

# Anwendung von Regressionsmodellen für adaptive digitale Lernspiele in der Bildauswertung

## BACHELORARBEIT

KIT – KARLSRUHER INSTITUT FÜR TECHNOLOGIE  
FRAUNHOFER IOSB – FRAUNHOFER-INSTITUT FÜR OPTRONIK,  
SYSTEMTECHNIK UND BILDAUSWERTUNG

**German Pustovojtovskij**

01. Februar 2021

Verantwortlicher Betreuer:	Prof. Dr.-Ing. Jürgen Beyerer
Verantwortlicher Zweitbetreuer:	Prof. Dr.-Ing. Thomas Längle
Betreuender Mitarbeiter:	Dipl.-Inf. Alexander Streicher



## Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die Arbeit selbständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht habe und die Satzung des Karlsruher Instituts für Technologie zur Sicherung guter wissenschaftlicher Praxis in der gültigen Fassung beachtet habe.

Karlsruhe, den 01. Februar 2021

---

(German Pustovojtovskij)





## Zusammenfassung

In der heutigen Zeit wird durch die COVID-19 Pandemie die Wichtigkeit von digitalem Lernen besonders deutlich aufgezeigt. Solche Krisen schließen offline-Lernmethoden, wie sie in Klassenzimmern oder Vorlesungssälen üblich sind, aus. E-Learning und digitale Lernspiele bieten gute Möglichkeiten, die Bildung der Menschen trotzdem weiter zu fördern.

Der Lernfortschritt kann mit Hilfe von digitalen Lernspielen verbessert werden, indem der Spieler im sogenannten Flow-Kanal gehalten wird. Um das zu erreichen, muss für eine Balance zwischen Über- und Unterforderung gesorgt werden. Dazu wurde am Fraunhofer IOSB das Adaptivitäts-Framework E-Learning Artificial Intelligence (ELAI) entwickelt.

Dieser Arbeit untersucht, wie der aktuelle Wissensstand des Nutzers mittels Regressionsmodellen bestimmt werden kann. Dabei werden die drei Modelle Knowledge Tracing (KT), Learning Factor Analysis (LFA) und Performance Factor Analysis (PFA) aufbauend auf dem Stand der Forschung und Technik im Hinblick auf ihre Einsetzbarkeit für die Berechnung eines sogenannten „Performance Scores“ genauer überprüft.

Die PFA zeigt sich als gut geeignete Methode zur Datenanalyse für das ELAI Framework. Aufbauend auf den entwickelten Konzepten, wird eine Modellierung für PFA auf Serious Games (SG) erstellt und anschließend im SG Streamlined Lost Earth (SLE) mit einem angepassten ELAI Interface implementiert.

Eine Evaluation in Form einer Nutzerstudie zeigt die Mehrwerte der Implementierung von PFA anhand eines Anwendungsszenarios auf. Abschließend werden noch mögliche Erweiterungen durch Anwendung der Selbsteinschätzung und andere Ergänzungen diskutiert.



## Abstract

At the present time, the COVID-19 pandemic highlights the importance of digital learning. Such crises flipped out offline learning methods which have long been common in classrooms or lecture halls. E-learning and digital learning games offer good opportunities to further people's education.

In order to improve learning progress with the help of digital learning games, it is important to keep the player in the so-called flow channel. To achieve this, a balance between frustration and boredom must be ensured. For this purpose, the adaptivity framework E-Learning Artificial Intelligence (ELAI) was developed at Fraunhofer IOSB.

By means of regression models, this work investigates how the user's current level of knowledge can be determined. In doing so, the three models Knowledge Tracing (KT), Learning Factor Analysis (LFA) and Performance Factor Analysis (PFA) are reviewed in more detail based on state of the art and technology in order to calculate a so-called "performance score".

The PFA is shown to be a well-suited method for data analysis for the ELAI framework. Building on the concepts developed, a modeling for PFA on Serious Games (SG) is created and then used in the SG Streamlined Lost Earth (SLE) with a customized ELAI interface implemented.

An evaluation in the form of a user study shows the additional benefits of the implementation of PFA based on an application scenario. Finally, possible extensions by applying self-assessment and other factors are discussed.



---

# Inhaltsverzeichnis

<b>Zusammenfassung</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Problemstellung . . . . .	2
1.2 Zielsetzung . . . . .	3
1.3 Lösungsansatz . . . . .	3
1.4 Aufbau und Gliederung der Arbeit . . . . .	4
<b>2 Stand der Forschung und Technik</b>	<b>7</b>
2.1 Adaptive Lernspiele und Serious Games . . . . .	7
2.2 Lernfortschrittmodellierung bei Serious Games . . . . .	10
<b>3 Grundlegende Konzepte</b>	<b>15</b>
3.1 Adaptive Lernspiele und Serious Games . . . . .	15
3.1.1 Digital Games und Serious Games . . . . .	15
3.1.2 Lernfortschrittoptimierung durch Adaptivität . . . . .	16
3.2 Beispiel für Serious Games: Lost Earth 2307 . . . . .	20
3.3 E-Learning A.I (ELAI) . . . . .	21
3.4 Experience Application Programming Interface (xAPI) . . . . .	23
3.5 Regressionsmodelle . . . . .	25
3.5.1 Logistisches Regessionsmodell . . . . .	25
3.5.2 Knowledge Tracing (KT) . . . . .	26
3.5.3 Learning Factor Analysis (LFA) . . . . .	29
3.5.4 Performance Factor Analysis (PFA) . . . . .	33
<b>4 Anwendung von Regressionsmodellen auf Serious Games</b>	<b>39</b>
4.1 Entwurf des adaptiven Assistenten . . . . .	40
4.2 Performance Factor Analysis Modellierung . . . . .	42

---

4.3	Entwurf des PFA-Moduls . . . . .	45
<b>5</b>	<b>Implementierung des Regressionsmodells</b>	<b>49</b>
5.1	Erster Prototyp . . . . .	49
5.1.1	ExpressJS Web-Framework . . . . .	49
5.1.2	JavaScript Library für Experience API: TinCanJS . . . . .	50
5.1.3	Berechnung von Performance Factor Analysis . . . . .	51
5.2	Implementierung des adaptiven Assistenten . . . . .	51
5.3	Implementierung des PFA-Moduls . . . . .	54
5.4	Diskussion . . . . .	56
<b>6</b>	<b>Evaluation</b>	<b>59</b>
6.1	Anwendungsszenario . . . . .	59
6.2	Nutzerstudie . . . . .	61
6.2.1	Hypothese . . . . .	61
6.2.2	Aufbau . . . . .	62
6.2.3	Ergebnisse . . . . .	64
<b>7</b>	<b>Fazit und Ausblick</b>	<b>69</b>
	<b>Literatur</b>	<b>71</b>
	<b>Glossar</b>	<b>75</b>

# 1 Einführung

Aktuell wird aufgrund der COVID-19 Pandemie allen Ausbildungsinstitutionen, wie z.B. Schulen und Universitäten, die Wichtigkeit von digitalem Lernen aufgezeigt [Radha et al. 2020]. E-Learning ist eine effektive Lehrmethode, die zu Zeiten des Lockdowns eine Möglichkeit bietet weiterhin Wissen zu vermitteln.

Um die Motivation und die Lernzielerreichung der Nutzer effektiv und effizient in Hinblick auf den Lernerfolg auszurichten, müssen sich die digitalen Lernspiele stärker an das Nutzungs- bzw. Lernverhalten anpassen. Ein wichtiger Faktor für den Lernfortschritt ist, ob die Nutzer im sogenannten Flow-Kanal sind [Perttula et al. 2017]. Im Flow-Kanal grenzt der Nutzer alle Emotionen und Gedanken aus, die nicht mit der Aktivität zu tun haben, um in der Tätigkeit vollkommen aufzugehen. Dabei wird der Nutzer in der optimalen Balance zwischen Frustration und Langeweile gehalten. In Abbildung 1.1 erkennt man, dass ein mögliches Reaktionsmodell für ein adaptives System für digitale Lernspiele demnach darin bestehen kann, den Nutzer immer wieder in den Flow-Kanal zu führen.

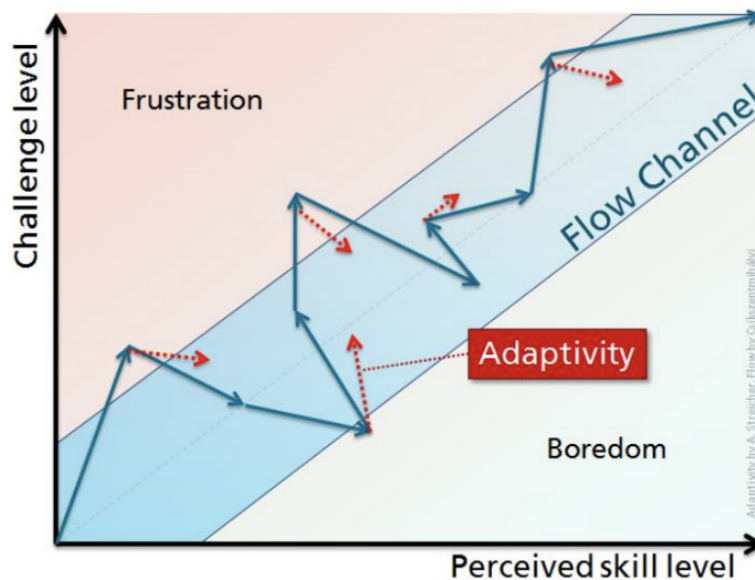


Abbildung 1.1: Flow-Kanal [Csikszentmihalyi 2009]

Digitales Lernen ist, bereits seit einigen Jahren, ein wichtiges Forschungsthema am Fraunhofer IOSB. Das E-Learning Artificial Intelligence (ELAI) (Abschnitt 3.3) Framework wurde entwickelt um die Adaptivität in sogenannten Serious Games (SG) zu steigern [Streicher und Roller 2017]. Dabei ist ein SG ein digitales Spiel, das zur Unterhaltung dient, aber zusätzlich ein weiteres Ziel, wie z.B. Sport oder Wissenslehre, verfolgt [Dörner et al. 2016, S.2].

Adaptivität (Abschnitt 3.1.2) in dieser Arbeit, ist die Adaptierung von SG an die Bedürfnisse der Nutzer. Exemplarisch durch Interaktionsmechanismen, bei denen der Inhalt und die Empfehlungen von einer automatischen intelligenten Tutor-Komponente personalisiert werden. Intelligente Tutor-Komponenten zeichnen das Verhalten des Nutzers auf und können Lernhinweise als Hilfestellung anzeigen. Um das zu realisieren, gibt es viele verschiedene Möglichkeiten: Beispielsweise wird in dieser Arbeit ein intelligenter Tutor erweitert, der dem Nutzer Hinweise anzeigt, falls der Nutzer auf die Hilfe angewiesen ist. Um herauszufinden, wann so ein Interface eingreifen muss, werden Maschinelles Lernen und Techniken zur Datenanalyse verwendet, um den bestmöglichen Einfluss auf die Personalisierung zu bewirken [Streicher und Roller 2015, S. 1]. Eine mögliche Methode, um Daten zu analysieren, ist die Verwendung von Regressionsmodellen. Modelle die für die Datenanalyse in SG oft verwendet werden [Pavlik Jr et al. 2009], sind Knowledge Tracing und Learning Factor Analysis. Ein daraus neu entstandenes Modell lautet Performance Factor Analysis.

## 1.1 Problemstellung

Als übergeordnete Ziel soll gezeigt werden wie die Adaptivität für Lernerfolgsoptimierung in Serious Games verbessert werden kann. Konkreter wird in dieser Thesis das PFA Modell angewendet, um den aktuellen Lernfortschritt des Nutzers berechnen zu können. Durch bereits vorhandene wissenschaftliche Artikel [Pavlik Jr et al. 2009; Gong, Beck, et al. 2010] wurde eine mögliche Verbesserung der Lernerfolgsoptimierung durch Performance Factor Analysis (PFA) vorgestellt. Performance Factor Analysis ist ein logistisches Regressionmodell, das zur Berechnung des Lernfortschritts eines bestimmten Nutzers verwendet wird. Daraus entstehen folgende Fragen, die in dieser Arbeit behandelt werden:

- Wie kann die Lernfortschrittperformance mittels PFA modelliert werden?
  - Wie sieht so eine Modellierung für SG allgemein aus?
  - Wie sieht so eine Modellierung für ein konkretes Anwendungsbeispiel aus?
- Kann das PFA Modell auf SG angewendet werden?
  - Wie sieht so eine Anwendung für ein konkretes Anwendungsbeispiel aus?



Das Anwendungsbeispiel dieser Arbeit ist das Serious Games Streamlined Lost Earth (SLE) [*Streamlined Lost Earth 2021*]. Das Spiel dient sowohl der Unterhaltung, als auch dazu den Nutzer im Bereich der Bildauswertung zu trainieren.

