



**Hochschule  
Bonn-Rhein-Sieg**  
University of Applied Sciences

## Abschlussarbeit

Bachelor of Science in Computer Science

# Sprachgesteuerte Kommunikation mit Nicht-Spieler-Charakteren in Computerspielen

vorgelegt von  
Kai Friese  
und  
Jasmin Knott

Eingereicht am: 19.06.2023  
Erstprüfer: Prof. Dr. Wolfgang Heiden  
Zweitprüfer: Prof. Dr. Ernst Kruijff

## **1. Abstract**

*Autor: Kai Friese und Jasmin Knott*

Dialoge in Computerspielen werden traditionell durch das Auswählen von einer oder mehreren Dialogoptionen gesteuert. Diese Steuerung bietet dem Spieler nur begrenzten Freiraum für die Gestaltung der Spielcharakterpersönlichkeit. Zudem können bei Unklarheiten keine Fragen gestellt werden und der Spieler kann nicht frei seine Meinung zu Geschehnissen äußern. Insgesamt wird die Kommunikation dadurch unrealistisch, da in einer realistischen Kommunikation jeder Beteiligte theoretisch frei äußern kann, was er oder sie möchte.

Das Ziel der vorliegenden Arbeit war es, zu prüfen, ob eine sprachgesteuerte Kommunikation die Dialoge in Computerspielen zwischen Spieler und Nicht-Spieler-Charakteren realistischer wirken lässt. Dazu wurde ein Kommunikation-intensives Computerspiel entwickelt, in welchem die Spracheingabe der Spieler ausgewertet wird und von einem Chatbot Antworten gegeben werden.

Um das Projekt auszuwerten, wurden Fragebögen von Testspielern ausgefüllt. Die gegebenen Antworten zeigen, dass die Kommunikation im Computerspiel durch Sprachsteuerung grundsätzlich realistisch wirken kann. Es hängt jedoch von der Umsetzung, der verwendeten Hardware und dem zugrundeliegenden Spiel ab, wie real die Kommunikation tatsächlich wirkt.

## **2. Vorwort**

*Autor: Kai Friese und Jasmin Knott*

Mit der vorliegende Bachelorarbeit behandeln wir die Fragestellung „Ist es möglich, durch Spracheingabe und deren Verarbeitung die Kommunikation in Computerspielen zwischen Spieler und Nicht-Spieler-Charakteren realistischer wirken zu lassen?“. Dazu entwickelten wir ein kleines Computerspielszenario, in welchem man sich mit einigen Nicht-Spieler-Charakteren unterhalten kann.

Wir haben uns für das Thema „Sprachgesteuerte Kommunikation zwischen Spieler und Nicht-Spieler-Charakteren“ entschieden, da wir in Computerspielen oft auf das Problem stoßen, in einem Dialog nur wenig Auswahl zu haben. Nicht alle Spieler können sich mit den angebotenen Dialogoptionen identifizieren und würden gerne anders im Gespräch reagieren. Außerdem werden die Auswahlmöglichkeiten häufig auf wenige kurze Worte reduziert, während der Spieler-Charakter jedoch weitaus mehr sagt, wodurch der Inhalt der Aussage von der getroffenen Auswahl abweicht.

Aufgrund unseres gemeinsamen Interesses an Computerspielen, vor allem an dem Genre Rollenspiele, sowie unserem gemeinsamen Informatikstudium an der Hochschule Bonn-Rhein-Sieg mit dem Fokus auf Game Development, entschieden wir uns, unsere Bachelorarbeit als Gruppe in diesem Bereich zu schreiben.

Das Verfassen von Abstract, Vorwort, Einleitung, Diskussion und Fazit sowie das Recherchieren von Hintergrund und Forschungsstand haben wir gemeinsam durchgeführt. Für die grundlegenden Spielmechaniken, Tools, APIs und Frameworks sowie die Spracheingabeverarbeitung und Verwaltung von Kommunikationsdaten war Kai Friese zuständig. Jasmin Knott kümmerte sich um die Grafiken, Schlüsselwörter und Dialoge sowie den Turing-Test [15] und die Ergebnisbewertung.

## 3. Verzeichnisse

### Inhaltsverzeichnis

1. Abstract.....	1
2. Vorwort.....	2
3. Verzeichnisse.....	3
3.1 Abkürzungsverzeichnis.....	4
4. Einleitung.....	5
5. Hintergrund und Forschungsstand.....	6
5.1 Chatbotsysteme.....	6
5.1.1 Chatbots.....	6
5.1.2 Emotionale Chatbots.....	8
5.1.3 ChatGPT.....	9
5.2 Game Pidgin Language.....	10
5.3 Dialoge in Computerspielen.....	12
5.4 Verwandte Arbeiten.....	16
6. Konzept.....	17
6.1 Grundlage.....	17
6.2 Spracheingabeverarbeitung.....	17
6.3 Schlüsselwörter.....	18
6.4 NPC-Erinnerungsvermögen.....	18
7. Beschreibung des Spiels.....	19
7.1 Allgemeines.....	19
7.2 Spielwelt.....	19
7.2.1 Die Welt.....	19
7.2.2 Spielorte.....	19
7.2.3 Charaktere.....	20
7.3 Die Handlung.....	22
7.3.1 Der Anfang.....	22
7.3.2 Der Hauptteil.....	22
7.3.3 Das Spielende.....	23
8. Umsetzung.....	24
8.1 Tools, API's und Frameworks.....	24
8.1.1 GitHub.....	24
8.1.2 ProCreate.....	24
8.1.3 Unity.....	24
8.1.4 Visual Studio.....	24
8.2 Spracheingabe, Schlüsselwörter und Dialoge.....	25
8.2.1 Spracheingabeverarbeitung.....	25
8.2.2 Schlüsselwörter.....	31
8.2.3 Dialoge.....	32
8.2.4 Gesprächsthemen.....	35
9. Ergebnisse.....	37
10. Diskussion.....	44
10.1 Eingebaute Schlüsselwörter.....	44
10.2 Implementierungsfehler.....	45
10.3 Kommunikationsschwierigkeiten.....	46
10.4 Begrenzte Anwendbarkeit.....	46

11. Fazit und Ausblick.....	48
11.1 Fazit.....	48
11.2 Ausblick.....	49
12. Eidesstattliche Erklärung.....	50
13. Literaturverzeichnis.....	51
14. Anhang.....	54
14.1 Analoger Anhang.....	54
14.2 Digitaler Anhang.....	56

***3.1 Abkürzungsverzeichnis***

CPL – Computer Pidgin Language

CPU – Central Processing Unit

DNN – Deep Neural Network

GPL – Game Pidgin Language

KI – Künstliche Intelligenz

NLP – Natural Language Processing

NPC – Non Player Character (Nicht-Spieler-Charakter)

## **4. Einleitung**

*Autor: Kai Friese und Jasmin Knott*

Die Videospielindustrie ist ein stetig wachsender Zweig. Viele Menschen spielen heutzutage digitale Spiele, um Spaß zu haben, sich zu entspannen oder spannende Geschichten zu erleben. Dabei tragen NPCs eine zentrale Rolle in den meisten Videospielen, sie erzählen die Geschichte, vergeben Aufgaben oder sind für deren Lösung essenziell. Selbst als Statisten im Hintergrund füllen NPCs die Spielwelt mit Leben. Durch ihr Verhalten und die Gespräche lassen NPCs ein Spiel immersiv wirken und der Spieler baut eine Verbindung zu den Charakteren auf. Jedoch sind in den meisten Videospielen alle Dialoge bereits vorherbestimmt. Es gibt häufig eine vorgeschriebene Auswahl an Worten, welche der Spieler an die NPCs richten kann und diese geben feste Antworten. Dadurch wird der Spieler in eine mehr oder weniger feste Rolle gezwungen und kann die Dialoge seiner Spielfigur nicht frei wählen.

Für eine real wirkende Kommunikation muss der Spieler in der Lage sein, freie Eingaben zu tätigen. Dazu bietet sich eine Spracheingabe durch ein Mikrofon an. Um die Eingabe des Spielers zu verstehen und eine passende Antwort zu finden, wird dann ein Chatbot eingesetzt. Chatbots sind Dialogsysteme mit zahlreichen Anwendungsgebieten. Sie werden für den Kundenservice<sup>1</sup>, als Unterhaltung oder in Logistik- und Buchungsprozessen<sup>2</sup> eingesetzt. Jedoch werden Chatbots bislang kaum in der Videospiel-Branche verwendet, obwohl die Kommunikation mit NPCs ein geeignetes Anwendungsfeld ist.

Eins der ersten Spiele, welche sich auf die Nutzung von KI und Chatbots konzentrierte, war „Eliza“ [17]. Es wurde von Joseph Weizenbaum entwickelt und 1966 erstmals veröffentlicht. Den Namen erhielt es vom gleichnamigen Psychotherapie-Chatbot.

Ein moderneres Spiel welches Chatbots verwendet, ist „Simulacra“<sup>3</sup> von Kaigan Games aus dem Jahr 2017. In diesem Spiel kann man mit der Hilfe eines Chatbots Rätsel lösen und Informationen sammeln.

Mit dem Fortschritt in KI und Chatbots werden jedoch in Zukunft weitere Videospiele mit diesem Ansatz entwickelt werden.

---

1 <https://www.ing.de/hilfe/>

2 <https://www.lufthansa.com/digitalassistant/webchat.html>

3 <https://www kaigangames com/game/games>

## **5. Hintergrund und Forschungsstand**

### **5.1 Chatbotsysteme**

*Autor: Kai Friese*

#### **5.1.1 Chatbots**

In Computerspielen spielen NPCs eine wichtige Rolle [6]. Sie lassen das Spiel lebendig wirken, vergeben Aufgaben an den Spieler, betreiben Handel oder geben wichtige Informationen an den Spieler. In der Regel agieren die NPCs dabei nach einem fest programmierten Verhalten, bei welchem durch Informationen wie Spielstand, Position und Aufgabenfortschritt ermittelt wird, welche Aktion der NPC ausführt.

Dies gilt auch für Dialoge zwischen dem Spieler und NPCs und auch für Dialoge zwischen zwei oder mehreren NPCs untereinander. Durch den Einsatz von einem Chatbot-System kann die Spielerfahrung noch weiter erhöht werden.

Es existieren verschiedene Techniken für Chatbots, welche verwendet werden können, um Dialoge in Videospielen zu verbessern. In einem einfachen Chatbot-System analysiert ein Chatbot in der Regel eine von einer Person gestellte Anfrage und holt sich eine bestimmte Antwort aus einer festen Datenbank. In der Regel wird die Antwort auf der Grundlage der Keywords abgeleitet und eine feste Antwort wird als Ausgabe gegeben.

Um nicht ausschließlich mit Keywords zu arbeiten, sondern auch den Inhalt einer Aussage verarbeiten zu können, wird die Technik "Natural Language Processing" [2][8] [9] (NLP) verwendet. Bei dieser Technik werden verschiedene Vorgänge verwendet. Für uns relevant sind zunächst die Tokenisierung [16] und die Kennzeichnung von Wortarten.

Die Tokenisierung ist ein Vorgang, bei dem ein Satz in einzelne Wörter oder Tokens zerlegt wird. Jedes Wort ist ein separates Token, das zur weiteren Verarbeitung übergeben wird.

Die Kennzeichnung von Wortarten wird verwendet, um die Wortart zu identifizieren, die jedes einzelne Wort in einem bestimmten Satz darstellt. Eine Wortart kann ein Substantiv, ein Adjektiv, ein Verb oder ein Adverb sein. Die richtige Wortart eines

Wortes in einem Satz muss identifiziert werden. Einige Wörter können mehr als eine Wortart haben, was ganz von der Satzstruktur abhängt.

NLP kann jedoch nicht nur zum Verständnis und zur Verarbeitung von natürlicher Sprache genutzt werden. Es ist auch möglich, unter Verwendung dieser Technik, automatische Antworten von einer KI generieren zu lassen. Zu Beginn war der Nutzen für die Videospiel-Branche jedoch gering, da Speicherplatz und CPU-Ressourcen nur limitiert zur Verfügung standen und NLP noch nicht ausgereift genug war [11].

Dank der Transformer-Architektur [5] und der Verfügbarkeit von großen Mengen an Textdaten im Internet sowie bereits trainierten Sprachmodellen, ist es heutzutage möglich, menschenähnliche Sätze und Texte als Antwort zu generieren. Die Verwendung dieser Technik in Videospielen ermöglicht es den Spielern, viel freiere Konversationen zu führen.

Jedoch treten dadurch neue Probleme auf: Die durch eine KI generierten Antworten lassen sich nicht genau kontrollieren.

Die von der KI generierten Antworten hängen von den Daten ab, mit welchen sie trainiert wurde [11]. Um den Spielcharakteren unterschiedliche Persönlichkeiten zu geben, können diese mit unterschiedlichen Trainingsdaten angelernt werden. Der KI sind so jedoch keine Werte und Grenzen bekannt, welche im Training nicht enthalten waren. So könnte ein böser Antagonist, entsprechend seiner Moral und ethischen Vorstellungen in der fiktiven Welt Antworten geben, welche die ethischen Grenzen der realen Welt überschreiten. Andersherum kann es auch sein, dass die KI mit „zu freundlichen“ Datensätzen trainiert wurde und so den Charakter eines bösen Antagonisten nicht glaubhaft wiedergeben kann, wodurch die Stimmung im Spiel zerstört wird. Es ist auch vorstellbar, dass der Spieler Markennamen verwendet, welche von der KI wieder aufgegriffen werden und dadurch rechtliche Probleme oder Rufschädigungen entstehen können [11]. Ebenfalls sensible Themen wie Diskriminierung, Gewalt und persönliche Informationen können zu Problemen führen.

Des Weiteren muss man darauf achten, dass die KI nicht über Dinge spricht, welche nicht Teil der Spielwelt sind. Die KI könnte beispielsweise historische Ereignisse ansprechen, welche zu dem Zeitpunkt, zu welchem das Spiel spielt, noch nicht geschehen sind.

Moderne Chatbots können in vier Kategorien unterteilt werden [10]: (1) knowledge (open/close domain), (2) response generation (retrieval or generative), (3) text processing (vector embedding oder latin alphabet) und (4) machine learning model.

Knowledge Chatbots verfügen über Wissen in bestimmten Bereichen (close domain), oder über eine Art Allgemeinwissen aus verschiedenen Datensätzen und Quellen (open domain).

Bei response generation Chatbots existieren im Grunde zwei Methoden, von welchen jedoch eine Vielzahl an Variationen entstanden sind. Bei der retrieval-Methode wird die beste Ausgabe aus kurzen Listen von Antworten ausgewählt, während bei der generative-Methode eine flexible Ausgabe anhand von Eingabesequenz und trainierten Klassifikatoren erstellt wird.

Text processing Chatbot-Systeme ermitteln eine semantische Beziehung zwischen Wörtern innerhalb eines bestimmten Vokabulars, dazu werden entweder reelle Zahlen (vector embedding) oder Buchstaben/Wörter (latin alphabet) verwendet.

Machine learning wird bei Chatbots vor allem mit Neuralen Netzwerken verwendet und kann als zentrale Komponente fast alle Aufgaben des Systems übernehmen. Dazu gehören die Eingabevorbereitung und -verarbeitung sowie die Ausgabeverarbeitung und -produktion.

### **5.1.2 Emotionale Chatbots**

Einige Forscher sind jedoch auf die Idee gekommen, nicht nur das Verständnis des Kontexts für die Generierung von Antworten zu verwenden [6]. Denn für die menschliche Kommunikation spielen auch Emotionen und Gefühle eine wichtige Rolle. Im Zuge dessen entstanden verschiedene Chatbots wie XiaoIce<sup>4</sup>, CAIRE<sup>5</sup> und Emotional Chatting Machine<sup>6</sup> (ECM). Diese sind in der Lage, einfühlsam und gefühlsbetont zu reagieren und die Interaktion durch das Verstehen der Stimmung zu verbessern.

