

Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering

Entwurf und Entwicklung eines Spiels zur Vermittlung ausgewählter Themen des Software Engineerings

Design and development of a game to present selected
software engineering topics

Masterarbeit

im Studiengang Informatik

von

Florian Schulz

Prüfer: Prof. Dr. rer. nat. Kurt Schneider
Zweitprüfer: Dr. rer. nat. Jil Ann-Christin Klünder
Betreuer: Alexander Specht

Hannover, 24.05.2024

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 24.05.2024

Florian Schulz

Zusammenfassung

Im Rahmen dieser Arbeit wurde ein Spiel zur Vermittlung von ausgewählten Themen des Fachgebiet Software Engineering entworfen und entwickelt. Das Spiel wurde für die 270° Projektionsfläche des InfoLabs, ein spezieller Raum der Leibniz Universität Hannover, konzipiert. Zur Identifikation von relevanten Themen für das Fachgebiet wurde eine Vorabrecherche der Fachgebietswebseite und ein Workshop durchgeführt. Die erarbeiteten Themen wurden priorisiert und mit Hilfe des SWEBOK (Software Engineering Body of Knowledge) in Wissensbereiche gruppiert. Dabei wurden drei Wissensbereiche identifiziert, welche anschließend in Form von Minispielen umgesetzt wurden. Die Minispiele sind durch ein Hauptspiel mit einer offenen Spielewelt verbunden. Das Spiel wurde mit der Unity Game Engine entwickelt, wobei insbesondere auf die Erweiterbarkeit und Wartbarkeit des Spiels geachtet wurde. Mit Hilfe einer kleinen Nutzerstudie wurde die Qualität des Spiels getestet und gefundene Fehler wurden behoben. Bei der Auswertung der Studie zeigte sich ein positiver Gesamteindruck bei den Probanden gegenüber des Spiels.

Abstract

In the context of this thesis, a game was designed and developed to present selected topics from the field of software engineering. The game was designed for the 270° projection screen of the InfoLab, a special room at Leibniz Universität Hannover. To identify relevant topics for the subject area, a preliminary search of the department's website and a workshop was conducted. The topics identified were prioritized and grouped into knowledge areas with the help of the SWEBOK (Software Engineering Body of Knowledge). Three knowledge areas were identified, which were then implemented in the form of mini-games. The mini-games are linked through a main game with an open game world. The game was developed using the Unity game engine, with particular attention paid to the expandability and maintainability of the game. The quality of the game was tested in a small user study and any errors found were corrected. The evaluation of the study revealed a positive overall impression of the game among the test subjects.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielsetzung	1
1.3	Lösungsansatz	2
1.4	Struktur der Arbeit	3
2	Grundlagen	5
2.1	Spieleentwicklung	5
2.1.1	Elemente in Spielen und ihr Zweck	5
2.1.2	Spielertypen	6
2.2	Themen des Fachgebiets Software Engineering	7
2.2.1	Vorlesungsinhalte	7
2.2.2	Aktuelle Forschungsschwerpunkte	8
2.2.3	Themenbereiche für Abschlussarbeiten	9
2.2.4	Einordnung der Themengebiete	10
2.3	Verwandte Arbeiten	13
3	Anforderungen an das Spiel	15
3.1	Anforderungen im Überblick	15
3.2	Priorisierung	18
4	Konzeptentwicklung	19
4.1	Vorbereitung	19
4.2	Auswahl der Methode	19
4.3	Workshop	21
4.3.1	Ziel des Workshops	21
4.3.2	Planung des Workshops	22
4.3.3	Durchführung des Workshops	24
4.3.4	Ergebnisse des Workshops	26
4.3.5	Erkenntnisse zur Spiele-Entwicklung	29
4.4	Entwurf des Spielkonzepts	30
4.4.1	Konzept des Spiels	30
4.4.2	Nutzung der räumlichen Gegebenheiten	32

4.4.3	Ideen für Minispiele	33
4.4.4	Ausgewählte Minispiele	39
4.4.5	Umsetzung der Anforderungen	40
5	Implementierung	43
5.1	Game Engine	43
5.2	Softwarearchitektur	44
5.2.1	Grundspiel	44
5.2.2	Minispiele	46
5.2.3	Szenen-Übergänge	49
5.3	Umsetzung im InfoLab	50
5.3.1	Eindruck aus dem InfoLab	50
5.3.2	Sitzordnung	50
5.4	Erweiterbarkeit	52
6	Studie	55
6.1	Planung und Durchführung	55
6.1.1	Planung	55
6.1.2	Durchführung	56
6.2	Evaluation	57
6.3	Gefundene Fehler und Wünsche	61
6.3.1	Umgesetzt	61
6.3.2	Nicht Umgesetzt	62
7	Diskussion	63
7.1	Umsetzung der Anforderungen	63
7.2	Limitationen	64
7.2.1	Workshop	64
7.2.2	Studie	64
8	Zusammenfassung und Ausblick	67
8.1	Zusammenfassung	67
8.2	Ausblick	68
A	Eindrücke aus dem Spiel	69
A.1	Dialoge	69
A.2	Szenenwechsel	71

Kapitel 1

Einleitung

1.1 Motivation

Das Fachgebiet Software Engineering vom Institut für Praktische Informatik der Leibniz Universität Hannover ist jährlich Teil von Veranstaltungen zur Wissenskommunikation, wie *Die Nacht, die Wissen schafft*, bei denen es darum geht, sich extern zu präsentieren und vorzustellen. Besucher der Veranstaltungen sind beispielsweise potenzielle Studierende und Personen mit Interesse an Technik. Unter anderem handelt es sich aber auch um Kinder oder Personen ohne Erfahrung mit Software Engineering Inhalten.

Um die vielen verschiedenen und unterschiedlichen Zielgruppen anzusprechen, wurden bereits in der Vergangenheit Spiele zur Umsetzung verschiedener Themen der Informatik verwendet. Zum Vermitteln der Logik von Binärzahlen wurde ein Spiel entwickelt, welches im Rahmen einer Abschlussarbeit konzipiert und implementiert wurde. In einer anderen Abschlussarbeit [17] wurde die der Informatik in Form eines Spiels umgesetzt. Dabei lag der Hauptfokus auf dem Vermitteln von Themen wie Binärzahlen, die Funktionsweise von des ersten physischen Rechners und geschichtlich relevante Abläufe im Themenfeld der Informatik.

Da die bisher entwickelten Spiele nur auf allgemeine Themen in der Informatik eingehen, soll im Rahmen dieser Arbeit ein Spiel zur Vermittlung von ausgewählten Themen des Software Engineering konzipiert und entwickelt werden.

1.2 Zielsetzung

Ziel dieser Arbeit ist es, ein Entwurf für ein Spiel zu entwickeln und umzusetzen, welche Themen aus dem Bereich Software Engineering vorstellt. Dabei muss identifiziert werden, welche Themen relevant sind und wie diese ansprechend vermittelt werden können. Dafür müssen relevante Themen des Software Engineering identifiziert und ausgewählt werden. Inhalte der

ausgewählten Themen sollen ansprechend in Form eines Spiels vermittelt werden. Bei der Umsetzung sind mehrere Vorbedingungen zu beachten. Zum einen muss die Zielgruppe (Schüler und angehende Studierende) berücksichtigt werden, um einen angemessenen Schwierigkeitsgrad für die Spiele zu schaffen. Außerdem weist der Veranstaltungsraum, in dem das Spiel primär gespielt werden wird, besondere technische Gegebenheiten auf, welche bei der Planung des Spiels berücksichtigt werden müssen. In dem Raum steht durch 3 Beamer eine Projektionsfläche von 270° zur Verfügung. Das Spiel soll auf der gesamten Projektionsfläche dargestellt werden. Da es für die Veranstaltung besser ist, wenn mehrere Spieler gleichzeitig spielen können, soll das Spiel für mindestens zwei Spieler als kooperatives oder kompetitives Spiel entworfen werden. Durch die große Projektionsfläche und die Anforderung von mindestens zwei Spielern ist die Bedienung des Spiels durch Maus und Tastatur ungeeignet und es sollen stattdessen Gamepad-Controller, in diesem Fall Xbox Controller, verwendet werden. Bei dem Spiel sollen geeignete Qualitätsfaktoren wie Spielspaß, Immersion und Attraktivität identifiziert und berücksichtigt werden. Ein wichtiges bereits feststehendes Qualitätsmerkmal des Spiels ist seine Wartbarkeit, da sich die relevanten Themen im Software Engineering ändern können. Deshalb soll das Spiel so entworfen werden, sodass Teile des Spiels austauschbar oder erweiterbar sind.

1.3 Lösungsansatz

Zu Beginn werden geeignete Qualitätsmerkmale für das Spiel erarbeitet, da diese essenziell für die Konzeptualisierung des Spiels sind. Die Qualitätsmerkmale werden mittels Literatur identifiziert und priorisiert.

Anschließend werden für das Fachgebiet Software Engineering relevante Themengebiete identifiziert. Hierfür wurde zuerst eine kleine Literatursuche durchgeführt, indem von Fachgebiet Software Engineering veröffentlichte Paper gesichtet wurden. Außerdem wurde bei Google Scholar nach relevanten Papern gesucht. Anschließend wurden die Informationsquellen des Fachgebiets durchsucht. Dazu gehören primär Lehrveranstaltungen wie Vorlesungen, Themen für Abschlussarbeiten und aktuelle Forschungs- und Projektthemen des Fachgebiets.

Im Anschluss wird ein Workshop durchgeführt, um die Suche der relevanten Themen zu vervollständigen und die Themen ihrer Relevanz nach zu priorisieren. Im Rahmen des Workshops findet außerdem eine offene Diskussionsrunde zur Ideenfindung für mögliche Spielkonzepte statt. Für die Durchführung des Workshops werden sowohl wissenschaftliche Mitarbeiter des Fachgebiets angefragt als auch studentische Hilfskräfte, welche mehrere abgeschlossene Veranstaltungen und Erfahrung mit der Lehre am Fachgebiet haben. Damit wird sichergestellt, dass alle Teilnehmenden einen guten

Überblick über die Themen des Fachgebiets haben.

Bei der abschließenden Konzeptualisierung und Entwicklung der spielrischen Umsetzung der zuvor identifizierten Theme wird insbesondere auf die zuvor identifizierten Qualitätsmerkmale für Spiele geachtet. Mittels abschließenden Testsessions und anschließender Nutzerstudie wird letztendlich die Lauffähigkeit und Qualität des Spiels sichergestellt.

1.4 Struktur der Arbeit

Diese Arbeit ist wie folgt strukturiert. In Kapitel 2 werden die Grundlagen zur Spieleentwicklung, die Themen des Fachgebiets Software Engineering und Verwandte Arbeiten behandelt. In Kapitel 3 werden die verschiedenen Anforderungen an das Spiel erläutert und anschließend priorisiert. Kapitel 4 beschäftigt sich mit dem Prozess der Identifikation relevanter Themen und es werden die erarbeiteten Konzepte vorgestellt. Der zu Identifizierung der relevanten Themen durch geführte Workshop wird samt seiner Ergebnisse besprochen. In Kapitel 5 wird die Umsetzung des Spielkonzepts beschrieben. Die Studie zum Spiel und ihre Ergebnisse werden in Kapitel 6 beschrieben. In Kapitel 7 werden die Umsetzung der Anforderungen und die Limitierungen der Arbeit diskutiert und Kapitel 8 beinhaltet eine Zusammenfassung der Arbeit und einen Ausblick auf mögliche Folgearbeiten.

Kapitel 2

Grundlagen

2.1 Spieleentwicklung

Es existieren viele verschiedene Definitionen zu dem Begriff **Spiel**, von denen die von Carl C. Abt [1, 21] am häufigsten verwendet wird. Es gibt aber auch heute noch Diskurs zu der Thematik.

Definition eines Spiels nach Abt

Jeder Wettbewerb (Spiel) zwischen Gegnern (Spielern), die unter Zwängen (Regeln) auf ein Ziel hinarbeiten (Gewinnen, Siegprämie).

Spiele, insbesondere Videospiele, können im Vergleich der verschiedenen digitalen Medientypen, von Filmen und Anwendungssoftware unterschieden werden, da jeweils unterschiedliche Aspekte im Fokus sind. Bei Anwendungssoftware ist der Hauptaspekt das Interagieren mit den Medien, um eine bestimmte Aufgabe zu erledigen. Dagegen dient der Film, als Medium der Unterhaltung, ohne besondere Interaktionen zu erfordern. Im Vergleich dazu, kombinieren Videospiele die jeweiligen Hauptaspekte, da nun die Interaktion mit den Medien der Unterhaltung dient.

In der Regel werden Spiele für die Nutzung in der Freizeit, ohne produktiven Zweck entwickelt. Es hat sich jedoch als effektiv erwiesen, Spiele auch in der Lehre zur Vermittlung von Inhalten einzusetzen [3].

2.1.1 Elemente in Spielen und ihr Zweck

Im Gegensatz zu Anwendungssoftware gibt es für Videospielen andere Anforderungen. Schell [18] berichtet über die Taxonomie von Spielstimulus nach LeBlanc. Dabei soll jedes Element in einem Spiel einen Zweck haben und zur Gesamterfahrung beitragen.

Der Spieledesigner Marc LeBlanc hat eine Liste von acht *Stimuli* (engl. „Pleasures“) vorgeschlagen, welche er für die primären Spielanreize hält [18].

Das **Empfinden** wird hauptsächlich durch das Auslösen von Emotionen stimuliert. Beispielsweise kann bei Horrorspielen durch schaurige Musik eine Spannung aufgebaut werden oder der Spieler durch ein lautes Geräusch erschreckt werden. Die **Fantasie** kann durch das Abtauchen als fiktiver Charakter in eine fiktive Welt angeregt werden. Eine **Handlung** kann durch fesselndes Schreiben und dem Einsatz von Dramatik das Gesamterlebnis verbessern. Durch **Herausforderungen** werden Spieler dazu ermutigt, bestimmte Fähigkeiten zu meistern. Dieser Anreiz wird üblicherweise bei Spieler gegen Spieler Spielen in den Fokus gesetzt. Bei der **Entdeckung** geht es um das Erkunden und Aufdecken von Geheimnissen. **Gemeinschaft** hat den Hauptfokus auf kooperative und soziale Interaktion. Hierbei muss die soziale Interaktion nicht mal unbedingt mit anderen realen Spielern erfolgen, sondern kann auch mit sogenannten Non-Playable Characters (NPC) sein. Durch Einflussnahme auf die Spielwelt kann das Spielen als **Ausdruck** zum Erlebnis werden. Hierbei geht es darum, dass die Spielenden die Welt nach ihren Wünschen verändern können und somit dem Spielerlebnis Ausdruck verleihen. **Hingabe** zum Spiel wird durch Einbindung des Spielers durch Routineaufgaben für Fortschritt als Zeitvertreib realisiert.

Bei der Entwicklung eines Spiels kommt es darauf an, eine gute Balance der verschiedenen Stimuli im Spiel darzustellen. Dies ist wichtig, um möglichst viele Personen für das Spiel zu begeistern und es nicht eindimensional wirken zu lassen.

2.1.2 Spielertypen

Schell [18] berichtet über die Taxonomie von Spielertypen nach Bartle. Bartle führte ein Modell ein, welches Spielende einer oder mehrerer von vier Klassen zuordnet. Die vier Klassen sind Achiever, Killer, Explorer und Socializer. Die **Achiever** möchten hauptsächlich die Ziele des Spiels erreichen, welche häufig durch Quests, besondere Schwierigkeitsgrade im Spiel oder durch Trophäen umgesetzt werden. Ihr größtes Bedürfnis in einem Spiel ist die Herausforderung. **Explorer** möchten die ganze Breite des Spiels entdecken. Ihr größtes Bedürfnis in einem Spiel ist das Erkunden der Welt und Umgebung. **Socializer** sind hauptsächlich an Interaktionen und Beziehungen mit anderen Spielenden interessiert. Der Hauptfokus liegt in der gemeinschaftlichen Interaktion. **Killer** haben Interesse, sich mit anderen Spielenden zu messen und diese zu besiegen. Dieser Gruppe lässt sich nur schwer eine einzelne Kategorie aus LeBlanc's Taxonomie zuordnen. Sie habe die Begierde, sich zu messen und Gegenstände und andere Spieler zu zerstören.

In Abbildung 2.1 werden die vier Typen und ihre jeweiligen Interessen dargestellt. Achiever sind daran interessiert, auf die Welt einzuwirken. Explorer sind daran interessiert, mit der Welt zu interagieren. Socializer sind daran interessiert, mit Spielern zu interagieren. Killer sind daran interessiert,

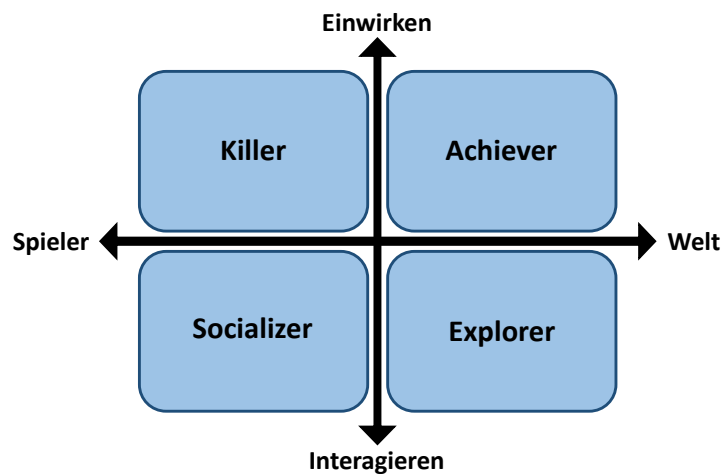


Abbildung 2.1: Darstellung der Interessen der Spielertypen nach Bartle [18]

auf Spieler einzuwirken [18]. Die Taxonomien von LeBlanc und Bartle helfen zwar die Stimuli von Spielern zu identifizieren und diese dann Spielergruppen zuzuordnen, jedoch reichen diese einfachen Klassifizierungen nicht aus, um das vollständige menschliche Verlangen zu beschreiben. Es gibt noch weitere Aspekte, welche für im Rahmen dieser Arbeit jedoch nicht relevant sind und somit auch nicht erwähnt werden.

2.2 Themen des Fachgebiets Software Engineering

Software Engineering beschäftigt sich vor allem mit der strukturierten und systematischen Entwicklung von Software und geht dabei über den reinen Implementierungsprozess hinaus.

2.2.1 Vorlesungsinhalte

Im Folgenden wird beschrieben auf welche Themengebiete sich das Fachgebiet Software Engineering im Bereich Lehre, Forschung und Abschlussarbeiten fokussiert.

Das Fachgebiet Software Engineering der Leibniz Universität Hannover bietet vier Grundlagenveranstaltungen an. Darunter fallen die Vorlesungen Softwaretechnik, Software-Qualität, Software Process Engineering und Requirements Engineering [11].

Als Grundlagenveranstaltung vermittelt Softwaretechnik ein Überblick über die Themengebiete des Software Engineering. In dieser Veranstaltung werden essenzielle Begriffe, Konzepte und Prinzipien, wie Design Patterns, Information-Hiding und die Ingenieursprinzipien erläutert.

Software-Qualität behandelt zielorientiertes Messen und Bewerten von Software. Ein wichtiger Teil der Veranstaltung ist die ISO 25010¹. Diese enthält unter anderem ein Qualitätsmodell, welches Softwareingenieuren beim Festlegen von Qualitätsaspekten zur Bewertung eines Softwareprodukts hilft. Das Qualitätsmodell² enthält neun Qualitätsaspekte wie z.B. Kompatibilität, Sicherheit, Interaktionsfähigkeit und Wartbarkeit und verfeinert jeden dieser Aspekte mit Hilfe von spezifischen Merkmalen. Die Interaktionsfähigkeit hat beispielsweise die Aspekte Lernbarkeit, Schutz vor Bedienungsfehlern und User Assistenz. Ein weiterer wichtiger Bestandteil der Veranstaltung ist das Testen von Softwaresystemen. Dabei wird das Konzept des Äquivalenzklassentests vermittelt, bei dem es darum geht, mit möglichst wenigen Testfällen möglichst viele potenzielle Fehler zu finden. Dafür werden Ein- und Ausgaben der Software in Klassen unterteilt, von den erwartet wird, dass diese sich gleichartig verhalten.

Software Process Engineering thematisiert den Lebenszyklus von Software-Prozessen. Dabei werden die Schritte der Planung, Analyse, Design, Implementierung, Testen, Integration und Wartung durchlaufen. Es werden außerdem die Unterschiede zwischen traditionellen, agilen und hybriden Methoden beschrieben. Bei traditionellen Methoden liegt der Fokus auf der Dokumentation und ihrer strikten Einhaltung. Bei den agilen Methoden liegt der Fokus auf dem flexiblen Umgang mit den Anforderungen und der damit verbundenen Möglichkeit, auf Änderungen der Anforderungen zu reagieren.

Requirements Engineering beschäftigt sich mit dem Management von Anforderungen. Dabei werden die Phasen Elicitation, Interpretation, Negotiation, Dokumentation und Verifizierung und Validierung durchlaufen. Im Anschluss daran werden die Änderungen von Anforderungen verfolgt. Kunden sollen unterstützt werden, Anforderungen für die Software aufzustellen und zu priorisieren und diese Wünsche letztendlich passend für die Entwickler zu dokumentieren.

2.2.2 Aktuelle Forschungsschwerpunkte

Das Fachgebiet Software Engineering der Leibniz Universität Hannover hat derzeit drei Forschungsschwerpunkte, welche von der DFG gefördert werden. Darunter fallen die Projekte ViViUse, softXplain und TraceSEC [10].

Das Projekt ViViUse [19] beschäftigt sich mit sogenannten Vision Videos, welche ein gemeinsames Projektverständnis unterstützen sollen. Vision Videos sind Videos, welche die Vision eines Systems veranschaulichen und so Stakeholdern helfen, ihre mentalen Modelle des zukünftigen Systems offenzulegen, zu diskutieren und abzugleichen [15]. In diesem Forschungsprojekt wird erforscht, wie eine effiziente und effektive Erstellung von Vision Videos

¹<https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

²Stand 15.05.24

umgesetzt werden kann [19]. Nagel et al. [15] erforschten beispielsweise die Möglichkeiten von interaktiven Vision Videos, in den Fragen zum Video und Auswahlmöglichkeiten zur nächsten Aktion im Video eingebunden wurden. Das Ergebnis der Studie war, dass interaktive Elemente in Vision Videos zu mehr Aufmerksamkeit und mehr Verständnis der Stakeholder führten.

