

ENTWICKLUNG EINES ADAPTIVEN ASSISTENTEN AUF BASIS VON UNÜBERWACHTEN MASCHINELLEN LERNENS MIT XAPI-NUTZUNGSINTERAKTIONSDATEN IN SERIOUS GAMES

GERALDINE SUTTER (6021014)

JADE HOCHSCHULE
Wilhelmshaven Oldenburg Elsfleth

 **Fraunhofer**
IOSB

Bachelorarbeit

30.04.2021

Erstprüfer

Prof. Dr.- Ing. Frank Wallhoff
Jade HS

Zweitprüfer/ Betreuer

Dipl.- Inf. Alexander Streicher
Fraunhofer IOSB

Betreuer

Dipl.- Inf. (FH) Daniel Atorf, M.Sc.
Fraunhofer IOSB

TGM

Technik und
Gesundheit für
Menschen

Jade Hochschule Wilhelmshaven/Oldenburg/Elsfleth
Fachbereich Bauwesen Geoinformation Gesundheitstechnologie
Abteilung Technik und Gesundheit für Menschen
Zeughausstraße 73a
D-26121 Oldenburg

Entwicklung eines adaptiven Assistenten auf Basis von unüberwachten maschinellen Lernens mit xAPI-Nutzungsinteraktionsdaten in Serious Games



Geraldine Sutter (6021014)

Bachelorarbeit

30.04.2021

Erstprüfer

Prof. Dr.- Ing. Frank Wallhoff
Jade HS

Zweitprüfer/ Betreuer

Dipl.- Inf. Alexander Streicher
Fraunhofer IOSB

Betreuer

Dipl.- Inf. (FH) Daniel Atorf, M.Sc.
Fraunhofer IOSB



Technik und
Gesundheit für
Menschen

Jade Hochschule Wilhelmshaven/Oldenburg/Elsfleth

Fachbereich Bauwesen Geoinformation Gesundheitstechnologie

Abteilung Technik und Gesundheit für Menschen

Zeughausstraße 73a

D-26121 Oldenburg

Ausgabe des Themas: 18.03.2021

Abgabe der Arbeit: 30.04.2021

Tag des Kolloquiums: 12.05.2021

Erstprüfer: Prof. Dr.- Ing. Frank Wallhoff
Jade HS

Zweitprüfer/ Betreuer: Dipl.- Inf. Alexander Streicher
Fraunhofer IOSB

Betreuer: Dipl.- Inf. (FH) Daniel Atorf, M.Sc.
Fraunhofer IOSB

ABSTRACT

English In the process of learning, everyone makes different progress and needs individual guidance accordingly. In times of digitalization, there is more and more reliance to Serious Games are set, which playfully bring the learning material closer to the user. This replaces the human assistance. It can lead to frustration, if a teacher cannot be on site to assist. Therefore, a virtual assistant would be beneficial, that adaptively adjusts to the user's learning behavior. The purpose of this work is the implementation of an intelligent virtual assistant. The existing assistant Smart LISA is to be improved, or other words, it is to be made „intelligent“. The research question is whether it is possible to create an Artificial intelligence (AI) that analyzes user behavior in a game, compares it with other users, and adaptively adjusts the assistant's actions accordingly. To implement such an assistant, an xAPI tracker was used to record the learning data of the users. The assistant communicates with a cluster server, which categorizes the user data and returns a matching cluster. For this clustering the Elkan algorithm was used. This assistant was built into an educational game and tested and evaluated in an online study. Through this study, it became apparent that users prefer passive assistance.

Deutsch In dem Prozess des Lernens macht jeder unterschiedliche Fortschritte und benötigt dementsprechend individuelle Anleitung. In Zeiten der Digitalisierung, wird vermehrt auf Serious Games gesetzt, welche dem Nutzer spielerisch den Lernstoff nahebringen. Dadurch entfällt die Notwendigkeit einer menschlichen Assistenz. Wenn ein Lehrer nicht vor Ort sein kann um zu unterstützen, kann dies zu Frustration führen. Deshalb wäre ein virtueller Assistent, der sich adaptiv an das Lernverhalten des Nutzers anpasst, vorteilhaft. Ziel dieser Arbeit ist die Umsetzung eines intelligenten virtuellen Assistenten. Hierbei soll der bereits vorhandene Assistent Smart LISA verbessert, also "intelligent" gemacht werden. Die Forschungsfrage ist, ob es möglich ist, eine künstliche Intelligenz (KI) zu erstellen, welche das Nutzerverhalten in einem Spiel analysiert, mit anderen Nutzern vergleicht und das Agieren des Assistenten darauf adaptiv anpasst. Um einen solchen Assistenten implementieren zu können wurde ein xAPI-Tracker genutzt, um die Lerndaten der Nutzer aufnehmen zu können. Der Assistent kommuniziert mit einem Cluster-Server, welcher die Nutzerdaten kategorisiert und ein passendes Cluster zurückgibt. Für dieses Clustering wurde der Elkan-Algorithmus genutzt. Dieser Assistent wurde in ein Lernspiel eingebaut und in einer Online-Studie getestet und bewertet. Durch diese Studie wurde deutlich, dass Nutzer eine passive Unterstützung bevorzugen.

Keywords - Serious Games, Lernspiel, Assistent, Maschinelles Lernen, Clustering, xAPI

Disclaimer In dieser Arbeit wird aus Gründen der besseren Lesbarkeit das generische Maskulinum verwendet. Weibliche und anderweitige Geschlechteridentitäten werden dabei ausdrücklich mitgemeint, soweit es für die Aussage erforderlich ist.

Abkürzungsverzeichnis

AI	-	Artificial intelligence
API	-	Application Programming Interface
EM	-	Erwartung, Maximierung
EVA	-	Eingabe, Verarbeitung, Ausgabe
IAS	-	Interoperabilität und Assistenzsysteme
ID	-	Identifikator
IOSB	-	Institut für Optronik, Systemtechnik und Bildauswertung
IPM	-	Ideal Path Model
IPO	-	Input, Processing, Output
KI	-	künstliche Intelligenz
LE	-	Lost Earth
LISA	-	Language Independent Ship Assistent
LRS	-	Learning Record Store
PCA	-	Principal Component Analysis
REST	-	Representational State Transfer
WiMo	-	Wissensmodelle
xAPI	-	Experience Application Programming Interface

INHALTSVERZEICHNIS

1	Einleitung	1
1.1	Motivation	1
1.2	Problemstellung	2
1.3	Lösungsansatz	2
1.4	Projektumfeld & Fraunhofer IOSB	3
1.5	Überblick	4
2	Grundlagen	5
2.1	Serious Games	5
2.2	Adaptivität	6
2.3	Lost Earth	6
2.4	Virtuelle Assistenten und LISA	8
2.5	RESTful Application Programming Interface	10
2.6	Experience API	11
2.7	Darstellung eines Nutzungspfades durch Ideal Path Model	12
2.8	Maschinelles Lernen	12
2.8.1	Clustering	14
2.8.2	K-Means	14
2.8.3	Elkan-Algorithmus	15
2.9	Verwendete Produkte und Software	17
3	Stand der Forschung und Technik	21
3.1	Maschinelle Unterstützung für nicht spezialisiertes Personal	21
3.2	Anwendungen mit xAPI	21
4	Konzept	23
4.1	Adaptiver Assistent	23
4.2	Datenkreislauf und Architektur	26
4.3	EVA-Prinzip	27
4.4	Methodik	28
5	Umsetzung	31
5.1	LISA	31
5.1.1	xAPI Tracker	32
5.1.2	Anfrage an den Server	32
5.1.3	Hilfe berechnen	32

5.1.4	Passende Hilfe anzeigen	33
5.2	Server	34
5.2.1	Anfrage an das LRS	34
5.2.2	IPM	35
5.2.3	Daten sammeln	35
5.2.4	Bewertungsalgorithmus	36
5.2.5	Bewertungsarrays	37
5.3	Clustering	38
5.4	Ergebnisse	38
6	Verifikation	41
6.1	Hypothesen/ Ziele der Studie	41
6.2	Versuchsaufbau	41
6.3	Durchführung	43
6.4	Ergebnisse	43
6.5	Diskussion	54
7	Fazit und Ausblick	57
	Literatur	59

Die Digitalisierung ist in vollem Gange und besonders in Zeiten von Covid-19 ist es fundamental Präsenzvorgänge wie das Lernen, in eine Online-Version umzuwandeln. Dadurch wird vermehrt auf Lernspiele oder Serious Games gesetzt, welche eine unabhängige und selbstständige Weiterbildung bieten. Serious Games haben in der Definition ein charakteristisches Ziel. Ein Beispiel für ein solches Spiel ist Lost Earth 2309 von Fraunhofer IOSB. Hierbei sollen Grundkenntnisse zur Luft- und Satellitenbilddauswertung vermittelt werden. Durch solche Spiele werden Lerninhalte online näher gebracht und ein präsenster Lehrer ist nicht notwendig. In dem Prozess des Lernens machen Lernende unterschiedliche Fortschritte und benötigen dementsprechend individuelle Anleitung. Dadurch kann es bei unabhängigem Lernen zu Problemen führen, falls der Lernende Schwierigkeiten mit dem Lerninhalt hat und dabei kein Lehrer vor Ort ist, um zu unterstützen. Dieses Problem soll mit einem Maschinellen-Clustering-Verfahren angegangen werden. Unter Clustering wird das Gruppieren von Daten mit gleichen Merkmalen verstanden.

1.1 Motivation

Diese Arbeit beschäftigt sich mit der adaptiven Unterstützung in Serious Games. In dem Lernspiel Lost Earth von Fraunhofer existiert ein Skript-gesteuerter Assistent, der sogenannte „Language Independent Ship Assistent“ (LISA). LISA kann nur in für sie vorgesehenen Momenten, Hinweise an den Nutzer weiterleiten. Dabei kann es dazu kommen, dass der Spieler unterbrochen wird, wenn er gerade keine Hilfe braucht, aber auch, dass LISA keine Hilfestellung gibt, wenn der Nutzer sie benötigt. Mit einem adaptiven Assistenten sollen solche Situationen vermieden und Lernerfolge optimiert werden. In dieser Arbeit wird der Widerspruch untersucht, dass der Nutzer Hilfe benötigt und möchte, aber zur gleichen Zeit nicht unterbrochen werden will. Das bedeutet der Assistent muss in den passenden Momenten helfen, in welchen der Spieler Unterstützung benötigt. Wenn ein solcher Zeitpunkt falsch eingeschätzt wird, dann kann es bei dem Nutzer zu Frustration oder Langeweile führen. Dies führt wiederum dazu, dass der Nutzer das Spiel frühzeitig beendet. Da in diesem Szenario kein Lehrer in den Vorgang eingreifen kann, soll eine KI das Nutzerverhalten in dem Spiel analysieren, mit denen von anderen Nutzern vergleichen und das Agieren des Assistenten darauf adaptiv anpassen. Also müssen Nutzer in Gruppen eingeteilt werden, die aussagen, welche Hilfenotwendigkeit diese Nutzer benötigen. Die darauf folgende Frage ist, wie kann eine KI über aufgenommene Daten kategorisieren und erkennen, dass Hilfe benötigt oder nicht benötigt wird?

1.2 Problemstellung

Das Ziel ist es, einen Assistenten zu entwickeln, der Nutzer in Lernspielen gezielt unterstützt. In der Masterthesis von Dillig [Dillig, 2020] wurde bereits der Assistent LISA konstruiert. Darin sind die Hilfestellungen in das Spiel fest integriert und kommen bei jedem Durchgang an demselben Zeitpunkt. In dieser Arbeit soll dieser Assistent „smart“ werden und zwischen Situationen differenzieren, in denen dem Nutzer schneller oder langsamer Hilfestellung zuteilwerden sollte. Um einen solchen Assistenten zu entwickeln, welcher Nutzer aufgrund ihres Verhaltens unterscheiden kann, wird maschinelles Lernen benötigt. In dieser Arbeit soll der Algorithmus selbst lernen, welches Verhalten für eine Hilfenotwendigkeit spricht. Das bedeutet, es stehen Unüberwachtes Lernen und Bestärkendes Lernen zur Auswahl. Da Bestärkendes Lernen sehr viel mehr Daten benötigt, um angelernt zu werden, ist für diese Arbeit ein Clustering-Algorithmus geeignet. In diesem Projekt soll hervorgehen, welche Clustering-Art für den Assistenten verwendet werden sollte, welcher Algorithmus für die Unterteilung der Daten und welche Metrik für die Berechnung dieser verwendet werden. Außerdem muss ein System erstellt werden, mit welchem der Assistent erkennt, wie weit der Spieler in dem Spiel fortgeschritten ist, um die nötige Orientierung zu bieten, um den Spieler auf den richtigen Weg zu führen.

Ziel ist es, die Lösungen dieser Fragen zu implementieren und zu einem Assistenten zusammenzuführen, welcher intern mit den einzelnen Aspekten kommuniziert und so den Nutzer adaptiv unterstützt. Es soll also eine „smart“ LISA entwickelt werden.

1.3 Lösungsansatz

Um das Ziel zu erreichen LISA „smart“ zu machen, muss ein Algorithmus entwickelt werden. Dieser Algorithmus wird in Spielen von Studenten und Personen aus dem Bekanntenkreis getestet. In unterschiedlichen Spielen wird ein Assistent eingebaut. Dieser Assistent lässt Objekte aufleuchten, mit welchen der Spieler interagieren muss, um weiterzukommen. Der Assistent kann über Sprechblasen, in welchen dem Nutzer hilfreiche Hinweise zu Spielmechaniken angezeigt werden, unterstützend eingreifen. Optimalerweise werden diese Hilfestellungen nur gegeben, wenn der Spieler sie benötigt. Dabei soll der Spielfluss sofern möglich nicht unterbrochen werden. Da jeder Spieler andere Fähigkeiten und Gedankengänge hat wird eine KI benötigt, welche unterschiedliche Nutzer in Bereiche einteilen kann. Konkret hier in die Bereiche „Braucht schnell Hilfe“, „Braucht demnächst Hilfe“ und „Braucht lange keine Hilfe“. Ein Machine Learning Algorithmus, welcher Daten in Bereiche einteilen kann, wäre ein Clustering-Algorithmus. Dabei sollte der Algorithmus die Daten der Spieler sammeln, unterschiedliche Persona erstellen und diese nach benötigtem Hilfegrad clustern. Zukünftige Teilnehmende werden mit den bereits ermittelten Nutzerdaten verglichen. Ein neuer Nutzer wird in ein Cluster eingeteilt und ihm werden dementsprechende Hilfeleistungen zuteil. Falls der Nutzer sich im Verlauf des Spiels anders verhält, so wird er in ein anderes Cluster eingeteilt. Ziel dieser Arbeit ist, dass LISA den Spielverlauf angenehmer gestalten soll. Das bedeutet, dass mit dem Assistenten die Serious Games für den Nutzer angenehmer durchgeführt werden können als ohne. Diese Software-Unterstützung

soll seitens des Nutzer gegenüber eine gescripteten Hilfestellung bevorzugt werden. Er soll besser funktionieren und von dem Nutzer bevorzugt werden als gescriptete Hilfestellungen.

1.4 Projektumfeld & Fraunhofer IOSB

Wie in der Abteilung Interoperabilität und Assistenzsysteme (IAS) in dem Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB) in Abbildung 1.1 zu sehen, werden Methoden zur Unterstützung bei der Interaktion mit komplexen Informationssystemen entwickelt. Dabei werden digitale Lernspiele konstruiert, welche sich adaptiv an die Nutzer anpassen. Die Gruppe Wissensmodelle (WiMo) in der Abteilung IAS entwickelt hierbei Modelle und Methoden für Assistenzsysteme mit dem Schwerpunkt, Konzepte in der Bildauswertung zu vermitteln.



Abbildung 1.1: Fraunhofer IOSB Standort Karlsruhe [Fraunhofer IOSB, 2021b]

1.5 Überblick

In dieser Arbeit werden zunächst die Grundlagen und der aktuelle Stand der Forschung erläutert. Dazu gehören die verwendeten Produkte, Modelle und Begriffe, die für das Verständnis nötig sind. Anschließend wird das Konzept der Arbeit und das nähere Vorgehen beschrieben. Dabei wird geklärt, was zu dem Assistenten gehört, damit er das gewünschte Ziel erreichen kann. Daraufhin wird die Umsetzung des Projekts dokumentiert. Wie wurden die Voraussetzungen erfüllt und wie geschieht die Berechnung und Kommunikation? Darauf folgt die Schilderung der durchgeführten Verifikation in Form einer Studie. Dabei wurde der Assistent in einem Spiel getestet und daraufhin bewertet. Mithilfe der Ergebnisse des Projekts und der Studie, wird eine Diskussion eingeleitet. Als Abschluss wird das Fazit, sowie der Ausblick dargelegt.

In diesem Kapitel werden die Grundlagen behandelt, die der Arbeit der folgenden Kapitel zugrunde liegen. Dabei werden die Komponenten die von Fraunhofer für diese Arbeit gestellt wurden, das Konzept von unterschiedlichen Modellen, KI und verwendete Produkte sowie Software zusammengefasst und beschrieben.

2.1 Serious Games

Um einen Assistenten für ein Lernspiel zu entwickeln, muss erst geklärt werden, auf was es in einem solchen ankommt. Wenn von einem Serious Game die Sprache ist, dann wird normalerweise von einem Spiel für Training, Simulationen oder der Weiterbildung gesprochen. Es geht darum den Spieler mit Hilfe von einem Computerspiel zu motivieren neues Wissen oder Fähigkeiten zu erlernen. Der Unterschied zwischen einem Serious Game und einem Unterhaltungsspiel, ist das angestrebte Ziel. Bei einem Unterhaltungsspiel soll der Spieler Spaß haben. [Susi, Johannesson und Backlund, 2007]

Das charakteristische Ziel eines Serious Games ist abhängig seines Anwendungsbereichs. Beispiele für solche Ziele wären, Lern- oder Trainingseffekte bei Bildungsspielen oder Änderungen des Vitalzustands durch ein Gesundheitsspiel. [DIN SPEC 91380:2018-06, 2018]

Die Theorie von Digital Game Based Learning besagt, dass die neuen Generationen, die momentan in der Ausbildung sind oder selbst Ausbilden mit Videospiele und der Digitalisierung aufgewachsen sind. Dadurch werden auch die Lernmethoden und Übungen dementsprechend angepasst. Das Lernen soll dadurch unterhaltsam und fesselnd werden, wodurch es um einiges effektiver wird. [Prensky, 2003]

Um ein immersives Serious Game zu entwickeln, das bedeutet dass eine virtuelle Umgebung als real wahrgenommen wird, soll dieses wie ein Unterhaltungsspiel wirken, welches indirekt Lernziele vermittelt. Um dies zu erreichen, muss die Zielgruppe des Serious Games bekannt sein. Somit kann das Spiel an die Vorlieben dieser Zielgruppe angepasst werden. Durch die Eingrenzung der Zielgruppe, können ein passendes Spiele-Genre sowie Spielmechaniken für das Serious Game gewählt werden. Dabei sollten die Lernziele des Serious Games durch reale Aufgaben erreicht werden, ohne das immersive Spielgefühl zu verlieren. Dafür können die angestrebten Lernziele mit den Spielzielen verbunden werden. [Atorf, Kannegieser und Roller, 2018]

2.2 Adaptivität

Der zu entwickelnde Assistent soll sich den Interaktionen der Nutzer anpassen, dies wird auch als Adaptivität bezeichnet. Adaptivität wird in Serious Games genutzt um den Nutzer auf lange Sicht motiviert zu halten. Durch die Nutzung von Adaptivität sollen im Fall von Serious Games, Spielinhalte automatisch und individuell angepasst werden. Dabei ist das Ziel das System an den Nutzer anzupassen. Damit kann Adaptivität durch einfache, schwellenbasierte Verfahren oder komplexe, sich entwickelnde Anwendermodellen realisiert werden. [Streicher und Smeddinck, 2016]

Durch einen automatisch anpassbaren Schwierigkeitsgrad sollen alle Nutzer ihr erlerntes Wissen festigen oder steigern. Durch eine hohe Leistung eines Spielers soll die Schwierigkeit der darauf folgenden Spielabschnitte schwieriger werden um ihn weiterhin zu fordern. Bei einem Spieler mit niedriger Leistung sollten die folgenden Abschnitte das bereits Gelernte vereinfacht wiederholen. [Hammel, Schade und Adelsberg, 2019]

Adaptivität sorgt dafür, dass sich eine Software an die Fähigkeiten des Nutzers anpasst. Hierbei ist der Zeitpunkt, wann sich die Software anpassen soll entscheidend. Dies kann über eine Messung des Nutzerfortschritts ermittelt werden. Bewegt sich der Nutzer in Richtung des Ziels oder entfernt sich von diesem, passt sich die Software entsprechend adaptiv an diesen Fortschritt an. Eine Möglichkeit den Nutzerfortschritt zu bestimmen stellt das Ideal Path Model dar. [Streicher, Leidig und Roller, 2018]

2.3 Lost Earth



Abbildung 2.1: In dieser Szene kann zwischen Räumen gewechselt werden. [Fraunhofer IOSB, 2021a]

In dieser soll ein Assistent für ein Lernspiel implementiert werden. Dafür wurde das folgende Spiel beschrieben. Lost Earth 2307 baut auf Digital Game Based Learning auf. Das heißt es vertritt die Ziele aus einem Serious Game als auch einem Unterhaltungsspiel. In dem Spiel sollen Lerninhalte vermittelt werden und dabei unterhalten. Das Spiel ist ein Lernwerkzeug, welches möglichst viele Aufgaben in dem Bereich zur Auswertung von Luft- und Satellitenbildern vermitteln soll. Dabei werden spielerisch optische, Infrarot- und Radarbilder ausgewertet. [Fraunhofer IOSB, 2020]

Das verwendete Lost Earth ist mit Unity entwickelt worden. Das Spiel ist in einzelne Szenen unterteilt in welchen sich Objekte und virtuelle Personen befinden, mit welchen der Nutzer interagieren kann. [Fraunhofer IOSB, 2021a] In der Abbildung 2.1 ist die Szene zu sehen, in welcher der Szenenwechsel bzw. die Raumauswahl stattfindet.