Da der Mensch eine Konversation multimodal wahrnimmt, ist dies auch ein wichtiger Ansatz von Chatbots. Der Mensch nimmt bei einer Unterhaltung neben dem gesprochenen Wort auch Mimik, Gestik und Tonlage wahr. Diese Modalitäten werden

4 <https://www.xiaoice.com/>

5 <https://demo.caire.ust.hk/chatbot/>

6 Der ECM steht nicht zur öffentlichen Verfügung und es gibt derzeit keine spezifischen wissenschaftlichen Veröffentlichungen zum ECM. Der ECM wurde von OpenAI entwickelt.  
<https://openai.com/>

verwendet, um die Bedeutung einer Aussage zu verstehen. In Chatbot-Systemen wird dies durch die Umwandlung von Text in Audio oder die Darstellung von animierten Gesichtern umgesetzt.

### **5.1.3 ChatGPT**

Einer der aktuell besten und bekanntesten Chatbots ist ChatGPT<sup>7</sup>. Es wurde am 30. November 2022 von OpenAI veröffentlicht und basiert auf einer Künstlichen Intelligenz. ChatGPT wurde für Dialoge optimiert und kann gesamte Gespräche verfolgen. Mithilfe von NLP kann die KI Fragen, welche in natürlicher Sprache formuliert werden, verarbeiten und in einem menschenähnlichen Text beantworten.

Als Struktur verwendet ChatGPT eine Transformer-Architektur [5], um natürliche Konversationen zu erzeugen. Die Architektur besteht prinzipiell aus einem Encoder/Decoder- (bei einigen Transformersystemen allerdings aus einem reinen Decoder-) und einem Selbstbeobachtungsmechanismus. Diese Struktur wurde entwickelt, um den Kontext einer Konversation besser zu erfassen und kohärente Antworten zu generieren [1].

ChatGPT ist in der Lage, verschiedene Aufgaben auszuführen. Einige beispielhafte Fähigkeiten sind:<sup>8</sup>

- Antworten auf Fragen zu vielen Themenbereichen wie Wissenschaft, Geschichte, Kunst, Musik, Sprachen und vielem mehr.
- Erstellen von Inhalten wie Gedichten, Geschichten, Songtexten oder sogar Programmcodes.
- Übersetzung von Texten von einer Sprache in eine andere.
- Unterstützung bei der Suche nach Informationen im Internet.
- Durchführung von Berechnungen und Konvertierungen von Einheiten.
- Analyse von Texten und Erkennung von Mustern.
- Interaktion mit Nutzern in natürlicher Sprache.

---

7 <https://openai.com/blog/chatgpt>

8 Von ChatGPT erstellte Auflistung der Fähigkeiten <https://chat.openai.com/>

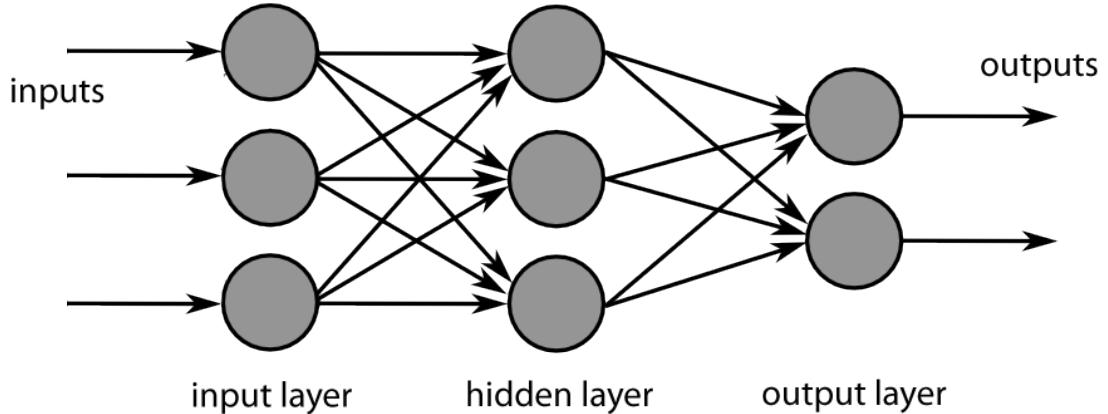


Abbildung 1: Deep Neural Network [18]

Um dies zu bewerkstelligen, basiert ChatGPT auf einem „Deep Neural Network“ [9] [10] (DNN). Dies ist ein künstliches Konstrukt, welches das Nervensystem von Lebewesen nachahmt. Ein solches Netz besteht, wie in Abbildung 1 zu sehen, aus verschiedenen Schichten von künstlichen Neuronen. Informationen werden durch die erste Schicht, welche als Eingabeschicht (input layer) fungiert, an das Netzwerk übermittelt. Alle Informationsdaten werden durch eine oder mehrere Zwischenschichten, oder auch verborgene Schichten (hidden layer) genannt, geleitet und verarbeitet. Die Ausgabe einer Schicht wird dabei als Eingabe der nächsten Schicht verwendet. Als letzte Schicht wird die Ausgabeschicht (output layer) verwendet. Sie gibt die Ergebnisse der vollständig verarbeiteten Daten aus. Ein DNN kann durch einen sogenannten Backpropagation-Algorithmus trainiert werden. In diesem Prozess werden die Neuronen schrittweise angepasst, um die Ergebnisse zu verbessern. Gesteuert wird dies durch eine Verlustfunktion, welche den Unterschied zwischen den Ergebnissen des Netzwerks und den tatsächlichen Werten misst.

## 5.2 Game Pidgin Language

*Autor: Kai Friese und Jasmin Knott*

Kommunikation ist nicht nur in natürlicher Sprache möglich, sondern unter anderem auch mit einer „Pidgin-Sprache“. Dabei handelt es sich um eine reduzierte Sprachform, welche verschiedensprachige Personen zur Kommunikation verwenden. Diese Pidgin-Sprache stammt aus der Kolonialzeit. Zu dieser Zeit trafen vermehrt Menschen mit unterschiedlichen Muttersprachen aufeinander, sodass die Notwendigkeit einer gemeinsamen Kommunikation entstand. Die schwächere Gruppierung passte ihre

Sprache dabei der Sprache der Kolonialmacht an. Dabei werden Grammatik und Vokabular stark vereinfacht und dienen nur zur rudimentären Kommunikation. Aus dieser Pidgin-Sprache wurde ein Konzept zur Spracherkennung und Sprachsteuerung entwickelt, die „computer pidgin language“ [7] (CPL). Bei diesem Konzept soll der menschliche Anwender eine Sprache lernen, welche ein Computer leichter verstehen kann als natürliche Sprache.

Aus der CPL wurde eine weitere Sprache für effiziente Dialoge in Computerspielen entwickelt, die „game pidgin language“ [13][14] (GPL).

„The GPL has a very limited set of grammatical rules. It primarily had eight rules [1]<sup>9</sup>, which are as follows -

1. The Sentence may begin with a noun or noun phrase followed by Verb.
2. A Sentence may begin with a Verb Phrase followed by a Noun.
3. An Adjective is a valid sentence.
4. A Noun is a valid sentence.
5. A Verb Phrase is a Valid Sentence.
6. The auxiliary „i“ (é) may be added as suffix to a noun that does not end with „i, „a“ or „y“ sounding alphabet.
7. There is no tense in this language as Computer games are played in real time.
8. The language has no gender.

In this paper, we have added three more rules. The first is there are no articles in the grammar. Second is a preposition phrase is a valid sentence. The third is, a sentence cannot be more than two words long except for proper nouns.“ ([13], 2005, Seite 2)

Ein Satz darf dabei nur aus den Wörtern des GPL-Wörterbuchs gebildet werden. Durch die Verwendung dieser Regeln kann eine schnelle und effiziente Kommunikation zwischen dem menschlichen Spieler und den NPCs entstehen. Da alle möglichen Wörter und Sätze dem System bereits bekannt sind, kann so auf jede gültige Eingabe eine passende Reaktion vom System ausgelöst werden.

---

<sup>9</sup> Die Quellenreferenz [1] des Originalzitats bezieht sich auf Quelle [14] dieser Thesis.

### 5.3 Dialoge in Computerspielen

Autor: Jasmin Knott

Es gibt verschiedene Varianten, wie Dialoge in Computerspielen aufgebaut sind, um sich mit den NPCs zu verständigen. Die meisten Videospiele verwenden dabei eine textbasierte Antwortauswahlmethode. Dabei erscheinen vorwiegend zwei bis vier, seltener auch mehr Antwortmöglichkeiten. Diese Möglichkeiten können aus einzelnen Wörtern oder auch ganzen Sätzen bestehen. Der Spieler kann sich für eine der Antwortmöglichkeiten entscheiden, woraufhin der NPC mit einer passend einprogrammierten Antwort reagiert. Auch bekannte Spiele wie „Hogwarts Legacy“<sup>10</sup> und „The Witcher 3: Wild Hunt“<sup>11</sup> benutzen diese Methode. In Abbildung 2 sieht man einen Ausschnitt von „Hogwarts Legacy“, wo man entscheiden kann, einer Aussage des NPCs zuzustimmen oder zu widersprechen. In Abbildung 3 sind Dialogoptionen aus „The Witcher 3: Wild Hunt“ dargestellt. Dort spricht der Spieler gerade mit einem Händler und hat die Möglichkeit etwas zu kaufen, nach Informationen zu fragen oder den NPC zu einer Runde Gwint, einem Sammelkartenminispiel, herauszufordern. Zusätzlich kann der Spieler sich verabschieden und das Gespräch beenden. In der Regel wird dem Spieler angezeigt, welche Antwort aktuell ausgewählt ist, bevor diese bestätigt wird. Bei „Hogwarts Legacy“ ist die Anzeige ein Rahmen um die Antwort, bei „The Witcher 3: Wild Hunt“ zeigt ein Pfeil darauf.

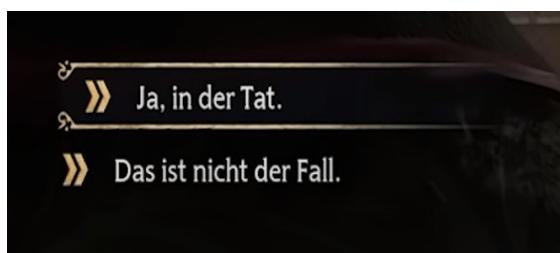


Abbildung 2: Dialog aus Hogwarts Legacy mit Auswahlanzeige

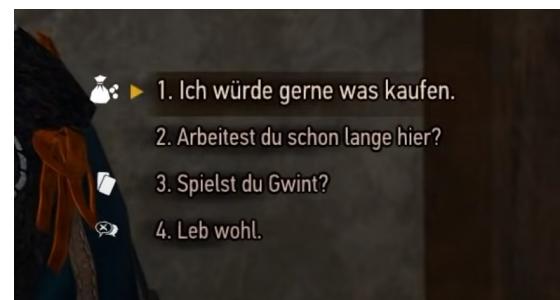
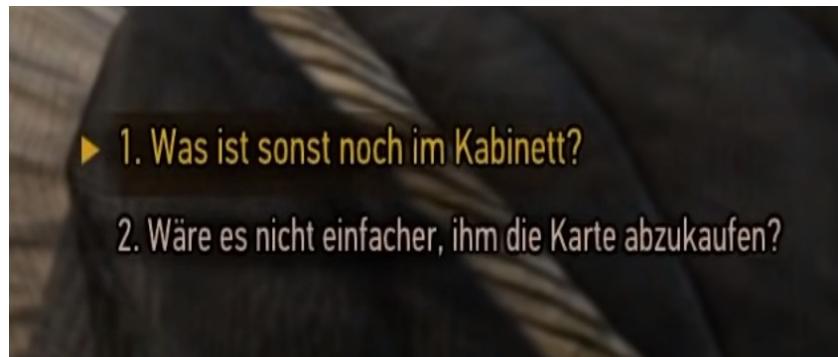


Abbildung 3: Dialog aus The Witcher 3: Wild Hunt mit Auswahlanzeige

Bei dieser Methode der Dialogform gibt es auch die Variante, für die Aufgabe relevante Antwortoption farblich zu kennzeichnen. Im Beispiel von „The Witcher 3: Wild Hunt“ werden diese Fragen gelb dargestellt, wie es in Abbildung 4 zu sehen ist. Reguläre Dialogoptionen werden dabei weiß angezeigt und können alles enthalten, was nicht zum Lösen der Aufgabe relevant ist.

10 <https://www.hogwartslegacy.com/de-de>

11 <https://www.thewitcher.com/us/de/witcher3>



*Abbildung 4: „The Witcher 3: Wild Hunt“ relevante Antworten*

In einigen Spielen gibt es Dialoge, in welchen der Spieler nur begrenzt Zeit hat, sich für eine Auswahl zu entscheiden. Dies wird häufig in speziellen Situationen verwendet, um dem Spieler zu vermitteln, dass der Charakter unter Druck steht. Diese Methode eignet sich sehr gut, um die Immersion und Aufmerksamkeit des Spielers zu fördern.

In „The Witcher 3: Wild Hunt“ wird dies ebenfalls angewendet. Dazu wird dem Spieler visuell mitgeteilt, dass er sich schnell entscheiden muss. In Abbildung 5 ist ein ablaufender Balken zu sehen, welcher ein Beispiel für die visuelle Darstellung aus „The Witcher 3: Wild Hunt“ ist.

Dieser Balken läuft in etwa zehn Sekunden herunter. Meist werden dem Spieler nur wenige, kurze Antwortmöglichkeiten zur Verfügung gestellt, damit dieser auch in der Lage ist sich in dem kurzen Zeitfenster für eine Antwort zu entscheiden. Sollte keine Antwort rechtzeitig gewählt werden, trifft das Spiel eine Entscheidung. Häufig tragen solche Entscheidungen relevante Konsequenzen für den weiteren Spielverlauf mit sich. Im Beispiel von „The Witcher 3: Wild Hunt“ löst die Antwort „Hau ab oder stirb.“ einen Kampf aus, wohingegen „Etwas Schnaps?“ die Situation deeskalieren kann.



Abbildung 5: „The Witcher 3: Wild Hunt“ zeitlich begrenzter Dialog

Das Computerspiel „Beyond two Souls“<sup>12</sup> benutzt eine ähnliche Methode, um mit NPCs zu kommunizieren. Da das Spiel für die Controllersteuerung ausgelegt ist, es gibt maximal vier Auswahlmöglichkeiten pro Dialog. Jedoch hat der Spieler dabei oft beispielsweise nur Emotionen oder Schlagwörter zur Auswahl. In der Spielsituation aus Abbildung 6 kann der Spieler sich beispielsweise zwischen den Möglichkeiten „Beschuldigen“, „Argumentieren“, „Rühren“ oder „Vorwurf“ entscheiden. Grundsätzlich hat der Spieler dadurch ein Gefühl, in welche Richtung die Antwort gelenkt wird. Doch kennt man die exakte Aussage und Bedeutung erst, sobald die Spielfigur die Antwort ausgesprochen hat. Das kann jedoch zu Missinterpretation der Antwortmöglichkeiten führen, beispielsweise könnte der Spieler für etwas argumentieren wollen, doch der Spielcharakter argumentiert dagegen. Da der Spieler seine Antwort nachträglich nicht mehr ändern kann, kann dies zu Frustration führen. Gerade wenn Antworten ähnliche Bedeutungen haben wie „Beschuldigen“ und „Vorwurf“ ist nicht klar absehbar, welche Antwort zu welchen Konsequenzen führt.