Das Projekt softXplain [20] beschäftigt sich mit Anforderungen für selbsterklärende Software. Die Schlüsselfragen in diesem Projekt sind: wie kann Software erfahren, welches Verhalten von ihr erwartet wird und was die richtige Form und Zeitpunkt für eine Erklärung sind [20]. Chazette et al. [5] haben eine Literaturanalyse und Interviewstudie durchgeführt, um herauszufinden, welche Aktivitäten und Praktiken bei der Entwicklung von selbst-erklärender Software genutzt werden sollten. Deters et al. [8] forschten an zielgerichteten Heuristiken, um die Bewertung von selbsterklärender Software zu ermöglichen.

Das Projekt TraceSEC [13] beschäftigt sich mit der Nachverfolgbarkeit von Sicherheitsanforderungen in der Softwareentwicklung. Hierbei soll es ermöglicht werden, sicherheitsrelevante Aktivitäten anhand von Artefakten und einem Qualitätsmodell zu verfolgen [13]. Ahrens et al. [2] führten eine Interviewstudie durch, um die Umsetzbarkeit von Eye-Tracking zur automatisierten Erkennung von Trace Links zu überprüfen. Unter Trace Links versteht man Verknüpfungen zwischen Artefakten, welche die mentale Anstrengung bei Software-Engineering Aktivitäten verringern können.

2.2.3 Themenbereiche für Abschlussarbeiten

Im Fachgebiet Software Engineering werden Abschlussarbeiten (Bachelor- und Masterarbeiten) für Studierende angeboten. Um das Anfragen nach Abschlussarbeitsthemen für die Studierenden zu vereinfachen, wird auf der Webseite des Fachgebiets eine Übersicht über die angebotenen Themenbereiche der Mitarbeiter angeboten [12]. Folgende Themen werden derzeit angeboten:

- Requirements Engineering
- Eye Tracking
- Soziale Aspekte
- Hybride Methoden
- Softwareprozesse
- ML im Software Engineering
- (Usable) Security & Privacy
- Vision Videos
- Erklärbarkeit
- Usability
- Agile Methoden
- Soziale Netzwerkanalyse
- Stimmungsanalyse
- User-Centered Design
- User Experience
- Traceability

2.2.4 Einordnung der Themengebiete

Eine Einordnung der Themengebiete ist wichtig, um diese in einen wissenschaftlichen Kontext zu setzen. Die Einordnung wird genutzt, um in Kapitel 4.3.4 die erarbeiteten Themengebiete einzuordnen und zu gruppieren. Um die Themenbereiche des Fachgebiets Software Engineering einzuordnen wird die Strukturierung des Software Engineering Body of Knowledge (SWEBOK) [4] genutzt. Der Guide „Strukturierung des Software Engineering Body of Knowledge (SWEBOK)“ erfasst allgemein anerkanntes Wissen über Software Engineering und unterteilt Software Engineering dabei in 15 Wissensbereiche [4]. Beispiele für diese Wissensbereiche sind der Umgang mit Softwareanforderungen, das Design von Software – insbesondere der Architektur, das Testen von Software, die Software Wartung, Modelle und Methoden des Software Engineerings und Software Qualität [4]. Im Fachgebiet Software Engineering der Leibniz Universität Hannover liegt ein besonderer Fokus auf den SWEBOK Wissensbereichen *Software Requirements*, *Software Quality* und *Software Engineering Process*, *Software Engineering Professional Practice* und *Software Engineering Models and Methods* [4, 10, 11]. Tabelle 2.1 veranschaulicht die vom Fachgebiet abgedeckten Wissensbereiche durch Vorlesungsinhalte, Forschungsschwerpunkte und Themenbereiche für Anschlussarbeiten. Die Vorlesungen sind in der Tabelle mit ihren Abkürzungen aufgelistet.

Einordnung Vorlesungen

In der Tabelle 2.1 ist die Veranstaltung **Softwaretechnik (SWT)** mit Klammern gekennzeichnet, da diese Grundlagenveranstaltung viele der SWEBOK Wissensbereiche anschnidet, jedoch nicht weiter in die Tiefe geht.

Die Vorlesung **Software Qualität (SWQ)** behandelt hauptsächlich Themen aus Kapitel 10 des SWEBOK. SWQ vermittelt Grundlagen der Software Qualität (Kapitel 10.1 *Software Quality Fundamentals*) und legt den Studierenden auch Vorgehen wie Reviews (Kapitel 10.2.3 *Reviews and Audits*) nahe. Außerdem werden Software Qualitätsanforderungen (Kapitel 10.3.1 *Software Quality Requirements*) behandelt. Die Veranstaltung lehrt außerdem Inhalte aus Kapitel 4 des SWEBOK (Software Testing), in dem Testverfahren behandelt werden. SWEBOK-Kapitel 4 ist in Tabelle 2.1 nicht weiter angeführt, da es weder Forschungsschwerpunkte noch Themen für Abschlussarbeiten in diesem Bereich gibt.

Die Veranstaltung **Software Process Engineering (SPE)** behandelt hauptsächlich Themen aus den SWEBOK-Kapiteln 8 und 9. Dabei behandelt die Veranstaltung die Grundlagen mit der Definition von Software Prozessen (Kapitel 8.1. *Software Process Definition*). Außerdem wird auf den Software-Lebenszyklus eingegangen (Kapitel 8.2 *Software Life Cycles*) und das Entwerfen und Verbessern von Software Prozessen (Kapitel 8.3 *Software*

SWEBOK Wissensbereich	Vorlesungs- inhalte	Forschungs- schwerpunkte	Themen für Abschlussarbeiten
1: Software Requirements	RE (SWT)	TraceSEC	Requirements Eng. Traceability Eye Tracking
8: Software Process Engineering	SPE (SWT)		Softwareprozesse
9: Software Engineering Models and Methods	SPE (SWT)		Hybride Methoden Agile Methoden
10: Software Quality	SWQ (SWT)	softXplain TraceSEC	Security & Privacy Erklärbarkeit User-Centered Design User Experience Usability
11: Software Engineering Professional Practive	(SWT)	ViViUse	Soziale Aspekte Vision Videos Soziale Netzwerkan. Stimmungsanalyse

Tabelle 2.1: Themengebiete des Fachgebiets Software Engineering zugeordnet zu SWEBOK Wissensbereichen [4]

Process Assessment and Improvement) gelehrt. Außerdem werden Hybride und Agile Methoden gelehrt (Kapitel 9.4.4 *Agile Methods*).

Die Vorlesung **Requirements Engineering (RE)** behandelt hauptsächlich Themen aus dem SWEBOK-Kapitel 1. In der Vorlesung werden Schritte des Referenzmodells gelehrt und Methodiken zur Ausführung der einzelnen Schritte diskutiert. Die Erhebung von Anforderungen ist in SWEBOK-Kapitel 1.3 *Requirements Elicitation* aufgeführt. Die Verhandlung mit dem Kunden wird in Kapitel 1.4.4. *Requirements Negotiation* behandelt. und die Dokumentation in Form einer Spezifikation ist in Kapitel 1.5.3 *Software Requirements Specification* zu finden. Kapitel 1.6 *Requirements Validation* beschreibt die Methodiken zur Validierung von Anforderungen und der Umgang mit Änderungen während des Projekts ist in Kapitel 1.7.2 *Change Management* aufgeführt.

Einordnung Forschungsprojekte

Das Forschungsprojekt **ViViUse** wurde SWEBOK-Kapitel 11 *Software Engineering Professional Practice* zugeordnet. Diese Zuordnung wurde basierend auf dem Unterkapitel 11.2 *Group Dynamics und Psychology* getroffen. Da ein Kernziel von Vision Videos das Erreichen eines gemeinsamen Verständnisses ist, werden in diesem Forschungsprojekt die Dynamiken von Teams (Kapitel 11.2.1 *Dynamics of Working in Teams/Groups*) und insbesondere die Interaktionen mit Stakeholdern (Kapitel 11.2.4 *Interacting with Stakeholders*) erforscht. Ebenfalls werden in diesem Forschungsprojekt andere Dynamiken, wie der Umgang mit multikulturellen Umgebungen (Kapitel 11.2.6 *Dealing with Multicultural Environments*), erforscht.

Das Forschungsprojekt **softXplain** wurde dem Wissensbereich 11 Software Qualität zugeordnet, da Erklärbarkeit selbst als Qualitätsaspekt angesehen wird [6]. Im Rahmen von softXplain wurde beispielsweise ein Qualitätsmodell für Erklärbarkeit von Deters et al. [7] erstellt, was dem SWEBOK Kapitel 10.1.3 *Models and Quality Characteristics* zuzuordnen ist.

TraceSEC wurde zum einen dem SWEBOK Wissensbereich 1: Software Requirements zugeordnet, da das Unterkapitel 1.7.4 *Requirements Tracing* die eine Hälfte des Forschungsschwerpunkts abdeckt. Die andere Hälfte wird mit dem SWEBOK Wissensgebiet 10 Software Quality abgedeckt, da Security selbst auch ein Qualitätsaspekt ist.

Einordnung Abschlussarbeiten

Das Fachgebiet bietet in verschiedenen Themengebieten Abschlussarbeiten für die Studierenden an. Der Hauptfokus von drei Abschlussarbeitsthemen lässt sich dem Wissensbereich Software Requirements zuordnen. **Requirements Engineering** lässt sich dem gesamten Kapitel 1 zuordnen.

Tracing wird durch das Kapitel 1.7.4 *Requirements Tracing* abgedeckt. **Eye Tracking** wird hauptsächlich für den Bereich *Requirements Tracing* (SWEBOK-Kapitel 1.7.4) genutzt und befasst sich auch mit Unterkapitel 1.8 Software Requirements Tools, da geforscht wird, wie Eye Tracker zur Untersuchung von Requirements Tracing verwendet werden kann.

Das Abschlussarbeitsthema **Software Prozesse** wird durch Wissensbereich Software Engineering Process (SWEBOK-Kapitel 8) abgedeckt.

Die zum Thema **Agile Methoden** und **Hybride Methoden** angebotenen Abschlussarbeiten lassen sich dem Wissensbereich 9: Software Engineering Models und Methods beziehungsweise Unterkapitel 9.4.4. *Agile Methods* zuordnen.

Die Themen zu Qualitätsanforderung wie **Security**, **Privacy**, **Erklärbarkeit** und **Usability** lassen sich dem Wissensbereich Software Quality zuordnen. Außerdem wird auch **User Experience** und **User-Centered Design** betrachtet. Diese Qualitätsanforderungen gehören zum SWEBOK-Kapitel 10.3.1 *Software Quality Requirements*.

Das Kapitel *Software Engineering Professional Practice* deckt vor allem die Abschlussarbeitsthemen zur Zusammenarbeit im Team ab. Die **Stimmungsanalyse** kann dem Unterkapitel 11.2 *Group Dynamics and Psychology* zugeordnet werden und die **Soziale Netzwerkanalyse** noch genauer dem Unterkapitel 11.2.1 *Dynamics of Working in Teams/Groups*.

Vision Videos sind im Unterkapitel 11.2.4 *Interacting with Stakeholders* angesiedelt und **Soziale Aspekte** lassen sich zum Unterkapitel 11.3.3 *Teams and Group Communication* zuordnen.

2.3 Verwandte Arbeiten

Es gibt bereits eine Vielzahl an Arbeiten, die sich mit Spielen und der Vermittlung von Wissen beschäftigen. Einige, für diese Arbeit besonders relevante, werden in diesem Abschnitt erläutert.

Es wurden zuvor am Fachgebiet Software Engineering der Leibniz Universität Hannover bereits zwei Spiele zur Vermittlung von wissenschaftlichen Inhalten entwickelt.

Das Spiel „Die Geschichte der Informatik“ wurde im Rahmen einer Masterarbeit von C.Prasuhn [17] entwickelt. Bei dem Spiel liegt der Hauptfokus auf wichtigen Meilensteinen in der Geschichte der Informatik. Das Spiel ist in vier Kapitel unterteilt und ist für die technischen Gegebenheiten im InfoLab, insbesondere die 270° Projektionsflächen entwickelt worden. In Kapitel 1 werden die Themen mechanische Rechenmaschinen und Binärzahlen behandelt. Kapitel 2 handelt von Analytical Engine als erste Idee eines universell programmierbaren Computers. Hauptinhalt von Kapitel 3 ist der erste funktionsfähige Digitalrechner Z3 von Konrad Zuse. In Kapitel 4 werden die Entwicklung der ersten Programmiersprache, Transistoren und

integrierte Schaltkreise und die Entwicklung des Internets behandelt [17].

Am Fachgebiet wurde im Rahmen einer Bachelorarbeit³ ein weiteres 270° Spiel zur Vermittlung der Logik von Binärzahlen entwickelt. Die Besonderheit bei diesem Spiel ist die Nutzung einer *Xbox Kinect*, wodurch Spielende das Spiel durch Bewegungen und Gesten steuern können, statt mit Maus und Tastatur oder einem Controller.

³Autor und Titel sind unbekannt

Kapitel 3

Anforderungen an das Spiel

In diesem Kapitel werden die verschiedenen Anforderungen an das im Rahmen dieser Arbeit zu entwickelnde Spiel erläutert. Im Anschluss an die Auflistung werden die Anforderungen miteinander verglichen und priorisiert.

3.1 Anforderungen im Überblick

In diesem Abschnitt wird ein Überblick über alle Anforderungen an das Spiel erstellt.

Anzahl der Spieler

Um die Vorstellung der Themen interaktiver zu gestalten, sollen zwei Gruppen, welche jeweils aus mindestens einer Person bestehen, an dem Spiel teilnehmen. Durch das Einbeziehen von mehreren Personen soll die Aufmerksamkeit der Anwesenden gesteigert werden. Bei Veranstaltungen wie *der Nacht, die Wissen schafft*, sind pro Vorstellung teilweise bis zu 30 Personen anwesend. Das reine Zugucken bei Spielen kann schnell langweilig oder frustrierend werden, daher sollen so möglichst viele Personen der Veranstaltung einbezogen werden, auch wenn dies nur durch teaminterne Absprachen erfolgt.

Bedienbarkeit

Durch die große Projektionsfläche und dem Wunsch von mehreren Spielern muss auf die Bedienbarkeit des Spiels geachtet werden. Es sollen Xbox Controller genutzt werden, da diese kompakt, handlich und einfach weiterzureichen sind. Durch die Verwendung dieser Controller, ist ein einfacher Einstieg in das Spiel, ohne Vorerfahrung möglich, da die Controller durch ihr Design auf Steuerungselemente suggerieren. Die komplette Steuerung des Spiels soll vom Spiel selbst vermittelt werden, sodass keine zusätzliche Erklärung von Experten benötigt wird.



Abbildung 3.1: Foto der 270° Projektionsfläche im Infolab

3 Bildschirme

Das InfoLab verfügt über 3 Full HD Beamer, welche zusammen als eine 270° Projektionsfläche bilden. Abbildung 3.1 zeigt die Projektionsfläche im InfoLab. Durch die Größe der Beamerprojektionen kann ein Großteil des Raums zur Darstellung der Spielwelt und ihrer Komponenten genutzt werden und somit die Immersion der Spieler erhöhen. Für ein optimales Spielerlebnis soll daher darauf geachtet werden, die gesamte Anzeigefläche zu nutzen.

Vermittlung von Software Engineering Konzepten

In Abschnitt 2.3 wurde die am Fachgebiet bereits entwickelten Spiele zum Vermitteln von Wissen vorgestellt. Die beiden Spiele vermitteln nur Konzepte zum allgemeinen Thema Informatik. Das im Rahmen dieser Arbeit entwickelte Spiel, soll die Themen und Inhalte des Fachgebiets Software Engineering beinhalten. Die ausgewählten Themen sollen klar ersichtlich sein und Wissen und Konzepte anschaulich vermitteln. Das Behandeln der Inhalte des Fachgebiets ist in diesem Spiel von besonderer Wichtigkeit, da es primär bei Veranstaltungen zur Wissensvermittlung genutzt werden soll.

Anpassung an die Zielgruppe

Das Spiel soll bei Veranstaltungen zur Öffentlichkeitsarbeit der Universität genutzt werden, um das Fachgebiet und den Themenbereich des Software Engineering vorzustellen. Da sich die Zielgruppen der verschiedenen Veranstaltungen unterscheiden, umfasst die Zielgruppe für dieses Spiel ein weites Spektrum. Da auch Personen ohne vorherige Informatik-Kenntnisse bei den Veranstaltungen teilnehmen und somit gegebenenfalls mit dem Spiel in Kontakt treten, soll das Spiel ohne Vorwissen spielbar sein und dennoch

Herausforderungen für alle möglichen Zielgruppen der Veranstaltungen bereitstellen.

Dadurch, dass das Spiel voraussichtlich nur ein einziges Mal gespielt wird, ist auf eine einfache Bedienung und angemessene Erklärungen zu achten. Die Zielgruppe ist nach der Abbildung 2.1 bei den eher Explorern einzuordnen, da diese Gruppe eher als neugierig und wissbegierig gilt. Durch das Interagieren mit der Welt lässt lassen sich Rätsel lösen und wertvolles Wissen erarbeiten.

Spielspaß

Für ein positives Spielerlebnis ist Spielspaß essenziell wichtig. Spiele werden primär freiwillig gespielt, weil sie Spaß machen. Wenn man am Spiel keinen Spaß hat, wird man dieses auch nicht spielen. Ebenfalls wird man solch ein Spiel dann auch nicht anderen Leuten vorstellen. Spielspaß ist ein subjektives Empfinden, da verschiedene Gruppen an verschiedenen Inhalten interessiert sind (Abschnitt 2.1). Durch den Einsatz von Design, Spielmechaniken und bestimmten Komponenten, lassen sich die Bedürfnisse der Spielertypen erfüllen, was bei Spielergruppen des jeweiligen Spielertypen zu einem erhöhten Spielspaß führt.

Spieldauer

Vorstellungen bei den Veranstaltungen zur Wissensvermittlung sind zumeist zeitlich begrenzt. Da die Teilnehmeranzahl durch die Raumgröße begrenzt ist, man jedoch auch möglichst vielen Personen ermöglichen möchte eine Vorstellung zu besuchen, gibt eine zeitliche Begrenzung der einzelnen Vorstellungen. Dies hilft auch Teilnehmenden der Veranstaltungen ihren Besuch einer Vorstellung besser zu planen, da es so feste Vorstellungszeiträume gibt. Aufgrund dessen soll das Spielerlebnis zwischen 20 und 30 Minuten dauern.

Zuverlässigkeit

Bei den Veranstaltungen, an den das Spiel genutzt wird, können nur bestimmt viele Personen pro Vorstellung teilnehmen und die Vorstellungen sind zeitlich begrenzt. Das Spiel soll zuverlässig und ohne Fehler oder Abstürze spielbar sein, da sich diese negativ auf das Gesamterlebnis der teilnehmenden Personen der jeweiligen Vorstellung auswirken würden.

Erweiterbarkeit

Durch das große Spektrum der Zielklasse, ist es von Vorteil, die Möglichkeit zu haben den Schwierigkeitsgrad vor den Vorstellungen auszutauschen, um den an der Vorstellung Teilnehmenden eine angepasste Herausforderung zu bieten. Das Erweitern des Spiels oder das Austauschen von Komponenten wäre sehr nützlich.

Wartbarkeit

Das Spiel soll für längere Zeit zum Vorstellen genutzt werden und sollte daher angenehm in der Wartung sein. Es können sich Inhalte im Software Engineering ändern und somit kann Bedarf entstehen, Inhalte des Spiels in der Zukunft auszutauschen. Das Warten des Programms und Anpassen der Fragen sollte somit möglich sein.

3.2 Priorisierung

Die im vorherigen Abschnitt genannten Anforderungen sind alle zu beachten, jedoch werden im Rahmen dieser Arbeit drei Anforderungen als besonders wichtig angesehen, um die Themen des Fachgebiets während Veranstaltungen zur Wissensvermittlung bestmöglich darzustellen.

Dabei handelt es sich um die Anforderungen **Spielspaß**, die Verwendung der **3 Bildschirme** und die **Vermittlung von Software Engineering Konzepten**.

Der Spielspaß ist wichtig, um Aufmerksamkeit und Interesse an den vermittelten Inhalten zu schaffen. Ein immersives Spiel bietet die Möglichkeit, dass sich Spieler, aber auch Zuschauer von dem Spiel fesseln lassen. Bereitet ein Spiel jedoch kein Spaß, wird es nicht mehr lange gespielt werden.

Um das volle Potenzial des Raums zu nutzen, sollen alle 3 Beamer und somit die vollen 270° verwendet werden. Die Nutzung einer großen Projektionsfläche hinterlässt bei einigen Teilnehmenden einen großen Eindruck, da ein solches Setup eher unüblich ist. Durch die Nutzung der gesamten Fläche, entsteht für Personen, die eine solche Technik nicht gewohnt sind, ein besonderes Spielerlebnis.

Bei Veranstaltungen zur Wissensvermittlung wurden bisher die zuvor entwickelten Spiele genutzt. Zwar sind die Spiele bei den Vorstellungen gut angekommen, jedoch haben diese nur wenig mit dem Themenbereich und Fachgebiet Software Engineering zu tun. Die Vermittlung von Software Engineering Konzepten steht bei dieser Arbeit thematisch im Mittelpunkt. Für das Fachgebiet Software Engineering besteht daher ein großes Interesse daran, Inhalte mittels eines Spiels vorzustellen, da sich dies für Veranstaltungen wie „Die Nacht, die Wissen schafft“ besonders gut eignet.

Kapitel 4

Konzeptentwicklung

In diesem Kapitel wird beschrieben, wie bestimmt wurde, welche Themen in dem Spiel vermittelt werden sollen. Vor dem Workshop wurde bereits eine Vorabrecherche über die Webseite des Fachgebiets Software Engineering durchgeführt. Dabei wurden bereits einzelne Themen identifiziert, welche dann in einem Workshop weiter diskutiert wurden.

