Abbildung 4.5 dargestellt zu sehen, die exaten Werte sind in Tabelle B.3 angegeben.

	3	**5**	**7**	**10**	**13**	**16**
5	1.000	-	-	-	-	-
7	0.007	0.289	-	-	-	-
10	0.017	0.562	1.000	-	-	-
13	0.051	1.000	1.000	1.000	-	-
16	0.454	1.000	1.000	1.000	1.000	-
20	0.066	1.000	1.000	1.000	1.000	1.000

Tabelle 4.6.: Paarweiser t-Test mit Bonferroni-Korrektur für Fehleranzahl.

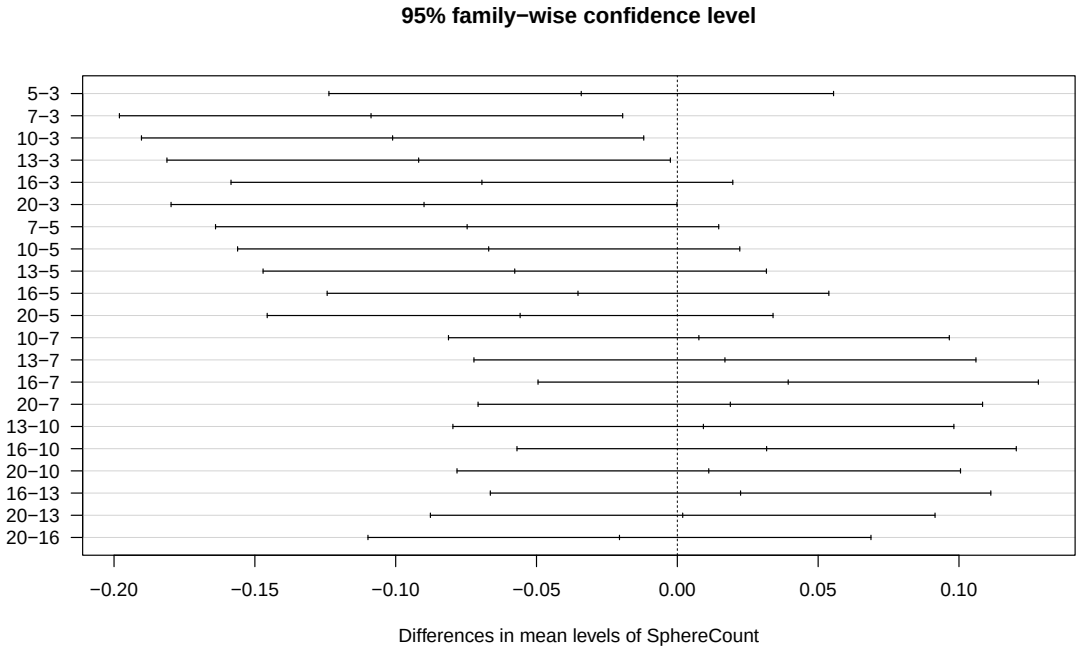


Abbildung 4.5.: Tukey's HSD für Fehleranzahl pro Kugelanzahl.

Ein robuster Post-Hoc Test, ebenfalls mit einem Anteil von 0.007 statt 0.2 verworfenen Daten pro Ende, hat einzig den Vergleich von 3 zu 7 Kugeln als signifikant unterschiedlich gefunden. Eine Abbildung die die Ergebnisse zeigt ist in Abbildung 4.6, eine Tabelle mit den exakten Ergebnissen ist in Tabelle B.4

Somit finden alle unsere drei durchgeführten Post-Hoc Tests einen signifikanten Unterschied der Fehleranzahl pro Kugelanzahl zwischen 3 und 7 Kugeln. Wie aus Tabelle 4.4 ersichtlich, ist dies jedoch eine Verringerung der durchschnittlichen Fehleranzahl, was unserer Hypothese **H2**: „Je mehr Kugeln zum Hervorheben verwendet werden, desto mehr Fehler passieren beim Anklicken.“ gänzlich widerspricht.

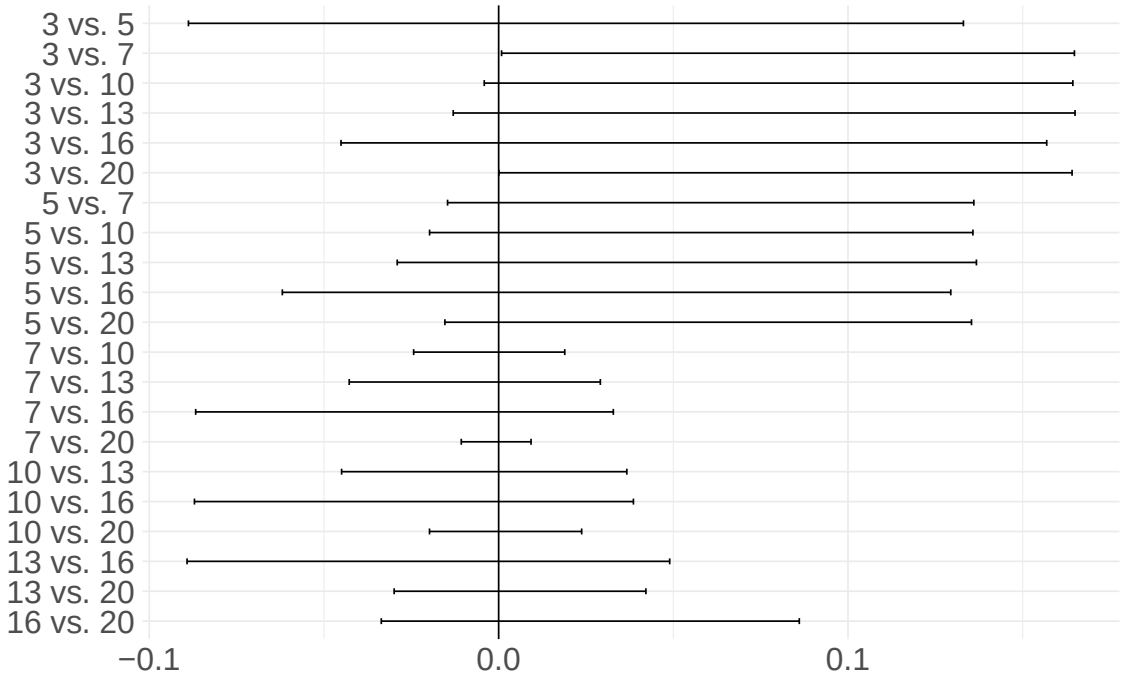


Abbildung 4.6.: Robuster Post-Hoc Test für Fehleranzahl pro Kugelanzahl.