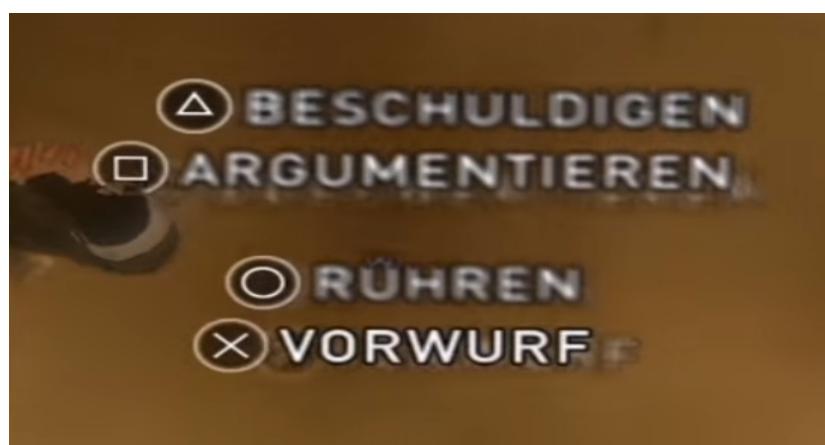


Abbildung 6: „Beyond two Souls“ Dialog

12 <https://www.quanticdream.com/en/beyond-two-souls>

Wie in Abbildung 6 zu erkennen können Dialoge auch auf andere Art, als durch die Worte, Einfluss auf das Spiel nehmen. So befindet sich der Spieler in dieser Situation in einer Art Sturm. Diese offensichtlich unangenehme Situation wird dadurch bestärkt, dass die Auswahloptionen wackeln, verschwimmen und ihre Größe verändern. Auch in dieser Situation ist die Auswahlzeit begrenzt. Dies wird dadurch erkennbar gemacht, dass mit der Zeit drei der Optionen immer schwerer zu lesen sind und langsam verschwinden. Die letzte Auswahlmöglichkeit wird dabei in den Vordergrund gerückt und klarer lesbar: wenn die Zeit abläuft, wird diese Option automatisch gewählt.

Eine weitere Variante an Dialogen zeigt das Detektivspiel „L.A. Noire“<sup>13</sup>. Dort bewegt der Spieler sich durch die Geschichte, ohne große Gesprächsauswahl. Es ist wie ein interaktiver Film, mit vielen Sequenzen, in welchen der Spielercharakter mit dem Spieler über Selbstgespräche kommuniziert. Bei der Untersuchung eines Tatorts zum Beispiel, gibt der Spielercharakter über ein Selbstgespräch alle Informationen über die Leiche an den Spieler weiter, sobald man mit dieser interagiert. Im Vergleich dazu hätte der Spieler in „The Witcher 3: Wild Hunt“ noch die Möglichkeit sich in Dialogform, die einzelnen Spuren am Opfer anzusehen und erklären zu lassen.

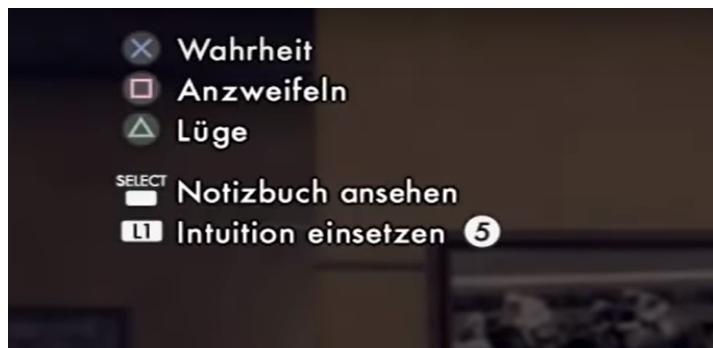


Abbildung 7: „L.A. Noire“ Dialog

In Gesprächen mit NPCs funktioniert „L.A. Noire“ ähnlich wie „Beyond two Souls“. Es stehen wenige Antworten mit einem Wort zur Auswahl. In Abbildungen 7 wurde der NPC befragt, ob er etwas gehört hat, was dieser kurz mit „Nein“ beantwortet. Der Spieler hat nur die Wahl aus „Wahrheit“, um dem NPC zu glauben, „Anzweifeln“, um sein Misstrauen deutlich zu machen, oder „Lüge“, um klar zumachen, dass der Spieler dem NPC nicht glaubt.

13 <https://www.rockstargames.com/lanoire/>

## **5.4 Verwandte Arbeiten**

*Autor: Kai Friese*

Auf dem YouTube-Kanal „Art from the Machine“<sup>14</sup> wurden zwei Videos<sup>1516</sup> veröffentlicht, welche Modifikationen für das Spiel „Skyrim VR“<sup>1718</sup> vorstellen. Diese Modifikationen sind Python Skripts, welche es dem Spieler ermöglichen sich mit den NPCs im Spiel zu unterhalten.

Die Umsetzung wurde mithilfe von den KIs ChatGPT, xVASynth<sup>1920</sup> und Whisper<sup>21</sup> realisiert. Zur Spracherkennung und Umwandlung in Text, dient die automatische Spracherkennungssoftware Whisper von OpenAI, welche mit 680.000 Stunden mehrsprachigen Daten aus dem Internet trainiert wurde. Mithilfe von ChatGPT werden die Eingaben ausgewertet und eine passende Antwort wird generiert. Dabei können auch Daten wie Zeit (im Spiel) und Ort (im Spiel) verwendet werden. Zum Schluss kann durch xVASynth die Textantwort in Sprache umgewandelt werden. Diese KI verfügt über Audio-Daten von zahlreichen NPCs aus verschiedenen Videospielen, unter anderem auch Skyrim. Mit diesen Daten können Texte in verschiedenen Stimmen vertont werden.

---

14 [https://www.youtube.com/@art\\_from\\_the\\_machine](https://www.youtube.com/@art_from_the_machine)

15 <https://www.youtube.com/watch?v=Gz6mAX41fs0>

16 [https://www.youtube.com/watch?v=u\\_Zn89\\_g7ok](https://www.youtube.com/watch?v=u_Zn89_g7ok)

17 <https://elderscrolls.bethesda.net/de/skyrim10>

18 [https://store.steampowered.com/app/611670/The\\_Elder\\_Scrolls\\_V\\_Skyrim\\_VR/](https://store.steampowered.com/app/611670/The_Elder_Scrolls_V_Skyrim_VR/)

19 <https://steamcommunity.com/app/1765720>

20 <https://github.com/DanRuta/xVA-Synth>

21 <https://openai.com/research/whisper>

## **6. Konzept**

### **6.1 Grundlage**

*Autor: Kai Friese*

Zur Beantwortung unserer Forschungsfrage: „*Ist es möglich, durch Spracheingabe und deren Verarbeitung die Kommunikation in Computerspielen zwischen Spieler und Nicht-Spieler-Charakteren realistischer wirken zu lassen?*“, wurde ein simples 2D-Spiel in Unity<sup>22</sup> entwickelt, in welchem die Kommunikation im Vordergrund steht. Der Spieler soll sich dabei in einer mittelalterlichen Taverne aufhalten und mit NPCs wie dem Wirt, einer Schankmaid und reisenden Gästen Gespräche führen können.

Das Spiel findet in einem Detektivsetting statt, in welchem einer der Tavernengäste in der ersten Nacht ums Leben kommt. Die Aufgabe des Spielers besteht darin, die Umstände des Todes aufzuklären.

Durch dieses Szenario wird der Spieler dazu angeregt, sich aktiv mit den NPCs zu unterhalten und die sprachgesteuerte Kommunikation zu verwenden. Des Weiteren ist das Szenario eines Detektivspiels gut für dieses Thema geeignet, da Dialoge in dieser Art von Spielen bislang, meistens nur eingeschränkt verwendet wurden. In den Spielen „L.A. Noire“<sup>23</sup> und „Police Simulator: Patrol Officers“<sup>24</sup> beispielsweise verkörpert der Spieler einen Polizisten, welcher verschiedene Fälle aufklären muss. In beiden Spielen stehen dem Spieler kaum, bis keine Dialogoptionen zur Verfügung.

Dem Spieler werden in der traditionellen Dialogform nur beschränkt Fragen und Antworten angeboten. Dadurch bietet sich den Spielern die Möglichkeit, durch schlichtes Ausprobieren aller Antworten die beste Antwort zu finden. Mit der sprachgesteuerten Kommunikation muss sich der Spieler intensiver mit der Geschichte beschäftigen, um die richtigen Fragen stellen zu können. Dadurch wird die Immersivität eines Detektivspiels gesteigert und der Spieler kann sich wie ein richtiger Detektiv fühlen.

### **6.2 Spracheingabeverarbeitung**

*Autor: Kai Friese*

22 <https://unity.com/de>

23 <https://www.rockstargames.com/lanoire/>

24 <https://www.patrol-officers.com/de-DE/>

Die Sätze des Spielers werden für die Spracheingabeverarbeitung in ihre Einzelteile aufgeteilt und analysiert. Um zu verstehen, was der Spieler sagen möchte, wurde ein Wörterbuch aus spiel relevanten Schlüsselwörtern angelegt. Wenn ein entsprechendes Schlüsselwort genannt wurde, soll der NPC im passenden Kontext antworten. Die Dialoge sollen dabei je nach Gesprächspartner und Fragen, Antworten und Aussagen des Spielers variieren, um eine realistisch wirkende Kommunikation zu simulieren.

Als mögliches Problem sehen wir die Vielfalt der menschlichen Sprache und möglicher Gesprächsthemen, sodass es kaum möglich sein wird, die Bedeutung aller Worte zu erfassen und passende Antworten auf alle Spieler-Eingaben zu haben. Um dieses Problem einzudämmen, werden die Gespräche inhaltlich auf unser Szenario reduziert. Die größte Herausforderung dabei war es, realistisch auf Themen zu reagieren, auf welche die NPCs nicht vorbereitet sind.

### **6.3 Schlüsselwörter**

*Autor: Jasmin Knott*

Die Wahl der Schlüsselwörter muss sorgfältig getroffen werden, da in der natürlichen Sprache Wörter in vielen verschiedenen Formen verwendet werden. Je nach Satzstellung können Wörter dabei auch ihre Bedeutung ändern. Es kann jedoch auch derselbe Kontext durch verschiedene Wörter ausgedrückt werden. Dazu kann der Spieler verschiedene Zeitformen verwenden und Verben konjugieren. Um das alles zu verbinden, werden zahlreiche Schlüsselwörter benötigt, deren Kontext aus dem Dialog und der Kombination mit anderen Wörtern gezogen werden muss.

### **6.4 NPC-Erinnerungsvermögen**

*Autor: Kai Friese*

Um eine reale Konversation zu imitieren, ist es wichtig, dass NPCs sich merken, was bereits gesagt wurde. Um dies zu erreichen, müssen einige Daten gespeichert werden. Zunächst werden alle ausgelösten Schlüsselwörter abgespeichert. Ergänzend können noch weitere Daten gespeichert werden, zum Beispiel, ob ein Charakter oder der Spieler sich bereits vorgestellt hat.

Mithilfe der gespeicherten Daten sollen Rückfragen, Themenwechsel, verspätete Antworten und Weiteres verarbeitet werden können.

## 7. Beschreibung des Spiels

### 7.1 Allgemeines

*Autor: Kai Friese*

Das Spiel fällt unter das Rollenspiel-Genre und ist im Stil eines Fantasy Detektivabenteuers aufgebaut. Um das Spiel spielen zu können, benötigt man Zugang zum Internet und ein Mikrofon mit akzeptabler Aufnahmegerät.

Den Spielercharakter steuert man über die Tasten „W“, „A“, „S“, und „D“. Um mit Objekten zu interagieren, kann die Taste „Enter“ verwendet werden. Damit der Spieler sich mit NPCs unterhalten kann, muss die Sprachaufnahme mit der „Leertaste“ gestartet werden. Dieselbe Taste stoppt die Aufnahme wieder. Oben links im Spiel ist ein Notizbuch angezeigt, welches durch einen Mausklick darauf geöffnet werden kann. Dort stehen Hinweise zur Spieldurchführung.

Falls der Charakter einmal durch andere Spielobjekte blockiert sein sollte und sich dadurch nicht mehr bewegen kann, hilft das gleichzeitige Drücken der rechten Taste „Shift“ und „Backspace“. Dadurch wird der Charakter an einen freien Platz im Zentrum der Szene gesetzt und die Bewegung ist wieder möglich.

### 7.2 Spielwelt

#### 7.2.1 Die Welt

*Autor: Kai Friese*

Das Spiel spielt in einer Fantasiewelt in der es verschiedene Rassen, wie Schleime (Illustrationen im Kapitel: 7.2.3 Charaktere), Menschen, Geister, Tiermenschen oder auch Einhörner gibt. Der Ort des Geschehens spielt in der Nähe eines kleinen Dorfes namens Rübenhain. Dort leben ausschließlich Schleime. In den restlichen Königreichen werden Schleime gejagt und vertrieben, sodass sie gezwungen waren, sich zurückzuziehen. Sie sammelten sich in einem kleinen Königreich, welches die Schleime akzeptierte und wurden dort schnell zur vorherrschenden Rasse.

#### 7.2.2 Spielorte

*Autor: Kai Friese*

In der Nähe des kleinen Dorfes befindet sich eine Taverne, in welcher das Spiel stattfindet. Die Taverne besteht aus dem Schankraum im Erdgeschoss und einem Obergeschoss mit Gästzimmern. Im Außenbereich der Taverne befindet sich auf der rechten Seite das Tor zum Stall, welchen man betreten kann, um mit einem Stallburschen zu sprechen.

### **7.2.3 Charaktere**

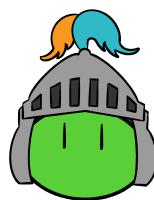
*Autor: Jasmin Knott*

Der Spielercharakter (Abbildung 8) wird durch einen bunten Schleim verkörpert.



*Abbildung 8:  
Spielercharakter*

Sobald man die Taverne betritt, begegnet man links, an einem Tisch sitzend, einem grünen Schleim mit Ritterhelm namens Farron (Abbildung 9). Farron ist Söldner und kommt manchmal zwischen seinen Aufträgen auf ein Bier in die Taverne. Ihm gegenüber sitzt ein roter Schleim namens Theodor (Abbildung 10). Dieser ist ein fahrender Weinhändler und hat Farron beauftragt, ihn auf seiner Reise durch das angrenzende Königreich zu beschützen.



*Abbildung 9:  
Farron*



*Abbildung 10:  
Theodor*

Weiter geradeaus an der Theke befindet sich der Tavernenwirt Brok Koli (Abbildung 11) und seine Tochter Jolanda (Abbildung 12), die Schankmaid.



*Abbildung 11:  
Brok Koli*



*Abbildung 12:  
Jolanda*

Auf der rechten Seite ist eine Gruppe aus vier Schleims, welche an einem Tisch sitzen. Der Schleim mit dem braunen Mantel heißt Albert (Abbildung 13) und ist ein Dorfbewohner, welcher sich seinen Unterhalt als Tagelöhner verdient. Rechts neben ihm sitzt ein blauer Schleim namens Alfred (Abbildung 14). Er ist der Jäger des Dorfes. Gegenüber von ihm mit dem Kleeblatt Tattoo sitzt sein bester Freund namens Francis (Abbildung 15). Francis arbeitet als Metzger im Dorf und ist mit Jolanda liiert. Auf seiner linken Seite befindet sich ein violetter Schleim namens Herbert (Abbildung 16). Er ist der Holzfäller des Dorfes.



*Abbildung 13:  
Albert*



*Abbildung 14:  
Alfred*

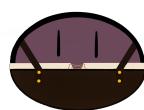


*Abbildung 15:  
Francis*



*Abbildung 16:  
Herbert*

Im Stall passt ein blass violetter Schleim als Stallbursche (Abbildung 17) auf das Pferd Plötze von Farron und den Esel Don von Theodor auf.