4.1 Vorbereitung

Zu Beginn der Entwicklung des Konzepts wurde eine Vorabrecherche durchgeführt. Dabei wurde die Webseite des Fachgebiets nach Themengebieten durchsucht, welche das Fachgebiet repräsentieren [10, 11, 12]. In Unterkapitel 2.2.3 werden die gefundenen Themenbereiche für Abschlussarbeiten aufgezählt und erläutert. Aus der Gesamtheit der aufgezählten Themengebiete wurde eine Vorauswahl von fünf Themen getroffen, welche in dem Workshop als Anregung dienen soll. Die Vorauswahl setzen sich aus folgenden Themen zusammen:

- Requirements Engineering
- Usability
- Softwareprozesse
- Agile Methoden
- Visionsvideos

Es wurde sich bei der Vorauswahl für diese Themen entschieden, da diese alle Wissensbereiche aus Kapitel 2.2.4 abdecken und auch feste Bestandteile der angebotenen Lehrveranstaltung sind.

4.2 Auswahl der Methode

In diesem Abschnitt werden verschiedene Methoden zum Erarbeiten von relevanten Themen für das Fachgebiet Software Engineering diskutiert.

Als mögliche Methoden zur Erarbeitung von Themen standen Workshop, Interview, Umfrage oder die sogenannte Klemmbrett-Methode zur Auswahl. In Tabelle 4.1 werden die Vor- und Nachteile der Methoden aufgelistet.

Das Durchführen eines Workshops liefert durch das Zusammenarbeiten der Teilnehmenden konkrete und gemeinsame Ideen [9]. In Einzelarbeitsphasen können vielen verschiedenen Ideen gebrainstormt werden, welche anschließend in der Gruppe diskutiert werden können [22]. Das Finden eines Termins wird mit steigender Teilnehmeranzahl jedoch zunehmend schwieriger. Hinzu kommt, dass durch offene Diskussionsrunden die individuellen Ideen von Teilnehmenden eingeschränkt werden können, da diese eventuell eingeschüchtert werden könnten und sich somit nicht mehr trauen, ihre Vorschläge zu äußern [9].

Im Gegensatz dazu erhält man bei Interviews viele unterschiedliche Meinungen, welche von der teilnehmenden Person ausführlich erläutert werden können [16]. Das Finden von Terminen ist hier eher einfach, da kein Gruppentermin gefunden werden muss. Aufgrund des fehlenden Austauschs können Ideen jedoch nicht weiterentwickelt werden und die Meinungen bleiben undiskutiert [22].

Bei Umfragen ist es, im Vergleich mit allen anderen genannten Methoden, leicht eine große Teilnehmeranzahl zu erreichen, da Teilnehmer die Umfrage zu beliebigen Zeitpunkten an beliebigen durchführen können [16, 9]. Dadurch kann eine große Anzahl an Antworten generiert werden, welche eine quantitative Analyse erlauben. Außerdem fühlen sich Teilnehmer nicht unter Druck gesetzt und können eigene Ideen ungestört durchdenken und aufschreiben [16]. Wie schon beim Interview können Ideen hier jedoch aufgrund des fehlenden Austauschs nicht diskutiert werden. Zusätzlich fehlen bei dieser Methode auch noch die Gedankenanstöße eines Gesprächspartners und es gibt keine Möglichkeit für Rückfragen [16].

Zu guter Letzt ist auch die Klemmbrett-Methode eine mögliche Vorgehensweise zur Identifikation von repräsentativen Themengebieten. Das ist ein kleines Interview, bei der Kandidaten kurzfristig ohne Termin befragt werden. Der Vorteil hierbei ist, dass diese Methode gut in den Büroalltag vieler Teilnehmenden passt, da das Teilnehmen nur einen geringen Aufwand für Teilnehmende darstellt. Außerdem können viele Teilnehmende mehrfach befragt werden, wodurch eine moderate Anzahl an Ideen entstehen kann. Wie auch schon beim Interview und der Umfrage gibt es jedoch auch hier keinen direkten Austausch mit anderen Teilnehmenden [22]. Zudem können Teilnehmende dadurch, dass die ganze Befragung ohne konkreten Rahmen geschieht, weniger fokussiert sein, was sich in weniger ausgereifteren Ideen widerspiegeln kann.

Method	Pro	Contra
Workshop	+ gemeinsame Ideen + konkrete Ideen + diverse Ideen	- Terminfindung - offene Diskussion schränkt individuelle Ideen ein
Interview	+ viele unterschiedliche Meinungen + fokussierte Ideen + individuelle Termine	- kein Austausch von Ideen - eindimensionale Meinungen
Umfrage	+ einfache Teilnehmerakquise da keine Terminfindung + quantitative Ergebnisse (große Anzahl) + eigene Ideen kommunizieren („sind für sich“)	- kein Austausch von Ideen - eindimensionale Meinungen - keine Gedankenanstöße für Kreativität - keine Rückfragen möglich
Klemmbrett-Methode („Kurz-Interviews“)	+ im Büroalltag möglich + geringer Aufwand für Teilnehmende + viele Teilnehmende + Mehrfachbefragung + moderate Anzahl Ideen	- wenig Austausch von Ideen - weniger fokussiert

Tabelle 4.1: Abwägung der verschiedenen Methoden zur Erarbeitung relevanter Themen und spielerischer Ansätze

Zur Konzipierung dieses Spiels wurde sich für die Durchführung eines Workshops entschieden, da dieser am vielversprechendsten für eine große Anzahl von gut ausgereiften Ideen und Meinungen scheint. Besonders die Kreativität kann durch die offene Diskussion angeregt werden. Das Zusammentragen der Ideen zur offenen Diskussion sorgt für ein gemeinsames Verständnis der Ideen der anderen Teilnehmenden und kann ggf. sogar zu noch weiteren Ideen führen. Hierbei ist darauf zu achten, dass die Diskussion weiterhin fokussiert verläuft. Das Einführen von Einzelarbeitsphasen vor der offenen Diskussion ermöglicht, dass jeder der Teilnehmenden unvoreingenommen seine eigenen Ideen und Meinungen erarbeiten und notieren kann.

4.3 Workshop

In diesem Abschnitt werden das Ziel, die Planung, die Durchführung und die Ergebnisse des Workshops beschrieben.

4.3.1 Ziel des Workshops

Ziel des Workshops war es, die Auswahl an Themen zu erweitern, zu diskutieren und zu priorisieren, sodass eine Menge von ausgewählten Themen hervorgeht, welche das Fachgebiet und den Bereich Software Engineering

repräsentiert. Ebenfalls sollen Ideen für ein grobes Spielkonzept oder einzelne Komponenten von Spielen gefunden werden.

4.3.2 Planung des Workshops

Zur Planung des Workshops werden zuerst die Rahmenbedingungen festgehalten und anschließend ein Ablaufplan erstellt. Die Rahmenbedingungen enthalten Informationen zum Durchführungsort, Uhrzeit, Dauer, benötigten Materialien, Teilnahmebedingungen, Teilnehmenden und zur Terminfindung.

Rahmenbedingungen

Die Teilnehmenden des Workshops sollten Personen mit guten Kenntnissen über die Arbeiten und Themen des Fachgebiets Software Engineering der Universität Hannover sein. Dafür werden alle wissenschaftlichen Mitarbeiter des Fachgebiets eingeladen. Zusätzlich wurden vier Studierende, welche als Tutoren von Lehrveranstaltungen des Fachgebiets viel Erfahrung gesammelt haben, eingeladen. Für den Workshop ist es wichtig, dass die Teilnehmenden Erfahrung mit den Lehr- und Forschungsinhalten des Fachgebiets haben, da im Workshop letztendlich die für das Fachgebiet relevanten Themen erarbeitet und auch anschließend priorisiert werden sollen. Durch Einzel- und Gruppenarbeitsphasen soll garantiert werden, dass alle Teilnehmenden ihre Ideen vorstellen und auch in der Gruppe offen diskutieren können.

Für ein passendes Datum wurde an alle potenziellen Teilnehmenden des Workshops ein Doodle¹ zur Terminplanung geschickt. Dort tragen die Kandidaten für die vorgegebenen Termine ein, ob diese für sie passen oder nicht. Anschließend findet der Workshop an dem Tag und der Uhrzeit statt, an dem die meisten Kandidaten Zeit haben.

Als Ort zur Durchführung des Workshops wird die Bibliothek des Fachgebiets Software Engineering der Leibniz Universität Hannover genutzt, da dieser Ort genug Platz für alle Teilnehmenden bietet, für alle gut erreichbar ist und auch die notwendigen technischen Gegebenheiten vorweist. Die Dauer des Workshops beträgt ca. 120 Minuten und soll in mehreren verschiedenen Sessions unterteilt werden und eine Pause enthalten.

Als technische Voraussetzung für den Raum wird ein Whiteboard benötigt, welches magnetisch ist. Dies ist wichtig, um die Vorschläge und Ergebnisse der Brainstorming-Sessions der einzelnen Kandidaten mittels Pappkarten und Magneten am Whiteboard zu befestigen und somit die Ergebnisse zu visualisieren. Magnete und ein Whiteboard eignen sich sehr gut, da so die Karten problemlos umsortiert werden können. Des Weiteren werden die zuvor genannten Pappkarten, Stifte und Klebepunkte benötigt. Mit den Klebepunkten sollen die Priorisierung erfolgen.

¹doodle.com

Ablaufplan

Begrüßung 10min	 Ziel des Workshops Ablauf des Workshops
Themengebiete finden 25min	 Vorstellung bisherige gefundene Themengebiete (SE-Website)
	 Ergänzung der Themengebiete Brainstorming (alleine)
	 Themengebiete vervollständigen Offene Diskussion
Themengebiete verfeinern 35min	 Verfeinerung der Themengebiete Brainstorming (alleine)
	 Themen gruppieren & zusammenfassen Offene Diskussion
Priorisierung 10min	 Themen priorisieren Jeder 5 Klebezettel
Pause 10min	 Pause
Ideenfindung Spiel 40min	 Ideenfindung spielerische Umsetzung Brainstorming
	 Erkenntnisse zusammentragen Offene Diskussion

Abbildung 4.1: Ablaufplan des Workshops

Der Workshop beginnt mit einer zehnminütigen Begrüßung, bei der das Ziel des Workshops und der Ablauf des Workshops vorgestellt werden. Dadurch konnten sich die Teilnehmenden auf den Ablauf einstellen und wussten, worauf hinausgearbeitet werden soll.

Im Anschluss an die Begrüßung gibt es eine Session zum Finden und eine weitere Session zum Verfeinern von Themengebieten. Dadurch haben Teilnehmende in der ersten Session die Möglichkeit, grobe Ideen zu Themen zu entwickeln und diese zu besprechen. Basierend auf den Ergebnissen der ersten Session, können in der zweiten Session Themenbereiche vervollständigt und weiterentwickelt werden. Diese Vorgehensweise ermöglicht es Teilnehmenden, Ideen auszutauschen und auf Ideen anderer Teilnehmenden aufzubauen.

In der ersten Session zur Themenfindung wird eine kleine Vorauswahl an Themen vorgestellt, welche durch eine Themensichtung der Webseite des Fachgebiets zusammengestellt wurde. [12] Diese Vorauswahl sollte als Grundlage zur Erarbeitung weiterer Themengebiete dienen. Jeder der Teilnehmenden des Workshops sollte vorerst für sich selbst weitere Themengebiete brainstormen und diese auf vorgefertigten Pappkarten notieren. In einer offenen Diskussionsrunde wurden die Ergebnisse der einzelnen Teilnehmenden anschließend an einem Whiteboard zusammengetragen und diskutiert.

In der darauffolgenden Session sollen die erarbeiteten Themengebiete verfeinert werden. Hierbei arbeiten die Teilnehmenden vorerst wieder alleine, um sich bei der Ideenfindung möglichst wenig gegenseitig zu beeinflussen. Die Ergebnisse der Einzelarbeitsphase werden wieder zusammengetragen und in der gesamten Runde diskutiert. In der Diskussionsrunde werden Duplikate entfernt und zusammengehörige Themengebiete in Gruppen zusammengefasst. Hierfür werden erneut die Pappzettel an dem Whiteboard neu angeordnet.

Die Gesamtauswahl der Themengebiete soll in der nächsten Session priorisiert werden. Dafür bekommt jeder der Teilnehmenden fünf Klebepunkte, welche nach Meinung der Teilnehmenden auf die 5 wichtigsten Themen verteilt werden sollen. Themen mit vielen Klebepunkten werden als wichtige Themen des Fachgebiets Software Engineering angesehen.

Als letzte Session sollen nach einer zehnminütigen Pause Ideen für mögliche spielerische Umsetzungen von den erarbeiteten Themen diskutiert werden. Wie in den vorherigen beiden Sessions arbeiten die Teilnehmenden zunächst alleine und anschließend in der gesamten Gruppe. Ziel dieser Session ist es, Anregungen und Erkenntnisse für Spielideen zu sammeln. Dabei müssen noch keine fertigen Spielkonzepte entstehen.

4.3.3 Durchführung des Workshops

Am Workshop nahmen 4 Tutoren und 3 wissenschaftliche Mitarbeiter des Fachgebiets Software Engineering der Leibniz Universität Hannover teil. Es gab eine weibliche Teilnehmerin und 6 männliche Teilnehmer. Der Workshop lief wie in Abschnitt 4.3.2 geplant ab und es wurde von allen Teilnehmenden aktiv mitgearbeitet. Abbildung 4.2 zeigt das mit Karten gefüllte Whiteboard nach der ersten Session zum Finden von Themen. Insgesamt wurden in den Sessions zum Finden und Verfeinern der Themengebiete 69 Karten erstellt. Abbildung 4.3 zeigt das Ergebnis der beiden Themenfindungs-Sessions und der Priorisierungs-Session. In der Session zum Brainstorming für Spielideen wurden 17 Karten erarbeitet.



Abbildung 4.2: Whiteboard nach der Themenfindungs-Session

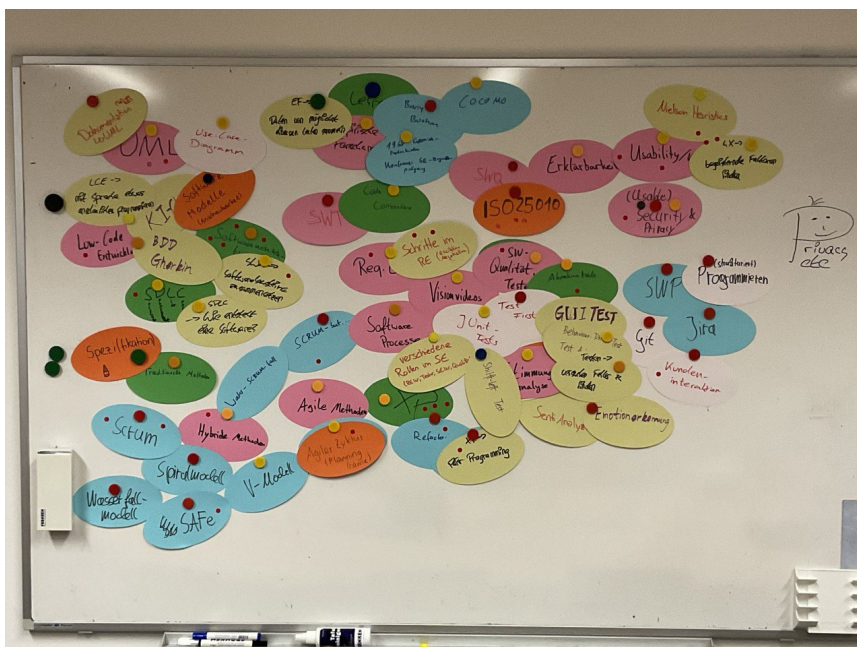


Abbildung 4.3: Whiteboard nach der Priorisierungs-Session

4.3.4 Ergebnisse des Workshops

In diesem Kapitel wird vorgestellt, welche Themen in den Brainstorming-Phasen gefunden wurden und wie diese Themen von den Teilnehmenden priorisiert wurden. Außerdem wird die Gruppierung der priorisierten Themen nach SWEBOK-Wissensbereichen aufgezeigt und erläutert.

Ergebnisse der Priorisierung

Bei der Priorisierung durfte jeder der Teilnehmenden 5 Klebepunkte auf verschiedene Pappkarten kleben. Dies führt zu einer Priorisierung der einzelnen Themen nach Relevanz, welche in Tabelle 4.2 dargestellt wird. Diese Tabelle führt nicht alle im Workshop erarbeiteten Themen auf, sondern lediglich die Themen, welche mindestens einen Priorisierungspunkt erhalten haben.

Themenbereich	# Priorisierungspunkte
Softwarearchitektur Usability / UX	4
ISO 25010 XP	3
Schritte im RE Scrum UML	2
Abnahmetest Agiler Zyklus Hybride Methoden Kundeninteraktion Low-Code Entwicklung Nielsen Heuristiken Pair Programming Refactoring Requirement Engineering SAFe Scrum – but... Software Development Life Cycle SW-Bausteine zusammensetzen SW-Qualität / Testing Useable Security & privacy	1

Tabelle 4.2: Priorisierungspunkteverteilung der Teilnehmenden

Ergebnisse der Gruppierung

In diesem Abschnitt werden die Ergebnisse aus dem vorherigen Abschnitt 4.3.4 gruppiert. Die Themen werden den in Kapitel 2.2.4 vorgestellten SWEBOK-Wissensbereichen zugeordnet. Themen, die keiner dieser Gebiete zugeordnet werden konnten, wurden der Kategorie „Sonstige“ zugeordnet. In Tabelle 4.3 ist die Zuordnung der Themen in die Wissensbereiche und die Priorisierungspunkte der einzelnen Gruppen dargestellt.

Die Themen, die dem Wissensbereich Software Engineering Models and Methods zugeordnet wurden, haben aufsummiert 13 Priorisierungspunkte. Die meisten Themen aus diesem Wissensbereich können dem Unterkapitel 9.4.4 *Agile Methods* zugeordnet werden. Lediglich das Thema UML lässt sich dem Unterkapitel 9.2 *Types of Models* zuordnen.

Themen aus dem Wissensbereich Software Quality haben aufsummiert 10 Priorisierungspunkte. Die meisten Themen spiegeln Qualitätsaspekte von Softwaresystemen wider und sind damit dem Unterkapitel 10.1.3 *Models and Quality Characteristics* zuzuordnen. Die Nielsen Heuristiken können auch dem Unterkapitel 4.1 *General User Interface Design Principles* zugeordnet werden. Da in diesem Fall mit der Themenkarte jedoch explizit auf den Qualitätsaspekt Usability abgezielt wurde, ist es in diesem Fall Teil des Unterkapitel 10.1.3 Das Thema Software Qualität / Testing findet sich im Unterkapitel 10.2.2 *Verification and Validation* wieder.

Der Wissensbereich Software Requirements erzielte aufsummiert 4 Priorisierungspunkte. Dabei wurden Themen aus den Unterkapiteln 1.2 *Requirements Process* und 1.6.4 *Acceptance Tests* angeführt.

Der Wissensbereich Software Engineering Process enthält mit dem Thema Software Development Life Cycle einen Priorisierungspunkt (Unterkapitel 8.2.2 *Software Life Cycle Models*). Der Wissensbereich Software Engineering Professional Practice erhielt einen Priorisierungspunkt. Dabei wurde die Kundeninteraktion (Unterkapitel 11.2.4 *Interacting with Stakeholders*) als Thema genannt.

Es wurden zudem zwei Themen aus dem Unterkapitel 2.3 Software Structure and Architecture und ein Thema aus dem Unterkapitel 13.4.4 High-Level Programming Languages angeführt. Damit enthält die Zusatzkategorie „Sonstiges“ insgesamt 6 Priorisierungspunkte.

Zur Umsetzung im Spiel werden dementsprechend folgende Wissensbereiche umgesetzt:

- Software Engineering Models and Methods
- Software Quality
- Software Requirements

Außerdem werden Themen aus der Kategorie Sonstige abgedeckt.

SWEBOK- Wissensbereich	Themen	Summierte Priorisierungspunkte
9: Software Engineering Models and Methods	Agiler Zyklus	13
	Hybride Methoden	
	Pair Programming	
	Refactoring	
	SAFe	
	Scrum	
	Scrum - but...	
	UML	
	XP	
10: Software Quality	ISO25010	10
	Nielsens Heuristiken	
	SW-Qualität / Testing	
	Usability / UX	
1: Software Requirements	Usable Security & Privacy	4
	Abnahmetest	
	Requirements Engineering	
8: Software Engineering Process	Schritte im RE	1
	Software Development Life Cycle	
11: Software Engineering Professional Practice	Kundeninteraktion	1
Sonstige	Low-Code-Entwicklung	6
	SW-Architektur	
	SW-Bausteine zusammensetzen	

Tabelle 4.3: Gruppierung der Themen in die SWEBOK-Wissensbereiche

4.3.5 Erkenntnisse zur Spiele-Entwicklung

In diesem Abschnitt werden die Ergebnisse aus der Findungsphase für Spielideen des Workshops präsentiert und besprochen. Es wurden in dieser Phase des Workshops viele verschiedene Ideen entwickelt und diskutiert. Dabei wurden die Ideen unterschiedlich konkret und umfänglich ausgearbeitet.

Eine Idee war es bereits bekannte Spiele als Vorlage zu nehmen und deren Spielkonzepte mit Software Engineering Themen nachzubauen. Bei dem häufiger genannten Spiel *Overcooked* sollen in gegebener Zeit möglichst viele gewünschte Gerichte in gegebener Zeit zubereitet und herausgegeben werden. Angepasst an den SE Kontext könnten hier statt Gerichte Abläufe in SE Bereichen wie z.B. der Software Development Life Cycle in bestimmten Reihenfolgen miteinander kombiniert und somit „zubereitet“ werden.

Allgemeine Spieltypen für eine Umsetzung von SE Inhalten in einem Spiel wurde ebenfalls diskutiert. Dabei wurden die folgenden Bereiche genannt:

- Simulation
- Storytelling
- Skill-Basierte
- Jump N Run
- Erkundung
- Spielesammlung

Die Bereiche Erkundung, Simulation und Spielesammlung wurden dabei häufiger genannt. Erkundung hat die Neugierde der Spielenden als motivierende Kraft. In Simulation können Aktionen und ihre Folgen dargestellt werden. Zum Beispiel, was passieren würde, wenn man die Testphase im Entwicklungsprozess weglassen würde. Eine Spielesammlung hat den Vorteil, dass dort viele verschiedene Themengebiete abgedeckt werden können.