5. Zusammenfassung und Ausblick

5.1. Zusammenfassung

In dieser Bachelorarbeit haben wir untersucht welchen Einfluss die Anzahl der verwendeten Kugeln auf die Hervorhebungsmethode HiveFive [LSGB20] hat. Dafür haben wir in Unity eine Möglichkeit die möglichen Effekte zu messen implementiert. Der Studienteil unserer Implementierung bestand aus einer Szene im dreidimensionalen Raum, mit fünf auf einem Gelände verteilten Bäumen an denen mehrere Äpfel hängen. Diese Studie haben wir mit Hilfe der Crowdsourcing-Platform Prolific durchgeführt. Unsere Auswertung ergab, dass nur von eine Erhöhung der Kugelanzahl von unter 10 auf 10 zu einer signifikant schnelleren Erkennzeit führt. Änderungen von 10 auf über 10 Kugeln führt zu einer signifikanten Verlangsamung der Erkennzeit (13 Kugeln), oder es ist keine signifikante Änderung erkennbar (16, 20 Kugeln).

Anders als vermutet wird die Fehleranzahl nicht signifikant größer durch Erhöhen der Kugelanzahl. Einzig bei bei der Erhöhung der Kugelanzahl von 3 auf 7 Kugeln ist ein signifikanter Unterschied gegeben. Hierbei fällt entgegen unserer Hypothese die Fehleranzahl.

5.2. Limitationen

Da wir unsere Studie per Crowdsourcing durchgeführt haben, hatten wir keinen Einfluss auf die Einhaltung der Anweisungen durch die Teilnehmer. So konnten wir nicht versichern, dass Teilnehmer die Studie wie angewiesen im Vollbildmodus durchführen oder, wie vor jeder Runde angemerkt, dass das Erkennen der Hervorhebung mit Drücken von „X“ registriert wird.

5.3. Ausblick

Da HiveFive originell für VR entwickelt wurde, ist es eventuell sinnvoll unsere Ergebnisse auch in VR zu überprüfen. Da wir außerdem die Erkennzeit nur durch eine teilweise nicht vorhandene Nutzereingabe, die durch biologische Gegebenheiten weiteren Verzögerungen unterliegt, messen konnten, ist ein Studienaufbau mit Eye-Tracking eine weitere Idee um genauere Ergebnisse zu erzielen.

Literaturverzeichnis

[ALI24] ALIyerEdon. *URP Tree Models*. 2024. URL: https://assetstore.unity.com/packages/3d/vegetation/trees/urp-tree-models-253340 (zitiert auf S. 17).

[And12] Z. F. Andy Field Jeremy Miles. *Discovering Statistics Using R*. März 2012. URL: https://studysites.uk.sagepub.com/dsur/main.htm (zitiert auf S. 29, 31).

[Ish08a] S. Ishihara. *Ishihara Plate 1 (12)*. Hrsg. von Nicoguaro. 15. Mai 2008. URL: https://commons.wikimedia.org/wiki/File:Ishihara_1.PNG (zitiert auf S. 14).

[Ish08b] S. Ishihara. *Ishihara Plate 11 (6)*. Hrsg. von Nicoguaro. 15. Mai 2008. URL: https://commons.wikimedia.org/wiki/File:Ishihara_11.PNG (zitiert auf S. 14).

[Ish08c] S. Ishihara. *Ishihara Plate 19 (Empty)*. Hrsg. von Nicoguaro. 15. Mai 2008. URL: https://commons.wikimedia.org/wiki/File:Ishihara_19.PNG (zitiert auf S. 14).

[Ish08d] S. Ishihara. *Ishihara Plate 23 (42)*. Hrsg. von Nicoguaro. 15. Mai 2008. URL: https://commons.wikimedia.org/wiki/File:Ishihara_23.PNG (zitiert auf S. 14).

[Ish08e] S. Ishihara. *Ishihara Plate 9 (74)*. Hrsg. von Nicoguaro. 15. Mai 2008. URL: https://commons.wikimedia.org/wiki/File:Ishihara_9.png (zitiert auf S. 14).

[Ish85] S. Ishihara. *Ishihara's test for colour-blindness*. Kanehara Shuppan Company, 1985 (zitiert auf S. 14).

[KW52] W. H. Kruskal, W. A. Wallis. „Use of Ranks in One-Criterion Variance Analysis". In: *Journal of the American Statistical Association* 47.260 (1952), S. 583–621. DOI: 10.1080/01621459.1952.10483441. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1952.10483441. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441 (zitiert auf S. 29).

[LSGB20] D. Lange, T. C. Stratmann, U. Gruenefeld, S. Boll. „HiveFive: Immersion Preserving Attention Guidance in Virtual Reality". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. , Honolulu, HI, USA, Association for Computing Machinery, 2020, S. 1–13. ISBN: 9781450367080. DOI: 10.1145/3313831.3376803 (zitiert auf S. 11, 17, 23, 37).

[LSGB22] D. Lange, T. C. Stratmann, U. Gruenefeld, S. Boll. *Research Repository for HiveFive*. 2022. URL: https://github.com/danllng/hivefive (besucht am 23. 07. 2024) (zitiert auf S. 18, 19).

[MW20] P. Mair, R. Wilcox. „Robust statistical methods in R using the WRS2 package". In: *Behavior Research Methods* 52.2 (2020), S. 464–488. ISSN: 1554-3528. DOI: 10.3758/s13428-019-01246-w. URL: https://doi.org/10.3758/s13428-019-01246-w (zitiert auf S. 29, 31).

[nik10] nikita. *Coastal Overlay [World Machine Marco]*. 2010. URL: https://www.world-machine.com/library/index.php?entry=47&type=0&sort=most&focus=1 (zitiert auf S. 17).

[NMH+16] L. T. Nielsen, M. B. Møller, S. D. Hartmeyer, T. C. M. Ljung, N. C. Nilsson, R. Nordahl, S. Serafin. „Missing the point: an exploration of how to guide users' attention during cinematic virtual reality". In: *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. VRST '16. Munich, Germany: Association for Computing Machinery, 2016, S. 229–232. ISBN: 9781450344913. DOI: 10.1145/2993369.2993405. URL: https://doi.org/10.1145/2993369.2993405 (zitiert auf S. 11).

[Pal24] Pallets. *Flask (Version 3.0.3) [Computer software]*. 2024. URL: https://palletsprojects.com/ (zitiert auf S. 20).

[Pos24] Posit team. *RStudio: Integrated Development Environment for R*. Posit Software, PBC. Boston, MA, 2024. URL: http://www.posit.co/ (zitiert auf S. 27).

[R C24] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2024. URL: https://www.R-project.org/ (zitiert auf S. 27).

[Rey87] C. W. Reynolds. „Flocks, herds and schools: A distributed behavioral model". In: *SIGGRAPH Comput. Graph.* 21.4 (Aug. 1987), S. 25–34. ISSN: 0097-8930. DOI: 10.1145/37402.37406 (zitiert auf S. 18, 19).

[Row17] Rowntec Assets. *Survival Kit Lite*. 2017. URL: https://assetstore.unity.com/packages/3d/props/tools/survival-kit-lite-92549 (zitiert auf S. 17).

[WB24] J. W. Woodworth, C. W. Borst. „Visual cues in VR for guiding attention vs. restoring attention after a short distraction". In: *Computers & Graphics* 118 (2024), S. 194–209. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2023.12.008. URL: https://www.sciencedirect.com/science/article/pii/S0097849323003023 (zitiert auf S. 11).