Die grundlegende Aufgabe in dem Spiel ist es Sensoren zu Planeten zu schicken und die Umgebungsbedingungen, wie zum Beispiel das Wetter zu beachten. Wenn der Sensor zurück kommt und erfolgreich Bilder aufnehmen konnte, werden diese in einer abschließenden Aufgabe bearbeitet oder untersucht. In der Abbildung 2.2 ist die geöffnete Konsole zu sehen über die der Sensor gestartet werden kann. Unter dem Abschnitt „Start Delay“, in diesem Bild mit dem Leuchten hervorgehoben, kann der Zeitpunkt verändert werden, an dem der Sensor startet. So kann schlechtes Wetter und somit eine fehlerhafte Bildaufnahme vermieden werden. Lost Earth 2309 ist eine erneuerte Version, welche web-fähig ist und in den abschließenden Aufgaben auf weitere Software verzichtet. [Fraunhofer IOSB, 2021a]



Abbildung 2.2: In diesem Fenster kann eingestellt werden wann der Sensor losgeschickt wird. [Fraunhofer IOSB, 2021a]

2.4 Virtuelle Assistenten und LISA

Es soll ein Assistent für Lernspiele entwickelt werden. Dafür ist es nützlich die Grundprinzipien eines solchen Agenten zu klären.

Virtuelle Assistenten Ein virtueller Assistent soll dem Nutzer bestmöglich bei der Arbeit unterstützen. Dafür sollte dieser Assistent in der Lage sein, Entscheidungen zu treffen und dementsprechend zu handeln. Die Aufgabe des Assistenten ist es, ohne Eingreifen einer weiteren Person dem Nutzer die Aufgaben einfacher zu gestalten. [Wallhoff, Vox und Theuerkauff, 2019] Eine Schwierigkeit in Serious Games ist es, den Nutzern das Spielprinzip zu Beginn zu erklären. Erst wenn Nutzer den Spielablauf verstehen, können sie völlig in dieses eintauchen. Unterhaltungsspiele setzen hierbei auf Tutorials. Diese sollen dem Spieler alle nötigen Fähigkeit zum erfolgreichen Spielen beibringen. Dabei zeichnen sich besonders die Tutorials ab, die sich nicht wie ein solches anfühlen. Hierbei kann ein virtueller Assistent helfen einem Nutzer Spielmechaniken beizubringen, ohne dass diese wie ein Tutorial wirken. Hierbei kann der Assistent über Dialoge Tipps geben oder Funktionen beim ersten Interagieren beschreiben. [Atorf, Kannegieser und Roller, 2019]

LISA LISA ist ein Assistent in dem Spiel Lost Earth. Sie ist ein schwebender Roboter, welcher den Spieler während den Missionen begleitet und unterstützt, falls dieser Hilfe benötigt. Dies macht sie mit Hinweisen in Form von Message-Boxen oder Sprechblasen und indem sie Objekte aufleuchten lässt, mit welchen als nächstes interagiert werden muss. Außerdem weist LISA darauf hin, wenn der Raum verlassen werden muss um die Mission fortführen zu können. [Fraunhofer IOSB, 2021a]

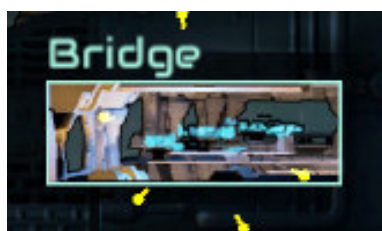


Abbildung 2.3: Das Aufleuchten eines Raums, der als nächstes besucht werden muss. [Fraunhofer IOSB, 2021a]



Abbildung 2.4: Um in der Mission weiter zu kommen, müssen Räume auch wieder verlassen werden. [Fraunhofer IOSB, 2021a]

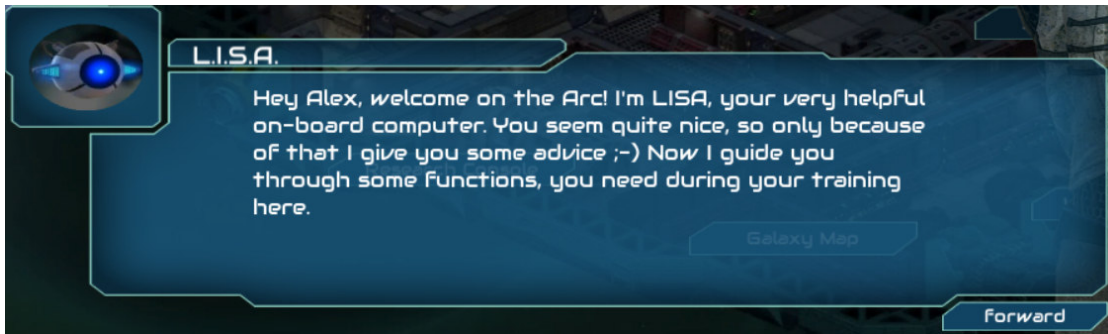


Abbildung 2.5: Für Mitteilungen, die nicht mit einfachen Zeichen dargestellt werden können, werden Message-Boxen genutzt. [Fraunhofer IOSB, 2021a]

Die Sprechblasen, wie in Abbildung 2.6 sind passiv aufgebaut. Da sie nur klein über LISA angezeigt werden und nicht weggeklickt werden müssen, unterbrechen sie nicht das Spiel. Die Message-Boxen wie in Abbildung 2.5, müssen dagegen über einen Knopf akzeptiert werden und nehmen einen großen Teil des Bildschirms ein. Das Leuchten ist in Abbildung 2.3 zu erkennen und gehört zu den passiven Hilfestellungen. Ähnlich ist dabei das Zeichen für „Raum muss verlassen werden“, in Abbildung 2.4 abgebildet. Es lässt das Zeichen in der unteren Mitte des Bildschirms anzeigen ohne dass dabei etwas aktiv angeklickt werden muss. [Danz, 2019]

In der Bachelor-Arbeit von Lobstedt [Lobstedt, 2020] wurde LISA und ihre Hilfestellungen weiterentwickelt. Mit einem separaten Framework von Fraunhofer wurde ein Assistenzlevel des Nutzers bestimmt. Durch dieses Assistenzlevel und einem Entscheidungsbaum sollten die gelieferten Hilfestellung angepasst werden. [Lobstedt, 2020]

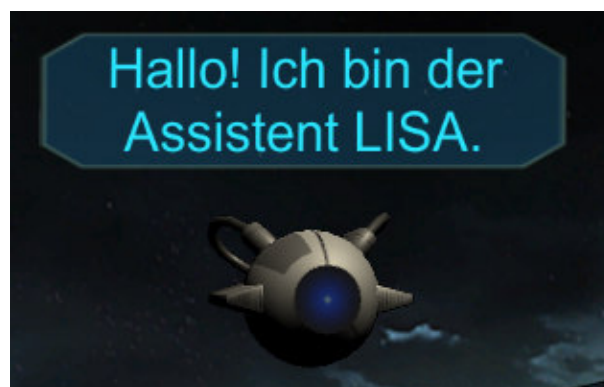


Abbildung 2.6: Passive Message-Box in Lost Earth, welche nicht weggeklickt werden muss [Fraunhofer IOSB, 2021a]

2.5 RESTful Application Programming Interface

In dieser Arbeit wird eine Programmierschnittstelle genutzt. Eine solche Schnittstelle wird als Application Programming Interface (API) bezeichnet. Sie verknüpft zwei unabhängige Anwendungen. Mit Hilfe einer API lässt sich bestimmen, welche Funktionen einer Anwendung angesprochen werden können und dürfen. Durch dieses gezielte Gewährleisten von Funktionalitäten, lässt sich über Authentifizierungen die Sicherheit einer API gewährleisten. Eine API enthält eine Anfrage (Request) und verarbeitet diese. Anschließend wird eine Antwort mit Ergebnissen der Verarbeitung zurückgeschickt. Die verschiedenen Arten der APIs beschreiben die Zugriffsvoraussetzungen sowie die Sicherheit einer API. [Sim, 2019]

open API Öffentliche APIs ohne Zugriffsbeschränkungen

public API Öffentliche APIs mit erforderlicher Authentifizierung

private API Organisationsinterne APIs

internal API APIs für nur ein bestimmtes System

[Sim, 2019]

RESTAPI gehört nicht zu den herkömmlichen Typen und Arten. Es beschreibt die Architektur von API's in Webanwendungen. [Dazer, 2012]

Representational State Transfer Application Programming Interface REST -oder auch RESTful API steht für Representational State Transfer Application Programming Interface. Dabei wird mit REST ein Architektur-Stil für Web API's beschrieben. Der Schwerpunkt von REST liegt auf Skalierbarkeit, Erweiterbarkeit und Interoperabilität. Das Ziel von REST ist eine plattformunabhängige Schnittstelle. In den folgenden Abschnitten sind einige wichtige Kernkonzepte von REST aufgeführt. [Dazer, 2012]

Client und Server müssen voneinander unabhängig sein.

Jede Funktion muss in sich selbst beendet sein. Das heißt das eine Funktion nicht Parameter, wie in einer Klasse als Klassenvariable, für die anderen Funktionen/Methoden speichern darf. Dies wird Zustandslosigkeit genannt.

Caching ermöglicht den Datenfluss zwischen Server und Client zu vermindern. Caching verhindert, dass der Server unnötig vom Client angefragt wird. Unter Caching wird das lokale

abspeichern von Serverdaten beim Client verstanden. Statt einer Anfrage an den Server werden einfach die Daten im Cach abgerufen, da der Server die gleichen Daten liefern würde. Vom Server selbst wird definiert, welche Funktionen Cach-bar sind.

Die einzelnen Ressourcen auf dem Server müssen eindeutig adressierbar sein. Dies wird mittels URI's umgesetzt. Dabei steht URI für Uniform Ressource Identifier und fasst URL's (Uniform Ressource Locator) und URN (Uniform Ressource Name) zusammen. Jede Funktion auf dem Server hat eine eindeutige URI.

Die Ressourcen werden von dem Server mittels verschiedener Dateitypen an den Client geschickt. Typische Dateitypen sind JSON und XML. Über die Header - Datei kann der Client dem Server eine Information über die präsentierbaren Dateitypen mitteilen. Die Header - Datei umfasst weitere Informationen für den Client oder Server, die nichts mit der eigentlichen Anfrage zu tun haben, wie zum Beispiel Authentifizierungen.

Die Funktionen müssen Mittels Standard HTTP Methoden angefragt werden. Darunter fallen **GET, PUT, POST, DELETE, ...** Beispielsweise sollte eine **GET** Funktion nur Daten Abfragen und über **POST** sollten nur Daten hinzugefügt werden können.

[Dazer, 2012]

Die zuvor genannten Punkte sind nicht zwingend einzuhalten. Dennoch gilt, die Anwendung erst als RESTful, falls alle Konzepte eingehalten wurden. Vorteile dieser Anwendung sind, dass sie einfach erweiterbar ist, da dies durch die Zustandslosigkeit und die Nutzung von HTTP-Methoden ermöglicht wird. Durch die Zustandslosigkeit lassen sich neue Funktion einbinden, ohne die Vorigen zu beeinträchtigen. Durch die HTTP-Methoden sind die Funktionen klar in ihrer Aufgabe definiert. [Dazer, 2012]

2.6 Experience API

Um in dieser Arbeit Daten aus dem Spiel nutzen zu können, werden diese in einem einheitlich Format abgespeichert, um die ausgeführten Interaktionen nachzuvollziehen. Experience API (Experience Application Programming Interface oder xAPI) wird weitreichend verwendet, um Lernendaten eines Nutzers aufzunehmen und abzuspeichern. Durch diese Lerndaten lassen sich Nutzerinteraktion und Lernfortschritte nachvollziehen. Dabei werden die einzelnen Interaktionen als einheitliche Statements zusammengefasst. Ein Statement wird als Nomen, Verb, Objekt zusammengefasst und abgespeichert. Diese Statements werden auf einem Learning Record Store (LRS) abgespeichert. Das LRS wird als Hauptspeicher für die statements genutzt. [Xiao u. a., 2020]

2.7 Darstellung eines Nutzungspfades durch Ideal Path Model

Damit eine Orientierung implementiert wird, werden Daten benötigt, die den Spielablauf beschreiben. Das Ideal Path Model (IPM) beschreibt die perfekte Durchführung einer virtuellen Umgebung. Bei einem Spiel wäre das zum Beispiel die Komplettlösung. Wie spielt der Spieler am schnellsten und effektivsten das Spiel durch. Bei einem IPM werden alle nötigen Schritte festgehalten, die nötig sind um in einer solchen Umgebung voran zu kommen. Dieses Modell kann mit einem xAPI-Tracker aufgenommen werden. [Streicher, Leidig und Roller, 2018]

2.8 Maschinelles Lernen

In dieser Arbeit wird ein Clustering-Algorithmus verwendet. Diese Arten von Algorithmen lassen sich unter dem Punkt Maschinelles Lernen zusammenfassen. Das Thema Maschinelles Lernen ist ein Teilgebiet von KI (künstlich Intelligenz). Es gibt vier Kategorien, in welchen künstliche Intelligenz beschrieben werden kann.

- **menschliches Handeln**

Menschliches Handeln wird mit dem *Turing Test* überprüft. Bei diesem Test schreibt ein Mensch mit einem Menschen und mit einer KI. Falls die Testperson nicht unterscheiden kann ob die Antworten von einem Menschen oder einer KI kam, hat die KI den Test bestanden und gilt als intelligent. Dabei muss die KI über folgende programmierte Eigenschaften verfügen: Natural Language Processing(NLP), Wissensrepräsentation, Logisches Schlussfolgern und Machine Learning.

- **menschliches Denken**

Eine KI „denkt“ wie ein Mensch, wenn das Input -und Outputverhalten dem eines Menschen ähnelt.

- **rationales Denken**

Sobald auf eine Logik zurückgegriffen werden kann und dadurch Schlüsse gezogen werden, kann es als rationales Denken bezeichnet werden.

- **rationales Handeln**

Es wird rational gehandelt, wenn das Handeln zu einem bestmögliche Ergebnis zu führt oder dieses zu erwarten ist. Um so agieren zu können, benötigt es rationales Denken. Aber nicht immer gibt es durch Rationalität eine eindeutige Lösung. Das bedeutet zu rationales Handeln gehört auch, eine Entscheidung zu fällen und dementsprechend zu agieren, auch wenn es nicht klar ist, ob das beste Resultat eintritt.

[Russell und Norvig, 2010]

Das Ziel von maschinellem Lernen ist das Entwerfen von Algorithmen, welche sich weiterentwickeln und an Veränderungen anpassen. Um dies erreichen zu können, werden Daten und Training genutzt. [Kapitanova und Son, 2012]

In der folgenden Auflistung werden die drei Hauptarten von maschinellem Lernen vorgestellt.

supervised learning

Supervised Learning oder überwachtes Lernen, hat Daten zur Verfügung, welche Eingabe -und die passenden Ausgabewerte beinhalten. Dadurch soll der Computer weitere Daten gleich einordnen können, wie in den zuvor bereitgestellten Daten. Hauptaufgaben eines supervised learning Algorithmus ist die Klassifikation und die Regression. [Russell und Norvig, 2010]

- **Klassifikation**

Bei der Klassifikation handelt es sich um eine feste Anzahl an Ausgaben, die bestimmt wird. Diese kann zum Beispiel das Wetter sein. Sollte es bei der Klassifikation nur zwei Ausgaben geben, so wird diese Bool- oder Binär - Klassifikation genannt.

- **Regression**

Bei der Regression handelt es sich um einen Zahlenwert, der bestimmt wird. Dies kann zum Beispiel die Temperatur am nächsten Tag sein.

[Russell und Norvig, 2010]

unsupervised learning

Unsupervised Learning oder unüberwachtes Lernen, hat Daten zur Verfügung, welche nur Eingabewerte beinhalten. Der Computer soll dabei eine Struktur in diesen Daten finden, um diese in Gruppen aufteilen zu können. Dabei müssen den Daten mehrere Merkmale zugeordnet werden. Dadurch kann der Computer die Daten unterscheiden und gegebenenfalls richtig einteilen. Je nach Merkmal können ganz unterschiedliche Gruppenarten entstehen. Denn das Ziel ist es ähnliche Dinge zusammenzufassen. Merkmale ausschlaggebend. [Russell und Norvig, 2010]

- **Clustering**

Clustering wird in dieser Arbeit verwendet. Dieses wird näher im Abschnitt 2.7 erläutert.

reinforcement learning

Reinforcement Learning oder bestärkendes Lernen, wird häufig ein Agent angelehrt etwas so zu tun wie gewollt, indem er mit Belohnungen und Bestrafungen auf den richtigen Weg geleitet wird. Dabei werden Aktionen bevorzugt, die den optimalsten Nutzen mit sich bringen.

[Frochte, 2019]

- **Klassifikation**

Ein Agent ist hierbei eine Software die in der Lage ist, ohne das Eingreifen eines Menschen eigenständig zu handeln. Hierbei lassen sich diese Agenten in 5 Klassen einteilen: Einfache reaktive Agenten, Beobachtende Agenten, Zielbasierte Agenten, Nutzenbasierte Agenten und Lernende Agenten.

