



**HOCHSCHULE
MITTWEIDA**
University of Applied Sciences

BACHELORARBEIT

Herr
Robin Schöppach

**Auswirkungen verschiedener
nicht-diegetischer
Hervorhebungsmethoden auf die
visuelle Objektwahrnehmung in
Videospielen**

Mittweida, Juni 2025



BACHELORARBEIT

Auswirkungen verschiedener nicht-diegetischer Hervorhebungsmethoden auf die visuelle Objektwahrnehmung in Videospielen

Autor:

Robin Schöppach

Studiengang:

Medieninformatik & Interaktives Entertainment

Seminargruppe:

MI20w3-B

Erstprüfer:

Prof. Dr.-Ing. Christian Roschke

Zweitprüfer:

M.Sc. Manuel Heintz

Einreichung:

Mittweida, 19.06.2025

BACHELOR THESIS

Effects of different non-diegetic highlighting methods on visual object perception in video games

Author:

Robin Schöppach

Course of Study:

Media Informatics & Interactive Entertainment

Seminar Group:

MI20w3-B

First Examiner:

Prof. Dr.-Ing. Christian Roschke

Second Examiner:

M.Sc. Manuel Heintzig

Submission:

Mittweida, 19.06.2025

Bibliografische Beschreibung

Schöppach, Robin:

Auswirkungen verschiedener nicht-diegetischer Hervorhebungsmethoden auf die visuelle Objektwahrnehmung in Videospielen. – 2025. – 48 S.

Mittweida, Hochschule Mittweida – University of Applied Sciences, Fakultät Angewandte Computer & Biowissenschaften, Bachelorarbeit, 2025.

Referat

Diese Arbeit behandelt den Einfluss von nicht-diegetischen Hervorhebungsmethoden auf die visuelle Wahrnehmung von Objekten in Videospielen. Hierfür wurden acht Videospiele untersucht und ihre nicht-diegetischen Hervorhebungsmethoden extrahiert. Es wurden vier verschiedene nicht-diegetische Hervorhebungstechniken erkannt, die im Anschluss verglichen werden sollten: die *Outline* (eine farbige Umrandung bzw. Kontur), der *Glow-Effekt* (ein weicher Schein bzw. Leuchten um das hervorzuhebende Objekt), die Hervorhebung mittels des *Fresnel-Effekts* und der *Color-Overlay-Effekt* (Einfärbung bzw. Farbüberlagerung des hervorzuhebenden Objektes).

Zum Zwecke des Vergleichs wurde eine Software entworfen und mit der Spiele-Engine Unity umgesetzt, in der die Methoden in einem videospielnahen Kontext verglichen werden können. Zusätzlich zu den zusammengetragenen Hervorhebungstechniken wurde eine weitere, texturierte Variante der Farbüberlagerung implementiert – die Hervorhebung mittels des sog. *Stripe-Effekts*, eine Variante des Color-Overlay-Effekts mit Streifentextur – um den Einfluss von Texturen auf die Wahrnehmung der Hervorhebungsmethoden zu erfassen. Mit der entstandenen Software wurde eine Studie mit $n = 10$ Probanden im Alter 21-25 durchgeführt, in denen den Teilnehmenden eine Anzahl von Objekten für einen begrenzten Zeitraum von zwei Sekunden gezeigt wurde. Dabei sollten sowohl die Objekte als auch die Umgebung den Kontext eines Videospieles möglichst getreu nachempfinden. Die Teilnehmenden wurden nach Verschwinden der Objekte gefragt, wie viele Objekte zu sehen waren. Dieser Versuchsablauf wurde 50 Mal pro Proband wiederholt. Anschließend sollten die Teilnehmenden einen Fragebogen ausfüllen, der die Einschätzung der Erkennbarkeit der Methoden und demographische Daten erfasste. Demnach wurden die Methoden Outline und Glow-Effekt als am besten erkennbar gewertet. Leicht schlechter erkennbar wurde die Hervorhebung durch den Fresnel-Effekt eingeschätzt. Die Hervorhebung mit Stripe-Effekt und vor allem die Hervorhebung durch die Farbüberlagerung wurden als am schlechtesten erkennbar erachtet. Unterschiede in den relativen Abweichungen der Schätzversuche der Probanden von den angezeigten Objektzahlen zwischen den Hervorhebungsmethoden waren statistisch nicht signifikant. Auch ließ sich kein Zusammenhang zwischen der Genauigkeit der Schätzversuche und der wöchentlichen Spielzeit der Probanden hergestellt werden.

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Ergänzende Verzeichnisse	III
1 Einführung und Motivation	1
1.1 Aufgabenbeschreibung	1
1.2 Aufbau der Arbeit	2
2 Grundlagen	3
2.1 Visuelle Wahrnehmung	3
2.2 Diegetische und nicht-diegetische Hervorhebungsmethoden	5
2.3 Technische Grundlagen	7
2.4 Hervorhebungsmethoden im Vergleich	8
3 Konzeption	11
3.1 Szenenaufbau	11
3.2 Wahrzunehmende Objekte	11
3.3 Objekt-Platzierung	12
3.4 Auswahl der nicht-diegetischen Hervorhebungsmethoden	12
4 Implementierung	21
4.1 System	21
4.2 Objekterstellung	22
4.3 Hervorhebungsmethoden	25
4.4 Logging	33
5 Evaluation	35
5.1 Teilnehmende	35
5.2 Beschreibung des Experiments	35
5.3 Ergebnisse	35
5.4 Fehlerquellen	43
5.5 Fazit	44
6 Zusammenfassung und Ausblick	47
6.1 Zusammenfassung	47
6.2 Weiterführende Arbeit	47
Anhang	49
A Fragebogen	49
Glossar	51
Literaturverzeichnis	53
Eidesstattliche Erklärung	57

Ergänzende Verzeichnisse

Abbildungsverzeichnis

2.1	Beispiele diegetischer Benutzeroberflächen in Videospielen	6
2.2	Diegetische Hinweiselemente, die den Spieler auf interagierbare Objekte oder den Weg hinweisen	7
3.1	Beispiel für eine Kamera in Ego-Perspektive	11
3.2	Skizze des Platzierungsebene	13
3.3	Beispiele von Outlines in Videospielen	15
3.4	Glow-Effekt in Counter-Strike 2 zum Hervorheben von Spielercharakteren	16
3.5	Schematische Darstellung der Funktionsweise des Fresnel-Effekts	16
3.6	Beispielhafte Darstellung einer Kugel und eines Würfels mit Fresnel-Effekt	17
3.7	Beispiele von Outlines in Videospielen	18
3.8	Beispiele von Outlines in Videospielen	19
4.1	Flussdiagramm Programmablauf	22
4.2	Aufnahmen der Unity-Szene	23
4.3	Benutzeroberflächenelemente und Menüs	24
4.4	Modellierter Schlüssel in Blender	24
4.5	Schlüssel mit goldenem Material in Unity-Szene	25
4.6	Beispiel Inverted-Hull-Methode	26
4.7	Linieneffekt via Kantenerkennung im Spiel Sable von Shedworks	27
4.8	Schlüssel mit Outline	29
4.9	Schlüssel mit Glow-Effekt	29
4.10	Screenshot aus dem Shader Graph des Fresnel-Shaders	30
4.11	Schlüssel mit Fresnel-Effekt	31
4.12	Shader Graph für Shader des Color-Overlay-Effekts	31
4.13	Schlüssel mit Color-Overlay-Effekt	32
4.14	ShaderGraph für Stripe-Effekt	32
4.15	Schlüssel mit Stripe-Effekt	33
4.16	Beispiel für ein simples Datenkonstrukt im JSON-Dateiformat	33
4.17	Beispiele für einzelne Logs aus den Protokollen	34
5.1	Beispiele von Schätzversuchen mit verschiedenen Objektzahlen N und verschiedenen Hervorhebungsmethoden	36
5.2	Ergebnisse der Umfrage zur Erkennbarkeit der Hervorhebungsmethoden; 5 = sehr gut, 1 = sehr schlecht	37
5.3	Schätzversuche ohne Hervorhebung, Differenz der eingegebenen Schätzungen zu tatsächlich gezeigter Anzahl von Objekten, Polynomiale Trendlinie 4. Grades . . .	38
5.4	Schätzversuche bei Hervorhebung durch Outline, Differenz der eingegebenen Schätzungen zu tatsächlich gezeigter Anzahl von Objekten, Polynomiale Trendlinie 4. Grades	38
5.5	Schätzversuche bei Hervorhebung durch Glow-Effekt, Differenz der eingegebenen Schätzungen zu tatsächlich gezeigter Anzahl von Objekten, Polynomiale Trendlinie 4. Grades	39

5.6	Schätzversuche bei Hervorhebung durch Fresnel-Effekt, Differenz der eingegebenen Schätzungen zu tatsächlich gezeigter Anzahl von Objekten, Polynomiale Trendlinie 4. Grades	39
5.7	Schätzversuche bei Hervorhebung durch Color-Overlay-Effekt, Differenz der eingegebenen Schätzungen zu tatsächlich gezeigter Anzahl von Objekten, Polynomiale Trendlinie 4. Grades	40
5.8	Schätzversuche bei Hervorhebung durch Stripe-Effekt, Differenz der eingegebenen Schätzungen zu tatsächlich gezeigter Anzahl von Objekten, Polynomiale Trendlinie 4. Grades	40

Tabellenverzeichnis

3.1	Übersicht der betrachteten Spiele	14
5.1	Zusammenfassung der Daten aus den Schätzversuchen nach Hervorhebungsmethode	41
5.2	Berechnungen zur einfaktoriellen Varianzanalyse der relativen Abweichungen, gruppiert nach Hervorhebungsmethoden	43
5.3	Auswertung Protokolldaten nach angegebener mittlerer Spielzeit pro Woche	44
A.1	Fragen und zugehörige Antwortmöglichkeiten des Fragebogens	49
A.2	Fragen und zugehörige Antwortmöglichkeiten des Fragebogens	50

1 Einführung und Motivation

Videospiele werden immer beliebter und erfreuen sich einer immer breiteren Spielerschaft. Sie sind schon lange in der Mitte der Gesellschaft angekommen und ziehen Spielende aus allen Bereichen an. Zudem sind Videospiele nicht nur immer leichter erreichbar, es treten auch durch Gamification Elemente von Spielen in neuen Bereichen auf.

Ein bedeutender Anteil der heutigen Spiele sind die sog. *fast-paced* (rasante, temporeiche) Games. Diese zeichnen sich durch den schnellen Ablauf des Spielgeschehens aus, die den Spielenden fordern sollen, ebenso schnell zu handeln. Ableger des Genres siedeln sich häufig im *Ego-Shooter*-Genre an, in dem der Spielende die Perspektive der von ihm gesteuerten Figur einnimmt. Zu bekannten Ablegern der Kategorie zählen hier die *Doom*-Reihe, *Ultrakill*, *Ghostrunner 1 & 2* und *Titanfall 2*. Daneben gibt es auch weiterhin nennenswerte fast-paced Spiele mit zweidimensionaler Optik wie *Hotline Miami 1 & 2*, die *Sonic*-Spiele oder *Pizza-Tower*.

Fast-paced Games im Speziellen verlangen von den Spielenden, dass sie rasch Entscheidungen treffen und auf das rasante Spielgeschehen rechtzeitig reagieren, um mit dem Spiel mithalten zu können. Diese Reaktionen müssen meist nicht nur schnell, sondern auch akkurat sein. Beispielsweise müssen Gegner ins Visier genommen werden oder Tastenkombinationen ausgeführt werden, um benötigte Fähigkeiten auszuführen. Entscheidend für die Schnelligkeit und Genauigkeit der Reaktionen ist, dass die Informationen, die das Spiel dem Spielenden mitteilt, von dem Spielenden in eben jenem begrenzten Zeitraum verstanden und korrekt interpretiert werden können.

Eine zu hohe Menge an Informationen könnte den Spielenden jedoch überfordern. Er muss aus dem Fluss der Informationen und Reize, die das Spiel ihm füttert, eben jene Komponenten herausfiltern können, die für ihn und das Spielgeschehen in dem Moment am wichtigsten sind. Deswegen benutzen viele Videospiele Werkzeuge, um die Aufmerksamkeit des Spielenden auf die wichtigsten und unmittelbar dringendsten Elemente des Blickfeldes zu lenken. Dafür verwenden sie mitunter visuelle Hervorhebungsmethoden (*highlighting methods*), die die Wichtigkeit dieser Elemente signalisieren sollen.

Es werden eine Reihe von verschiedenen Hervorhebungsmethoden eingesetzt, deren Effektivität, Einflüsse und Auswirkungen bisher in diesem Kontext aber nur begrenzt untersucht wurden. Ziel dieser Arbeit soll es sein, diese Hervorhebungsmethoden zu vergleichen. Es soll herausgefunden werden, wie bestimmte Hervorhebungsmethoden die Informationsaufnahme und aus dem Spiel und die Verarbeitung dieser Informationen beeinflussen.

1.1 Aufgabenbeschreibung

Es gilt herauszufinden, welche Hervorhebungsmethoden benutzt werden, wie sie zur Anwendung kommen und ob sie den Spielenden Informationen vermitteln, die akkurat interpretiert werden können. Angesichts dessen sollen die Methoden auf ihren Eindruck auf die Spielenden untersucht werden und wie sie in einem begrenzten Zeitraum wahrgenommen werden

können. Zu diesem Zweck wird eine Auswahl von Spielen, die schnelle Reaktionen erfordern, untersucht, um aus ihnen die Hervorhebungsmethoden abzuleiten, die später verglichen werden sollen. Es wird eine Software mittels der *Unity Engine* umgesetzt, damit die ausgewählten Hervorhebungsmethoden und deren Auswirkungen auf die Probanden in einem anwendungsbezogenen Kontext verglichen werden können.

1.2 Aufbau der Arbeit

Zunächst sollen in Kapitel 2 die theoretischen Grundlagen erklärt werden, auf denen die nachfolgenden Kapitel aufbauen. Hierbei sind sowohl psychologische als auch technische Sachverhalte gemeint, die einerseits der visuellen Wahrnehmung des Menschen und andererseits der informatischen Umsetzung der Software zugrunde liegen. Außerdem wird auf den Begriff 'diegetisch' und seine Bedeutung eingegangen. In Kapitel 3 wird eine Auswahl von Spielen dargelegt und untersucht, um eine Auswahl von Hervorhebungstechniken zu erhalten, die verglichen werden können. Es werden die Anforderungen dargelegt, die ein solcher Vergleich der Hervorhebungsmethoden voraussetzt und ein Konzept vorgestellt, um eine Software umzusetzen, die jene Anforderungen erfüllt. Diese Umsetzung wird anschließend in Kapitel 4 erläutert und es wird beleuchtet, wie diese Software zustande kommt und wie die einzelnen Hervorhebungstechniken implementiert wurden. Die Evaluation der Software und damit der Vergleich der Hervorhebungsmethoden wird in Kapitel 5 in ihrer Planung, Umsetzung und statistischen Auswertung erörtert. Abschließend wird in Kapitel 6 ein Ausblick gegeben, wie zukünftige Arbeiten auf diesem Themengebiet an die vorliegende anknüpfen könnten.

2 Grundlagen

In diesem Kapitel soll die theoretische Basis für die nachstehende Konzeption und Umsetzung der Software zum Vergleich der Hervorhebungsmethoden geschaffen werden. Die Grundlagen der menschlichen Wahrnehmung werden dargelegt, um einen Überblick über das Themengebiet zu geben und zum Verständnis der folgenden Kapitel beizutragen. Es werden ähnliche, themenbezogene Arbeiten angesprochen, um den Stand der Forschung zu zeigen. Ferner findet sich hier auch Platz, um die technischen Grundlagen, auf dem der Rest der Arbeit aufbaut, zu erläutern.

2.1 Visuelle Wahrnehmung

Menschliche Wahrnehmung

Die menschliche Wahrnehmung ist ein komplexer Prozess mit dem Ziel, die Umwelt zu Erfassen, der in mehreren Stufen und an mehreren Orten des Körpers stattfindet. Dabei ist der Ausgangspunkt ein physikalischer Stimulus, ein sog. *distaler* Reiz, der auf eine der vielen verschiedenen Rezeptorzellen des Körpers trifft. Durch dieses Aufeinandertreffen werden neuronale Reaktionen ausgelöst, welche weiterverarbeitet werden können. Aufgrund dieser neuronalen Aktivität kann im Gehirn der nun *proximale* Reiz analysiert und mithilfe von kognitiven Abläufen verstanden, interpretiert und ein Abbild des externen Reizes rekonstruiert werden. Diese Einzelreize werden mithilfe Assoziationen, Erwartungen und Emotionen interpretiert und nach Verständniseinheiten durchsucht. [Sch21, S.27f]

Für die Verarbeitung eines Reizes sind verschiedene Eigenschaften der Sinneswahrnehmung eine von Bedeutung [Les20, S.12]:

- Modalität (z.B. Sehen, Hören, Gleichgewichtssinn etc.)
- Intensität
- Dauer
- Lokalisation

Farben und Kontraste

Für die menschliche visuelle Wahrnehmung spielen Farben und Kontraste eine zentrale Rolle. Kanten von Objekten werden durch visuelle Kontraste wahrgenommen, die im Bild nahe beieinander liegen [Les20, S. 9]. Diese Kontraste können unterschiedlicher Natur sein und werden verschieden wahrgenommen. So ist wird eine Hell-Dunkel-Kontrast primär durch die Stäbchen im Auge wahrgenommen, während farbliche Kontraste durch die Zapfen erfasst werden. Für die menschliche Wahrnehmung ist der Hell-Dunkel-Kontrast der stärkste. Demnach fallen beispielsweise helle Objekte vor dunklem Hintergrund schneller ins Auge. Dabei wird der Hell-Dunkel-Kontrast auch von Farben beeinflusst. So wird gelb als heller wahrgenommen als blau. Dies liegt an den Zäpfchen Im Gegensatz dazu sind Komplementärkontraste Unterschiede im Farbton, Muster Der Farbkontrast sorgt dafür, dass Gegenfarben wie

Grün und Rot oder Gelb und Blau sich gegenseitig verstärken, wenn sie von benachbarten Punkten im Raum ausgehen, wie etwa bei einem Objekt vor einem Hintergrund, der in der entsprechenden Gegenfarbe gefärbt ist [Les20, S.15].

Bildkonstruktion

Die Aufnahme und Verarbeitung visueller Reize läuft in zwei getrennten Prozessen ab. Der *präattentive* (ungesteuerte) Prozess, der sehr schnell und hochgradig parallel abläuft und für das Erkennen [BKV22, S.22] der einfachsten und wichtigsten Umrisse und Merkmale von Objekten sorgt. Dazu gibt es einen langsameren *attentiven* (gesteuerten) Prozess, der feinere Details erfasst und durch eine gerichtete Aufmerksamkeit gesteuert wird. [BW17, S.131]; [Les20, S.13]

Kognitive Prozesse

Kognitive Elemente spielen bei der visuellen Wahrnehmung und spezifisch im attentiven Prozess eine zentrale Rolle, um die wahrgenommenen Reize verarbeiten und interpretieren zu können. Dabei greifen sie auf Erfahrungen und Erinnerungen zurück, um das Gesehene als "bekanntöder unbekanntëinzustufen. Auch ist die Erwartung des Betrachtenden von Bedeutung. Diese lenkt die Aufmerksamkeit und beeinflusst, welche Reize in welchem Ausmaß wahrgenommen und welche ignoriert werden. Dadurch, dass Erwartungen und Erfahrungen die Wahrnehmung beträchtlich steuern und prägen ist anzunehmen, dass so das Wahrgenommene sich interindividuell unterscheidet. Wahrnehmungen sind ebenso Ergebnis logisch-analytischer Abläufe der Kognition wie sie auch psychologisches Produkt der Aufmerksamkeit sind und damit auch von Wünschen, Erwartungen und Emotionen gestaltet werden. [Sch21, S.31]

Prozesse des Mengenerfassung

Das Erfassen von Mengen und deren Zahlen läuft über verschiedene Prozesse ab, die abhängig von Kontext, Anzahl und Erscheinung der Elemente zu tragen kommen [WR00].