*Abbildung 17:  
Stallbursche*

### **7.3 Die Handlung**

#### **7.3.1 Der Anfang**

*Autor: Jasmin Knott*

Der Spieler startet als Abenteurer auf einem Weg vor der Taverne. Darin kann er sich direkt mit den NPCs unterhalten. Er kann sich vorstellen oder nach etwas zu essen, trinken und allgemeinen Informationen über das Dorf fragen. Die Charaktere geben auch Auskunft über ihre Berufe, ihre Stimmung, das Wetter und mehr. Der Wirt schickt den Spieler mit dem Schlüssel über die Wendeltreppe hoch zu seinem Zimmer. Dort schläft der Spielercharakter auch direkt ein.

#### **7.3.2 Der Hauptteil**

*Autor: Jasmin Knott*

Der Spieler wird von einem Schrei geweckt. Im Notizbuch ist nun die Mission aufgetaucht herauszufinden, wer diese Nacht geschrien hat. Im Stall begegnet der Spieler dem Stalljungen. Dieser gibt zu, geschrien zu haben, da er Albert tot aufgefunden hat. Wenn er gefragt wird, was passiert ist, erzählt er einem, dass er ihn zuletzt gestern Abend gesehen hat, da er von ihm verlangte ihm noch eine Flasche Wein zu holen. Das Notizbuch weist einen indessen an, sich die Leiche und den Tatort genauer anzusehen. Neben dem toten Albert findet sich eine Weinflasche, die fast leer ist. Untersucht man danach die Leiche, erscheint plötzlich der Geist von Albert. Leider weiß dieser jedoch nicht, wer ihn umgebracht hat.

Der Spieler muss die anderen NPC's nach einem möglichen Mordmotiv und einem Alibi befragen. Von Theodor und Farron kann man erfahren, dass die Weinflasche und Albert vergiftet worden war.

Werden alle anwesenden über die Flasche befragt, antwortet Francis, dass Albert die Flasche von einem Fremden aus dem Wald geschenkt bekommen hat. Von Theodor erfährt der Spieler, dass Albert ihm den Wein angeboten hat, damit Theodor ihm seine Schulden erlässt. Theodor hat sie dann weiter an den Wirt verkauft. Vom Wirt erfährt der Spieler, dass er die Flasche unter den Thresen gestellt habe, sie nun aber nicht mehr da sei.

Es muss also diese Flasche gewesen sein, die der Stalljunge Albert am Vorabend

gebracht hatte. Der Wirt und Theodor behaupten beide die Flasche nicht vergiftet zu haben. Die Flasche muss schon vorher vergiftet worden sein.

### **7.3.3 Das Spielende**

*Autor: Jasmin Knott*

Wenn der Spieler denkt, ausreichend Informationen zusammengetragen zu haben, kann er Brok Koli bitten, alle NPCs zusammenzuholen, um den Mord aufzuklären. Er versammelt alle an der Bühne und gibt dem Spieler das Zeichen, das Mikrofon zu aktivieren und seine Theorie zu präsentieren. Sobald der Spieler fertig erklärt hat, muss er die Taste „Enter“ drücken, um fortzufahren. Als Nächstes soll der Spieler nun noch auflösen, dass keiner der Anwesenden der Mörder ist, sondern der Fremde aus dem Wald die Flasche vergiftet haben soll. Danach drückt er wieder „Enter“ und das Spiel zeigt einem, wie viel Prozent der richtigen Lösung erreicht wurden. Die Prozentzahl richtet sich nach der Präsentation der Mordtheorie und der Aufklärung des Mordes.

## 8. Umsetzung

### 8.1 Tools, API's und Frameworks

#### 8.1.1 GitHub

*Autor: Kai Friese*

Der Versionsverwaltungsdienst GitHub<sup>25</sup> von Microsoft erlaubt es, gemeinsam an der Entwicklung des Spiels zu arbeiten, Features zu verwalten und Konflikte, welche durch die Bearbeitung einer Datei von zwei Entwicklern entstehen können, zu vermeiden.

#### 8.1.2 ProCreate

*Autor: Jasmin Knott*

Die Raster-Grafik-Editor-App ProCreate<sup>26</sup> von Savage Interactive erlaubt es, Skizzen und Grafiken zu erstellen. Diese verwenden wir zum Planen und Konzeptionieren sowie zur Erstellung von Ingame-Grafiken des Spiels.

#### 8.1.3 Unity

*Autor: Kai Friese*

Wir verwenden die Laufzeit- und Entwicklungsumgebung Unity, mit der Langzeit-Support Version 2021.3.22f1, von Unity Technologies<sup>27</sup> zur Entwicklung unseres Spiels. Neben der Entwicklung eines 2D-Computerspiels bietet die UnityEngine vor allem das Erkennen und Aufnehmen von Spracheingaben des Spielers, welche wir für unsere Arbeit benötigen.

#### 8.1.4 Visual Studio

*Autor: Kai Friese*

Die Entwicklungsumgebung Visual Studio<sup>28</sup> von Microsoft wird verwendet, um die C# Scripts für Unity zu programmieren.

---

25 <https://github.com/>

26 <https://procreate.com/>

27 <https://unity.com/de>

28 <https://visualstudio.microsoft.com/de/>

## **8.2 Spracheingabe, Schlüsselwörter und Dialoge**

### **8.2.1 Spracheingabeverarbeitung**

*Autor: Kai Friese*

Das Ziel der Spracheingabeverarbeitung ist die korrekte Erkennung von Schlüsselwörtern. Unser erster Ansatz dafür war die Verwendung des KeywordRecognizer<sup>29</sup> aus dem Paket UnityEngine.Windows.Speech. Diesem Recognizer können Schlüsselwörter zugewiesen werden, welche einzeln ausgesprochen gut erkannt werden. Bei der Erkennung eines Schlüsselwortes wird ein Event aufgerufen, auf welches mithilfe der Methode „OnPhraseRecognized([...])“ reagiert werden kann. Problematisch an diesem Ansatz ist, dass lediglich ein KeywordRecognizer auf einmal aktiv sein kann. Somit können einzelne Schlüsselwörter nicht bestimmten NPCs zugeordnet werden, sondern müssen alle in einem Wörterbuch abgespeichert werden. Dadurch muss bei der Erkennung eines Schlüsselwortes zunächst einmal geprüft werden, ob und wie die Gesprächspartner darauf reagieren. Des Weiteren konnten Schlüsselwörter, welche in einem Satz eingebaut waren nicht erkannt werden, wodurch der KeywordRecognizer für unser Vorhaben ausschied.

Für unsern nächsten Ansatz und die finale Implementierung verwendeten wir ebenfalls aus dem Paket UnityEngine.Windows.Speech den DictationRecognizer<sup>30</sup>. Dieser kann keine Schlüsselwörter erkennen, sondern nimmt die gesamte Spracheingabe auf und wandelt sie in einen Text um. Die verwendete API greift dabei auf die Spracherkennungsfunktion von Windows zu. Dabei handelt es sich um eine Online-Spracherkennung, welche Microsoft Azure Cognitive Services – Speech Services<sup>31</sup> verwendet, einen cloudbasierten Spracherkennungsdienst.

---

29 <https://docs.unity3d.com/ScriptReference/Windows.Speech.KeywordRecognizer.html>

30 <https://docs.unity3d.com/ScriptReference/Windows.Speech.DictationRecognizer.html>

31 <https://azure.microsoft.com/de-de/products/cognitive-services/speech-services/>

Dementsprechend ist es notwendig, das entwickelte Spiel mit Internetzugang und den richtigen Freigaben auszuführen. Dazu muss, wie in Abbildung 18 veranschaulicht, in den Windows Einstellungen unter der Kategorie „Datenschutz“ der Punkt „Spracherkennung“ ausgewählt werden und die Online-Spracherkennung aktiviert werden. Die Spracherkennung nimmt zunächst die Audiodaten auf, komprimiert diese und überträgt sie an die Microsoft Azure-Cloud-Dienste. Dort analysieren fortschrittliche Sprachmodelle die Audiodaten und wandeln sie in Text um. Der fertige Text wird dann zur weiteren Verwendung zurückgesendet.

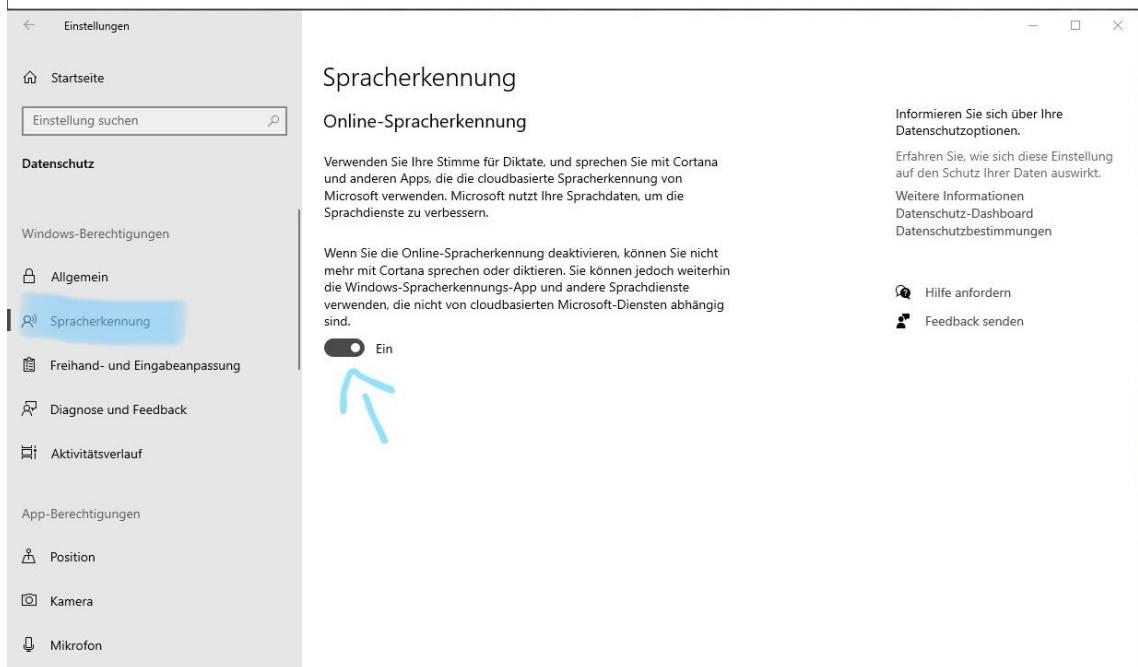


Abbildung 18: Anleitung zum Aktivieren der Online-Spracherkennung

Während der Spracheingabe können verschiedene Events ausgelöst werden, wobei das DictationResult Event für diese Arbeit am wichtigsten ist. Dieses Event wird ausgelöst, wenn der Spieler eine Pause beim Sprechen macht, in der Regel passiert dies am Ende eines Satzes. Nach der Eventauslösung kann man in einer Methode des DictationRecognizer darauf reagieren und die Eingabe verarbeiten.

Weitere Events, welche vom DictationRecognizer ausgelöst werden, sind DictationHypothesis, welches kontinuierlich ausgelöst wird, während der Spieler spricht. Dabei wird während der Eingabe bereits ein Text aus den gesprochenen Worten erzeugt. Das Event DictationComplete wird aufgerufen, wenn die Spracherkennung beendet wird. Dabei spielt es keine Rolle, ob bewusst die Methode „Stop()“ verwendet wird, oder ob ein Fehler oder Timeout auftritt. Zuletzt gibt es noch das Event DictationError, welches bei auftretenden Fehlern ausgelöst wird.

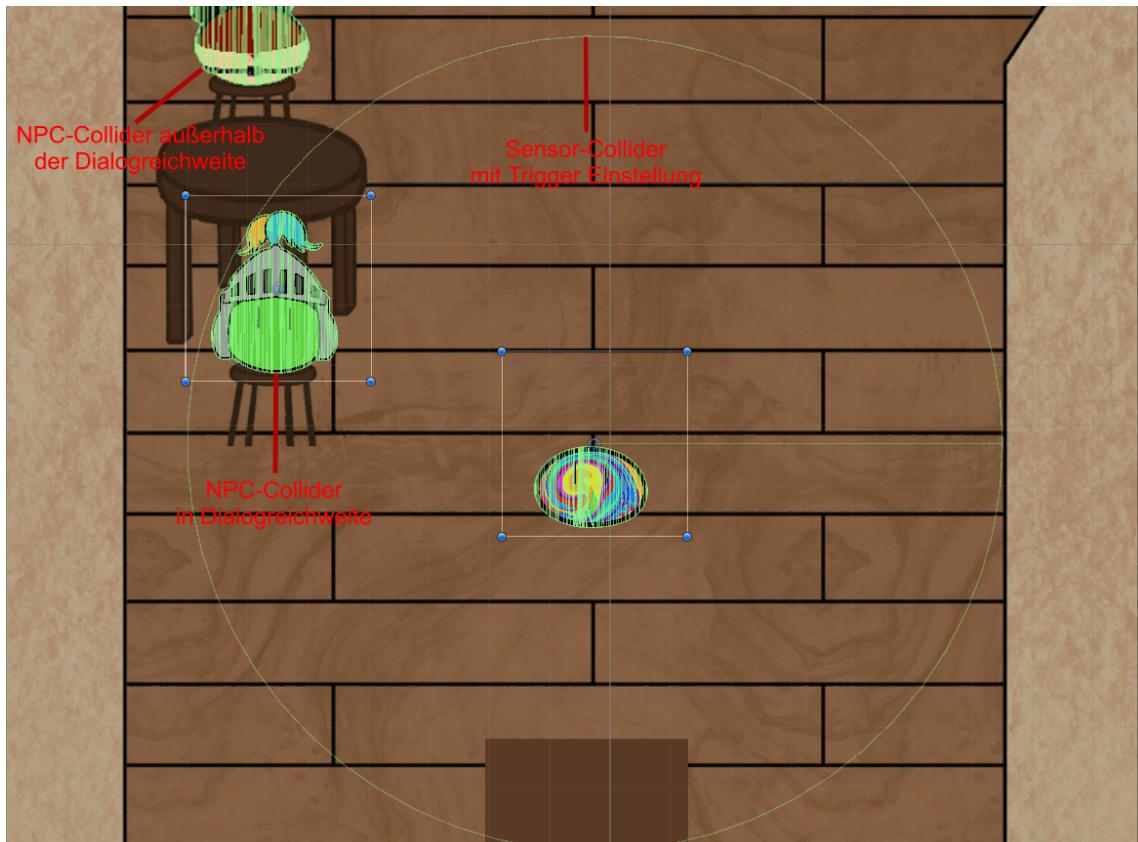


Abbildung 19: Collider Ansicht von Spieler Charakter, Sensor und NPCs

Um eine Konversation führen zu können, werden zunächst mehrere Parteien benötigt. Dafür verfügt der Spieler-Charakter, neben seinem regulären Collider, über einen Collider2D<sup>32</sup>, welcher als Trigger eingestellt ist. Dies bedeutet, dass Kollisionen mit anderen Collidern zwar erkannt werden und Events auslösen. Jedoch wird die Bewegung des Spieler Charakters nicht blockiert. Der Collider2D ist in Abbildung 19 als grüner Kreis um den Spieler Charakter dargestellt. Er ist viereinhalb mal so groß, wie der Spieler Charakter und fungiert als Sensor. Sobald der Collider eines anderen GameObjects<sup>33</sup> in den Sensor Colider eindringt, wird überprüft, ob es sich dabei um den Collider eines Objekts mit dem Tag<sup>34</sup> „NPC“ handelt. Ist dies der Fall, wird der zugehörige NPC dem Dialog hinzugefügt.