[Frochte, 2019]

2.8.1 Clustering

Das Clustering beschäftigt sich mit dem Thema, einzelne Daten eines Datensatzes in Gruppen einzuordnen. Diese Art des maschinellen Lernens befindet sich im Gebiet des unüberwachten Lernen, da die untersuchten Daten zu Beginn nicht einzuordnen(gelabelt) sind. Erst durch das Clustering selbst können den Daten Bezeichnungen gegeben werden, je nachdem was ihre Cluster aussagen sollen. [Russell und Norvig, 2010]

Diese Cluster von Gruppen werden durch Eigenschaften der einzelnen Daten bestimmt. Dabei werden die Daten welche ähnliche oder unähnliche Ausprägungen der Eigenschaften aufweisen zu einem Cluster zusammengesetzt. [Kuntze, 2007]

Die existierenden Cluster Algorithmen lassen sich in partitionierende und hierarchische Verfahren unterteilen. Bei den partitionierenden Verfahren ist zu Beginn die Anzahl der Cluster festgesetzt. Die Daten werden daraufhin auf genau diese Anzahl an Clustern gruppiert. Hierbei muss jedes Cluster mindestens ein Element aus dem Datensatz enthalten und jedes Element darf nur einem Cluster zugeordnet sein. Ein Beispiel wäre hierbei der k-Means Algorithmus. Bei dem hierarchischen Verfahren wird eine Clusterstruktur erstellt. Hierbei lässt sich verfolgen, zu welchen weiteren Clustern, einzelne Cluster zusammengefasst werden. [Böhm u. a., 2004]

2.8.2 K-Means

In dieser Arbeit wird für das Einordnen von Nutzerdaten ein K-Means-Algorithmus verwendet. K-Means ist ein Clustering Algorithmus, mit dessen Hilfe ein Datensatz mit einer Anzahl von N Werten in eine zuvor definierte Anzahl K Cluster gruppiert werden kann. Dabei werden die einzelnen Daten x^n (für $1 \leq n \leq N$) des Datensatzes D in das Cluster gruppiert, zu dessen Cluster Zentrum μ_k (für $k = 1, \dots, K$) sie den geringsten Abstand haben. Die einzelnen Daten x^n dürfen jeweils nur in einem Cluster vorkommen. Dies wird über die boolesche Variable r_{nk} ($\in \{1,0\}$) für $n = 1, \dots, N$ und $k = 1, \dots, K$) ausgesagt. Sollte ein Wert x_n in einem Cluster k für $r_{nk} = 1$ sein, also in dem Cluster enthalten sein, so muss er in einem Cluster j für $r_{nj} = 0$ sein. Dabei gilt $k \neq j$ und stellt somit die Exklusivität zu einem Cluster sicher. Die Aufgabe des K-Means Algorithmus ist nun, die einzelnen Daten x^n den nächsten Clusterzentren μ_k zuzuordnen. Dies wird über die euklidische Distanz in Formel 2.1 berechnet. Der Wert x_n wird dem Cluster zugeordnet, zu dessen

Clusterzentrum μ_k er den geringsten euklidischen Abstand hat. [Bishop, 2006]

$$\|x_n - \mu_k\| = \sqrt{(x_n - \mu_k)^2} \quad (2.1)$$

Um nun die korrekte Cluster Zuweisung über r_{nk} und die korrekten Clusterzentren μ_k zu erhalten, wird die Summe des quadratischen Fehlers J berechnet. Diese ist in Formel 2.2 zu sehen. [Bishop, 2006]

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2 \quad (2.2)$$

Je niedriger der Wert J umso genauer sind die Daten den passenden Clustern zugewiesen. Um r_{nk} und μ_k anzupassen, wird jeweils nacheinander zuerst nur der Wert für r_{nk} und dann der Wert für μ_k angepasst. Dies wird so lange durchgeführt, bis die Summe des quadratischen Fehlers J gegen einen Wert konvergiert. Dann ist die finale Verteilung der Daten auf die Cluster abgeschlossen. Zu Beginn des Clusterings werden die Clusterzentren μ_k zufällig gewählt. Im Anschluss wird der Wert für die Cluster Zuordnung r_{nk} mit Hilfe der Formel 2.3 optimiert. Hier muss jeweils nur für jeden Wert x_n bestimmt werden, welchem Clustercenter er am nächsten ist. [Bishop, 2006]

$$r_{nk} = \begin{cases} 1 & \text{für } k = \arg \min_j \|x_n - \mu_j\|^2 \\ 0 & \text{sonst} \end{cases} \quad (2.3)$$

Danach wird der Wert für die Clusterzentren μ_k mit Hilfe von Formel 2.4 optimiert. Hierbei muss lediglich der Mittelwert aller Daten in einem Cluster bestimmt werden. Durch diese Mittelwerts - Berechnung der Clusterzentren heißt der Algorithmus K-Means. [Bishop, 2006]

$$\mu_k = \frac{\sum_n r_{nk} \cdot x_n}{\sum_n r_{nk}} \quad (2.4)$$

Diese Verbesserung der Werte r_{nk} und μ_k im Wechsel entsprechen genau den **E (Erwartung)**- und **M (Maximierungs)**-Schritten des **EM-Algorithmus**. Dieser steht für Erwartungs-Maximierungs-Algorithmus und beschreibt ein Verfahren, um die Lösung mit der höchsten Wahrscheinlichkeit zu finden (maximum likelihood solution). [Bishop, 2006]

2.8.3 Elkan-Algorithmus

In dieser Arbeit wurde der Elkan-Algorithmus genutzt, da er die Berechnung der Daten aus dieser Arbeit schneller berechnet, als der herkömmliche EM-Algorithmus.

Der Ansatz ist hierbei, mit Hilfe der Dreiecksungleichung unnötige Distanzberechnungen zu vermeiden. Die Dreiecksungleichung sagt aus, dass in einem Dreieck die Summe der Längen zweier Seiten, größer oder gleich der Länge der verbleibenden Seite sein muss. Diese Dreiecksungleichung ist in Formel 2.5 zu sehen, wobei \mathbf{x} , \mathbf{y} , \mathbf{z} die Punkte in einem Dreieck und $\mathbf{d}()$ die Distanz zwischen diesen angibt. [Elkan, 2003]

$$d(x, z) \leq d(x, y) + d(y, z) \quad (2.5)$$

Dieses Dreieck ist in Abbildung 2.7 dargestellt. Sollte die Summe $d(\mathbf{x},\mathbf{y})+d(\mathbf{y},\mathbf{z})$ gleich $d(\mathbf{x},\mathbf{z})$ sein, so liegt der Punkt \mathbf{y} genau auf der Geraden zwischen Punkt \mathbf{x} und \mathbf{z} . [Elkan, 2003]

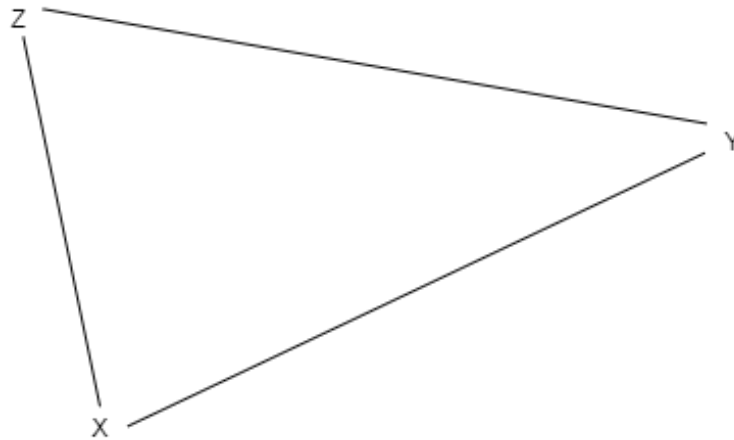


Abbildung 2.7: Es gilt $d(X,Z) \leq d(X,Y) + d(Y,Z)$

Um nun den K-Means Algorithmus zu verbessern, wird ein Datenpunkt \mathbf{x} sowie zwei Clusterzentren \mathbf{c} und \mathbf{c}' betrachtet. \mathbf{c} steht für das Clusterzentrum zu dem \mathbf{x} zugewiesen ist und \mathbf{c}' steht für alle anderen Clusterzentren. Ziel ist es zu beweisen, dass $d(\mathbf{x},\mathbf{c}) \leq d(\mathbf{x},\mathbf{c}')$ ist. Durch die Dreiecksungleichung ist bekannt, dass der Abstand zwischen den zwei Clusterzentren \mathbf{c} und \mathbf{c}' größer oder gleich dem doppeltem Abstand von dem Punkt \mathbf{x} zu einem Clustercenter \mathbf{c} oder \mathbf{c}' sein muss. Durch Formel 2.6 lässt sich beweisen, dass $d(\mathbf{x},\mathbf{c}) \leq d(\mathbf{x},\mathbf{c}')$ ohne den $d(\mathbf{x},\mathbf{c}')$ berechnen zu müssen. [Elkan, 2003]

$$d(x, c) \leq \frac{1}{2}d(c, c') \quad (2.6)$$

Um die Anzahl an Distanzberechnung weiter zu verringern wird ein obere Grenze $\mathbf{u}(\mathbf{x})$ und eine untere Grenze $\mathbf{l}(\mathbf{x},\mathbf{c})$ definiert. Diese Grenzen schließen den Wert $d(\mathbf{x},\mathbf{c})$ ein. Für $\mathbf{u}(\mathbf{x})$ gilt, $\mathbf{u}(\mathbf{x}) \geq d(\mathbf{x},\mathbf{c})$. Das bedeutet eine neue Distanzmessung von $d(\mathbf{x},\mathbf{c})$ ist nur dann erforderlich wenn $\mathbf{u}(\mathbf{x}) > \frac{1}{2}d(\mathbf{c},\mathbf{c}')$ gilt. Auf alle Clusterzentren betrachtet, bleibt der Datenpunkt \mathbf{x} zu dem Clusterzentrum \mathbf{c} zugewiesen, falls $\mathbf{u}(\mathbf{x}) \leq \frac{1}{2}d(\mathbf{c},\mathbf{c}')$ für alle Clusterzentren \mathbf{c}' zu denen \mathbf{x} nicht zugewiesen ist. Dies ist in Formel 2.7 dargestellt. [Elkan, 2003]

$$u(x) \leq \frac{1}{2} \min_{c'} d(c, c') \quad (2.7)$$

$\mathbf{u}(\mathbf{x})$ wird zu Beginn des Clusters auf $d(\mathbf{x},\mathbf{c})$ gesetzt, also den Abstand von \mathbf{x} zu dessen nächstem Clustercenter \mathbf{c} . Sollte sich das entsprechende Clustercenter \mathbf{c} verschieben, so wird $\mathbf{u}(\mathbf{x})$ um die Differenz dieser Verschiebung erhöht. Dies wird in Formel 2.8 dargestellt, wobei $\mathbf{m}(\mathbf{c})$ für das verschobene Cluster \mathbf{c} steht. [Elkan, 2003]

$$u(x) = u(x) + d(m(c), c) \quad (2.8)$$

Diese Erhöhung beschreibt die Annahme, dass sich das Clustercenter \mathbf{c} vom Punkt \mathbf{x} wegbewegt. Für $\mathbf{l}(\mathbf{x}, \mathbf{c})$ gilt, $\mathbf{l}(\mathbf{x}, \mathbf{c}) \leq \mathbf{d}(\mathbf{x}, \mathbf{c})$. Hierbei wird $\mathbf{l}(\mathbf{x}, \mathbf{c})$ zu Beginn ebenfalls auf $\mathbf{d}(\mathbf{x}, \mathbf{c})$ gesetzt. Dieser Wert verändert sich ebenfalls, sobald das Clustercenter \mathbf{c} sich verschiebt. Dabei wird $\mathbf{l}(\mathbf{x}, \mathbf{c})$ hierbei um Differenz der Verschiebung verringert und es gilt $\mathbf{l}(\mathbf{x}, \mathbf{c}) \geq \mathbf{0}$. Dies wird in Formel 2.9 dargestellt. [Elkan, 2003]

$$l(x, c) = \max(0, l(x, c) - d(m(c), c)) \quad (2.9)$$

Dies Verringerung beschreibt die Annahme, dass sich das Clustercenter \mathbf{c} zum Punkt \mathbf{x} hinbewegt.

$\mathbf{l}(\mathbf{x}, \mathbf{c})$ berechnet jeden Abstand von jedem Punkt \mathbf{x} zu jedem Clustercenter \mathbf{c} . Somit kann die Bedingung $\mathbf{u}(\mathbf{x}, \mathbf{c}) \leq \mathbf{l}(\mathbf{x}, \mathbf{c}')$ überprüft werden. Ist demnach die obere Grenze des Punktes \mathbf{x} zu dem Clustercenter \mathbf{c} kleiner als die untere Grenze des gleichen Punkten \mathbf{x} zu einem nicht zugeordneten Clusterzentrum \mathbf{c}' , so müssen aufgrund von Formel 2.10 keine Distanzberechnungen für den Punkt \mathbf{x} durchgeführt werden. Bei jeder Berechnung von $\mathbf{d}(\mathbf{x}, \mathbf{c})$ wird $\mathbf{l}(\mathbf{x}, \mathbf{c})$ mit dem berechneten Wert gleich gesetzt. Falls der Punkt \mathbf{x} einem neuen Clusterzentrum \mathbf{c} zugewiesen wird, wird $\mathbf{u}(\mathbf{x})$ gleich $\mathbf{d}(\mathbf{x}, \mathbf{c})$ gesetzt. [Elkan, 2003]

$$d(x, c) \leq u(x) \leq l(x, c') \leq d(x, c') \quad (2.10)$$

Sollte jedoch $\mathbf{u}(\mathbf{x}, \mathbf{c}) \geq \mathbf{l}(\mathbf{x}, \mathbf{c}')$ sein und Formel 2.7 gelten, so muss $\mathbf{d}(\mathbf{x}, \mathbf{c})$ neu ermittelt und Formel 2.6 für eine richtige Clusterzuordnung überprüft werden. Durch diesen Ansatz lassen sich unnötige Distanzberechnungen vermeiden. [Elkan, 2003]

2.9 Verwendete Produkte und Software

In der folgenden Auflistung werden alle verwendeten Produkte und Software aufgeführt, die für diese Arbeit nötig waren.

Unity

Unity ist eine Spiele-Engine, welche eine breite Zielplattformauswahl hat, neben Betriebssystemen wie Windows, Mac OS, Linux, Android oder IOS, auch für Spielekonsolen wie Playstation oder Xbox und für Webseiten. [Unity, 2019]

Unity bietet eine Plattform, von welcher Spiele einfach entwickelt werden können und das auch mit weniger Implementierung als bei herkömmlichen Vorgehen. [Unity, 2019]

Diese Engine besteht aus zwei Komponenten, den Einstellungen in der Unity-Oberfläche und den C#-Skripten, welche der Unity Oberfläche hinzugefügt werden. Bei der Nutzung von Unity werden Objekte erstellt. Diese werden in dem Inspector bearbeitet. Das wird durch die vorgefertigten Skripte von Unity oder durch eigene Skripte umgesetzt. [Unity, 2019]

Unity WebGL

Um die Unity-Spiele in einer Studie nutzen zu können, ohne dass die Teilnehmer eine

Unity-Lizenz und die entsprechende Hardware dafür besitzen, werden alle Programme als Web-Version implementiert. Unity bietet dafür die Zielplattform Unity WebGL an. [Unity Technologies, 2021]

Visual Studio 2019

Eine passende Plattform um die C#-Skripte für Unity zu implementieren ist Visual Studio 2019. Auf dieser Plattform kann mit unterschiedlichen Programmiersprachen implementiert werden. Dabei werden zur Übersicht zum Beispiel Variablen und Klassennamen markiert und auftretende Compile-Fehler angezeigt. [Microsoft, 2021]

xAPI-Tracker von ADL.NET

Der xAPI-Tracker von der Advanced Distributed Learning (ADL) Initiative wurde für Unity entwickelt. Dadurch muss kein eigener implementiert werden. Herkömmliche xAPI Tracker werden von Unity WebGL nicht unterstützt. [Advanced Distributed Learning, 2020]

JSON.net von Unity

JSON.net ist ein Asset aus dem Asset Store von Unity. Durch diese Bibliothek können JSON und BSON Dateien in Unity gelesen und bearbeitet werden. Es unterstützt WebGL-Versionen. [parentElement, 2017]

Flask & Gunicorn

Flask ist ein Package für die Programmiersprache Python. Flask ist ein Webframework mit dessen Hilfe sich Webanwendungen erstellen lassen. Dies wird mithilfe von Python Decorators umgesetzt. Mit dessen Hilfe lassen sich einzelne Webrouen definieren. Um nun anzugeben, welche Web Route wie triggern soll, wird in den Decorator die URL, die nach dem Domain Namen steht, angegeben. Wird nun diese URL angefragt, wird der Code der hinterlegten Python Methode ausgeführt. Um die Flask Webanwendung verwenden zu können, muss auf einem Webserver laufen. Dafür wird das Python Package Gunicorn verwendet. Dieses ist ein in Python geschriebener HTTP Webserver. Er bietet eine einfache Verbindung zu Flask. [Ronacher, 2021], [Chesneau, 2021]

Heroku

Heroku ist ein Webservice, mit dessen Hilfe sich Webserver hosten lassen können. Dabei bietet Heroku ein breites Spektrum an unterstützten Programmiersprachen an. Heroku bietet verschiedenen Preismodelle an, je nach Größe und Auslastung des Webserver. Um einen Webserver auf Heroku hochzuladen, müssen lediglich ein Git Repository verwendet werden. Dieses enthält für eine Flask Webanwendung folgende Inhalte. Alle Python Dateien, die für die Flask Routen erforderlich sind; alle verwendeten Packages der Python Dateien in einer Requirements Datei; sowie ein Procfile für Gunicorn um die Flask Anwendung zu identifizieren. [Salesforce, 2021]

SQLAlchemy

SQLAlchemy ist ein Package von Python. Mit diesem lässt sich eine Datenbank an das Py-

thon Programm anbinden. SQLAlchemy hat den Vorteil, dass es einfach zu handhaben ist, da Daten ohne „typische“ SQLAnweisungen mit der Datenbank ausgetauscht werden können. Des Weiteren bietet SQLAlchemy eine Vielzahl an Unterstützten SQLDialekten. Für die Anwendung mit Heroku kann auf Seiten von Heroku eine Online Postgres Datenbank erstellt werden, die mit SQLAlchemy über eine Datenbank URI verbunden wird. [Bayer, 2021]

Numpy & Sklearn

SKlearn ist ein Python Package um verschieden machine learning Algorithmen zu Implementieren. Die Daten für diese Algorithmen sind Numpy Arrays. Numpy ist hierbei ebenfalls ein Python Package für das Numerische Rechnen und das Arbeiten mit mehrdimensionalen Arrays. [Oliphant, 2021], [Cournapeaul, 2021]

Matplotlib

Matplot ist ein Package für Python für die grafische Darstellung von Daten. Hierbei benötigen viele Funktionen eingaben von Numpy Daten. [Hunter, 2021]

In folgendem Kapitel wird der momentane Stand der Forschung erläutert, welcher Bezug auf maschinelle Unterstützung und xAPI-Tracker in unterschiedlichen Anwendungen hat. Dadurch wird die Motivation, sowie der Nutzen dieser Arbeit verdeutlicht.

3.1 Maschinelle Unterstützung für nicht spezialisiertes Personal

In der Arbeit „Adaptive Human-Machine Interfaces in Cognitive Production Environments“ wurde eine Mensch-Maschine-Schnittstelle genutzt, um einen nicht spezialisierten Industriearbeiter ohne Vorkenntnisse durch eine Produktionsaufgabe zu führen. Dafür wurde der Arbeiter mit Gesten- und Objekterkennung überwacht und mit einem Retina-Display, Text-to-Speech-Befehlen und auch akustische Signalen jeweils auf den nächsten Schritt in der Aufgabe hingewiesen. Der Arbeiter wird überwacht, um die Produktionslinie nachzuvollziehen und die darauf angepasste Anweisung zu übermitteln. Hier wurden die Möglichkeiten und die auftretenden Nachteile einer solchen Systems untersucht. In der Durchführung wurde ein virtueller Arbeitsplatz genutzt, um konkrete Anwendungsfälle der Mensch-Maschine-Schnittstelle zu testen und zu evaluieren. [Wallhoff, Ablaßmeier u. a., 2007]

3.2 Anwendungen mit xAPI

In vielen Bereichen der Lernanalyse müssen Nutzerdaten überwacht und für spätere Auswertungen gespeichert werden. In der Anwendung werden xAPI-Tracker in unterschiedlichen Bereichen verwendet und mit variierenden Erweiterungen angepasst, um für die entsprechende Disziplin geeignet zu sein.

Clustering In der Arbeit „Research on Adaptive Learning Prediction Based on XAPI“ einer Studie wurden gezielte Unterstützungsdienste für Studierende getestet. Dabei gab es zwei Arten von Datensätzen. Bei dem ersten Versuch wurden die Lerndaten mithilfe der Onlineplattform aufgenommen und ein weiteres mal wurde das Lernverhalten der Studierenden mit xAPI-statements aufgenommen. Dadurch sollte die Bestimmung des Lernzustandes möglich sein und die per-

sonalisierten Lernzustände sollten erkannt werden. Mit einer Vielzahl an Machine-Learning-Algorithmen, wie zum Beispiel Clustering sollten nun weitere Lernzustände vorhergesehen werden. Das Vorhersagemodell mit den xAPI Daten, wies eine höhere Genauigkeit auf, als das Modell mit den herkömmlichen Online-Lerndaten. [Xiao u. a., 2020]

Lernanalyse In dem Bereich der Lernanalyse ist es von Bedeutung große Mengen an Lern-
daten zu sammeln, um den Erfolg des Lernens zu verbessern. In dem Beitrag „An enhanced xapi
data model supporting assessment analytics“ wurde die Anwendung eines xAPI-Trackers in der
Lernanalyse untersucht. Es wurden die Funktionen, Beurteilungsdaten zu verfolgen und zu
speichern, überprüft. Dabei wurde versucht das bestehende Datenmodell des xAPI-Standards
zu verbessern, um die Bewertungsanalyse zu fördern und effektiv zu unterstützen. Dies wurde
durch Logikregeln, um diesen Vorgang zu bestärken, erweitert. Dieses neue Modell bietet die
Möglichkeit mehrere Bewertungsdaten zusammenzufassen und zu speichern. Dadurch sollte der
Prozess der Bewertungsanalyse wirksam durchgeführt werden. [Nouira, Cheniti-Belcadhi und
Braham, 2018]

Serious Games Bei der Arbeit „Applying standards to systematize learning analytics in se-
rious games“ gibt einen Überblick über den gegenwärtigen Stand von Learning Analytics, Daten-
standards und Serious Games. Dabei wird untersucht wie die Interaktionen der Spieler in Serious
Games verfolgt werden könne. Dadurch wird ermittelt welche Metriken sich daraus entwickeln
könnten. Darauf aufbauend wird ein Interaktionsmodell entwickelt, welche als Basis für Lern-
analysen in Serious Games verwendet werden kann. Bei der Implementierung dieses Modells
wurde ein xAPI-Tracker für die Datenaufnahme verwendet. Dieses Modell soll möglichst viele
Anwendungsfälle abdecken, um für die Mehrzahl der Serious Games nutzbar zu sein. Abschlie-
ßend wurde eine verwendbare Demo für Serious Games entwickelt, welche Interaktionen verfolgt
und speichert. [Serrano-Laguna u. a., 2017]

In der Arbeit „Systematizing game learning analytics for serious games“ soll ein xAPI-Tracker
in ein Lernspiel eingebaut werden. Die Wirksamkeit von Serious Games wird erhöht, wenn die-
se messbar und überprüfbar gemacht werden. Denn werden die Interaktionsdaten von Schülern
erfasst, kann ihr Lernprozess verstanden und verbessert werden. Um diese Datensammlung in-
tuitiv nutzbar zu machen, wird ein Visualisierung für Lehrende eingebaut. Die Datensammlung
und ihre Darstellung wurden beide so konstruiert, dass sie einfach für sämtliche Spiele-Engines
in andere Lernspiele eingebaut werden können. [Alonso-Fernandez u. a., 2017]

In dem vierten Kapitel werden die Konzepte der einzelnen Komponenten des Assistenten zusammengeführt. Diese Komponenten bestehen aus dem Spiel als Server mit eingebautem Assistenten sowie einem xAPI-Tracker, eine Verbindung zu dem LRS und einem Cluster-Server. Der intelligente Assistent soll die Nutzer unterscheiden können und ob diese Hilfe benötigen oder nicht. In den folgenden Abschnitten werden die Konzepte zur Entwicklung eines solchen Assistenten beschrieben und die Methodik, mit welcher er validiert wird.