Simultanerfassung (eng. *Subitizing* beschreibt die Fähigkeit, die Anzahl von mehreren Elementen in einer Gruppe akkurat und schnell zu erfassen, ohne diese einzeln abzählen zu müssen. Dies ist vor allem der Fall, wenn die Anzahl der zu erfassenden Objekte vier oder kleiner ist [Kau+49]. Mengen mit mehr als vier oder fünf Objekten können nicht simultan erfasst werden, außer sie erscheinen in Mustern die dem Betrachtenden bereits bekannt sind.

Simultanerfassung ist dabei kein vollständig unterbewusster, präattentiver Prozess, der keine gesonderte Aufmerksamkeit benötigt. Denn sie ist wesentlich akkurater, wenn der Betrachtende seine Aufmerksamkeit auf die zu erfassende Menge und die Aufgabe des Erfassens richtet und damit auch kognitive Prozesse involviert [Rai+08].

In Studien, die das Bestimmen und Einschätzen der Anzahl von schnell aufblinkenden Elementen erforschten, kamen zu dem Schluss, dass das Spielen von Videospielen mit einem besseren visuellen Erfassungsvmögen korreliert [GB03; GB05]. Demnach ist die Anzahl von

Objekten, die simultan erfasst werden können, bei Spielenden (*video game players*, VGPs) oft höher als bei nicht-spielenden Personen (*non-video game players*, NVGPs). Diese Schlüsse sind zwar in der Wissenschaft nicht unumstritten [MS09], jedoch scheint der generelle Konsens den oben genannten Befunden zu entsprechen [SF10; CG23]. Dieser Sachverhalt soll auch in dieser später beschriebenen Studie betrachtet werden.

Wenn sich Mengen in regelmäßige Untergruppen unterteilen lassen, können sie wesentlich schneller und einfacher verarbeitet werden, da die Stückzahl in der Untergruppe bestimmt und auf die Gesamtmenge extrapoliert werden kann. Dieser Prozess des Unterteilen und Aufaddieren ist schneller als simples Zählen. [WR00, S. 11]

2.2 Diegetische und nicht-diegetische Hervorhebungsmethoden

Diegese und Diegetisch

Der Fokus dieser Arbeit liegt in erster Linie auf nicht-diegetischen Hervorhebungsmethoden. Da die Begriffe '*diegetisch*' bzw. '*nicht-diegetisch*' im allgemeinen Sprachgebrauch, in der Wissenschaft und in der Spieleentwicklung unterschiedlich angewandt und bisweilen mit abweichenden Bedeutungen verstanden werden, soll an dieser Stelle für den Kontext dieses Projektes kurz die Bedeutung des Begriffs nicht-diegetisch erläutert werden.

Der Begriff der 'Diegese' bzw. 'diegetisch' findet seinen Ursprung in der Filmwissenschaft und kann sich auf Etienne Souriau bzw. seine Tochter Anne Souriau zurückführen lassen, die den Begriff Anfang der 1950er als *diégèse* prägten. Er leitet sich aus dem Altgriechischen *διηγησις* (*dihegesis*, dt. Erzählung, Erörterung, Ausführung) ab:

"Diegetisch ist alles, was man als vom Film dargestellt betrachtet und was zu Wirklichkeit, die er in seiner Bedeutung *voraussetzt*, gehört." [Sou97]

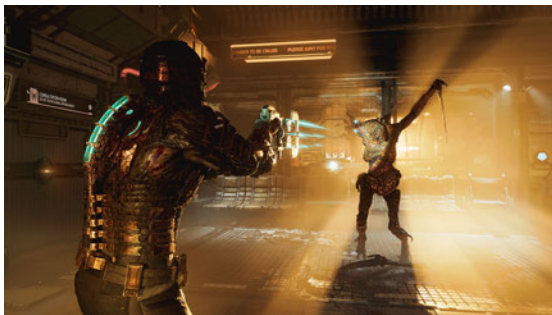
Mit anderen Worten ausgedrückt beschreibt diegetisch alles, was in der erzählten Welt existiert. Charaktere, Gegenstände, Handlungsstränge sind alle diegetisch, solange sie innerhalb der Erzählung stattfinden. Nicht-diegetisch ist hingegen alles das, was zwar vom Film vermittelt wird, aber nicht teil der Erzählung ist. Als Beispiele hierfür werden meist Sequenzen wie die Vorstellung der beteiligten Produktionsstudios im Vorspann des Filmes, die Credits oder auch Einblendungen mit Orts- bzw. Zeitangaben angeführt. [Fux07]

Diegetische Spielelemente

Zwar bezieht Souriau die Definitionen der Begriffe nur auf den Film, jedoch kann 'diegetisch' in jeder Erzählung unabhängig des Mediums in dem sie vermittelt wird angewandt werden. Solange eine Erzählung vermittelt wird, gibt es auch eine Diegese. Im Game-Design lässt sich der Begriff diegetisch meist im Bezug auf Objekte finden, die für gewöhnlich nicht diegetisch sind und wird dementsprechend geradezu als Ausnahme zur Norm definiert. Da Videospiele schließlich auch nur eine Software sind, mit der interagiert werden muss, gibt

es meist Elemente der Benutzeroberfläche, die sich nur unter erheblichem kreativem Aufwand als diegetische Gebilde einbringen lassen. Deswegen sind Menüs und Anzeigen für Lebenspunkte, Erfahrungspunkte etc. häufig nicht-diegetisch dargestellt. Aus diesem Grund wird es als Konventionsbruch empfunden, wenn solche Elemente als Teil der erzählten Spielwelt integriert sind. Ein anschauliches Beispiel liefert *Dead Space* (siehe Abb. 2.1a) oder auch *Gloomwood* (siehe Abb. 2.1b), wo Informationen wie Lebenspunkte oder auch das Inventar des Spielercharakters in die erzählte Welt eingebaut sind.

Es lässt sich hier feststellen, dass einige Hervorhebungen und generell Spielelemente sich in beiden Kategorien wiederfinden können und die Zuordnung nicht immer gänzlich eindeutig treffen lässt. Die Entscheidung, ob ein spezifisches Spielelement diegetisch ist oder nicht, hängt substanziell von der erzählten Spielwelt, von ihren Regeln und ihrer Darstellung ab. Diese können zuweilen recht skurril sein und keinen bedeutenden Realitätsanspruch stellen. Stellenweise lässt sich auch ein Spielercharakter als solcher nicht identifizieren. Deshalb lässt sich in Fällen wie *Balatro* oder *Nubby's Number Factory* schwer bis kaum entscheiden, ob gewisse Spielelemente nur vom Spielenden gesehen werden oder auch von der gesteuerten Figur.



(a) Screenshot aus *Dead Space* (2023)¹ – die blauen Elemente des Raumanzuges stellen die Anzeige der Lebenspunkte der Spielerfigur dar.



(b) Screenshot der Inventaransicht in *Gloomwood*²

Abbildung 2.1: Beispiele diegetischer Benutzeroberflächen in Videospielen

Nicht-Diegetische Hervorhebungsmethoden

Diegetisch lässt sich in Videospielen also als alles das verstehen, was der vom Spieler gesteuerte Charakter in der erzählten Welt sehen und erfahren kann. Hervorhebungsmethoden können diegetisch sein, wenn sie auch dem Charakter selbst Hinweise und Informationen liefern sollen. Das können Animationen in Fighting Games sein, in denen die Figur übertrieben ausholt, um zu signalisieren, dass sie gleich zuschlagen wird. Es können aber auch Lichtquellen sein, die dem Spieler den Weg zum nächsten Gebiet aufzeigen sollen. Ein berühmteres Beispiel, welches in letzter Zeit in Verrufung geraten ist, sind die gelben Hinweislinien in *Final Fantasy VII Rebirth* und der Neuauflage von *Resident Evil 4* von 2023 (siehe Abb. 2.2).

¹Bildquelle – *Dead Space* via Steam Store: https://store.steampowered.com/app/1693980/Dead_Space/

²Bildquelle – *Gloomwood* via Steam Store: <https://store.steampowered.com/app/1150760/Gloomwood/>

³Bildquelle – *Final Fantasy VII Rebirth* Screenshot von @DaveOshry via X/Twitter:

<https://x.com/DaveOshry/status/1755469586999505137>

⁴Bildquelle – *Resident Evil 4* (2023) Screenshot, Forumbeitrag via *ResetEra*:

<https://www.resetera.com/threads/debate-over-resident-evil-4-remakes-yellow-paint-resurfaces-as-devs-explain-just-how-much-help-players-really-need-realism-is-less-immersive.772340/page-10>



(a) Gelbe Farbe an Felsvorsprüngen sollen dem dem Spielenden auf den Kletterweg hinweisen³



(b) Gelbe Farbe auf Fässern zeigt dem Spielenden an, dass mit ihnen interagiert werden kann⁴

Abbildung 2.2: Diegetische Hinweislelemente, die den Spieler auf interagierbare Objekte oder den Weg hinweisen

Nicht-diegetische Hervorhebungen sind dementsprechend nun alle die, die dem Spieler, aber *nicht* der Spielerfigur selbst übermittelt werden. Hervorhebungen wie Umrisse oder Einfärbungen können dabei diegetisch sein, wenn sie durch eine zusätzliche Ebene entstehen, die auch der Spielerfigur vorgehalten wird, wie etwa die Fähigkeit des *Witchersinnes*, das besondere Hinweise markiert. Im Bezug auf die Spieleentwicklung wird teilweise nochmals danach unterschieden, ob die zu beurteilende Benutzeroberfläche auch in die Spielwelt integriert ist – als *spatial* (räumlich) betitelt – oder ob diese gänzlich getrennt vom Spielraum, wie etwa eine übergeordnete, zweidimensionale Ebene mit Menüs, agiert. Letzteres wird meistens als *true non-diegetic*, also als wahrhaftig nicht-diegetisch bezeichnet, da diese Art der Interaktion komplett unabhängig von der Spielwelt existieren kann. In dieser Arbeit soll sich nun auf die Kategorie nicht-diegetische Hervorhebungsmethoden als Gegensatz zu diegetischen Techniken konzentriert werden. Die nochmalige Unterteilung findet hier keine Anwendung.

2.3 Technische Grundlagen

3D-Modelle

Ein 3D-Modell besteht aus einem *Mesh* (Netz) aus Polygonen. Diese Polygone sind üblicherweise Dreiecke, die von drei Knotenpunkten (sog. *Nodes*) aufgespannt werden. Zusätzlich hat jedes Polygon einen Normalenvektor, der orthogonal auf diesem steht und angibt, in welche Richtung die Fläche zeigt.

Unity-Engine

Zur Erstellung von Videospielen werden häufig sogenannte *Engines* benutzt, die, ähnlich wie ein Werkzeugkasten, Frameworks, Funktionen und Komponenten als Werkzeuge bereitstellen, welche häufig in der Spielprogrammierung vorkommen und so die Spieleentwicklung erheblich erleichtern. Für die Umsetzung dieses Programms wurde die Engine Unity benutzt. Sie ist einer der beliebtesten und verbreitetsten Werkzeugkästen in der Videospielebranche.

Shader

Shader sind zentrale Werkzeuge der Videospieleentwicklung und maßgeblich an der visuellen Präsentation und Darstellung von Videospielen beteiligt. Grundsätzlich ist ein Shader ein kleines Programm, das auf der Grafikkarte (*Graphics Processing Unity*, GPU) des Systems ausgeführt wird. Shader verwandeln die Eingabe der 3D-Szene in ein zweidimensionales Bild, was auf dem Bildschirm ausgegeben werden kann. Dieser Prozess wird *Rendern* genannt. Dabei muss der Shadercode üblicherweise über jeden Vertex der Szene und über jeden Pixel des entstehenden Bildes laufen. Da sich die jeweilige Shader-Sequenz nur auf je einen Vertex oder Pixel konzentriert, können die einzelnen Berechnungen unabhängig voneinander und zeitgleich ausgeführt werden, um die Vielzahl der Schritte zu stemmen und das Bild in Echtzeit darstellen zu können [Dop18].

2.4 Hervorhebungsmethoden im Vergleich

Es gibt bereits einige Vergleiche von Hervorhebungsmethoden in Videospielen. Dabei wird meistens eine Reihe von sowohl diegetischen als auch nicht-diegetischen Hervorhebungstechniken untersucht.

Nicht-Diegetische Methoden scheinen dabei die Aufmerksamkeit im Kontext der Virtuellen Realität besser einzufangen als statische diegetische Hervorhebungsmethoden, wie z.B. Lichtstrahlen. Darüber hinaus werden Konturen bzw. Einfärbungen auch von den Spielenden auch als immersiver bewertet. [Bel24]

Studien, die Eye-Tracker benutzen, um die Augenbewegungen und damit die Aufmerksamkeit von Spielenden besser nachvollziehen zu können, kommen jedoch zu dem Schluss, dass Animationen gegenüber Konturen und Einfärbungen die Aufmerksamkeit von Spielenden besser einfangen und halten können [Lin23].

Auch wurden bereits Untersuchungen angestellt, welche Hervorhebungsfarbe die Aufmerksamkeit von Spielenden am effektivsten einfängt. Diese kommen zu mitunter widersprüchlichen Ergebnissen. Zum einen scheint die Farbe der Hervorhebung nur eine untergeordnete Rolle zu spielen, wenn auch die Farbe Rot von den Teilnehmenden als am besten sichtbar genannt wurde [OS20]. Weitere Forschungen kommen unter Nutzung von Blickerfassung jedoch zu dem Schluss, dass Gelb oder Grün die Aufmerksamkeit der Spielenden besser einfängt und halten kann [Lin23].

Die Faktoren der Ablenkung durch Hervorhebungstechniken besonders im Anwendungsfall der Videospiele sind aktuell noch recht unerforscht. Im Gegensatz zu den Merkmalen der Wahrnehmbarkeit der Hervorhebungen, fokussieren sich vergleichsweise wenige Studien auf die der Ablenkungen. Ersteres ist einfacher zu definieren und zu messen und es gibt in dieser Hinsicht bereits etablierte Verfahren und Experimente, um die Wahrnehmbarkeit zu messen. Zudem ist die Wahrnehmbarkeit von Hervorhebungen auch besser erforscht, da die Hervorhebungsmethoden dazu ausgelegt sind, wahrgenommen zu werden. Ablenkung durch Hervorhebungstechniken ist hingegen weniger erforscht und es gibt nur sehr wenige Ansätze, um diese zu untersuchen. Richtlinien zum Entwerfen von Hervorhebungstechniken, die die Ablenkung mit der kommunzierten Wichtigkeit des hervorzuhebenden Elements verbinden, sind noch kaum wesentlich belegt und bedürfen weiterer Forschung. [Leu17]

Diese Arbeit soll sich nun, aufgrund der erläuterten Kenntnislage, mit dem Einfluss von Hervorhebungsmethoden in größeren Anzahlen beschäftigen und erforschen, wie sich eine Menge von hervorgehobenen Objekten auf die Wahrnehmung eben dieser Objekte auswirkt. In den obigen Arbeiten wurde entweder der Einfluss von eines einzelnen hervorgehobenen Objektes oder wenigen Objekten beobachtet. In den betrachteten Studien wird die Hervorhebung als Hilfsmittel oder Ablenkung benutzt, um daran die Erfüllung einer bestimmten Aufgabe zu messen. Hier werden nun größere Objektanzahlen betrachtet und untersucht, ob es einen Zusammenhang zwischen der Darstellung von Mengen von Objekten bzw. die Hervorhebung dieser mit verschiedenen visuellen Techniken und der wahrgenommenen, geschätzten Anzahl dieser Objekte gibt. Auch wird hier spezifisch eine Verbindung zu einem Videospielkontext gezogen und die Hervorhebungsmethoden in einer dreidimensionalen Umgebung getestet, die einem Videospiel mit First-Person-Perspektive nachempfunden wurde. Aufgrund der dargestellten Kenntnislage lässt sich die Vermutung aufstellen, dass sich ein Zusammenhang zwischen dem Kontrast der Darstellung von Objekten und der Einschätzung der Anzahl dieser Objekte herstellen lässt.

3 Konzeption

Im Sinne dieser Arbeit soll eine Software entstehen, die es erlaubt, die Wahrnehmung von verschiedenen Hervorhebungsmethoden zu evaluieren. In diesem Kapitel wird nun erläutert, welche Anforderungen diese Software hat und es wird ein Konzept vorgestellt, mit dem diese Anforderungen erfüllt werden sollen.

3.1 Szenenaufbau

In Unity ist ein Spiel in Szenen aufgeteilt. Diese stellen eine digitale, räumliche Umgebung dar, in dem sich verschiedene Elemente wie 3D-Modelle, Kameras, Lichtquellen und Benutzeroberfläche befinden. Um die Studie möglichst auf den Kontext von Videospielen beziehen zu können, soll die entstehende Software einer Videospieldumgebung visuell möglichst nahe kommen. Aus diesem Grund wurde eine Konstellation gewählt, die der eines Spiels in Ego-Perspektive nachempfunden wurde. Demnach übernimmt der Spielende die Sichtweise der von ihm gesteuerten Figur. Für die Szenenerstellung bedeutet dies, dass die Kamera und damit der Spielende annähernd horizontal auf Augenhöhe in eine dreidimensionale Umgebung blickt (siehe Abb.3.1).

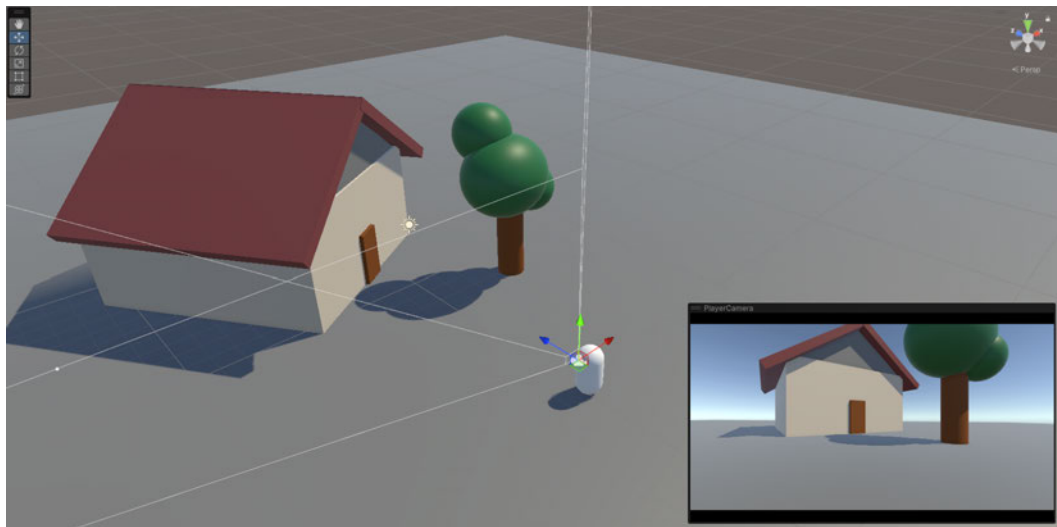


Abbildung 3.1: Beispiel für eine Kamera in Ego-Perspektive, im Bildausschnitt unten rechts die Kameraperspektive des Spielers; Screenshot aus Unity

3.2 Wahrzunehmende Objekte

Da die Ergebnisse der Studie dadurch beeinflusst werden könnten, dass verschiedene Objekte verschieden gut erkennbar sein können, wurde sich dazu entschieden, ein einzelnes Objekt in verschiedenen Anzahlen auf dem Bildschirm zu präsentieren. Wie im Grundlagenkapitel unter Abschnitt 2.1 bereits erwähnt, hängt die Wahrnehmung und Erkennung von Objekten auch davon ab, ob das spezifisch zu erkennende Objekt dem Betrachter bereits

bekannt ist. So soll für dieses Projekt ein Gegenstand zum Erkennen gewählt werden, den die Probanden mit an Sicherheit grenzender Wahrscheinlichkeit bekannt ist und den sie von allen Blickwinkeln erkennen können.