[Wel51] B. L. Welch. „On the Comparison of Several Mean Values: An Alternative Approach". In: *Biometrika* 38.3/4 (1951), S. 330–336. ISSN: 00063444, 14643510. URL: http://www.jstor.org/stable/2332579 (besucht am 31. 07. 2024) (zitiert auf S. 29).

[Wic16] H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4. URL: https://ggplot2.tidyverse.org (zitiert auf S. 27).

[Wor22] World Machine Software, LLC. *World Machine*. 2022. URL: https://www.world-machine.com/ (zitiert auf S. 17).

Alle URLs wurden zuletzt am 31. 07. 2024 geprüft.

A. Ishihara-Ergebnisse

Hier geben wir die Verteilung aller 348 Antworten pro Tafel an. Wie erwähnt wurden die Antworten zweier Teilnehmer nicht hochgeladen und somit nicht betrachtet. Für die Ergebnisse von Tafel 19 in Abbildung A.4 gilt, dass diese Tafel original kein sichtbares Muster haben soll. In dieser Version sehen jedoch einige Menschen ohne Farbsehschwäche eine Zwei oder Acht.

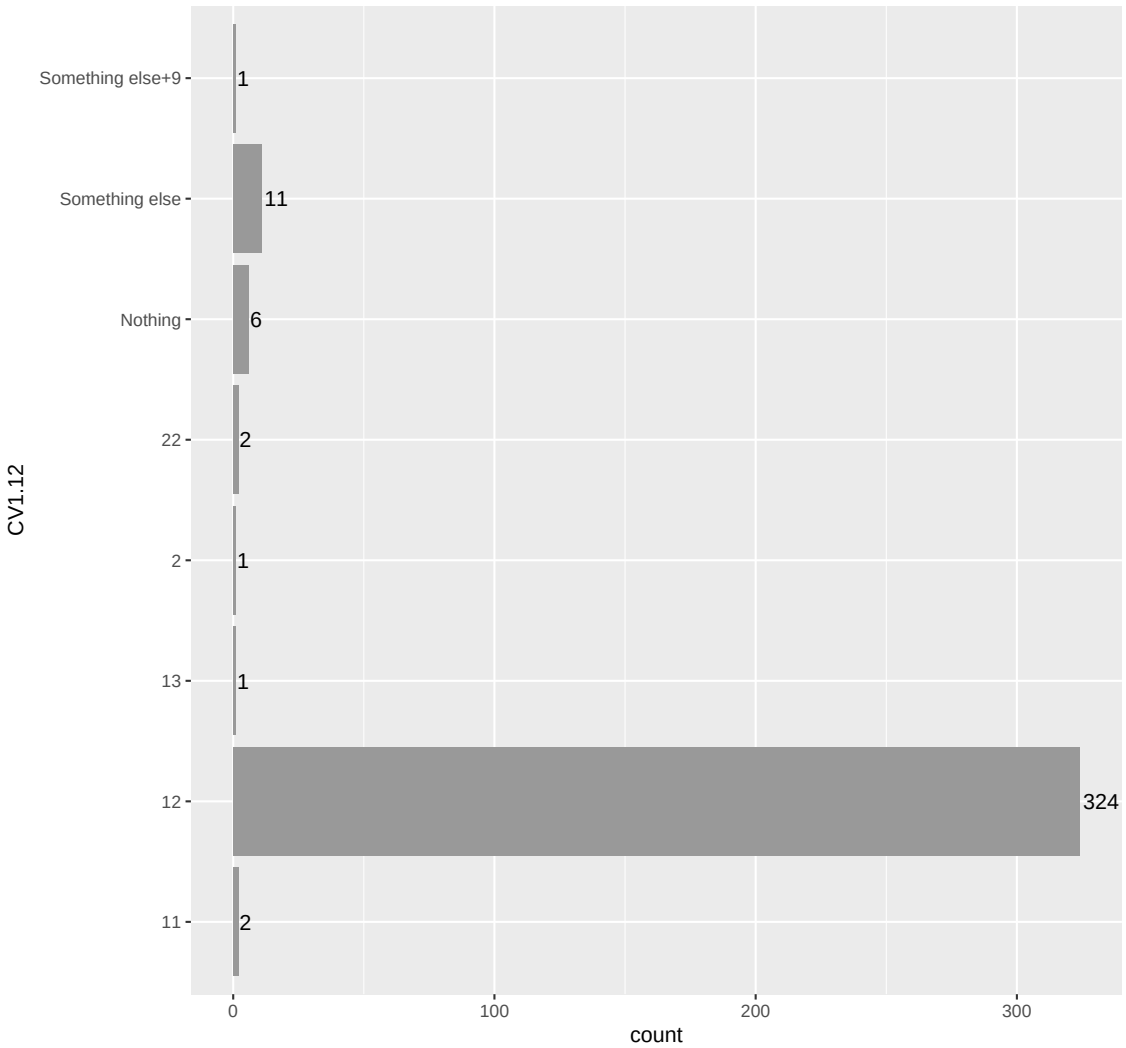


Abbildung A.1.: Ishihara Tafel 1 (12)