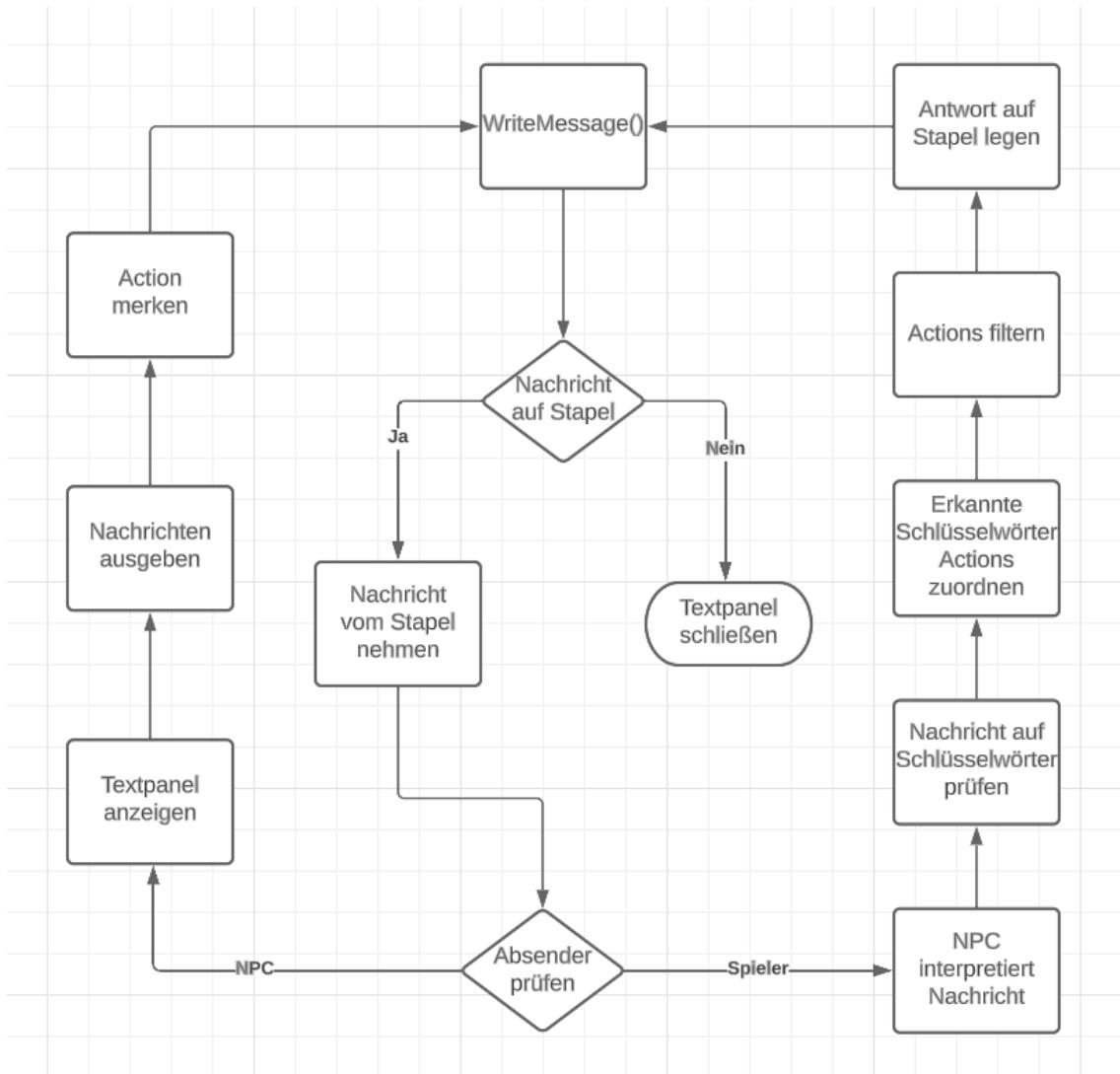


Abbildung 20: Ablauf der Nachrichtenverarbeitung

32 <https://docs.unity3d.com/Manual/Collider2D.html>

33 <https://docs.unity3d.com/ScriptReference/GameObject.html>

34 <https://docs.unity3d.com/Manual/Tags.html>

Wenn der Spieler eine Spracheingabe tätigt, wird der vom Cloud Service zurückgesendete Text in eine Nachricht umformatiert. Eine Nachricht ist dabei ein Datentransferobjekt, welches zwei Informationen als Strings abspeichert. Die Informationen sind der Name des Absenders der Nachricht und der zu übermittelnde Text. Die Nachricht wird an den Dialog weitergegeben und zunächst auf den Stapel der aktuellen Nachrichten abgelegt, um ein Durcheinander des Gesprächs zu vermeiden. Falls genau eine Nachricht auf dem Stapel liegt, wird sofort die Methode zum Verarbeiten der Nachricht aufgerufen.

Die Verarbeitung von Nachrichten beginnt durch die Methode „WriteMessage()“, deren Ablauf in Abbildung 20 dargestellt wird. In dieser Methode wird die unterste Nachricht des Stapels bearbeitet. Zunächst wird geprüft, ob das Ausgeben der eventuell vorherigen Nachricht abgeschlossen ist. Wenn die Nachricht nicht vom Spieler stammt, wird als Nächstes das Textpanel zum Anzeigen von Nachrichten aktiviert und der zu übermittelnde Text wird im Textpanel auf dem Bildschirm sichtbar. Der Name des Absenders wird dabei ebenfalls angezeigt.

Wenn die Nachricht jedoch vom Spieler stammt, interpretiert jeder NPC, welcher sich im Dialog befindet, die Nachricht separat. Dafür wird eine Coroutine<sup>35</sup> gestartet, welche es Unity ermöglicht, die Verarbeitung über mehrere Frames zu verteilen und so einen flüssigen Spielfluss sicherzustellen. Bei der Interpretation wird der Text zunächst in Kleinbuchstaben umgewandelt. Dann wird für jedes allgemeine und einzigartige Schlüsselwort mit der Methode „Text.Contains(s)“, wobei *s* die Schlüsselwörter repräsentiert, geprüft, ob der Text bekannte Schlüsselwörter beinhaltet. Bei einem Treffer wird die Action<sup>36</sup> welche dem Schlüsselwort zugeordnet ist in eine Liste aufgenommen. Wenn die Action bereits in der Liste ist, wird sie kein zweites Mal hinzugefügt. Dies verhindert, dass beispielsweise die Eingabe „Moin Moin“, zweimal die Begrüßung auslöst.

Im Anschluss wird die Action Liste durch Filter aussortiert. Diese Filter wurden in allgemeinen Fällen für alle NPCs gemeinsam und in speziellen Fällen für einige ausgewählte NPCs zusätzlich separat erstellt. Ein Filter besteht dabei aus einem Array von Actions, welche Schlüsselwörtern zugeordnet sind, die Konflikte verursachen können. Konflikte entstehen durch ähnliche oder mehrfach verwendete Schlüsselwörter, sowie Schlüsselwörter, welche Teilwörter von anderen Schlüsselwörtern sind. Das

35 <https://docs.unity3d.com/Manual/Coroutines.html>

36 <https://docs.unity3d.com/ScriptReference/Events.UnityAction.html>

Aussortieren durch den Filter funktioniert so, dass falls alle Einträge des Filters ausgelöst wurden, lediglich die erste Action im Filter beibehalten wird. Alle anderen Actions werden entfernt, wenn sie sowohl im Filter als auch in der Action Liste vorhanden sind.

Im Anschluss werden alle übrigen Actions in der Liste aufgerufen und somit auch die passenden Methoden ausgeführt. In einer solchen Methode stehen ein oder mehrere Antworten als String in einer Liste zur Verfügung. Die Antworten können sich auch bei den allgemeinen Schlüsselwörtern je nach NPC unterscheiden. Die Differenzierung der Antworten, wird dabei durch einen String-Vergleich mit dem Namen des NPCs, welcher die Nachricht interpretiert, ermittelt. Die Antwort wird aus einer der möglichen Antworten zufällig ausgewählt und nun ebenfalls in einer Nachricht an den Dialog übermittelt und auf den Stapel der bisherigen Nachrichten gelegt.

Im Anschluss wird der Methodename vom NPC abgespeichert, um sich zu merken, welche Antworten bereits gegeben wurden.

Das Anzeigen von Nachrichten findet ebenfalls in einer Coroutine statt. Dort wird jeder Buchstabe der Nachricht einzeln zur Anzeige hinzugefügt, um die Dauer und den Prozess des Sprechens darzustellen. Wenn dies abgeschlossen ist, wird das Event „onTextFinished“ ausgelöst. Der Dialog achtet auf dieses Event und beginnt bei der Auslösung damit, eine weitere Nachricht zu verarbeiten.

Damit der Spieler seine Eingaben nachvollziehen und überprüfen kann, ob er korrekt verstanden wurde, gibt es eine Eingabehistorie. Alle Eingaben vom Spieler werden dort notiert und am Bildschirmrand angezeigt, wobei frühere Nachrichten ab einem bestimmten Punkt nicht mehr angezeigt werden. Im Speicher sind die Nachrichten jedoch weiterhin bekannt. Dazu liegt eine Maske über der Eingabehistorie, welche den Bereich angibt, in welchem die Eingaben angezeigt werden. Sobald eine Nachricht, durch neuere Eingaben so weit nach unten verschoben wurde, dass sie aus dieser Maske „herausrutscht“ wird sie nicht mehr sichtbar. Dies geschieht spätestens nach 16 Eingaben durch den Spieler, wobei längere Eingaben dies früher auftreten lassen können.

Damit der Spieler auch weiß, wann er vom Spiel aufgenommen wird, kann er die Aufnahme mit der Leertaste starten und stoppen. Dabei wird während der Aufnahme ein rotes Symbol in der rechten, oberen Ecke angezeigt. Wenn der DictationRecognizer keine Verbindung zum Internet aufbauen kann, versucht dieser sich, solange neu zu

starten, bis eine Internetverbindung gegeben ist. Da der DictationRecognizer ohne Internet nicht richtig startet, lässt er sich auch nicht stoppen. Dadurch entsteht fälschlicherweise der Eindruck, dass sich die Spracherkennung im Spiel nicht mehr stoppen lässt.



Abbildung 21: Spieldia aus einem Dialog mit Eingabehistorie und Aufnahmesymbol

Die Abbildung 21 zeigt unten das Textpanel (Violett) mit dem Namen des Sprechenden NPC (rot) und einer gegebenen Antwort (gelb). Auf der rechten Seite ist die Eingabehistorie zu sehen. Neue Spielereingaben werden dort immer als oberstes angezeigt. Das rote Aufnahmesymbol befindet sich in der oberen rechten Bildschirmecke.

### 8.2.2 Schlüsselwörter

Autor: Jasmin Knott

Jeder NPC verfügt über drei verschiedene Arten von Schlüsselwörtern. Zunächst haben alle NPCs die allgemeinen Schlüsselwörter, wobei es sich um Wörter handelt, mit denen jeder umgehen kann (zum Beispiel: Begrüßungen und Verabschiedungen).

Neben den allgemeinen Schlüsselwörtern verfügen einige NPCs über einzigartige Schlüsselwörter, diese bestehen aus Wörtern, welche nicht für jeden NPC Sinn ergeben (zum Beispiel: Beim Gastwirt kann man mit dem Schlüsselwort „Zimmer“ einen

Schlafplatz für die Nacht buchen, bei einem anderen Gast ist dies nicht möglich.)

Zuletzt gibt es noch Antwort-Schlüsselwörter, welche zum Einsatz kommen, wenn der NPC Fragen stellt, auf welche der Spieler reagiert. Dazu ist es notwendig, den bisherigen Gesprächsverlauf abzuspeichern. Dies geschieht, indem sich jeder NPC separat die Antworten, welche vom NPC gegeben wurden, in einer Liste abspeichert. Dadurch können Reaktionen wie „Ja“ und „Nein“ als Antwort auf die letzte passende Frage zugeordnet werden.

Des Weiteren werden viele Schlüsselwörter gruppiert, da sie inhaltlich dieselbe Aussage tätigen und der NPC reagiert auf alle Wörter einer Gruppe gleich. So werden beispielsweise Begrüßungen wie "Hallo", "Guten Tag" und "Moin" zusammengefasst und der NPC grüßt zurück.

### **8.2.3 Dialoge**

*Autor: Jasmin Knott*

Um einen Dialog führen zu können, benötigt man zunächst mehrere Teilnehmer. Dazu fügen wir jeden NPC, welcher in die „Hörreichweite“ des Spielers eintritt, dem Dialog hinzu. Beim Verlassen der Hörreichweite wird der entsprechende NPC wieder aus dem Dialog entfernt.

Um Nachrichten der NPCs auszugeben, verwenden wir ein Textpanel, welches eingeblendet wird, sobald neue Nachrichten generiert werden. Da auf diese Art nur ein NPC auf einmal sprechen kann, werden alle Nachrichten auf einem Stapel gespeichert und nacheinander abgearbeitet. Dadurch geht keine Nachricht verloren und alle Nachrichten bleiben in der richtigen Reihenfolge.

Wenn eine Nachricht an der Reihe ist ausgegeben zu werden, wird zunächst sichergestellt, dass das Textpanel eingeblendet ist. Dann wird eine Coroutine gestartet, welche das Schreiben weiterer Nachrichten verhindert, bis diese Nachricht abgearbeitet ist. Daraufhin wird in der Coroutine der Text aus der Nachricht langsam im Textpanel ausgegeben. Zum Schluss wird das Schreiben weiterer Nachrichten wieder ermöglicht und ein Event ausgelöst, dass diese Nachricht fertig verarbeitet wurde. Die Nachricht wird vom Stapel genommen und der Stapel wird weiter abgearbeitet.

In den nachfolgenden Abbildungen 22 bis 32, ist ein beispielhafter Dialog aus dem Spiel dargestellt. Das Gespräch wird vom Spieler mit den Worten „Guten Tag“

begonnen, worauf die NPCs „Brok Koli“ und „Jolanda“ in Abbildungen 22 und 23 antworten.

Als Nächstes will der Spieler wissen, wo er sich befindet (Abbildungen 24). In Abbildungen 25 antwortet Brok Koli und in Abbildung 26 erfährt der Spieler zusätzlich von der Schankmaid Jolanda, dass dies der einzige Ort zum Übernachten in der Gegend ist. Der Spieler bestellt sich daraufhin ein Zimmer in Abbildung 27 und bekommt in Abbildung 28 ein passendes Angebot vom Wirt Brok Koli. Die Abbildung 29 zeigt die Bestätigung des Spielers. In Abbildung 30 erfährt der Spieler, wo das Zimmer ist und bedankt sich in Abbildung 31. Das Gespräch wird in Abbildung 32 von Brok Koli beendet.

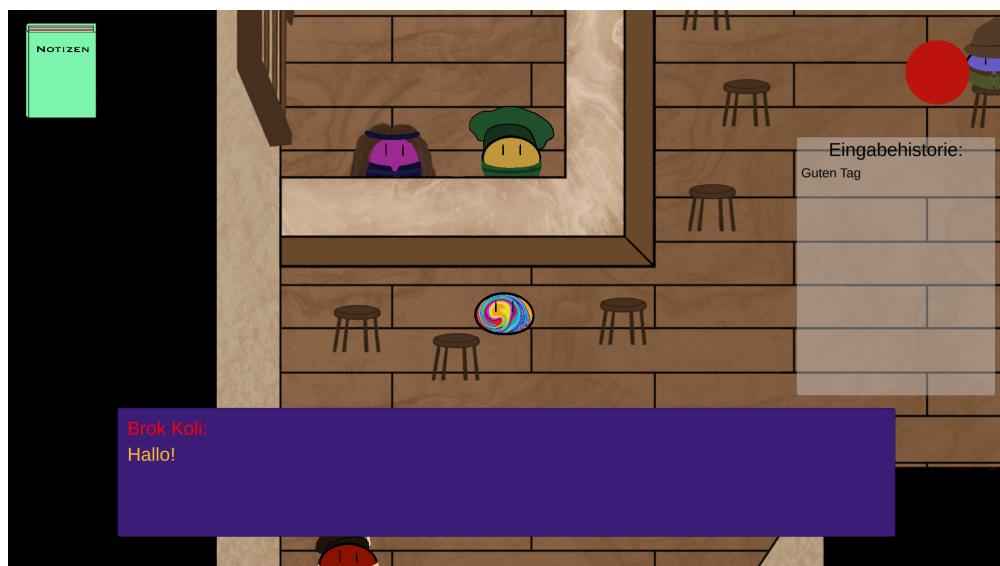


Abbildung 22: Dialog mit der Eingabe "Guten Tag"

Jolanda:  
Guten Tag!