Um jedoch die Ergebnisse der Evaluation bestmöglich auf den Kontext eines Videospiele beziehen zu können, sollen sowohl das wahrzunehmende Objekt als auch der Hintergrund einem Videospiel bestmöglich nahekommen. Für dieses Experiment soll erforscht werden, wie Menschen Mengen von Objekten in kürzester Zeit wahrnehmen können, wenn diese mit verschiedenen Methoden hervorgehoben wurden. Dafür ist es nicht gewünscht, dass die Probanden die einzelnen Objekte in diesen Mengen zählen können und darüber die Zahl der Objekte erfassen. Deswegen dürfen die Objekte nur für eine begrenzte Zeit auf dem Bildschirm bleiben, um das Zählen zu verhindern. Dies ist auch für die Nähe zum Spielekontext günstiger, da hier nur selten Objekte direkt gezählt werden müssen, statt die Anzahl der Objekte schnell grob zu überschlagen.

3.3 Objekt-Platzierung

Die Schlüssel müssen nun in der Szene platziert (eng. *spawning*) werden. Um Verdeckungen auszuschließen eine konstante Größe der Objekte auf dem Bildschirm zu garantieren, werden die Objekte in einer Ebene parallel zur Ebene der Kamera platziert. Es wird ein Rechteck gewählt, welches sich in ausreichendem Abstand im Pyramidenstumpf des Kamerasichtfeldes befindet (siehe Abb. 3.2). Dieses Rechteck ist etwas kleiner als die volle Höhe und Breite der Bildebene an diesem Punkt, um auszuschließen, dass Objekte sich übermäßig außerhalb des Sichtfeldes aufhalten. Die Schlüsselpositionen müssen randomisiert sein, um mögliche Muster auszuschließen, die die Erfassung der Menge vereinfachen könnten.

Zusätzlich müssen sich die Schlüssel in einer zufälligen Rotation zum Betrachtenden befinden. Dies simuliert die Realität von herkömmlichen Spielwelten besser, da dort Objekte meist in allen möglichen Orientierungen befinden können. Jedoch sollten die Schlüssel nicht um eine Achse rotiert werden, die sich senkrecht

3.4 Auswahl der nicht-diegetischen Hervorhebungsmethoden

Zentral für dieses Projekt ist die Auswahl der Hervorhebungsmethoden. Zu diesem Zweck werden acht Spiele betrachtet und ihre nicht-diegetischen Hervorhebungstechniken ausgemacht. Die Spiele wurden aufgrund ihres hohen Tempos und der Schnelligkeit der Reaktionen, die vom Spielenden erwartet werden ausgewählt. Dabei kommen primär aus zwei verschiedenen Genres, die jedoch obiges gemeinsam haben. Einerseits finden sich hier Spiele aus dem *Fast-Paced* (dt. temporeich, schnellebig) Genre. Ihr Spielgeschehen läuft häufig sehr schnell ab und die Spielenden müssen rasch Entscheidungen treffen und auf Reize reagieren, die diese Spiele senden. Vertreter des Fast-Paced Genres zeichnen sich durch ein actionreiches Spielerlebnis aus, welches durch recht simple und einfach verständliche Spielmechaniken verursacht werden, da hier der Fokus auf dem Spielfluss liegt, der durch die Aktionen und Reaktionen des Spielenden entsteht. Diese Spiele sind oft nur alleine spielbar, die Spielenden

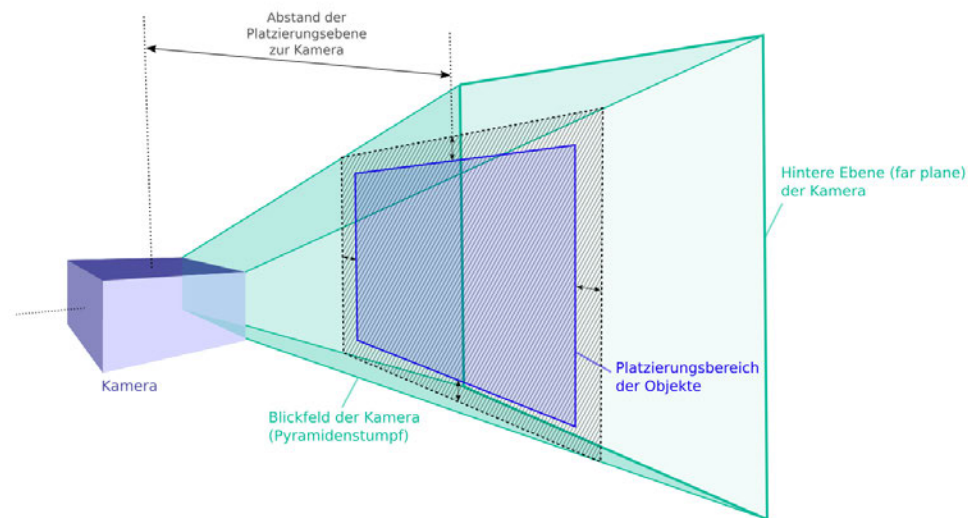


Abbildung 3.2: Skizze des Platzierungsebene

sind also Einzelspieler. Die zweite Kategorie an Spielen, die betrachtet wurde sind Mehrspielertitel, in denen Spieler gegeneinander antreten (Player versus Player, *PvP*). Hierbei wurde ebenso darauf geachtet, dass diese Spiele auch auf ein rasantes Spielgefühl hinarbeiten, welches durch schnell gefragte Reaktionen und Entscheidungen entstehen soll. Diese entstehen hierbei häufig durch die Aktionen anderer Spieler, auf die passende Antworten gefunden werden müssen. Bei beiden Kategorien wurden dabei Spiele priorisiert, welche auf einen peniblen Realitätsanspruch verzichten, um den Einsatz von nicht-diegetischen Hervorhebungsmethoden zu fördern. Aufgrund dieser Kriterien wurden folgende Spiele untersucht (siehe Tab. 3.1).

Schlüsse aus der Untersuchung

Durch die Untersuchung der Spiele konnten neben diversen diegetischen vier verschiedene nicht-diegetische Hervorhebungstechniken gefunden werden. Diese sollen im Folgenden näher vorgestellt werden.

Outline

Eine der in den betrachteten Beispielen am häufigsten auftretenden Methoden ist die *Outline* (engl. Kontur), eine solide, farbige Umrandung des Objektes. Allgemein wird sie in Spielen häufig zur Markierung von interagierbaren Gegenständen oder Figuren eingesetzt. Sie besitzt sowohl auf der Innenseite zum Objekt als auch auf der Außenseite zur Umgebung eine solide Kante, die starke Kontraste zwischen dem Objekt selber und der Outline als auch zwischen

Spiel	Jahr	Einzel- / Mehrspieler	Genre	Perspektive
<i>Doom Eternal</i> Entwickler: id Software Publisher: Bethesda Softworks	2020	Einzelspieler	Action, Shooter	Ego-Perspektive
<i>Hades</i> Entwickler: Supergiant Games Publisher: Supergiant Games, Private Division	2020	Einzelspieler	Rogue-like, Action-RPG	Isometrisch
<i>Ghostrunner 2</i> Entwickler: One More Level Publisher: 505 Games	2023	Einzelspieler	Action, 3D-Platformer	Ego-Perspektive
<i>League of Legends</i> Entwickler: Riot Games Publisher: Riot Games	2009	Mehrspieler, PvP	MOBA	Isometrisch
<i>Overwatch 2</i> Entwickler: Blizzard Entertainment Publisher: Blizzard Entertainment	2023	Mehrspieler, PvP	Hero Shooter	Ego-Perspektive
<i>Valorant</i> Entwickler: Riot Games Publisher: Riot Games	2020	Mehrspieler, PvP	Hero Shooter, Taktik-Shooter	Ego-Perspektive
<i>Rocket League</i> Entwickler: Psyonix Publisher: Psyonix	2015	Mehrspieler, PvP	Sports	Third-Person-Perspektive
<i>Counter-Strike 2</i> Entwickler: Psyonix Publisher: Psyonix	2023	Mehrspieler, PvP	Taktik-Shooter	Ego-Perspektive

Tabelle 3.1: Übersicht der betrachteten Spiele

der Outline und dem Umfeld des Objektes erlaubt. Somit ist es im Prinzip eine der auffälligsten Methoden, da die menschliche Wahrnehmung starke Kontraste leichter und intuitiver erfassen kann.

⁵Bildquelle – Screenshot aus League of Legends – Spielaufzeichnung von *Skin Spotlight* via YouTube: <https://www.youtube.com/watch?v=i2zmFjRM70k>

⁶Bildquelle – Screenshot aus Overwatch 2 – Spielaufzeichnung von *Orbital Gameplay* via YouTube: <https://www.youtube.com/watch?v=GxVUa6vwGYw>

⁷Bildquelle – Screenshot aus Hades – Spielaufzeichnung von *Gamer's Little Playground* via YouTube: <https://www.youtube.com/watch?v=dPmMCzvLs9E>



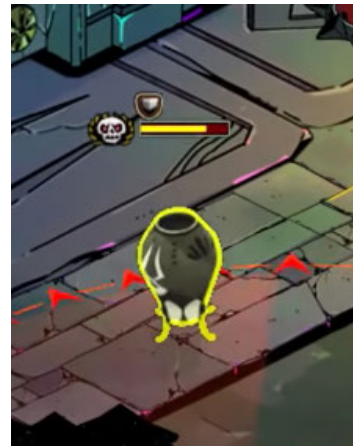
(a) Verwendung einer blauen Outline in League of Legends zum Markieren ausgewählter verbündeter Einheiten⁵



(b) Verwendung einer roten Outline in League of Legends zum Markieren ausgewählter gegnerischer Einheiten⁵



(c) Verwendung der Outline in Overwatch 2 zum Markieren von verbündeten Figuren, auch hinter Objekten⁶



(d) Verwendung einer gelben Outline in Hades zum Markieren von gepanzerten Gegnern⁷

Abbildung 3.3: Beispiele von Outlines in Videospielen

Glow-Effekt

Der *Glow*-Effekt (engl. Schein) ist wie die Outline ein Umriss des hervorzuhebenden Objektes, jedoch besitzt er im Gegensatz nach außen keine harte Kante, sondern wird von innen nach außen transparenter. Dadurch sieht das Objekt so aus, als würde eine farbiger Schein von ihm ausgehen. Der Glow-Effekt wird ähnlich wie die Outline benutzt, jedoch hat er den Vorteil, dass er durch seine graduellen Verlauf sanfter in das restliche Bild einfügt und so weniger aufdringlich wirkt.

Fresnel

Der Fresnel-Effekt ist ein visueller Effekt, der in Videospielen unterschiedliche Anwendungen findet. Durch diese Darstellungsweise werden Flächen, welche sich in einem spitzen Winkel zum Betrachtenden befinden, heller bzw. reflektierender dargestellt werden. Wie in Abb. 3.5 schematisch dargestellt, führt eine größere Differenz zwischen dem Einfallswinkel und der Normalen der Fläche zu einer höheren Intensität des Effektes.



Abbildung 3.4: Glow-Effekt in Counter-Strike 2 zum Hervorheben von Spielercharakteren

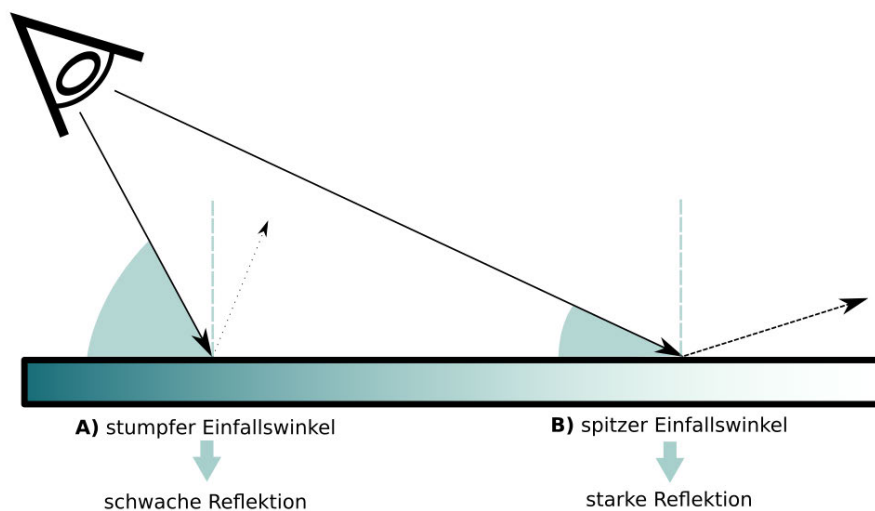


Abbildung 3.5: Schematische Darstellung der Funktionsweise des Fresnel-Effekts

Ein Objekt mit Fresnel-Effekt sieht demnach meist reflektiver oder glatter aus und kann mitunter anders beleuchtet dargestellt werden, als es seine Umgebung und ihre Lichtverhältnisse zulassen würde. Das Objekt wirkt durch seine hervorgehobenen Kurven und Ausprägungen in seiner Form räumlicher, es wird ihm verliehen mehr Tiefe verliehen, so dass es mehr aus seiner Umgebung hervorsticht. Der Fresnel-Effekt ist klassischerweise subtiler als die Outline- und Glow-Methoden, da hier das Aussehen des Objektes beeinflusst wird, statt ein zusätzliches visuelles Element um das Objekt zu erstellen, welches beim Betrachten sofort auffällt. Als Hervorhebungstechnik wird der Fresnel-Effekt meist mit einer Signalfarbe dargestellt, in manchen Spielen wird er auch aus artistischeren Gründen benutzt, um Dinge

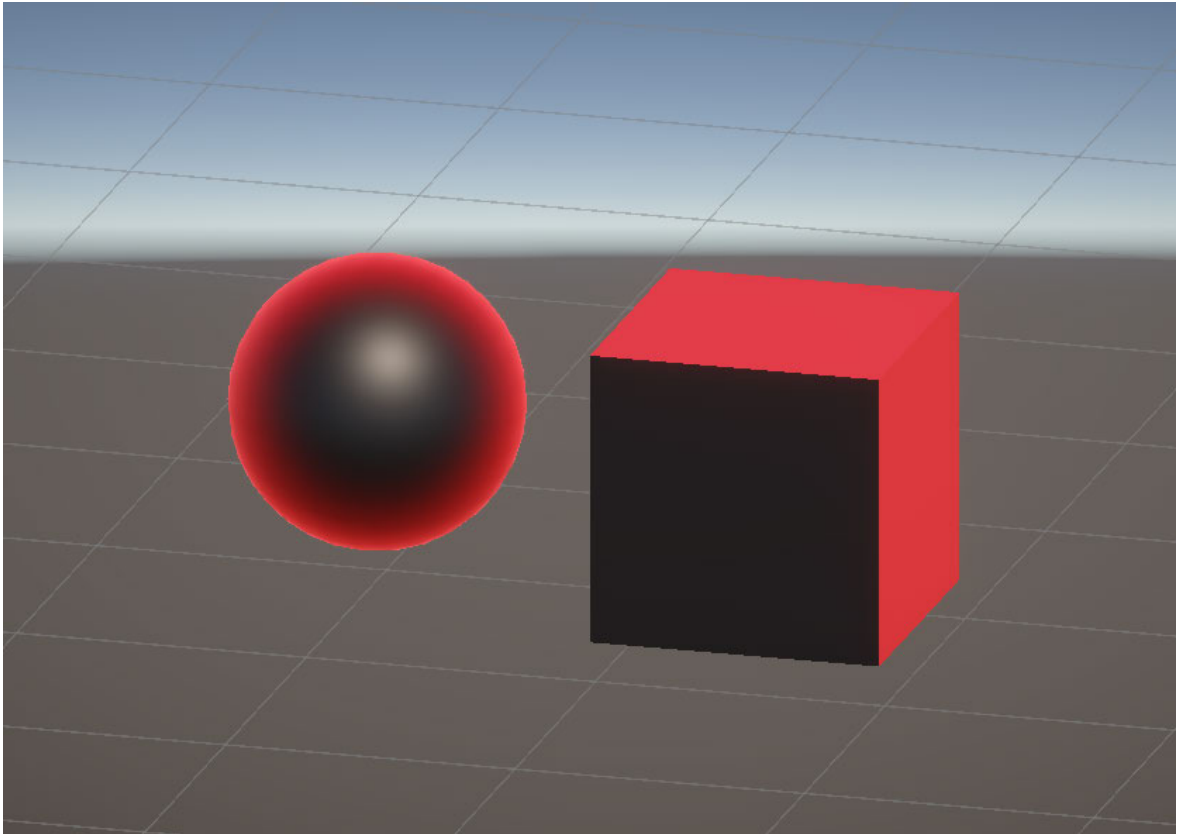


Abbildung 3.6: Beispielhafte Darstellung einer Kugel und eines Würfels mit einem roten Fresnel-Effekt, dieser kommt bei Objekten mit bzw. ohne harte Kanten unterschiedlich zur Geltung

leuchtend darzustellen. Der Fresnel-Effekt funktioniert am besten, wenn das Objekt über Rundungen verfügt und damit der graduelle Übergang sichtbar wird, der durch die Krümmung der Oberfläche, weg vom Betrachter, entsteht. Bei spitzen, kantigen Objekten mit scharfen Winkeln und harten Kanten ist der Fresnel-Effekt jedoch nur begrenzt von Nutzen, da hier der Übergang nicht sichtbar ist und die Hervorhebung im schlimmsten Fall nicht als solche wahrgenommen wird (siehe Abb.3.6).

Color-Overlay-Effekt

Die *Color-Overlay*-Methode (engl. Farbüberlagerung) ist eine einfarbige, halbdurchsichtige Färbung des Objektes. Sie wird üblicherweise dazu eingesetzt, Spielelemente sichtbar zu machen, die eigentlich durch andere Objekte wie etwa die Umgebung verdeckt werden. Da hier das Augenmerk auf der Silhouette des hervorzuhebenden Elementes liegt und nicht auf den Details, muss dieses Element nur anhand seiner Silhouette identifizierbar sein. Da die meisten Charaktere in PvP-Spielen sowieso nach diesem Kriterium entworfen werden, sind diese für eine Anwendung des Color-Overlay-Effektes gut geeignet. Eine weitere Variante des Effektes kann in Kombination mit Animationen genutzt werden, um den Effekt blinken zu lassen, wodurch zwischen dem Aufblitzen des Effektes das Objekt erkennbar bleibt.

⁸Bildquelle – Screenshot aus Doom Eternal – Spielaufzeichnung von *SourceSpy91* via *YouTube*:
<https://www.youtube.com/watch?v=Glr03B45YLQ>

⁹Bildquelle – Screenshot aus Valorant – Spielaufzeichnung von *GTX 1050 Ti* via *YouTube*:
https://www.youtube.com/watch?v=yhSNWzt_WCY



(a) Leuchtend blaue Fresnel-Hervorhebung von verwundbaren Gegnern in Doom Eternal⁸



(b) Leuchtend orange Fresnel-Hervorhebung von verwundbaren Gegnern in Doom, wenn sich diese in Reichweite des Spielers befinden⁸



(c) Verwendung des Fresnel-Effekts in Valorant zum Hervorheben von Spielercharakteren⁹



(d) Verwendung des Fresnel-Effekts zum Hervorheben interagierbarer Objekte in Ghostrunner 2¹⁰

Abbildung 3.7: Beispiele von Outlines in Videospielen

Diese werden für die entstehende Software als Grundlage verwendet und dementsprechend umgesetzt.