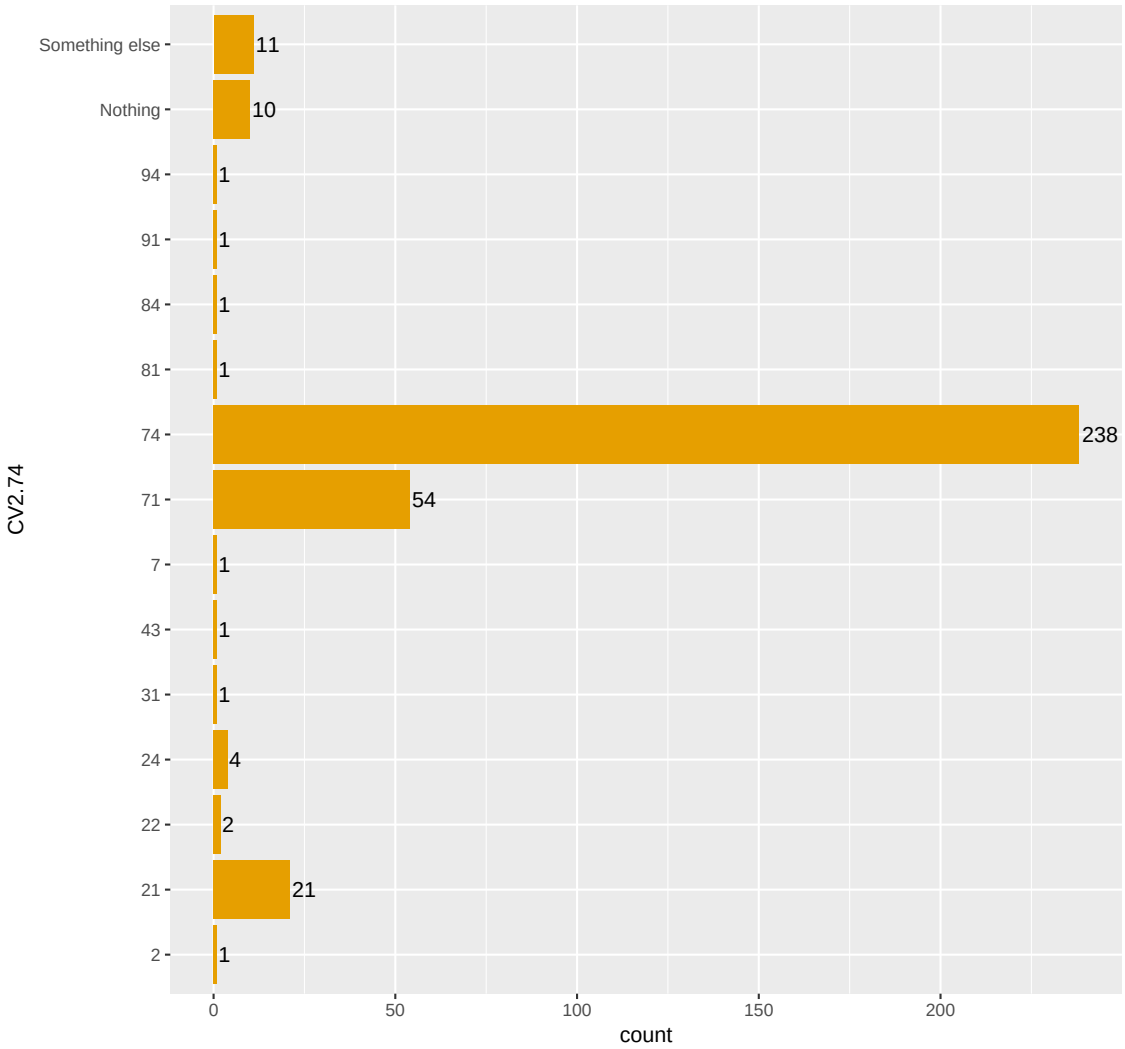


Abbildung A.2.: Ishihara Tafel 9 (74)

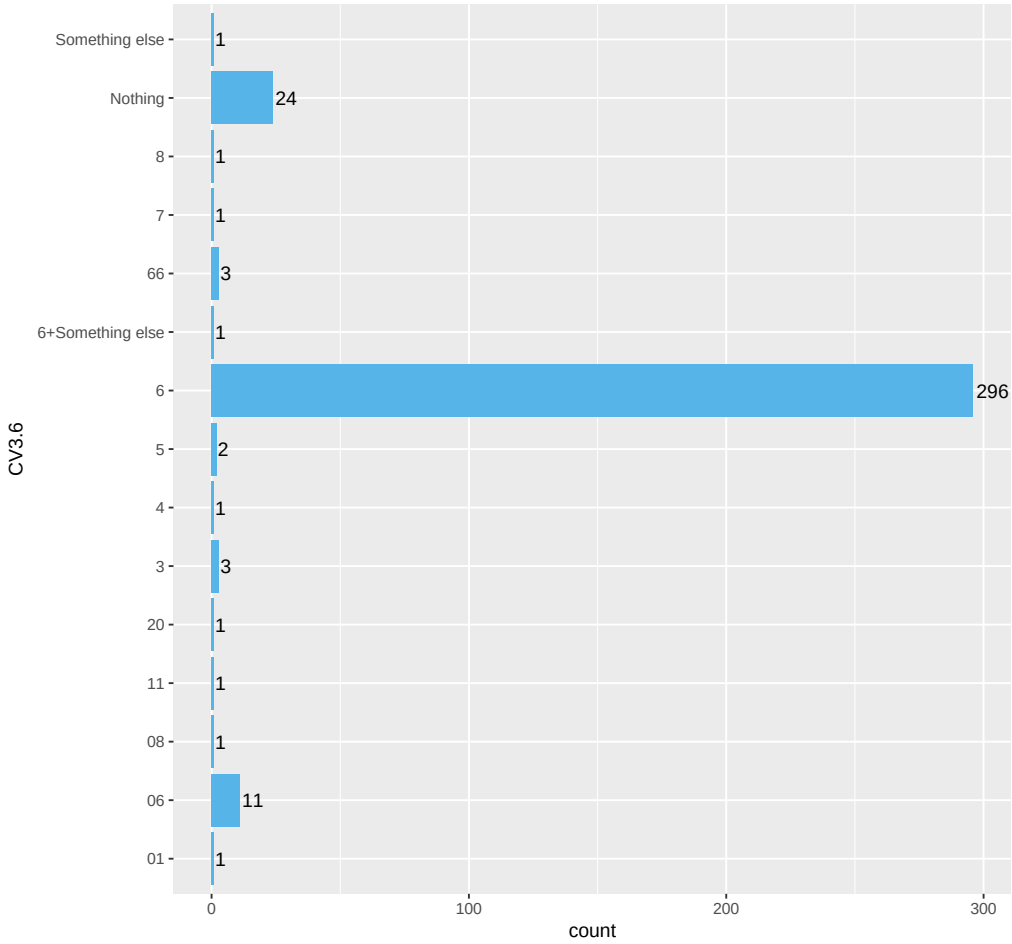


Abbildung A.3.: Ishihara Tafel 11 (6)