4.1 Adaptiver Assistent

In diesem Abschnitt werden die einzelnen benötigten Komponenten zu LISA erläutert. Der Assistent soll dem Spieler automatisch Hilfe anbieten, wenn diese benötigt wird. Außerdem sollen je nach Aufgabe passende Hilfestellungen gegeben werden. Der Algorithmus orientiert sich an Trainingsdaten, sowie dem IPM.

In Abbildung 4.1 ist das Aktivitätsdiagramm des Assistenten LISA angezeigt. Dabei werden die Fälle berücksichtigt, in welchen der Assistent Hilfenotwendigkeit erkennt oder wenn nicht. Diese beiden Fälle spalten sich in korrekter und falscher Vermutung auf und welche Folgen das für den Nutzer mit sich bringt. Im besten Fall spielt der Nutzer, ohne unterbrochen zu werden weiter und im schlimmsten Fall beendet der Nutzer das Spiel aus Frustration oder Langeweile.

Clustering der Nutzerdaten Die Überlegung ist, dass der Assistent Nutzer in verschiedene Cluster einteilt. Die Nutzer in demselben Cluster haben hierbei die gleiche Hilfenotwendigkeit. Ein Nutzer kann jedoch während des Spielens die Cluster wechseln, je nachdem wie er sich verhält. Folgt er den Interaktionen des IPMs, also ist er auf dem richtigen Weg, wird er in ein besseres Cluster verschoben. Sollte er sich viel Verlaufen und von dem idealen Weg abkommen, wird er in ein schlechteres Cluster verlegt. Dabei wird zwischen drei Clustern unterschieden. Das erste Cluster sagt der Nutzer **Braucht lange keine Hilfe**, das zweite Cluster **Braucht demnächst Hilfe** und das dritte Cluster **Braucht schnell Hilfe** aus. Um die Nutzer nun auf die Cluster zu verteilen, werden verschiedene Features des Nutzers erfasst: Hat der Nutzer gerade den richtigen Raum betreten und mit dem richtigen Objekt interagiert? Wie lange hat er für diese Interaktion benötigt und hat er auf nicht interagierbare Objekte geklickt?

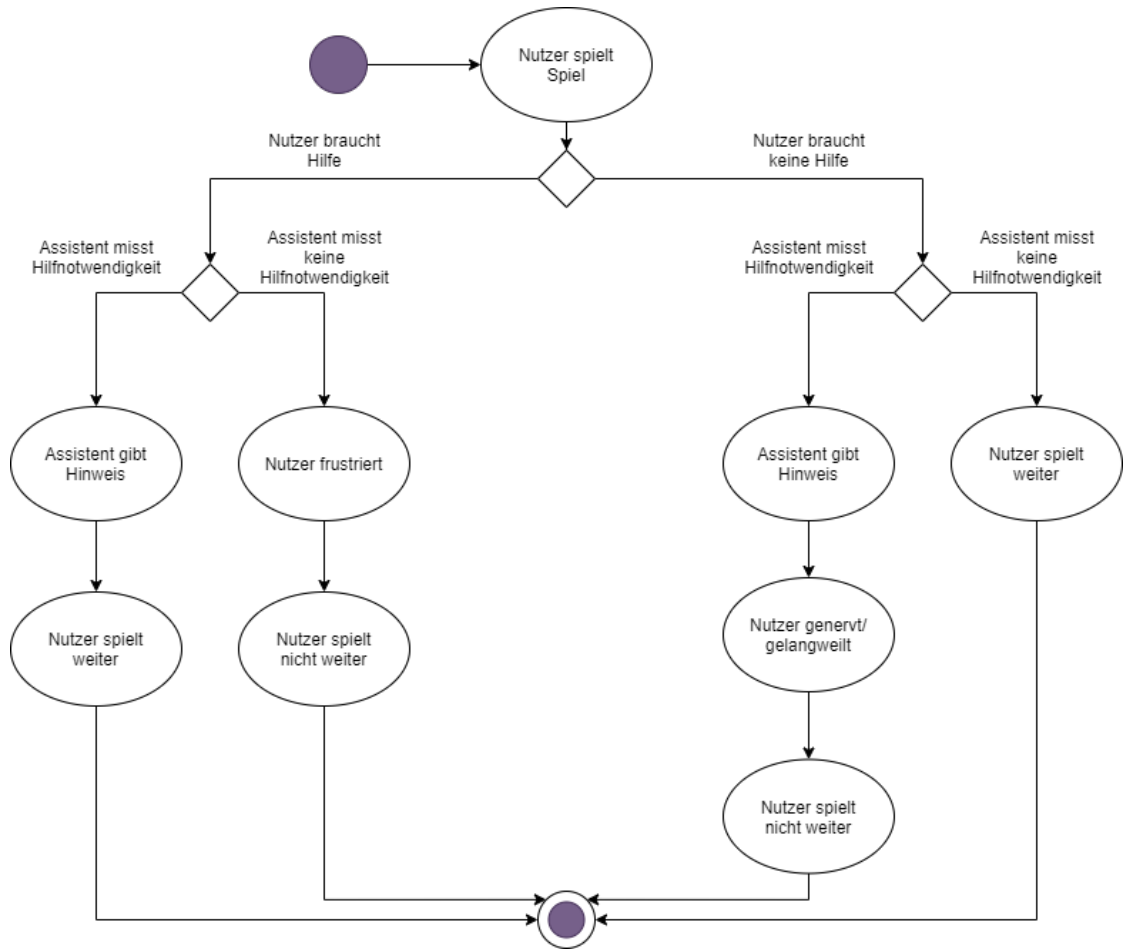


Abbildung 4.1: Die Abbildung zeigt das Aktivitätsdiagramm zu dem Assistenten

Die Aufgaben des Nutzers in einem Spiel bestehen meistens aus verschiedenen Tätigkeiten, die zum Abschließen der Aufgabe erforderlich sind. Deshalb wird für jede Tätigkeit ein separates Clustering durchgeführt, denn es kann Nutzer geben die zunächst Hilfe brauchen, aber anschließend keine mehr benötigen. Würde das nicht berücksichtigt werden und die komplette Aufgabe als ein Clustering behandelt werden, würden solche Nutzer durch einen Misserfolg am Anfang, durchgehend schlechter eingestuft als nötig.

Orientierung im Lost Earth Die Position des Nutzers im Spiel, sowie die nächste erforderliche Interaktion, müssen dem Algorithmus bekannt sein. Um dies zu erreichen, wird ein IPM erstellt. Dieses repräsentiert den idealen Nutzer, der immer den direkten und effizientesten Weg geht und mit allem korrekt interagiert. Dieses IPM wird als Referenz für den Algorithmus genutzt und mit der momentanen Position des Nutzers verglichen. Damit wird definiert, wo der Nutzer sich gerade befindet und wo er als nächsten hinmüsste.

Hilfestellungen durch den Assistenten Der Assistent soll den Nutzer nur dann unterstützen, falls dieser Hilfe benötigt. Das Problem besteht darin was passieren soll, wenn der Nutzer mit nichts interagiert. Deshalb wird das Clustering durch einen Timer unterstützt. Je nach Cluster steigt dieser Timer sehr schnell oder langsam bis zu einem Schwellwert an. Erst bei Erreichen dieses Schwellwerts, soll der Assistent dem Nutzer Hilfe zukommen lassen. Somit wird gewährleistet, dass Nutzer, die mit nichts interagieren ebenfalls Hilfe erhalten. Auch wird so Nutzern aus dem Cluster **schnell Hilfe** etwas Zeit zur Verfügung gestellt sich zu verbessern, bevor die nächste Hilfestellung gegeben wird. Die einzelnen Hilfestellungen werden durch den Assistenten durch das Aufleuchten von Objekten, Messageboxen mit Informationen zur nächsten Tätigkeit, sowie dem Anzeigen der rechten Maustaste zum Verlassen von Räumen gegeben.

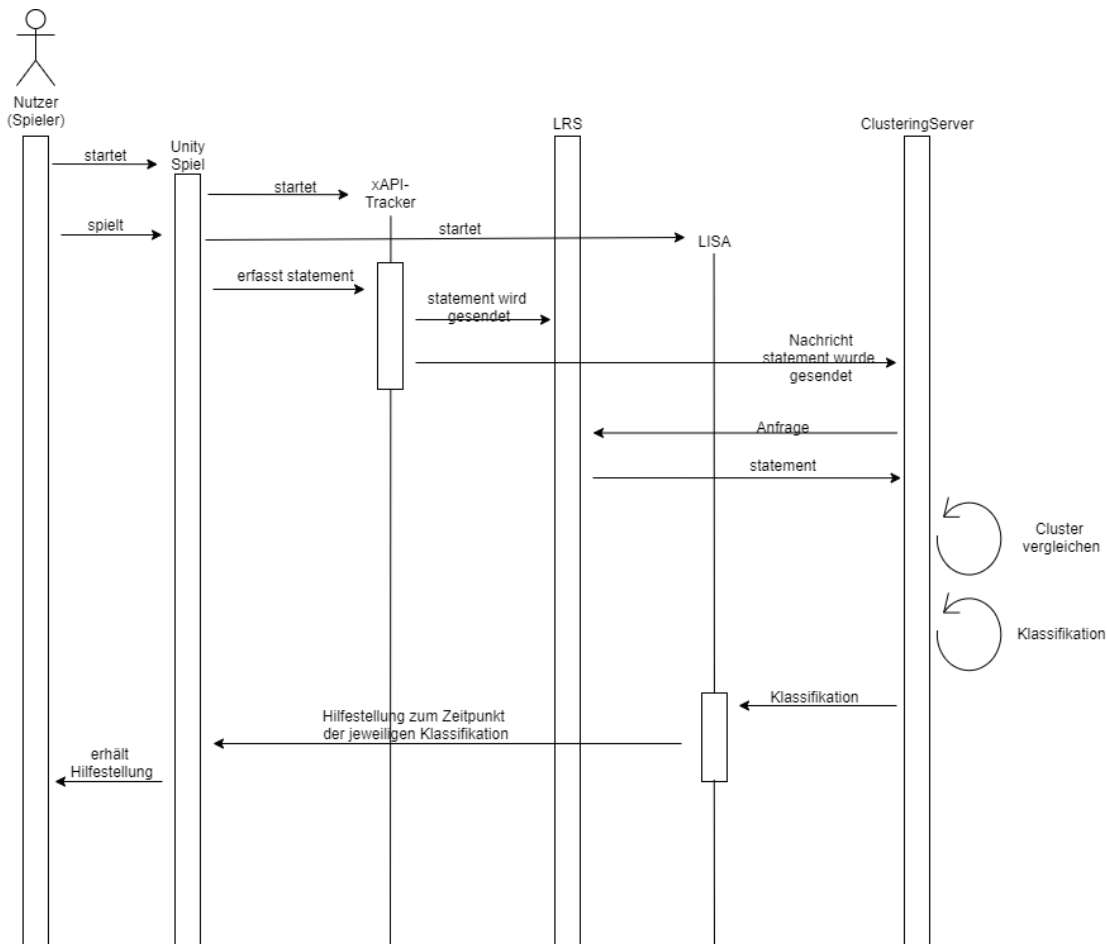


Abbildung 4.2: Die Abbildung zeigt das Sequenzdiagramm zu dem Assistenten

Auslagerung auf einen Server Der Algorithmus wird als Server implementiert. Dies hat mehrere Vorteile. So kann die ganze Rechenleistung für das Clustern ausgelagert werden. Des Weiteren sind so der Algorithmus und der Assistent voneinander getrennt, was eine Wartung

und Erweiterung vereinfacht. Somit ließe sich auch der Algorithmus mit einem anderen Assistenten nutzen. Ebenfalls ist der Assistent selbst ein Package, was eine Einbindung in weitere Spiele vereinfacht. Damit der Server alle notwendigen Informationen erhält, sendet der Assistent Nutzerdaten über xAPI Statements an ein LRS. Dieses LRS dient als Hauptspeicher von allen Nutzerinteraktionen, die vom Server abgefragt werden.

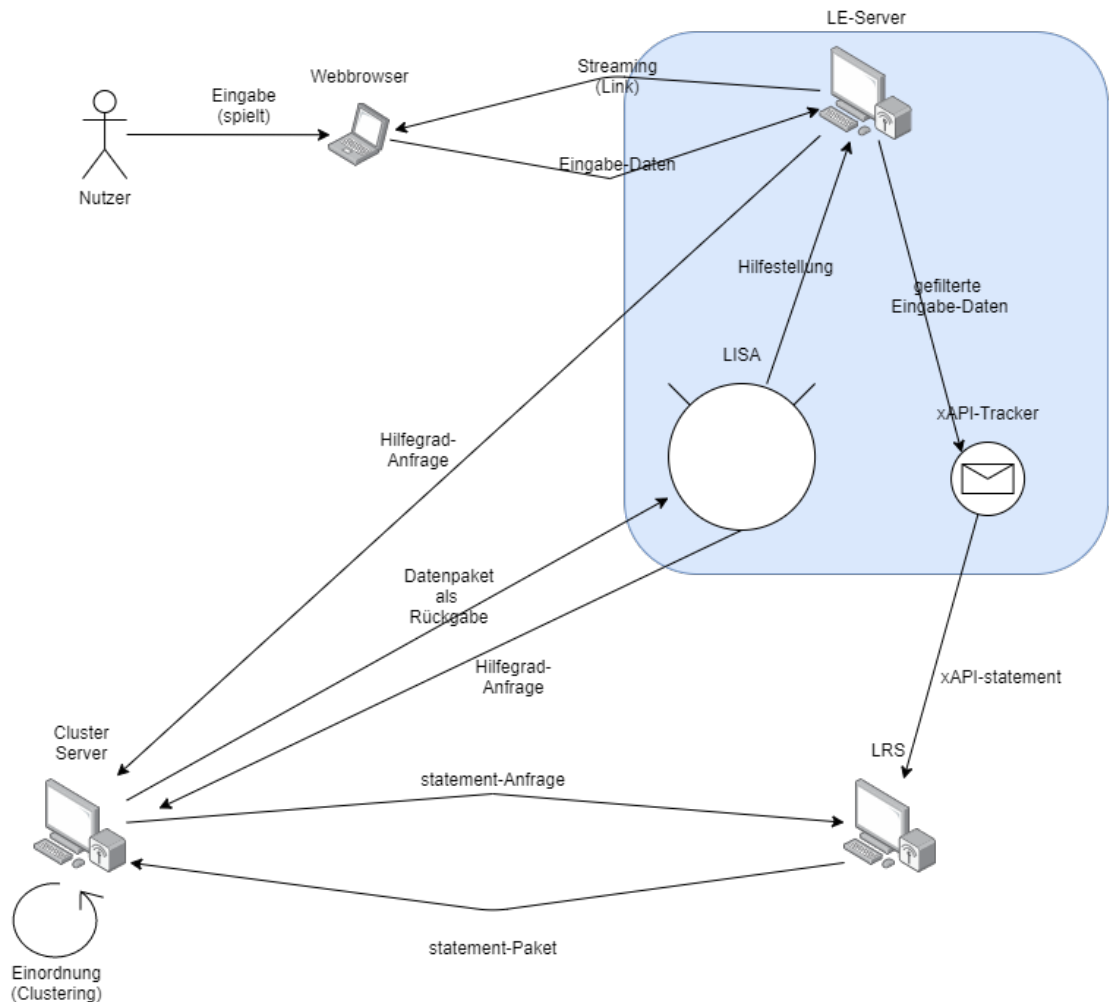


Abbildung 4.3: Die Abbildung zeigt das Architekturdiagramm zu dem System des Assistenten

4.2 Datenkreislauf und Architektur

Um einen Überblick auf die Datenströme und die Architektur zu erhalten, werden diese im Folgenden erläutert. In Abbildung 4.2 ist das Sequenzdiagramm zu sehen, welches einen Durchlauf

der Interaktion des Nutzers mit dem Spiel zeigt. Dabei werden die Vorgänge, welche im Hintergrund laufen abgebildet. Der Datenstrom erfolgt von der Eingabe des Nutzers in das Spiel. Dabei werden über den eingebauten xAPI-Tracker Daten, statements an das LRS gesendet. Zeitgleich bekommt der Clustering-Server eine Nachricht, dass ein neues statement des momentanen Nutzers an das LRS gesendet wurde. Der Assistent, hört auf jedes statement, welches von dem xAPI-Tracker gesendet wurde und sendet eine Anfrage auf den Hilfegrad an den Cluster-Server, damit der neue Nutzer in eines der Cluster klassifiziert werden kann. Daraufhin fragt der Clustering-Server den LRS-Server an und bekommt die neuen statements des zugehörigen Nutzers zurück. Diese neuen Daten werden verarbeitet und das neue Bild des Nutzers wird mit den vorhandenen Clustern verglichen, die Daten anderer Nutzer beinhalten. Dann wird der Nutzer klassifiziert und somit in eines dieser Cluster eingeteilt. Die bestimmte Klassifikation und einige weitere Daten zur Weiterverarbeitung, werden in Form einer Rückgabe an LISA gesendet. LISA gibt über das Spiel eine passende Hilfestellung an den Nutzer weiter, wenn diese benötigt wird. Daraufhin tätigt der Nutzer weitere Eingaben und der Datenkreislauf wiederholt sich. In Abbildung 4.3 sind die Datenströme und die dazugehörigen Aktoren abgebildet. Zu dem Spiel gehören alle Aktoren, die mit der grauen Umrahmung kenntlich gemacht wurden.

4.3 EVA-Prinzip

Typischerweise erfolgt die Datenverarbeitung nach dem EVA-Prinzip, EVA steht hierbei für Eingabe-Verarbeitung-Ausgabe. IPO ist die englische Version und steht für Input, Processing und Output. Mit der **Eingabe** wird gestartet. Dazu gehören alle Vorgänge, die Daten in das System eingeben. Darunter fallen meist die Eingaben von Nutzern, Sensordaten oder auch externe Programme. Nach der Eingabe folgt die **Verarbeitung**. Hier werden mit den Eingaben meist Prozesse gestartet oder beendet. Ein Prozess kann jede mögliche Aufgabe übernehmen, wobei er jede Eingabe für die Anforderung des Prozesses entsprechend umwandelt oder weiterverarbeitet. Mit der **Ausgabe** wird der Vorgang beendet und das Ergebnis ausgegeben. Zu den Ausgaben zählt meistens Feedback an den Nutzer oder externe Programme. Das Feedback können z.B. Daten oder Ausgaben auf dem Bildschirm oder bestimmtes Reagieren der Hardware sein. [Printer Care Service GmbH, 2020]

Im Folgenden wird das EVA-Prinzip auf diesen Assistenten LISA angewandt. Oft wird die Schreibweise als IPO für den Vorgang eines Computers oder Programms gewählt. In Abbildung 4.4 wird das IPO-Prinzip dargestellt.