Texturierte Methoden

Hervorhebungen können über Texturen verfügen, welche sich auf die Wahrnehmung von diesen Hervorhebungen auswirken können. Aus diesem Grund soll für dieses Projekt eine weitere Hervorhebungsmethode implementiert werden, die stellvertretend texturierte Hervorhebungen repräsentieren soll. Da der Color-Overlay-Effekt über die größte Fläche verfügt und somit die Textur am deutlichsten zur Geltung kommt, wird eine alternative Variante dieser Hervorhebung implementiert, die über eine diagonale Streifentextur verfügt. Demnach soll die entstehende Software zu den vier Hervorhebungstechniken diese fünfte Methode des *Stripe-Effekts* beinhalten. Außerdem soll eine Kontrollmethode beigelegt werden, die die Objekte ohne gesonderte Hervorhebung darstellt.

¹⁰Bildquelle – Screenshot aus Ghostrunner 2 – Spielaufzeichnung von Shirrako via YouTube:
<https://www.youtube.com/watch?v=LxHTRn2d3mY>



(a) Verwendung des Color-Overlay-Effekts in Valorant zum Hervorheben von verbündeten Figuren, die durch die Umgebung verdeckt sind ⁹



(b) Verwendung des Color-Overlay-Effekts in Valorant zum Hervorheben von interagierbaren Gegenständen ⁹



(c) Verwendung des Color-Overlay-Effekts in Overwatch 2 zum Hervorheben von gegnerischen Figuren, welche verdeckt sind und durch eine Fähigkeit aufgedeckt wurden ⁶

Abbildung 3.8: Beispiele von Outlines in Videospielen

4 Implementierung

Dieses Kapitel erläutert, wie die Software zum Testen der Hervorhebungsmethoden entstanden und aufgebaut ist. Es wurde die Unity Version 2022.3.9f1 zur Entwicklung des Programms verwendet.

4.1 System

Überblick

Die Logik der Software wird im Wesentlichen von dem Skript *GuessManager* gesteuert. Es erbt von der Klasse *MonoBehaviour*, welche von Unity bereit gestellt wird, um einfache Funktionalitäten, wie etwa Methoden, die bei jeder Berechnung eines Frames aufgerufen werden, mit niedrigem Aufwand implementieren zu können. Zur Übersicht des Systems wurde ein Flussdiagramm beigelegt (siehe Abb. 4.1). *GuessManager* organisiert den Ablauf der Rateversuche, berechnet die Parameter des jeweiligen Rateversuchs und gibt diese an die entsprechenden Komponenten des Systems weiter. So etwa wird die Hervorhebungsmethode und Objektanzahl für den nächsten Rateversuch randomisiert und an die Skripte *ObjectSpawner* und *LoggingManager* weiter. Der *ObjectSpawner* ist hierbei für die Positionierung der Objekte entsprechend der überreichten Anzahl verantwortlich. Das Skript *GuessUI* ist für die Funktionalität der grafischen Benutzeroberfläche zuständig. Es nimmt Eingaben des Probanden entgegen und gibt weiter. Eingegebene Rateversuche werden an den *LoggingManager* gereicht, um sie in der Log-Datei zu protokollieren. Zudem wird der Start des nächsten Rateversuchs an den *GuessManager* gegeben.

Der *GuessManager* bestimmt eine zufällige Objektanzahl. Dabei ist die Wahrscheinlichkeit der randomisierten Zahl nicht gleich verteilt, sondern so gewichtet, dass niedrige Zahlen häufiger gewählt werden. Diese Gewichtung wurde gewählt, da der gleiche absolute Unterschied zwischen niedrigen Objektzahlen mehr auffällt als bei höheren. Zunächst wurde eine zufällige Gleitkommazahl $0 \leq x \leq 1$ generiert. Diese Zahl wurde anschließend durch folgende Formel in eine Objektzahl $0 \leq N \leq 100$ konvertiert.

$$N = \lceil 70^x + 30x \rceil \quad (4.1)$$

Die Aufrundungsfunktion $\lceil \dots \rceil$ gibt hierbei die kleinstmögliche natürliche Zahl zurück, welche größer als der innere Ausdruck ist. Demnach ergibt sich folgende Verteilung der Wahrscheinlichkeiten

Szenenerstellung

Um Ergebnisse zu erhalten, welche möglichst realistisch und nah an dem tatsächlichen Spielkontext sind, wurde für die Umsetzung ein Spielumgebung entworfen, in dem die Hervorhebungsmethoden verglichen werden sollen. Hierfür wurde eine Szene im Stil eines First-Person-Spiels erstellt, dessen Setting sich am beliebten Genre der Fantasy orientiert. Zu

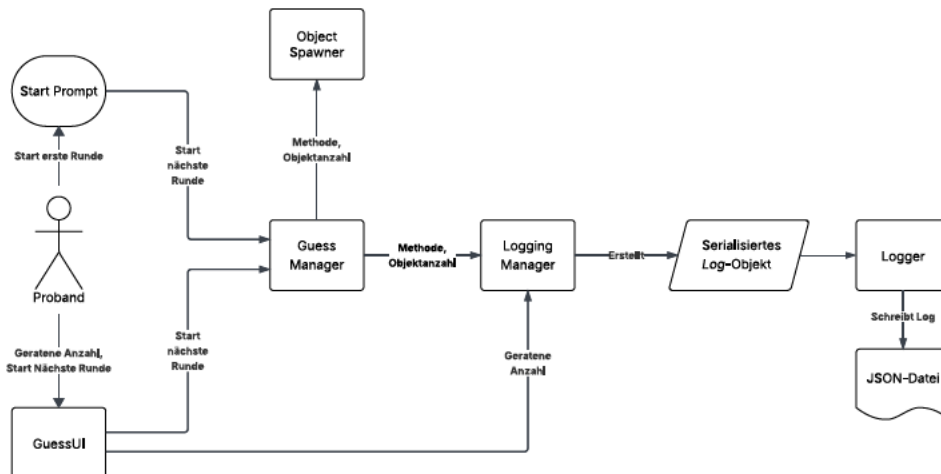


Abbildung 4.1: Flussdiagramm Programmablauf

diesem Zweck wurde ein modulares Asset-Paket benutzt, das bereits texturierte 3D-Modelle zur Verfügung stellt.¹¹ Mit Hilfe dieser Assets wurde eine Umgebung gebaut, welche einen Kerker (*Dungeon*) darstellt und als Kulisse für die eigentliche Evaluation gelten soll (siehe Abb. 4.2a). Zudem wurde die Beleuchtung der Szene so angepasst, um die düstere Umgebung eines von Fackelschein beleuchteten Dungeonlevels nachzuempfinden (siehe Abb. 4.2b). Dies konnte über das *Rendering*-Fenster des Unity Editors bewerkstelligt werden. Zur Navigation wurde eine simple Benutzeroberfläche implementiert, damit die Probanden den ersten bzw. nächsten Schätzversuch starten, die Schätzungen abgeben und das Protokollverzeichnis öffnen können (siehe Abb. 4.3).

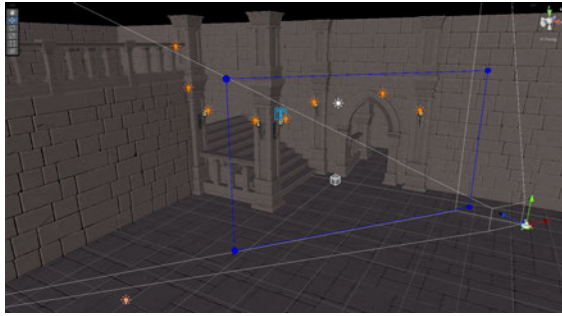
4.2 Objekterstellung

Das hervorzuhebende Objekt des Schlüssels wurde in *Blender* modelliert. Blender ist ein 3D-Grafikprogramm, welches von der Blender Foundation entwickelt wird¹². Es ist Open Source, kostenlos und wird durch Spenden finanziert [Ble]. Deshalb wird es oft von kleineren, unabhängigen Spieleentwicklern (sog. *independent game developers*, oder kurz *indies*) für die Modellierung, Texturierung und Animation von 3D-Objekten benutzt.

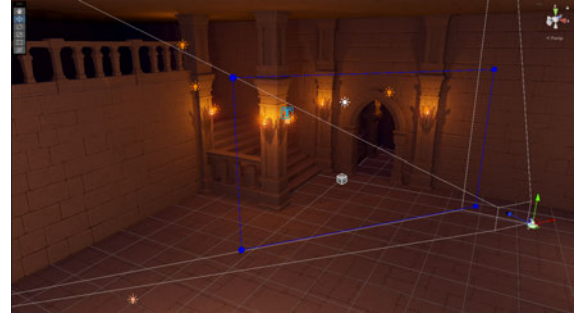
¹¹Benutztes Asset-Paket: *Dungeon Modular Pack* von *logicalbeat* via Unity Asset Store:

<https://assetstore.unity.com/packages/3d/environments/dungeons/dungeon-modular-pack-295430>

¹²Website Blender: <https://www.blender.org/>



(a) Unity-Szene ohne Lighting



(b) Unity-Szene mit Lighting



(c) Beleuchtete Unity-Szene aus Sicht der Spieler-Kamera

Abbildung 4.2: Aufnahmen der Unity-Szene

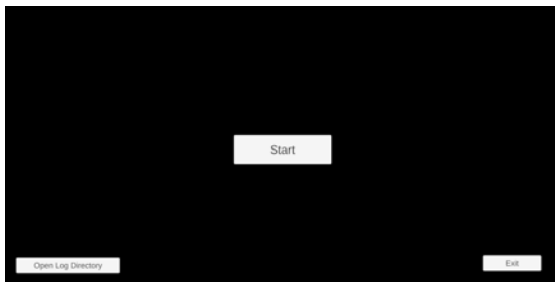
Für diesen Anwendungsfall war es wichtig, dass der Schlüssel auch als ein einzelnes Objekt exportiert wurde. Diese erstellten 3D-Assets können dann mit dem Dateiformat FBX in die Unity Engine übertragen werden.

UV-Mapping und Texturierung

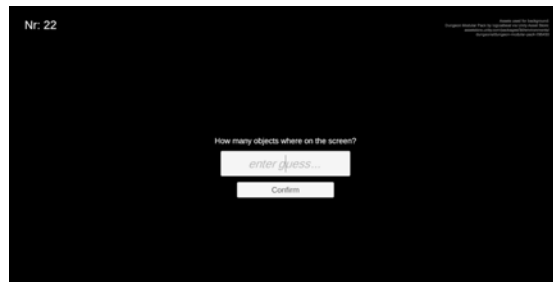
Nachdem das Modell für die Testobjekte erstellt wurde, kann es nun texturiert werden. Dafür muss jedoch erst eine sog. UV-Map erstellt werden, die die Oberfläche des 3D-Objektes auf eine 2D-Fläche projiziert. Hierbei wird jedem Punkt der Polygone des Meshes ein Punkt auf einem zweidimensionalen Koordinatensystem zugewiesen. Um Verwechslung mit dem ursprünglichen Vektorraum des 3D-Modells, heißen die namensgebenden Koordinaten U und V. Ergebnis dieser Zuweisung ist die sogenannte UV-Map.

Die Funktionalität zur Erstellung einer UV-Map ist meist fester Bestandteil der 3D-Modellierungssoftware. In diesem Beispiel wurde für die Erstellung des Testobjekts mit Blender die UV-Map erstellt.

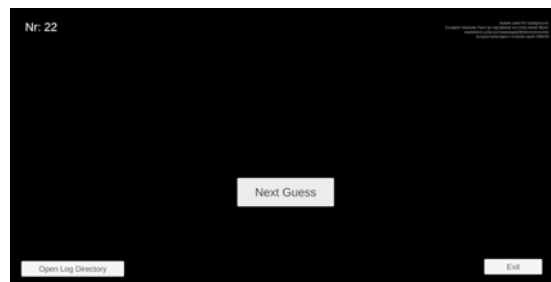
Mithilfe der UV-Abbildung kann nun eine oder mehrere Texturen des 3D-Objekts mit einem beliebigen Bildbearbeitungsprogramm erstellt werden.



(a) Startmenü, welches sich nach dem Ausführen der Software öffnet



(b) Eingabefeld mit Abfrage zum Schätzversuch



(c) Menü nach Eingabe eines Schätzversuchs

Abbildung 4.3: Benutzeroberflächenelemente und Menüs

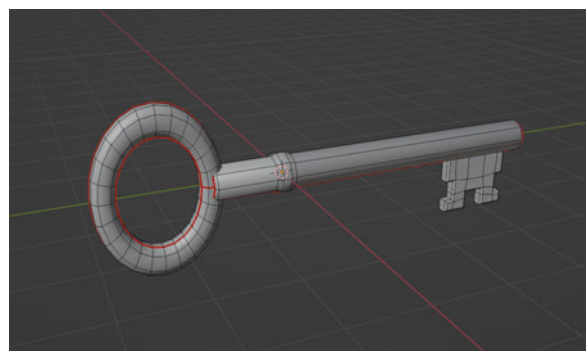


Abbildung 4.4: Modellierter Schlüssel in Blender

Export und Import

Das 3D-Modell ist nun fertig modelliert und hat auch entsprechende Texturen erhalten. Es muss nun aus dem Modellierungsprogramm in das Unity-Projekt importiert werden. Dazu wird zuerst das Modell als FBX-Datei exportiert. FBX ist ein Dateiformat, welches 2006 von Kaydara entwickelt wurde und ist derzeit eines der gängigen Formate, um 3D-Modelle inklusive Animationen und Texturen zwischen verschiedenen Programmen auszutauschen.

Um die Schlüssel dem Kontext eines Videospiele getreu darzustellen, musste Ihnen noch ein Material gegeben werden. Für diesen Anwendungszweck wurde ein goldenes Material mit dem Shader Graph erstellt. Es wurde eine Noise Texture (Rauschfunktion) benutzt, um die Unregelmäßigkeit von abgenutztem Metall nachzuempfinden. Diese Noise Textur wurde in eine Normal Map umgewandelt, um jene Unebenheiten auf die Oberfläche zu übertragen. Normal Maps funktionieren, in dem Sie die Normalvektoren für jeden Pixel der Fläche neu ausrichten.



Abbildung 4.5: Schlüssel mit goldenem Material in Unity-Szene

4.3 Hervorhebungsmethoden

Farbe

Um die Hervorhebungsmethoden als Antezedenzvariable möglichst zu isolieren, wurde für jede Methode die gleiche Hervorhebungsfarbe gewählt. Bei der Einfärbung der Hervorhebung wurde darauf geachtet, mit Kontrasten zu arbeiten. Die Hervorhebung sollte sich stark von dem dunklen, orangenen Hintergrund der Szene abheben. Deshalb wurde für die Hervorhebungsfarbe ein helles Blau gewählt, um sich sowohl Komplementärkontrast, einen Hell-Dunkel-Kontrast als auch einen Kalt-Warm-Kontrast zunutze zu machen.

Fläche

Zudem wurde darauf geachtet, dass die Hervorhebungsmethoden einen ähnlichen Anteil des Bildes einnehmen. So soll die Summe der Deckkraftwerte der eingefärbten Pixel pro Objekt im Mittel gleich sein. Das heißt, dass beispielsweise die Anzahl der Pixel der Outline verglichen mit dem Color-Overlay-Effekte dem Flächeninhalt des Objektes multipliziert mit der Deckkraft des Effektes entsprechen soll. Da das Verhältnis von Flächeninhalt des Abbilds des Objektes und Länge seiner Kontur nicht konstant ist und mitunter von der Orientierung des Objektes abhängt, kann dies nicht immer ohne erheblichen Mehraufwand garantiert werden. Deswegen soll es in diesem Projekt ausreichend sein, dass die summierte Deckkraft der Hervorhebungsmethoden annähernd die Gleiche ist.

Unity-Integration

Die Hervorhebungsmethoden wurden in Unity mit je einem *Renderer Feature* umgesetzt, welches die hervorzuhebenden Objekte mit einem spezifischen Shader in einem zusätzlichen Durchlauf (*Render Pass*) rendert. Wenn nun dem jeweiligen *Renderer Feature* mitgeteilt wird, dass er nur Objekte rendern soll, die sich auf diesem Layer befinden, können diese Objekte mit der gewünschten Hervorhebungsmethode versehen werden.

Im Folgenden soll sich nun damit befassen werden, wie die einzelnen Hervorhebungsmethoden als Shader umgesetzt wurden, damit sie vom Renderer Feature dargestellt werden können.

Outline-Methoden

Es gibt verschiedene Weisen, um 3D-Objekte in Spielen mit einer Kontur zu versehen [Gol20]. Diese sollen hier kurz umrissen werden.

Zunächst können Manipulationen am 3D-Objekt, beispielsweise über Vertex-Shader, eingesetzt werden. Ein prominenter Vertreter dieser Variante ist die *Inverted Hull Method* (Methode der gestülpten Hülle). Hierbei wird das Mesh des Objektes dupliziert und umgestülpt, sodass die Normalen des Polygone nach innen zeigen. Im nächsten Schritt wird dieses Mesh vergrößert, indem die Vertices entlang der Normalen nach außen bewegt werden. In den meisten Fällen wird in Videospielen das Polygon nur der Seite gerendert, um nicht sichtbare Flächen zu entfernen und so die Zahl der zu rendernden Polygone zu verringern. Dieses Verfahren wird *Backface Culling* genannt und wird auch bei der Inverted Hull Methode eingesetzt. Beim Backface Culling wird der Winkel zwischen dem Normalenvektor des Polygons und dem Vektor zum Betrachtungspunkt bemessen. Ist dieser Winkel größer als 90° , so ist die Fläche nicht zu sehen und das Polygon wird nicht dargestellt [RS]. Da die Normalvektoren des invertierten Meshs nach innen zeigen, werden nur die Polygone hinter dem hervorzuhebenden Objekt angezeigt. Wenn das vergrößerte Mesh nun in der Outline-Farbe dargestellt wird, entsteht so eine Outline des Objektes. Diese Variante ist ziemlich effizient, hat jedoch die Schwäche, dass Spitzen und Kanten des 3D-Modells Artefakte und Lücken in der Outline entstehen lassen (siehe Abb. 4.6).