Abbildung 23: Antwort von Jolanda auf "Guten Tag"

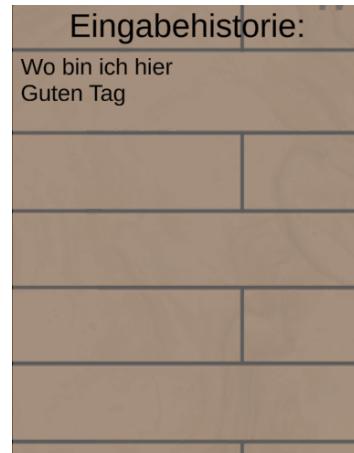


Abbildung 24: Eingabe "Wo bin ich hier"

**Brok Koli:**

Rübenhain ist wohl das nächste Dorf in der Gegend.

Abbildung 25: Antwort von Brok Koli auf "Wo bin ich hier"

**Jolanda:**

Nicht weit von hier liegt das Dorf Rübenhain, doch das hier ist das einzige Gasthaus in der Umgebung.

Abbildung 26: Antwort von Jolanda auf "Wo bin ich hier"

Eingabehistorie:	
na wenn das so ist dann würde ich gerne einen Schlafplatz haben	
wo bin ich hier Guten Tag	

Abbildung 27: Eingabe "Na wenn das so ist dann würde ich gerne einen Schlafplatz haben"

**Brok Koli:**

Möchtest du gerne die Nacht hier verbringen? Ich hätte da noch ein Zimmer für einen Silbertaler pro Nacht frei.

Abbildung 28: Antwort von Brok Koli auf "Na wenn das so ist dann würde ich gerne einen Schlafplatz haben"

Eingabehistorie:	
Das klingt super das nehme ich na wenn das so ist dann würde ich gerne einen Schlafplatz haben	
wo bin ich hier Guten Tag	

Abbildung 29: Eingabe "Das klingt super das nehme ich"

**Brok Koli:**

Sehr schön, hier hast du deinen Schlüssel, das Zimmer ist Oben. Ich wünsche eine gute Nacht!

Abbildung 30: Antwort von Brok Koli auf "Das klingt super das nehme ich"

Eingabehistorie:	
Danke sehr	
Das klingt super das nehm ich	
Na wenn das so ist dann würde	
ich dir gerne einen Schlafplatz	
nehmen	
Wo bin ich hier	
Guten Tag	

Abbildung 31: Eingabe "Danke sehr"

**Brok Koli:**

Es war mir ein Vergnügen.

Abbildung 32: Antwort von Brok Koli auf "Danke sehr"

## 8.2.4 Gesprächsthemen

*Autor: Jasmin Knott*

Da es endlos viele Möglichkeiten für Gesprächsthemen gibt, wurde der Rahmen für diese Bachelorarbeit auf das Szenario eines mittelalterlichen Gasthauses in einer Fantasiewelt begrenzt. Dadurch wurden Gesprächsthemen der Moderne, wie Politik, Technik und generelle Themen der realen Welt ausgeschlossen. Da die Informationsverbreitung im Mittelalter nicht sehr ausgeprägt war, können die NPCs nur über lokale Ereignisse sprechen. Das umfasst die Kenntnis der Ortschaft und Umgebung sowie die vorhanden NPCs.

Zu Spielbeginn hat der Spieler die Möglichkeit, sich mit der Welt, den NPCs und der Sprachsteuerung etwas vertraut zu machen. Mögliche Gesprächsthemen sind dabei allgemeiner Smalltalk wie zum Beispiel Begrüßungen und Fragen zu dem Beruf, der Herkunft und dem Wohlergehen. Bei einigen NPCs sind auch Fragen über besondere äußere Merkmale der Charaktere möglich, dazu zählt beispielsweise der Helm des

Söldners. Beim Wirt kann man auch über die Grundbedürfnisse, wie Essen, Trinken und einen Schlafplatz sprechen. Werden diese Themen bei anderen NPCs angesprochen, reagieren diese nicht. Die meisten dieser Gesprächsthemen sind jedoch nicht relevant, um die Aufgabe des Spiels zu lösen. Lediglich die Frage nach einem Schlafplatz ist relevant, um die Geschichte voranzutreiben.

Das Schließen gehen initialisiert die Hauptaufgabe unseres Szenarios, den Tod von Albert. Dieser Todesfall muss vom Spieler aufgeklärt werden, indem er die NPCs befragt und so das Geschehen rekonstruiert.

Dadurch werden auch neue Gesprächsthemen freigeschaltet. Dazu gehören Fragen zum Opfer, zur Leiche und dem Tatort, Tatmotive und Alibis. Auch mit dem Geist des Toten kann man sich nun unterhalten.

Wenn man ausreichend Fragen gestellt hat, kann der Gastwirt die NPCs an der Bühne versammeln, sodass der Spieler seine Lösung präsentieren kann. Dabei soll er die relevanten Informationen der NPCs zusammenfassen und den Schuldigen stellen. Nach dem Spiel kann man anhand einer Prozentzahl ablesen, wie viel der Spieler von der Geschichte korrekt rekonstruiert hat. Dazu hat er auch die Möglichkeit, sich die richtige Lösung sowie die eventuell fehlenden Informationen anzeigen zu lassen.

## 9. Ergebnisse

*Autor: Jasmin Knott*

Um nach der Fertigstellung des Spiels eine geeignete Bewertungsgrundlage zu haben, wurden Fragebögen für Testspieler erstellt und im Anschluss analysiert. Der Fokus der Fragen liegt darauf, wie die sprachgesteuerte Kommunikation in Computerspielen im Vergleich zu bisherigen Modellen wirkt. Dazu wurden acht Testspieler, davon eine weibliche und sieben männliche, im Alter von 20 bis 30 Jahren organisiert. Die Testspieler setzten sich aus Freunden und Bekannten zusammen. Diese Testspieler haben alle viel bis sehr viele Erfahrungen in Computerspielen allgemein. Die Testspieler haben dazu einen Fragebogen ausgefüllt, welcher den Fokus auf die Kommunikation legt. Ursprünglich war auch ein Turing-Test [15] angedacht, doch diesen kann unser System zum aktuellen Zeitpunkt nicht bestehen. Der Grund liegt in der vergleichsweise geringen Menge an Schlüsselwörtern, dazu sind die Gespräche nicht tiefgehend genug. Zudem antworten die NPCs nicht auf alle Eingaben und das Erinnerungsvermögen ist nicht ausgereift genug, um eine konsistente Kommunikation aufrechtzuerhalten.

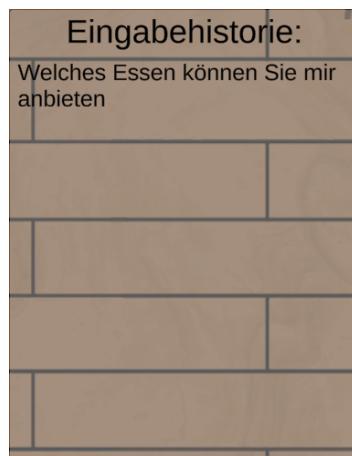


Abbildung 33: Eingabe "Welches Essen können Sie mir anbieten"



Abbildung 34: Antwort von Brok Koli auf "Welches Essen können Sie mir anbieten"

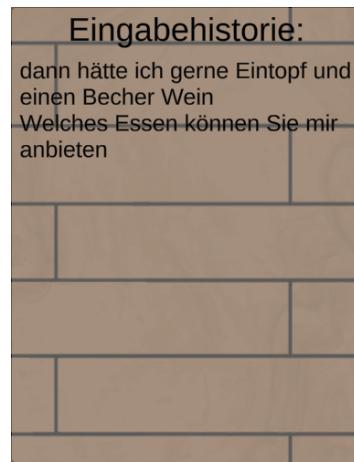


Abbildung 35: Eingabe "dann hätte ich gerne Eintopf und einen Becher Wein"

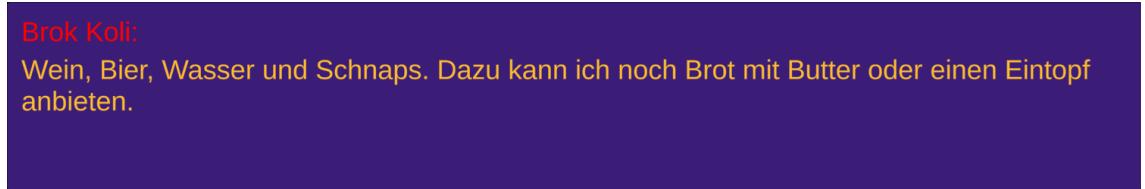


Abbildung 36: Antwort von Brok Koli auf "dann hätte ich gerne Eintopf und einen Becher Wein"

Die Abbildungen 33 bis 36 zeigen einen beispielhaften Dialog, welcher in einem Turing-Test unseren Chatbot als KI entlarven könnte. Zunächst erfragt der Spieler in Abbildungen 33, welches Essen der Wirt ihm anbieten kann. Dieser benennt in Abbildung 34 die Möglichkeiten, woraufhin der Spieler in Abbildungen 35 Eintopf und Wein bestellt. Der Chatbot erkennt das nach Essen und Trinken angesprochen wurden und gibt so als Antwort eine Aufzählung von Getränken und Speisen in Abbildungen 36. Da die Bestellung des Spielers nicht dem Angebot widerspricht, gibt es keinen Grund, das Angebot noch einmal aufzuzählen. Ein Mensch würde in dieser Situation anders antwortet, wodurch die KI generierte Antwort offensichtlich wird.

Der Fragebogen für Testspieler gab uns zuerst einmal Informationen über die Testspieler selber, indem demografische Daten wie Alter und Namen erfasst wurden. Nach den allgemeinen Informationen über die Spieler, beschäftigt sich der Fragebogen mit spiel relevanten Informationen. Dazu konnte in einem Ranking System unter Anderem die Spielmechanik oder auch die Kommunikation bewertet werden. So konnten wir ablesen, wie viel Spaß das Spiel machte oder wie gut die Kommunikation im Allgemeinen funktioniert hat.

Da Testspieler häufig eigene Ideen und Verbesserungsvorschläge haben oder auch gerne tiefer auf einige Punkte eingehen möchten, können auf der zweiten Fragebogenseite

Freitext antworten geschrieben werden. Um dort ein paar Hilfestellungen zu geben, zu welchem Thema etwas geschrieben werden kann, wurden drei Fragen vorgegeben. Durch diese Fragen konnten wir Schlüsse darüber ziehen, was den Testspielern besonders gut oder schlecht gefallen hat, sowie die Ideen sammeln, wie das Spiel verbessert oder geändert werden könnte.

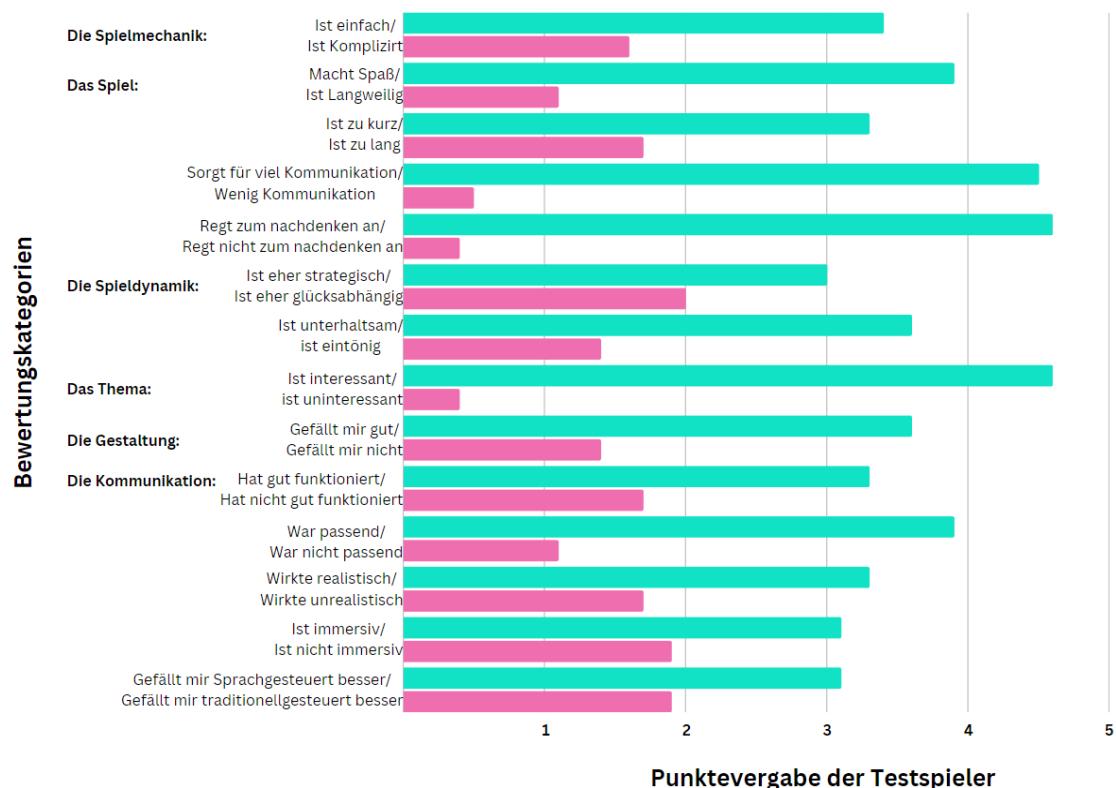


Abbildung 37: Quantitative Auswertung der Fragebögen

Explizierte Hilfestellungen für die Testspieler gab es nicht. Sie mussten das Spiel selbstständig kennenlernen und herausfinden, wie sie mit den NPC's zu interagieren haben. Dadurch konnten wir einen unverfälschten Eindruck erhalten, wie schwierig, intuitiv und natürlich die Spielmechanik ist. Lediglich ein Tipp wurde gegeben. Bei den Gruppen der NPCs welche am Tisch saßen, muss der Spielercharakter sehr nah dran stehen, um in der Reichweite für alle NPCs an diesem Tisch zu sein. Das Balkendiagramm in Abbildungen 37 stellt die Ergebnisse der Fragebögen dar. Dabei ist zu sehen, dass alle Bewertungskriterien im Schnitt positiv ausfielen. Dies ist daran zu erkennen, dass der grüne Balken größer ist als der pinke Balken.