Es wurden auch Spielmechaniken besprochen, wie das Zusammenarbeiten von beiden Spielern, um weiterzukommen. Dies ist eine in Coop-Spielen häufig genutzte Technik, um beide Spielende mehr in das Spiel einzubeziehen und das „Wir-Gefühl“ zu stärken. Als mögliches Thema zur Umsetzung mit dieser Technik wurde das Pair Programming genannt. Dabei sollten beide Spielenden zusammen arbeiten, um Code schreiben zu können und somit ein höheres Ziel zu erreichen.

Es wurden auch einige Ideen für Storybasierte Spiele erarbeitet und diskutiert. Eine der Ideen war, die Reise von einem fiktiven Charakter namens „Q“ (für Qualität) durch eine Software Engineering Welt voller Abenteuer, mit vielen NPCs und Hauptfokus auf Storytelling Komponenten. In diesem Zusammenhang wurden auch Rollenspielkomponenten diskutiert, sodass Spielende ihre Charaktere mit verschiedenen Gegenständen ausrüsten können. Dabei kommen die Ausrüstungsgegenstände aus dem Software Engineering Bereich, als Beispiel wurde eine Scrum Master Ritterrüstung genannt. In der Vorlesung Grundlagen Softwaretechnik wird in einer Abbildung ein Scrum Master durch eine Person in Ritterrüstung dargestellt.

Schlussendlich war sich die Teilnehmergruppe des Workshops einig, dass ein kurzes und belohnendes Gameplay für den Einsatzzweck des Spiels geeigneter ist, als ein langes progressives Gameplay. Ebenso wurden auch Storytelling und Simulation als Hauptkomponenten für das Spiel als ungeeignet angesehen, da ein Spieldurchlauf zeitlich auf maximal 30 Minuten begrenzt ist und solche Spieltypen zumeist umfangreicher sind.

4.4 Entwurf des Spielkonzepts

Dieser Abschnitt handelt von der Entwicklung des Spielekonzepts. Dabei wird ein Grundkonzept erarbeitet, welches die groben Rahmen des Spiels vorgibt. Anschließend werden Ideen für mögliche Umsetzungen von Inhalten beschrieben. Danach wird eine kleine Auswahl an Spielen, welche sich besonders zur Umsetzung im Spiel eignen, erläutert. Im darauf folgenden Abschnitt werden weitere geeignete Elemente für das Spiel mit Blick auf die Anforderungen diskutiert.

4.4.1 Konzept des Spiels

Es wurde sich entschieden, ein Spiel mit mehreren Minispielen zu entwickeln. Dies hat den Vorteil, dass auf diese Weise viele verschiedene Themengebiete eingebunden werden können und das Spiel einfach erweiterbar ist. Außerdem unterstützt diese Art von Spiel die Erkenntnisse, die aus dem Workshop gezogen wurden. Bei kleinen Minispielen ist es einfacher, die Aufmerksamkeit der Teilnehmenden aufrechtzuerhalten. Durch die Einbettung dieser Minispiele in eine Story bleibt dabei gleichzeitig der Spielfluss erhalten.

Die Spielenden sollen in dem Spiel die Welt erkunden können und an bestimmten ausgewählten Orten die Minispiele starten können. Ideen für die Minispiele werden genauer in dem Abschnitt 4.4.3 beschrieben. Ein Weg zurück ins Hauptmenü ist wichtig, damit der Spieldurchlauf abgebrochen werden kann, falls die Spielenden das Spiel beenden oder neu starten möchten oder die Vorstellungssession vorbei ist.

Abbildung 4.4 zeigt die erste Skizze, wie der Aufbau eines Spiels mit drei Minispielen geplant wurde. Dort ist nur ein Spieler im Spiel, welcher sich in der Welt, welche hier als das Schwarze Rechteck dargestellt ist, bewegen kann. Dabei kann der Spieler zu den eingezeichneten Orten gehen, um Minispiele zu starten oder ins Hauptmenü zurückzukehren. Die Minispiele sind in dem Bild mit R1-R3 gekennzeichnet. Hierbei steht z.B. das R1, für „Raum 1“.

Eine erste Skizze dieser Abläufe ist in Abbildung 4.5 zu sehen. In der Skizze wird der Weg vom Desktop und Spielstart bis hin zum Starten der einzelnen Minispiele dargestellt. Auf dem Desktop kann man das Spiel starten und kommt somit in das Hauptmenü des Spiels. Im Hauptmenü hat man die Auswahl zwischen Start, Credits und Beenden. Durch Beenden

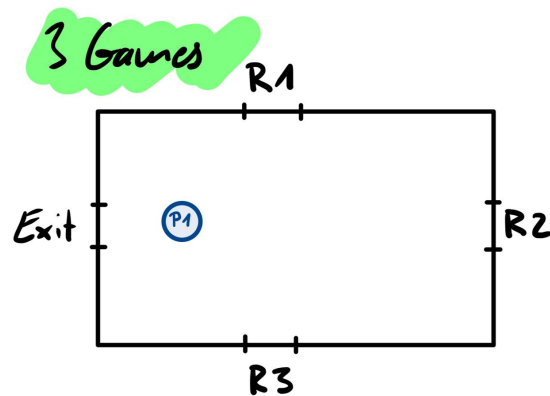


Abbildung 4.4: Skizze für grobes Konzept des Spiels

kommt man wieder auf den Desktop des Computers. Durch die Auswahl von Credits werden einem die Credits angezeigt und durch Start wird das eigentliche Spiel gestartet. Von dort kann man wieder zurück ins Hauptmenü oder eins der Spiele 1 bis 3 starten, indem einer der Räume 1-3 betreten werden. Sobald man ein beliebiges Minispiel gestartet hat, kann man dieses nur durch Durchspielen beenden und landet anschließend danach wieder im Hauptspiel. Von dort aus können erneut Minispiele gestartet oder ins Hauptmenü zurückgegangen werden.

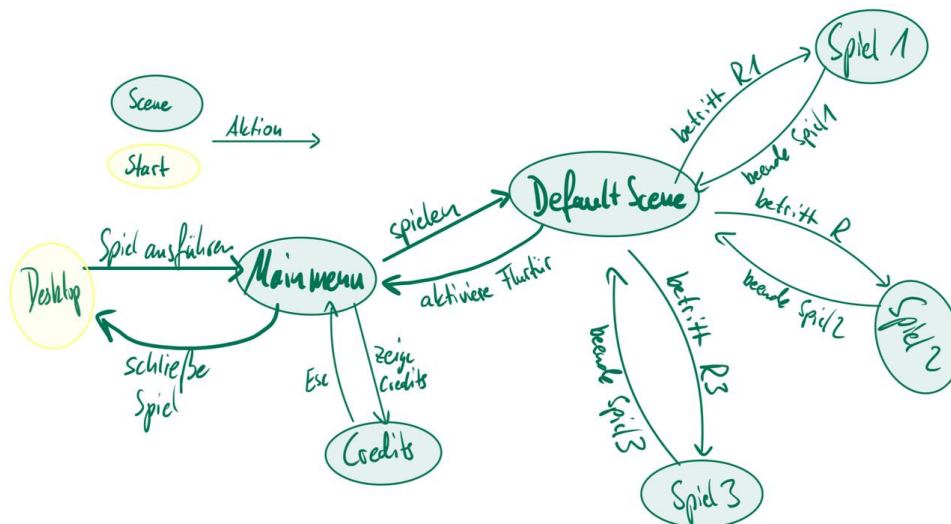


Abbildung 4.5: Darstellung der Szenen mit ihren Übergängen

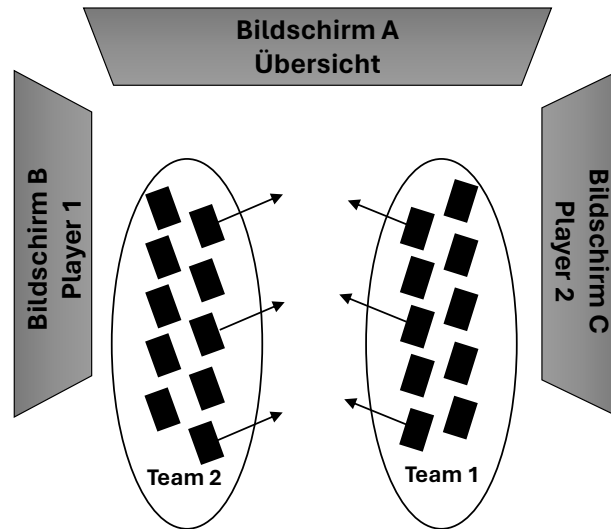


Abbildung 4.6: Räumliche Gegebenheiten inklusive Sitzplan für das InfoLab

4.4.2 Nutzung der räumlichen Gegebenheiten

Um die 270° Projektionsfläche der drei Bildschirme optimal zu nutzen, werden die Teilnehmenden der Vorführung in zwei Teams aufgeteilt und wie in Abbildung 4.6 platziert. Um das Spiel zu spielen, benötigt jedes Team einen Spieler. Es bedient immer genau eine Person des jeweiligen Teams den jeweiligen Teamcontroller. In der Abbildung stellen die 3 großen Trapeze die Bildschirme dar, welche auch mit Bildschirm A bis C beschriftet sind. Bildschirm A ist für beide Teams einsehbar und gibt ein Überblick über das Gesamtgeschehen oder den Punktestand, je nachdem ob man im Hauptspiel oder Minispiel ist. Bildschirm B zeigt die Inhalte für Team 1 bzw. Spieler 1 an. Bildschirm C zeigt die Inhalte für Team 2 bzw. Spieler 2 an. Die Teams sind so platziert, dass sie ihren eigenen Spielbildschirm sehen, jedoch mit dem Rücken zum Spielbildschirm des anderen Teams sitzen. Dies führt auch dazu, dass die beiden Teams sich gegenüber sitzen. Schon durch dieses Setup wird für viele Teilnehmenden das Spielen etwas Besonderes, da eine solch große Projektionsfläche unüblich ist und das Gegenübersitzen den Spielenden ermöglicht, sich bei dem Spiel beim Kommunizieren anzusehen und somit die Mimik der anderen ggf. zu deuten. Außerdem führen auch die großen Projektionsflächen und das Gefühl „in dem Spiel zu sitzen“ zu Begeisterung bei den Spielenden.

4.4.3 Ideen für Minispiele

In diesem Abschnitt werden Ideen für Minispiele diskutiert. Die Ideen für die Minispiele wurden basierend auf Workshopdiskussionen und der eigenen Erfahrung mit Spielen entwickelt. Dafür wird jeweils auf das Spielprinzip, Aktionsmöglichkeiten der Spielenden, Belohnungen, Variationen und Beispiele eingegangen.

Quizduell

Spielprinzip: Beiden Spielenden werden die gleichen Fragen gestellt und sie sollen aus vier möglichen Antworten, die richtige auswählen.

Mögliche Aktionen der Spielenden: Navigation durch Bewegen des linken Analog-Sticks über die einzelnen Antwortmöglichkeiten. Drücken von A zum Auswählen

Belohnung: Ein Punkt für jede richtige Antwort.

Inputvariation:

- Navigation eines Cursors oder zwischen hervorgehobenen Buttons mit Analog-Stick und bestätigen mit A Knopf
- Steuern einer Hand, welche oben auf der X-Achse bewegt werden kann und einen Ball hält. Antwortmöglichkeiten werden als beschriftete Schalen dargestellt und der Ball kann so fallen gelassen werden, dass dieser in die Schalen fällt.
- Zerstören von Objekten: Ein Objekt ist vor den Antworten platziert und muss zerstört werden, um die Antwort auszuwählen. (Schuss auf jede richtige Antwort)
- 1-2-3 Prinzip: Antworten stehen auf Feldern nebeneinander auf dem Boden. Es wird von 5 heruntergezählt und bei 0 muss man auf dem richtigen Feld stehen. Das Feld mit der richtigen Antwort wird durch ein Feuerwerk o.Ä. hervorgehoben.

Beispiel: Wer leitet ein Scrum-Meeting?

Antwortmöglichkeiten:

- | | |
|----------------------|------------------|
| 1. Der Projektleiter | 3. Scrum Master |
| 2. Der Kunde | 4. Captain Scrum |

Schätzspiel

Spielprinzip: Spielende bekommen eine Frage, welche darauf ausgelegt ist, dass die Spielenden keine genaue Antwort darauf haben, sondern Antworten geraten oder geschätzt werden können. Die Person, die Näher an der richtigen Lösung dran ist, gewinnt und bekommt den Punkt.

Mögliche Aktionen der Spielenden: Eingabe einer Freitextantwort mittels Bildschirmtastatur oder bei Zahlen die Eingabe mittels eines Buttons zum Erhöhen und eines Buttons zum Senken.

Belohnung: Wer am nächsten an der Lösung dran ist, bekommt einen Punkt. Bei gleicher Antwort bekommen beide ein Punkt oder keinen.

Beispiel: Wann prägte F.L. Bauer den Begriff des Software Engineerings?
Antwort:

- 1968

Inverse Quiz

Spielprinzip: Spieler bekommen eine Antwort und muss die richtige Frage dazu auswählen.

Mögliche Aktionen der Spielenden: Navigation über die Antwortmöglichkeiten und Einloggen der Antwort.

Belohnung: Ein Punkt für jede richtige Antwort.

Beispiel: Parnas

Fragen:

- Wer erfand das Information Hiding?
- Wer entwickelte den ersten Digitalrechner?
- Wer entwickelte das Konzept der Turing-Maschine?
- Wie lautet im Phasenmodell die letzte Phase?

Elicitation-Spiel

Spielprinzip: Bei dem Spiel soll die Elicitation-Phase im Requirements Engineering veranschaulicht werden. Ein Team ist in der Rolle des Kunden und das andere in der Rolle des Requirements Engineers. Der Kunde beschreibt ein Bild und der Requirements Engineer muss aus einer Menge von Antwortbildern das richtige auswählen. Nach einer festen Anzahl an zu beschreibenden Bildern tauschen die Teams die Aufgaben.

Mögliche Aktionen der Spielenden: Kunden beschreiben und agieren sonst nicht. Requirements Engineer wählt durch Drücken von A-, B-, X-, Y-Buttons ein Bild aus.

Belohnung: Ein Punkt für jede richtig ausgewählte Antwort des Requirements Engineers.

Variation 1: Der Requirements Engineer darf keine Fragen stellen. Dabei erleben die Kunden, wie schwierig es ist, die eigenen Anforderungen ausführlich genug zu formulieren. Dadurch wird den Teilnehmenden gezeigt, wie wichtig die Rolle des Requirements Engineers ist, um den Kunden dabei zu helfen, die Anforderungen richtig zu äußern.

Variation 2: Der Requirements Engineer darf je nach Schwierigkeitsgrad wenige Fragen (z.B. 2-3) stellen. Dabei erleben die Requirements Engineers, wie schwierig es ist, die richtigen Fragen zu stellen, um die Anforderungen aus dem Kunden herauszukitzeln.

Beispiel: In Abbildung 4.7 ist ein Beispiel für das Elicitation-Spiel abgebildet.

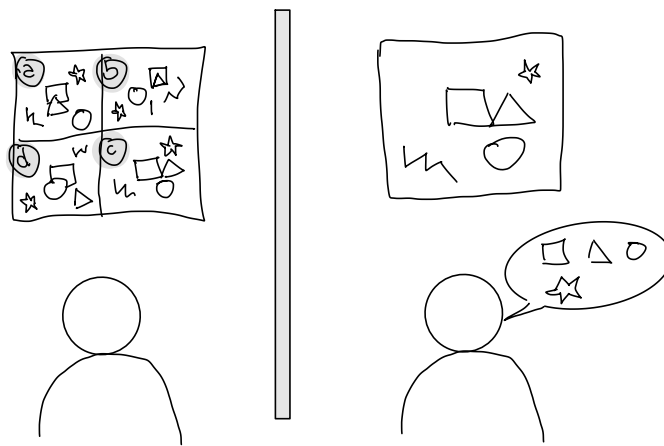


Abbildung 4.7: Visualisierungsbeispiel für das Elicitation-Spiel

Memory Spiel

Spielprinzip: Die Spielenden spielen gegeneinander Memory und decken immer zwei Karten auf, mit dem Ziel ein Paar zu finden. Es handelt sich hier bei einem Paar nicht um zwei identisch aussehende Karten, sondern um zwei Karten, die eine identische Sache beschreiben (z.B. durch Text und Bild). Dieses Minispiel ist nicht einem bestimmten Themenbereich zugewiesen und kann ein beliebiges Thema darstellen.

Mögliche Aktionen der Spielenden: Navigation durch Bewegen des linken Analog-Sticks über die einzelnen Karten. Drücken von A zum Auswählen, um Karten umzudrehen

Variation 1: Beide Teams teilen sich ein Spielfeld. Wenn die umgedrehten Karten kein Paar sind, werden sie wieder verdeckt und das andere Team ist dran. Wenn ein Paar aufgedeckt wird, darf das Team erneut zwei Karten aufdecken. Es gewinnt, wer mehr Paare gefunden hat.

Variation 2: Jedes Team hat ein eigenes Spielfeld und muss alle Paare finden. Bei dieser Variation wird mitgezählt, wie häufig Karten aufgedeckt werden, bis alle Paare gefunden wurden. Nachdem beide Teams alle Paare gefunden haben, gewinnt das Team, das weniger Aufdeckungen benötigt hat. **Belohnung:** Person mit mehr Paaren oder geringeren Anzahl an aufgedeckten Karten bis Ende des Spiels gewinnt

Doordash-Spiel

Spielprinzip: Es stehen mehrere Türen zur Auswahl, durch die man laufen kann. Die Spielenden müssen sich innerhalb von einem festgelegten Zeitraum für eine der Türen entscheiden. Es ist immer nur eine der Türen korrekt.

Mögliche Aktionen der Spielenden: Navigation der Spielfigur nach links oder rechts durch Bewegen des linken Analog-Sticks

Belohnung: Ein Punkt für jedes durchqueren einer korrekten Tür

Variation 1: Statt Punkte für richtige Türen kann es auch Punkte über Zeit geben und für falsche Türen Zeitstrafen.

Variation 2: Falsche Türen können auch für Game Over des Spielenden Sorgen, sodass einfach die Person das Spiel gewinnt, die länger im Spiel bleibt.

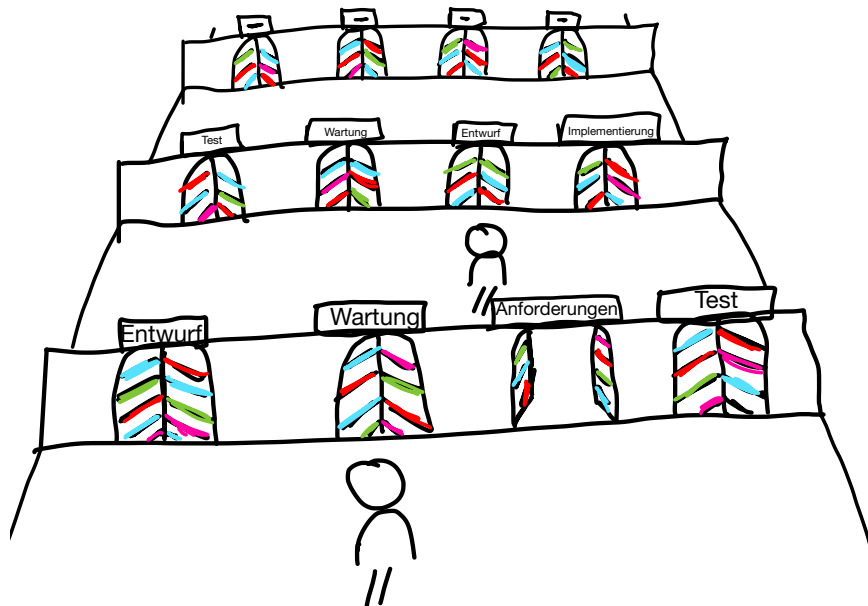


Abbildung 4.8: Visualisierungsbeispiel für das Doordash-Spiel

Beispiel: Durchlaufe die Phasen des Phasenmodells
Beschriftungen der Türen in Runde 1:

- Entwurf
- Anforderungsanalyse
- Test
- Implementierung

Beschriftungen der Türen in Runde 2:

- Entwurf
- Wartung
- Test
- Implementierung

Eine Visualisierung des Beispiels wird in Abbildung 4.8 dargestellt.

Sortieren auf Zeit

Spielprinzip: Phasen in richtiger Reihenfolge auswählen auf Zeit

Mögliche Aktionen der Spielenden: Navigation durch Bewegen des linken Analog-Sticks über die einzelnen Phasen. Drücken von A zum Auswählen und B zum rückgängig machen der letzten Auswahl

Belohnung: Wer zuerst die Reihenfolge korrekt abschließt, bekommt Punkt

Beispiel: Phasenmodell

Phasen zu ordnen:

- Test
- Wartung
- Implementierung
- Anforderungsanalyse
- Design

Nielsens Heuristiken Spiel

Spielprinzip: Es wird eine Beschreibung von einer der Nielsen Heuristiken gegeben, und die Spielenden sollen aus vier Bildern das Bild auswählen, in dem die Heuristik umgesetzt wurde.

Mögliche Aktionen der Spielenden: Navigation durch Bewegen des linken Analog-Sticks über die einzelnen Antwortmöglichkeiten. Drücken von A zum Auswählen einer Antwort.