Abbildung 4.6: Beispiel Inverted-Hull-Methode¹³

¹³Bildquelle: [Gol20]

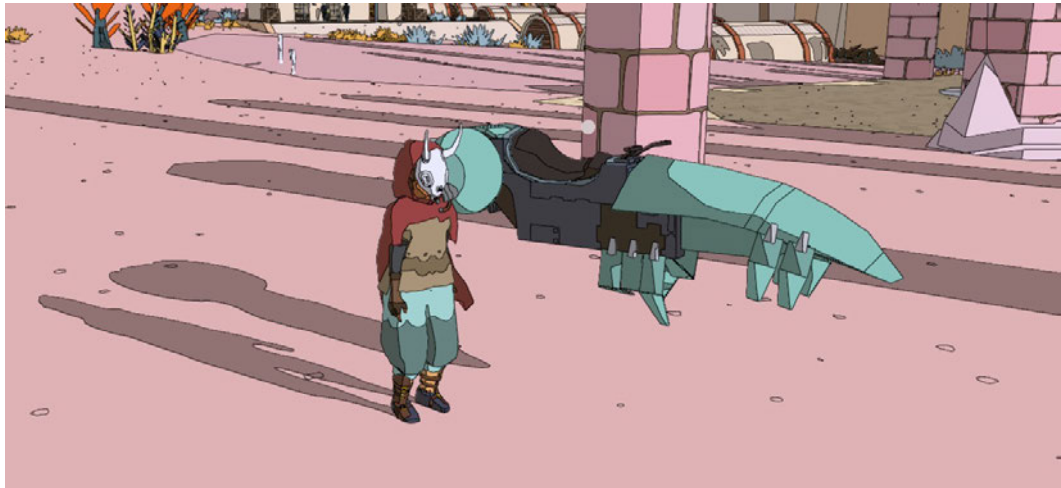


Abbildung 4.7: Linieneffekt via Kantenerkennung im Spiel Sable von Shedworks

Eine weitere Variante stellen Post-Processing-Effekte dar, die das gesamte Bild durchlaufen, in diesem Diskontinuitäten und Kanten erkennen und diese kennzeichnen. Dabei kann neben der eigentlichen Farbe des Bildes auch der sogenannte Depth-Buffer verwendet werden, der die Entfernung eines jeden Bildpunktes zur Kamera, also die Tiefe, markiert. Genauso kann der Normal-Buffer integriert werden, der die Normalen der jeweilig betrachteten Polygone zurückgibt. Die Kantenerkennung geschieht meist über den Gradienten des Bildes, der den Unterschied der Farbwerte der Pixel zu ihren benachbarten Pixeln darstellt. Derartige Algorithmen wie z.B. der Roberts-Cross-Algorithmus oder auch der Sobel-Algorithmus können in ihrem benötigten Rechenaufwand unterschiedlich anspruchsvoll sein, können aber mitunter zu interessanten visuellen Effekten beitragen, die an die Zeichnungen des Künstlers und Comiczeichners Jean Giraud ("*Moebius*") erinnern (siehe Abb. 4.7). Da bei diesen Effekten häufig nicht nur Konturen mit Linien versehen werden, sondern auch andere Linien entstehen, beispielsweise innerhalb der Objekte, soll sich hier auf andere Outline-Methoden konzentriert werden.

Zuletzt können die gerenderten Abbilder der Objekte mit Fragment-Shadern bearbeitet werden.

Für den Zweck dieser Arbeit wurde sich für die Methode des eines Shaders unter Benutzung des Jump-Flood-Algorithmus (JFA) entschieden. Dieser liefert einerseits die robustesten Ergebnisse, ohne eine reelle Gefahr, dass ungewollte Artefakte, wie etwa bei der Inverted-Hull-Methode auftreten. Zwar berechnet der JFA nicht immer das korrekte Ergebnis, sondern nur eine näherungsweise Approximation dessen, jedoch sind die Fehler in ihrem Ausmaß in der Praxis zu vernachlässigen. Auch ist er vom Rechenaufwand vor allem im Vergleich mit anderen Fragment-Shadern sowie mit Post-Processing-Effekten sehr effizient, da er sich die parallel ablaufenden Rechenprozesse der GPU und zunutze machen kann [Gol20].

Für den Jump-Flood-Algorithmus wird ein zweidimensionales Raster mit der Größe $n \times n$ angenommen, in dem sich eine Menge von *seeds points* (Saatpunkten) befinden, die jeweils einen zugewiesenen Datenpunkt, wie z.B. eine Farbe haben. Für jeden Punkt bzw. Pixel dieses Rasters soll nun jeweils ermittelt werden, zu welchem dieser Seeds er sich am nächsten befindet und die Daten jenes Seeds übernommen werden.

Eine mögliche Lösung dieses Problems wäre der Floodfill, bei jeder Pixel des Rasters seinen Wert an seine direkten Nachbarn weitergibt, der nun selbiges tut, bis Pixel mit unterschiedlichen Seeds aufeinandertreffen. Diese Lösung mit der Propagation um 1 kann jedoch nur seriell ausgeführt werden, da die für jeden Pixel erst der Wert eines Nachbarn bestimmt werden muss. [FB85] Beim JFA wird nun, anstelle des Propagieren um 1, Schritte gewählt, die logarithmisch skaliert werden. Mit jedem Durchlauf mit der Schrittlänge $k \in \{\frac{n}{2}, \frac{n}{4}, \dots, 1\}$ werden für jeden Seedpunkt mit den Koordinaten (x, y) die Punkte mit den Koordinaten $(x + i, y + j)$ betrachtet, wobei $i, j \in \{-k, 0, k\}$ gilt. Für jeden dieser maximal acht Punkte mit den Koordinaten (x', y') wird nun überprüft, ob dieser schon einen Seed zugeteilt bekommen hat. Sollte dies nicht der Fall sein, wird diesem Punkt der Seed s zugewiesen und der Punkt wird selbst zum Seedpunkt. Wenn dieser Punkt schon einen zugewiesenen Seed s' hat, wird für diesen Punkt überprüft, welche der Distanzen zu den Ursprungspunkten von s und s' geringer ist und der zugewiesene Seed dementsprechend gegebenenfalls aktualisiert. Mit jedem Durchlauf kann sich der zugewiesene Seed eines jeden Punktes mehrfach ändern. Nach $\log n$ Durchläufen ist das Raster gefüllt und jedem Punkt die Distanz zu seinem nächsten Seedpunkt zugewiesen. [RT06]

Da für jeden betrachteten Punkt die Distanz zu den Seedpunkten errechnet wird, kann damit nun eine Distanztransformation oder ein Distanzfeld erstellt werden. Diese gibt an, welchen Abstand ein jeder Pixel zu einer bestimmten Fläche hat, welche in diesem Kontext das zweidimensionale Abbild des zu konturierenden Objektes ist. Im Anwendungsfall der Outline wird nun eine Distanztransformation erstellt, welche den Abstand r eines jeden betrachteten Pixel außerhalb des hervorzuhebenden Objektes zu dem je nächsten Pixel der Silhouette des Objektes angibt. Anhand dieser kann dann jeder Pixel mit $r \leq d$ eingefärbt werden, wobei d die gewünschte Dicke der Outline angibt. Da nur Pixel betrachtet werden müssen, die eine Distanz r kleiner oder gleich der Outlinedicke besitzen, muss der JFA auch kein Distanzfeld für das gesamte Bild berechnen. So muss die größte Schrittlänge k , mit der der JFA arbeitet, nur größer sein als die Outlinedicke d . Dementsprechend benötigt der JFA $\lceil \log_2 d \rceil$ Durchläufe. Dieser logarithmische Zusammenhang ermöglicht es, auch enorm dicke Outlines ohne erheblichen rechnerischen Mehraufwand zu generieren.

Die Outline mit Jump-Flood-Algorithm wurde in Unity mit einem *Renderer Feature*¹⁴ umgesetzt, welches die hervorzuhebenden Objekte in einem zusätzlichen Durchlauf (*Render Pass*) mit dem gewünschten Shader rendert. Der Code für Shader und Renderer Feature selbst basiert auf dem URP Outline Renderer Feature von Feral Pug¹⁵.

Glow

Der Glow-Effekt basiert auf einer ähnlichen wie die Outline. Hier wird ebenfalls der Jump-Flood-Algorithmus benutzt, um eine Distanztransformation für die Silhouette des Objektes zu erstellen. Ihre Distanzwerte können dann verwendet werden, um einen graduellen Übergang zu erhalten. Hierfür wird die folgende Funktion verwendet, die Deckkraftwerte a eines Pixels in Abhängigkeit des Distanzwertes r und der gewünschten Breite des Glow-Effektes d angibt:

¹⁴Unity-Dokumentation zu Renderer Features: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@10.1/manual/urp-renderer-feature.html>

¹⁵Implementation einer Outline mit JFA in Unity von Feral Pug via GitHub: <https://github.com/FeralPug/URP-Outline-Render-Feature>

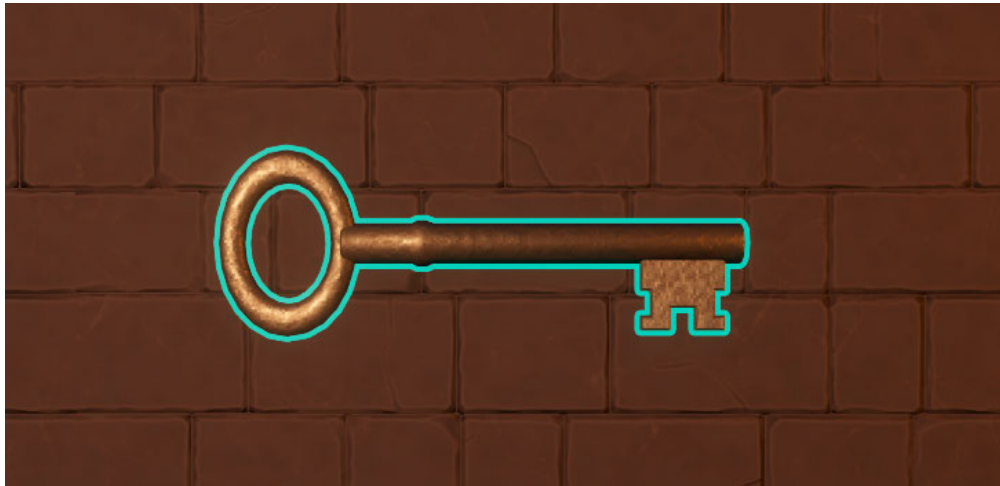


Abbildung 4.8: Schlüssel mit Outline

$$a = \text{saturnate} \left[1 - \left(\frac{r}{d+1} \right) \right] \quad (4.2)$$

Dabei wird die Sättigungsfunktion *saturnate* verwendet, die alle Werte größer als 1 oder kleiner als 0 auf diese Grenzen des Intervalls rundet. Demnach werden Pixel, welche sich näher an dem Objekt befinden, und eine dementsprechend geringe Distanz r aufweisen, auch mit einer höheren Deckkraft a versehen, als Pixel höheren Distanzwerten, also größerem Abstand zum Objekt. Die Deckkraft a fällt auf 0, wenn der Bruch 1 erreicht, wenn also die Distanz vom Objekt größer ist als die gewünschte Breite des Glow-Effektes.

So werden aus der Distanztransformation die Alphawerte, die für den weichen Gradienten von innen nach außen des Glow-Effektes gewünscht werden.



Abbildung 4.9: Schlüssel mit Glow-Effekt

Fresnel-Effekt

Für die restlichen Hervorhebungsmethoden wurde die Umgebung *Shader Graph* von Unity benutzt, ein visuelles Coding-Werkzeug der Unity Engine, welche es dem Nutzer erlaubt, Shader auch ohne das Schreiben von Shader-Code zu erstellen. Es funktioniert über das Verbinden von Knotenpunkten (sog. Nodes), um die Ausgabe einer Node in die Eingabe einer anderen zu geben.

Die Unity Engine stellt bereits eine Fresnel-Node bereit, was den Aufwand für die Erstellung dieses Effekts erheblich vereinfacht¹⁶. Die Fresnel-Node von Unity gibt für den eindimensionalen Farbwert c eines Pixels folgenden Zusammenhang aus:

$$c = \left\{ 1 - \text{saturnate}(\vec{n}_n \cdot \vec{v}_n) \right\}^p \quad (4.3)$$

Der Vektor $\vec{n}_n$ beschreibt hierbei die Normale des Polygons, welches betrachtet wird. $\vec{v}_n$ ist die Blickrichtung, mit der auf dieses Polygon geschaut wird. Beide Vektoren sind normiert, sie sind also so skalar multipliziert, dass ihre Beträge gleich 1 sind: $|\vec{n}_n| = |\vec{v}_n| = 1$

Der Exponent p beschreibt die Stärke (*power*) mit der dieser Effekt wirken soll. Je größer p desto mehr konzentriert sich der Fresnel-Effekt auf die spitzeren Winkel. Anhand der Formel der Fresnel-Node von Unity kann erkannt werden, dass die Stärke des Farbwertes c am größten ist, wenn die Vektoren $\vec{n}_n$ und $\vec{v}_n$ möglichst orthogonal zueinander sind, also das betrachtete Polygon von einem möglichst spitzen Winkel aus betrachtet wird.

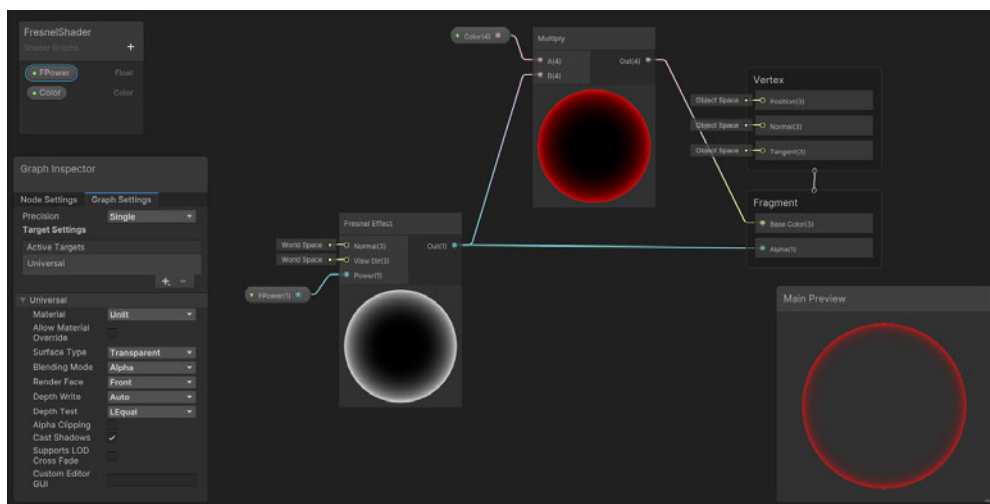
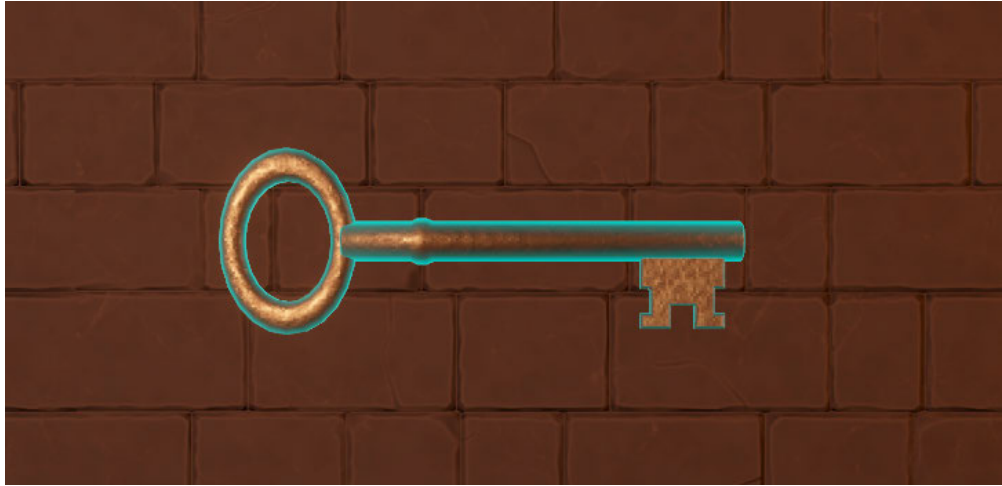
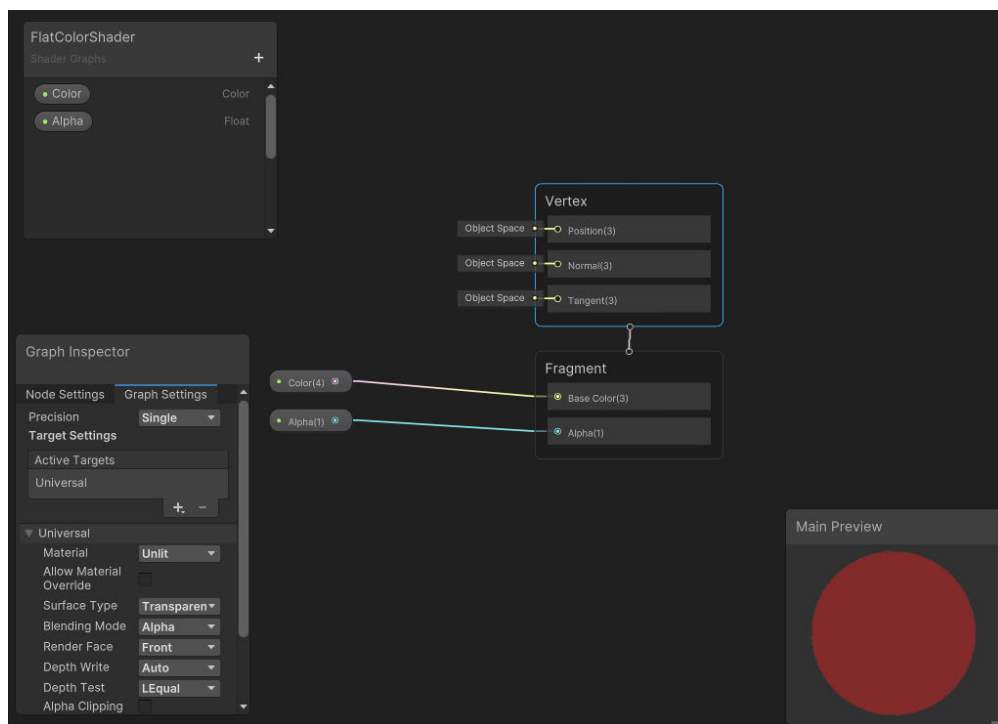


Abbildung 4.10: Screenshot aus dem Shader Graph des Fresnel-Shaders

Color Overlay

Die Erstellung des Color-Overlay-Effekts ist vergleichsweise simpel, da nur eine Farbe mit geringer Deckkraft über die Maske des Objekts gelegt werden muss. Im Shader Graph wird die Farbe mit der gewünschten Transparenz ausgegeben (siehe Abb. 4.12).