Input

Der Input oder die Eingaben des Algorithmus sind die Merkmale jedes Nutzers. Dazu zählen die Zeit, welche der Nutzer zwischen korrekten Schlüsselaktionen benötigt, der momentane Raum in welchem der Spieler sich befindet, ob und wie mit Objekten interagiert wurde und wie oft der Nutzer einen NoProgressClick tätigt, also wie oft er etwas mit der Maus anklickt, wenn es nicht nötig oder keinen Nutzen mit sich bringt. Wenn er zum Beispiel auf ein Objekt klickt, mit welchem nicht interagiert werden kann. Weitere Eingaben

sind, ob Missionen erfolgreich abgeschlossen wurden und Zeit zwischen Aktionen. Um diese Daten als Eingabe-Werte nutzen zu können, werden xAPI statements genutzt. Dabei werden zum Beispiel das Abschließen der Mission mit der benötigten Zeit zusammengefügt und als Statement an das LRS gesendet. Weitere statements sind interagierbaren Objekten hinterlegt. Falls diese angeklickt werden, wird ein entsprechendes statement gesendet. Um herauszufinden, ob ein Nutzer daneben klickt, wird während des Mausclicks abgefragt, ob ein interagierbares Objekt ausgewählt wurde. Nachdem der Nutzer den Raum wechselt, wird dieses Abgefragt und ebenfalls ein statement gesendet. Diese statements werden von der KI anfragt, ausgewertet und dann an Lisa gesendet.

Processing

In dem Processing oder dem Vorgang des Algorithmus werden mit den Merkmalen der Nutzer Gruppen eingeteilt. Diese werden in ihren benötigten Hilfegrad gruppiert. Dieser Hilfegrad wird durch ein Cluster bestimmt. Je nach eingeteiltem Cluster erhält der Nutzer früher oder später Hilfe. Grundsätzlich soll LISA zwischen Situationen unterscheiden können, in welchen der Spieler Hilfe braucht oder nicht unterbrochen werden will. Wenn LISA die Daten an die KI weitergibt, werden diese dort verarbeitet. Das bedeutet ein Hilfegrad wird mit den erhaltenen Merkmalen berechnet, indem diese mit Daten von vorherigen Usern verglichen werden. Für jede Tätigkeit in jeder Mission wird so ein Hilfegrad berechnet. Eine Tätigkeit ist z.B. ein Aufgabenabschnitt.

Output

Der Output oder die Ausgaben geschehen über den Assistenten LISA. Dies geschieht durch das Aufleuchten lassen von Objekten mit welchen als nächstes interagiert werden muss oder mit Sprechblasen, in denen entweder eine Mechanik erklärt wird oder ein Hinweis zum weiteren Vorgehen gegeben werden kann.

Der zuvor berechnete Hilfegrad wird an LISA weitergegeben. Dem Hilfegrad entsprechend werden die Hilfestellungen ausgegeben oder nicht, falls der Algorithmus erkannt hat, dass der Nutzer keine Hilfe benötigt.

4.4 Methodik

In dieser Arbeit wird ein Experiment als wissenschaftliche Methode genutzt. Dabei werden in dem Experiment zwei zufällig zugeordnete Gruppen gebildet. Die beiden Gruppen bekommen dieselbe Umfrage mit denselben Fragen und denselben Aufgaben in den Videospiele. Gruppe A beginnt mit dem Assistenten mit adaptiven Hilfeangebot. Gruppe B beginnt ohne den Assistenten und bekommt zu festgelegten Zeitpunkten, festgelegte Hilfestellungen. Nach jeder Mission werden die Teilnehmenden zurück zur Umfrage geführt. Hier bewerten sie die Hilfestellungen und den Assistenten. Anschließend legen sie die zweite Mission ab. Hierbei bekommt Gruppe A festgelegte Hilfestellungen und Gruppe B Unterstützung durch den adaptiven Assistenten. Daraufhin sollen die Teilnehmenden nochmals den Assistenten bewerten.

Makro

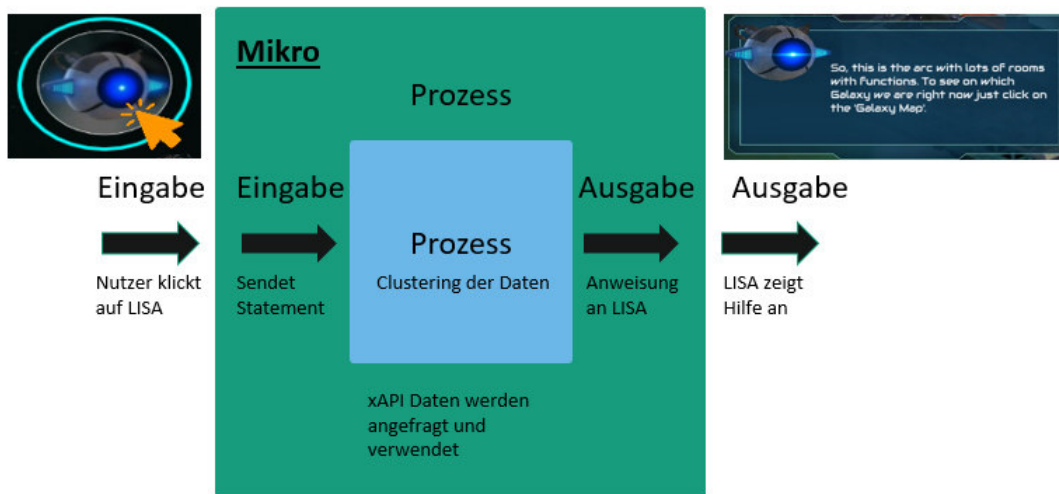


Abbildung 4.4: Das Makro-IPO wird von dem Nutzer gesehen. Innerhalb ist das Mikro-IPO welches die technische Seite beschreibt.

Dieses Experiment wird hauptsächlich quantitativ mit einigen qualitativen Aspekten durchgeführt. Deshalb wird von dem Mixed-Method-Ansatz gesprochen. In diesem Fall ist es das eingebettete Design, da die quantitative Forschung in dem Experiment genutzt wird und nur für einige individuelle Antworten, qualitative Fragen in der Umfrage gestellt werden. Die Auswertung der Ergebnisse erfolgt deduktiv, da die zu Beginn gestellte Forschungsfrage überprüft wird.

Die Gütekriterien Validität, Reliabilität und Objektivität, sowie Intersubjektivität, Transparenz und Reichweite werden erfüllt.

Validität: Das Experiment ist valide, da die interne und externe Validität erfüllt ist. Das Experiment ist intern valide, da es kontrolliert stattfindet und die Probanden in zwei Versuchsgruppen dieselben Aufgaben haben.

Reliabilität: Das Experiment ist reproduzierbar, also kann mehrmals mit unterschiedlichen Personengruppen durchgeführt werden. Dadurch wird das Kriterium Reliabilität erfüllt.

Objektivität: Das Experiment ist ohne Objektleiter möglich. Dadurch wird mögliches subjektives Eingreifen vermieden und das Kriterium Objektivität ist erfüllt.

Transparenz: Die einzelnen Schritte werden dokumentiert, daher wird von Transparenz gesprochen.

Intersubjektivität: Das Experiment ist intersubjektiv, da die gewonnenen Daten ausgewertet und diskutiert werden.

Reichweite: Das Experiment kann mit anderen Teilnehmenden erneut durchgeführt werden.

In dem folgenden Kapitel wird die Umsetzung des Assistenten erläutert. In dieser Arbeit werden die Nutzerdaten, die über xAPI aufgenommen wurden, genutzt, um den passenden Zeitpunkt für Hilfestellungen zu bestimmen. Dabei werden die Nutzer in 3 Cluster eingeteilt. Diese werden „Braucht lange keine Hilfe“, „Braucht demnächst Hilfe“ und „Braucht schnell Hilfe“ genannt. Je nach Cluster wird in größeren oder kleineren Abständen hochgezählt. Falls eine bestimmte Zahl dabei erreicht wird, wird eine Hilfestellung abgespielt.

Dabei soll ein Assistent diese Hilfestellung koordinieren. Das heißt es werden Hilfestellungen bestimmt, die passend zu dem Fortschritt des Nutzers sind. Für diese Koordination werden ebenfalls die aufgenommenen xAPI-Nutzerdaten verwendet und auch ein IPM zur Orientierung benötigt. Durch den Vergleich von dem zurückgelegten Weg, so wie den momentanen Standort des Nutzers und dem IPM, wird erkannt wie weit der Fortschritt im Spiel ist und auch mit wem oder was daraufhin interagiert werden muss.

Um diese Voraussetzungen zu erfüllen, braucht es einen Server für die Einteilung in Cluster, einen Assistenten, der in der Zielsoftware eingespeichert ist und einen xAPI-Tracker, der gewählte Nutzerinteraktionen aufnimmt und an ein LRS sendet. Eine Übersicht auf diese Komponenten ist in Abbildung 4.3 zu sehen.

In den folgenden Abschnitten wird die Umsetzung dieser Anforderungen beschrieben.

5.1 LISA

Der Assistent LISA wurde als Hauptzentrale für Hilfestellung für den Nutzer gewählt. Der Nutzer interagiert mit dem Spiel. Dadurch werden xAPI Statements dieser Interaktionen an das LRS gesendet. Daraufhin fragt der Assistent den Server an, ob der Nutzer gerade Hilfe benötigt und bei welcher Interaktion. Gleichzeitig läuft ein Timer, der bei Erreichen einer Schwelle die Hilfestellung ausgibt. Je nach Antwort des Servers steigt dieser Timer sehr schnell oder langsam an.

Implementierung Die Funktionen Gegenstände aufleuchten und Sprechblasen anzeigen zu lassen, waren in dem Assistenten LISA bereits implementiert. Zu Beginn wurde der Assistent LISA zu einem Unity Package umgewandelt. Hierbei bietet Unity eine Funktion, Objekte und Skripte als Package zu exportieren. Dadurch lässt sich der Assistent auch in weitere Unity Spiele übertragen. Dem Assistenten wurde die Unity Funktion **DontDestro-**

yOnLoad hinzugefügt. Diese ermöglicht es, dem Assistenten in jeder Unity Szene vorhanden zu sein, ohne neu geladen zu werden. Dies hat den Vorteil, dass der Timerwert sowie die Ergebnisse bei einer Serveranfrage immer vorhanden sind.

5.1.1 xAPI Tracker

Alle Informationen über Interaktionen des Nutzers werden mithilfe des xAPI Trackers an das LRS von Fraunhofer gesendet. Dieses speichert alle Statements unter einer individuellen ID ab. Durch diese ID können die Statements gezielt vom LRS angefragt werden.

Implementierung Für das Senden von xAPI Statements an das LRS wurde der **Unity-xAPI-Wrapper** von ADLnet verwendet. Dieser stellt ein **xAPIMessenger** Skript zur Verfügung, das das Senden der Statements an das LRS übernimmt. Diesem müssen lediglich die Authentifizierungsdaten für das Anfragen des LRS mitgeteilt werden. Anschließend muss ein xAPI Statement erstellt werden. Dafür wurde die Klasse **xAPIUtils** erstellt, die mit Methoden des **Unity-xAPI-Wrapper** Package das Statement richtig aufbaut. Der **xAPI-Messenger** bietet im Anschluss die Möglichkeit, Statements aus jedem anderen Skript zu versenden.

5.1.2 Anfrage an den Server

Der Assistent fragt den Server nach der Hilfebedürftigkeit des Nutzers sowie weiteren Informationen über die Position des Nutzers an. Diese Anfrage geschieht nach jeder getätigten Interaktion des Nutzers.

Implementierung In Unity wurden die Anfragen an den Server über Unitys **WebRequest** durchgeführt. Diese **WebRequests** wurden als **CoRoutine** ausgeführt. Das bedeutet, sie werden parallel zum anderen Code bearbeitet. Da das Unity Spiel selbst als WebVersion implementiert wurde ist die Methode über **CoRoutinen** die Einzige, um die Daten mit dem Server auszutauschen. Als Anfrage an den Server wird die ID des xAPI Statements und der Name des Nutzers gesendet. Die Antwort des Servers ist in Json formatiert. Sie wird in Unity über einen Json Parser zu einem Objekt konvertiert. Die einzelnen Json Parameter werden zu Klassen Variablen, die sich einfacher ansprechen lassen.

5.1.3 Hilfe berechnen

Damit der Server die richtigen statements abrufen kann, auch wenn der Nutzer viele Interaktionen hintereinander tätigt, wird eine Reihe, eine **Queue** genutzt. Dabei werden alle statement-IDs in eine

Queue abgespeichert und dieses wird von dem Server abgearbeitet. Dabei wird von der ersten zur letzten ID gearbeitet, die in diese Reihe eingespeichert wird. Dadurch kann der Server gezielt den Nutzer bewerten und dabei die passende Hilfestellung abrufen. Ohne eine solche Reihe, könnte es dazu kommen, dass einige statements übersprungen werden und damit die Orientierung mit dem IPM fehlschlägt.

Um zu vermeiden, dass nun die Hilfestellung ganz schnell hintereinander angezeigt werden, wird eine Abfrage vor dem Aufrufen der Hilfe-Methoden implementiert. Falls sich IDs in der **Queue** befinden, werden keine Hilfestellungen angezeigt. Erst wenn die **Queue** leer ist, werden die Hilfestellungen angezeigt. So ist der Server mit der Bewertung und Orientierung immer auf dem aktuellen Stand, wenn eine Hilfestellung angezeigt wird.

5.1.4 Passende Hilfe anzeigen

Damit dem Nutzer die korrekte Hilfe angezeigt wird, muss mit der Orientierung des Servers gearbeitet werden. Hierzu werden die drei Variablen **userPosition**, **lastInteraction** und **nextInteraction**, welche als Rückgabe von dem Server gesendet werden. In der Abbildung 5.1 werden Beispiele für die Zuordnung der einzelnen Hilfemethoden angezeigt. Im Normalfall ist **userPosition** der momentane Standort des Nutzers und **nextInteraction** ist die nächste Hilfestellung, die er bekommen sollte. In Zeile 1 der Abbildung 5.1 ist der momentane Standort **LISA_Bridge** und die Hilfestellung soll dem Nutzer zeigen wo er **MissionList** öffnet. In der Methode werden diese beiden strings zusammengeführt und es entsteht der neue string **LISA_BridgeMissionList**. So lautet auch der Name der Methode, die diese Hilfestellung anzeigen würde. Die Methode wird mit folgendem Befehl aufgerufen **lisahelper.Invoke(helpMethod, 0)**; Die Methoden sind unter der Klasse **lisahelper** abgespeichert und mit dem Befehl **Invoke** können unter Verwendung von strings, Methoden aufgerufen werden.

Wenn nun aber in **lastInteraction** etwas abgespeichert ist, wird **nextInteraction** nicht berücksichtigt. Ein Beispiel dafür ist in Abbildung 5.1 Zeile 2 zu sehen. Der zusammengefügte string würde dabei **LISA_ArkLISA_Bridge** sein.

Zeile	userPosition	LastInteraktion	NextInteraktion	Hilfe-Methode die aufgerufen wird:
1	LISA_Bridge	None	MissionList	LISA_BridgeMissionList();
2	LISA_Ark	LISA_Bridge	MissionList	LISA_ArkLISA_Bridge();

Abbildung 5.1: Durch dieses Vorgehen wird der Methoden-Name für die anstehende Hilfe definiert.

5.2 Server

Der Algorithmus zur Entscheidung der Hilfenotwendigkeit eines Nutzers wurde auf einen Server ausgelagert. Der Assistent schickt eine Anfrage an den Server. Der Server bestimmt die momentane Interaktion des Nutzers und verarbeitet diese. Die momentane Interaktion des Nutzers wird beim LRS angefragt. Anschließend wird durch einen Bewertungsalgorithmus ein Bewertungsarray erstellt. Des Weiteren wird bestimmt, welche Interaktionen seitens Nutzer korrekt wären und die ideale Zeit der Tätigkeit, in der sich der Nutzer befinden. Das Bewertungsarray wird im Anschluss geclustert und das Ergebnis dieses Clusters sowie weitere Informationen werden als Antwort vom Server an den Assistenten geschickt. Auf dem Server werden die Bewertungsarrays der Nutzer, sowie erforderliche Informationen zur Berechnung dieser, in einer Datenbank gespeichert. Diese Datenbank enthält ebenfalls alle Daten der Cluster sowie das IPM abgespeichert.

Implementierung Der Server wurde in Python mithilfe der Flask und Gunicorn Packages erstellt. Die so entstandene WebAPI befolgt das RESTful Design. Hierbei wurden verschiedene Webrouen für verschiedene Anfragen an den Server erstellt. Dieser Webserver wurde auf Heroku hochgeladen, um ihn über das Internet ansprechen zu können. Die Datenbank ist eine Postgres SQL-Datenbank, die über Heroku der Webanwendung hinzugefügt wurde.

5.2.1 Anfrage an das LRS

Die im LRS gespeicherten statements werden benötigt um den aktuellen Nutzer zu bewerten und zu leiten. Die Anfrage des Assistenten an den Server enthält als übergebene Informationen den Namen des Nutzers sowie die ID, unter der die aktuelle Interaktion des Nutzers auf dem LRS gespeichert ist. Über die ID wird diese Interaktion angefragt und alle für diese Anwendung nicht erforderlichen Informationen aus den enthaltenen Daten werden entfernt.

Implementierung Die Anfrage an das LRS wurde über das Requests Package von Python implementiert. Für die Anfrage an das LRS wurde die entsprechende URI mit der ID der Interaktion angefragt. Die Einzige zusätzliche Information waren die Authentifizierungsdaten, die über den Request-Header angegeben wurden. Die notwendigen Informationen wurde mithilfe von List Slicing, Indexing und Regular Expressions aus der Antwort vom LRS extrahiert und in ein Array geschrieben. Ein Beispiel dieses Arrays ist in Formel 5.1 zu sehen.

$$['selected', 'Lisa - Bridge', '20 40'] \quad (5.1)$$

5.2.2 IPM

Das IPM dient als Referenz um zu wissen, welche Interaktionen der Nutzer als nächstes Tätigen muss. Diese Information nutzt der Assistent für die korrekte Ausgabe der Hilfestellung und der Algorithmus auf dem Server für die Bewertung des Nutzers. In Abbildung 5.2 ist ein Diagramm aller Tätigkeiten aus Mission eins zu sehen.

Implementierung Das IPM wurde mit Hilfe von xAPI Statements aufgenommen. Das Spiel wurde als idealer Nutzer durchgespielt und die so gewonnen Daten wurden als IPM gespeichert.

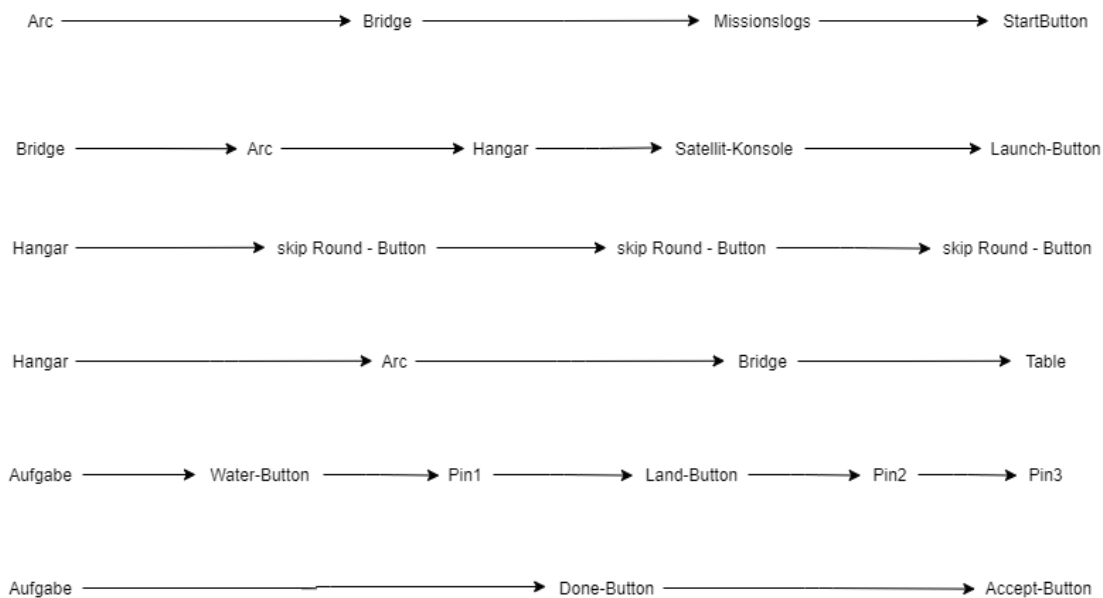


Abbildung 5.2: Das Ideal Path Model des Spiels Speciare Lunar Research Facility.