Belohnung: Ein Punkt für jedes richtig ausgewählte Bild

Beispiel: Heuristik: Error Prevention + *Beschreibung*

- Bild von Abfrage, ob man sicher ist, dass man ein Programm schließen möchte
- Bild von einem QR-Code zum schnelleren Anmelden im WLAN Netzwerk
- Bild von mit einem Ladebalken beim Download
- Bild von einer Webseite mit Breadcrumbs

Erstellen Sie Testfälle zur folgender Funktion:

```
int add (int a, int b)
```

Wert für a:

Wert für b:

Sollwert:

Bisherige Testfälle:

```
add(1,1) -> 2    add(-1,1) -> 0    add(1,-1) -> 0
add(-1,2) -> 1    add(2,2) -> 4
```

Abbildung 4.9: Visualisierungsbeispiel für das Äquivalenzklasse-Spiel

Äquivalenzklassen-Spiel

Spielprinzip: Auswählen von einer Menge an Testfällen, bei der man denkt, dass damit die minimale Menge erreicht ist, um Fehler aus allen Äquivalenzklassen zu finden.

Mögliche Aktionen der Spielenden: Navigation durch bewegen des linken Analog-Sticks über die einzelnen Testfälle. Drücken von A zum Auswählen und B zum rückgängig machen der letzten Auswahl

Belohnung: Spielenden erhalten pro richtigen Testfall einen Punkt und pro falschen wird ihnen einer abgezogen. Aus einer Aufgabe kann nicht mit weniger als 0 Punkten gegangen werden

Nach 1-2 Spielrunden gibt es eine kurze Vorstellung von Äquivalenzklassentests mit einem Beispiel und weitere Spielrunden.

Variation: Für einen erhöhten Schwierigkeitsgrad kann das Spiel ohne Vorauswahl an Testfällen gespielt werden, sodass die Personen selbst Testfälle erstellen müssen

Beispiel: Wähle geeignete Testfälle für die Funktion `int add(int a, int b)`. Versuche dabei, mit möglichst wenigen Testfällen möglichst viele potenzielle Fehler zu finden.

- add(1,1)
- add(2,0)
- add(-1,1)
- add(1,-1)
- add(0,0)
- add(0,1)
- add(-1,-1)
- add(-4,-1)
- add(-2,-1)

Eine Visualisierung der Variante 1 wird in Abbildung 4.9 dargestellt.

4.4.4 Ausgewählte Minispiele

Von allen in Abschnitt 4.4.3 beschriebenen Ideen für Minispiele wurde sich für vier Minispiele entschieden. Die Auswahl der vier Minispiele, welche implementiert werden sind das Quizduell, das Elicitation Spiel, das Usability Spiel und das Memory der Modelle. Die Spiele sind den vier verschiedenen Regionen im Spiel zugeordnet.

Quizduell

In der ersten Region ist das Quizduell mit allgemeinen Fragen über Software Engineering und Informatik Wissen. Bei diesem Spiel soll es darum gehen, ein Mix an Themen der Zusatzkategorie „Sonstige“ darzustellen.

Elicitation-Spiel

Die zweite Region hat das Thema Requirements Engineering und beinhaltet das Elicitation Spiel. Bei dem Elicitation Spiel liegt der Fokus auf der Kommunikation der Spielenden. Dabei muss einer der Spielenden ein Bild beschreiben und der andere aufgrund der Beschreibung das korrekte Bild aus vier Auswahlmöglichkeiten auswählen. Nach einer festen Anzahl an beschriebenen Bildern werden die Rollen der beiden Spielenden getauscht. Da bei diesem Spiel die Elicitation-Phase des Requirements Engineering Prozesses veranschaulicht wird, wird damit der SWEBOK-Wissensbereich *Software Requirements* abgedeckt.

Usability-Spiel

Das Thema der dritten Region ist Usability und enthält dementsprechend auch das Usability-Spiel. Bei diesem Spiel wird eine Beschreibung von einer Nielsen Heuristik gegeben und die Spielenden sollen auswählen, bei welcher der Antwortmöglichkeiten die gefragte Heuristik umgesetzt wurde. Damit sollen die zehn Heuristiken von Nielsen vermittelt werden, welche ein wichtiger Bestandteil der Usability sind. Der Qualitätsaspekt Usability ist einer der Grundbausteine der ISO25010:2011. Daher soll mit diesem Spiel der SWEBOK Wissensbereich *Software Quality* abgedeckt werden.

Memory-Spiel

Der letzten Region ist das Memory der Modelle zugeordnet, da es in dieser Region um die verschiedenen Software-Entwicklungs-Modelle geht. Bei dem Memory der Modelle werden nicht wie für Memory-Spiele üblicher zwei gleich Bilder gesucht, sondern immer eine Karte mit einem Wort und eine Karte mit dem passenden Bild. Ein Pärchen könnte zum Beispiel eine Karte, mit der Aufschrift „Bär“ und eine Karte, mit dem Bild eines Bären, sein. Auf den Karten sind keine Tiere, sondern Software Entwicklungsmodelle dargestellt,

sodass dort zwischen verschiedenen Modellen unterschieden wird. Damit wird der SWEBOK-Wissensbereich *Software Engineering Models and Methods* behandelt.

4.4.5 Umsetzung der Anforderungen

Spielrinzip

Wie schon in den vorherigen Abschnitten beschrieben, soll das Spiel ein Grundspiel und mehrere Minispiele enthalten. Durch diesen Aufbau wird es ermöglicht, das Spiel um weitere Minispiele zu erweitern oder Minispiele einfacher auszutauschen. Bei der Implementierung der Minispiele soll auch darauf geachtet werden, das Austauschen der Fragen und Bilder zu ermöglichen, sodass der Schwierigkeitsgrad des Spiels angepasst werden kann. Details zur Umsetzung der Erweiterbarkeit erfolgt in Abschnitt 5.4.

Raum und Teams

Durch die Anordnung der Teams wie in Abbildung 4.6 ist eine sinnvolle Nutzung der 270° Projektionsfläche gegeben, was zu einem besseren Spielerlebnis führt. Die Anordnung der Teams hat ebenfalls auch einen positiven Einfluss auf den Spielspaß und durch die Nutzung von Controllern, welche im Team herumgereicht werden können, ist eine gute Bedienbarkeit grundsätzlich erreicht. Auf die Steuerung und Bedienung der Controller wird in Kapitel 5 genauer eingegangen.

Minispiele und Spielstand

In den Minispielen treten die Spielenden immer gegeneinander an, und erarbeiten sich durch das Gewinnen von Minispielen Punkte im Gesamtspielstand, welcher während und am Ende des Spiels angezeigt wird. Die Spielstände in den Minispielen und der Gesamtspielstand sollen dafür sorgen, dass die Spielenden eine Herausforderung haben und sich auch mit dem gegenüberstehenden Team messen können. Die Spielstände können aber auch vernachlässigt werden, da diese keinen Einfluss auf das Spielgeschehen haben.

Story

Die Geschichte des Spiels ist, dass die Welt „Digitalia“ von „dem digitalen Zerfall“ heimgesucht wird. Dieser wird im Spiel als dichter Nebel dargestellt, der sich über die ganze Karte streckt. Um die Welt zu retten, müssen die beiden Helden die Minispiele absolvieren und somit die Welt vom digitalen Zerfall befreien. Während des Erkundens des Spielfelds wird der Nebel um die Spielfiguren temporär vertrieben. Die Karte des Grundspiels wurde so designt, dass sie aus mehreren einzelnen Regionen besteht, die durch Brücken

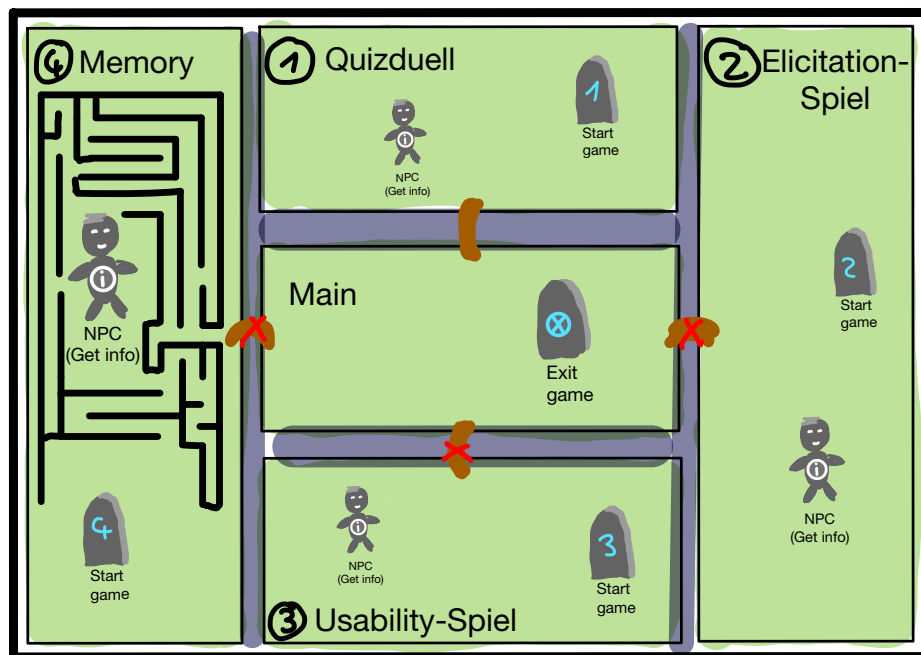


Abbildung 4.10: Konzeptentwurf des Spielaufbaus

miteinander verbunden sind. Auf der Abbildung 4.10 kann man erkennen, dass die Spielwelt in fünf verschiedene Regionen unterteilt ist. Jede der außenliegenden Regionen enthält NPCs, mit denen interagiert werden kann und Objekte an den die Minispiele der Regionen gestartet werden können. In der Region in der Mitte der Welt erscheinen die beiden Spielfiguren, und von dort aus kommt man in die anderen Regionen. Jedoch sind die Wege in die Regionen rechts, unten und links von Spielbeginn an blockiert. Somit kann vor dem ersten Spiel nur die mittlere und obere Region erkundet werden. Sobald einer der Spielfiguren in die Nähe des NPCs kommt, erscheint ein Dialog mit Informationen zum nächsten Minispiel. Nachdem das ein Minispiel durchgespielt wurde, wird der Nebel aus der jeweiligen Region verbannt und die Spielfiguren erscheinen wieder in der mittleren Region. In Abbildung 4.11 ist eine Skizze des Grundspiels nach dem Beenden des ersten Minispiels zu sehen. Es verschwindet ebenfalls die Blockade des Wegs in die nächste Region, in der neue interagierbare Objekte wie NPCs und Minispiele auf die Spielenden warten. Nachdem die ersten drei Minispiele gespielt wurden und nur noch die linke Region vom Nebel betroffen ist, muss dort ein Weg durch ein Labyrinth gefunden werden. Die Schwierigkeit hierbei ist, dass sich die Umgebung im Labyrinth verändert und somit der Weg durch das Labyrinth zum Palast nicht leicht zu finden ist.

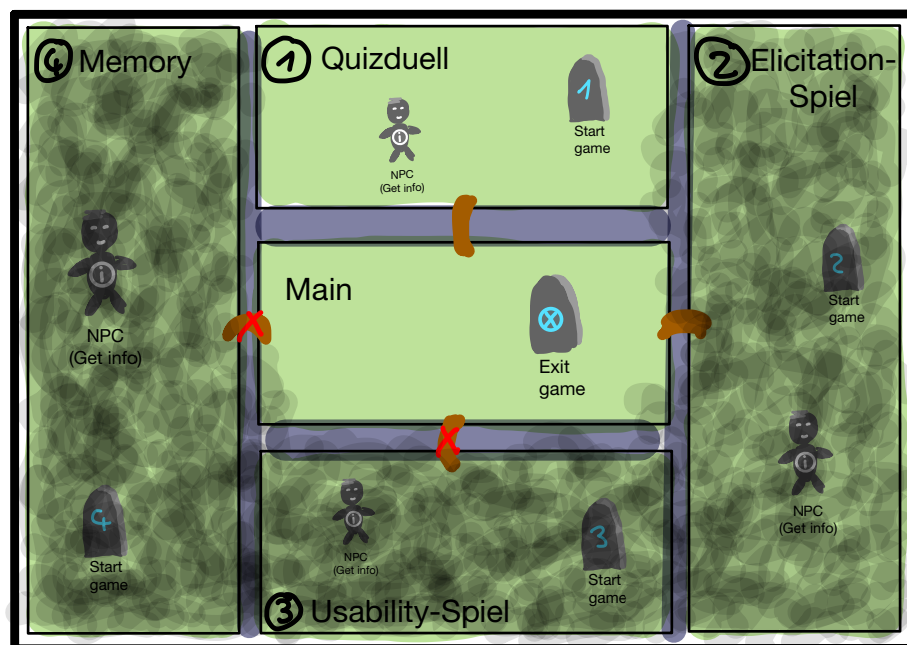


Abbildung 4.11: Grundspiel nach dem Beenden des ersten Minispiels

Kapitel 5

Implementierung

5.1 Game Engine

Zur Entwicklung des Spiels wurde die Game Engine **Unity**¹ verwendet, welche auf das Entwickeln von 2D- und 3D-Spielen spezialisiert ist. Es wurde sich für die Verwendung von Unity entschieden, da bereits in vorherigen Veranstaltungen (Grundlagen der Spieleentwicklung und AppLab) damit gearbeitet wurde. Die Unity Game Engine wurde auch zur Entwicklung von beliebten Spielen wie *Pokémon GO*, *Overcooked* und *Beat Saber* genutzt. Hiermit zeigt sich, dass mit der Unity Engine verschiedene Arten von Spielen entwickelt werden können.

Des Weiteren ist auch die Nutzung des **AssetStore**² sehr praktikabel, sodass nicht alle Elemente wie Bilder für Spieler und Gegenstände selbst geschaffen werden müssen, sondern frei verfügbare genutzt werden können. Der Assetstore ist ein Marktplatz, auf der jeder angemeldete Benutzer verschiedenste Inhalte, wie Musik, Templates oder Werkzeuge, zur Spieleentwicklung anbieten kann. Der Vorteil beim Nutzen von Inhalten aus dem AssetStore ist, dass die Zeit gespart wird, Inhalte selbst zu gestalten und die angebotenen Inhalte häufiger in Projekten genutzt werden und somit auch schon getestet sein können. Dies entspricht dem Prinzip der Wiederverwendung von bewährten Inhalten. Bei den angebotenen Inhalten handelt es sich um Texturen, 3D-Modelle, Sprites, Bilder, Musik, Soundeffekte, Spezialeffekte, einzelne Tools oder sogar ganze Systeme für Spiellogiken. Der Großteil der angebotenen Inhalte muss gegen Entgelt erworben werden, jedoch gibt es auch kostenfreie Inhalte, die zur Entwicklung von Spielen genutzt werden dürfen.

Neben dem offiziellen Unity AssetStore gibt es noch weitere Marktplätze wie **itch.io**³, auf denen Inhalte, zur Spieleentwicklung angeboten werden.

¹<https://unity.com/de>

²<https://assetstore.unity.com/>

³<https://itch.io/>

Die Webseite itch.io ist jedoch nicht nur ein Marktplatz, sondern dort können auch Spiele veröffentlicht werden und es kann in einem Forum diskutiert werden. Durch den Austausch von Usern ist der Einstieg in die Spieleentwicklung etwas leichter, da auf diese Weise typische Fehler schnell entdeckt und beseitigt werden können.

Bei der Entwicklung des Spiels wurden verschiedenste Inhalte aus den zuvor genannten Marktplätzen genutzt. Folgende Inhalte werden genutzt:

- **Tilemaps:** Zum Bau der Umgebung im Grundspiel.
- **Musik:** Mehrere Songs als Hintergrundmusik in den verschiedenen Szenen
- **Sprites:** Als Darstellung des Spielers und von den NPCs.

5.2 Softwarearchitektur

Um den Aufbau und Ablauf des Spiels genauer zu betrachten, werden die einzelnen Szenen in Unity beschrieben, da diese einzelne Spielabschnitte repräsentieren. Das Spiel besteht aus neun Szenen, wovon fünf Szenen für die Minispiele, eine für das Hauptspiel und die restlichen für Intro, Credits und Hauptmenü verwendet werden.

5.2.1 Grundspiel

Die erste Szene, die mit dem Spielstart geladen wird, ist die **MainMenu**-Szene. Wie in Abbildung 5.1 dargestellt, wird in der Szene auf Bildschirm



Abbildung 5.1: Screenshot von Bildschirm A im Hauptmenü

A ein Titelbild mit Titel und Untertitel des Spiels angezeigt. Zudem sind drei Buttons zu sehen, wovon einer bereits ausgewählt ist. Die Buttons sind von oben nach unten mit den Aktionen „Spielen“, „Credits“ und „Beenden“ beschriftet. Durch das Auswählen und Bestätigen des „Spielen“-Knopfs wird die Szene gewechselt und es wird die Intro-Szene geladen. Wird der „Credits“-Knopf ausgewählt und bestätigt, findet ein Szenenwechsel zur Szene Credits statt. Durch Auswählen und Bestätigen von „Beenden“ wird das Spiel geschlossen und man kehrt zurück auf den Desktop.

In der Szene **Credits** wird aufgezeigt, welche Personen in welcher Rolle bei der Entwicklung des Spiels gearbeitet haben.

In der Szene **Intro** färben sich alle drei Bildschirme schwarz und es erscheint auf dem Bildschirm A ein Text, welcher Buchstabe für Buchstabe geschrieben wird. In dem Text wird die Geschichte und das Setting des Spiels erzählt. Diese Szene ist mit einer abenteuerlichen Musik unterlegt, um eine spannende Atmosphäre zu schaffen. Die Texte verblassen nach einer gewissen Zeit und es wird die Szene MainGame geladen.

Die Szene **MainGame** beinhaltet das Hauptspiel, welches der Mittelpunkt des Spiels ist. Sobald diese Szene geladen wird, erscheint eine Nebelwolke über dem ganzen Spielfeld und die Nebelwolke wird innerhalb von ein paar Sekunden immer dichter, sodass diese nicht mehr durchblickt werden kann. Dadurch können Spieler vorerst das gesamte Spielfeld erblicken, um einen groben Überblick zu erhalten, was auf sie zukommen wird. Wie in Abbildung 5.2 dargestellt ist, ist nur in der Mitte der Karte eine nebelfreie Stelle zu sehen. Dort ist der Spawnpunkt der beiden Spielfiguren, welche in einem kleinen Umkreis den Nebel temporär vertreiben können.



Abbildung 5.2: Screenshot von Bildschirm A während des Hauptspiels mit Nebel und ohne Spieler

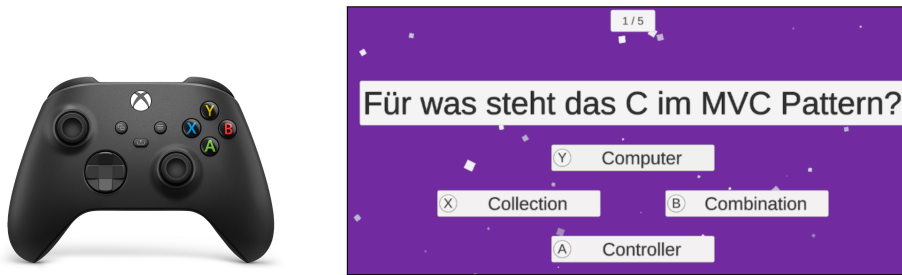
In dieser Szene können die Spielenden durch Tastendruck auf dem Controller eine Spielfigur erscheinen lassen. Sobald die Spielfigur des Teams erschienen ist, kann diese durch Bedienung des linken Analog-Sticks, bewegt werden. Durch das Bewegen der Spielfigur verschwindet auch überall in der Nähe des Spielers der Nebel. In der Spielwelt sind Objekte, mit unterschiedlichen Eigenschaften, platziert. Mit einigen dieser Objekte kann interagiert werden, andere dienen einfach als Blockaden, um die Wege in andere Regionen zu versperren. Um mit Objekten zu interagieren, muss die Spielfigur in die Nähe des Objekts bewegt werden. Manche interagierbare Objekte, wie z.B. NPCs, haben die Aufgabe, den Spielenden Informationen zu vermitteln. Es gibt zudem auch Objekte, die neben der Anzeige von Information auch die Möglichkeiten bieten, Minispiele zu starten oder zurück ins Hauptmenü zu wechseln.

Der Nebel in dieser Szene wird durch ein Partikelsystem generiert, welches zu den Spezialeffekten gehört. Dabei entstehen über die Zeit viele kleine einzelne Partikel, die sich langsam über die Karte bewegen und verteilen, bis eine bestimmte Menge erreicht ist. Diese Partikel haben eine bestimmte Lebensdauer von wenigen Sekunden und wenn diese vorüber ist, verschwinden sie woraufhin neue Partikel generiert werden. Durch diese Umsetzung wirkt der Nebel realistischer, was zu einer besseren Immersion beiträgt.

5.2.2 Minispiele

Es gibt in jeder Region ein Minispiel und mindestens ein interagierbares Objekt mit Hinweisen zum regionalen Minispiel. Mit manchen Objekten muss agiert werden, um Blockaden auf dem Weg zum Minispiel zu lösen. So wird sichergestellt, dass die Spielenden das benötigte Wissen vor dem Starten der Minispiele erhalten.

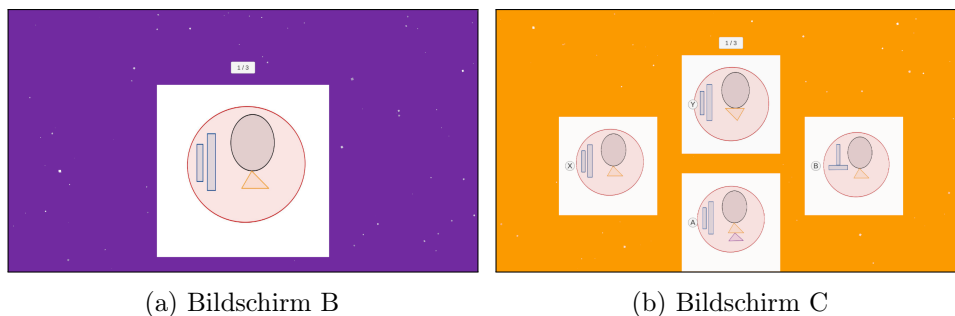
Die Szene **TriviaGame** beinhaltet das erste Minispiel, bei dem beide Teams Fragen bekommen und aus vier Antwortmöglichkeiten die Richtige ausgewählt werden soll. Um die Auswahl der Antworten möglichst intuitiv zu gestalten, wurden die Gegebenheiten des Controller-Designs berücksichtigt. Dafür wurden die Antworten wie die vier Buttons auf der rechten Seite des Controllers angeordnet und dementsprechend gelabelt. In Abbildung 5.3 wird veranschaulicht, wie die Anordnungen der Antworten und Controller-Buttons ist. Bevor das Quizduell Minispiel gestartet werden kann, muss mit dem *Brunnen der Weisheit* interagiert werden. Dadurch erhalten Spielende Informationen zu Konrad Zuse und David Parnas. Diese Hinweise können im Minispiel zum Beantworten der Fragen genutzt werden. Ein Screenshot von diesem Dialog ist im Anhang in Abbildung A.1 zu sehen.