¹⁶Unity-Dokumentation zur Fresnel-Node des Shader Graphs: <https://docs.unity3d.com/Packages/com.unity.shadergraph@6.9/manual/Fresnel-Effect-Node.html>

**Abbildung 4.11:** Schlüssel mit Fresnel-Effekt**Abbildung 4.12:** Shader Graph für Shader des Color-Overlay-Effekts

Stripes

Zusätzlich wurde noch eine weitere Hervorhebungsmethode implementiert. Diese stellt eine texturierte Alternative zum Color-Overlay-Effekt dar. Im Vergleich mit der Farbüberlagerung soll hiermit Für den Stripe-Effekt wurde ebenfalls der Shader-Graph benutzt. Hier wurde die Rectangle-Node von Unity benutzt, um einen Streifen zu erstellen. Danach wurde dieser um 45° rotiert, um die schräge Schraffur zu kreieren. Anschließend konnte die Fraction Node

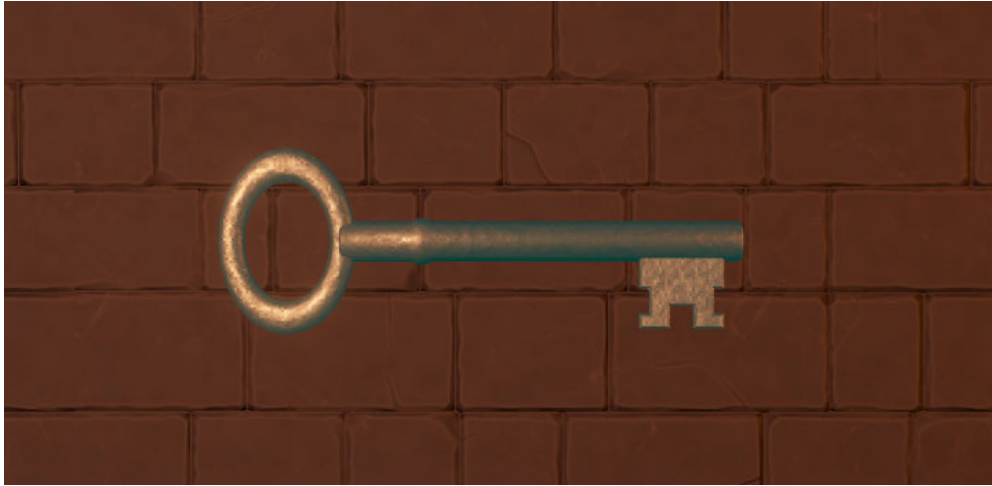
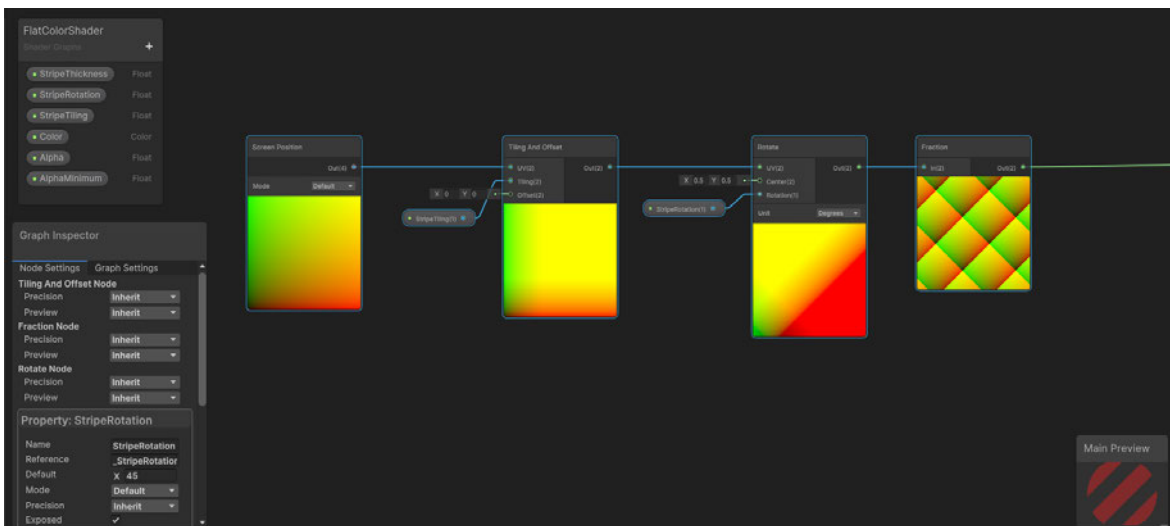


Abbildung 4.13: Schlüssel mit Color-Overlay-Effekt

eingesetzt werden, die jeweils die Vorkommastelle der UV-Werte aus dem Screen Space streicht (siehe Abb. 4.14a) und somit nur die Nachkomastelle erhält. Somit lässt sich die Repetition der Streifentextur erzeugen (siehe Abb. 4.15).



(a) Teil 1: Fraction-Node



(b) Teil 2: Streifenenerstellung

Abbildung 4.14: ShaderGraph für Stripe-Effekt

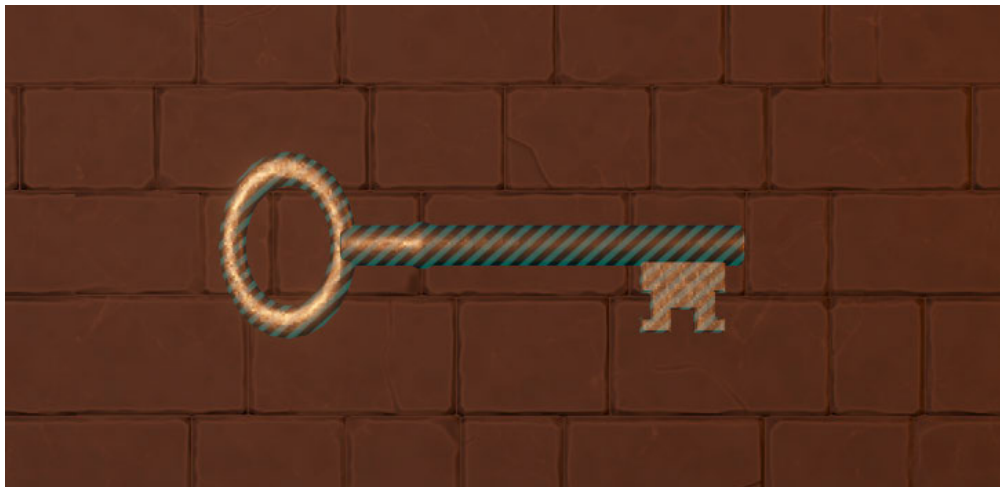


Abbildung 4.15: Schlüssel mit Stripe-Effekt

4.4 Logging

Um Daten während der Evaluation auch automatisch ohne manuelles Zutun erfassen zu können, wurde eine Protokollfunktion implementiert. Diese protokolliert mit jeder Schätzung die Nummer des Versuchs, die geratene Anzahl, die tatsächliche Anzahl und die entsprechende Highlighting-Methode in einer separaten sog. *Log-Datei*. Hierfür wurde das Dateiformat JavaScript Object Notation (JSON) gewählt. Dies ist ein gängiges Dateiformat für den kompakten Austausch und die Übertragung von strukturierten Daten [Ecm17]. Es kann sowohl von Menschen als auch maschinell geschrieben und gelesen werden. Dabei speichert es serialisierte Daten als Objekte ab, die verschachtelt werden können. Diese Objekte bestehen aus Paaren, welche wiederum jeweils aus einem *Schlüssel* und einem *Wert* bestehen. Diese Schlüssel-Wert-Paare werden wiederum durch Kommas getrennt (siehe Abb. 4.16).

```
1 {  
2   "fruit": "Apple",  
3   "size": "Large",  
4   "color": "Red"  
5 }  
6  
7
```

Abbildung 4.16: Beispiel für ein simples Datenkonstrukt im JSON-Dateiformat

Erstellen der Log-Dateien

Die Log-Dateien bestehen aus einer geordnete Liste (*Array*) von Log-Objekten. Für jeden Schätzversuch wird ein neues Log-Objekt erstellt. Dieses besitzt je ein Schlüssel-Wert-Paar für die Nummer des Schätzversuchs, die geratene Anzahl von Objekten, die tatsächlich an-

gezeigte Anzahl von Objekten und die Hervorhebungsmethode, die für die Objekte genutzt wurde. Die Log-Dateien werden in einem Ordner gespeichert, den Unity bereitstellt unter

```
1  { "logs": [
2    {
3      "number": 1,
4      "method": 1,
5      "objectCount": 55,
6      "guessedNumber": 26
7    },
8    {
9      "number": 2,
10     "method": 1,
11     "objectCount": 38,
12     "guessedNumber": 31
13   },
14   {
15     "number": 3,
16     "method": 5,
17     "objectCount": 16,
18     "guessedNumber": 15
19   },
20   {
```

Abbildung 4.17: Beispiele für einzelne Logs aus den Protokollen

Application.persistentDataPath. Dieser Ordner ermöglicht es, Dateien auch zwischen verschiedenen Instanzen zu speichern. Dabei werden die Dateien mit dem Erstellungsdatum im Namen nach dem Schema *BA-LogFile_YYYY-MM-DD-hh-mm-ss.json* abgespeichert, um sie später wieder identifizieren zu können.

Auslesen der Logging-Dateien

JSON-Dateien lassen sich von Maschinen parsen und können so auch automatisch in andere verwertbare Dateiformate konvertiert werden. Um die Daten auszuwerten, wurden sie in Tabellenkalkulationssoftware Google Slides importiert und somit an eine übergreifende Instanz übergeben, in der die Daten ausgewertet werden können.

5 Evaluation

Das folgende Kapitel wird sich nun mit der entstandenen Software und der durchgeführten Evaluation beschäftigen. Das Versuchsszenario wird erläutert

5.1 Teilnehmende

An der Evaluation nahmen 10 Probanden teil. Diese fanden sich alle im Altersbereich 21-25 wieder und bei keinem der Teilnehmenden waren Farbblindheiten oder Sehbehinderungen bekannt, die das Ergebnis der Evaluation beeinträchtigen könnten. Jeder der Probanden war laut Umfrage mit dem Medium des Videospieles vertraut und spielt jede Woche im Durchschnitt mehrere Stunden. Dabei gaben 3 der Probanden an, dass sie 5-10 Stunden pro Woche spielen. 6 gaben an, dass sie im Mittel zwischen 10 und 20 Stunden pro Woche spielen. Ein weiterer gab an, dass er mehr als 30 Stunden pro Woche mit dem Spielen von Videospielen verbringt. Das Geschlecht der Teilnehmenden war überwiegend männlich. So gaben 7 der 10 Probanden in der Umfrage ihr Geschlecht als männlich an, 2 als weiblich und ein Teilnehmender als divers.

5.2 Beschreibung des Experiments

Die Software wurde an die Teilnehmenden über einen Link zum Herunterladen zur Verfügung gestellt. Dieser Link war aus Gründen der Einfachheit in einer Umfrage enthalten, welche mit Tally¹⁷ erstellt wurde und Anweisungen zum Installieren und Ausführen des Programms enthielt. Nachdem Herunterladen und Ausführen des Programms öffnet sich dem Probanden ein Menü mit Buttons zum Starten, zum Öffnen des Ordners mit den Log-Dateien und zum Beenden des Programms (siehe Abb. 4.3a). Die Probanden wurden aufgefordert, auf Start zu klicken. Danach verschwindet das Menü und die erstellte Dungeon-Szene wird sichtbar. In ihr befinden sich eine zufällige Anzahl von Schlüsseln, die mit einer zufällig ausgewählten Hervorhebungsmethode hervorgehoben sind (siehe Abb. 5.1). Zwei Sekunden nach dem Sichtbarwerden der Szene, erscheint vor ihr eine schwarze Blende und es öffnet sich erneut ein Menü mit einem Eingabefeld und der Frage, wie viele Schlüssel auf dem Bildschirm zu sehen waren (siehe Abb. 4.3b). Nach Eingabe der Antwort verschwindet das Eingabefeld und es erscheinen drei Buttons zum Starten des nächsten Versuchs, zum Öffnen des Ordners mit den Log-Dateien und zum Beenden des Programms (siehe Abb. 4.3c). In der linken oberen Ecke wird die Anzahl der getätigten Versuche angezeigt.

5.3 Ergebnisse

Bei der Evaluation werden Daten aus zwei verschiedenen Quellen erhoben. Erstens werden während der Ausführung der Software die Logs aufgezeichnet, welche für jeden Schätzversuch entsprechende Werte wie Nummer des Versuchs, geschätzte Anzahl der Objekte, tatsächliche Anzahl der gezeigten Objekte und die Hervorhebungsmethode festhält. Zusätzlich füllten die Probanden im Anschluss eine Umfrage aus.

¹⁷Tally-Webseite: <https://tally.so/>



(a) Schätzversuch mit $N = 5$, Darstellung ohne Hervorhebung



(b) Schätzversuch mit $N = 3$, hervorgehoben durch die Stripes-Methode



(c) Schätzversuch mit $N = 26$, hervorgehoben durch die Glow-Methode



(d) Schätzversuch mit $N = 84$, hervorgehoben durch die Outline-Methode

Abbildung 5.1: Beispiele von Schätzversuchen mit verschiedenen Objektzahlen N und verschiedenen Hervorhebungsmethoden

Fragebogen

Zusätzlich dazu wurde von den Probanden je eine Einschätzung der Erkennbarkeit der einzelnen Hervorhebungsmethoden auf einer Skala von 1 (sehr schlecht) bis 5 (sehr gut) eingeholt.

Am Ende dieses Fragebogens sollte dann die erstellte Logging-Datei mit hochgeladen werden, sodass deren Daten trotz Anonymität den jeweiligen Antwortsätzen aus den Fragebögen zugeordnet werden konnten.

Wie in Abb. 5.2 zu erkennen ist, wurde die Outline vielfach als am besten erkennbar gewertet. Auch die Glow- und Fresnel-Effekte sind bei der subjektiven Einschätzung der Probanden als gut erkennbar bewertet worden, wobei die Umfrageergebnisse für den Fresnel-Effekt durchwachsener sind. Sowohl Stripe-Effekt und Color-Overlay-Effekt wurden als schlechter erkennbar erachtet und wurden, besonders im Falle des Color-Overlays, als nur minimal besser als die Erkennbarkeit ohne Hervorhebung eingeschätzt.

Protokolle

Wie im Kapitel Implementierung unter 4.4 ausgeführt wurde zum Aufzeichnen der Schätzungen eine Protokoll-Funktion implementiert, die die Eckdaten eines jeden Schätzversuches abspeichert.

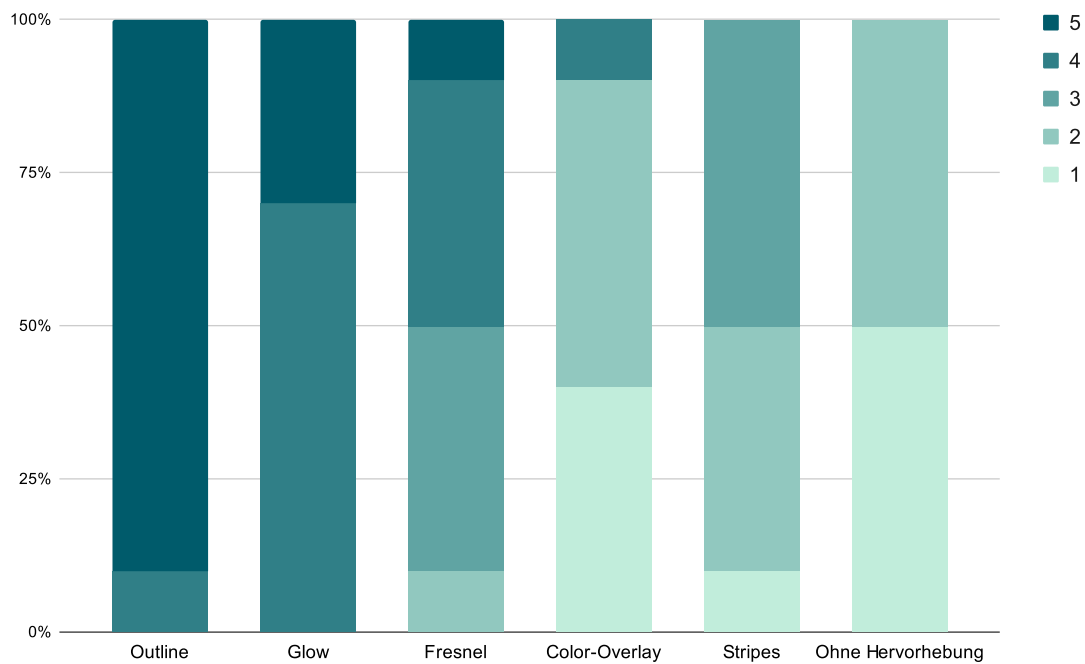


Abbildung 5.2: Ergebnisse der Umfrage zur Erkennbarkeit der Hervorhebungsmethoden; 5 = sehr gut, 1 = sehr schlecht

Um die Methoden miteinander zu vergleichen wurden die Daten auf verschiedenen Weisen ausgewertet. Zunächst wurde die Differenz d eines jeden abgegebenen Schätzversuchs zu der Anzahl der tatsächlich angezeigten Objekte gebildet. Diese wird im Zusammenhang mit der zugehörigen Objektanzahl N in den folgenden Diagrammen nach Methode aufgeteilt abgebildet.

Zunächst lässt sich aus den Diagrammen wie erwartet erkennen, dass die Beträge der Differenzen $|d|$ mit steigender gezeigter Objektzahl N pauschal zunehmen. Das heißt, die Schätzversuche werden mit steigender Objektzahl N ungenauer. Dabei sind die Differenzen mehrheitlich negativ, was bedeutet, dass die Objektzahlen meist unterschätzt statt überschätzt wurden. Dies lässt sich besonders gut am Fresnel- und Color-Overlay-Effekt erkennen, die fast ganzheitlich negative Differenzen oder allenfalls nur sehr kleine positive Differenzen aufweisen. Auch in den Kontrollversuchen ohne explizite Hervorhebung spiegelt sich dieser Sachverhalt wider, wo die Mehrheit der Schätzungen unter der tatsächlichen Objektzahl lag. Daraus könnte geschlossen werden, dass bei hohen Objektzahlen unabhängig von Darstellungsmethode bzw. Hervorhebung die schiere Menge von wahrzunehmenden Objekten dazu führt, dass die menschliche Wahrnehmung die Anzahl mehr und mehr unterschätzt.

Bei niedrigen Objektzahlen N ist im Gegensatz zu beobachten, dass Schätzversuche mit $N \leq 6$ in 97,8% der Fälle eine Differenz $d = 0$ aufwiesen, also akkurat waren. Nur zwei Schätzversuche in diesem Bereich waren falsch. Dies bestätigt, dass sich niedrige Objektzahlen simultan erfassen lassen und somit in der kurzen Zeit akkurat eingeschätzt werden können. Darüber hinaus können Objektzahlen im Bereich 7-10 immer noch recht akkurat eingeschätzt werden. Die Ungenauigkeit ist hier schon größer, jedoch sind diese meist sehr klein. So machen Schätzversuche mit $|d| \leq 1$ 98% der Versuche mit $N \leq 10$ aus.