5.2.3 Daten sammeln

Um Anfangsdaten für die Bildung der Cluster zu erhalten, mussten zuerst Szenarien erstellt werden. Diese Szenarien repräsentieren Nutzer, die bewusst nicht ideal sind. Die Szenarien zeigen Nutzer, die weniger bis viel Hilfe benötigen. Durch diese Szenarien werden die drei Cluster zu Beginn gewährleistet. Da diese Szenarien jeweils bewusst für die Einordnung in eines dieser Cluster erstellt wurden.

Implementierung Die Daten wurden mit Hilfe des xAPI Trackers gesammelt und im Anschluss als Anfangsnutzer dem Clustering hinzugefügt.

5.2.4 Bewertungsalgorithmus

Der Bewertungsalgorithmus bestimmt die Bewertungsarrays des Nutzers für die momentane Tätigkeit. Die vom LRS erhaltenen Daten werden mit dem IPM abgeglichen. Durch diesen Vergleich wird bestimmt, an welcher Position sich der Nutzer gerade befindet und was er als nächstes tun sollte. Sollte er eine Tätigkeit abgeschlossen haben, hat er einen Checkpoint erreicht. Seine weiteren Interaktionen werden dann mit der nächsten Tätigkeit verglichen.

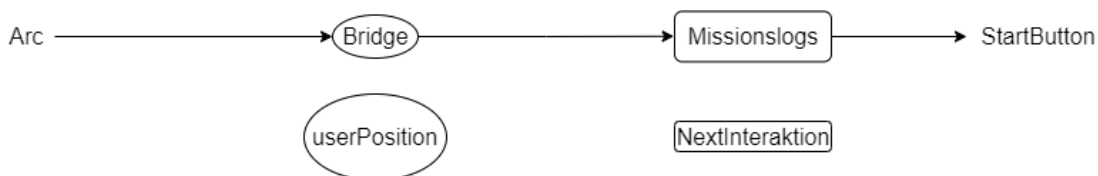


Abbildung 5.3: Die Orientierung bei einem Nutzer mit korrektem Verlauf.

Implementierung Der Algorithmus wurde mithilfe einer if - else - Verschachtelung umgesetzt. Dabei spielen die Variablen **nextInteraktion**, **lastInteraktion** eine zentrale Rolle. **nextInteraktion** zeigt an, mit was der Nutzer als nächstes interagieren muss. In Abbildung 5.3 ist zu sehen, dass sich der Nutzer in der Brücke befindet und als nächstes mit den Missionslogs interagieren muss. Sollte er dies tun, wird dieser entsprechend positiv bewertet. In Abbildung 5.4 ist zu sehen, dass der Nutzer als nächste Interaktion mit dem Start Knopf interagieren sollte. Dies macht er jedoch nicht und kehrt zurück bis in Ark. Entsprechend dieses Verhaltens wird er negativ bewertet. Die Variable **lastInteraktion** sagt nun aus, dass wenn der Nutzer sich wieder zur Brücke bewegen würde, er wieder auf dem richtigen Weg wäre. **lastInteraktion** gibt demnach den Weg zurück zu **nextInteraktion** an. Die Interaktionen einer Tätigkeit des IPM sind in einem Array in der richtigen Reihenfolge aufgelistet. Je nach erreichtem Checkpoint wird dieses Tätigkeitsarray aus der Datenbank, zum Vergleich mit dem Nutzer, geladen. **lastInteraktion** und **nextInteraktion** sprechen durch Indexing jeweils bestimmte Stellen des Arrays an. Sollte **nextInteraktion** außerhalb des Arrays liegen, wurde diese Tätigkeit vollständig abgeschlossen. Der Checkpoint wird um eins erhöht und zum Vergleich mit dem Nutzer wird nun die nächste Tätigkeit geladen.

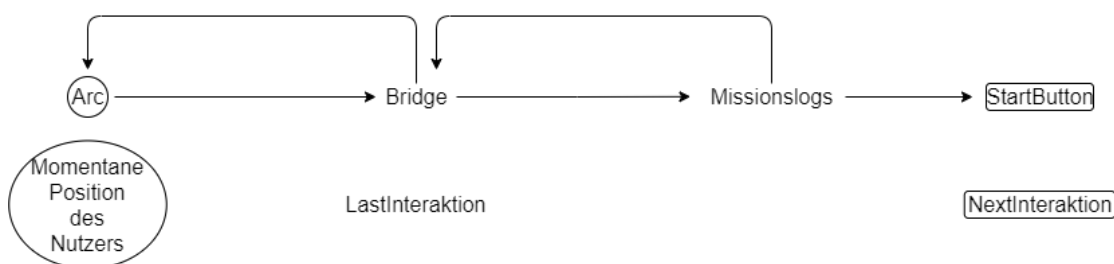


Abbildung 5.4: Die Orientierung bei einem Nutzer, der aus dem Raum geht, in der die Mission weitergeht.

5.2.5 Bewertungsarrays

Die zu clusternden Daten sind jeweils die einzelnen Bewertungsarrays eines Nutzers für eine Tätigkeit. In diesem Bewertungsarray werden die Features, Interaktion mit Raum und Objekt, benötigte Zeit zwischen Interaktionen und Klicks auf nicht interagierbare Objekte, ermittelt. Dieses Bewertungsarray ist in Formel 5.2 zu sehen und setzt sich wie folgt zusammen.

$$[Raum, Objekt, Zeit, NoProgressClick] \quad (5.2)$$

Für den Anfang jeder Tätigkeit werden alle Werte in dem Array auf null gesetzt.

Raum und Objekt Sollte sich der Nutzers dem IPM entsprechend in einen richtigen Raum bewegen oder mit einem richtigen Objekt interagieren, dann wird der Wert für **Raum** beziehungsweise **Objekt** um eins Reduziert. Sollte er sich in einen falschen Raum bewegen oder mit einem falschen Objekt interagieren, so wird der Wert um zwei erhöht. Das heißt je niedriger ein Wert in dem Array umso besser. Der Wert wird bei einer falschen Interaktion um zwei und nicht um eins erhöht, da sonst Fehler des Nutzers nicht erkannt werden. Eine falsche und eine richtige Interaktion würden sich somit auflösen.

Implementierung In Unity wurden zwei neue Skripte **OnScene** und **OnClick** erstellt. Das **OnScene** Skript wird in jeder Scene in Unity platziert. Es sendet ein xAPI statement beim Wechseln der Scene, mit dem aktuellen Raum, in dem sich der Nutzer befindet. Das **OnClick** Skript wird allen interagierbaren Objekten im Spiel hinzugefügt. Es sendet ein xAPI statement mit dem momentan interagierten Objekt. Die Berechnung der Werte für das Bewertungsarray findet auf dem Server statt.

Zeit Der Wert für die **Zeit** wird immer nur bei einer Richtigen Interaktion ermittelt. Dieser Wert sagt aus, wie schnell der Nutzer im Vergleich zum IPM handelt. Sollte der Nutzer schneller als das IPM sein, wird der Wert um eins reduziert. Ist der Nutzer doppelt so langsam, wird der Wert um eins erhöht. Sollte der Nutzer für die Interaktionen länger als das IPM brauchen, aber nicht länger als die doppelte Dauer des IPMs, bleibt der Wert gleich. Hier ist es nicht erforderlich den Wert bei zu langsamem Handeln um zwei statt um eins zu erhöhen, da hierbei zu langsamem und schnellem Handeln sich für die Bewertung ausgleichen dürfen.

Implementierung Jedes xAPI statement hat einen Zeitstempel. Durch dieses lässt sich die Zeit zwischen zwei Interaktionen berechnen. Auf dem Server wird die Differenz zwischen der neuen Interaktion und der vorherigen Interaktion berechnet und durch die Differenz des IPMs an dieser Stelle geteilt. Das erhaltene Verhältnis bestimmt die Anpassung des **Zeit** Wertes im Bewertungsarray auf dem Server.

No Progress Click Der Wert für **No Progress Click** zählt, wie oft der Nutzer versucht hat, mit einem nicht Interagierbarem Objekt zu interagieren. Für jede Interaktion steigt der Wert um eins und kann für diese Tätigkeit nicht gesenkt werden.

Implementierung In Unity wurde ein neues Skript **Clickable** erstellt. Es dient nur dem Zweck, interagierbare Objekte zu markieren. Dieses Skript ist erforderlich, da das **OnClick** Skript nicht allen interagierbaren Objekten hinzugefügt werden soll. Dies hat den Grund, dass diese Objekte kein xAPI statement senden sollten, jedoch trotzdem interagierbar sind. Über ein weiteres Skript **NoProgressClick** wird nach jedem Klick des Nutzers bestimmt, ob das geklickte Objekt das **Clickable** Skript enthält. Falls dies nicht der Fall ist, wird ein xAPI statement gesendet das aussagt, dass nichts interagierbares geklickt wurde. Die Berechnung der Werte für das Bewertungsarray findet auf dem Server statt.

5.3 Clustering

Für die Art des Clusterings wurde K-Means verwendet. Dies hat den Grund, dass die Anzahl von gesuchten Clustern von Beginn an bekannt ist. Als Algorithmus für K-Means wurde der Standard Elkan - Algorithmus gewählt. Es wurde dieser Algorithmus genutzt, da er die Rechenzeit im Vergleich zum Standard EM-Algorithmus verringert. Die Metrik für die Distanzmessung während des Clusterings ist der euklidische Distanz zwischen den einzelnen Daten. Dies bietet sich an, da die Daten alle im euklidischen Raum sind.

Implementierung Das Clustering wurde mithilfe des Python Package **sklearn** umgesetzt. Dieses bietet eine einfache Möglichkeit, verschiedene Machine Learning Algorithmen in Python zu implementieren. Dabei findet das Clustern auf dem Server statt. Die Daten für die einzelnen Cluster befinden sich in der Datenbank. Bei einer Cluster Anfrage werden die Daten aus der Datenbank geladen und mittels des K-Means Algorithmus von sklearn geclustert. Anschließend wird durch die Methode **predict()** das Cluster für den momentanen Nutzer ermittelt. Sollte der Nutzer einen Checkpoint erreicht haben, so wird der Nutzer dem Clustering dieser Tätigkeit hinzugefügt.

5.4 Ergebnisse

In dieser Arbeit wurde ein Assistent entwickelt, der mit Hilfe eines Servers, Nutzer in drei Kategorien einteilt und dem Nutzer dementsprechend Hilfe zukommen zu lässt. Der Assistent wurde mit C# in Unity eingebaut, außerdem besitzt er einen Sprite der im Spiel sichtbar ist. Der xAPI-Tracker wurde ebenfalls in Unity eingebaut und mit C#, wie auch der Unity-Oberfläche an das Spiel angepasst. Der Server für das Backend des Assistenten wurde mit Python implementiert. Dieser Server ist zuständig für das Clustering, sowie die Bewertung und Orientierung der Nutzer. Falls ein Nutzer eine Interaktion ausführt, wird diese über den xAPI-Tracker erfasst und als statement an das LRS gesendet. LISA hört durchgehend auf den xAPI-Tracker, sobald ein statement gesendet wird, schickt LISA eine Anfrage zur Einteilung an den Clusterserver und übergibt da-

bei die ID des jeweiligen statements. Der Server fragt daraufhin das LRS nach dem statement mit dieser ID an und filtert die wichtigen Informationen, die zur Berechnung nötig sind, heraus. Anschließend wird der Nutzer bewertet. Dabei wird mit dem IPM verglichen, ob die Interaktion korrekt war. Anschließend wird das berechnete Cluster, die momentane Aktion, die nächste erforderliche Interaktion und die letzte korrekte Interaktion des Nutzers an LISA zurückgegeben. Mit diesen Informationen kann LISA die Hilfestellung koordinieren. Passend zum Cluster wird die Schrittweite des Counters gesetzt. Außerdem wird mit den Informationen zu dem Nutzer ein string geformt, welcher dem Namen einer Methode entspricht, die als nächstes aufgerufen wurde. Diese Methode ist die passende Hilfestellung für den momentanen Nutzer.

Sonderfälle wurden berücksichtigt, in welchen der Nutzer in einen falschen Raum geht oder eine falsche Konsole öffnet, wird der Nutzer wieder auf den richtigen Weg geführt und entweder dazu aufgefordert den Raum zu verlassen oder die momentane Konsole zu schließen. Auch falls eine Mission fehlschlägt, werden die Checkpoints der aktuellen Mission zurückgesetzt und der Spieler wird wieder wie zu Beginn der Mission unterstützt. Auch wenn eine Konsole geöffnet, aber noch nicht mit ihr interagiert wurde und sie einfach wieder geschlossen wurde, wird der Nutzer wieder dazu aufgefordert diese Konsole zu öffnen. Dabei erkennt der Assistent, dass der Nutzer im ersten Moment korrekt reagiert hat und er lediglich Hilfe bei der folgenden Interaktion benötigte.

In diesem Kapitel wird die Verifikation von LISA in Form einer Studie erläutert. Dabei wird der zuvor entwickelte Assistent getestet und anschließend bewertet. Zuletzt werden die Ergebnisse der Studie ausgewertet und mit den Ergebnissen des Assistenten in der Diskussion aufgeführt.

6.1 Hypothesen/ Ziele der Studie

Ist der Assistent smart LISA dem Spieler in Spielen nützlich? Kann dieser Assistent sich adaptiv an den Nutzer anpassen?

- LISA unterscheidet zwischen Situationen, in welchen der Nutzer Hilfe benötigt oder nicht und passt ihr Agieren darauf an.
- LISA führt den Nutzer erfolgreich durch die Missionen.
- Nutzer empfinden LISA als hilfreich.
- LISA kann als **smart** bezeichnet werden.

6.2 Versuchsaufbau

In diesem Abschnitt wird der Aufbau der Studie erläutert. Um die Hypothesen zu überprüfen benötigt es neben der Testgruppe eine Kontrollgruppe. Sobald das Spiel über den Startknopf gespeichert wird, wird mit dem zuvor eingetragenen Namen in dem Spiel ein *user* erstellt. Jeder *user* erhält eine ID. Über die ID in der Datenbank wird der Nutzer in eine Gruppe der zwei Gruppen eingeteilt. Bei einer geraden ID ist der momentane Nutzer in Gruppe 1 und erhält in Mission 1 smart LISA als Unterstützung. Bei einer ungeraden ID ist der momentane Nutzer in Gruppe 2 und erhält statische Hilfe einer „dumb“ LISA. Diese *dumb* LISA gibt dieselben Hilfestellungen wie smart LISA, nur fällt der Timer weg, welcher an den Nutzer angepasst ist. Das bedeutet die Hilfestellungen werden sofort abgespielt und es wird keine Zeit gelassen, dass der Nutzer die Interaktion selbst bewältigt. Nachdem die erste Mission abgeschlossen ist, werden die Assistenten

gewechselt. Gruppe 1 erhält Unterstützung von *dumb* LISA und Gruppe 2 erhält Unterstützung von *smart* LISA.

Der Aufbau der Studie ist Folgender. Zu Beginn werden dem Nutzer einige Fragen gestellt und über einen Link kann das Spiel geöffnet werden. Bereits in den Fragen soll ein Name eingetragen werden. Dies dient der Zuordnung der Umfrage zu den Spieldaten. Denn der eingetragene Name in dem Spiel erhält eine Namenserverweiterung. Diese Endung ist der Tag und die Uhrzeit, bei der auf den Startknopf gedrückt wurde. Ein Beispiel wäre, wenn ein Nutzer am 19.04. um 13.07 Uhr das Spiel über den Knopf startet, ist die Namenserverweiterung: *19137*. So können auch bei einer Namendopplung die Ergebnisse in dem Spiel, denen aus der Umfrage zugeordnet werden. Diese Zusammensetzung ist in Formel 6.1 verdeutlicht.

$$\begin{aligned} \text{username} := & \langle \text{name} \rangle + \langle \text{DateTime.Now.Day} \rangle + \dots \\ & \dots \langle \text{DateTime.Now.Hour} \rangle + \langle \text{DateTime.Now.Minute} \rangle \end{aligned} \quad (6.1)$$

Anschließend kann das Spiel bis zu dem Ende der ersten Mission gespielt werden. Dort erhält der Nutzer einen Code, welchen er in der Studie eingeben muss. Daraufhin kann der Fragebogen weiter ausgefüllt werden. Nachdem die einzelnen Hilfestellungen des Assistenten bewertet wurden, spielt der Nutzer die zweite Mission aus dem Spiel und erhält anschließend einen Code, um die Studie abschließen zu können. Dabei wird erst der Assistent bewertet und anschließend aufgelöst, dass nur in einer Mission der Assistent sich dem Nutzerverhalten angepasst hat. Der Teilnehmende wird aufgefordert, einzuschätzen in welcher Mission er adaptiv unterstützt wurde. Daraufhin soll entschieden werden, ob der Assistent auch in anderen Systemen genutzt werden könnte und welche das sein könnten. Eine der abschließenden Fragen ist, ob der Nutzer LISA als „smart“ bezeichnen würde. Ein Feld für weitere Anmerkungen wird als letzte, optionale Frage zur Verfügung gestellt.

Damit die Daten von Nutzern in der Studie abgespeichert werden können, die den ersten oder zweiten Code nicht freigeschaltet haben, wurde die Angabe des Codes zu einer optionalen Frage. Dadurch konnte auf den Weiterknopf geklickt werden, auch wenn kein Code eingegeben wurde. Falls der Code nicht oder fehlerhaft eingetragen wurde, wird der Teilnehmende auf die letzte Seite der Studie geführt und hat damit die Umfrage frühzeitig beendet. Dadurch werden die Daten, die auf der ersten Seite eingetragen wurden, gespeichert. Nur wenn der Code richtig eingegeben wird, kann der Teilnehmende auf die darauffolgenden Fragen zugreifen.

6.3 Durchführung

Um die erfolgreiche Durchführung der Studie zu gewährleisten, müssen einige Aspekte beachtet werden. Damit die Teilnehmenden von ihrem privaten PC einfach auf die Studie zugreifen können, wurde das Spiel Lost Earth auf einen Server von Fraunhofer hochgeladen. Dieses Spiel kann über einen Link in der Studie geöffnet werden. Dabei wird automatisch ein weiterer Tab geöffnet, damit auf die Studie auch nach dem Spielen zugegriffen werden kann. Der Link zu der Studie wurde per Mail an zwei Studiengänge der Jade Hochschule gesendet. Außerdem konnten auch Einzelpersonen teilnehmen, welche den Link zugeschickt bekamen.

6.4 Ergebnisse

In diesem Abschnitt werden die Ergebnisse der durchgeführten Studie präsentiert. Bei allen Bewertungen, die durch Balkendiagramme dargestellt werden, steht „1“ für nicht zutreffend oder nicht hilfreich und „4“ für sehr zutreffend oder sehr hilfreich.

Umfrage Erfolgreich abgeschlossen

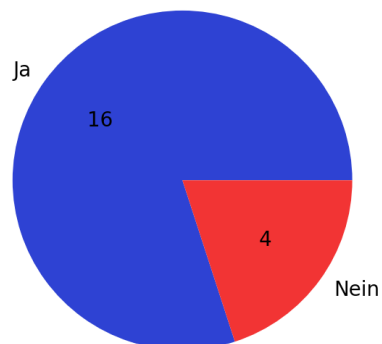


Abbildung 6.1: Die Umfrage wurde erfolgreich abgeschlossen

Durch die Abbildung 6.1 geht hervor, wie viele der Teilnehmenden die Studie erfolgreich abgeschlossen haben. Vier Personen haben entweder das Spiel nicht fertig spielen können oder haben einen der Codes falsch eingegeben und haben somit die Studie frühzeitig beendet. Dabei haben 16 Teilnehmende die Studie erfolgreich abschließen können und haben so auch das Spiel durchspielen können.

Ist Ihnen aufgefallen, dass der Assistent sich in den beiden Spielabschnitten nur einmal an ihr Nutzerverhalten angepasst hat?

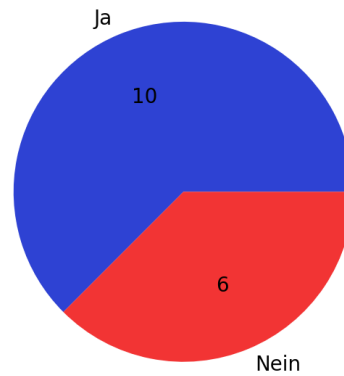


Abbildung 6.2: Ist aufgefallen, dass der Assistent nur in einer Mission adaptiv unterstützt hat?

In der Abbildung 6.2 ist zu sehen, dass von den 16 Teilnehmenden, die beide Missionen abgeschlossen haben, zehn bemerkt haben, dass der Assistent nur bei einer von zwei Mission adaptiv unterstützt hat.