(a) Xbox Series X/S Controller⁴ (b) Screenshot der Antwortfelder im Spiel

Abbildung 5.3: Vergleich von Xbox Controller Knopf Anordnung und Design der Antworten im Spiel

Als zweites Minispiel kommt das **ElicitationGame**, bei dem es um die Anforderungserhebung im Requirements Engineering geht. Dieses Minispiel ist in zwei Szenen unterteilt, da es zwei verschiedene Phasen gibt. In der ersten Phase wird Team 1 ein Bild angezeigt und Team 2 sieht vier Bilder, von denen eins dem Bild von Team 1 entspricht. Team 1 beschreibt das Bild und Team 2 muss aufmerksam zuhören und darf anschließend je nach Spielvariation maximal 3 Fragen stellen. Daraufhin wählt Team 2 von den vier verschiedenen Bildern das korrekte Bild aus. Nachdem alle Bilder von Team 2 ausgewählt wurden, erfolgt ein Wechsel zur zweiten Phase und die Rollen der beiden Teams werden getauscht. Die Auswahl der Antwortmöglichkeiten wurde wie schon bei der vorherigen *TriviaGame* Szene beschrieben implementiert. Abbildung 5.4 zeigt die beiden Bildschirme der Teams während der ersten Frage. Bevor das Elicitation-Minispiel gestartet werden kann, muss mit dem *RE Opi* interagiert werden (siehe Anhang Abbildung A.2). Dieser NPC erklärt den Spielenden, worauf bei der Anforderungserhebung zu achten ist. Dadurch werden die Spielenden darauf vorbereitet, worauf es bei dem nächsten Minispiel ankommt.



(a) Bildschirm B

(b) Bildschirm C

Abbildung 5.4: Screenshot aus dem Spiel mit verschiedenen Ansichten der beiden Teams

⁴https://assets.xboxservices.com/assets/fa/b8/fab86afd-63d6-45a9-862f-798e5e45cda2.jpg?n=111101_Gallery-0_4_1350x759.jpg

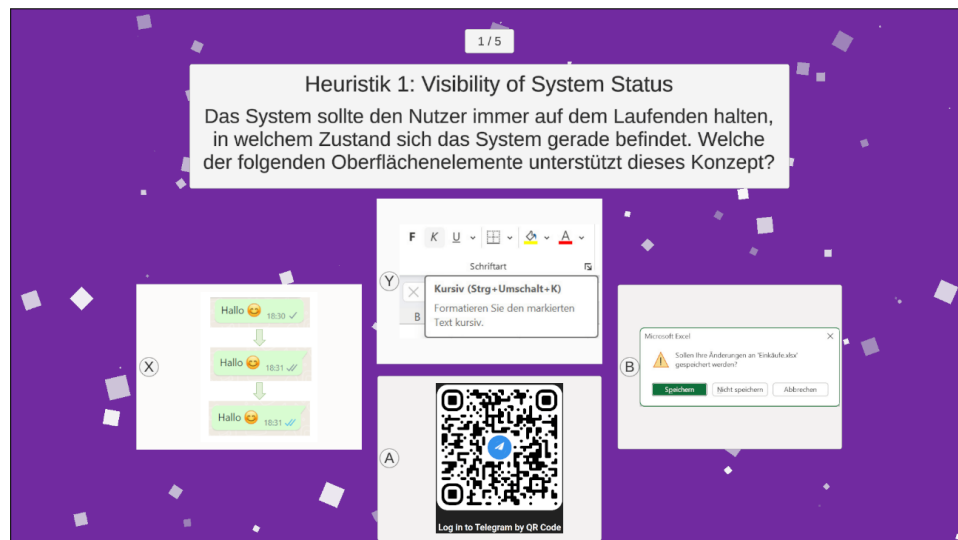


Abbildung 5.5: Screenshot aus dem Usability-Minispiel

Die Szene des dritten Spiels ist die **UsabilityGame**-Szene, bei der die zehn Heuristiken von Nielsen im Fokus stehen. Bei dieser Szene wird im oberen Textfeld die Heuristik genannt und in einem kurzen Text darunter beschrieben. Unter dem Textfeld gibt es die Auswahl von 4 Antwortfeldern, die jeweils Oberflächenelemente aufzeigen. Von diesen vier Auswahlmöglichkeiten soll die richtige ausgewählt werden. Die Auswahl der Antworten erfolgt hier ebenfalls wie bei den vorherigen Minispielen über die A-, B-, X-, Y-Buttons. In Abbildung 5.5 ist beispielhaft die Frage zur ersten Heuristik von Nielsen abgebildet. Bevor das Usability-Minispiel gestartet werden kann, muss mit einer Statue von *Prinzessin Usability* interagiert werden. Da bei diesem Minispiel die benötigten Informationen im Minispiel selbst gegeben werden, vermittelt diese Statue nur die Wichtigkeit des Themenbereichs. Ein Screenshot von diesem Dialog ist im Anhang in Abbildung A.3 zu sehen.

Das letzte Spiel wurde in der **MemoryGame**-Szene umgesetzt. Bei diesem Spiel sollen die Spielenden auf ihren eigenen Spielfeldern die Spielkarten umdrehen und mit möglichst wenigen Aufdeckungen alle Paare der Spielkarten finden. In diesem Spiel wird nicht die Anzahl der gefundenen Paare gezählt, sondern die Anzahl der benötigten Versuche, bis alle Paare gefunden wurden. Es können von den Spielenden immer maximal zwei Karten aufgedeckt werden. Wenn diese ein Paar sind, bleiben sie aufgedeckt, falls nicht, können Karten wieder verdeckt werden, um neue aufdecken zu können. In Abbildung 5.6 ist ein Screenshot des Spiels mitten aus einer Spielrunde abgebildet. Bevor das Memory-Minispiel gestartet werden kann, muss mit einer Vielzahl von NPCs interagiert werden. Dabei vertritt



Abbildung 5.6: Screenshot aus dem Memory-Minispiel

jeder NPC sein Lieblingsmodell/Lieblingsmethodik zur Entwicklung von Software. Beispielsweise erzählt der *V-Opi* vom V-Modell. Insbesondere wird von jedem Modell/Methodik eine Darstellungsform beschrieben, welche im folgenden Minispiel genutzt wird. Ein Screenshot von einem dieser Dialoge ist im Anhang in Abbildung A.4 zu sehen.

5.2.3 Szenen-Übergänge

Um die Übergänge zwischen den Szenen flüssiger zu gestalten, wurden Animationen verwendet. Es werden auf allen drei Bildschirmen die gleichen Animationen abgespielt.

Zur Animation werden die folgenden drei Elemente genutzt: Ein Bild mit dem Logo des Fachgebiets und zwei Rechtecke, welche zusammen exakt die Größe des Bildschirms haben.

Ein Übergang eines Szenenwechsels besteht aus zwei Teilen. Die erste Animation spielt beim Starten einer Szene und die andere Animation beim Verlassen einer Szene. Der erste Teil der Animation wird beim Verlassen der Ursprungsszene abgespielt, der zweite Teil wird beim Starten der Zielszene abgespielt. Bei dem ersten Teil der Animation werden die beiden Rechtecke jeweils vom linken und rechten Bildschirmrand auf den Bildschirm geschoben, sodass sie sich genau in der Mitte des Bildschirms treffen und die Ursprungsszene verdecken. Sobald die Rechtecke ca. 75 Prozent des Bildschirms verdeckt haben, wird das Logo des Fachgebiets langsam eingeblendet und wird kleiner. In dem zweiten Teil der Animation vergrößert sich das Logo wieder langsam auf die Ursprungsgröße und verblasst. Währenddessen schieben sich die beiden Rechtecke aus dem Bild

und sodass die Zielszene zum Vorschein kommt. Die zweite Animation ist das Rückwärtsabspielen der ersten Animation.

Eine Visualisierung des Ablaufs eines Szenenwechsels von der MainMenu-Szene in die Credits-Szene ist in Anhang A.2 dargestellt.

5.3 Umsetzung im InfoLab

In diesem Abschnitt wird darauf eingegangen, wie die Gegebenheiten im InfoLab letztendlich genutzt wurden. Hierbei wird ein Eindruck aus echten Spielsituationen gegeben.

5.3.1 Eindruck aus dem InfoLab

Die Abbildung 5.7 zeigt die Nutzung der kompletten 270° Projektionsfläche während des Spiels⁵. Der mittlere Bildschirm wird als Übersicht über das gesamte Spielfeld genutzt. Dieser Bildschirm kann von beiden Teams eingesehen und zur Orientierung genutzt werden. Die anderen beiden Bildschirme sind teamspezifisch und sollen jeweils nur für ein Team einsehbar sein.



Abbildung 5.7: Überblick des InfoLabs während des Spiels

5.3.2 Sitzordnung

In Abbildung 5.8 ist die Sitzordnung der Teams aus der Vogelperspektive dargestellt. Hierbei wurden beispielhaft insgesamt 22 Stühle verwendet, welche die Raumkapazitäten jedoch noch nicht voll beansprucht haben. Somit wäre eine Umsetzung dieser Anordnung mit 30 Stühlen auch ohne weiteres möglich. Bei dieser Anordnung der Stühle sitzen sich die Teams

⁵Das Spielfeld ist zu diesem Zeitpunkt noch komplett durch Nebel verdeckt

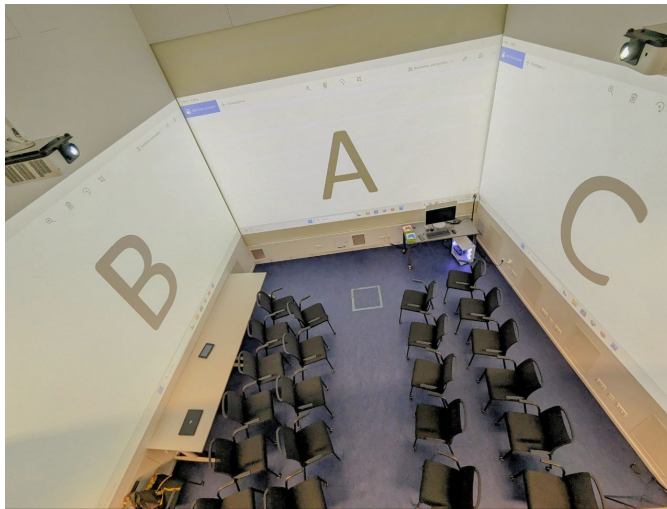


Abbildung 5.8: Sitzordnung der Teams im InfoLab

gegenüber und haben keinen Blick auf den Bildschirm des jeweils anderen Teams. Somit können keine Antworten abgeguckt werden und das Elicitation Spiel kann direkt ohne umsetzen gespielt werden.

Abbildung 5.9 zeigt das Sichtfeld eines Mitglieds von Team 2. Hierbei sind aus der Perspektive des Spielenden nur Bildschirm A und C sichtbar. Durch die Darstellung des Spiels gepaart mit der Positionierung der Spieler wird die Immersion der Spielenden gesteigert, da das Gefühl entsteht, sich direkt in dem Spiel zu befinden.



Abbildung 5.9: Sichtfeld von Mitgliedern des Team 2 während des Spiels

5.4 Erweiterbarkeit

In diesem Abschnitt wird auf die Erweiterbarkeit des Spiels eingegangen. Um das Spiel bestmöglich vom Schwierigkeitsgrad des Spiels und der Aktualität der Themen anzupassen oder einfach um neue Schwerpunkte zu setzen, ist es wichtig, auf eine gute Erweiterbarkeit zu achten.

Für die Spiele Quizduell, Elicitation-Spiel und Usability-Spiel werden in dem Unityprojekt scriptable Objekte genutzt. Diese scriptable Objekte werden verwendet, um die Frage und alle Antwortmöglichkeiten zu einer Frage eines Spiels zu speichern und diese bei Bedarf im Spiel effizient zu laden. In Abbildung 5.10 wird beispielhaft für das Quizduell gezeigt, wie die Fragenobjekte angeordnet sind und wo diese zu finden sind. Jedes Minispiel hat in dem Ressource-Ordner einen eigenen Unterordner für die Fragen des jeweiligen Minispiels.

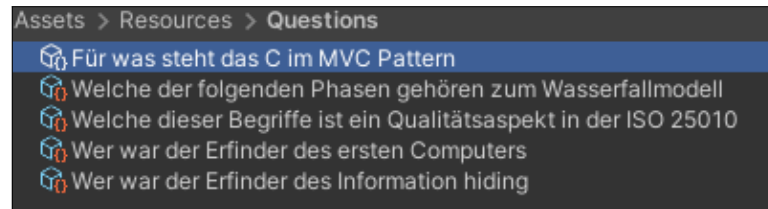


Abbildung 5.10: Darstellung der scriptable Fragenobjekte für das Quizduell Minispiel

Der beispielhafte Aufbau eines Fragenobjekts ist in Abbildung 5.11 zu sehen. Dort wird ein Text für eine Frage angegeben und darunter sind vier Felder für Antwortmöglichkeiten. Beim Erstellen der Objekte ist darauf zu achten, dass unter dem Eintrag *Element 0* die korrekte Antwort einzutragen ist. Die Reihenfolge der restlichen Antworten spielt keine Rolle, da die Antwortmöglichkeiten in den Minispielen zufällig angeordnet werden.

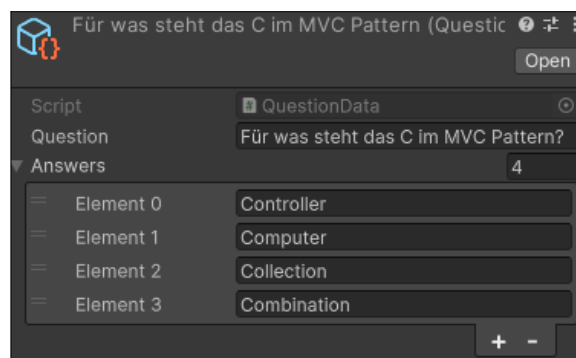


Abbildung 5.11: Beispiel des Aufbaus eines Fragenobjekts für das Quizduell Minispiel

Für die Minispiele Elicitation und Usability sind die Fragenobjekte ähnlich aufgebaut. Dort werden bei den Antwortmöglichkeiten Bilder statt Texte genutzt. Beim Erstellen neuer Fragenobjekte ist darauf zu achten, den richtigen Script-Typen zu hinterlegen. Es ist außerdem zu beachten, dass die neuen Fragen in den richtigen Ordnern platziert werden, da sich die Fragenobjekte der drei Minispiele etwas unterscheiden. Die *Question* Scriptable Objekte sind für das Quizduell und haben als Frage und als Antwort Text. Dagegen werden bei *QuestionElicitation*-Objekten Bilder für die Fragen und Antworten genutzt. Bei den *QuestionUsability*-Objekten werden zwei Textfelder bei der Frage genutzt und für die Antworten werden Bilder genutzt. Somit sind die verschiedenen Scriptable Objekte für die Fragen nicht identisch und müssen im richtigen Ordner abgelegt werden.

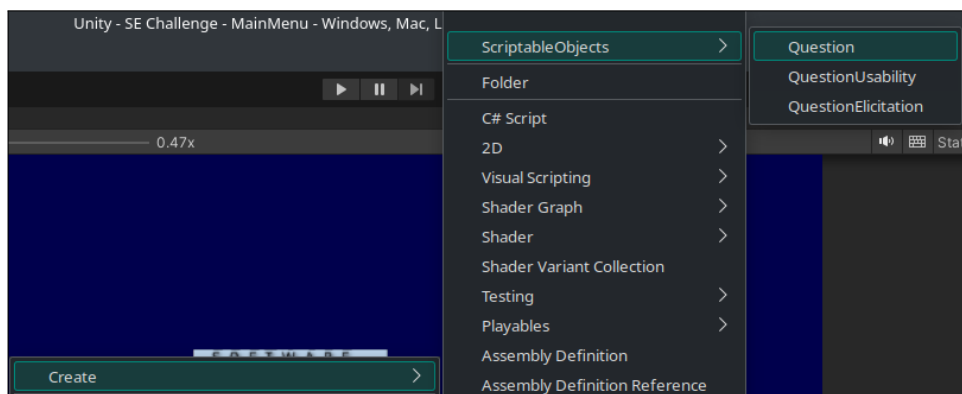


Abbildung 5.12: Ablauf zum Erstellen von neuen Frageobjekten

Um weitere Spiele in das Gesamtspiel einzubinden, kann ein neues Minispiel in einer neuen Szene komplett unabhängig entwickelt werden und anschließend eingebunden werden. Im persistent angelegten GameManager-Objekt können die minispielinternen Spielstände vermerkt werden und der Gesamtspielstand angepasst werden. Das ist notwendig, da diese Informationen bei lokaler Speicherung nach einem Szenenwechsel verloren gehen würden.

Kapitel 6

Studie

In diesem Kapitel wird die Studie zum Bewerten und Testen des Spiels beschrieben. Eine Besonderheit hierbei ist, dass diese nicht mit dem fertigen Spiel durchgeföhrt wurde, sondern mit verschiedenen Inkrementen, sodass die Testdurchläufe der Teilnehmer dazu dienten Stellen mit Fehlern und Verbesserungsbedarf zu finden und zu beheben. Bei der Studie fielen nur kleine Fehler auf und das Spiel blieb vom Grundkonzept her gleich.

6.1 Planung und Durchführung

Ziel der Studie war die Evaluation des Spiels und das Finden von Fehlern während des Entwicklungsprozesses.

6.1.1 Planung

Die Evaluation des Spiels sollte in Form einer Umfrage nach einer Spielrunde durchgeföhrt werden. Hierbei erhielten die Teilnehmenden eine kurze Einführung in die Studie und spielten das Spiel. Anschließend wurden die Teilnehmenden gebeten an einer Umfrage teilnehmen, bei der Sie verschiedene Komponenten des Spiels bewerten sollen. Bei der Einführung gab es eine kleine Vorstellungsrunde, gefolgt von einer kurzen Einweisung und Positionierung der Teilnehmenden im Raum. Darauf folgend sollten die Teilnehmenden das Spiel einmal durchspielen und anschließend einen Umfragebogen ausfüllen. Während die Teilnehmenden spielten wurden sie gebeten laut zu denken, da zeitgleich Notizen bezüglich Fehlern, benötigten Hilfestellungen oder verwirrenden Stellen gemacht wurden.

Es wurden insgesamt 14 Personen zur Studie eingeladen. Im Rahmen dieser Studie wurde keine besonders große Teilnehmerzahl benötigt, da es bei der Studie hauptsächlich darum ging, die Qualität des Spiels durch mehrere Spieldurchläufe während der Entwicklungsphase zu sichern.

Fragen		Antwortmöglichkeiten	
QID	Frage	Von	Bis
Q1	Hat es Ihnen Spaß gemacht, das Spiel zu spielen?	Sehr viel Spaß	Sehr wenig Spaß
Q1.a	Begründung	Freitext	
Q2	Die Story des Spiels war ... zu verstehen	zu einfach	zu schwer
Q3	Das Spiel hängt ab von ...	eigener Entscheidung	Zufall
Q4	Das Spiel ist ...	sehr spannend	sehr langweilig
Q5	Die Spieldauer habe ich als ... wahrgenommen.	zu lang	zu kurz
Q6	Die Spielidee ist ...	sehr originell	wenig originell
Q7	Die Aufmachung des Spiels ist ...	sehr ansprechend	wenig ansprechend
Q8	Der Titel des Spiels passt ...	sehr gut	sehr schlecht
Q9	Die Spielregeln (vom Spiel selbst und den Minispielen) sind ...	sehr klar	sehr unklar
Q10	Die Spielregeln (vom Spiel selbst und den Minispielen) sind ...	vollständig	unvollständig
Q11	Die Spielregeln (vom Spiel selbst und den Minispielen) sind ...	sehr intuitiv	unintuitiv
Q12	Die Steuerung ist ... zu lernen	einfach	schwer
Q13	Der Schwierigkeitsgrad des Spiels war für mich ...	sehr einfach	sehr schwer
Q14	Für mich gab es ...	zu viel Wartezeiten	zu wenig
Q15	Das Spiel war ...	nicht spannend	sehr spannend
Q16	Das Spiel erinnert mich an...	Freitext	
Q16.a	Wegen	Freitext	
Q17	Ich möchte das Spiel ...	nie wieder spielen gerne nochmal spielen nochmal spielen unter folgender Bedingung	
Q18	Vorschläge / Bemerkungen	Freitext	

Tabelle 6.1: Fragenkatalog zur Bewertung des Spielerlebnisses

Für die Umfrage wurde ein Fragenkatalog entworfen, welcher in drei Bereiche unterteilt wurde. Dabei gibt es Fragen zur Demografie, Vorerfahrung mit dem Spielen von Videospielen und Bewertung des Spielerlebnisses. In Tabelle 6.1 sind die Fragen zur Bewertung des Spielerlebnisses aufgelistet. Dabei handelt es sich um 18 Hauptfragen, wobei zwei Fragen noch jeweils eine Zusatzfrage haben. Von den somit insgesamt 20 Fragen und Unterfragen sind vier vom Typ Freitext. Die restlichen Fragen sind über eine 5-Punkte-Skala zu bewerten, wo bei jeder dieser Fragen eine Möglichkeit zur Enthaltung hatte.