## 1.2 Zielsetzung

Zu zeigen ist, wieso PFA als Regressionsmodell bevorzugt wird und welche Vorteile PFA gegenüber anderen Modellen, wie z.B. Knowledge Tracing (KT) (Abschnitt 3.5.2) und Learning Factor Analysis (LFA) (Abschnitt 3.5.3), hat. Dazu muss PFA untersucht und die Vor- und Nachteile erläutert werden (Abschnitt 3.5.4). Als direkter Vergleich wird oft das Knowledge Tracing Modell gewählt, da es bisher das beliebteste Modell ist, welches zur Steigerung des Lernerfolges verwendet wurde [Schultz et al. 2014]. Weiterhin müssen auch die Nachteile und Risiken des PFA Modells in Betracht gezogen und abgewogen werden. Anschließend muss ein Modell für die Performance Berechnung konzipiert werden. Es ist zu klären, inwieweit die PFA für die ELAI und ein Fraunhofer IOSB Lernspiel, etwa Lost Earth 2307 [*Lost Earth 2307 - Fraunhofer IOSB 2021*], angewendet werden kann.

Die Anwendung von PFA auf SLE wird mit einer Evaluation untersucht (Abschnitt 6). Die Mehrwerte der PFA Modellierung werden aufgezeigt und die Anwendung von PFA auf Adaptivität für Serious Games wird in einem Anwendungsszenario demonstriert (Abschnitt 6).

## 1.3 Lösungsansatz

Zur Ermittlung der Nutzerperformance kann die auf der Learning Factor Analysis [Cen, K. Koe-dinger, et al. 2006] basierte Performance Factor Analysis genutzt werden [Pavlik Jr et al. 2009]. Dabei wurde das LFA Modell gewählt, weil es, anders als das Knowledge Tracing (Abschnitt 3.5.2), Lernfortschrittmodellierung erlaubt trotz Abfragen mit mehreren Wissenskomponente (WK) (Abschnitt 3.5.2). Um aus dem LFA das PFA Modell zu konstruieren, muss das LFA Modell die Nutzerperformance nicht mehr allgemein für alle Nutzer, sondern für jeden Einzelnen, ermitteln. Mittels PFA kann ein aktueller „Performance Score“ berechnet werden, der anzeigt, wie der Nutzer die gestellte Frage zu einem bestimmten Themenbereich beantworten wird. ELAI (Abschnitt 3.3) kann an dieser Berechnung anknüpfen um das SG an die Bedürfnisse des Nutzers anzupassen (Adaptivität, Abschnitt 3.1.2). Zusätzlich können durch Beobachtungen der Nutzerinteraktionen und Berechnung der Lernfortschrittperformance in SG passende Zeitpunkte berechnet werden, um ein adaptives System aktiv werden zu lassen. Die mittels der PFA berechnete, akkumulierte Lernfortschrittperformance kann für die Klassifikation der

Nutzerlernperformance in der ELAI genutzt werden, um darauf basierend Entscheidungen für den Adaptivitätszeitpunkt abzuleiten. Um die Güte für den „Performance Score“ zu bewerten, wird in der Evaluation eine Korrelation zwischen der Selbsteinschätzung des Nutzers und dem berechneten Wert hergestellt. Natürlich gibt es auch weitere Einflüsse und Faktoren, wie beispielsweise die Zeit, die zwischen den Lerneinheiten vergeht. Dabei kann ein Nutzer Wissen nach bestimmter Zeit wieder vergessen. Jedoch beschränkt sich diese Arbeit auf die zuvor genannten Faktoren. Die hier nicht berücksichtigten Faktoren, wie die Zeit zwischen Lerneinheiten, könnten Themen zukünftiger Arbeiten werden.

## **1.4 Aufbau und Gliederung der Arbeit**

In den folgenden Absätzen wird der Aufbau der Arbeit erläutert.

### **Einleitung**

Zum Einstieg in diese Arbeit wird beschrieben, welche Motivation zur Behandlung der „Anwendung von Regressionsmodellen für adaptive digitale Lernspiele in der Bildauswertung“ geführt hat. Anschließend werden Problemstellung, Zielsetzung und Lösungsansatz dieser Bachelorarbeit vorgestellt.

### **Stand der Technik**

Hier wird der aktuelle Stand der Forschung und Technik zu den Themen SG, Adaptivität und Regressionsmodelle dargestellt.

### **Grundlagen**

In diesem Kapitel werden einige grundlegende Begriffe und Konzepte erklärt, die in dieser Arbeit verwendet werden. Dazu gehören die Erklärungen von adaptiven Lernspielen und Serious Games, dem Fraunhofer IOSB ELAI-Framework, Experience API (xAPI) und den Regressionsmodellen Knowledge Tracing, Learning Factor Analysis und Performance Factor Analysis.

### **Anwendung von Regressionsmodellen auf Serious Games**

Kapitel 4 stellt die Modellierung der PFA mit möglicher Parameterbelegungen vor. Zusätzlich werden die grundlegende Konzeption des adaptiven Assistenten LISA und des PFA-Moduls dargestellt.

### **Implementierung des Regressionsmodells**

Hier wird die Implementierung des PFA Prototypen dieser Arbeit behandelt. Weiterhin werden die Konzepte aus Kapitel 4 realisiert. Abschließend werden die benötigten Anpassungen am PFA-Modell diskutiert, um die Berechnung für SLE zu gewährleisten.

### **Evaluation**

In der Evaluation wird das Anwendungsszenario dieser Thesis vorgestellt und anschließend die Ergebnisse der durchgeführten Nutzerstudie ausgewertet.

### **Fazit und Ausblick**

Im letzte Kapitel werden die Ergebnisse der gesamten Arbeit zusammengefasst. Abschließend wird ein Ausblick für zukünftige Arbeiten geliefert. Dazu gehören andere Möglichkeiten der Evaluation der Ergebnisse aus der Nutzerstudie und mögliche Erweiterungen der angewandten PFA Formel durch die Selbsteinschätzung.



## 2 Stand der Forschung und Technik

In diesem Kapitel wird der aktuelle Stand der Technik im Hinblick auf die Anwendung von Regressionsmodellen im Bereich e-Learning behandelt. Zunächst werden bereits vorhandene Publikationen und Forschungsarbeiten zum Thema adaptive Lernspiele und Serious Games vorgestellt. Dann folgt ein Überblick über die Lernfortschrittmodellierung bei Serious Games.

### 2.1 Adaptive Lernspiele und Serious Games

Adaptive Lernspiele (auch tutoring systems) und Serious Games sind heutzutage sehr beliebt [Gong, Beck, et al. 2010]. Dabei stehen die Verbesserung des Lernerfolgs und die Steigerung der Motivation im Vordergrund. Beide Aspekte profitieren von der Adaptivität an den Nutzer [Gong, Beck, et al. 2010].

#### **Serious Games**

Menschen lieben es, unterhalten zu werden und Spiele zu spielen [Dörner et al. 2016, S. 2]. Aus diesen Gründen ist es verständlich, wieso der Computer ebenfalls zum Spielen verwendet wird. Spiele, die eine beliebige Rechenmaschine verwenden (z.B. Smartphone, Laptop, Gaming Konsole) nennt man digitale Spiele [Dörner et al. 2016, S. 2]. Abgesehen von der Unterhaltung können Spiele die Intention haben einen seriösen Zweck zu erfüllen. Beispielsweise dient das Spiel Basketball nicht nur ausschließlich der Unterhaltung, sondern auch der Fitness und der Kommunikation des Spielers. Sogenannte Serious Games (Abbildung 2.1) sind digitale Spiele, die nicht nur zur reinen Unterhaltung entwickelt wurden, sondern auch einen weiteren zusätzlichen Zweck zu erfüllen, wie z.B. Fitness, Gesundheit, Lernen. Diesen zusätzlichen Zweck nennt man charakterisierenden Zweck [Dörner et al. 2016, S. 2f.]. In Abschnitt 3.1 wird das Thema Serious Games genauer ausgeführt. Das Fraunhofer IOSB hat mit Lost Earth 2307 (LE2307) ein SG nach dem Konzept des Digital Game Based Learning entwickelt. Sowohl die Bildungslehre als auch die Spielentwicklung, die zur Unterhaltung dienen soll, integrieren dieses Modell nach [Kiili 2005] erfolgreich.

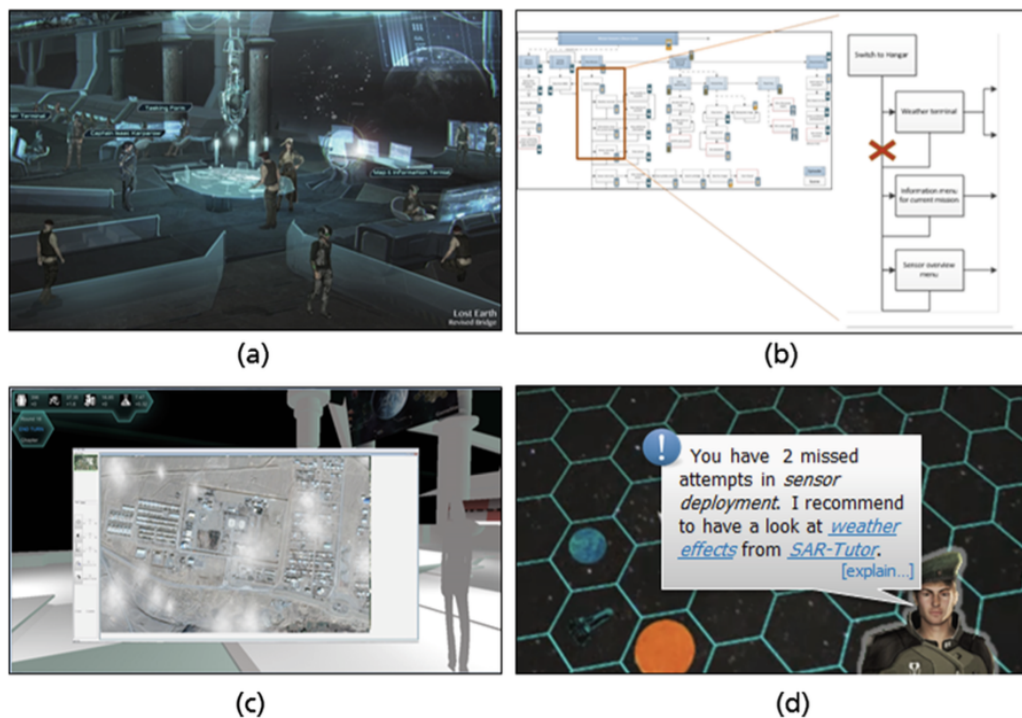


Abbildung 2.1: Szenen des Serious Game "Lost Earth 2307"; (a) Brücke um Missionen zu starten; (b) Pfad eines adaptiven Spielverlaufes; (c) Bild Interpretation mit dynamischer Modifikation der Schwierigkeit; (d) virtueller Assistent, der Hilfe und Tipps bietet [Streicher und Smeddinck 2016, S. 360]

## Adaptivität

Ein adaptives System wird aktiv, wenn in das Spielgeschehen eingegriffen wird, um Spielverhalten zu manipulieren. Techniken der Künstlichen Intelligenz (KI) bzw. des Machine Learnings (ML) können genutzt werden, um den Lernfortschritt des Nutzers zu schätzen und somit passende Adaptivitätszeitpunkte und -strategien zu entwickeln [Frutos-Pascual et al. 2017]. Eine zentrale Frage ist, wann Adaptivität passieren soll, also wann ein adaptives System aktiv wird [Frutos-Pascual et al. 2017].

In einem Artikel von Andersen [Andersen 2012] wird eine Methode erläutert, mit der sich SG individuell an den Nutzer anpassen können, indem das Spiel den Lernfortschritt messbar macht und anhand von Schwellenwerten richtige Zeitpunkte bestimmt. Diese Methode nennt man Knowledge Tracing (KT) (Kapitel 3.5.2). KT wird weiterhin im wissenschaftlichen Schreiben von [Doroudi et al. 2019] untersucht. Dabei gilt KT als eines der beliebtesten Modelle [Pavlik Jr

et al. 2009, S. 1]. In den wissenschaftlichen Arbeiten [Pavlik Jr et al. 2009], [Gong, Beck, et al. 2010] wird eine neue Alternative für KT vorgestellt. Dieses Modell heißt Performance Factor Analysis (PFA) (Kapitel 3.5.4) und basiert auf dem logistischen Regressionsmodell Learning Factor Analysis (LFA). Die genannten Regressionsmodelle bieten sich zur Datenanalyse an. Die Ergebnisse aus diesen Arbeiten weisen darauf hin, dass PFA den Lernfortschritt besser modellieren kann als KT. Was in diesen Schreiben jedoch fehlt, ist die Anwendung von diesen Regressionsmodellen auf ein konkretes Spiel.