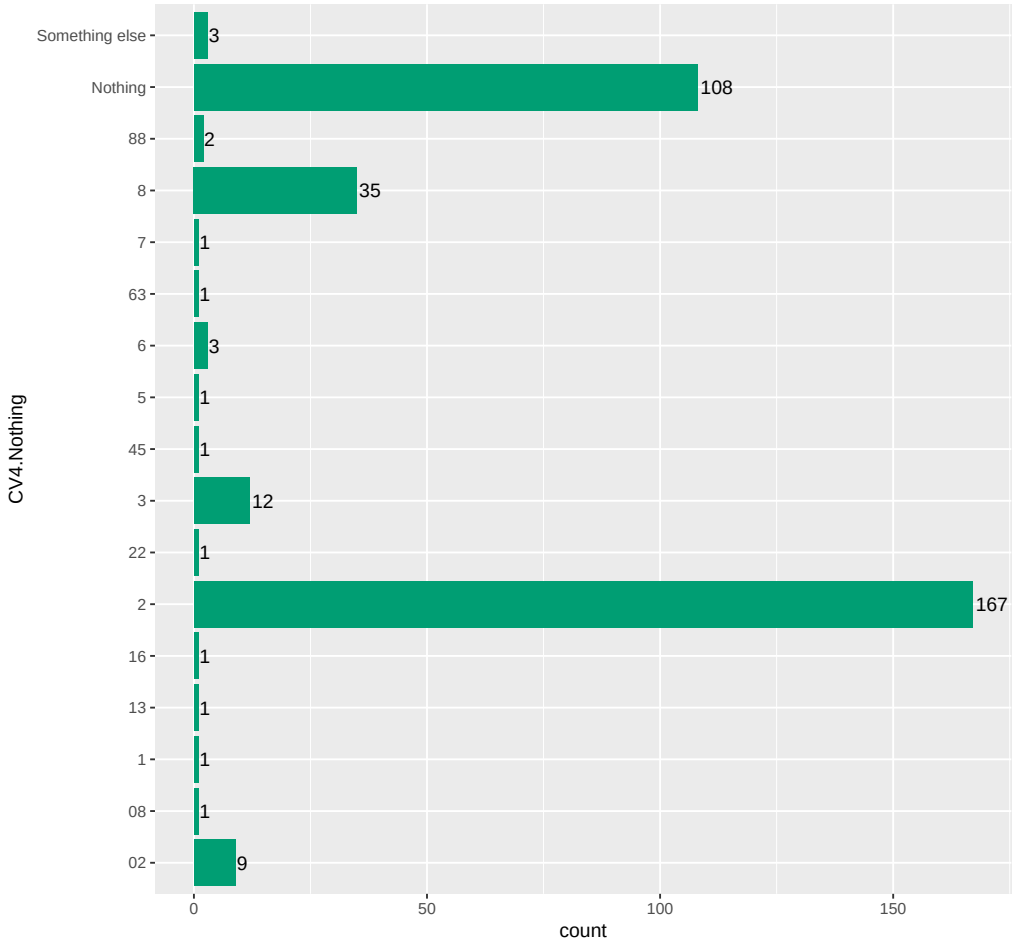


Abbildung A.4.: Ishihara Tafel 19 (Leer/2/8)

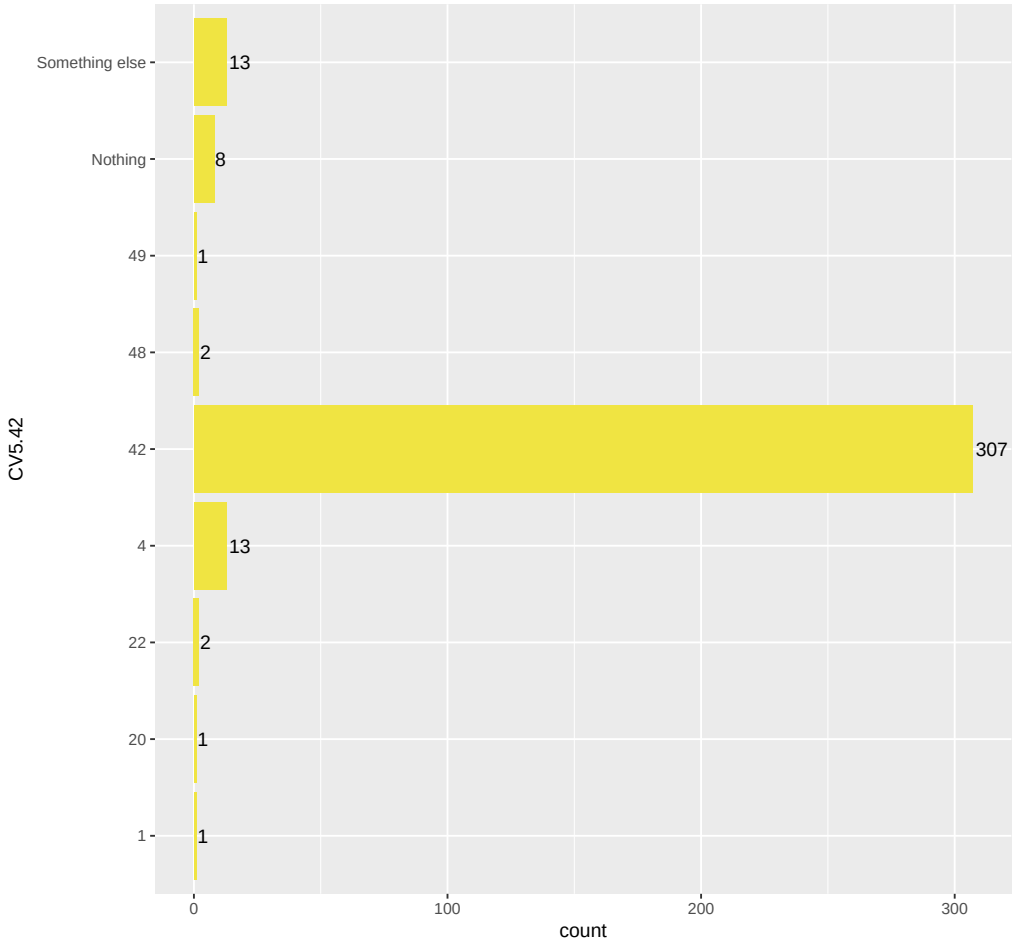
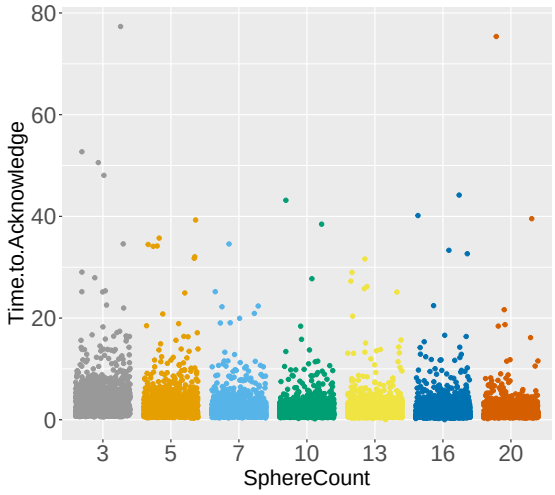


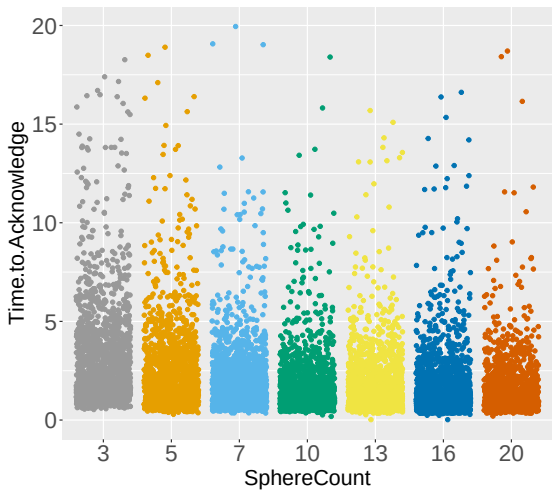
Abbildung A.5.: Ishihara Tafel 23 (42)