In welcher Mission war der Assistent adaptiv?
adaptiver Assistent aufgefallen

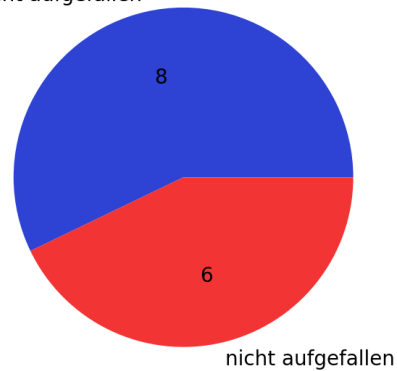


Abbildung 6.3: Unterstützung in den Missionen geschätzt.

In Abbildung 6.3 wird angezeigt, dass acht von vierzehn Teilnehmenden anschließend erfolgreich schätzen konnten, in welcher Mission sie adaptiv unterstützt werden sollten.

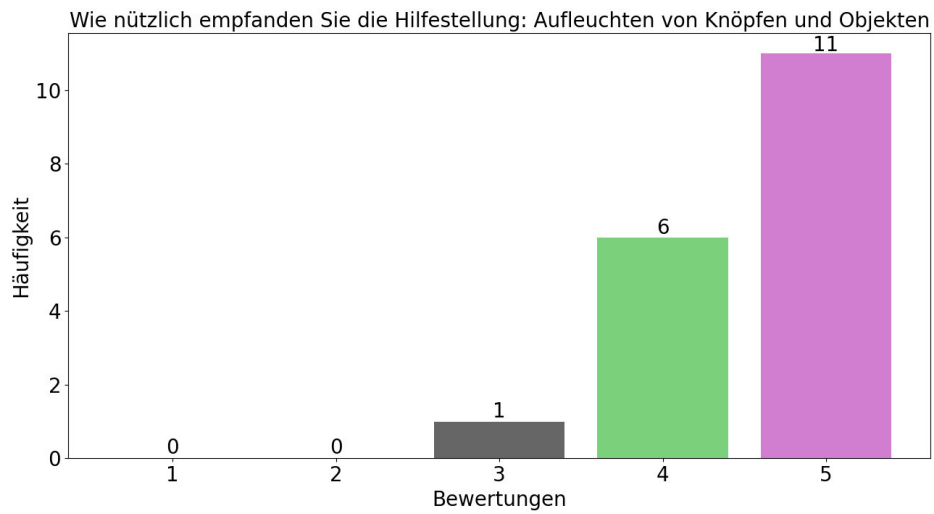


Abbildung 6.4: Wie hilfreich wurde das Aufleuchten von Objekten empfunden

Aus Abbildung 6.4 geht die Bewertung der Hilfestellung in Form von Aufleuchten von Objekten hervor.

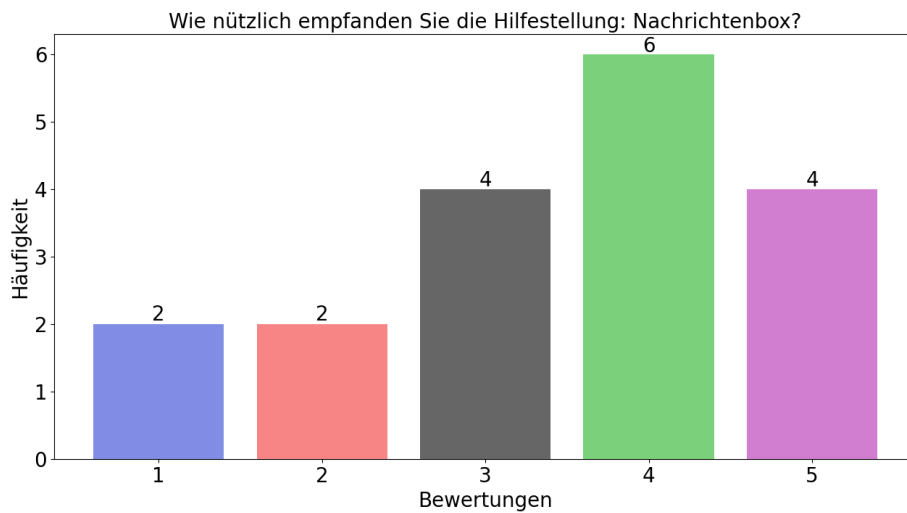


Abbildung 6.5: Wie Hilfreich wurden die Nachrichtenboxen empfunden

Aus Abbildung 6.5 geht die Bewertung der Hilfestellung in Form von Nachrichtenboxen hervor.

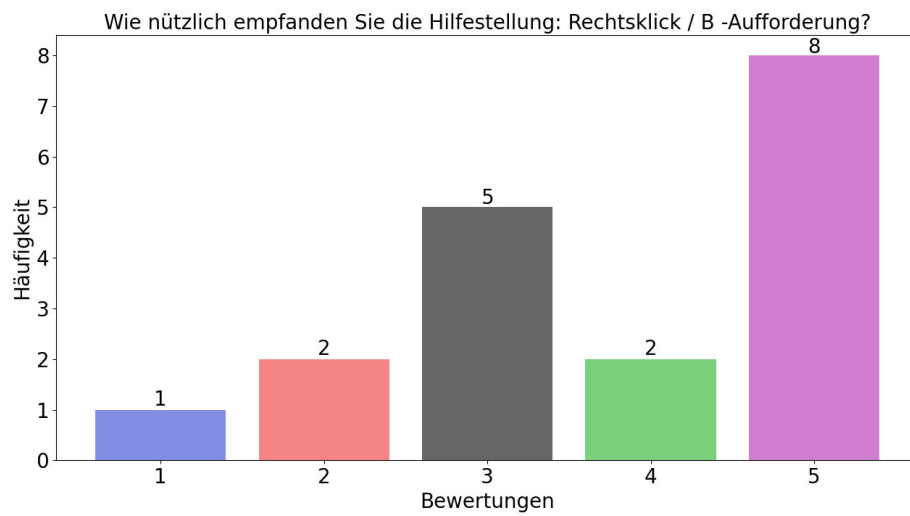


Abbildung 6.6: Wie hilfreich wurde das Raum Verlassen-Symbol empfunden

Aus Abbildung 6.6 geht die Bewertung der Hilfestellung in Form der Aufforderung den Raum zu verlassen, hervor.

Könnte der vorgeführte Assistent LISA, "smart" genannt werden?

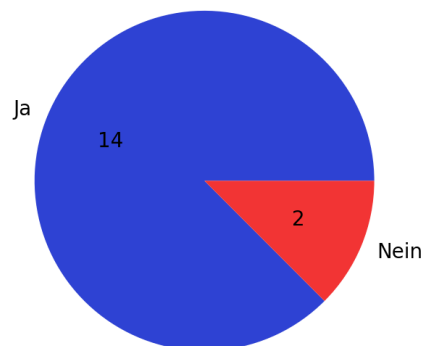


Abbildung 6.7: Wurde LISA abschließend als smart bezeichnet

Durch die Abbildung 6.7 wird gezeigt, dass 14 von 16 Personen den Assistenten LISA als smart bezeichnen würden.

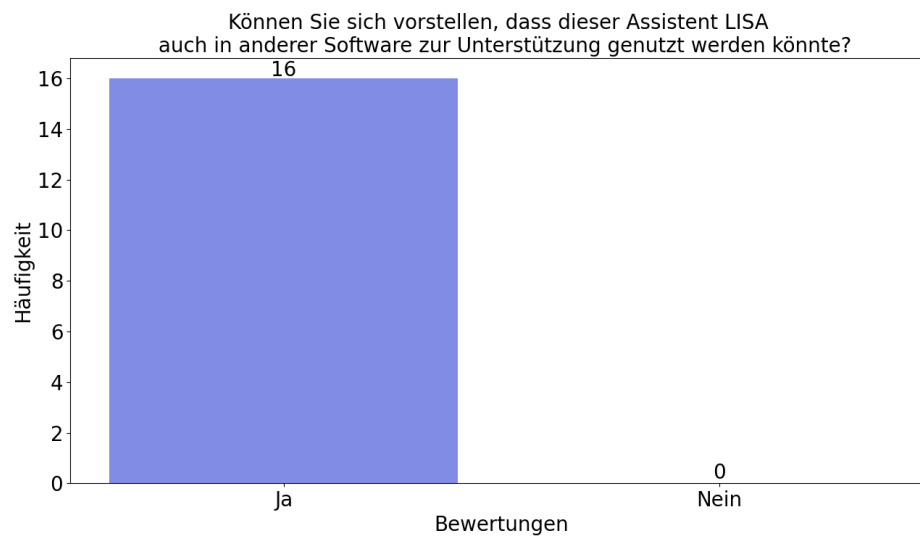


Abbildung 6.8: LISA wäre auch in anderer Software hilfreich.

In Abbildung 6.8 gaben von den Teilnehmenden die, beide Missionen erfolgreich abschließen konnte, alle an, dass LISA auch in anderer Software genutzt werden könne. Als möglich Software wurde ILIAS, Moodle, CAD, Datenbanken auf Userseiten, Lernprogrammen (z. B. Inklusionsarbeiten, SmartHome Dienste), andere Spiele, Entwicklungsumgebungen, Software für nicht technik-affine Personen, Bildverarbeitungssoftware, Office Tools, Grafikprogramme, und bei unbekannter Software zu Anfang genannt.

Ein Zitat aus den Antworten: „...eigentlich würde es fast allen Anwendungen gut tun, so einen Assistenten zu integrieren, am besten neben weiterer Gamifizierung. ... bei Software kann der Anfang doch sehr zermürend sein...“.

Bei den folgenden Ergebnissen wird viel auf die einzelnen Gruppen eingegangen. In Mission eins Gruppe eins ist eine Person mehr als in Mission zwei, da dieser Teilnehmende die Studie frühzeitig beendet hat.

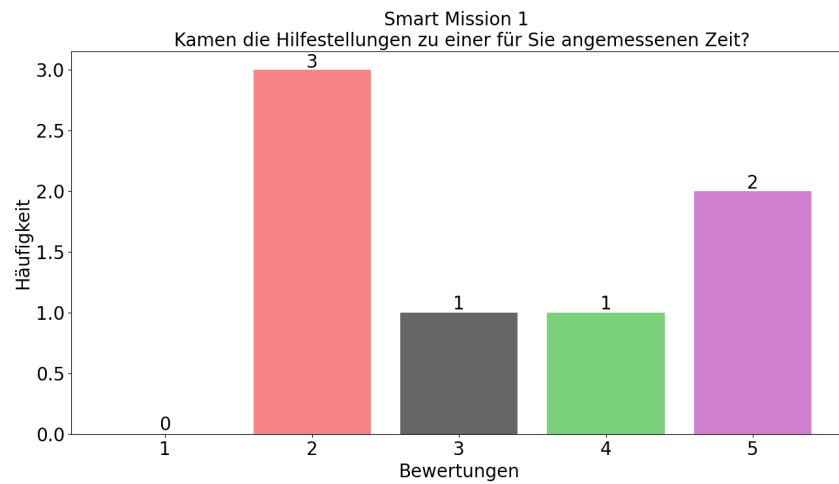


Abbildung 6.9: Das sind die Teilnehmenden die in Mission eins Unterstützung von smart LISA erhalten haben.

In Gruppe eins wurde wie in Abbildung 6.9 der Zeitpunkt der Hilfestellungen in Mission eins von den entsprechenden Teilnehmenden bewertet.



Abbildung 6.10: Das sind die Teilnehmenden die in Mission eins Unterstützung von der statischen LISA erhalten haben.

In Gruppe zwei wurde wie in Abbildung 6.10 der Zeitpunkt der Hilfestellungen in Mission eins von den entsprechenden Teilnehmenden bewertet.



Abbildung 6.11: Das sind die Teilnehmenden die in Mission zwei Unterstützung von der statischen LISA erhalten haben.

In Gruppe eins wurde wie in Abbildung 6.11 der Zeitpunkt der Hilfestellungen in Mission zwei von den entsprechenden Teilnehmenden bewertet.

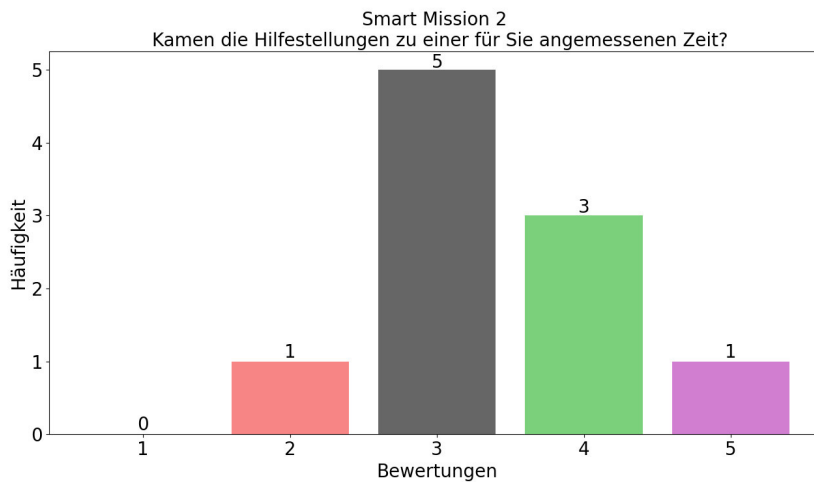


Abbildung 6.12: Das sind die Teilnehmenden die in Mission zwei Unterstützung von smart LISA erhalten haben.

In Gruppe zwei wurde wie in Abbildung 6.12 der Zeitpunkt der Hilfestellungen in Mission zwei von den entsprechenden Teilnehmenden bewertet.

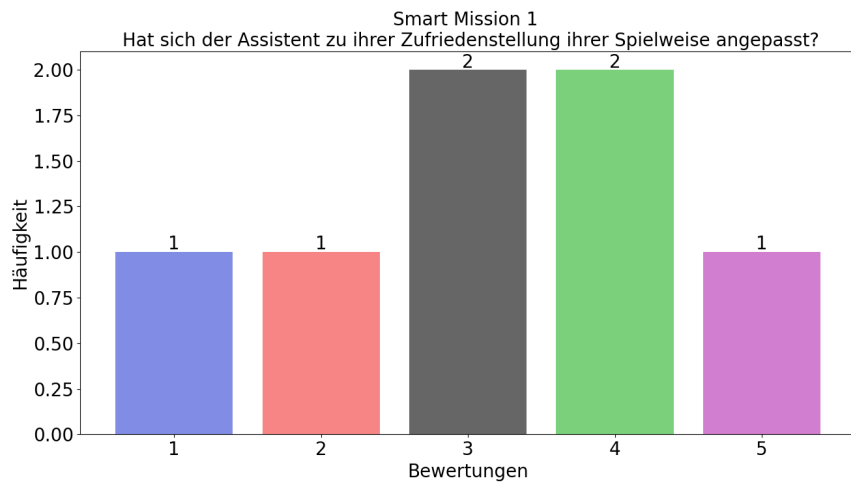


Abbildung 6.13: Das sind die Bewertungen der Teilnehmenden in Mission eins mit adaptiver Hilfestellung.

In Abbildung 6.13 ist zu sehen wie zufrieden die Teilnehmenden mit den gebotenen, adaptiven Hilfestellungen in Mission eins waren.

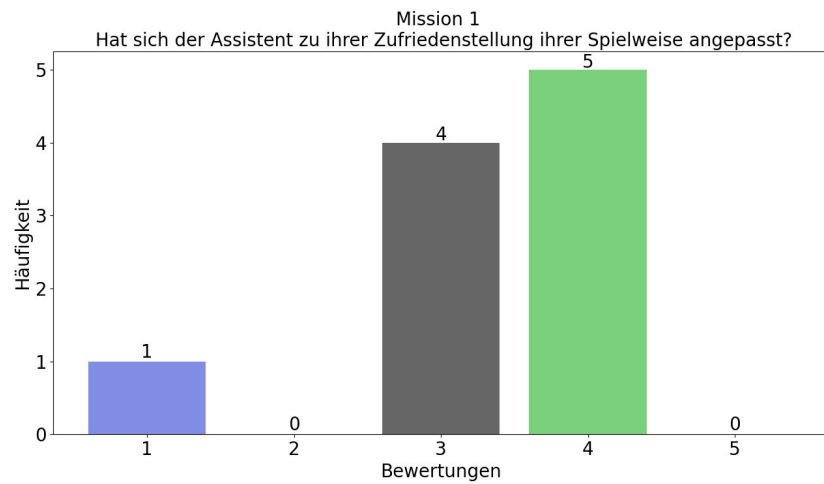


Abbildung 6.14: Das sind die Bewertungen der Teilnehmenden in Mission eins mit statische Hilfestellung.

In Abbildung 6.14 ist zu sehen wie zufrieden die Teilnehmenden mit den gebotenen, statischen Hilfestellungen in Mission eins waren.

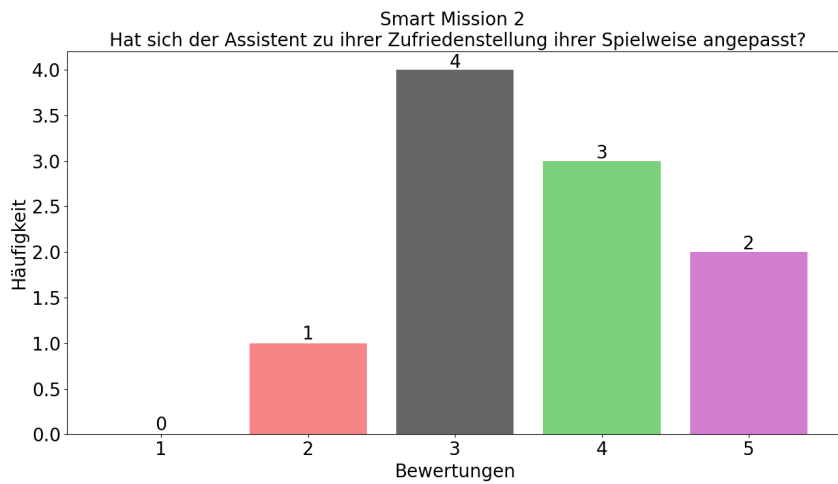


Abbildung 6.15: Das sind die Bewertungen der Teilnehmenden in Mission zwei mit adaptiver Hilfestellung.

In Abbildung 6.15 ist zu sehen wie zufrieden die Teilnehmenden mit den gebotenen, adaptiven Hilfestellungen in Mission zwei waren.

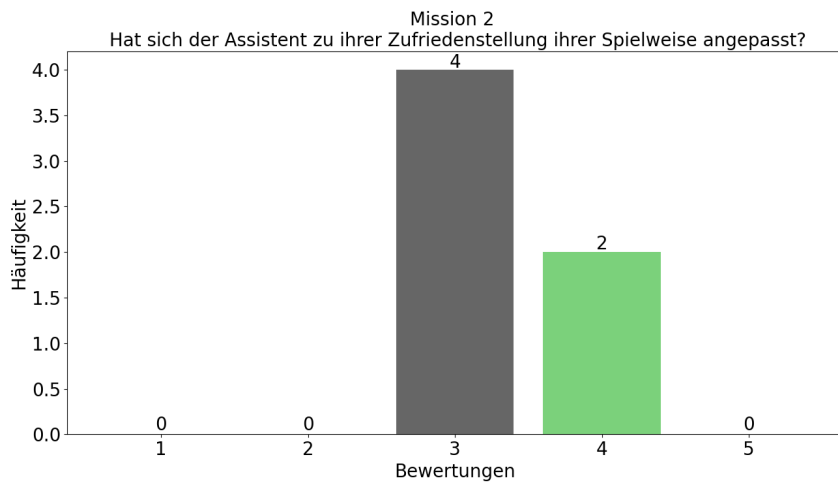


Abbildung 6.16: Das sind die Bewertungen der Teilnehmenden in Mission zwei mit statische Hilfestellung.

In Abbildung 6.16 ist zu sehen wie zufrieden die Teilnehmenden mit den gebotenen, statischen Hilfestellungen in Mission zwei waren.

Empfinden Sie den Assistenten LISA hilfreicher als statische Hilfestellungen?

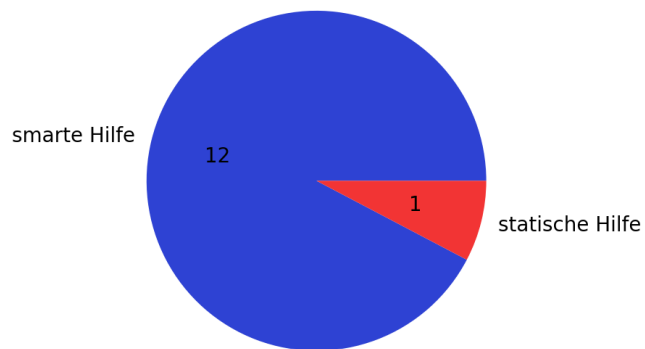


Abbildung 6.17: Das sind die Teilnehmenden die zwischen statischer und adaptiver Hilfe entscheiden.