Um Adaptivität in einem SG zu gewährleisten, muss eine Interaktion mit dem Nutzer stattfinden. Eine weitere wichtige Frage ist, wie Adaptivität eingesetzt wird, also welche Möglichkeiten des Eingriffs in das Spiel existieren. Im Artikel von [Sampayo-Vargas et al. 2013] wird folgendes vorgestellt: Eine verbreitete Methode sind Hinweise, die von dem adaptiven System zur Verfügung gestellt werden. Bei Hinweisen handelt es sich um eine Möglichkeit, dem Nutzer zu helfen und ihm den gestellten Sachverhalt zu vereinfachen. Das kann beispielsweise durch eine Herleitung zum Lösungsweg oder das Vorgeben einer möglichen Lösung geschehen [Sampayo-Vargas et al. 2013]. Darüber hinaus kann bei Serious Games der Schwierigkeitsgrad des Spiels gesteuert werden. Diese Arbeit orientiert sich am erweiterten vier-phasigen Adaptivitätszyklus [Shute et al. 2012] mit den Prozessen Erfassung, Analyse, Selektion und Darstellung (Abbildung 2.2).

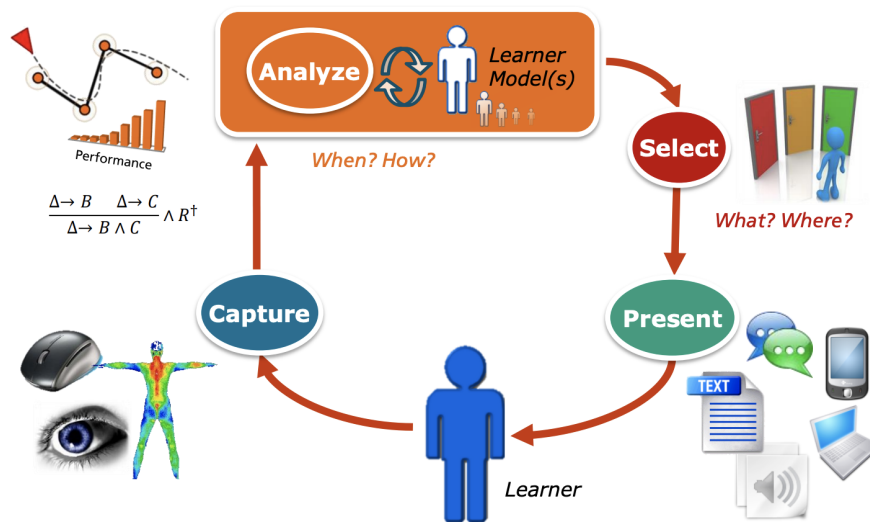


Abbildung 2.2: Vier-Phasen Adaptivitätszyklus [Streicher und Smeddick 2016]

Das Ziel der Arbeit ist es sowohl einen Lernfortschritt für den Nutzer mittels der oben

genannten Methoden zu berechnen als auch diese Berechnung dann zu nutzen, um Anpassungen an dem Spiel zu treffen. Im folgenden Abschnitt werden die bereits existierenden wissenschaftliche Schreiben und Ergebnisse zu den aufgeführten Modellen erläutert.

## 2.2 Lernfortschrittmodellierung bei Serious Games

Wie im Artikel von [Pavlik Jr et al. 2009] beschrieben, muss der Lernfortschritt des Nutzers messbar gemacht werden, um das Spiel an den Nutzer zu adaptieren. Das kann mithilfe von Regressionsmodellen realisiert werden. Wie von C. Romero and S. Ventura in „Educational data mining: A survey from 1995 to 2005“ [C. Romero et al. 2007] erläutert, wird im Bereich des adaptiven e-Learnings zwischen zwei grundsätzlichen Modellen unterschieden:

- Modelle zum Analysieren von bereits vorhandenen Lerndaten (Educational Data Mining, 2.3)
- Modelle zum Anzeigen und Anpassen von Lernfortschritt (Adaptive Tutoring System, Abbildung 2.4)

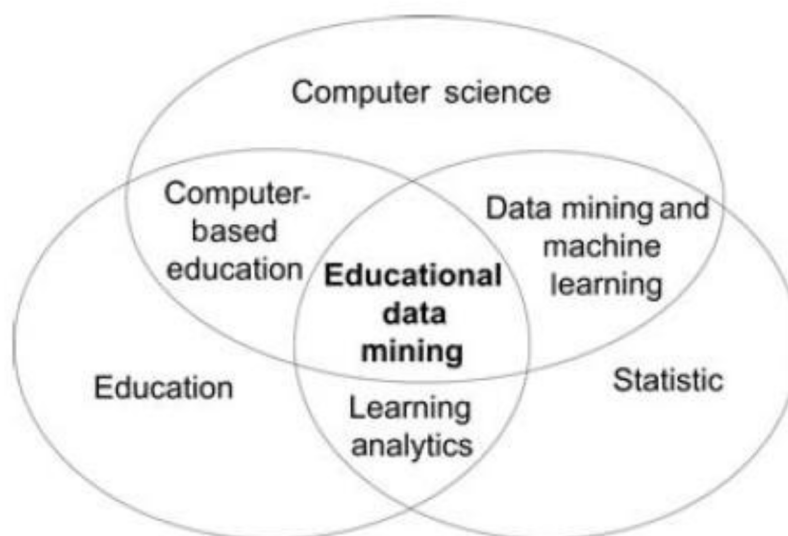


Abbildung 2.3: Bereiche aus denen das Educational Data Mining entstanden ist [Cristobal Romero et al. 2020].

Dabei ist Learning Factor Analysis ein Educational Data Mining Modell und Knowledge Tracing ein adaptives Tutoring System [Pavlik Jr et al. 2009].



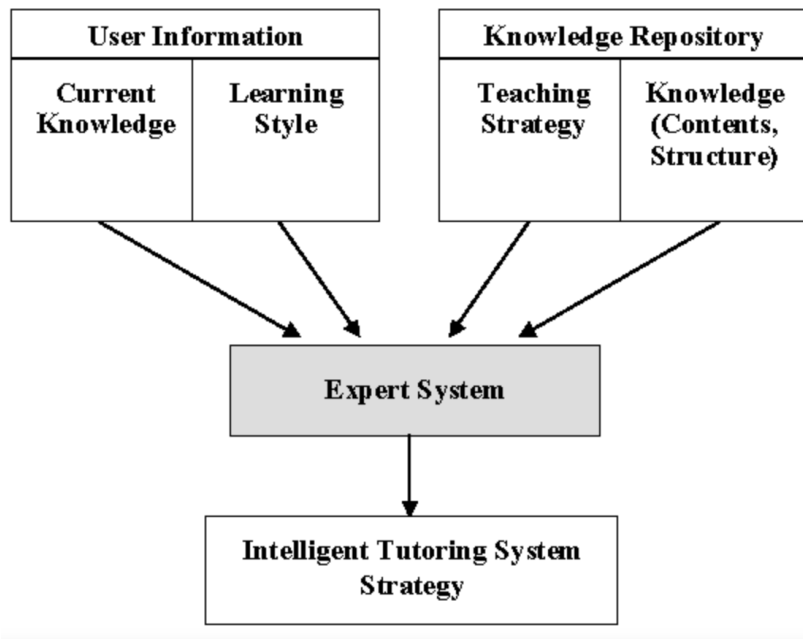


Abbildung 2.4: Aufbau eines adaptiven Tutoring Systems [Liegler et al. 2000]

Bevor es mit den Lernfortschrittmodellierungen weitergeht, müssen die Begriffe Daten und WK definiert werden. Wenn in dieser Arbeit von Daten geredet wird, sind damit die gesammelten Daten des Nutzers in einem adaptiven Lernspiel z.B. LE2307 gemeint. Diese Daten bestehen aus verschiedenen Wissenskomponente (WK). Die Definition einer WK orientiert sich in dieser Arbeit an der Erklärung von [K. R. Koedinger et al. 2012]. Eine WK ist eine Beschreibung einer Fähigkeit, die vom Nutzer verwendet werden kann. Eine WK kann alleine oder in Kombination mit anderen WK dazu verwendet werden, erfolgreich ein Problem oder ein Teilproblem zu lösen [K. R. Koedinger et al. 2012]. Einen genaueren Einblick zum Thema WK gibt es in Abschnitt 3.5.2.

In dieser Arbeit liegt der Fokus auf den adaptiven Tutoring Systemen. Um nachvollziehen zu können, wie das Performance Factor Analysis (PFA) Modell aus dem Artikel von Pavlik Jr, Cen und Koedinger entstanden ist und wieso es in wissenschaftlichen Arbeiten wie [Gong, Beck, et al. 2010; Pavlik Jr et al. 2009] genauer untersucht wurde, erläutert Kapitel 3 das Modell im Detail.

Das Problem bei LFA ist, dass die gespeicherten Daten keinem konkreten Nutzer zugewiesen werden können. Wie in [Pavlik Jr et al. 2009] bereits ausgearbeitet, ist das LFA Modell für adaptive Tutoring Systeme nicht geeignet, weil die Analyse der Daten nicht an einen bestimmten Nutzer angepasst werden, sondern nur für den Durchschnitt aller Nutzer gilt.

Mit KT ist es möglich, den aktuellen Wissensstand eines konkreten Nutzers einer bestimmten WK zu berechnen. Falls die gestellte Frage jedoch mehrere WK gleichzeitig abfragt, kann der Lernfortschritt mit Knowledge Tracing nicht mehr angezeigt oder angepasst werden. Da die meisten fortgeschrittenen adaptiven Lernspiele mehr als nur eine WK gleichzeitig abfragen, scheint das KT für moderne SG nicht optimal zu sein [Pavlik Jr et al. 2009].

Da LFA sich für Analyse von Datensätzen anbietet und die Möglichkeit besteht mehrere WK einzubeziehen, wird LFA als Basis für das, in [Pavlik Jr et al. 2009] neu entwickelte Modell, verwendet (Abbildung 2.5). Im wissenschaftlichen Schreiben [Pavlik Jr et al. 2009] zu Performance Factor Analysis (PFA) wird das LFA Modell umkonfiguriert. Das PFA Modell soll die Daten, wie in LFA, analysieren und die Erkenntnisse daraus an einen konkreten Nutzer binden [Pavlik Jr et al. 2009]. In Abschnitt 3.5.4 wird das gesamte PFA Modell genauer erklärt.

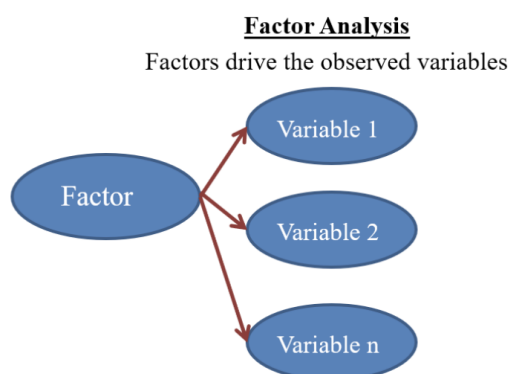


Abbildung 2.5: LFA und PFA betrachten für eine Fähigkeit (Faktor) diverse WK (Variablen)[*Introduction to PCA and EFA 2021*]

In [Pavlik Jr et al. 2009] und [Gong, Beck, et al. 2010] werden die beiden Modelle KT und PFA in mehreren Projekten miteinander verglichen und beide wissenschaftlichen Artikel kommen zum dem Fazit, dass in diesen konkreten Umgebungen PFA besser abschneidet als KT und damit besser für adaptive Tutoring Systeme geeignet ist [Gong, Beck, et al. 2010; Pavlik Jr et al. 2009].

In dieser Arbeit wird eine mögliche Umsetzung für das Performance Factor Analysis Modell dargestellt. Dieses Modell kann dann generisch für unterschiedliche Serious Games verwendet werden, um einen sogenannten „Performance Score“ zu berechnen. Weiterhin wird in der Implementierung dieser Arbeit ein intelligenter Tutor erweitert, der anhand des berechneten Wertes mit dem Nutzer über das SG Interface interagieren kann. Damit soll anhand eines konkreten Beispiels gezeigt werden, dass das PFA Modell für die Lernfortschrittsoptimierung verwendet werden kann. Im folgenden Kapitel werden die Grundlagen für die Modellierung

und die Implementierung vorgestellt.



## 3 Grundlegende Konzepte

In diesem Kapitel wird auf verschiedene, bereits entwickelte, Konzepte und Technologien eingegangen. Die hier vorgestellten Themen sind von großer Relevanz für den theoretischen sowie den praktischen Teil dieser Arbeit.

Zuerst wird das Konzept von SG und das Prinzip der Adaptivität erläutert. Anschließend werden diese Begriffe in einen gemeinsamen Kontext gesetzt. Danach wird die vom Fraunhofer IOSB entwickelte intelligente Tutor Komponente ELAI vorgestellt. Sodann wird der Service, mit dem die Nutzerdaten gesichert werden, vorgestellt. Zuletzt folgt eine Auseinandersetzung mit möglichen Modellen und Methoden der Datenanalyse, wie z.B. KT, LFA oder PFA.

### 3.1 Adaptive Lernspiele und Serious Games

Im folgenden Abschnitt wird auf die Begriffe Serious Games und Adaptivität eingegangen. Die Motivation hinter diesen Themen wird erläutert und die Begriffe in einen gemeinsamen Kontext geführt. Zum Schluss wird das konkrete Beispiel eines SG dieser Arbeit vorgestellt.