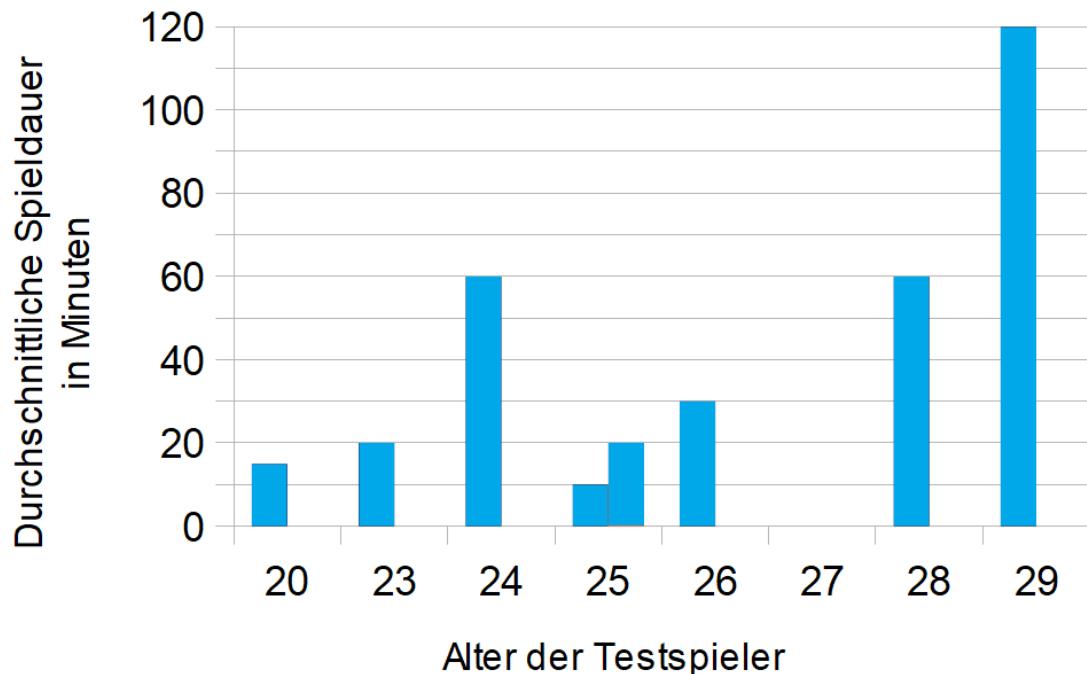


Abbildung 38: Durchschnittlich Spieldauer und Alter der Testspieler

Im Allgemeinen kam das Spiel bei den Spielern gut an und hat Spaß gemacht. In Abbildung 38 ist dargestellt, wie lang die Testspieler im Durchschnitt pro Spieldurchlauf nach eigenen Angaben benötigt haben. Die Länge der Spieldauer wurde als passend eingestuft, durchschnittlich wurden etwa 50 Minuten benötigt, um das Spiel abzuschließen. Dabei wurde der schnellste Durchlauf nach zehn Minuten und der langsamste nach zwei Stunden abgeschlossen. Es ist also nicht möglich, wie in traditionell gesteuerter Kommunikation, durch einfach nur ausprobieren aller Antworten die richtige Lösung zu finden. Eher im Gegenteil, die Spieler müssen sich aufmerksamer mit den NPCs und ihrer Spielwelt auseinandersetzen. Dadurch wird jedoch auch der Schwierigkeitsgrad des Spiels angehoben. Ein aufmerksamer Spieler, der schnell auf die richtigen Fragen kommt, hat das Spiel auch entsprechend schnell durchgespielt. Wobei hingegen ein Spieler, welcher für die Aufgabe des Spiels irrelevante Fragen stellt, deutlich länger braucht. Dies wurde von den Spielern jedoch als positiv wahrgenommen und alle Testspieler sind der Meinung, dass das Spiel zum Nachdenken anregt. Ebenso fanden alle Tester, bis auf einen, dass das Spiel für viel Kommunikation innerhalb und außerhalb der Spielwelt sorgt. Einige Testspieler haben eigene Freunde und Verwandte hinzugezogen, um das Spiel zu präsentieren und darüber zu sprechen.

Die Spielmechanik wurde von den Testspielern als passend, tendenziell jedoch eher

einfach eingestuft. Die Spieldynamik wurde dabei vom Großteil als vom Glück abhängig eingestuft, da aufgrund der begrenzten Schlüsselwörter „gehofft“ werden musste, dass man die richtigen Worte verwendet. Einige Spieler hatten beispielsweise schon die richtige Frage gestellt, jedoch falsch formuliert und mussten daraufhin die Fragestellung abändern. Dazu wurde sich auch gewünscht, mehr Hinweise zu geben, oder besser erkenntlich zu machen, wenn relevante Informationen auftauchen. Positiv wurde dabei jedoch die Eingabehistorie bewertet, welche anzeigt, was das Spiel von den Worten des Spielers verstanden hat. So wusste der Spieler, ob er nur nicht verstanden wurde oder ob der NPC keine Antwort auf die Eingabe hatte. Ebenfalls positiv wurde die Anzeige des roten Aufnahmesymbols empfunden, da so klar war, ob der Spieler aufgenommen wurde oder nicht.

Der Unterhaltungsfaktor des Spiels wurde dabei eher durchschnittlich eingestuft, wobei das Thema als sehr interessant empfunden wurde. Mit den NPCs richtig sprechen zu können, ist eine neue Erfahrung und die Tester denken, dass dies in der Zukunft eine große Rolle spielen kann. Diese Art der Kommunikation passt zwar nicht zu allen Genres, doch im richtigen Genre angewendet hat es sehr viel Potenzial. Laut Testspielern kann die Technik gerade bei Spielen wie Detektiv- oder Rollenspielen und in Kombination mit virtueller Realität oder erweiterter Realität die Immersion eines Spiels enorm verbessern.

Das Kernelement, also die sprachgesteuerte Kommunikation, hat bei der Hälfte der Testspieler gut und bei der anderen Hälfte eher schlecht funktioniert. Im Allgemeinen wurde sie für den ersten Ansatz in diesem Thema als in Ordnung, jedoch noch ausbaufähig eingestuft. Es sind einige Fehler aufgetreten, in welchen die Wörter des Spielers nur falsch oder gar nicht interpretiert wurden, was zu Frustration führen kann. Positiv wurde dazu vermerkt, dass Schimpfwörter vom verwendeten DictationRecognizer automatisch zensiert wurden. Dadurch werden toxisches Verhalten und Hass in Videospielen nicht unterstützt. Insgesamt wurde die sprachgesteuerte Kommunikation eher durchschnittlich angenommen. Einige finden die traditionelle Art der Dialoge in Videospielen besser, andere empfinden die sprachgesteuerte Dialogform besser. Alle sind jedoch der Meinung, dass es auch stark auf das Spiel ankommt, in welchem diese Technik verwendet wird. Ebenfalls entstehen durch diese Kommunikationsart Einschränkungen, da es an öffentlichen Plätzen oder unterwegs eher schlecht angewendet werden kann.

Die Spieler fanden es gut, dass trotz der verhältnismäßig niedrigen Zahl an Schlüsselwörtern die NPCs unterschiedliche Antworten gegeben haben und so eine eigene Persönlichkeit verkörperten. Die Antworten und Dialoge wurden auch als größtenteils passend und realistisch eingestuft, wobei es vorkommt, dass Antworten nicht genau auf die Frage passen.



Abbildung 39: Dialog mit der Eingabe "Warum sollte jemand Albert vergiften?"

Die Abbildung 39 zeigt ein Beispiel, in welchem vom Spieler gefragt wurde „Warum sollte jemand Albert vergiften?“. Die Antwort des NSCs passt zwar zum Thema der Vergiftung, geht jedoch nicht passend auf die Frage des Spielers ein.

Jedoch freuten sich die Spieler, wenn sie auf kreative, oder außergewöhnliche Fragen auch eine Antwort bekamen, wodurch die Motivation, weiter zu fragen, gesteigert wurde. Ein Beispiel dafür ist es, dass man sich zu Anfang des Spiels von dem NPC Albert einen Kartentrick zeigen lassen kann.

Das Setting eines Detektivspiels wurde als sehr passend für die sprachgesteuerte Kommunikation empfunden. Dazu wurde sich jedoch gewünscht, selber Notizen machen zu können, um so gesammelte Informationen zu vermerken. Eine passende Idee eines Testspielers dafür war die Möglichkeit, seinem Notizbuch eigene Einträge diktieren zu können. Für diesen Ansatz wäre es erforderlich zu differenzieren, ob eine Spracheingabe an NPCs gerichtet ist, oder an das Notizbuch. Dies wäre dadurch

realisierbar, verschiedene Tasten zur Aktivierung der Spracheingabe zu verwenden oder die Eingabe automatisch immer ans Notizbuch zu leiten, wenn dieses geöffnet ist.

Die Gestaltung des Spiels wurde als gewöhnungsbedürftig, jedoch passend, beschrieben und kam grundlegend bei den meisten Testern gut an. Optisch wirken die Charaktere niedlich und süß, allerdings war nicht in allen Situationen klar, wie welcher Charakter hieß. Bei einigen Grafiken führte die räumliche Perspektive zur Kritik, hatte jedoch keinen großen negativen Einfluss auf das Spiel.

Die Immersion des Spiels wurde wieder sehr unterschiedlich wahrgenommen. Insgesamt wurde sie jedoch aufgrund von mangelnden Antworten und Schlüsselwörtern nicht so gut wie erhofft. Dabei sind Testspieler, welche das Spiel als weniger immersiv angegeben haben, der Meinung, dass größere Sprachpakete dies verbessern könnten.

## **10. Diskussion**

### **10.1 Eingebaute Schlüsselwörter**

*Autor: Jasmin Knott*

Bei der Entwicklung der Schlüsselwörter sind drei Probleme aufgetreten. Zum einen gibt es einige problematische Wörter, wie das Wort „hi“. Diese sind vor allem kurze Wörter, welche in anderen Wörtern auftreten können. So wird beispielsweise bei den Wörtern „hier“ und „hinten“ das Schlüsselwort „hi“ herausgelesen und die NPCs begrüßen den Spieler. Der Grund dafür ist, dass die Spracheingabe zunächst in Text umgewandelt wird, bevor die Überprüfung nach Schlüsselwörtern stattfindet. So beinhaltet der String „hier“ eins zu eins das Schlüsselwort „hi“ und löst dieses aus.

Des Weiteren entstand ein Problem dadurch, dass einige Schlüsselwörter in einem anderen Zusammenhang ebenfalls Verwendung finden. Ebenso ist es vorgekommen, dass allgemeine Schlüsselwörter auch in einzigartigen Schlüsselwörtern auftraten.

Die Lösung für diese Probleme sind Filter, welche bei der Auslösung mehrerer Schlüsselwörter in einem Satz prüfen, welche korrekt ausgelöst wurden und welche nicht verwendet werden sollen.

Ein weiteres Problem sind die verschiedenen Zeitformen von Wörtern. So müssen für alle verwendeten Verben, zahlreiche Schlüsselwörter angelegt werden, um alle Zeitformen abzufangen.

Im Rahmen dieser Bachelorarbeit ist es nicht möglich alle Wörter, bei denen diese Probleme auftreten könnten, abzufangen, jedoch wurde das Prinzip zumindest für einige wenige dieser Fälle angewendet. Einige Beispiele für verarbeitete problematische Wörter sind:

- „Wein“, welches in Konflikt mit „weinen“, „Weinflasche“ und „Weinhändler“ steht
- „Geist“ und „passiert“, sind jeweils Schlüsselwörter, welche, in Kombination mit weiteren Wörtern, sowohl als allgemeine Schlüsselwörter Antworten bei allen NSCs auslösen können, als auch als einzigartige Schlüsselwörter beim Stallburschen verwendetet werden
  - Das Schlüsselwort „Geist“ ist in den allgemeinen Schlüsselwörtern

vorhanden, um Beispielsweise die Frage: „Hast du schon mal einen Geist gesehen?“, stellen zu können. Den Stallburschen kann zusätzlich noch gefragt werden, ob er Alberts Geist sieht, da sich beide am gleichen Ort befinden.

- Mit dem Schlüsselwort „passiert“ können alle NPCs nach dem Schrei befragt werden, welcher in der Nacht zu hören ist. Der Stallbursche kann zusätzlich darauf reagieren, indem er beispielsweise gefragt wird „Was ist dann passiert?“, um weitere Information zum Geschehen zu geben.

## ***10.2 Implementierungsfehler***

*Autor: Kai Friese*

In der Implementierung besteht ein Problem, welches die Verarbeitung der Spracheingabe behindert. Es kann passieren, dass die Sprachverarbeitung während des Spiels langsamer wird und sogar durch einen Timeout aufhört zu funktionieren. Der Grund für den Fehler liegt bei dem aus der UnityEngine verwendeten, DictationRecognizer. Dieser ist für die Aufnahme der Sprache und die Umwandlung in Text zuständig. Durch unsere Analyse haben wir festgestellt, dass wenn dieser Fehler auftritt, die Verarbeitungsgeschwindigkeit der einzelnen Wörter deutlich abnimmt. Dazu kann der DictationRecognizer die Verarbeitung nicht mehr abschließen und wirft schließlich einen Timeoutfehler.

Sobald der Fehler auftritt, lässt sich der DictationRecognizer nicht mehr neu starten. Ab diesem Punkt hilf ein Neustart des Spiels, wodurch das Problem zumindest vorübergehend beseitigt ist. Testspieler haben berichtet, dass es helfen kann, das Spielfenster aus dem Fokus zu nehmen und nach kurzer Zeit wieder zum Spiel zu wechseln.

Der Fehler ist nicht gut reproduzierbar, da er bereits nach wenigen Minuten auftreten kann, oder auch gar nicht. Zudem konnte kein festes Ereignis festgestellt werden, wodurch der Fehler auftritt.

Trotz verschiedener Lösungsansätze, wie regelmäßiges neu Starten, Freigeben der verwendeten Ressourcen sowie automatische Timeouts bei längerer Stille, war es nicht möglich, das Problem im gegebenen Zeitraum zu beheben.

### **10.3 Kommunikationsschwierigkeiten**

*Autor: Kai Friese*

Das Spiel, welches für diese Arbeit entwickelt wurde, verfügt über mehr als 1000 Schlüsselwörter. Im Vergleich zu den Möglichkeiten, welche die deutsche Sprache und auch viele andere bieten, ist diese Zahl, verschwindet gering. Dadurch gibt es sehr viele Eingaben, auf welche der Schlüsselwort-basierte Chatbot des Spiels keine Antwort hat. Wenn ein Spieler solche Eingaben tätigt und keine Reaktion erhält, wirkt sich das negativ auf das Spielerlebnis aus.

Diese Problematik kann zunächst dadurch verbessert werden, eine Standardantwort einzubauen, falls der NPC keine Antwort auf die Eingabe des Spielers hat. Nachdem die Rückmeldungen der Testspieler eingeholt wurden, wurden solche Standardantworten implementiert. Behoben würde das Problem durch mehr Schlüsselwörter, oder große Sprachmodelle in Kombination mit tiefen neuralen Netzwerken und KI generierten Antworten.

### **10.4 Begrenzte Anwendbarkeit**

*Autor: Jasmin Knott*

Die sprachgesteuerte Kommunikation ist nicht für alle Plattformen und Spielgenres geeignet. Zunächst ergibt die Anwendung dieser Technik nur in Spielen Sinn, in welchen die Kommunikation mit NPCs eine tragende Rolle spielt. Sprachgesteuerte Kommunikation sollte beispielsweise nicht in Spielen eingesetzt werden, in welchen die Kommunikation schnell gehen muss, nur zur reinen Informationsweitergabe dient oder erst gar keine Kommunikation stattfindet.

Spielgenres, welche sich besser für die Verwendung von sprachgesteuerter Kommunikation eignen, sind:

- Abenteuerspiele
- Rollenspiele
- Simulationen
- Horrorspiele
- Lernspiele

- Detektiv- und Rätselspiele
- Einzelspielerspiele

Spiele aus folgenden Genres eignen sich tendenziell weniger für diese Art der Kommunikation:

- Geschicklichkeitsspiele
- Fighting Games
- Shooter
- Plattformspiele
- Party Spiele
- Strategiespiele
- Mehrspieler-Spiele
- E-Sport-Spiele

Dennoch ist eine eventuell angepasste Anwendung in allen Spielgenres möglich.

Zusätzlich wird die Verwendung von sprachgesteuerter Kommunikation auch durch die Hardware eingeschränkt. Ohne Mikrofon oder mit einem Mikrofon mit schlechter Sprachqualität ist diese Kommunikation nicht zu verwenden. Des Weiteren kann diese Technik nicht bei lauter Umgebung eingesetzt werden, generell ist die Anwendung zu Hause in einem stillen Raum am besten geeignet.