Die Abbildung 6.17 zeigt dass zwölf von 16 Teilnehmenden LISA hilfreicher als statische Hilfe empfinden. Eine Person bevorzugt statische Hilfe und drei Teilnehmende legten sich nicht fest.

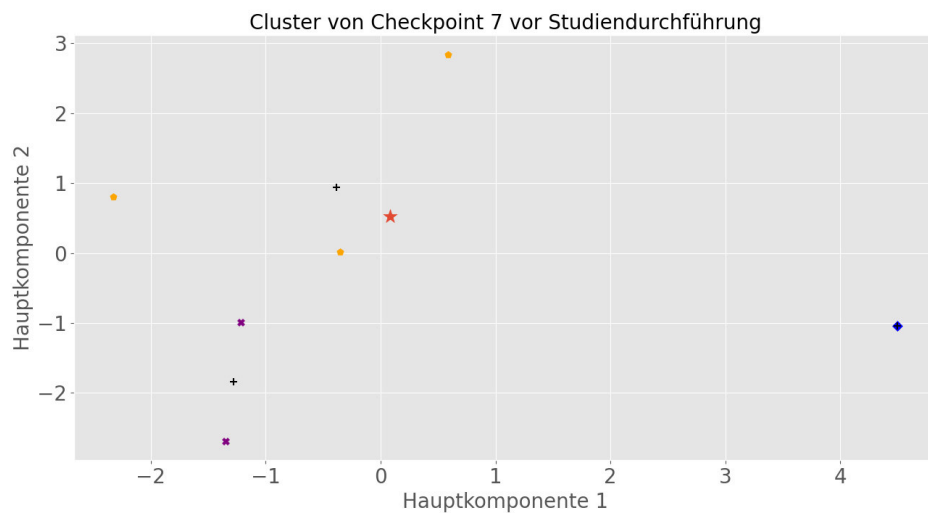


Abbildung 6.18: Das sind die Cluster vor der Studie mit selbst aufgenommenen Testdaten.

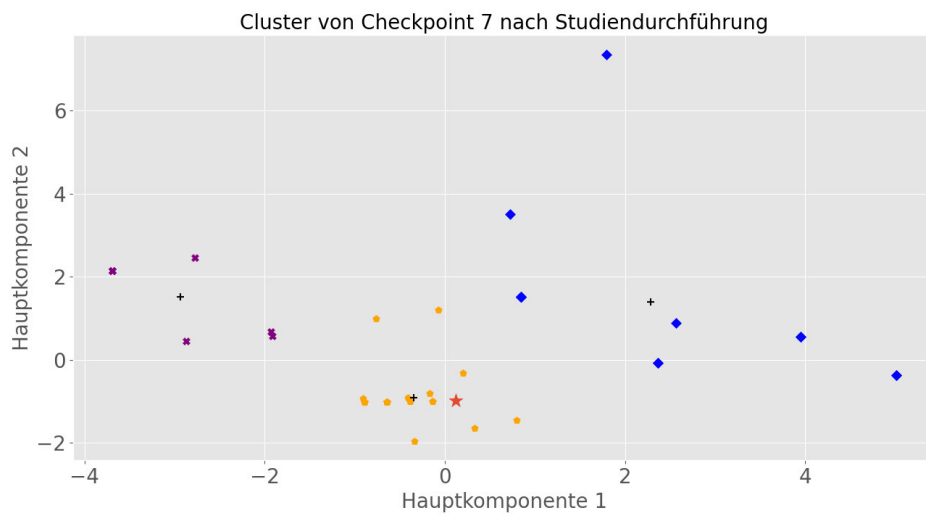


Abbildung 6.19: Das sind die Cluster nach der Studie mit den Testdaten und den Daten der Teilnehmenden.

In der Abbildung 6.18 ist das Cluster zu sehen, wie es vor der Studie berechnet wurde. In Abbildung 6.19 ist das Cluster mit den Daten aus der Studie berechnet. Der rote Stern ist dabei das IPM und die schwarzen Kreuze zeigen die Clusterzentren. Die Daten der einzelnen Cluster sind durch ihre Farbe und Form aufgeteilt. Diese Abbildungen wurde mithilfe der Principal Component Analysis (PCA) erstellt [Wold, Esbensen und Geladi, 1987]. Hierbei wurde PCA verwendet um die Dimensionalität der Daten zu reduzieren und sie Darstellen zu können.

6.5 Diskussion

Entsprechend der Ergebnisse der Studie, lassen sich einige Schlussfolgerungen zu dem Assistenten und den dessen Hilfestellungen ziehen. In dieser Arbeit sollten die folgenden Hypothesen beantwortet werden.

- LISA unterscheidet zwischen Situationen, in welchen der Nutzer Hilfe benötigt oder nicht und passt ihr Agieren darauf an.
- LISA führt den Assistenten erfolgreich durch die Missionen.
- Nutzer empfinden LISA als hilfreich.
- LISA kann als **smart** bezeichnet werden.

Um die Ergebnisse nachvollziehen zu können und damit neue Erkenntnisse erkennbar werden, werden in den folgenden Abschnitten die Ergebnisse interpretiert, die Begrenzungen und wie weiterführende Arbeiten aussehen könnte, werden erläutert.

Interpretation Laut den Ergebnissen in Abbildung 6.2 gaben mehr als die Hälfte an, dass sie den adaptiven Assistenten erkennen konnten, aber in 6.3 konnten weniger Personen korrekt schätzen, in welcher Mission der Assistent adaptiv agierte. Durch Abbildung 6.19 wird klar, dass LISA die Teilnehmenden einteilen konnte, jedoch war der Unterschied zu einem statischen Assistenten zu gering, als dass es von außen deutlich erkennbar wäre. Auch die Zufriedenheit mit der Anpassung an die Spielweise in 6.13 und 6.15, ist mittelmäßig ausgefallen. Dies ließe sich durch die Änderung der einzelnen Timer beheben.

Durch Abbildung 6.1 wird klar, dass LISA die Mehrheit der Teilnehmenden erfolgreich durch die Missionen führen konnte.

Abbildung 6.17 zeigt, dass die Mehrheit LISA hilfreicher empfunden haben, als statische Hilfe. Damit erweist sich LISA den Teilnehmenden gegenüber als nützlich.

In Abbildung 6.8 ist deutlich, dass der Assistent in unterschiedlichen Bereichen nutzbar wäre. Durch 6.7 wird klar, dass diese LISA als smart bezeichnet werden kann.

Durch einige Antworten, wie in Abbildung 6.12 ist erkennbar, dass die Zeit bis zu einer Hilfestellung für einige zu kurz gewählt war. Auch durch die Beliebtheit des Aufleuchtens in 6.4 und der Unbeliebtheit der Nachrichtenseiten in 6.5, ist erkennbar, dass die Mehrheit der Teilnehmenden eine passive Hilfestellung bevorzugen um nicht unterbrochen zu werden.

Einige würden gerne komplett auf die Nachrichtenseiten verzichten und nur die passive Aktionen von LISA, als automatisch angezeigte Hilfestellungen akzeptieren. Diese wollten die tiefgreifenden Informationen aktiv aufrufen. Zum Beispiel über einen Klick auf LISA.

Begrenzungen Die Zeit für eine Interaktion, wurde zu knapp gewählt. Für einen Überblick, wie lange ein Nutzer für solche Interaktionen benötigt, hätte durch viele Nutzerdaten oder eine weitere vorgehende Studie getestet werden müssen. Es wäre über den Rahmen dieser Arbeit hinausgegangen, eine weitere Studie im Vorfeld durchzuführen.

In dem Algorithmus zur Orientierung konnten nicht alle Sonderfälle berücksichtigt werden. Ein Beispiel dafür wäre das ein Nutzer eine Tätigkeit überspringen könnte, da er die daraus folgenden Informationen bereits kennt. Bei diesem Szenario würde der Assistent den Nutzer auch nach Weiterführung der korrekten Interaktionen zu dieser übersprungenen Information führen. Es wäre möglich gewesen, den Algorithmus umzuordnen, indem einzelne Tätigkeiten als unwichtig eingestuft werden und somit übersprungen werden können.

Weiterführende Untersuchungen Eine weiterführende Untersuchung könnte sein, LISA in ein weiteres Serious Game oder in eine andere Software einzubauen. Es könnte ein Recommender System, ähnlich wie bei Streaming- oder Shopping-Seiten eingebaut werden, um die Hilfsgrade vorherzusehen, indem ähnliche Nutzer als Richtwert gelten.

Es könnte eine weitere KI eingebaut werden, welche entscheidet ob der Nutzer gerade eine passive Hilfestellung bevorzugt oder sich mehr konkrete Informationen wünscht.

In dieser Arbeit sollte der Assistent LISA „smart“ gemacht werden. Des Weiteren sollte der Assistent lernen, einzelne Nutzer voneinander zu unterscheiden, welche Hilfe benötigen oder nicht. Dafür musste der Elkan-Algorithmus verwendet und die gewählte Metrik war die euklidische. Um zu gewährleisten, dass die Nutzenden von LISA wieder auf den richtigen Weg geleitet werden, wurde der Bewertungsalgorithmus verwendet. Dieser dient zur Bewertung für das Clustering, als auch für die Berechnung der nächsten erforderlichen Hilfestellung. In dieser Arbeit wurde erfolgreich ein funktionsfähiger Assistent entwickelt, welcher Nutzer nach ihrem Verhalten einteilt und daraufhin unterstützt. Da in den Bewertungen LISA als „smart“ und hilfreich bezeichnet wurde, kann der Assistent smart LISA genannt werden. Durch die hohe Abschlussrate und die genannten Beispiele, in welchen Anwendungen LISA noch verwendet werden könnte, wird deutlich dass sie zum erfolgreichen Abschließen von festen Zielen eine geeignete Unterstützung ist. Durch die in den Ergebnissen gezeigten Cluster wird kenntlich, dass der Assistent die Teilnehmenden erfolgreich über ihre Interaktionen bewertet und einteilt. Durch die Bewertungen der Adaptivität und dem gewählten Zeitpunkt einzelner Hilfestellungen, wird klar dass Parameter wie der Timer verändert werden müssten, um einen ausgereiften Assistenten zu schaffen, welcher passendere Hilfestellungen bietet. In der Diskussion wurde bereits erwähnt, dass es Sonderfälle in Spielen gibt, die nicht alle berücksichtigt werden konnten. Es gibt noch weitere solcher Ausnahmefälle, die bei einer Weiterentwicklung des Assistenten berücksichtigt werden sollten.

Ein leistungsfähiger Server, welcher mehrere Berechnungen parallel vollziehen kann, könnte angewandt werden, damit die Antworten und Einteilungen schneller berechnet und gesendet werden können. Der Assistent könnte in weiteren Spielen eingebaut und getestet werden. In einer Weiterentwicklung könnte der Assistent mit weiterer Software kommunizieren, die das Verhalten von Nutzern beurteilt. In einer Weiterentwicklung von smart LISA könnte die Funktion verbessert werden, in welcher der Nutzer aktiv um Hilfe bittet. Die in Szenen eingebaute LISA, könnte als Kontrolle für den berechneten Hilfeegrad dienen. So könnte der Nutzer den Hilfeegrad bis zu einem gewissen Maß selbst wählen. Dadurch könnte der Nutzer sich mehr oder weniger Zeit bis zur nächsten Hilfestellung geben und so seinen Assistenten individuell auf sich selbst anpassen.

Einen perfekten adaptiven Assistenten zu konstruieren ist aus heutiger Sicht nahezu unmöglich, da jeder Anwender diese Assistenz individuell wahrnimmt und eigene Präferenzen hat. Dennoch ist es möglich, mit passiven Hilfestellungen eine breite Masse an Personen zu ihrer Zufriedenheit zu unterstützen.

LITERATUR

- Advanced Distributed Learning (2020). *GitHub: Unity-xAPI-Wrapper*. URL: <https://github.com/adlnet/Unity-xAPI-Wrapper> (besucht am 30. 04. 2021).
- Alonso-Fernandez, Cristina u. a. (2017). „Systematizing game learning analytics for serious games“. In: *2017 IEEE global engineering education conference (EDUCON)*. IEEE, S. 1111–1118.
- Atorf, Daniel, Ehm Kannegieser und Wolfgang Roller (2018). „Balancing realism and engagement for a serious game in the domain of remote sensing“. In: *International Conference on Games and Learning Alliance*. Springer, S. 146–156.
- Atorf, Daniel, Ehm Kannegieser und Wolfgang Roller (2019). „Study on Enhancing Learnability of a Serious Game by Implementing a Pedagogical Agent“. In: *International Conference on Games and Learning Alliance*. Springer, S. 158–168.
- Bayer, Michael (2021). *SQLAlchemy: Overview*. URL: <https://docs.sqlalchemy.org/en/14/intro.html> (besucht am 30. 04. 2021).
- Bishop, Christopher M (2006). *Pattern recognition and machine learning*. springer.
- Böhm, Christian u. a. (2004). „Immer größere und komplexere Datenmengen: Herausforderungen für Clustering-Algorithmen.“ In: *Datenbank-Spektrum* 9, S. 11–17.
- Chesneau, Benoit (2021). *Gunicorn.org*. URL: <https://gunicorn.org/> (besucht am 30. 04. 2021).
- Cournapeaul, David (2021). *Scikit-Learn.org*. URL: <https://scikit-learn.org/stable/> (besucht am 30. 04. 2021).
- Danz, Marcel (2019). *Enhancing User Experience in a Serious Game by Revising the UI And Implementing a Pedagogical Agent for Improving Learnability*. eapartment of Informatics, Karlsruhe Institute of Technology, Masterthesis.
- Dazer, Michael (2012). *RESTful APIs-Eine Übersicht*.
- Dillig, Marlene (2020). *Optimierung des Game Designs und intelligente Weiterentwicklung des pädagogischen Agenten in einem Serious Game Framework*. Hochschule Furtwangen University, Masterthesis.
- DIN SPEC 91380:2018-06 (2018). „Serious Games Metadata Format“. In: DOI: [10.31030/2853739](https://doi.org/10.31030/2853739).
- Elkan, Charles (2003). „Using the triangle inequality to accelerate k-means“. In: *Proceedings of the 20th international conference on Machine Learning (ICML-03)*, S. 147–153.
- Fraunhofer IOSB (2020). *Fraunhofer IOSB: Lost Earth 2307*. URL: <https://www.iosb.fraunhofer.de/de/projekte-produkte/lost-earth-2307.html> (besucht am 24. 04. 2021).
- Fraunhofer IOSB (2021a). *Lost Earth 2309*. Serious Game.

- Fraunhofer IOSB (2021b). *Fraunhofer IOSB: Standort Karlsruhe*. URL: <https://www.iosb.fraunhofer.de/de/ueber-uns/standorte/anfahrt-standort-karlsruhe.html> (besucht am 24. 04. 2021).
- Frochte, Jörg (2019). *Maschinelles Lernen: Grundlagen und Algorithmen in Python*. Carl Hanser Verlag GmbH Co KG.
- Hammel, Katharina, Cornelia Schade und Bjorn Adelsberg (2019). „Workshop adaptive Serious Games: Workflow zur Generierung einer Spielidee“. In: *Mensch und Computer 2019-Workshopband*.
- Hunter, John D. (2021). *Matplotlib.org*. URL: <https://matplotlib.org/stable/index.html> (besucht am 30. 04. 2021).
- Kapitanova, Krasimira und S Son (2012). „Machine learning basics“. In: *Intelligent Sensor Networks*. CRC Press, S. 3–29.
- Kuntze, Daniel (2007). „Untersuchung von Clustering-Algorithmen für die Kullback-Leibler-Divergenz“. In: *Fakultät für Elektrotechnik*. Informatik und Mathematik der Universität, Diplomarbeit.
- Lobstedt, Andreas (2020). *Smarte digitale Lernspiele mit interoperabler Kopplung und Adaptivität*. Fakultät für Informatik, Bachelorthesis.
- Microsoft (2021). *Visual Studio 2019*. URL: <https://visualstudio.microsoft.com/de/vs/> (besucht am 30. 04. 2021).
- Nouira, Azer, Lilia Cheniti-Belcadhi und Rafik Braham (2018). „An enhanced xapi data model supporting assessment analytics“. In: *Procedia Computer Science* 126, S. 566–575.
- Oliphant, Travis (2021). *Numpy.org*. URL: <https://numpy.org/> (besucht am 30. 04. 2021).
- parentElement (2017). *Unity Asset Store: JSON.Net for Unity*. URL: <https://assetstore.unity.com/packages/tools/input-management/json-net-for-unity-11347> (besucht am 30. 04. 2021).
- Prensky, Marc (2003). „Digital game-based learning“. In: *Computers in Entertainment (CIE)* 1.1, S. 21–21.
- Printer Care Service GmbH (2020). *PrinterCare: Was ist das EVA-Prinzip?* URL: <https://www.printer-care.de/de/drucker-ratgeber/was-ist-das-eva-prinzip> (besucht am 30. 04. 2021).
- Ronacher, Armin (2021). *Flask: Welcome to Flask*. URL: <https://flask.palletsprojects.com/en/1.1.x/#> (besucht am 30. 04. 2021).
- Russell, Stuart Jonathan und Peter Norvig (2010). *Artificial Intelligence: A Modern Approach (3rd Edition)*. Prentice Hall.
- Salesforce (2021). *Heroku.com*. URL: <https://www.heroku.com/> (besucht am 30. 04. 2021).
- Serrano-Laguna, Ángel u. a. (2017). „Applying standards to systematize learning analytics in serious games“. In: *Computer Standards & Interfaces* 50, S. 116–123.
- Sim, Ian (2019). „The 6 Key Benefits of Being an API-Centric Organization“. In:
- Streicher, Alexander, Sebastian Leidig und Wolfgang Roller (2018). „Eye-tracking for user attention evaluation in adaptive serious games“. In: *European Conference on Technology Enhanced Learning*. Springer, S. 583–586.
- Streicher, Alexander und Jan D. Smeddinck (2016). „Personalized and adaptive serious games“. In: *Entertainment computing and serious games: International GI-Dagstuhl seminar 15283, dagstuhl castle, germany, july 5-10, 2015, revised selected papers*. Hrsg. von Ralf Dörner u. a.

- ZSCC: NoCitationData[s0]. Cham: Springer International Publishing, S. 332–377. ISBN: 978-3-319-46152-6. DOI: [10.1007/978-3-319-46152-6](https://doi.org/10.1007/978-3-319-46152-6). URL: <https://doi.org/10.1007/978-3-319-46152-6>.
- Susi, Tarja, Mikael Johannesson und Per Backlund (2007). *Serious games: An overview*. Institutionen för kommunikation och information.
- Unity (2019). *Unity Echtzeit-Entwicklungsplattform | 3D-, 2D-VR- & AR-Engine*. URL: <https://unity.com/de> (besucht am 09. 02. 2021).
- Unity Technologies (2021). *Unity Manual: Getting started with WebGL development*. URL: <https://docs.unity3d.com/Manual/webgl-gettingstarted.html> (besucht am 30. 04. 2021).
- Wallhoff, Frank, Markus Ablaßmeier u. a. (2007). „Adaptive human-machine interfaces in cognitive production environments“. In: *2007 IEEE international conference on multimedia and expo*. IEEE, S. 2246–2249.
- Wallhoff, Frank, Jan Paul Vox und Tobias Theuerkauff (2019). „Assistenz-und Servicerobotik–die Gestaltung der Mensch-Maschine-Schnittstelle als Grundlage des Anwendungserfolgs“. In: *Gesundheit digital*. Springer, S. 99–122.
- Wold, Svante, Kim Esbensen und Paul Geladi (1987). „Principal component analysis“. In: *Chemometrics and intelligent laboratory systems* 2.1-3, S. 37–52.
- Xiao, Jun u. a. (2020). „Research on Adaptive Learning Prediction Based on XAPI“. In: *International Journal of Information and Education Technology* 10.9.

EIGENSTÄNDIGKEITSERKLÄRUNG

Erklärung gemäß §1 (5) Allgemeiner Teil (Teil A) der Prüfungsordnung für die Bachelor-Studiengänge (BPO) an der Jade Hochschule Wilhelmshaven/Oldenburg/Elsfleth in der Fassung vom 21.10.2014 (Verköndungsblatt Nr. 56/2014 der Jade Hochschule)

Diese Bachelorarbeit ist eine Einzelarbeit. Ich erkläre hiermit, diese Bachelorarbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben.

Schopfheim 30.04.2021



Ort, Datum, Unterschrift (Geraldine Sutter (6021014))