#### 3.1.1 Digital Games und Serious Games

Menschen lieben es, Spiele zu spielen [Dörner et al. 2016, S.2]. Spiele wie Backgammon werden heute immernoch gespielt, obwohl sie schon 5.000 Jahre alt sind. Nachdem der Mensch angefangen hat Gegenstände wie Bälle, Karten und Würfel zum Spielen zu verwenden, ist es nicht überraschend, dass der Mensch auch den Computer zu spielerischen Zwecken verwendet [Dörner et al. 2016, S.2]. Daraus sind die Digital Games entstanden. Spiele, die eine bestimmte Art von Computer, beispielsweise Smartphone, Laptop, Konsole usw. nutzen, nennt man Digital Games [Dörner et al. 2016, S.2].

Digitale Games sind sehr erfolgreich und erreichen mittlerweile sehr viele Altersgruppen [Dörner et al. 2016, S.2]. Aber wenn Digital Games so erfolgreich sind, wieso sollte man sie nicht auch für andere Zwecke neben der Unterhaltung einsetzen? Im Bereich des Fernsehens, der ebenfalls zur Unterhaltung dient, wurden auch Fortschritte im Bereich des Lernens festgestellt. Ein Beispiel für eine Kindersendung, die sowohl der Unterhaltung als auch dem Lernfortschritt

dient, ist die Sesamstraße [Dörner et al. 2016, S.2]. Das gleiche könnte auch im Bereich der Digital Games funktionieren. Somit schließen wir auf die Definition von SG. Ein Serious Games (SG) ist ein Digital Game, das sowohl Unterhaltung als auch ein weiteres zusätzliches Ziel (z.B. Gesundheit oder Lerneffekt) hat [Dörner et al. 2016, S.2]. Diese zusätzlichen Ziele nennt man charakterisierende Ziele. In Abbildung 3.1 ist ein Schaubild zu den oben definierten Begriffen zu sehen. Dabei sind SG der Schnitt zwischen Lernen, Simulation und Spielen. Diese drei Bereiche bilden gemeinsam SG.

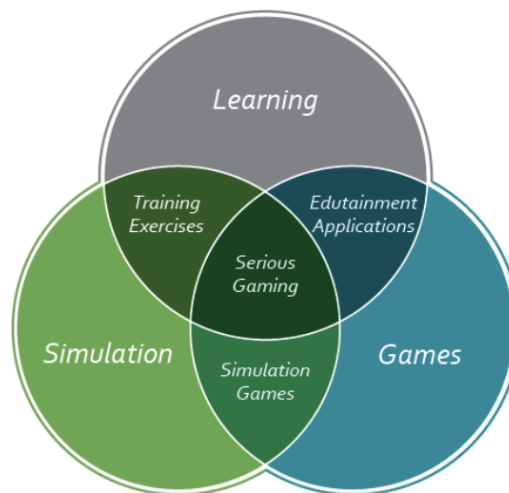


Abbildung 3.1: Ein Schaubild zum Thema SG (Bildquelle [*Defining serious games - The primary objective is not entertainment* 2016])

Das charakterisierende Ziel dieser Arbeit ist der Lerneffekt. Dabei soll der Nutzer des Spiels nicht nur unterhalten werden, sondern auch einen Lernfortschritt in einem konkreten Themenbereich vermittelt bekommen. Abbildung 3.2 zeigt eine Bildschirmaufnahme eines SG mit dem Lerneffekt als charakterisierendes Ziel. Das Spiel soll den Wissensstand des Nutzers im Bereich der Bildauswertung verbessern.

### 3.1.2 Lernfortschrittsoptimierung durch Adaptivität

Wie im vorherigen Abschnitt bereits erläutert, ist das Ziel eines SG den Nutzer zu unterhalten und ein weiteres charakterisierendes Ziel zu erfüllen. Im Fall dieser Arbeit ist das Ziel, einen Lerneffekt zu erzielen. Hierfür sind Entwurf und Design des Spielverlaufs entscheidend [Streicher und Smeddinck 2016]. Ist das Spiel zu starr und hat der bereits Nutzer grundlegende Kenntnisse zu dem erlernenden Wissen, ist der Nutzer unterfordert. Ein Nutzer, der von der Geschwindigkeit des Spiels zu sehr herausgefordert ist und an den sich die Lernge-

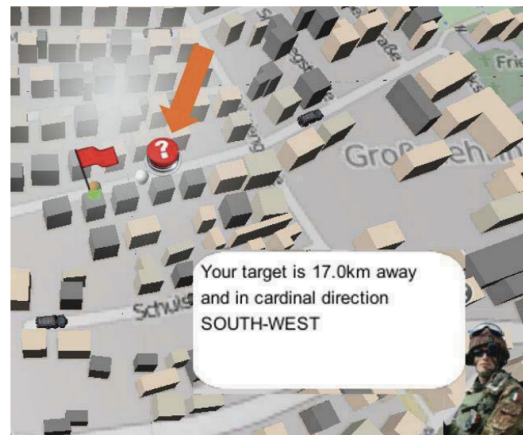


Abbildung 3.2: Das SG SaFIRA, das auf einem Suchen-und-Finden Spiel basiert (Bildquelle [Streicher, Roller, und Biegemeier 2017]).

schwindigkeit nicht anpasst, ist wiederum überfordert. In beiden Fällen werden sowohl die Motivation als auch der Lerneffekt negativ beeinflusst [Streicher und Smeddinck 2016]. Gibt es eine Möglichkeit um den Spielverlauf in Abhängigkeit vom Nutzer zu ändern? Bei dem gewünschten Verhalten kann das Prinzip der Adaptivität helfen [Streicher und Smeddinck 2016]. Adaptivität in dieser Arbeit ist die dynamische Adaptierung von Serious Games an die Bedürfnisse des Nutzers [Streicher und Smeddinck 2016]. Bei einem Interaktionsmechanismus können Inhalte und Empfehlungen von einer intelligenten Tutor Komponente an den Nutzer angepasst werden. Mit Hilfe von Mitteln der Datenanalyse und des maschinellen Lernens wird versucht, den positiven Einfluss durch die Interaktion zu maximieren [Streicher und Roller 2015]. Ein Beispiel für eine intelligente Tutor Komponente ist ELAI. Um adaptives Verhalten zu ermöglichen, müssen Auslöser für solche Modifikationen fest definiert sein. Diese Auslöser sind feste Schwellenwerte konkreter Metriken bei der Analyse, wie z.B. Anzahl an Spielzeit oder eine Analyse des Spielverhaltens. Bei der Analyse des Spielverhaltens könnte man beispielsweise die Anzahl an korrekten oder falschen Antworten verwenden. Mithilfe dieser Auslöser kann dann ein Zeitpunkt bestimmt werden, um den Spielverlauf an die Bedürfnisse des Nutzers anzupassen. Ein Spiel, das sich an die Bedürfnisse des Nutzers anpasst, um die Lernperformance zu optimieren, nennt man adaptives Lernspiel.

## Flow

Wieso Adaptivität eine so wichtige Rolle für die Lernfortschrittsoptimierung spielt, kann durch das Konzept des Flows begründet werden. Der Begriff Flow wurde von Mihály Csíkszent-

mihályi [Csikszentmihalyi 2009] im psychologischen Sinne geprägt. Mihály Csikszentmihályi beschreibt Flow als einen psychologischen Zustand, in dem die gesamte Aufmerksamkeit eines Menschen auf eine Aufgabe gerichtet wird [Csikszentmihalyi 2009]. Um in diesem Zustand zu bleiben, muss der Mensch sich im sogenannten Flow Kanal/ in der Flow Zone aufhalten. Diese Zone befindet sich zwischen der Überforderung und der Langeweile [Csikszentmihalyi 2009]. Unterschiedlichen Grund- und Vorkenntnissen erschweren es, den jeweiligen Flow Kanal des Menschen zu bestimmen. Anhand von Informationen über den Wissenstand des Menschen können Anpassungen an der Aufgabe getroffen werden, um jeden Menschen individuell in die Flow Zone zu führen. Csikszentmihályi beschreibt den Flow Zustand als harmonisches Zusammenspiel von Körper und Geist. Laut Csikszentmihályi kann im Flow Zustand Höchstleistung erbracht und das gesamte Potential des Menschen entfaltet werden [Csikszentmihalyi 2009].

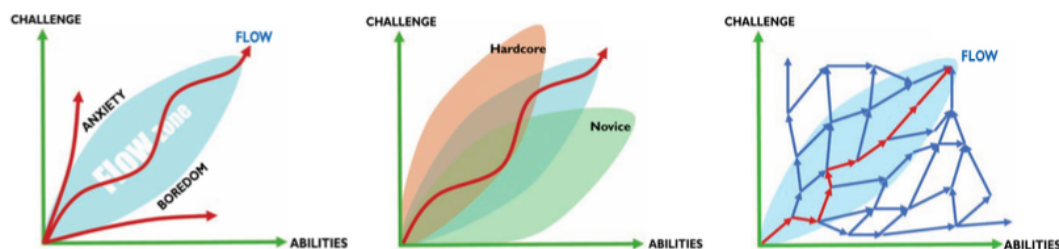


Abbildung 3.3: (a) Spieler können während des Spielverlaufs unterschiedliche Verhaltensmuster annehmen; (b) Es muss eine Klassifizierung des Spielers erfolgen um seine Fähigkeiten bestimmen zu können; (c) Durch Anpassung im Spielverlauf kann der Spieler im Flow Kanal gehalten werden (Bildquelle: [Chen 2007])

Um den Lernfortschritt zu optimieren, ist das Ziel den Nutzer in dieser Flow Zone zu halten. Das heißt, dass sich der Spielverlauf des Serious Games anpassen muss, um den Nutzer nicht zu über- oder unterfordern. Diese Anpassungen werden mittels Adaptivität erreicht. Aber woran wird diese Über- oder Unterforderung erkannt? Beispielsweise könnte eine falsche Antwort auf eine Frage ein Indikator dafür sein, dass der Nutzer überfordert ist. Anders könnte eine Reihe von richtigen Antworten darauf hinweisen, dass der Nutzer unterfordert ist. Im folgenden Abschnitt wird eine Möglichkeit erläutert, wie man Adaptivität in eine Lernanwendung integrieren kann, um mithilfe der Adaptivität den Nutzer im Flowkanal zu halten, um so den Lernfortschritt zu optimieren.



## Adaptivitätszyklus

Nachdem im vorherigen Abschnitt motiviert wurde, wieso Adaptivität ein wichtiger Faktor in der Lernfortschrittsoptimierung ist, wird in diesem Abschnitt eine Methode erläutert, wie man Adaptivität in Serious Games integrieren kann. Der vierphasige Adaptivitätszyklus (Abbildung 3.4 von Shute und Zapata-Rivera [Shute et al. 2012]) beschreibt ein iteratives Verfahren, um den Lernvorgang an die Bedürfnisse des Nutzers anzupassen. Dabei besteht das Modell aus vier verschiedenen Phasen:

- Erfassung („Capture“): In dieser Phase werden während des Lernprozesses Informationen über den Nutzer gesammelt.
- Analyse: Hier werden die gesammelten Daten mittels intelligenter Analysetechnik ausgewertet und anhand der Ergebnisse Entscheidungen über den Lernfortschritt getroffen.
- Selektion („Select“): In diesem Schritt werden Anpassungen des Lernprozesses ausgewählt, die für die Realisierung der Adaptivität relevant sind.
- Darstellung („Present“): Die ausgewählten Anpassungen werden dem Nutzer im Kontext des Lernprozesses vorgestellt.

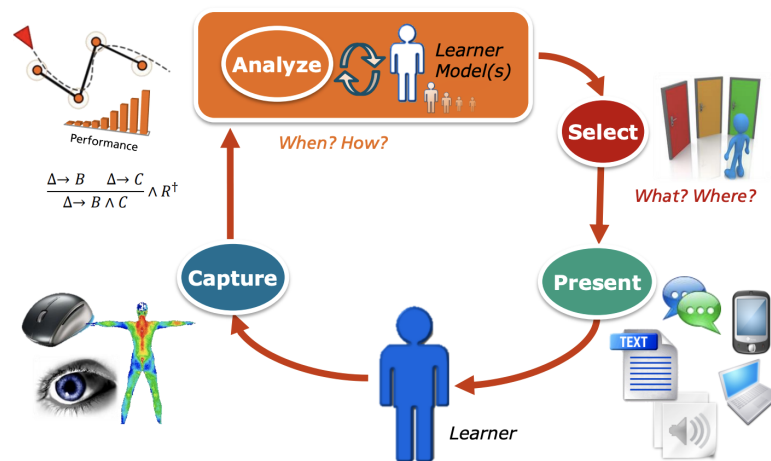


Abbildung 3.4: Der vierphasige Adaptivitätszyklus von Shute und Zapata-Rivera (Bildquelle: [Streicher und Smeddinck 2016])

Nach der Darstellungsphase geht der gesamte Zyklus wieder von vorne los. Dabei werden neue Informationen erfasst, da der Nutzer sein Verhalten wieder an das Dargestellte anpasst.