B. Weitere Daten der Auswertung

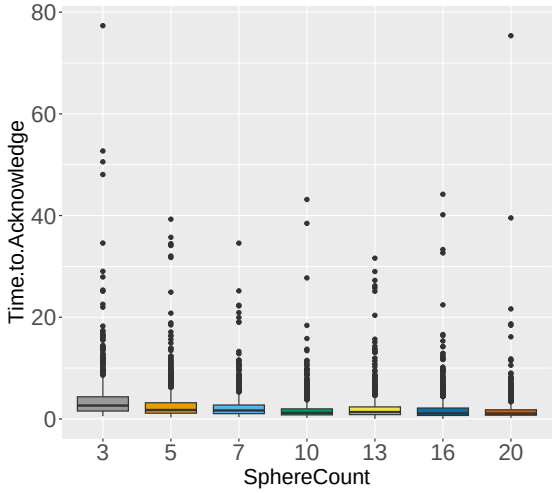
B.1. Erkennzeit



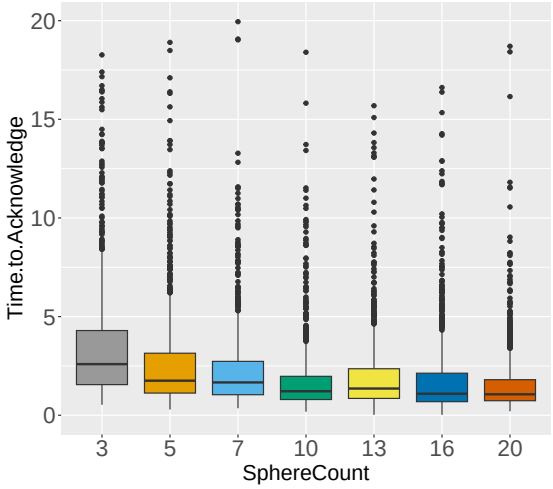
(a) Alle validen Datenpunkte.



(b) Nur Erkennzeit unter 20 s.



(c) Alle validen Datenpunkte.



(d) Nur Erkennzeit unter 20 s.

Abbildung B.1.: Übersicht über alle Datenpunkte mit einer Erkennzeit über null, mittels Scatterplots und Boxplots.

	diff	lwr	upr	p adj
5-3	-0.9662879	-1.28219735	-0.65037844	0.0000000
7-3	-1.3653438	-1.68040208	-1.05028560	0.0000000
10-3	-1.8529565	-2.16738675	-1.53852619	0.0000000
13-3	-1.6334950	-1.94865877	-1.31833126	0.0000000
16-3	-1.7749155	-2.08903507	-1.46079587	0.0000000
20-3	-1.9987615	-2.31548386	-1.68203919	0.0000000
7-5	-0.3990559	-0.71406028	-0.08405160	0.0035560
10-5	-0.8866686	-1.20104485	-0.57229229	0.0000000
13-5	-0.6672071	-0.98231700	-0.35209723	0.0000000
16-5	-0.8086276	-1.12269311	-0.49456202	0.0000000
20-5	-1.0324736	-1.34914235	-0.71580490	0.0000000
10-7	-0.4876126	-0.80113353	-0.17409173	0.0000931
13-7	-0.2681512	-0.58240767	0.04610532	0.1534133
16-7	-0.4095716	-0.72278094	-0.09636231	0.0022266
20-7	-0.6334177	-0.94923723	-0.31759814	0.0000001
13-10	0.2194615	-0.09416549	0.53308839	0.3747202
16-10	0.0780410	-0.23453664	0.39061865	0.9903431
20-10	-0.1458051	-0.46099817	0.16938806	0.8209436
16-13	-0.1414205	-0.45473591	0.17189501	0.8374824
20-13	-0.3652665	-0.68119133	-0.04934169	0.0116392
20-16	-0.2238461	-0.53872924	0.09103712	0.3547887

Tabelle B.1.: Tukey's HSD mit Konfidenzintervallen. Signifikante Unterschiede sind markiert.

	psihat	ci.lower	ci.upper	p.value
3 vs. 5	0.82722	0.60914	1.04530	0.00000
3 vs. 7	1.00478	0.80202	1.20754	0.00000
3 vs. 10	1.46809	1.27761	1.65857	0.00000
3 vs. 13	1.28919	1.08983	1.48855	0.00000
3 vs. 16	1.52185	1.32451	1.71919	0.00000
3 vs. 20	1.60386	1.41556	1.79217	0.00000
5 vs. 7	0.17756	0.00928	0.34584	0.00424
5 vs. 10	0.64087	0.48761	0.79414	0.00000
5 vs. 13	0.46197	0.29781	0.62614	0.00000
5 vs. 16	0.69463	0.53293	0.85634	0.00000
5 vs. 20	0.77665	0.62610	0.92720	0.00000
7 vs. 10	0.46331	0.33276	0.59386	0.00000
7 vs. 13	0.28441	0.14122	0.42760	0.00000
7 vs. 16	0.51707	0.37671	0.65744	0.00000
7 vs. 20	0.59909	0.47174	0.72643	0.00000
10 vs. 13	-0.17890	-0.30410	-0.05370	0.00008
10 vs. 16	0.05376	-0.06819	0.17572	0.18182
10 vs. 20	0.13578	0.02906	0.24250	0.00048
13 vs. 16	0.23266	0.09726	0.36806	0.00000
13 vs. 20	0.31468	0.19282	0.43653	0.00000
16 vs. 20	0.08201	-0.03651	0.20054	0.07235

Tabelle B.2.: Robuster Post-Hoc Test mit Konfidenzintervallen. Signifikante Unterschiede sind markiert.

B.2. Fehleranzahl

	diff	lwr	upr	p adj
5-3	-0.034117800	-0.12370527	0.0554696697	0.9212429
7-3	-0.108750883	-0.19809696	-0.0194048075	0.0061458
10-3	-0.101095993	-0.19026399	-0.0119279961	0.0145769
13-3	-0.091844664	-0.18122066	-0.0024686636	0.0393863
16-3	-0.069388823	-0.15846871	0.0196910686	0.2453474
20-3	-0.089935178	-0.17975317	-0.0001171884	0.0494490
7-5	-0.074633083	-0.16396388	0.0146977084	0.1727606
10-5	-0.066978193	-0.15613088	0.0221744893	0.2870908
13-5	-0.057726864	-0.14708759	0.0316338575	0.4769356
16-5	-0.035271023	-0.12433559	0.0537935389	0.9062385
20-5	-0.055817378	-0.14562016	0.0339854079	0.5256169
10-7	0.007654890	-0.08125522	0.0965649984	0.9999781
13-7	0.016906219	-0.07221249	0.1060249329	0.9978557
16-7	0.039362060	-0.04945969	0.1281838073	0.8490470
20-7	0.018815705	-0.07074627	0.1083776777	0.9962168
13-10	0.009251329	-0.07968885	0.0981915086	0.9999331
16-10	0.031707170	-0.05693545	0.1203497849	0.9409543
20-10	0.011160815	-0.07822351	0.1005451388	0.9998049
16-13	0.022455841	-0.06639601	0.1113076898	0.9896956
20-13	0.001909486	-0.08768234	0.0915013115	1.0000000
20-16	-0.020546355	-0.10984279	0.0687500773	0.9937756

Tabelle B.3.: Tukey's HSD für Fehleranzahl pro Kugelanzahl. Signifikante Unterschiede sind markiert.