6.1.2 Durchführung

An der Studie haben zwölf Personen teilgenommen. Von allen Teilnehmenden waren drei Personen weiblich, und die restlichen männlich. Der Altersdurchschnitt der Teilnehmenden liegt bei 26,3 Jahren. Dabei beträgt die

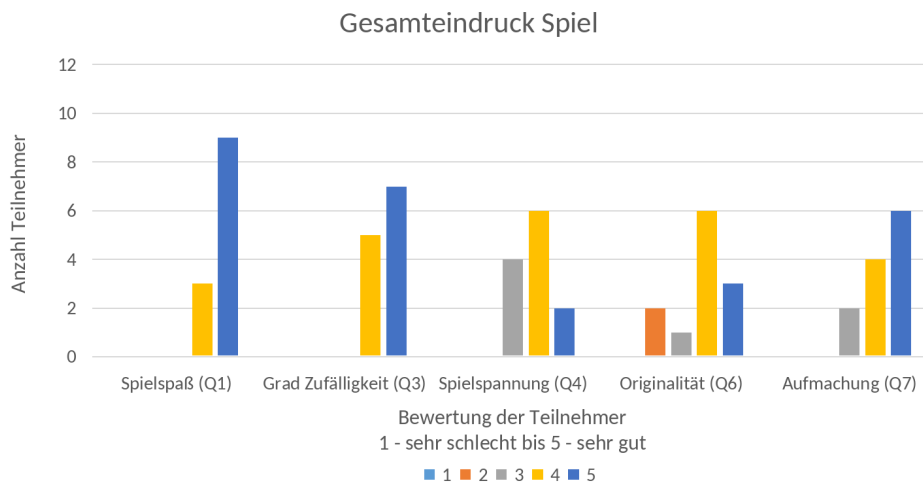


Abbildung 6.1: Ergebnisse der Kategorie Gesamteindruck des Spiels

Standardabweichung einen Wert von 2,1. Alle Teilnehmenden sind Studierende und haben bereits Vorkenntnisse im Bereich Software Engineering und Informatik. Während der Spielesessions wurde die Zeit gestoppt.

6.2 Evaluation

In diesem Abschnitt werden die Ergebnisse der Studie vorgestellt. Dabei werden die Fragen der Umfrage thematisch gruppiert und anschließend analysiert.

Abbildung 6.1 zeigt die Ergebnisse der Fragen, die den **Gesamteindruck** des Spiels repräsentieren. Dazu wurden Fragen zum Spielspaß (Q1), zur empfundenen Zufälligkeit (Q3), zur Spielspannung (Q4), Originalität (Q6) und der Aufmachung (Q7) gruppiert. Durch eine Betrachtung des Median der einzelnen Fragen zeigt sich, dass das Spiel bei den Teilnehmern einen insgesamt positiven Gesamteindruck hinterließ. Bei den Fragen Q1 und Q3 liegt der Median bei 5, was der höchsten Wertung entspricht. Der Median der Frage Q7 liegt bei 4,5 und bei Q4 und Q6 liegt der Median bei 4. Insgesamt wurde nur zweimal eine Bewertung von 2 oder schlechter bei der Frage zur Originalität gegeben. Die gute Bewertung des Spielspaßes ist mit Hinblick auf die in Kapitel 3 festgelegten Anforderungen besonders hervorzuheben. Zwei Teilnehmende haben in den Begründungen bei Frage Q1.a für das Elicitation-Spiel besonderes Lob ausgesprochen. Ein Teilnehmer empfand das Gesamtspiel als „unterhaltsames Erlebnis für Jung und Alt“. Ein weiterer Teilnehmer äußerte sich mit den Worten, „leichte Lerneffekte mit gutem Gameplay verknüpft“. Außerdem wurde die Atmosphäre, insbesondere die Musik und Optik, positiv angemerkt.

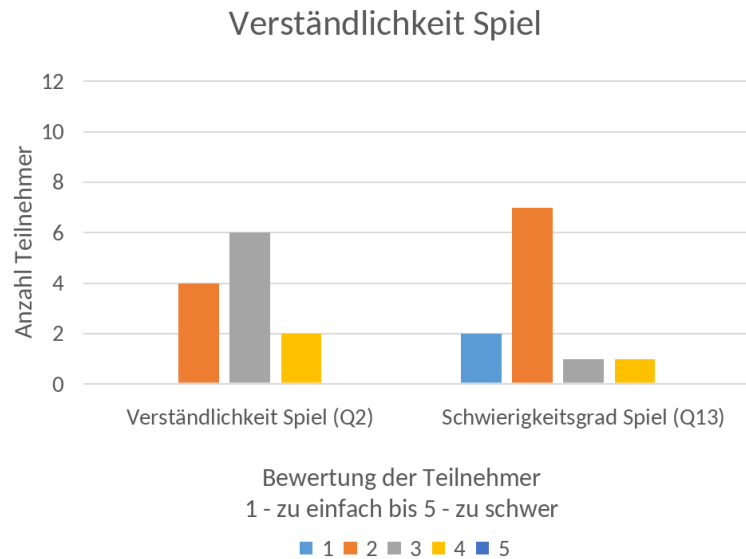


Abbildung 6.2: Ergebnisse der Kategorie Verständlichkeit des Spiels

In der Abbildung 6.2 sind die Ergebnisse der Fragen zur **Verständlichkeit** des Spiels dargestellt. Hierzu zählen die Verständlichkeit des Spiels (Q2) und der Schwierigkeitsgrad des Spiels (Q13). Bei den Fragen ist zu beachten, dass eine mittlere Wertung von 3 dem optimalen Ergebnis entspricht, da die extreme 1 bzw. 5 für die Wertung zu einfach bzw. zu schwer entspricht. Die Verständlichkeit erzielt ein Median von 3, was dafür spricht, dass das Spiel ausgeglichen ist und weder zu einfach noch zu schwer zu verstehen ist. Der Schwierigkeitsgrad des Spiels wurde von den meisten Teilnehmenden als eher einfach empfunden, jedoch auch von einer teilnehmenden Person als eher schwer. Da die Zielgruppe jedoch auch aus Personen ohne Vorkenntnissen besteht und die Teilnehmer der Umfrage alle Vorkenntnisse hatten, ist eine Einordnung als eher einfach positiv anzusehen.

Abbildung 6.3 zeigt die Ergebnisse bezüglich der **Lernbarkeit der Steuerung** (Q12). Diese Frage wurde von elf Teilnehmenden als „5 - sehr einfach“ bewertet. Eine Person hat diese Frage nicht beantwortet. Insgesamt deuten die Ergebnisse darauf hin, dass es keine Probleme mit der Steuerung im Spiel gab.

In Abbildung 6.4 sind die Ergebnisse der Fragen zur Kategorie **Spielregeln** dargestellt. Dazu zählen die Fragen zur Verständlichkeit (Q9), der Vollständigkeit (Q10) und der Intuitivität (Q11) der Spielregeln. Fragen Q9 und Q11 haben beide ein Median von 5 und Frage Q10 hat ein Median von 4. Daraus lässt sich schließen, dass die Spielregeln von den Teilnehmenden als verständlich und intuitiv wahrgenommen wurden. Fast alle Teilnehmenden hielten die Spielregeln außerdem für vollständig. Eine Person bewertete Q10

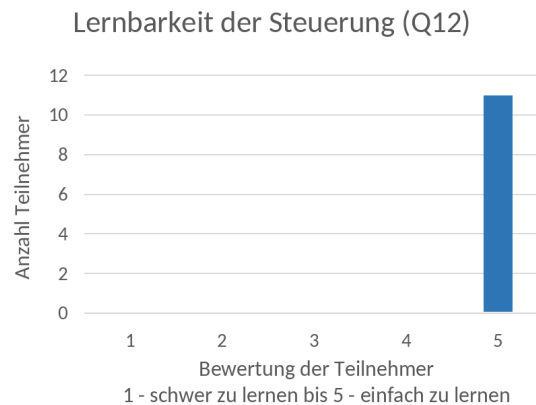


Abbildung 6.3: Ergebnis der Frage zur Lernbarkeit der Steuerung

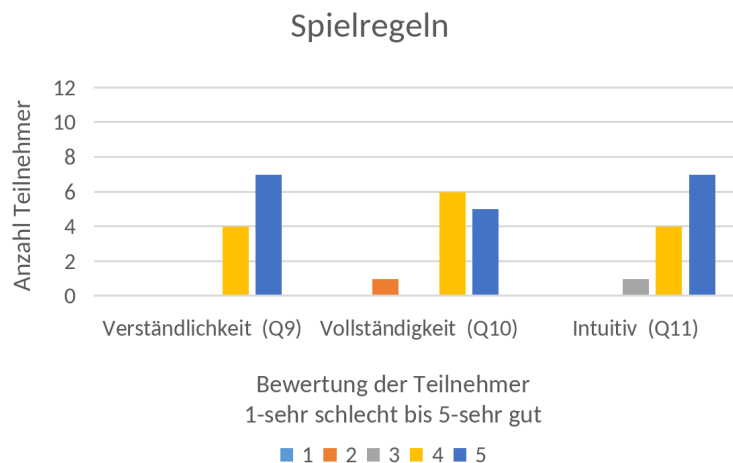


Abbildung 6.4: Ergebnisse der Fragen zur Kategorie Spielregeln

mit eher schlecht, und schrieb unter Bemerkungen (Q18) „Bereit Button vor den Minispielen einfügen“. Die Bemerkung könnte darauf hinweisen, dass Einleitungstexte vor den Spielen nicht vollständig gelesen werden konnte, bevor der Mitspieler das Minispiel gestartet hat.

Abbildung 6.5 zeigt die Fragen der Kategorie **Spieldauer**. Diese Kategorie beinhaltet eine Frage zum Empfinden der Spieldauer (Q5) und eine zum Empfinden der Wartezeiten (Q14). Die Frage Q5 hat in den Bewertungen einen Median von 3 erzielt, was bedeutet, dass die Spieldauer weder als zu kurz noch als zu lang empfunden wurde. Vier der zwölf Teilnehmenden haben angegeben, dass sie gerne länger gespielt hätten. Die Wartezeit wurde von den Teilnehmenden in Q14 ebenfalls als passend eingestuft. Bei der Frage Q14 wurde ein Median von 3 erzielt und zwei Teilnehmende empfanden die Wartezeiten als zu lang. Die Teilnehmenden haben zwischen 17,5 und 35,5

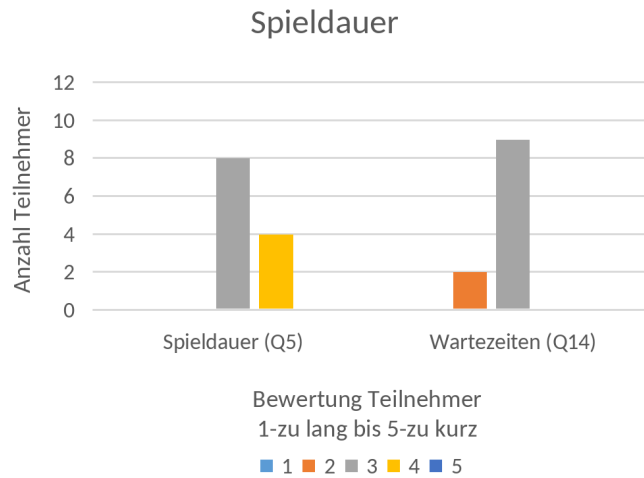


Abbildung 6.5: Ergebnisse zu Fragen der Kategorie Spieldauer

Minuten für einen Spieldurchlauf gebraucht. Im Durchschnitt benötigten die Teams ca. 22 Minuten.

Abbildung 6.6 zeigt das Ergebnis der Frage, für wie passend der **Titel des Spiels** empfunden wird. Der angezeigte Titel des Spiels während der Studie war „Who wants to be a Software Engineer?“. Der Titel des Spiels wurde an die TV-Sendung „How wants to be a millionaire?“ (deutsch: Wer wird Millionär?) angelehnt. Diesen Titel empfanden die meisten Personen als eher passend, jedoch gab es auch eine Person, die diesen Titel als eher unpassend empfand. Inzwischen wurde der Titel zu „Software Engineering Heroes“ geändert. Es wurde noch der Titelzusatz „Der Kampf gegen den digitalen Zerfall“ hinzugefügt.

Bei der Frage an welche Spiele das getestete Spiel erinnert (Q16) wurden folgende Antworten gegeben:

- Memory und Quizduell
- Animal Crossing
- RPG Maker
- Pokemon
- Mario Party
- Zelda
- Stardew Valley
- Vampires Dawn Teil 1 bis 3
- Evoland

Bei dieser Frage wurde von drei Personen Pokémon und von zwei Personen Stardew Valley genannt. Als Begründung warum das Spiel an in Q16 gegebene Antwort erinnert, wurde angegeben, dass dies an der Aufmachung, Optik bzw. Grafik und „niedlichen Atmosphäre“ liegt.

Alle Teilnehmenden der Studie gaben an, dass sie das Spiel gerne

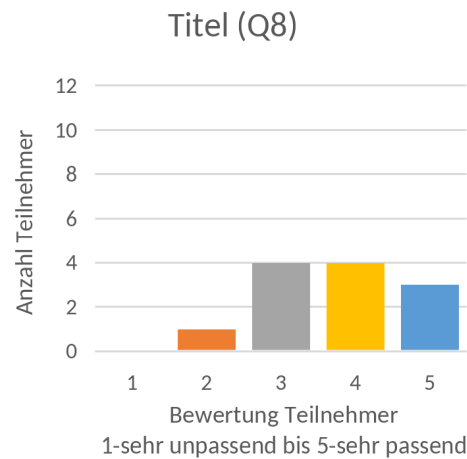


Abbildung 6.6: Ergebnis zur Frage bezüglich des gewählten Titels (Q8)

erneut spielen würden. Sechs davon würden es jedoch nur unter bestimmten Bedingungen erneut spielen wollen. Zu den Bedingungen zählen:

- mehr Minispiele und mehr fragen
- mit anderen Personen nochmal spielen
- mehr Zusatzfeatures
- mehr Interaktionsmöglichkeiten und Eastereggs

6.3 Gefundene Fehler und Wünsche

Bei der Frage nach Vorschlägen und Bemerkungen (Q18) gab es Bemerkungen zum Spiel, zu den Minispielen und zur Wiederspielbarkeit.

Sowohl die Notizen während der Spielsessions als auch Vorschläge und Bemerkungen (Q18) aus der Umfrage, lieferten eine Vielzahl an Verbesserungsvorschlägen und Möglichkeiten zur Verbesserung des Spielerlebnisses. Die Notizen, Vorschläge und Bemerkungen werden in diesem Abschnitt je nach Umsetzung unterteilt.

6.3.1 Umgesetzt

Grundsätzlich zum Spiel wurde vorgeschlagen, die Texte in den Dialogen kleiner zu machen, damit der Kopf beim Lesen nicht so viel gedreht werden muss. Außerdem gab es Probleme mit der Positionierung von einzelnen NPCs, da die Wege im Labyrinth zu eng für zwei Spieler waren und die Interaktionsreichweite der NPCs zu klein war. Deshalb konnten beide Spielenden nicht gleichzeitig mit dem jeweiligen NPC interagieren. Zeitweise

wurde auch der Gesamtpunktestand doppelt erhöht, wenn Antworten kurz nacheinander gegeben wurden. Dieses Problem konnte durch den Einsatz von Semaphoren gelöst werden.

Eine Person wünschte sich für eine bessere Wiederspielbarkeit mehr Minispiele und auch mehr Fragen bei den Spielen 1 und 3. Bei dem Spieldurchlauf, in dem das angemerkt wurde, war das Memory Spiel noch nicht korrekt eingebunden und die anderen Spiele hatten weniger Fragen. Das Problem wurde bis zum nächsten Spieldurchlauf behoben und die Anzahl der Fragen wurde bei allen Spielen erhöht.

Bei einem Durchlauf kam es zu einem Problem beim Starten eines erneuten Spieldurchlaufs. Dort war der Spielstand vom vorherigen Durchlauf noch vorhanden und somit kein Nebel in der Spielwelt. Dieser Fehler wurde ebenfalls behoben.

6.3.2 Nicht Umgesetzt

Es wurde sich eine Aktion zum schnelleren Laufen im Grundspiel gewünscht. Dieses Feature könnte umgesetzt werden, indem beim Drücken und Halten eines bestimmten Knopfes die Laufgeschwindigkeit erhöht wird. Sobald dieser Knopf wieder losgelassen wird, wird auch die Laufgeschwindigkeit wieder auf den Standardwert zurückgesetzt. Es wurde sich ein Bestätigungsmechanismus zum Starten von Spielen gewünscht, sodass Team 1 nicht mehr alleine die Spiele starten kann, sondern noch die Bestätigung von Team 2 benötigt. Außerdem wurde sich gewünscht, den Spielern Namen zu geben und die Farben der Spielfiguren selbst auszuwählen. Diese Features würden zu einer Personalisierung des Spielers beitragen, wodurch dieser sich zur Spielfigur mehr verbunden fühlt und das Gefühl von Immersion verstärkt wird. Im Intro wurde sich eine Animation wie im Starwars-Intro gewünscht. Ebenfalls wurde sich gewünscht, dass der Introtext vorgelesen wird. Zu den Minispielen wurde sich mehr visuelles Feedback bezüglich richtigen und falschen Antworten gewünscht. Die in diesem Abschnitt genannten Verbesserungsvorschläge und Wünsche sind generell alle umsetzbar, jedoch war es im zeitlichen Rahmen dieser Arbeit nicht mehr möglich alle hier aufgezählten Vorschläge zu berücksichtigen.

Kapitel 7

Diskussion

In diesem Kapitel wird auf die Umsetzung der Anforderungen aus Kapitel 3 und auf Limitierungen im Workshop und der Studie eingegangen.

7.1 Umsetzung der Anforderungen

Die erste Anforderung an das Spiel war der **Spielspaß**. Um für die Spielenden ein möglichst gutes Spielerlebnis zu schaffen, wurden ein Teil der in Abschnitt 2.1.1 genannten Stimuli umgesetzt. Das Hauptspielprinzip basiert auf den Anreiz der *Entdeckung*. Durch die Geschichte des Spiels in einer fiktiven Welt mit einer Bedrohung, werden Anreize in den Bereichen **Empfinden**, *Fantasie* und *Handlung* gegeben. Durch Nutzung von situationsabhängig passender Musik und Spezialeffekten wie dem Nebel wird die Immersion verstärkt und die zuvor genannten Stimuli werden intensiviert. Da Spielspaß sehr subjektiv ist, kann dieser zwar nicht objektiv gemessen werden, jedoch wurden die subjektiven Meinungen der Teilnehmer erhoben. Die Umfrageergebnisse bezüglich des Spielspaßes zeigen, dass die Teilnehmenden viel Spaß beim Spielen des Spiels hatten, was darauf schließen lässt, dass diese Anforderung angemessen umgesetzt wurde.

Wie gefordert wurden die **drei Bildschirme** zur Darstellung des Spiels genutzt. Das Spielkonzept wurde auf die 270° Projektionsfläche und die Sitzanordnung maßgeschneidert.

Zur **Vermittlung der Konzepte des Fachgebiets Software Engineering** wurden durch eine Vorabrecherche der Fachgebiets Webseite und einem Workshop relevante Themen identifiziert. Die gefundenen Themen wurden mit Hilfe des SWEBOK [4] gruppiert und priorisiert. Im Spiel wurden letztendlich drei der SWEBOK-Wissensbereiche umgesetzt. Bei den umgesetzten Kapiteln handelt es sich um Kapitel 1: Requirements Engineering, Kapitel 9: Software Engineering Models and Methods und Kapitel 10: Software Quality.

Die Anforderung an die **Zuverlässigkeit** und Qualität des Spiels wurde

durch die Testdurchläufe während der Studie sichergestellt. Bei der Systemarchitektur wurden durch einen modularen Aufbau die Anforderungen, **Erweiterbarkeit** und **Wartbarkeit** berücksichtigt.

Bei der **Anpassung an die Zielgruppe** wurden primär die Spielertypen Explorer und Socializer aus Kapitel 2.1.2 berücksichtigt. Somit liegt der Hauptfokus auf dem Interagieren mit anderen Spielern und der Spielwelt. Die Anpassung wird durch die in diesem Abschnitt zuvor genannten Stimuli unterstützt. Des Weiteren kann durch Änderung der Fragen das Spiel auch an weitere Zielgruppen angepasst werden. Zum Beispiel durch Anpassen des Schwierigkeitsgrads der Fragen der einzelnen Minispiele.

Das Spiel wurde so konzipiert, dass die geforderte **Anzahl an Teilnehmern** (2) an dem Spiel teilnimmt. Durch die Verwendung von Teams werden indirekt mehr Personen beim Spielen des Spiels involviert. Damit keine Eingewöhnung an die **Bedienbarkeit** des Spiels nötig ist, wurde bei der Konzipierung und Umsetzung des Spiels darauf geachtet ein intuitives Design zu verwenden. Das Design der Controller wurde genutzt und die Anordnung der Antworten bei drei und vier Spielen im gleichen Muster wie die Buttons auf dem Controller angeordnet. Die Ergebnisse der Studie zeigen, dass die Steuerung sehr intuitiv ist. Die Anforderung an die **Spieldauer** von 20 bis 30 Minuten wurde eingehalten. Das Einhalten der Spieldauer wurde durch die Testdurchläufe während der Studie bestätigt und die Spieldauer ist laut den Ergebnissen der Studie angemessen.

7.2 Limitationen

In diesem Abschnitt wird die Gültigkeit der Ergebnisse eingeordnet und insbesondere darauf eingegangen, welche Gegebenheiten die Ergebnisse beeinflusst haben könnten.

7.2.1 Workshop

Die geringe Anzahl an wissenschaftlichen Mitarbeitern als Teilnehmer des Workshops kann dazu geführt haben, dass nicht alle Themen erarbeitet wurde, die für das Fachgebiet relevant sind. Es nahmen beispielsweise nicht aus jedem Forschungsprojekt Mitarbeiter teil. Dadurch kann auch die Priorisierung der Themen beeinflusst worden sein. Durch das Einbeziehen von erfahrenen studentischen Hilfskräften wurde versucht, die geringe Anzahl der verfügbaren wissenschaftlichen Mitarbeiter auszugleichen.

7.2.2 Studie

Da die Anzahl der Teilnehmenden (12) an der Studie für eine quantitative Analyse mittels Bewertungsbogen gering ist, konnten keine statistischen Tests durchgeführt werden. Dadurch konnte die Qualität des Spiels nicht

statistisch bewiesen werden. Da das Ziel des Workshops primär das Finden von Fehlern war und die Umfrage nur die Eindrücke der Probanden einfangen sollte, wurde eine statistische Analyse auch nicht angestrebt.