Im Rahmen dieser Arbeit ist das Konzept der Adaptivität sehr interessant, weil es das Ziel der Arbeit ist, Adaptivität in ein SG zu integrieren. Um Adaptivität zu integrieren, muss aber zuerst bestimmt werden, ob der Nutzer eine Anpassung benötigt. Der Kernteil der Arbeit befasst sich mit der Analysephase in dem Adaptivitätszyklus. Dort wird mittels Performance Factor Analysis (PFA) anhand der erfassten Daten bestimmt, ob eine Anpassung im Spielverhalten notwendig ist.

### 3.2 Beispiel für Serious Games: Lost Earth 2307

In dieser Arbeit wird das Konzept der Adaptivität in ein konkretes SG integriert. In diesem Abschnitt wird das SG vorgestellt. Das SG Lost Earth 2307 (LE2307) ist ein 4X Strategiespiel [*Lost Earth 2307 - Fraunhofer IOSB 2021*]:

- Untersuchen (eXplore)
- EXpandieren
- Auswerten (eXploit)
- Vertreiben (eXterminate)

Das SG wurde nach dem Konzepten des Digital Game Based Learning entwickelt. Das bedeutet, dass das Spiel zur Unterhaltung dienen und die Lernmotivation für die Lernziele erhöhen soll. Der Spieler befindet sich auf einem Raumschiff im Weltraum und erkundet fremde Planeten. Dabei lernt er optische, Infrarot- und Radarbilder auszuwerten. Außerdem wird der Spieler mit unterschiedlichen Sensorplattformen wie Drohnen, Flugzeugen und Satelliten konfrontiert.

Die Aufgabe dieses Spiels ist es, zukünftigen Bildauswertern Grundkenntnisse in der Bildauswertung zu vermitteln. Die Unterschiede der verschiedenen Bildaufnahmen und die Vor- und Nachteile der Sensorplattformen sollen verstanden werden. Mit diesen Aufgaben sollen die Kenntnisse der Lernziele verbessert werden.

Für die Implementierung dieser Arbeit wurde das SG Streamlined Lost Earth (SLE) verwendet. SLE ist im Rahmen der Masterarbeit von Marlene Dillig [*Streamlined Lost Earth 2021*] entstanden und basiert auf der Grundidee von LE2307. In Abbildung 3.5 sieht man die Raumschiffansicht und in Abbildung 3.6 wird der Nutzer vor die Auswahl verschiedener Sensoren gestellt. Abbildung 3.7 zeigt ein Beispiel für eine Mission aus dem Spiel SLE.

Der im folgenden Kapitel vorgestellte intelligente Tutor Assistent ELAI wurde dabei als Lernhilfe für den Spieler entwickelt, um eine hohe Lernmotivation sowie einen nachhaltigen Lernprozess sicherzustellen.



Abbildung 3.5: Das Raumschiff, auf dem sich der Spieler im Weltraum befindet.

### 3.3 E-Learning A.I (ELAI)

Diese Arbeit baut auf dem am Fraunhofer IOSB [Streicher und Roller 2017] entstandenen Framework E-Learning Artificial Intelligence (ELAI) auf. ELAI ist als eigenständige Software entwickelt und dient der Entkopplung von SG und Adaptivität. Spiele, wie beispielsweise Lost Earth 2308 und SaFIRA, verwenden intern ELAI, um adaptive Lernmethoden zu nutzen [Streicher und Roller 2017].

Die dabei zugrunde liegende Software Architektur verwendet Experience API (xAPI), um das Nutzerverhalten zu erfassen und zu speichern. Die ELAI Software Architektur besteht aus generischen und spezifischen Komponenten. Die spezifische Komponente ist der ELAI Adapter, welcher Nutzerinteraktionen in adaptiven Lernspielen und SG beobachtet und erfasst [Streicher, Roller, und Biegemeier 2017]. Der Adapter muss in diesem Kontext spezifisch sein, da er sich für die Datenerfassung an das konkreten Spiel oder das Genre anpassen muss. Die generische Komponente ist als externes Werkzeug dafür zuständig, die über die xAPI aufgezeichneten Daten zu interpretieren, eine Adaptivitätsstrategie auszuwählen und diese dem Adapter zu senden [Streicher, Roller, und Biegemeier 2017].

In Abbildung 3.8 ist die ELAI Framework Architektur dargestellt. Dabei ist das SG mit dem Adapter ("ELAI Interface") verbunden. Der Adapter kommuniziert über eine Kommunikationsebene ("Communication Layer"xAPI) mit der ELAI. Weil die ELAI generisch ist, kann sie auf



Abbildung 3.6: Der Bereich, in dem Spieler ihre Sensoren für die Bilder auswählen.

verschiedene Spiele angewendet werden [Streicher, Roller, und Biegemeier 2017]. Die ELAI Komponente lässt sich in zwei weitere Teile trennen:

- Interpretationsmaschine („Interpretation Engine“): Ist dafür zuständig die Daten aus der Datenbank auszuwählen (z.B. LRS) und die Daten anschließend zu analysieren.
- Einflussmaschine („Influence Engine“): Ist dafür zuständig eine Adaptivitätsstrategie, falls nötig, auszuwählen und die Änderungen anzuzeigen (z.B. ein Hinweis).

Dabei werden die, in der Erfassungsphase, aufgenommenen Daten des vierphasigen Adaptivitätszyklus [Shute et al. 2012] in der Analysephase von der Interpretationsmaschine analysiert. Es werden verschiedene Methoden, wie z.B. KT, LFA oder PFA, der Datenanalyse eingesetzt, um die Daten in, für die Maschine interpretierbare, Informationen umzuwandeln. In der Selektionsphase ist die Einflussmaschine dafür zuständig nach einer konkreten Adaptivitätsstrategie in den Spielverlauf einzugreifen und den Verlauf an die Bedürfnisse des Nutzers anzupassen. Diese Adaptivitätsstrategie ist spezifisch für das jeweilige SG definiert. In Abbildung 3.9 ist der Ablauf einer Nutzerinteraktion in einem SG mit der zusammenhängenden ELAI Komponente zu sehen.



Abbildung 3.7: Ein optisches Bild mit einer Frage zur Bildauswertung.

### 3.4 Experience Application Programming Interface (xAPI)

Wie bereits in den vorherigen Abschnitten beschrieben, werden für die Lernfortschrittsoptimierung durch Adaptivität Daten des Nutzers benötigt. In dieser Arbeit wird die Service Experience API (xAPI) verwendet, um die benötigten Daten des Nutzers zu speichern. xAPI ist ein Service für Lernumgebungen, der es ermöglicht, Daten über die Erfahrungen des Nutzers, sowohl online als auch offline, zu sammeln [What is xAPI aka the Experience API or Tin Can API 2021]. Hierbei werden Daten in einem festgelegten Format an Learning Record Stores (LRS) übermittelt. Das festgelegte Format sind sogenannte Statements, die sich aus Actor, Verb und Object zusammensetzen. Dabei ist Actor eine eindeutige Information, die das Subjekt der Aktivität beschreibt, wie z.B. ein Student oder eine Gruppe an Nutzern. Das Verb repräsentiert eine Aktivität, die der Actor durchführt, z.B. spielen, lesen, springen. Die Object Eigenschaft ist meistens ein Nebenereignis, das zu der Aktivität gehört. Ein solches Nebenereignis könnte beispielsweise ein SG, ein Buch oder ein Baum sein [Kevan et al. 2016]. LRS bieten damit einen Zugang zu großen Datenmengen, die man anschließend zum Zweck der Datenanalyse nutzen kann.

Statements können dennoch weitere Informationen enthalten, wie z.B. Kontext oder Ergebnisse der aktuellen Aktivität [Kevan et al. 2016]. xAPI Statements orientieren sich stark an dem vom Menschen lesbaren Format. Beispielsweise wird für einen Nutzer, der eine Lektüre liest,

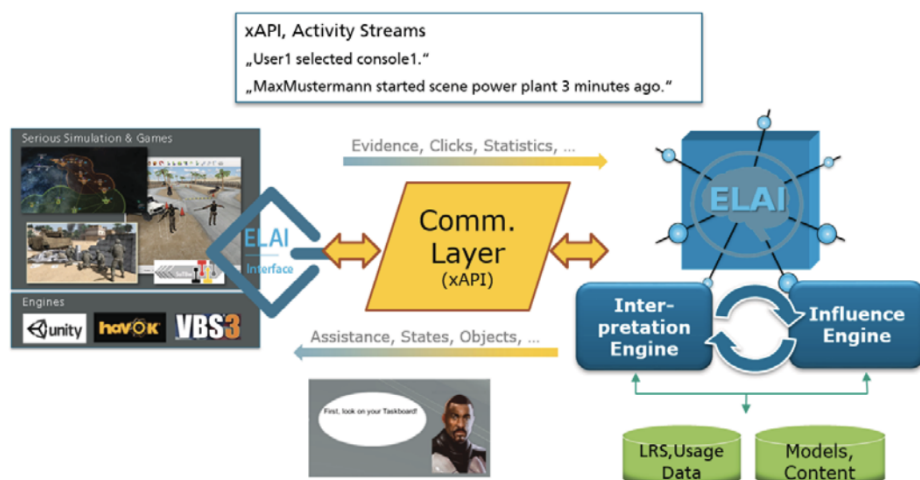


Abbildung 3.8: E-Learning Artificial Intelligence (ELAI) Framework Architektur mit Anbindung an ein SG [Streicher, Roller, und Biegemeier 2017]

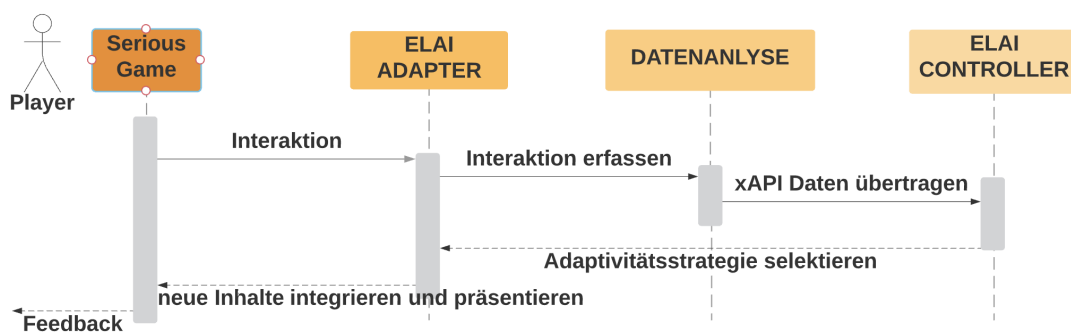


Abbildung 3.9: Ablauf der Interaktion eines Nutzers mit dem SG und dem ELAI Framework

folgendes Statements generiert: Max Mustermann (Actor) liest (Verb) Faust. Eine Tragödie (Object). Diese Eigenschaft macht diese Statements für Menschen gut leserlich und leicht verständlich. xAPI ist der Nachfolger von Shareable Content Object Reference Model (SCORM) desselben Herstellers Advanced Distributed Learning (ADL). Anders als seine Vorgänger, ist die Experience API zwischen verschiedenen Geräten und Anwendungen interoperabel. Dadurch bietet die xAPI deutlich mehr Freiheiten und Möglichkeiten im Zugriff auf die Statements. Das Statement Objekt wird im JavaScript Object Notation (JSON) Format gespeichert. Die JSON wird aus vier Syntax Regeln aufgebaut:

- Daten sind Name/Value Paare
- Daten werden durch ein Komma getrennt

- Geschweifte Klammern stehen für Objekte
- Eckige Klammern stehen für Felder

Alle JSON Objekte sind innerhalb geschweifter Klammern geschrieben. Alle Datenpaare haben ebenfalls eine festgelegte Struktur. Ein Datenpaar besteht aus einem Namensfeld in Anführungszeichen, gefolgt von einem Doppelpunkt und beendet mit einem Wert.

In Abbildung 3.16 ist ein xAPI Statement zum „Performance Score“ (Abschnitt 3.5.3) mit zusätzlichen Informationen aufgeführt. Nachdem das System vorgestellt wurde, mit dem die Datenmengen gespeichert werden, werden im folgenden Unterkapitel Methoden und Modelle zur Datenanalyse vorgestellt.