	psihat	ci.lower	ci.upper	p.value
3 vs. 5	0.02213	-0.08879	0.13304	0.82911
3 vs. 7	0.08282	0.00084	0.16481	0.04718
3 vs. 10	0.08012	-0.00412	0.16436	0.07609
3 vs. 13	0.07599	-0.01300	0.16499	0.17541
3 vs. 16	0.05588	-0.04512	0.15688	0.82911
3 vs. 20	0.08211	0.00009	0.16414	0.04924
5 vs. 7	0.06070	-0.01462	0.13602	0.25044
5 vs. 10	0.05800	-0.01977	0.13576	0.35952
5 vs. 13	0.05387	-0.02903	0.13676	0.68733
5 vs. 16	0.03375	-0.06192	0.12942	0.82911
5 vs. 20	0.05999	-0.01538	0.13535	0.25593
7 vs. 10	-0.00270	-0.02433	0.01893	0.82911
7 vs. 13	-0.00683	-0.04277	0.02911	0.82911
7 vs. 16	-0.02695	-0.08672	0.03283	0.82911
7 vs. 20	-0.00071	-0.01070	0.00928	0.82911
10 vs. 13	-0.00413	-0.04495	0.03669	0.82911
10 vs. 16	-0.02425	-0.08707	0.03858	0.82911
10 vs. 20	0.00199	-0.01979	0.02377	0.82911
13 vs. 16	-0.02011	-0.08919	0.04896	0.82911
13 vs. 20	0.00612	-0.02991	0.04215	0.82911
16 vs. 20	0.02623	-0.03359	0.08606	0.82911

Tabelle B.4.: Robuster Post-Hoc Test für Fehleranzahl pro Kugelanzahl. Signifikante Unterschiede sind markiert.

B.3. Likertfragebogen

Zur Vollständigkeit geben wir hier noch die Antworten auf unseren Likertfragebogen an, dabei gilt 1 als „Stimme überhaupt nicht zu“ und 5 als „Stimmt vollständig zu“

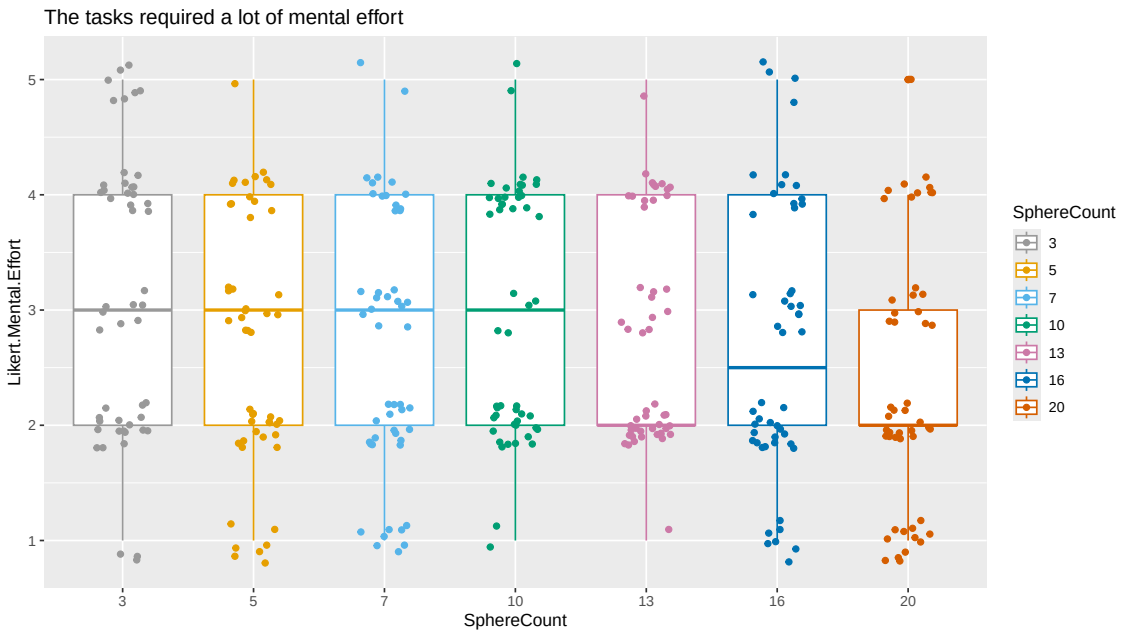


Abbildung B.2.: Antwortsverteilung auf Likertfrage 1.

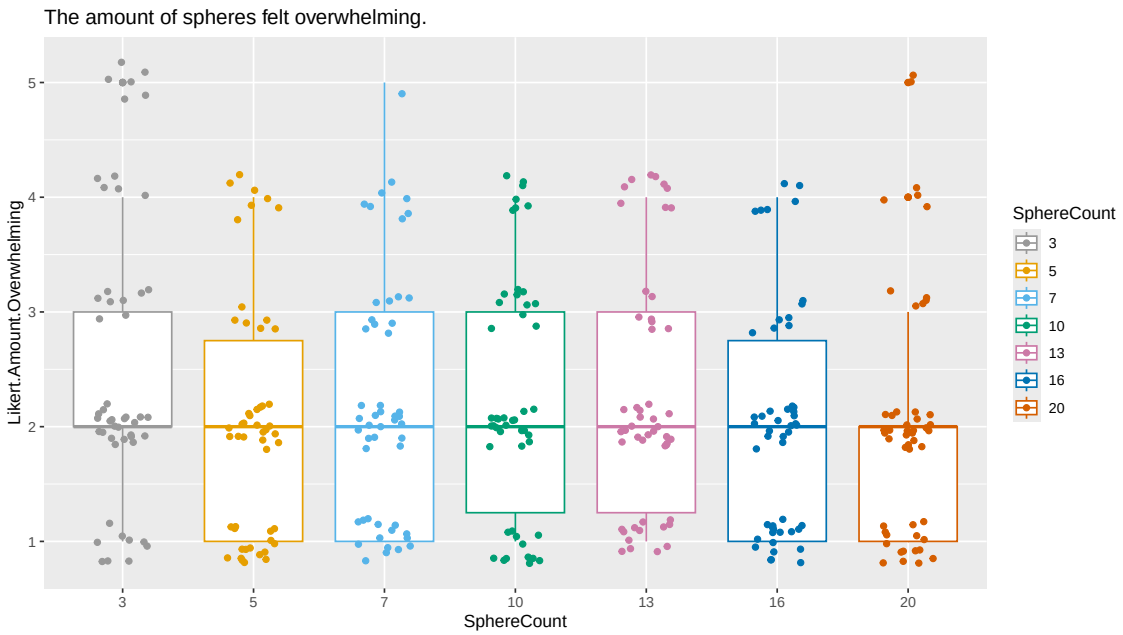


Abbildung B.3.: Antwortsverteilung auf Likertfrage 2.