Ohne Hervorhebung

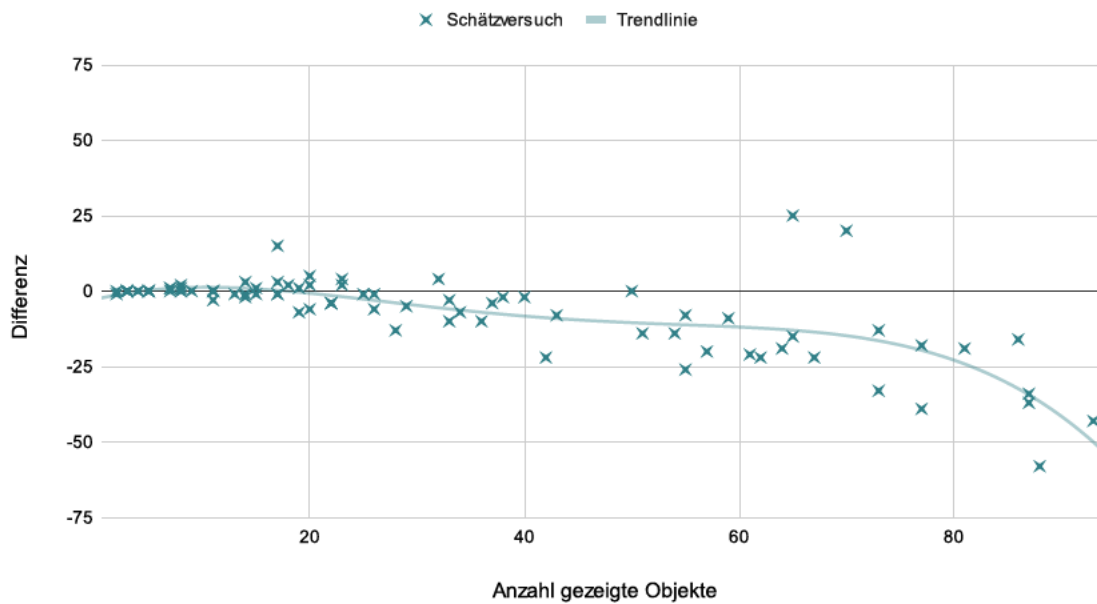


Abbildung 5.3: Schätzversuche ohne Hervorhebung, Differenz der eingegebenen Schätzungen zu tatsächlich gezeigter Anzahl von Objekten, Polynomiale Trendlinie 4. Grades

Hervorhebung durch Outline

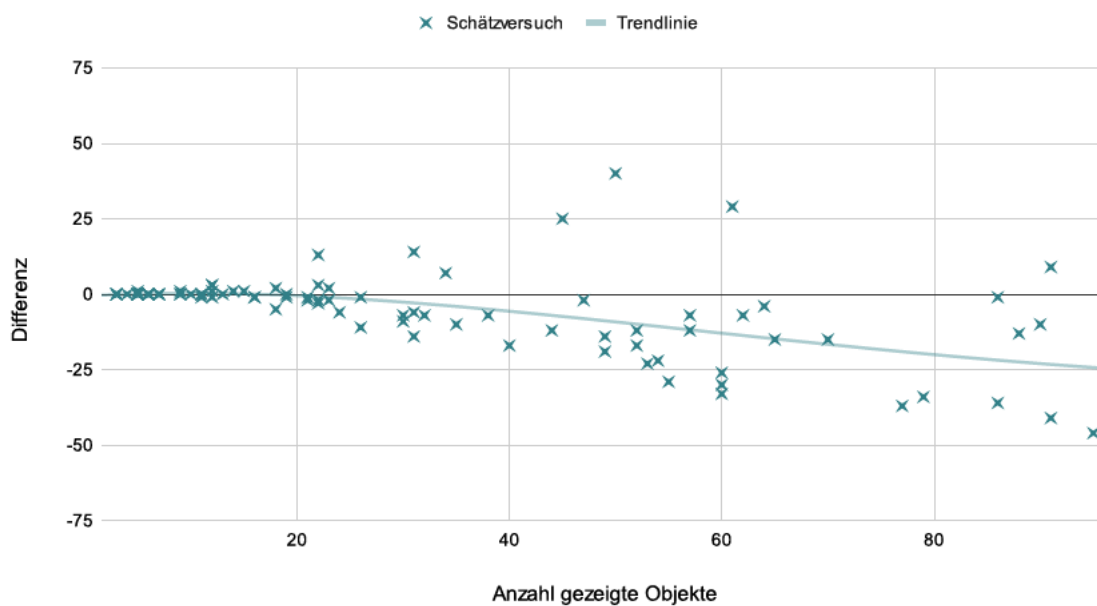


Abbildung 5.4: Schätzversuche bei Hervorhebung durch Outline, Differenz der eingegebenen Schätzungen zu tatsächlich gezeigter Anzahl von Objekten, Polynomiale Trendlinie 4. Grades

Hervorhebung durch Glow-Effekt

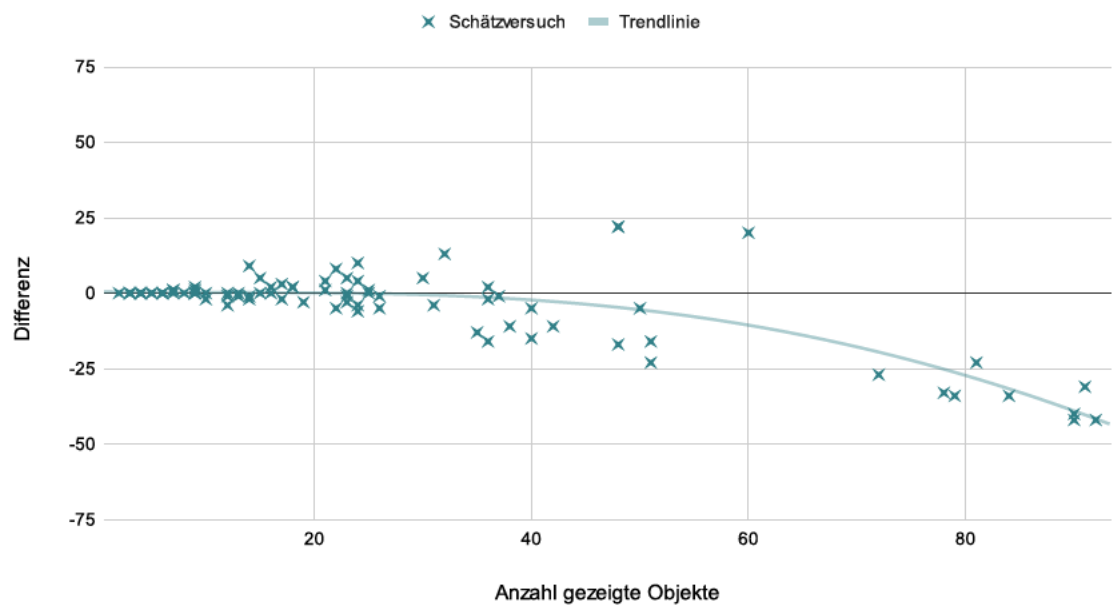


Abbildung 5.5: Schätzversuche bei Hervorhebung durch Glow-Effekt, Differenz der eingegebenen Schätzungen zu tatsächlich gezeigter Anzahl von Objekten, Polynomiale Trendlinie 4. Grades

Hervorhebung durch Fresnel-Effekt

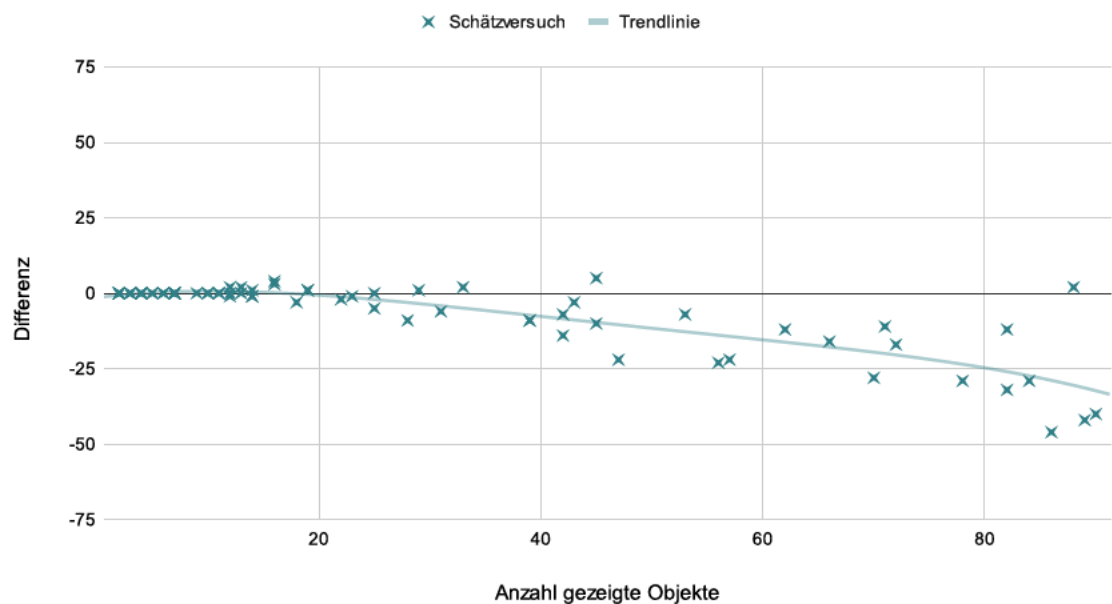


Abbildung 5.6: Schätzversuche bei Hervorhebung durch Fresnel-Effekt, Differenz der eingegebenen Schätzungen zu tatsächlich gezeigter Anzahl von Objekten, Polynomiale Trendlinie 4. Grades

Hervorhebung durch Color-Overlay-Effekt

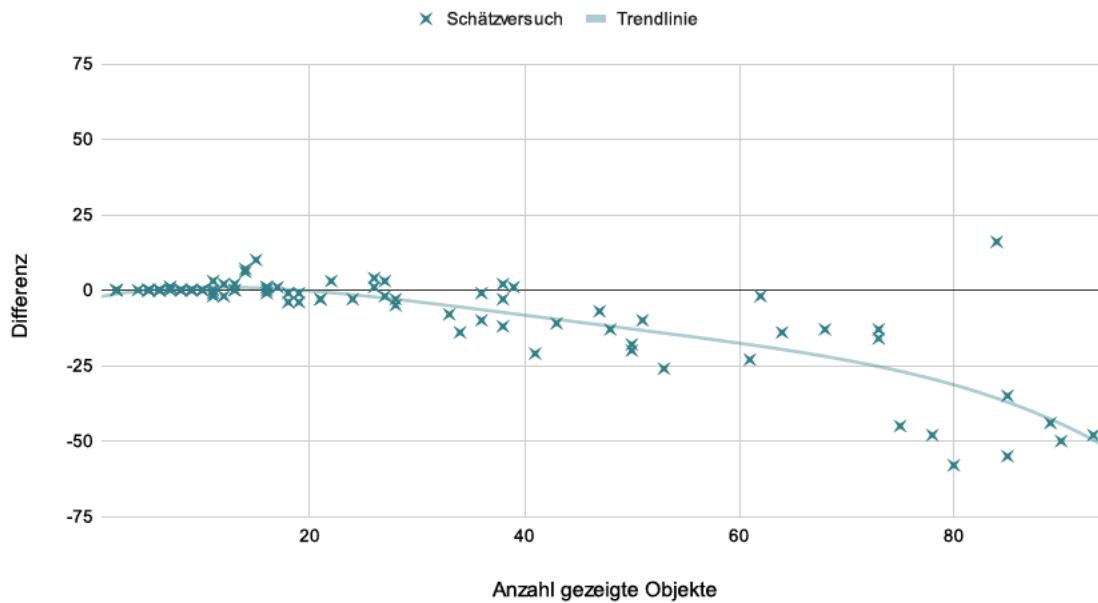


Abbildung 5.7: Schätzversuche bei Hervorhebung durch Color-Overlay-Effekt, Differenz der eingegebenen Schätzungen zu tatsächlich gezeigter Anzahl von Objekten, Polynomiale Trendlinie 4. Grades

Hervorhebung durch Stripe-Effekt

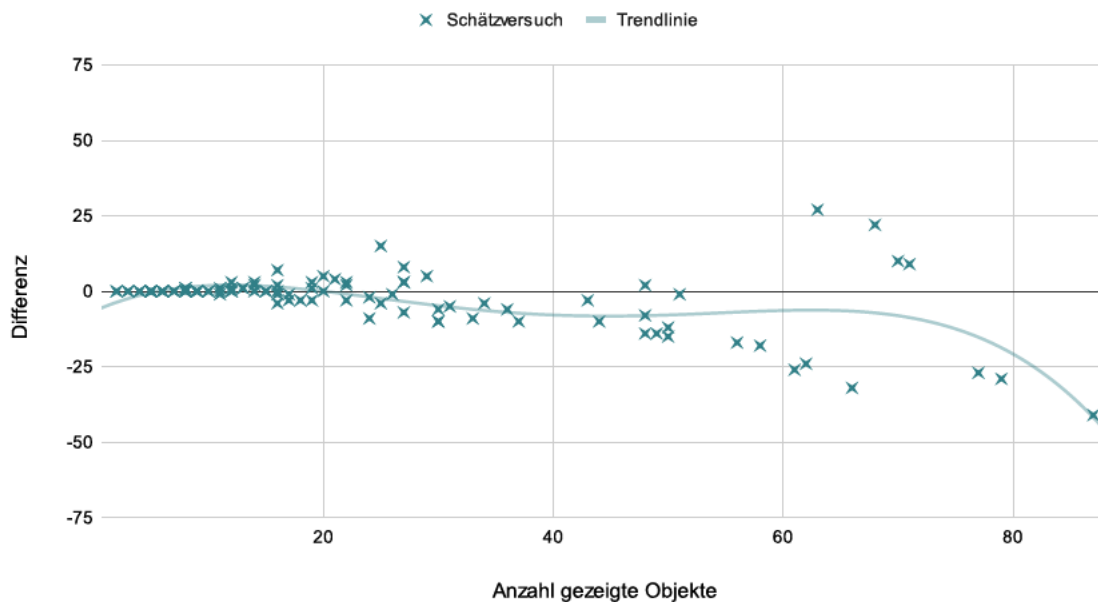


Abbildung 5.8: Schätzversuche bei Hervorhebung durch Stripe-Effekt, Differenz der eingegebenen Schätzungen zu tatsächlich gezeigter Anzahl von Objekten, Polynomiale Trendlinie 4. Grades

Auch wurde für jeden Schätzversuch das Verhältnis der Differenz d zu der zugehörigen angezeigten Objektzahl ermittelt. Wie in Tabelle 5.1 zu sehen ist, wurde diese *relative Abweichung* für jede Methode im Mittel berechnet. Hierbei wurde nicht der Betrag gewählt, wie etwa bei der Berechnung einer Varianz der Fall, da auch betrachtet werden soll, ob die Abweichungen positiv oder negativ, also die Schätzversuche zu hoch oder zu niedrig ausfielen. Es lässt sich beobachten, dass die mittlere relative Abweichung für Glow- und vor allem Stripe-Effekt verglichen mit den restlichen Methoden nur sehr gering negativ ist. Die Abweichungen der Kontrollversuche liegen dagegen im Schnitt fast 10% unter der eigentlichen Objektzahl.

Methode	Gesamt	Ohne Hervor- hebung	Outline	Glow- Effekt	Fresnel- Effekt	Color- Overlay- Effekt	Stripe- Effekt
Schätz- versuche	526	86	89	85	74	91	101
Summe gezeigte Objekte ΣN	15047	2696	2890	2326	2096	2578	2461
Mittl. Ob- jektanzahl $\bar{N}$	28,61	31,35	32,47	27,36	28,32	28,33	24,37
Mittl. Differenz $\bar{d}$	-5,05	-6,26	-5,36	-4,01	-6,01	-6,68	-2,46
Mittl. Ob- jektanzahl $\bar{N}$ wenn $d = 0$	7,49	7,76	7,00	8,44	6,56	7,32	7,95
Mittl. rel. Abwei- chung $\frac{\bar{d}}{\bar{N}}$	-0,0726	-0,0997	-0,0835	-0,0437	-0,0900	-0,0903	-0,0356
Summe rel. Abw. $\Sigma \frac{d}{N}$	-38,19	-8,57	-7,43	-3,71	-6,66	-8,22	-3,59
Varianz $s^2(\frac{d}{N})$	0,050	0,055	0,064	0,056	0,029	0,056	0,038

Tabelle 5.1: Zusammenfassung der Daten aus den Schätzversuchen nach Hervorhebungsmethode

Bemerkenswert ist, dass nach diesem Vergleich die Stripe-Methode flächendeckend besonders gut abschneidet. Dieser Unterschied fällt besonders dann auf, wenn die Stripe-Methode mit der des Color-Overlay-Effekts verglichen wird.

Im Gegensatz zu den anderen Methoden ist sowohl in den Diagrammen als auch den Mittelwerten zu erkennen, dass vor allem bei hohen Objektzahlen die Anzahl der Objekte durch diese Hervorhebungsmethode zum einen weniger unterschätzt wurde, zum anderen aber teilweise auch sehr viel öfter um signifikante Beträge überschätzt wurde. Dies ergibt im Mittel eine niedrigere Abweichung von den tatsächlich angezeigten Objektzahlen.

Entgegen der Vermutung führt die Hervorhebung durch die kontrastreicheren Methoden der Outline und des Glow-Effektes dazu, dass die Objektzahlen nur leicht weniger unterschätzt bzw. auch teilweise überschätzt werden. In Tabelle 5.1 ist zu erkennen, dass die Mittlere Differenz der beiden Methoden weniger negativ bzw. im Betrag kleiner ist als ohne Hervorhebung. Auch im relativen Verhältnis findet sich dieser Sachverhalt wieder, wo vor allem der Glow Effekt mit einem Wert von -0,0437 nur knapp die Hälfte der restlichen Differenz-Objektzahl-Verhältnisse verursacht hat. Dies spiegelt sich teilweise auch in der Umfrage wider, in der die Outline- und Glow-Methoden von den Teilnehmenden als besser erkennbar eingestuft wurden. Weiterhin lässt sich feststellen, dass die Anzahl der Objekte durch die Hervorhebung des Color-Overlay-Effektes besonders stark unterschätzt wurde. Hier ist die mittlere absolute Differenz mit -6,68 besonders groß, und die zugehörige mittlere relative Differenz wird nur durch die Darstellung ohne Hervorhebung übertroffen.

Dies fällt besonders auf, da die Hervorhebung durch die texturierte Variante des Stripe-Effekts eine sehr geringe mittlere absolute Abweichung aufweist. Einerseits könnte die recht geringe mittlere Objektzahl der Grund sein, jedoch lässt die weiterhin niedrige relative Abweichung auch den Schluss zu, dass diese Methode im vor allem im Vergleich mit der Color-Overlay-Methode eben durch die Streifentextur stärker wahrgenommen wird und die Objektanzahl dementsprechend in geringerem Ausmaß unterschätzt wird. Der Grund für diesen Gegensatz lässt sich durch diese Studie nicht eindeutig bestimmen, jedoch bestätigen die Ergebnisse des Fragebogens die letztere Annahme. Denn hier wurde die Stripe-Methode gegenüber der des Color-Overlay-Effekts als besser erkennbar eingestuft (siehe Abb. 5.2). Es gilt weitere Untersuchungen anzustellen, ob und spezifisch wie Muster und Texturen in Hervorhebungsmethoden sich auf die Wahrnehmung dieser ausüben.

Varianzanalyse

Um die Ergebnisse auszuwerten und einordnen zu können wurde eine einfaktorielle Varianzanalyse (**analysis of variances**, ANOVA) durchgeführt. Diese ist ein statistisches Verfahren, um die Abhängigkeit metrischer Messdaten von kategorischen Variablen bzw. Gruppen zu prüfen. Umgangssprachlich wird sie häufig als Vergleich der Varianzen zwischen Gruppen mit den Varianzen innerhalb der betrachteten Gruppen beschrieben. In dieser vorliegenden Auswertung wurde eine ANOVA durchgeführt, um einen möglichen Zusammenhang der relativen Abweichungen zu den entsprechenden Hervorhebungsmethoden $\frac{d}{N}$ nachweisen oder widerlegen zu können. Hierfür wird die Nullhypothese H_0 formuliert, nach der es keinen signifikanten Zusammenhang zwischen den erhobenen relativen Abweichungen $\frac{d}{N}$ und den Gruppen der Hervorhebungsmethoden gibt. Um den ausschlaggebenden F -Wert zu erhalten, müssen zunächst die Summe der Abweichungssquadrate SQ , die Freiheitsgrade df als auch die mittleren quadratischen Fehler MQF jeweils innerhalb der Gruppen und zwischen den Gruppen bestimmt werden (siehe Tab. 5.2). [Fah23]

Streuungsursache	Summe der Abweichungsquadrate SQ	Freiheitsgrade fg	Mittl. Quadr. Fehler MQF
Variabilität zwischen d. Gruppen	0,33	5	0,067
Variabilität innerhalb d. Gruppen	25,97	520	0,0499
Gesamt	26,30	525	-

Tabelle 5.2: Berechnungen zur einfaktoriellen Varianzanalyse der relativen Abweichungen, gruppiert nach Hervorhebungsmethoden

Damit ein Ergebnis mit einem Signifikanzwert von $p < \alpha = 0,05$ signifikant ist, muss der errechnete F -Wert über dem Schwellwert $F_{crit} = 2,231$ liegen, welcher sich aus den Freiheitsgraden des Experiments ergibt. Um den F -Wert zu erhalten, wird der mittlere quadratische Fehler zwischen den Gruppen $MQF_{zwischen}$ durch den mittleren quadratischen Fehler innerhalb der Gruppen $MQF_{innerhalb}$ geteilt:

$$F = \frac{MQF_{zwischen}}{MQF_{innerhalb}} \quad (5.1)$$

Damit ergibt sich ein F -Wert von $F = 1,338$ und ein p -Wert von $p = 0,247$, der deutlich über einem α -Wert von $\alpha = 0.05$ liegt. Die Nullhypothese H_0 , nach der es keinen Zusammenhang zwischen den relativen Abweichungen der Schätzversuche zu den Objektzahlen und den jeweiligen Hervorhebungsmethoden gibt, ließ sich mit $p > \alpha$ nicht ausreichend widerlegen.