## 3.5 Regressionsmodelle

In diesem Abschnitt werden die verwendeten und zusammenhängenden Regressionsmodelle erläutert. Zuerst sollte jedoch der Begriff der Regressionsanalyse erklärt werden. Regressionsanalyse ist eine konzeptuelle Methode um funktionelle Beziehungen zwischen Variablen zu untersuchen [Chatterjee et al. 2015]. Dabei wird die Beziehung der Variablen in Form einer Gleichung oder eines Modells dargestellt. Die Beziehung besteht dabei zwischen einer abhängigen und einer oder mehreren unabhängigen Variablen. Solche Gleichungen oder Modelle nennt man Regressionsmodelle. Es gibt viele unterschiedliche Regressionsmodelle. Die in dieser Arbeit behandelten Performance Factor Analysis (PFA) und Learning Factor Analysis (LFA) basieren auf dem klassischen logistischen Regressionsmodell, das im folgenden Unterkapitel behandelt wird. Aus diesem Grund wird nur auf dieses Modell weiter eingegangen und die Weiteren werden nicht erwähnt.

### 3.5.1 Logistisches Regessionsmodell

Das logistische Regressionmodell, oder auch Logit-Modell, ist eine Erweiterung des linearen Regressionsmodells. Das Logit-Modell wird verwendet, um die Wahrscheinlichkeit für das Eintreten eines bestimmten Ereignisses darzustellen. Dabei wird das Ereignis in zwei Klassen unterteilt. Diese beiden Klassen sind Treffer/Niete, 0/1, true/false, usw. Den Vorgang, das Ereignis einer Klasse zuzuordnen, nennt man Klassifikation. Die logistische Regression wird in zwei Regressionsmodelle unterteilt:

- Binär logistische Regression: die abhängige Variable des Regressionsmodells hat zwei Ausprägungen (dichotom)



- Multinomiale logistische Regression: die abhängige Variable des Regressionsmodells hat mehr als zwei Kategorien

Die vorliegende Arbeit behandelt ausschließlich die binäre logistische Regression, da es in dieser Arbeit darum geht, ob die PFA das Wissen des Nutzer als verstanden oder nicht verstanden klassifiziert. In Abbildung 3.10 sieht man den Unterschied zwischen der logistischen und der linearen Regressionsfunktion. Die logistische Funktion wird auch als Sigmoidfunktion bezeichnet. Diese Funktion wird in Abschnitt 3.5.3 nochmal aufgegriffen.

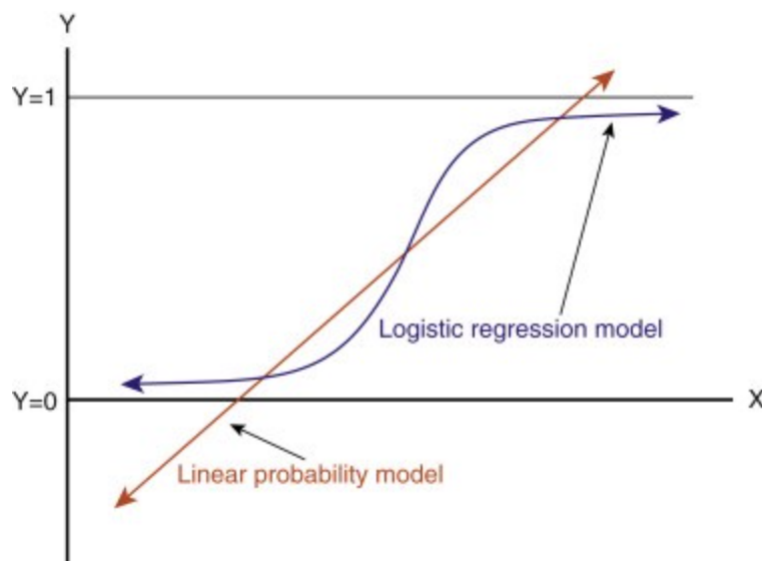


Abbildung 3.10: Verteilungsfunktionen der linearen und logistischen Regressionsmodelle [Lobo et al. 2010]

### 3.5.2 Knowledge Tracing (KT)

Knowledge Tracing (KT) ist eine weit verbreitete Methode für adaptive Tutoring Systeme [De Sande 2013]. KT kommt in zwei unterschiedlichen Formen vor:

- Bayesian Knowledge Tracing (BKT) Hidden Markov Modell (HMM): Gibt die Wahrscheinlichkeit für die korrekte Anwendung einer Fähigkeit an, wenn der Nutzer die Möglichkeit erhält, die Fähigkeit anzuwenden.
- Knowledge Tracing Algorithmus: Bestimmt die Wahrscheinlichkeit, dass ein Nutzer eine Fähigkeit beherrscht, in Echtzeit.



Das Hidden Markov Modell ist die bekannteste Technik für Verhaltensmustererkennung, die in Tutoring Systemen verwendet wird [Falakmasir et al. 2013; Yadav et al. 2020]. In dieser Ausprägung des BKT wird die Wahrscheinlichkeit berechnet, mit der ein Nutzer eine bestimmte Fähigkeit anwenden kann. Zur Berechnung werden die vergangenen erfolgreichen und fehlgeschlagenen Anwendungen der Fähigkeit des Nutzers verwendet. Diese Wahrscheinlichkeit wird in adaptiven Tutoring Systemen dazu verwendet, um herauszufinden, ob der Nutzer eine konkrete Fähigkeit beherrscht. Solche Fähigkeiten nennt man auch Wissenskomponente (WK). Beispielsweise kann es im SG Lost Earth 2307 folgende Wissenskomponenten geben:

- "Nutzer erkennt optische Bilder"
- "Nutzer erkennt Radarbilder"
- "Nutzer erkennt Infrarotbilder"

In der Praxis ist das HMM ein zwei-stufiges Verfahren. In der ersten Stufe wird der vergangene Verlauf der Wissenskomponente ausgegeben. Dabei wird eine Menge an Parametern ausgegeben:

- Initialer Lernfortschritt: Die Wahrscheinlichkeit, dass der Nutzer bei der ersten Anwendung die Wissenskomponente bereits beherrscht.
- Lernrate: Die Wahrscheinlichkeit, dass ein Nutzer, bei der Gelegenheit die WK zu nutzen, die Fähigkeit erwirbt.
- Schätzung (Guess Parameter): Beschreibt die Möglichkeit, dass der Nutzer eine korrekte Antwort generiert hat, obwohl er die WK nicht beherrscht.
- Ausrutscher (Slip Parameter): Sogar Nutzer, die eine WK beherrschen, können Fehler machen (z.B. Konzentrationsschwäche,...).

Wie in Abbildung 3.11 zu sehen ist, kann mithilfe der Parameter der aktuelle Wissensstand der WK iterativ berechnet werden. Abbildung 3.12 zeigt einen möglichen Pseudocode für die Bayesian Knowledge Tracing Hidden Markov Modell Berechnung.

Es gibt zwei Probleme, die beim Verwenden von KT auftreten und welche dazu führten, dass dieses Modell nicht für die Implementierung in das Serious Games Streamlined Lost Earth ausgewählt wurde. In den folgenden beiden Abschnitten werden die beiden Probleme von KT erläutert.

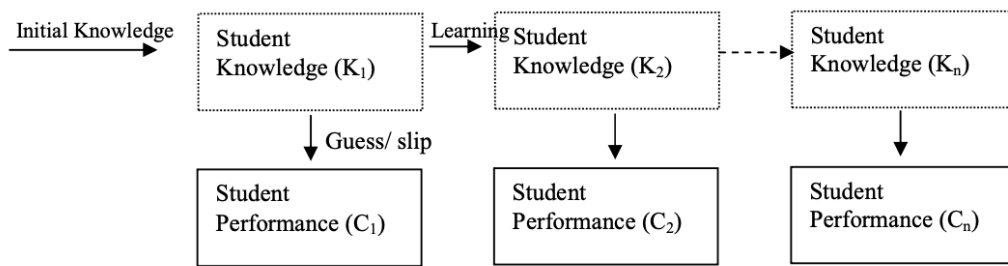


Abbildung 3.11: Knowledge Tracing Modell [Gong, Beck, et al. 2010, S. 3]

```

1   if responset-1 == correct
2       k = kt-1 * (1-slip) / (kt-1 * (1-slip) + guess * (1-kt-1));
3   else
4       k = kt-1 * slip / (kt-1 * slip + (1-kt-1) * (1-guess));
5   kt = k + (1-k) * learning;
  
```

Abbildung 3.12: Pseudocode für die HMM Berechnung [Gong, Beck, et al. 2010, S. 3]

### Modell Anpassung

Wie bereits in den wissenschaftlichen Schreiben von [Beck et al. 2007; Pardos et al. 2010; Gong, Rai, et al. 2009] erwähnt, gibt es bei KT zwei Probleme bei der Bestimmung von Parametern: lokale Minima und diverse globale Maxima. Das Problem der lokalen Minima wird durch die sogenannte „Error Surface“ hervorgerufen und kann durch mehrere Neustarts gelöst werden. Das zweite Problem ist auch bekannt als Mehrdeutigkeit. Diese liegt vor, wenn für die selben vorliegenden Daten mehrere Mengen an Parameterwerten zu den Daten passen. Da die unterschiedlichen Lösungsansätze als gleichwertig betrachtet werden, werden Eigenschaften wie Genauigkeit und Parameterplausibilität hinzugezogen, um für das Modell eine geeignete und eindeutige Lösung zu finden. Für dieses Problem gibt es mehrere Lösungsansätze wie z.B. der Expectation Maximization (EM) Algorithmus [Moon 1996] oder der Brute Force Ansatz [Baker et al. 2008]. In der Praxis sind diese Ansätze jedoch schwer umzusetzen, da sie sehr rechenintensiv bei der Modellsuche sind. Im Gegensatz zu KT bestehen für die, auf der logistischen Regression basierenden, Verfahren LFA und PFA keine Schwierigkeiten beim Erreichen von globalen Maxima.

### **Problem mit diversen Wissenskomponenten**

Das Bayesian Knowledge Tracing HMM hat einen weiteren großen Nachteil: Wenn eine Frage gestellt wird, die erfordert, dass diverse Wissenskomponenten gleichzeitig geprüft werden, ist das Modell unfähig alle Fähigkeiten gleichzeitig zu behandeln. Das Problem entsteht, weil KT bei einer Abfrage nur den vergangenen Verlauf einer einzelnen WK überprüft. Für ein Tutoring System ist es jedoch wichtig die zuvor genannte Eigenschaft zu besitzen, da viele Fragen meist nicht nur auf eine, sondern auf mehrere WK abzielen. Für dieses Problem existieren ebenfalls Lösungen wie z.B. die Frage bei der Berechnung zu wiederholen, bis alle Wissenskomponenten der Frage einmal geprüft wurden. Das bedeutet, dass die Frage mit mehreren WK in viele Fragen mit nur einer WK aufgeteilt wird. Dabei wird die Wahrscheinlichkeit auf Overfitting erhöht. Der zuvor genannte Ansatz ist eine mögliche Lösung für das Problem. Dieses Problem stellt sich bei der LFA und PFA Modellierung jedoch gar nicht erst, da beide diese Eigenschaft bereits implizit besitzen.

### **3.5.3 Learning Factor Analysis (LFA)**

Learning Factor Analysis (LFA) ist eine Educational Data Mining Methode und das Fundament für das im nächsten Unterkapitel vorgestellte Performance Factor Analysis Modell [Pavlik Jr et al. 2009]. Zunächst wird der Begriff Educational Data Mining bestimmt.

Durch das ständig wachsende Interesse im Bereich Data Mining und der Lernumgebung ist ein neuer Aufgabenbereich entstanden: Educational Data Mining [C. Romero et al. 2007]. Mit Lernumgebung ist das „Educational System“ gemeint. Darunter zählen Schulen, Universitäten, Hochschulen, Kindergärten, usw. Dabei wird klassisches Data Mining auf die Lernumgebung angewendet. Nach der Vorverarbeitung der vorliegenden Daten zur Lernumgebung können weitere Data Mining Techniken, wie z.B. Clustering, Klassifikation, Ausreißerererkennung und viele mehr, auf den Datensatz angewendet werden. Die Ergebnisse führen anschließend zu neuen Erkenntnissen, um die Lernumgebung weiterhin zu fördern [C. Romero et al. 2007]. In Abbildung 3.13 wird der Zyklus zur Anwendung von Data Mining auf Lernumgebungen dargestellt. Die Lernumgebungen werden von Pädagogen/innen entwickelt und von Lernenden genutzt. Data Mining wird verwendet um den Entwicklern dieser Systeme neue Erkenntnisse zu liefern oder zu bestätigen und Lernenden eine Empfehlung auszusprechen.