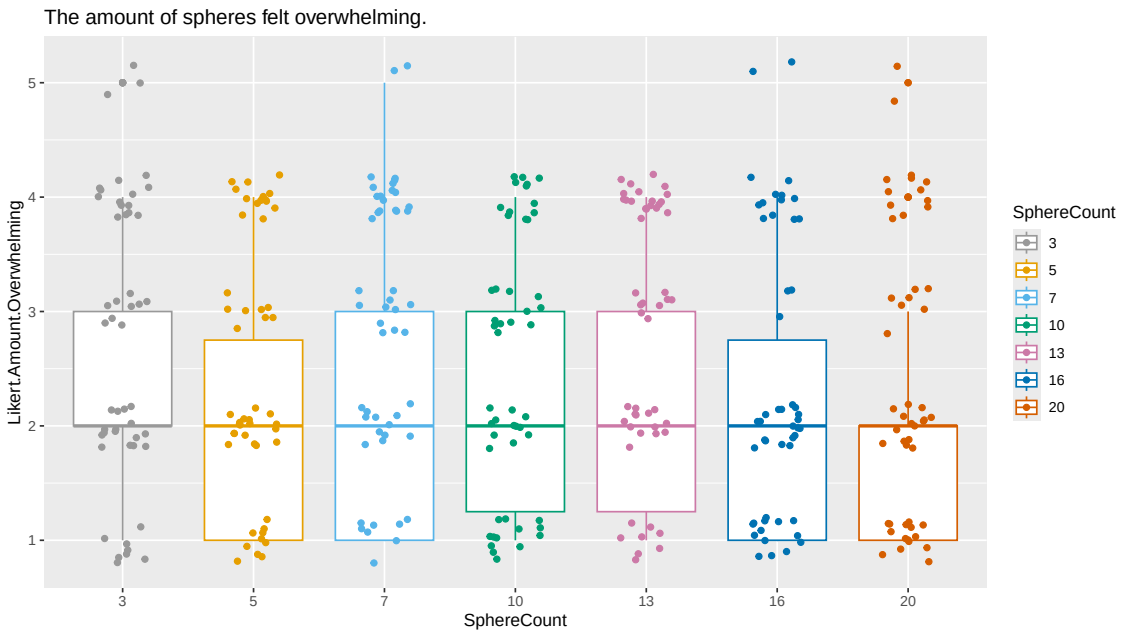


Abbildung B.4.: Antwortsverteilung auf Likertfrage 3.

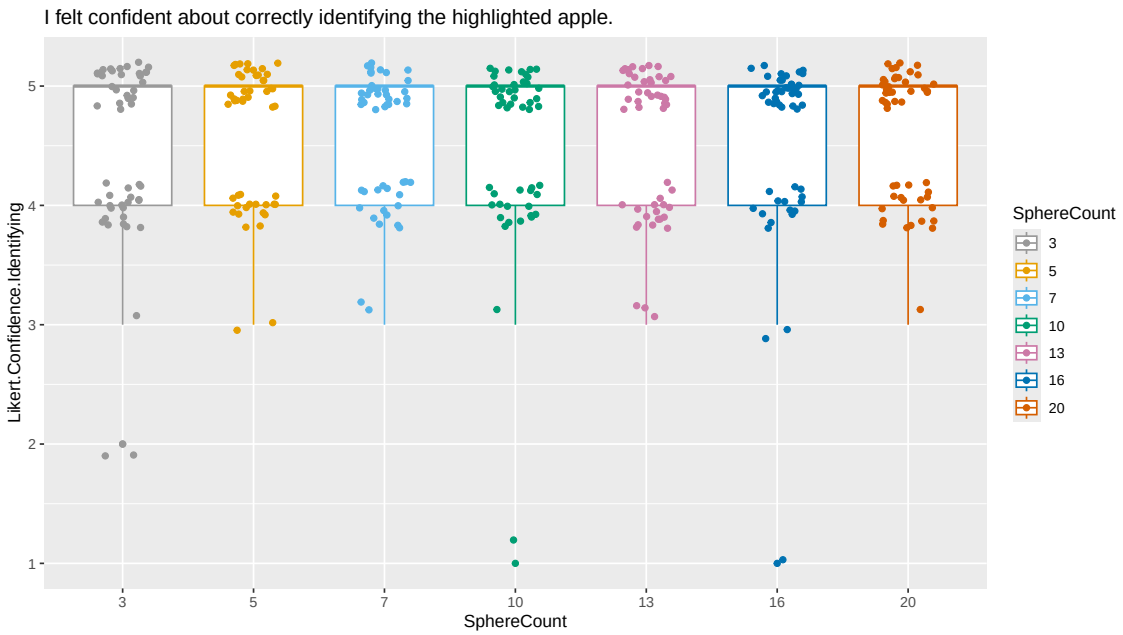


Abbildung B.5.: Antwortsverteilung auf Likertfrage 4.

Abbildung B.6.: Antwortsverteilung auf Likertfrage 5.

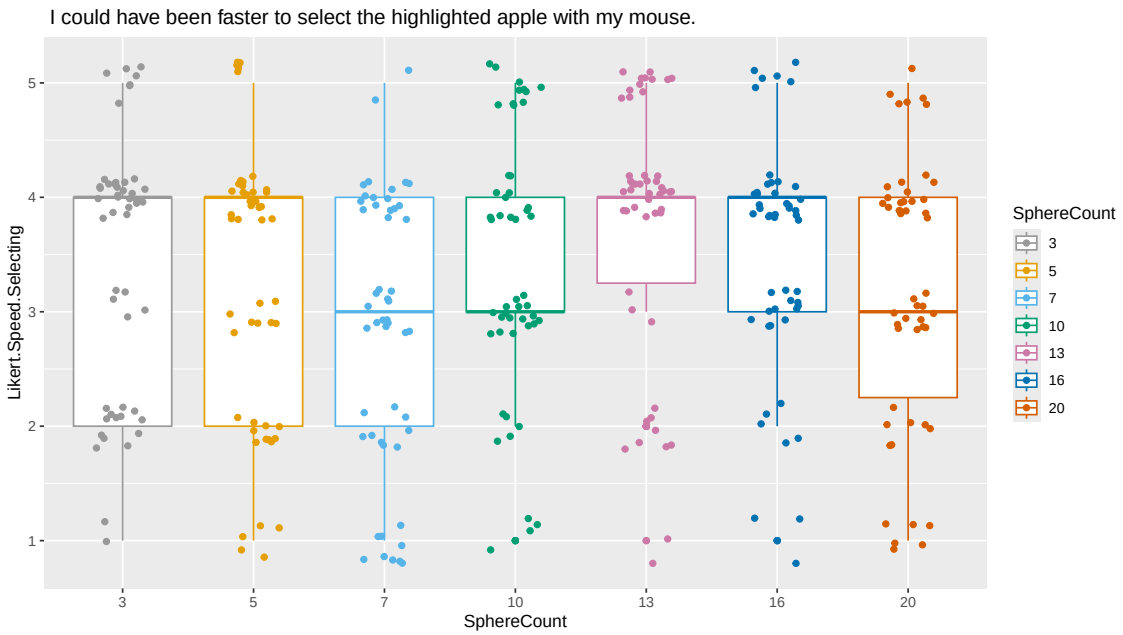


Abbildung B.7.: Antwortsverteilung auf Likertfrage 6.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Druck-Exemplaren überein.

Ort, Datum, Unterschrift