Doch wenn man die sprachgesteuerte Kommunikation mit virtueller Realität oder erweiterter Realität kombiniert, kann die Immersion eines Spiels neue Dimensionen erreichen.

# **11. Fazit und Ausblick**

*Autor: Kai Friese und Jasmin Knott*

## **11.1 Fazit**

In diesem Projekt wird die Forschungsfrage „*Ist es möglich, durch Spracheingabe und deren Verarbeitung die Kommunikation in Computerspielen zwischen Spieler und Nicht-Spieler-Charakteren realistischer wirken zu lassen?*“ behandelt. Dazu wurde ein Computerspiel mit Schlüsselwort-basierten Chatbot entwickelt, welcher auf eingaben durch das Mikrofon reagiert. Dieser Chatbot ist jedoch zum aktuellen Zeitpunkt nicht ausgereift genug, um einen Turing-Test zu bestehen. Doch die sprachgesteuerten Dialoge bieten dem Spieler eine deutlich freiere und dadurch auch realistischere Kommunikation, als es traditionelle Dialoge mit vorgegebenen Antworten können.

Wie realistisch die Kommunikation tatsächlich ist, hängt jedoch stark von der zugrunde liegenden Menge an Schlüsselwörtern ab. Durch zu wenig Schlüsselwörter tritt Frustration bei den Spielern auf und die Immersion sowie der Realismus der Kommunikation leiden darunter. Mit großen Mengen an Schlüsselwörtern kann die Eingabe des Spielers besser verstanden werden, wodurch die Kommunikation immersiver und realistischer wird.

Ebenso wichtig wie das Verständnis der Spracheingabe ist die Ausgabe passender Antworten. Fehlende oder unpassende Antworten mindern den Realismus der Kommunikation. Witzige oder unerwartete Antworten können hingegen zu einem Erfolgserlebnis führen und den Spieler dazu anregen, weitere Fragen zu stellen.

Die Wahl des Spielgenres und der Spieleplattform hat Einfluss darauf, wie sinnvoll es ist, die sprachgesteuerte Kommunikation zu verwenden und wie sie auf den Spieler wirkt. Einige Genres und Plattformen eignen sich dabei besser für den Einsatz von sprachgesteuerter Kommunikation als andere.

Die Spielumgebung kann das Spielerlebnis stark beeinflussen, so ist ein ruhiger Raum, in welchem man ungestört Sprechen kann, am besten für die Anwendung von sprachgesteuerter Kommunikation geeignet. Hintergrundgeräusche können die Spracherkennung beeinflussen, vor allem Gespräche können dazu führen, dass Eingaben stattfinden, welche nicht vom Spieler stammen.

Von den Testspielern wurde die Kommunikation insgesamt als durchschnittlich immersiv wahrgenommen, was mit zu wenig Schlüsselwörtern und mangelnden Antworten begründet wurde.

Durch eine gute Umsetzung dieser Technik entstehen auch weitere Vorteile. Durch den Mangel an vorgegebenen Dialogoptionen wird der Spieler zum Nach- und Mitdenken angeregt. Dies führt dazu, dass der Spieler das Spielgeschehen aufmerksamer verfolgt. Außerdem wird die Kommunikation angeregt, sodass Spieler sich mehr mit den NPCs unterhalten. Nach Angaben weniger Testspieler wurde auch außerhalb des Spiels die Kommunikation mit Freunden und Familienmitgliedern angeregt.

Grundsätzlich ist es also möglich, die Kommunikation in Computerspielen zwischen Spieler und Nicht-Spieler-Charakteren durch den Einsatz von sprachgesteuerter Kommunikation realer wirken zu lassen. Doch nur die Sprachsteuerung allein kann dies nicht vollständig bewerkstelligen. Es gehören noch weitere Faktoren als die reine Eingabeform dazu, um eine Kommunikation real wirken zu lassen.

## **11.2 Ausblick**

Die sprachgesteuert Kommunikation in Computerspielen kann durch die Anwendung weiterer Techniken noch deutlich verbessert werden. Anstelle eines Schlüsselwort-basierten Chatbots könnten große Sprachmodelle mit einer KI, welche auf DNN basiert, zum Einsatz kommen. Dazu kann auch die automatische Generierung von Antworten eingesetzt werden. Dadurch kann die Zahl der unpassenden Antworten eingeschränkt werden und es wird verhindert, dass auf eine Eingabe des Spieler keine Antwort gegeben wird. Jedoch muss bei der automatischen Generierung von Antworten auf ethisch problematische Ausgaben geachtet werden.

Um ein möglichst immersives Spielerlebnis zu erhalten, ist vor allem die Kombination von sprachgesteuerter Kommunikation mit virtueller Realität interessant.

## **12. Eidestattliche Erklärung**

Name: Kai Friese

Adresse: Auf der Höh 57, 53819 Neunkirchen-Seelscheid

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbst angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

---

Ort, Datum

---

Unterschrift

Name: Jasmin Knott

Adresse: Daimlerstraße 17, 53757 Sankt Augustin

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbst angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

---

Ort, Datum

---

Unterschrift

## 13. Literaturverzeichnis

1. M. Abdullah, A. Madain and Y. Jararweh, "ChatGPT: Fundamentals, Applications and Social Impacts," 2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS), Milan, Italy, 2022, pp. 1-8, doi: 10.1109/SNAMS58071.2022.10062688.  
<https://ieeexplore.ieee.org/document/10062688>
2. E. Cambria and B. White, "Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]," in IEEE Computational Intelligence Magazine, vol. 9, no. 2, pp. 48-57, May 2014, doi: 10.1109/MCI.2014.2307227. <https://ieeexplore.ieee.org/document/6786458>
3. CD Projekt Red: The Witcher 3: Wild Hunt (2015)  
<https://www.thewitcher.com/de/witcher3> (letzter Zugriff: 25.03.2023)
4. G. Daniel and J. Cabot, "The Software Challenges of Building Smart Chatbots," 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 2021, pp. 324-325, doi: 10.1109/ICSE-Companion52605.2021.00138.  
<https://ieeexplore.ieee.org/document/9402515>
5. A. Gillioz, J. Casas, E. Mugellini and O. A. Khaled, "Overview of the Transformer-based Models for NLP Tasks," 2020 15th Conference on Computer Science and Information Systems (FedCSIS), Sofia, Bulgaria, 2020, pp. 179-183, doi: 10.15439/2020F20. <https://ieeexplore.ieee.org/document/9222960>
6. M. F. Hasani and Y. Udjaja, "Immersive Experience with Non-Player Characters Dynamic Dialogue," 2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI), Jakarta, Indonesia, 2021, pp. 418-421, doi: 10.1109/ICCSAI53272.2021.9609725.  
<https://ieeexplore.ieee.org/document/9609725>
7. J. S. Hinde and G. Belrose, "Computer Pidgin Language: A new language to talk to your computer?," 2001 <https://www.hpl.hp.com/techreports/2001/HPL-2001-182.pdf> (letzter Zugriff: 15.05.2023)
8. H. Isahara, "Resource-based Natural Language Processing," 2007 International Conference on Natural Language Processing and Knowledge Engineering, 2007,

- pp. 11-12, doi: 10.1109/NLPKE.2007.4368002.  
<https://ieeexplore.ieee.org/document/4368002>
9. Y. LeCun, Y. Bengio and G. Hinton, "Deep Learning," *Nature*, 2015, vol. 521, no. 7553, pp. 436-444, doi: 10.1038/nature14539.  
<https://www.nature.com/articles/nature14539>
10. A. S. Lokman, M. A. Ameedeen, (2019). "Modern Chatbot Systems: A Technical Review". In: K. Arai, R. Bhatia, S. Kapoor (eds) *Proceedings of the Future Technologies Conference (FTC) 2018*. FTC 2018. Advances in Intelligent Systems and Computing, vol 881. Springer, Cham. [https://doi.org/10.1007/978-3-030-02683-7\\_75](https://doi.org/10.1007/978-3-030-02683-7_75)
11. Y. Mori and Y. Miyake, "Ethical Issues in Automatic Dialogue Generation for Non-Player Characters in Digital Games," 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 5132-5139, doi: 10.1109/BigData55660.2022.10020271.  
<https://ieeexplore.ieee.org/document/10020271>
12. U. Poznanski: Erebos (2011) <https://ursula-poznanski.de/produkt/erebos/> (letzter Zugriff: 25.03.2023)
13. T. Rudra, D. Tien and T. Bossomaier, "Spoken communication with computer game characters," Third International Conference on Information Technology and Applications (ICITA'05), Sydney, NSW, Australia, 2005, pp. 494-497 vol.1, doi: 10.1109/ICITA.2005.259. <https://ieeexplore.ieee.org/document/1488854>
14. T. Rudra, D. Tien and M. Kavakli, "Analysis of Conversational Game Pidgin Language," TENCON 2005 - 2005 IEEE Region 10 Conference, Melbourne, VIC, Australia, 2005, pp. 1-4, doi: 10.1109/TENCON.2005.300857.  
<https://ieeexplore.ieee.org/document/4084871>
15. A. M. Turing, "I.—COMPUTING MACHINERY AND INTELLIGENCE", *Mind*, Volume LIX, Issue 236, October 1950, pp. 433–460, <https://doi.org/10.1093/mind/LIX.236.433>
16. M. Virkar, V. Honmane and S. U. Rao, "Humanizing the Chatbot with Semantics based Natural Language Generation," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 891-894, doi: 10.1109/ICCS45141.2019.9065723.

<https://ieeexplore.ieee.org/document/9065723>

17. J. Weizenbaum, "ELIZA-a computer program for the study of natural language communication between man and machine," Communications of the ACM, 1966, vol. 9, no. 1, pp. 36-45, doi: 10.1148/365153.365168  
<https://dl.acm.org/doi/10.1145/365153.365168>

18. Von MultiLayerNeuralNetwork\_english.png: Chrislbderivative work: — HELLKNOWZ | TALK | enWP TALK MultiLayerNeuralNetwork\_english.png, CC BY-SA 3.0,  
<https://commons.wikimedia.org/w/index.php?curid=11397827>

## 14. Anhang

### 14.1 Analoger Anhang

Anhang A: Fragebogen für Testspieler Seite 1

**Fragebogen für Testspieler**

Datum:

Name:

Alter:

Das Spiel wurde:

Mal gespielt  
 Bestes Ergebnis

Durchschnittliche Spieldauer:

**Die Spielmechanik**

Ist einfach	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Ist kompliziert
-------------	---	-----------------

**Das Spiel**

Macht Spaß	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Ist Langweilig
Ist zu kurz	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Ist zu lang

Sorgt für viel Kommunikation	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Wenig Kommunikation
------------------------------	---	---------------------

Regt zum nachdenken an	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Regt nicht zum nachdenken an
------------------------	---	------------------------------

**Die Spieldynamik**

Ist eher strategisch	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Ist eher glücksabhängig
Ist unterhaltsam	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Ist eintönig

**Das Thema**

Ist interessant	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Ist uninteressant
-----------------	---	-------------------

**Die Gestaltung**

Gefällt mir gut	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Gefällt mir nicht
-----------------	---	-------------------

**Die Kommunikation**

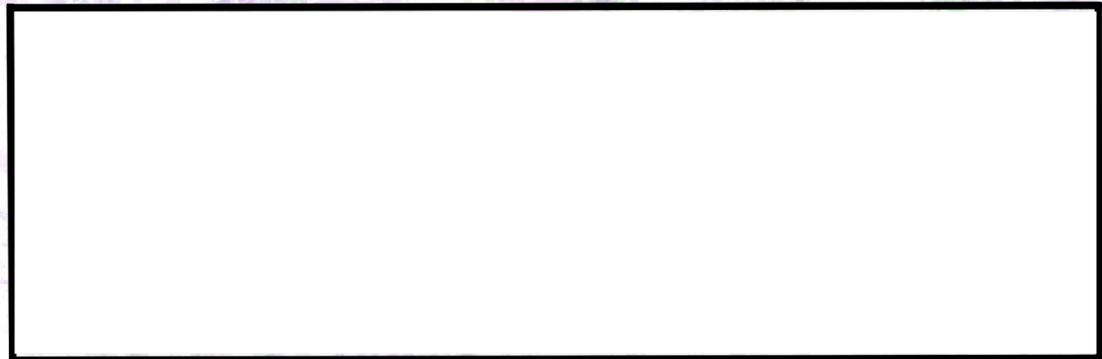
Hat gut funktioniert	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Hat nicht gut funktioniert
War passend	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	War nicht passend
Wirkte realistisch	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Wirkte unrealistisch
Ist immersiv	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Ist nicht immersiv

Gefällt mir Spragesteuert besser	<input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/>	Gefällt mit traditionellgesteuert besser
----------------------------------	---	--

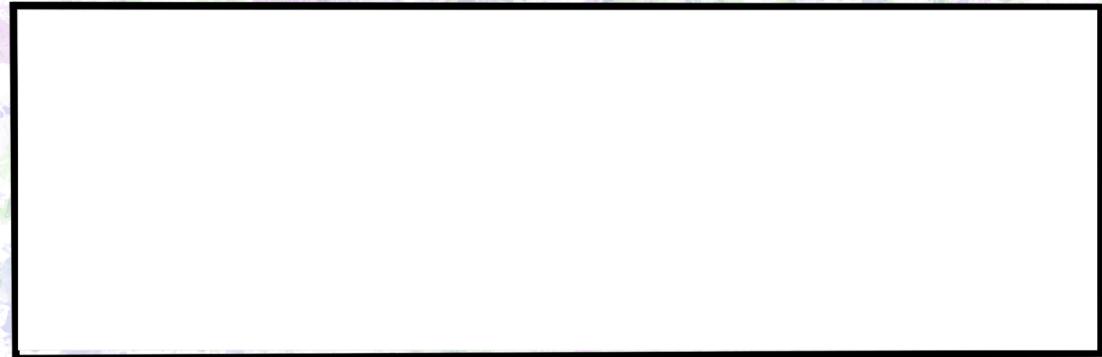
Anhang B: Fragebogen für Testspieler Seite 2

## **Fragebogen für Testspieler**

**Wie findest du es, Sprache zur  
Kommunikation in Computerspielen zu  
verwenden?**

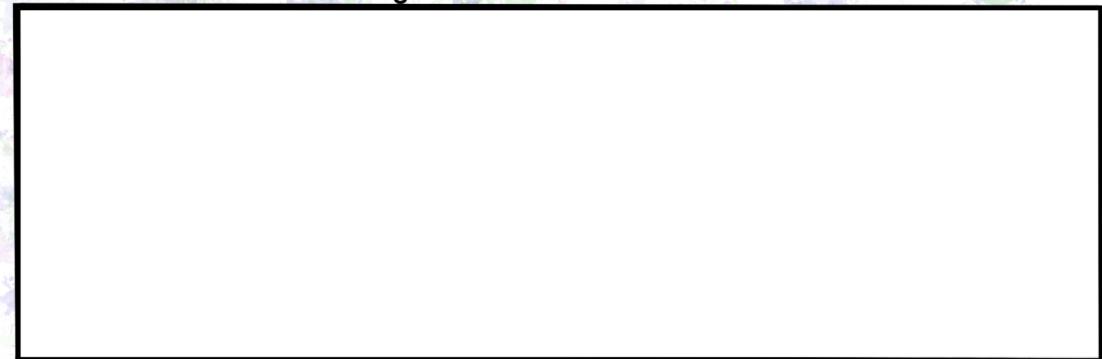


**Was würdest du anders machen?**



**Was fandest du gut bzw. schlecht?**

**Begründe deine Antwort bitte.**



## ***14.2 Digitaler Anhang***

Anhang C: Digitale Version der vorliegenden Arbeit

Anhang D: Spiel Projekt Ordner

Anhang E: Auflistung der Schlüsselwörter

Anhang F: Ausgefüllte Fragebögen der acht Testspieler

Anhang G: Videoaufnahmen eines Spieldurchlaufs