Learning Factor Analysis wird als Educational Data Mining Methode verwendet, um kognitive Modelle und Daten zwischen Studenten und Tutoren zu untersuchen. Dabei sind kognitive Modelle entweder Beschreibungen oder berechnete Repräsentationen des menschlichen Denkens über bestimmte Konzepte oder Fähigkeiten [Lane 2012]. Mittels LFA ist es somit möglich, die Schwierigkeit und die Lernrate von Wissenskomponenten zu bestimmen und eine Vorher-

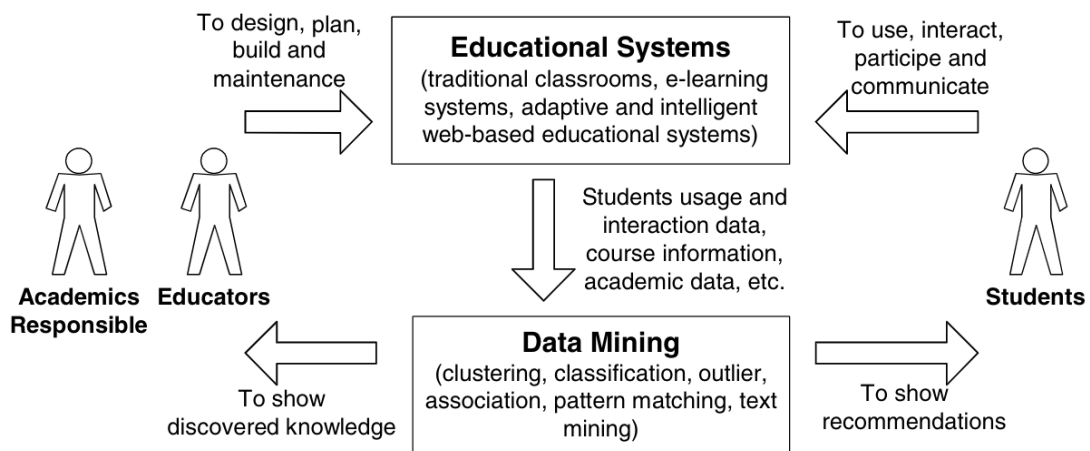


Abbildung 3.13: Zyklus zur Anwendung von Data Mining auf Lernumgebungen [C. Romero et al. 2007]

sage für die Performance jeder einzelnen WK zu treffen [Cen, K. R. Koedinger, et al. 2007]. Zum Beispiel wird LFA in einem Suchalgorithmus verwendet, um WK anhand diverser Faktoren zu trennen. Die Wissenskomponenten werden daraufhin Objekten zugewiesen, die den Datensatz bestmöglich repräsentieren und zusätzlich die Komplexität des Modell kontrollieren [Pavlik Jr et al. 2009]. Diese Zuweisung ist auch unter dem Begriff Q-Matrix bekannt. In Abbildung 3.14 ist ein Beispiel für eine Zuweisung von Variablen zu Faktoren dargestellt.

Im LFA Modell werden drei wichtige Eigenschaften festgehalten [Pavlik Jr et al. 2009]:

- Fähigkeit des Fachs: Für jedes Fach wird in einem Parameter festgehalten, was die Fähigkeit des Nutzers ist.
- Leichtigkeit der einzelnen WK: Gibt einen Schwierigkeitsgrad für unterschiedliche WK an.
- Lernrate der einzelnen WK: Für jede WK wird mithilfe eines Parameters die Lernrate festgehalten.

Bevor die Vor- und Nachteile der Learning Factor Analysis gegenüber anderen Modellen diskutiert werden, wird im nächsten Abschnitt die Berechnung des Modells vorgestellt.

### Berechnung der Learning Factor Analysis

Die klassische LFA Berechnung ist in Formel 3.1 zu sehen. Im folgenden werden die einzelnen Parameter erläutert:

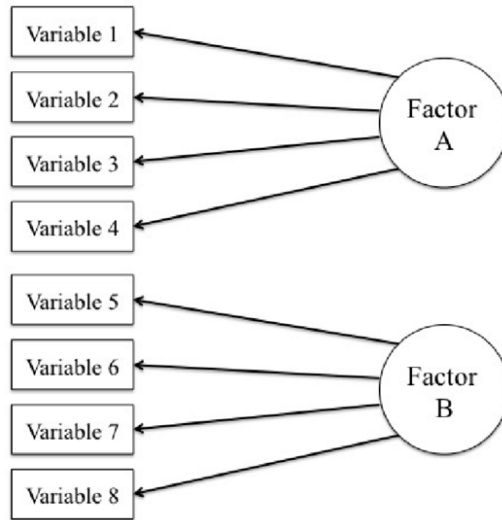


Abbildung 3.14: Zuweisung von Wissenskomponenten zu Faktoren [Costello et al. 2005]

$$m(i, j \in KCs, n) = \alpha_i + \sum_{j \in KCs} (\beta_j + \gamma_j n_{i,j}) \quad (3.1)$$

- Logit Wert  $m$ , der den akkumulierten Lernfortschritt oder auch „Performance Score“, für einen Nutzer  $i$ , repräsentiert:  $m \in \mathbb{R}^+$ , dabei steht 0 dafür, dass der Nutzer kein Wissen zu dem Fach besitzt.  $m$  kann für eine oder mehrere WK  $j$  berechnet werden.
- Nutzer  $i$ : ist eine eindeutige Kennzeichnung für einen Nutzer.
- Wissenskomponente  $j$ : Eine oder mehrere WK, die ein Fach oder Thema, das erlernt werden soll, repräsentiert.
- Vorteil von häufigen vorherigen Abfragen  $n$ : Eine Funktion  $n$  für vorherige Beobachtungen eines Nutzers  $i$  mit WK  $j$ .  $\gamma_{i,j}$  ist dabei eine Gewichtsfunktion von  $n$  für jede WK.
- Fähigkeit oder auch Lernobjekt  $\alpha$ : Für jeden Nutzer  $i$  die Fähigkeit, die eine oder mehrere WK enthalten kann.
- Schwierigkeitsgrad der WK  $\beta \in [0,1]$ : Gibt die Leichtigkeit der Wissenskomponente an.

Zunächst muss noch der Begriff Logit Wert erklärt werden: Der Logit Wert beschreibt den natürlichen Logarithmus einer Wahrscheinlichkeit  $p$  durch eine Gegenwahrscheinlichkeit  $1 - p$  [Becker et al. 2016].

Nachdem  $m$  berechnet wurde, kann mithilfe der Sigmoidfunktion (Formel 3.2) die Wahrscheinlichkeit der jeweiligen Vorhersagen der Analyse berechnet werden. Die Sigmoidfunktion, oder kurz S-Funktion, ist eine logistische Funktion mit einem S-förmigen Graphen (Abbildung 3.10). Die Funktion beschreibt eine stetig eindimensionale Wahrscheinlichkeitsverteilung und bietet eine Darstellung von Sättigungsprozessen. Dabei beschreibt  $e$  die Eulersche Zahl. Mithilfe dieser Funktion kann  $m$  in eine Vorhersage beobachteter Wahrscheinlichkeiten konvertiert werden.

$$p(m) = \frac{1}{1 + e^{-m}} \quad (3.2)$$

In Abbildung 3.15 ist ein Beispiel für die Anwendung von Learning Factor Analysis zu sehen. Im wissenschaftlichen Schreiben von [Cen, K. R. Koedinger, et al. 2007] wird ein Experiment zur Berechnung von rechteckigen und trapezförmigen Flächen durchgeführt. Dabei werden die tatsächlichen Fehlerraten der Nutzer mit den berechneten Fehlerraten durch LFA verglichen.

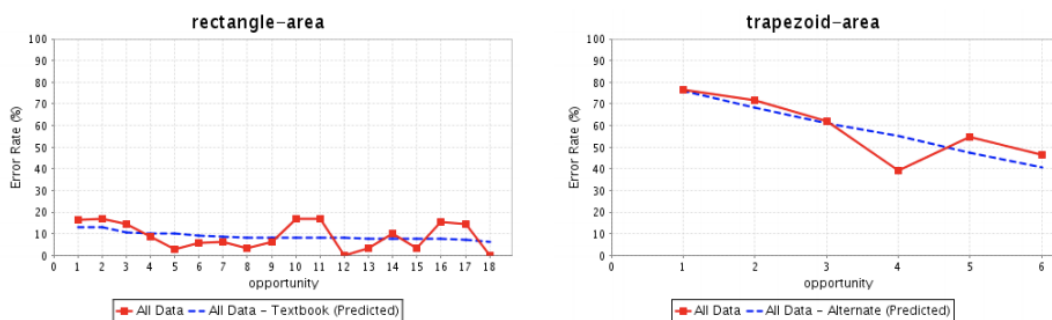


Abbildung 3.15: Lernkurve von rechteckigen und trapezförmigen Flächen: Die durchgezogenen (rot) Linien sind die tatsächlichen Fehlerraten und die gestrichelten (blau) Linien die vorhergesagten Fehlerraten des LFA Modells ([Cen, K. R. Koedinger, et al. 2007])

### Vor- und Nachteile von LFA gegenüber anderen Modellen

Wie bereits im Unterkapitel 3.5.2 beschrieben, ist das KT Modell nicht besonders gut dafür geeignet, den Lernfortschritt für Fähigkeiten mit diversen Wissenskomponente zu berechnen.

Im Vergleich dazu ist das LFA Modell eine nachgewiesene Lösung für das Problem von diversen WK innerhalb einer Fähigkeit [Leszczenski et al. 2007; Cen, K. Koedinger, et al. 2008]. Diese Lösung ist sehr hilfreich, da viele adaptive Tutoring Systeme Antworten erwarten, in denen mehrere WK involviert sind. Ein großer Nachteil, den LFA mit sich bringt, ist, dass dieses Modell Schwierigkeiten damit hat, einzelne WK bestimmten Nutzern zuzuschreiben. Das liegt daran, dass das Modell korrekte und inkorrekte Aussagen eines bestimmten Studenten ignoriert [Pavlik Jr et al. 2009]. Daher ist es dem Modell nicht möglich, einen „Performance Score“ für einen konkreten Nutzer zu berechnen.

LFA ordnet lediglich jedem Nutzer den gleichen Lernfortschritt zu. Damit ist dieses Modell zwar ungeeignet für diese Arbeit, bietet jedoch ein gutes Fundament für die Performance Factor Analysis. Im nächsten Abschnitt wird das Learning Factor Analysis Modell abgeändert, um das Modell adaptiv für jeden einzelnen Nutzer verwenden zu können.

#### 3.5.4 Performance Factor Analysis (PFA)

Die Performance Factor Analysis (PFA) ist nach [Pavlik Jr et al. 2009] eine auf der Learning Factor Analysis basierende Methode, um den Lernfortschritt des Nutzers zu berechnen. Vorab muss der Begriff Lernfortschritt definiert werden: Der Lernfortschritt des Nutzers beschreibt den aktuellen Wissensstand zu einem konkreten Lernobjekt oder einer Wissenskomponente, wie in Abschnitt 3.5.3 bereits erläutert.

Ein Vorteil der LFA ist, dass die Berechnung des Lernerfolgs mehrere WK betrachten kann. Aus diesem Grund bildet die Learning Factor Analysis die Basis für die Performance Factor Analysis. Ein großer Nachteil der LFA liegt darin, dass sich die Datenanalyse nicht auf einen konkreten Nutzer bezieht, sondern alle vorhandenen Daten zur Berechnung verwendet und daraus einen durchschnittlichen Lernfortschritt für alle Nutzer berechnet. Das hat zur Folge, dass dieses Modell sich nicht mehr an einen Nutzer anpasst, sondern an den Durchschnitt aller Nutzer. Deshalb wird das PFA Modell um einen zusätzlichen Parameter erweitert. Dieser neue Parameter repräsentiert den Nutzer und ermöglicht es somit, den Lernfortschritt eines konkreten Nutzers zu berechnen. Der Unterschied zur LFA besteht also darin, dass bei der PFA die Datenanalyse auf einen konkreten Nutzer zurückzuführen ist. Mit dem vorgestellten PFA Modell gibt es nun eine Methode zum Berechnen des Lernfortschritts. Solche Modelle nennt man adaptive Tutoring Systems.

Nach [Pavlik Jr et al. 2009; Gong, Beck, et al. 2010] soll PFA eine vielversprechende Alternative zu dem bekannten Modell KT bieten. Im Gegensatz zu PFA kann KT bei der Berechnung nicht mehrere WK gleichzeitig erfassen und somit kommen ausschließlich Fragen mit einer einzigen Komponente zur Analyse in Betracht.