Die Teilnehmenden der Studie entsprachen demografisch gesehen nicht der Zielgruppe. Beispielsweise war der Altersdurchschnitt und das Vorwissen der Teilnehmenden deutlich über dem der Zielgruppe. Das Akquirieren von Schülern war im Rahmen dieser Studie aus organisatorischen Gründen nicht möglich.

Da der Teilnehmerkreis der Studie hauptsächlich aus Freunden und Bekannten besteht, kann der *Social Desirability Bias* [14] verstärkt worden sein. Um diesen Einfluss zu verringern, wurden die Probanden gebeten ehrliches Feedback zu geben.

Eine weitere Verzerrung der Ergebnisse der Studie kann durch die Teilnahme mancher Probanden an Workshop und Studie entstanden sein. Durch Berücksichtigung der Ideen der für die Studie wieder eingeladenen Probanden können die Ergebnisse der Studie beeinflusst worden sein.

Insgesamt liefert die Studie nur einen ersten Eindruck vom Spiel. Um eine statistische Aussagekraft zu gewährleisten, müsste eine größer angelegte Studie durchgeführt werden.

Kapitel 8

Zusammenfassung und Ausblick

Im letzten Kapitel dieser Arbeit werden zunächst alle wichtigen Ergebnisse dieser Arbeit zusammengefasst. Abschließend wird ein Ausblick auf mögliche weiterführende Forschung gegeben.

8.1 Zusammenfassung

Für das Fachgebiet Software Engineering sollte ein Spiel zur Vermittlung von Themengebieten aus dem Bereich Software Engineering entwickelt werden. Dieses Spiel soll bei Veranstaltungen wie „Die Nacht, die Wissen schafft“ zum Einsatz kommen. Um das Spiel konzipieren und entwickeln zu können, mussten vorerst relevante Themen identifiziert werden. Hierfür wurden die öffentliche Webseite des Fachgebiets gesichtet und eine Vorauswahl an potenziell relevanten Themen zusammen gestellt. Mit Hilfe dieser Vorauswahl an Themen wurde ein Workshop mit Mitarbeitern des Fachgebiets durchgeführt. Dabei wurden weitere Themen identifiziert und eine Priorisierung aller Themen wurde mittels Priorisierungspunkten vorgenommen. Die priorisierten Themen wurden mit Hilfe des SWEBOK in Wissensbereiche gruppiert und die Priorisierungspunkte der Themen jedes Wissensbereichs wurde aufsummiert. Dadurch wurden Themen der Bereiche Software Models and Methods, Software Quality und Requirements Engineering als relevante Themen für das Fachgebiet bestimmt. Um die relevanten Themen zu vermitteln, wurde ein Spiel mit vier Minispielen konzipiert. Drei der Minispiele decken jeweils Themen eines Wissensbereichs ab. Das vierte Minispiel deckt ein Mix relevanter Themen anderer Wissensbereiche ab.

Zur Implementierung des Spiels wurde die Game Engine Unity verwendet, welche das Einbinden geeigneter Assets ermöglichte. Darauf folgend wurde eine Studie durchgeführt, in der 12 Probanden das Spiel testweise spielten. Bei der Studie wurden kleine Fehler entdeckt, welche dann iterativ

behalten wurden. Außerdem wurde während der Studie ein subjektives Meinungsbild der Probanden eingeholt, welches darauf schließen lässt, dass das Spiel die gewünschten Anforderungen angemessen umsetzt.

Im zeitlichen Rahmen dieser Arbeit war es nicht möglich, weitere Themenbereiche im Spiel umzusetzen und eine umfangreichere Studie durchzuführen.

8.2 Ausblick

Neben den im Spiel umgesetzten Minispielen wurden noch weitere Ideen erarbeitet, die jedoch nicht eingebunden werden konnten, ohne den zeitlichen Rahmen des Spiels zu überschreiten. Die zusätzlichen Minispiele könnten als Bonus-Minispiele umgesetzt werden, welche nicht direkt zu Story des Spiels gehören. Mit den Bonus-Minispielen könnte die Spieldauer von 30 Minuten überschritten werden. Das könnte für Veranstaltungen mit längeren Vorstellungszeiträumen nützlich sein.

Ebenso wurden nur drei Wissensbereiche und ein Mix aus verschiedenen Bereichen umgesetzt, sodass weitere Bereiche in der Zukunft hinzugefügt werden könnten. Auch diese könnten mit Hilfe von Bonus-Minispielen umgesetzt werden. Da bei unterschiedlichen Veranstaltungen, auf den das Spiel gespielt werden soll, unterschiedliche Themen im Fokus liegen könnten, ist eine Anpassung der Spiele und Fragen möglich. Das Anpassen der Fragen und Spiele kann auch zur Veränderung des Schwierigkeitsgrades der Fragen genutzt werden. Hierfür müssen die Fragen wie in Abschnitt 5.4 beschrieben angepasst und eine neue Version des Spiels gebaut werden.

In Abschnitt 6.3.2 wurden von Probanden der Studie gewünschte Features erwähnt, welche im Rahmen dieser Arbeit zeitlich nicht umgesetzt werden konnten. Beispielsweise könnten die Personalisierung der Spielfiguren und das beschleunigte Laufen in das Spiel integriert werden.

Da die in dieser Arbeit durchgeführte Studie aufgrund der geringen Teilnehmeranzahl keine statistische Aussagekraft hat, könnten durch eine große Studie die verschiedenen Qualitätseigenschaften des Spiels überprüft werden.

Anhang A

Eindrücke aus dem Spiel

A.1 Dialoge



Abbildung A.1: Dialog vor dem ersten Minispiel

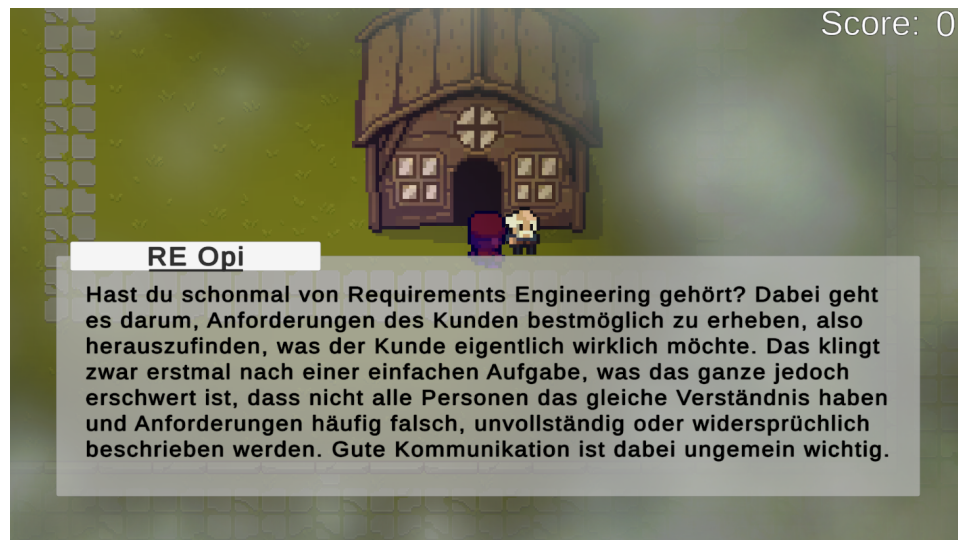


Abbildung A.2: Dialog vor dem zweiten Minispiel

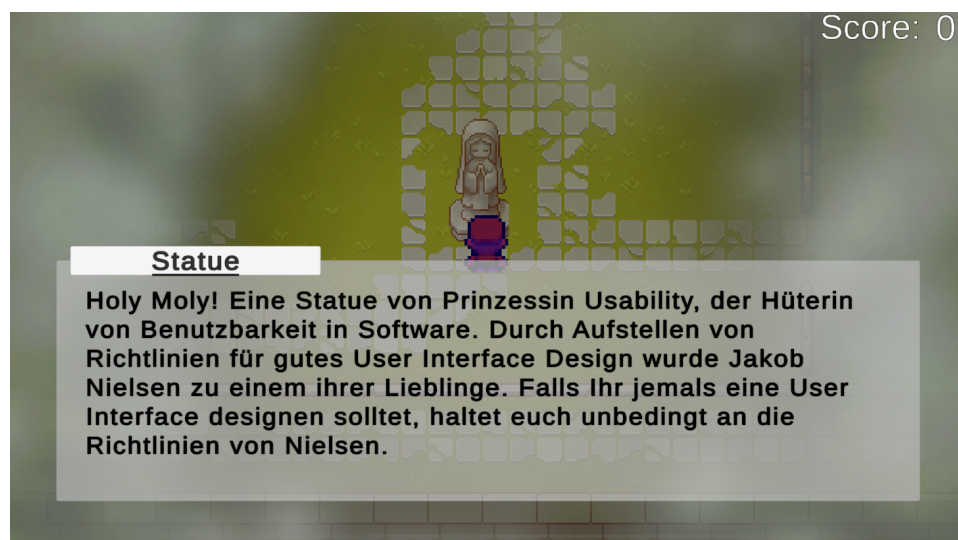


Abbildung A.3: Dialog vor dem dritten Minispiel

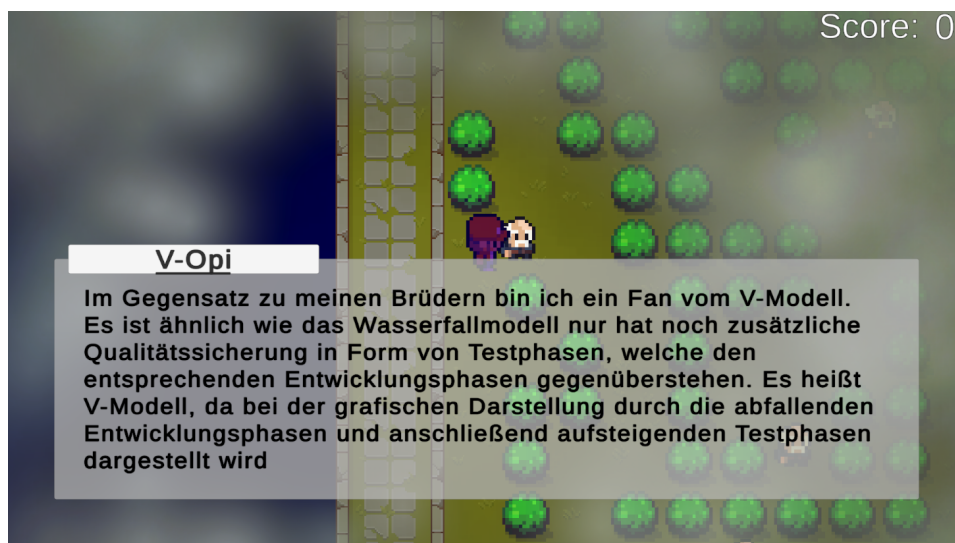


Abbildung A.4: Einer der Dialoge vor dem vierten Minispiel

A.2 Szenenwechsel



Abbildung A.5: Szenenwechsel – Teil 1.1



Abbildung A.6: Szenenwechsel – Teil 1.2

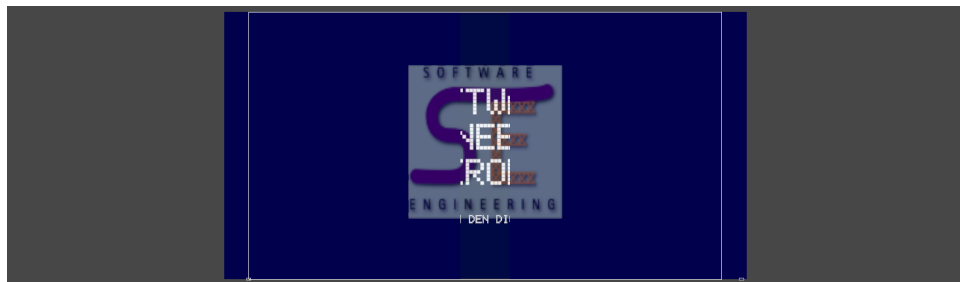


Abbildung A.7: Szenenwechsel – Teil 1.3

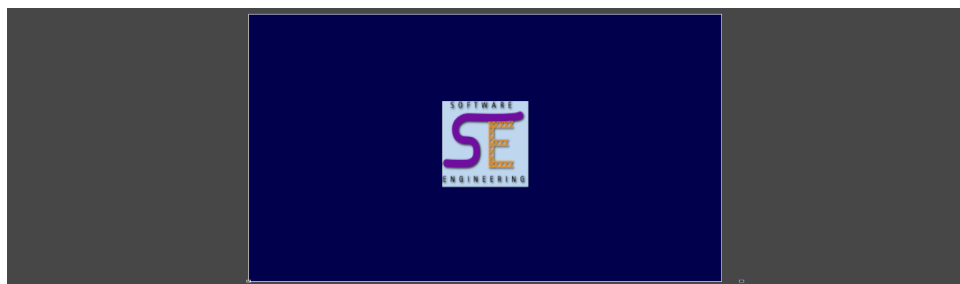


Abbildung A.8: Szenenwechsel – Mitte



Abbildung A.9: Szenenwechsel – Teil 2.1

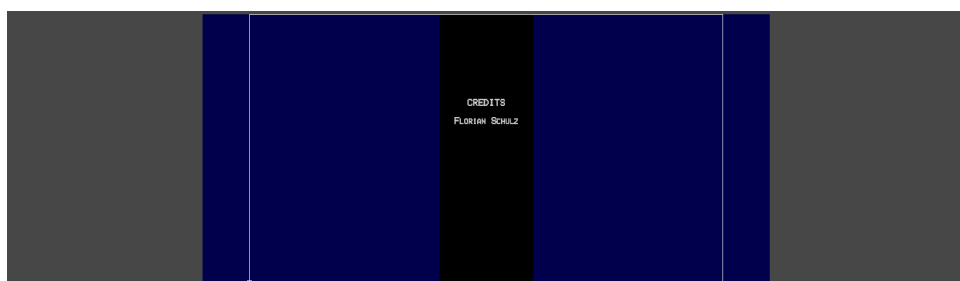


Abbildung A.10: Szenenwechsel – Teil 2.2



Abbildung A.11: Szenenwechsel – Teil 2.3

Tabellenverzeichnis

2.1	Themengebiete des Fachgebiets Software Engineering zugeordnet zu SWEBOK Wissensbereichen [4]	11
4.1	Abwägung der verschiedenen Methoden zur Erarbeitung relevanter Themen und spielerischer Ansätze	21
4.2	Priorisierungspunkteverteilung der Teilnehmenden	26
4.3	Gruppierung der Themen in die SWEBOK-Wissensbereiche	28
6.1	Fragenkatalog zur Bewertung des Spielerlebnisses	56

Abbildungsverzeichnis

2.1	Darstellung der Interessen der Spielertypen nach Bartle [18]	7
3.1	Foto der 270° Projektionsfläche im Infolab	16
4.1	Ablaufplan des Workshops	23
4.2	Whiteboard nach der Themenfindungs-Session	25
4.3	Whiteboard nach der Priorisierungs-Session	25
4.4	Skizze für grobes Konzept des Spiels	31
4.5	Darstellung der Szenen mit ihren Übergängen	31
4.6	Räumliche Gegebenheiten inklusive Sitzplan für das InfoLab	32
4.7	Visualisierungsbeispiel für das Elicitation-Spiel	35
4.8	Visualisierungsbeispiel für das Doordash-Spiel	36
4.9	Visualisierungsbeispiel für das Äquivalenzklasse-Spiel	38
4.10	Konzeptentwurf des Spielaufbaus	41
4.11	Grundspiel nach dem Beenden des ersten Minispiels	42
5.1	Screenshot von Bildschirm A im Hauptmenü	44
5.2	Screenshot von Bildschirm A während des Hauptspiels mit Nebel und ohne Spieler	45
5.3	Vergleich von Xbox Controller Knopf Anordnung und Design der Antworten im Spiel	47
5.4	Screenshot aus dem Spiel mit verschiedenen Ansichten der beiden Teams	47
5.5	Screenshot aus dem Usability-Minispiel	48
5.6	Screenshot aus dem Memory-Minispiel	49
5.7	Überblick des InfoLabs während des Spiels	50
5.8	Sitzordnung der Teams im InfoLab	51
5.9	Sichtfeld von Mitgliedern des Team 2 während des Spiels	51
5.10	Darstellung der scriptable Fragenobjekte für das Quizduell Minispiel	52
5.11	Beispiel des Aufbaus eines Fragenobjekts für das Quizduell Minispiel	52
5.12	Ablauf zum Erstellen von neuen Frageobjekten	53

6.1	Ergebnisse der Kategorie Gesamteindruck des Spiels	57
6.2	Ergebnisse der Kategorie Verständlichkeit des Spiels	58
6.3	Ergebnis der Frage zur Lernbarkeit der Steuerung	59
6.4	Ergebnisse der Fragen zur Kategorie Spielregeln	59
6.5	Ergebnisse zu Fragen der Kategorie Spieldauer	60
6.6	Ergebnis zur Frage bezüglich des gewählten Titels (Q8)	61
A.1	Dialog vor dem ersten Minispiel	69
A.2	Dialog vor dem zweiten Minispiel	70
A.3	Dialog vor dem dritten Minispiel	70
A.4	Einer der Dialoge vor dem vierten Minispiel	71
A.5	Szenenwechsel – Teil 1.1	71
A.6	Szenenwechsel – Teil 1.2	71
A.7	Szenenwechsel – Teil 1.3	72
A.8	Szenenwechsel – Mitte	72
A.9	Szenenwechsel – Teil 2.1	72
A.10	Szenenwechsel – Teil 2.2	72
A.11	Szenenwechsel – Teil 2.3	73

Literaturverzeichnis

- [1] C. Abt. *Serious Games*. University Press of America, 1987.
- [2] M. Ahrens and L. Nagel. All eyes on traceability: An interview study on industry practices and eye tracking potential. In *2023 IEEE 31st International Requirements Engineering Conference (RE)*, pages 77–88, 2023.
- [3] C. F. Barreto and C. França. 2021 iee/acm 13th international workshop on cooperative and human aspects of software engineering (chase). pages 105–108, 2021.
- [4] P. Bourque, R. E. Fairley, and I. C. Society. *Guide to the Software Engineering Body of Knowledge (SWEBOK): Version 3.0*. IEEE Computer Society Press, Washington, DC, USA, 3rd edition, 2014.
- [5] L. Chazette, J. Klünder, M. Balci, and K. Schneider. How can we develop explainable systems? insights from a literature review and an interview study. In *Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering*, ICSSP’22, page 1–12, New York, NY, USA, 2022. Association for Computing Machinery.
- [6] L. Chazette and K. Schneider. Explainability as a non-functional requirement: challenges and recommendations. *Requirements Engineering*, 25(4):493–514, Dec 2020.
- [7] H. Deters, J. Droste, M. Obaidi, and K. Schneider. How explainable is your system? towards a quality model for explainability. In *Requirements Engineering: Foundation for Software Quality: 30th International Working Conference, REFSQ 2024, Winterthur, Switzerland, April 8–11, 2024, Proceedings*, page 3–19, Berlin, Heidelberg, 2024. Springer-Verlag.
- [8] H. Deters, J. Droste, and K. Schneider. A means to what end? evaluating the explainability of software systems using goal-oriented heuristics. In *Proceedings of the 27th International Conference on Evaluation and*

- Assessment in Software Engineering*, EASE '23, page 329–338, New York, NY, USA, 2023. Association for Computing Machinery.
- [9] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian. *Selecting Empirical Methods for Software Engineering Research*, pages 285–311. Springer London, London, 2008.
 - [10] L. U. H. Fachgebiet Software Engineering. Webseite: Forschung. <https://www.pi.uni-hannover.de/de/se/forschung>. Zuletzt zugegriffen: 12.05.2024.
 - [11] L. U. H. Fachgebiet Software Engineering. Webseite: Lehre. <https://www.pi.uni-hannover.de/de/se/lehre>. Zuletzt zugegriffen: 12.05.2024.
 - [12] L. U. H. Fachgebiet Software Engineering. Webseite: Studentische arbeiten – themen. Link: <https://www.pi.uni-hannover.de/de/se/studentische-arbeiten/themen>. Zuletzt zugegriffen: 12.05.2024.
 - [13] J. Jürjens and K. Schneider. Tracesec – tracing and explaining security in software engineering. Deutsche Forschungsgemeinschaft (DFG) - Project number 500462081, 2022. <https://gepris.dfg.de/gepris/projekt/500462081>.
 - [14] M. King and G. Bruner. Social desirability bias: A neglected aspect of validity testing. *Psychology and Marketing*, 17:79 – 103, 02 2000.
 - [15] L. Nagel and O. Karras. Keep your stakeholders engaged: Interactive vision videos in requirements engineering. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pages 51–57, Los Alamitos, CA, USA, sep 2021. IEEE Computer Society.
 - [16] C. N. Phellas, A. Bloch, and C. Seale. Structured methods: interviews, questionnaires and observation. *Researching society and culture*, 3(1):23–32.
 - [17] C. Prasuhn. Entwurf und entwicklung eines 270° abenteuerspiels über die geschichte der informatik, May 2017. Master thesis, Leibniz University Hanover.
 - [18] J. Schell. *The Art of game Design – A Book of Lenses*. Morgan Kaufmann Publishers, imprint of Elsevier, 2008.
 - [19] K. Schneider. Using vision videos for requirements engineering. Deutsche Forschungsgemeinschaft (DFG) - Project number 289386339, 2016. <https://gepris.dfg.de/gepris/projekt/289386339>.

- [20] K. Schneider. softxplain: Requirements for self-explaining software. Deutsche Forschungsgemeinschaft (DFG) - Project number 470146331, 2021. <https://gepris.dfg.de/gepris/projekt/470146331>.
- [21] J. Stenros. The game definition game: A review. *Games and Culture*, 12(6):499–520, 2017.
- [22] K. Thoring, R. M. Mueller, and P. Badke-Schaub. Workshops as a research method: Guidelines for designing and evaluating artifacts through workshops. In *Proceedings of the 53rd Annual Hawaii International Conference on System Sciences, HICSS 2020*, volume 2020. IEEE, 2020.