Spielzeit pro Woche

Ein Zusammenhang zwischen der durch die Probanden angegebene wöchentliche Videospielzeit und der Genauigkeit ihrer Schätzungen ließ sich ebenfalls nicht feststellen. Sowohl die mittlere Differenz d als auch die Objektanzahl N im Mittel bei $d = 0$ korrelierten nicht mit der verbrachten Spielzeit. Zwar waren die Schätzversuche der Probanden aus der Kategorie 10-20 Stunden pro Woche leicht akkurater als die der Probanden aus der Kategorie 5-10 Stunden, jedoch setzt sich dieser Trend nicht in die Kategorie 30+ Stunden pro Woche fort, in der die Genauigkeit wieder nachlässt (siehe Tab. 5.3). Auch ist die Stichprobengröße zu gering, um derartige Aussagen über den Zusammenhang zwischen wöchentlicher Spielzeit und Genauigkeit der Schätzversuche zu treffen. So etwa gibt es nur einen einzigen Probanden, welcher eine wöchentliche Spielzeit von mehr als 30 Stunden angab, was eine entsprechende Auswertung nur schwer möglich macht. Es wird weitere Untersuchung empfohlen.

5.4 Fehlerquellen

Die Ergebnisse dieser Arbeit könnten durch verschiedene Faktoren beeinflusst oder verzerrt werden. Einerseits ist hier die eher geringe Stichprobengröße von $n = 10$ zu nennen, wodurch die Aussagekraft des Experiments nachlässt. Dies übt sich besonders auf die Ergebnisse aus,

Mittlere Anzahl Spielstunden pro Woche	5 - 10	10 - 20	30 +
Anz. Probanden	3	6	1
Mittl. Differenz $\bar{d}$	-6,08	-4,10	-6,53
Mittl. rel. Abweichung $\frac{\bar{d}}{\bar{N}}$	-0,080	-0,058	-0,068
Mittl. Objektanzahl $\bar{N}$ wenn $d = 0$	6,90	7,73	6,61

Tabelle 5.3: Auswertung Protokolldaten nach angegebener mittlerer Spielzeit pro Woche

da die Methodik der Studie für jedes Experiment eine Reihe von zufällig generierten Parametern wie Anzahl, Rotation und Position der Schlüssel bestimmt, die sich auf die Wahrnehmung der Objekte ausüben können. Dazu zählt auch, wie in Abschnitt 2.1 bereits erläutert, dass die menschliche Wahrnehmung auch interindividuellen Unterschieden unterliegt, wodurch zusätzliche, nicht voraussagbare Variationen eintreten können, die die Ergebnisse weiter beeinträchtigen. Diese randomisierten Parameter würden durch häufigere Wiederholung der Experimente an Einfluss verlieren und vernachlässigbar werden, wodurch eine Studie dieser Art auf deutlichere Unterschiede zwischen den Methoden stoßen könnte.

Andererseits muss dazu erwähnt werden, dass alle Probanden aus der Altersgruppe 21-25 Jahren stammen und regelmäßig Videospiele spielen. Dementsprechend könnte eine Verallgemeinerung der Evaluationsergebnisse auf die Gesamtbevölkerung verzerrt sein.

Ebenso ist weitere Forschung nötig, um zu ermitteln, ob die relative Abweichung $\frac{d}{N}$ eine geeignete Größe zur Untersuchung dieser Art darstellt. Denn dies setzt voraus, dass die Verteilung der Objektzahlen N für jede der Messreihen gleich verteilt ist. So könnte beispielsweise auch eine Beleuchtung der Größe $\frac{d}{N}$ in Betracht gezogen werden. Eine invers quadratische Beziehung zwischen Differenz d und Objektzahl N könnte sich so als sinnvoller herausstellen als eine von linearer Natur.

In diesem Zusammenhang müsste mehr erforscht werden, um die oben genannten Faktoren ausschließen und die hier dargelegten Erkenntnisse validieren zu können.

5.5 Fazit

Aus der Evaluation der erstellten Software lässt sich schließen, dass sich keine signifikanten Unterschiede zwischen den relativen Abweichungen der Schätzversuche und den jeweiligen Hervorhebungsmethoden zeigen. Es lassen sich jedoch Tendenzen ablesen, die weiterer Forschung bedürfen. So lässt sich festhalten, dass die subjektive Einschätzung der Methoden durch die Probanden über die Umfrage nur bedingt mit den Daten aus den Protokollen korreliert. Die Methoden Outline und Glow-Effekt wurden als besonders gut erkennbar eingestuft,

erzielten aber in den Schätzversuchen Ergebnisse, die nur eine leichte Abweichung von den restlichen Hervorhebungsmethoden abweichen und nicht statistisch signifikant sind. Die texturierte Methode des Stripe-Effekts erzielte geringere Abweichungen und damit akkuratere Schätzversuche als ihr untexturiertes Gegenstück der Color-Overlay-Methode, was sich auch in der Bewertung der Erkennbarkeit durch die Probanden widerspiegelt.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

Diese Arbeit beschäftigt sich mit den Unterschieden in der Wahrnehmung von nicht-diegetischen Hervorhebungsmethoden in Videospielen. Zu diesem Zweck wurde eine Software konzipiert und umgesetzt, die eine videospiegelnahe Umgebung simulieren soll. In dieser Umgebung wurden den Probanden eine Anzahl von Objekten gezeigt, die durch die verschiedenen Methoden hervorgehoben wurden. Nach einer Zeit von zwei Sekunden wurden die Objekte durch eine schwarze Blende verborgen und die Probanden wurden gebeten, die Anzahl der Objekte zu schätzen. Dieser Ablauf wurde je Proband 50 Mal wiederholt und die Daten der Schätzversuche automatisch protokolliert. Aus diesen Protokollen konnte geschlossen werden, dass es keine signifikanten Unterschiede in der wahrgenommenen Anzahl der Objekte zwischen den betrachteten Methoden gibt.

Es lassen sich allenfalls Tendenzen erkennen, die weiterer Forschung benötigen.

6.2 Weiterführende Arbeit

Ansätze zu weiteren Arbeiten auf diesem Gebiet lassen sich in zahlreicher Ausführung finden. Mit Blick auf die Ergebnisse ist in jeden Fall zu erwähnen, dass diese Arbeit eine begrenzte Stichprobengröße aufweist. So könnte das Experiment mit einer repräsentativeren Zahl an Teilnehmenden zu eindeutigeren Ergebnissen gelangen. Zusätzlich ist zu empfehlen, auch Probanden aus anderen Alters- und Bevölkerungsschichten zu inkludieren, um ein vollständigeres Bild zu erlangen. Für weitere Studien dieser Art könnten Untersuchungen hilfreich sein, die die Genauigkeit der Schätzungen in Abhängigkeit anderer Parameter misst. So könnte geschätzt werden, wie eine geringere oder höhere gegebene Zeit die Schätzungen verbessert oder verschlechtert. Es könnte auch betrachtet werden, wie Hervorhebungsmethoden, die sich zeitlich verändern, wie Animationen oder blinkende Effekte die Einschätzung beeinflussen. Auch können Parameter der spezifischen Methoden erforscht werden. So könnten beispielsweise die Auswirkungen verschiedener Dicken der Outline bzw. des Glow-Effektes, verschiedene Stärken des Fresnel-Effektes so wie Opazitäten für den Color-Overlay-Effekt untersucht werden.

Neben den Parametern der Hervorhebungsmethoden können auch andere Einflussfaktoren untersucht werden. So etwa können andere Objekte vor verschiedenen Hintergründen gezeigt werden und die Auswirkungen auf die Wahrnehmung der Probanden untersucht werden. Auch könnten sich zukünftige Arbeiten mit anderen Methoden als den hier betrachteten beschäftigen. Neben weiteren nicht-diegetischen Methoden, wie Menüelementen oder gar Symbolen können auch diegetische Methoden wie Spotlights oder Animationen betrachtet werden. Hierbei ist zu erwähnen, dass

Anhang A: Fragebogen

Frage	Antwortmöglichkeiten
Alter	unter 20 21 - 25 26 - 30 30 - 40 über 40
Geschlecht	Männlich Weiblich Divers
Wie viele Stunden pro Woche spielst du im Durchschnitt Videospiele?	0 0 - 5 5 - 10 10 - 20 20 - 30 über 30
Ist bei dir eine Farbblindheit, Sehbehinderung oder eine sonstige Sehschwäche bekannt?	Keine Bekannt Ja, und zwar: [Textfeld]

Tabelle A.1: Fragen und zugehörige Antwortmöglichkeiten des Fragebogens, Teil 1: Demografische Erfassung

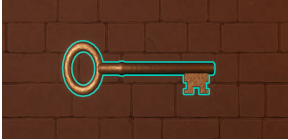
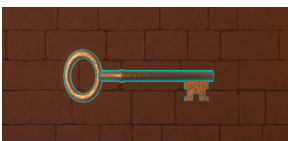
Frage	Antwortmöglichkeiten
Schätze abschließend bitte die Erkennbarkeit der verschiedenen Hervorhebungsmethoden ein. (Wie akkurat fandest du deine eigenen Schätzungen je nach Methode?)	
Wie schätzt du die Erkennbarkeit der Outline ein? 	1 (sehr schlecht) 2 3 4 5 (sehr gut)
Wie schätzt du die Erkennbarkeit des Glow-Effekts ein? 	1 (sehr schlecht) 2 3 4 5 (sehr gut)
Wie schätzt du die Erkennbarkeit des Fresnel-Effekts ein? 	1 (sehr schlecht) 2 3 4 5 (sehr gut)
Wie schätzt du die Erkennbarkeit des Color-Overlay-Effekts ein? 	1 (sehr schlecht) 2 3 4 5 (sehr gut)
Wie schätzt du die Erkennbarkeit des Stripe-Effekts ein? 	1 (sehr schlecht) 2 3 4 5 (sehr gut)
Wie schätzt du die Erkennbarkeit ohne Hervorhebung ein? 	1 (sehr schlecht) 2 3 4 5 (sehr gut)

Tabelle A.2: Fragen und zugehörige Antwortmöglichkeiten des Fragebogens, Teil 2: Einschätzung der Erkennbarkeit der Hervorhebungsmethoden

Glossar

Hero Shooter ist ein Subgenre der Shooter-Spiele, in denen die Spielenden Charaktere auswählen können, die sich durch eigene Fähigkeiten auszeichnen.

MOBA (*Multiplayer Online Battle Arena*) ist ein Subgenre der Echtzeit-Strategiespiele, in denen Teams von Spielenden in einer Arena gegeneinander antreten.

Platformer sind Spiele, in denen sich der Spielercharakter durch präzises Springen und Laufen fortbewegt, auch als *Jump'n'Run* bekannt.

Rogue-like ist ein Spielegenre, dessen Vertreter sich durch prozedural generierte Abschnitte, sog. *Permadeath* (keine Lademöglichkeit bei Tod der Spielfigur) und hohem Wiederspielwert auszeichnen.

RPG (*Role-Playing Game*) ist ein Rollenspiel.

Taktik-Shooter ist ein Shooter-Subgenre, in denen Planung und Strategie zentrale Rollen spielen.

Literaturverzeichnis

- [Sch21] Kai Schuster. „Grundlagen der menschlichen Wahrnehmung und Stärken und Risiken visueller Kommunikation“. In: *Handbuch Methoden Visueller Kommunikation in der Räumlichen Planung*. Hrsg. von Diedrich Bruns u. a. Wiesbaden: Springer Fachmedien Wiesbaden, 2021, S. 27–43. ISBN: 978-3-658-29862-3. DOI: [10.1007/978-3-658-29862-3_2](https://doi.org/10.1007/978-3-658-29862-3_2). URL: https://doi.org/10.1007/978-3-658-29862-3_2 (besucht am 22. 05. 2025).
- [Les20] Andreas Leschnik. *Visuelle Wahrnehmung: Grundlagen, Clinical Reasoning und Intervention im Kindes- und Jugendalter*. Springer Fachmedien Wiesbaden, 2020. ISBN: 9783658308773. DOI: [10.1007/978-3-658-30877-3](https://doi.org/10.1007/978-3-658-30877-3). URL: <http://dx.doi.org/10.1007/978-3-658-30877-3> (besucht am 24. 05. 2025).
- [BKV22] Andreas Butz, Antonio Krüger und Sarah Theres Völkel. *Mensch-Maschine-Interaktion*. ger. 3. Auflage. De Gruyter Studium. Berlin, Boston: De Gruyter Oldenbourg, 2022. ISBN: 9783110753325. DOI: [10.1515/9783110753325](https://doi.org/10.1515/9783110753325). URL: <https://doi.org/10.1515/9783110753325> (besucht am 07. 05. 2025).
- [BW17] Christian Becker-Carus und Mike Wendt. *Allgemeine Psychologie eine Einführung*. ger. 2., vollständig überarbeitete und erweiterte Neuauflage, Online-Ausgabe. SpringerLink Bücher. Springer Berlin, Heidelberg, 2017. ISBN: 9783662530061. DOI: [10.1007/978-3-662-53006-1](https://doi.org/10.1007/978-3-662-53006-1).
- [WR00] Karl F. Wender und Rainer Rothkegel. „Subitizing and its subprocesses“. In: *Psychological Research* 64.2 (Dez. 2000), S. 81–92. ISSN: 1430-2772. DOI: [10.1007/s004260000021](https://doi.org/10.1007/s004260000021). URL: <https://doi.org/10.1007/s004260000021> (besucht am 23. 05. 2025).
- [Kau+49] E. L. Kaufman u. a. „The Discrimination of Visual Number“. In: *The American Journal of Psychology* 62.4 (1949), S. 498–525. ISSN: 00029556. URL: <http://www.jstor.org/stable/1418556> (besucht am 24. 05. 2025).
- [Rai+08] Henry Railo u. a. „The role of attention in subitizing“. In: *Cognition* 107.1 (2008), S. 82–104. ISSN: 0010-0277. DOI: <https://doi.org/10.1016/j.cognition.2007.08.004>. URL: <https://www.sciencedirect.com/science/article/pii/S0010027707002302>.
- [GB03] C. Shawn Green und Daphne Bavelier. „Action video game modifies visual selective attention“. In: *Nature* 423.6939 (Mai 2003), S. 534–537. ISSN: 1476-4687. DOI: [10.1038/nature01647](https://doi.org/10.1038/nature01647). URL: <https://doi.org/10.1038/nature01647>.
- [GB05] C S Green und D Bavelier. „Enumeration versus multiple object tracking: the case of action video game players“. en. In: *Cognition* 101.1 (Dez. 2005), S. 217–245.
- [MS09] Karen Murphy und Amy Spencer. „Playing video games does not make for better visual attention skills“. In: *J. Articles Support Null Hypothesis* 6 (Jan. 2009).
- [SF10] Ian Spence und Jing Feng. „Video Games and Spatial Cognition“. In: *Review of General Psychology* 14 (Juni 2010), S. 92–104. DOI: [10.1037/a0019491](https://doi.org/10.1037/a0019491).

- [CG23] Emma G. Cunningham und C. Shawn Green. *Cognitive Skills Acquired From Video Games*. Sep. 2023. DOI: [10.1093/acrefore/9780190228613.013.1468](https://doi.org/10.1093/acrefore/9780190228613.013.1468). URL: <https://oxfordre.com/communication/view/10.1093/acrefore/9780190228613.001.0001/acrefore-9780190228613-e-1468>.
- [Sou97] Étienne Souriau. „Die Struktur des filmischen Universums und das Vokabular der Filmologie“. In: *montage AV. Zeitschrift für Theorie und Geschichte audiovisueller Kommunikation*. montage AV. Zeitschrift für Theorie und Geschichte audiovisueller Kommunikation 6.2 (1997), S. 140–157. DOI: [10.25969/mediarep/2](https://doi.org/10.25969/mediarep/2).
- [Fux07] Anton Fuxjäger. „Diegese, Diegesis, diegetisch. Versuch einer Begriffsentwerrung“. In: *montage AV. Zeitschrift für Theorie und Geschichte audiovisueller Kommunikation* 16.2 (2007), S. 17–37. URL: <https://doi.org/10.25969/mediarep/272> (besucht am 12.05.2025).
- [Dop18] Claudia Doppioslash. *Physically Based Shader Development for Unity 2017 : Develop Custom Lighting Systems*. eng. 1st ed. 2018. Berkley, California: Apress, 2018. ISBN: 9781484233092. URL: <https://link.springer.com/book/10.1007/978-1-4842-3309-2> (besucht am 08.05.2025).
- [Bel24] Tobias Belling. *Investigating and Comparing Video Game-Based Highlighting Techniques in Realistic Situated Environments*. 2024. URL: <https://d-nb.info/1359700021/34> (besucht am 20.05.2025).
- [Lin23] Joaquin Lindkvist. *Assessing Visual Attention Methods in Video Games : A Comparative Study of Effectiveness and Colorblind Accessibility*. 2023. URL: <https://urn.kb.se/resolve?urn=urn:nbn:se:bth-25014> (besucht am 12.05.2025).
- [OS20] Alexander Olsson und Daniel Sundahl. *An Experimental Study on Highlighting Objects in Video Game : The Visual Attention Between the Colors Red, Green, Blue and Yellow*. 2020. URL: <https://urn.kb.se/resolve?urn=urn:nbn:se:bth-20166> (besucht am 12.05.2025).
- [Leu17] Joshua Leung. „Understanding the Noticeability and Distraction of Interactive Highlighting Techniques“. Diss. University of Canterbury, 2017. URL: <https://ir.canterbury.ac.nz/items/411e2c3e-a1fc-4625-92ea-fc5ef9d3a19e> (besucht am 18.05.2025).
- [Ble] Blender Foundation. *Blender Development Fund*. URL: <https://fund.blender.org/> (besucht am 07.05.2025).
- [Gol20] Ben Golus. *The Quest for Very Wide Outlines. An Exploration of GPU Silhouette Rendering*. 2020. URL: <https://bgolus.medium.com/the-quest-for-very-wide-outlines-ba82ed442cd9> (besucht am 16.04.2025).
- [RS] Stefan Rossbach und René Siewert. *Backface Culling*. URL: https://www.inf.fu-berlin.de/lehre/WS06/19605_Computergrafik/doku/rossbach_siewert/backfaceculling.html (besucht am 12.05.2025).
- [FB85] Kenneth P. Fishkin und Brian A. Barsky. „An Analysis and Algorithm for Filling Propagation“. In: *Computer-Generated Images*. Hrsg. von Nadia Magnenat-Thalmann und Daniel Thalmann. Tokyo: Springer Japan, 1985, S. 56–76. ISBN: 978-4-431-68033-8.

- [RT06] Guodong Rong und Tiow-Seng Tan. „Jump flooding in GPU with applications to Voronoi diagram and distance transform“. In: *Proceedings of the 2006 symposium on Interactive 3D graphics and games - SI3D '06*. SI3D '06. ACM Press, 2006, S. 109. DOI: [10.1145/1111411.1111431](https://doi.org/10.1145/1111411.1111431). URL: <https://www.comp.nus.edu.sg/~tants/jfa.html> (besucht am 24.05.2025).
- [Ecm17] Ecma International. *ECMA-404 - Ecma International*. 2017. URL: <https://ecma-international.org/publications-and-standards/standards/ecma-404/> (besucht am 29.04.2025).
- [Fah23] Ludwig Fahrmeir. *Statistik der Weg zur Datenanalyse*. ger. 9., überarbeitete und ergänzte Auflage. Springer Spektrum, 2023, S. 519–528. ISBN: 9783662675267.

Eidesstattliche Erklärung

Hiermit versichere ich – Robin Schöppach – an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.

Mittweida, 19. Juni 2025

Ort, Datum

Robin Schöppach