### Berechnung der Performance Factor Analysis

Die klassische PFA Berechnung und die Erklärung der einzelnen Parameter ist in folgendem Abschnitt zu sehen [Pavlik Jr et al. 2009]:

$$m(i, j \in KCs, s, f) = \sum_{j \in KCs} (\beta_j + \gamma_i s_{i,j} + \rho_j f_{i,j}) \quad (3.3)$$

Die Erklärung der einzelnen Variablen:

- $m, i, j$  und  $\beta$  entsprechen den Parameter bei LFA.
- $s_{i,j}$ : Vorherige Erfolge der  $j$  für Nutzer  $i$ ;  $s_{i,j} \in 0,1$  oder  $s_{i,j} \in [0,1]$
- $f_{i,j}$ : Vorherige Misserfolge der Wissenskomponente  $j$  für Nutzer  $i$ ;  $f_{i,j} \in 0,1$  oder  $f_{i,j} \in [0,1]$
- $\gamma, \rho$ : Skalierungsfaktoren für Erfolg und Misserfolg der WK (Gewichtsfunktionen);  $0 \leq \gamma, \rho \leq 1$

Nachdem  $m$  berechnet wurde kann genauso wie im Learning Factor Analysis Modell (Abschnitt 3.5.3) mithilfe der folgenden Formel der Logit Wert in die Wahrscheinlichkeitsverteilung der Nutzer Performance konvertiert werden [Gong, Beck, et al. 2010]:

$$p(m) = \frac{1}{1 + e^{-m}}$$

### Unterschiede von PFA zu LFA

Nach [Pavlik Jr et al. 2009] ist der ausschlaggebende Unterschied zwischen den beiden Modellen, dass PFA empfindsam für den wichtigsten Indikator des Lernverhaltens des Nutzers ist: die Performance. Die Performance ist aus zwei Gründen besonders wichtig für das Lernverhalten des Nutzers. Erstens sind korrekte Antworten ein starkes Anzeichen dafür, dass das Wissen bereits sehr hoch ist. Dementsprechend ist diese Information bereits sehr hilfreich, um den Wissenstand des Nutzers zu bewerten. Zweitens führen korrekte Antworten zu besserem Lernfortschritt als falsche Antworten. Das kann unterschiedliche Gründe haben. Der Prozess einer richtigen Antwort wird besser verarbeitet oder das eine Überprüfung einer falschen Antwort weniger aufschlussreich ist. Es hat viele Vorteile, korrekte Antworten als Informationsquelle miteinzubeziehen, aber es mag auch sinnvoll erscheinen, fehlerhafte Antworten



zu betrachten [Pavlik Jr et al. 2009]. Die Sensibilisierung auf falsche Antworten kann einen weiteren Anhaltspunkt für das Lernverhalten bieten. Dabei repräsentiert der Misserfolg das Inverse des Erfolges. Gemeinsam können Erfolg und Misserfolg nicht nur die Quantität der Antworten, sondern auch eine Relation zwischen den beiden liefern [Pavlik Jr et al. 2009]. Ein weiterer Unterschied ist in der Gleichung 3.3 zu sehen: Der Parameter  $\alpha$  aus der LFA (Formel 3.1) ist weggefallen, da in einer Situation, in der Adaptivität gelten muss, dieser Parameter als schwer zu bestimmen gilt [Corbett et al. 1994].

### Nachteile von PFA

Bei der Performance Factor Analysis werden Erfolge und Misserfolge jedes einzelnen Lernobjektes bzw. jeder einzelnen Wissenskomponente festgehalten. Bei der Iteration über die WK wird die Reihenfolge der einzelnen Antworten jedoch nicht berücksichtigt [Gong, Beck, et al. 2010]. Zur Veranschaulichung ein Beispiel: Ein Nutzer soll eine Frage zum Satz des Pythagoras beantworten. Zuvor hat der selbe Nutzer vier Fragen zum selben Lernobjekt beantwortet. Von diesen vier Fragen wurden zwei richtig und zwei falsch beantwortet. Wie im Problem bereits beschrieben, ist die Reihenfolge der Antworten nicht einzusehen. Nun gibt es genau sechs Möglichkeiten, die zu dieser Konstellation an Antworten führen. In der Performance Berechnung sollte es einen Unterschied machen, ob ein Nutzer zuerst zwei mal falsch und dann zwei mal richtig oder ein Nutzer zuerst zwei mal richtig und dannach zwei mal falsch antwortet. Beide Fälle bieten abweichende Erkenntnisse über den vorherigen Performance Score des Nutzers.

Für dieses Problem bietet [Gong, Beck, et al. 2010] jedoch eine Lösung: Die Grundidee ist es, „Data Aging“ in die Berechnung von PFA einzubinden. Die modifizierte Berechnung basiert auf der Annahme, dass Antworten die weiter in der Vergangenheit liegen, weniger Einfluss auf den Performance Score haben sollten. In demselben wissenschaftlichen Schreiben wird der Verfallfaktor („decay factor“)  $\delta$  ( $0 < \delta \leq 1$ ) vorgestellt. Mit Hilfe des Verfallfaktors werden Antworten, die weiter in der Vergangenheit liegen, mit absteigender Bedeutung versehen. Die modifizierte Berechnung der Erfolge und Misserfolge ist in Formel 3.4 zu sehen:

$$\begin{aligned} success\_count_t &= \sum_{1 \leq k \leq t-1} P_k * \delta^{t-1-k} \\ failure\_count_t &= \sum_{1 \leq k \leq t-1} |P_k - 1| * \delta^{t-1-k} \end{aligned} \quad (3.4)$$

Diese Anpassungen ersetzen das  $s_{i,j}$  und  $f_{i,j}$  in der klassischen PFA Formel 3.3. Dabei bezeichnet  $t$  die  $t$ -te Frage, die er beantworten wird.  $P_k$  repräsentiert den Erfolg der  $k$ -te Frage.

Angenommen wir betrachten einen Erfolg als 1 und einen Misserfolg als 0, dann gelten für das obere Beispiel folgende Leistungen:

- Nutzer A: (0,0,1,1)
- Nutzer B: (1,1,0,0)

Weiter soll gelten, dass  $\delta = 0.9$ . Dann gilt folgende Berechnung für Nutzer A:

$$success\_count_4 = 0 * 0.9^3 + 0 * 0.9^2 + 1 * 0.9^1 + 1 * 0.9^0 = 1.9$$

$$failure\_count_4 = 1 * 0.9^3 + 1 * 0.9^2 + 0 * 0.9^1 + 0 * 0.9^0 = 1.5$$

Für Nutzer B würde dann für den Erfolg 1.5 und für den Misserfolg 1.9 gelten. Diese modifizierte Berechnung bietet nun die Möglichkeit die Reihenfolge der Antworten zu berücksichtigen [Gong, Beck, et al. 2010].

Nachdem alle Modelle und Methoden erläutert und analysiert wurden, wurde im Rahmen dieser Arbeit entschieden, dass das Performance Factor Analysis (PFA) Modell sowohl für das Serious Games (SG) Streamlined Lost Earth (SLE) als auch für andere intelligente adaptive Tutoring Systeme am besten geeignet ist. Deshalb wird im folgendem Kapitel 4 nur noch das PFA Modell behandelt.

```
1  {
2    "status" : "success",
3    "assistance" : {
4      "id" : "https://www.wikidata.org/wiki/Q1179505",
5      "display" : {
6        "en-US" : "AssistanceLevel"
7      },
8      "result" : {
9        "score" : {
10         "scaled" : 0.122791159524815
11       }
12     }
13   },
14   "result" : "created",
15   "skill" : {
16     "result" : {
17       "score" : {
18         "scaled" : 0.332473600971586
19       }
20     },
21     "display" : {
22       "en-US" : "SkillLevel"
23     },
24     "id" : "https://www.wikidata.org/wiki/Q205961"
25   },
26   "performance" : {
27     "id" : "https://www.wikidata.org/wiki/Q35140",
28     "result" : {
29       "score" : {
30         "scaled" : 0.729444897249172
31       }
32     },
33     "display" : {
34       "en-US" : "PerformanceScore"
35     }
36   }
37 }
```

Abbildung 3.16: Ein Beispiel für ein xAPI Statement im Learning Record Stores



## 4 Anwendung von Regressionsmodellen auf Serious Games

Das erarbeitete Konzept dieser Thesis besteht aus zwei Komponenten. Die erste Komponente ist der intelligente adaptive Assistent ELAI (Abschnitt 3.3) innerhalb eines Serious Games (Abschnitt 3.1). Dabei ist das ELAI Interface für die Interaktionen mit dem Nutzer zuständig. Die ausgewählten Interaktionen hängen hierbei von der Antwort der Interpretationsebene von ELAI ab, also ob der „Performance Scores“ einen konkreten Grenzwert überschreitet. Ein intelligenter adaptiver Assistent ist im Kontext dieser Arbeit eine Entität, die anhand eines berechneten „Performance Scores“ dem Nutzer gegebenenfalls hilft indem es das Spielniveau mit Hilfe von Hinweisen senkt. Die zweite Komponente dieser Arbeit ist das sogenannte PFA Modul, welches dafür zuständig ist, mithilfe von vorliegenden Daten über den Nutzer einen „Performance Score“ zu berechnen. Anhand dieses Scores soll anschließend entschieden werden, ob ELAI in das Spielgeschehen eingreifen soll oder nicht.

In Abbildung 4.1 ist der Entwurf, der durch diese Arbeit entstanden ist. Der Nutzer spielt ein Serious Games (SG), beispielsweise SLE oder LE2307. Aus Nutzerinteraktionen, wie z.B. Aufgaben lösen oder Missionen spielen, werden innerhalb des Spiels xAPI Statements (Abschnitt 3.4) generiert, die im Learning Record Stores gespeichert werden. Dabei ist das ELAI Interface ein integrierter Teil des SG. Anhand der vorliegenden xAPI Statements wird ein JSON Format mit Informationen, wie z.B. „Performance Score“ und Parameter über den Nutzer, erstellt. Das JSON Objekt wird von der ELAI über eine HTTP Anfrage angefragt und anhand des „Performance Scores“ bekommt das ELAI Interface eine Benachrichtigung, um einen Hinweis für den Nutzer im Spiel anzuzeigen. Die Hinweise werden dabei von der ELAI „Influence Engine“ erstellt und der „Performance Scores“ wird mittels eines Grenzwertchecks in der Interpretationsebene überprüft.

In Abbildung 4.2 ist ein vollständiges Aktivitätsdiagramm von einer gespielten Mission dargestellt. Dabei wird das ELAI Interface in Abschnitt 4.1 und das Performance Factor Analysis Modul in Abschnitt 4.3 nochmal aufgegriffen und detailliert erläutert. In Abbildung 4.3 sieht man ein Sequenzdiagramm für einen Spieldurchlauf einer Mission. Der linke Teil des Diagramms wird im folgenden Abschnitt und der rechte Teil in Abschnitt 4.3 besprochen.

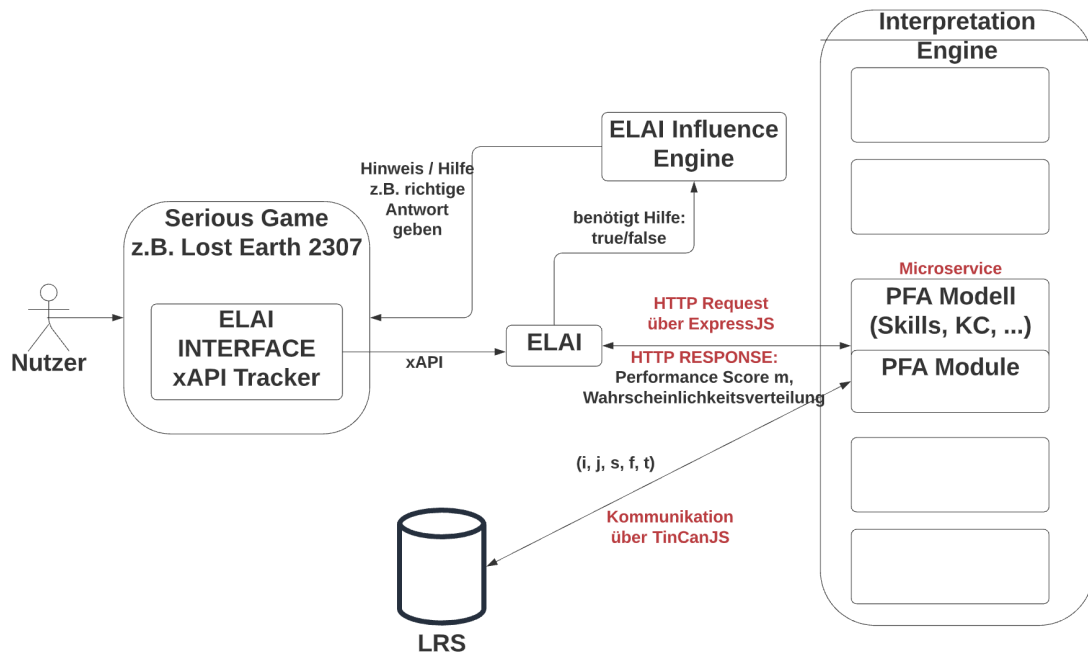


Abbildung 4.1: Architekturdiagramm: in Rot die Kommunikation

## 4.1 Entwurf des adaptiven Assistenten

Im Rahmen dieser Arbeit ist Streamlined Lost Earth (SLE) das Serious Games, in dem der adaptive Assistent und das PFA Modul implementiert werden. Der intelligente Tutor für SLE heißt LISA. LISA wird in Unity 19.2.21 entwickelt. Dazu wird die bereits vorliegende Arbeit von Marlene Dillig in SLE aufgegriffen und diese um weitere Funktionalitäten erweitert. Die Entwicklung von LISA findet in der objektorientierten Programmiersprache C-Sharp (C#) statt. Um im SG Streamlined Lost Earth Adaptivität zu gewährleisten, müssen folgende Schritte durchgeführt werden:

- Erfolge und Misserfolge des Nutzers bei einer Mission müssen festgehalten werden, um den „Performance Score“ zu berechnen.
- Eine Anfrage an das PFA Modul senden.
- Den „Performance Score“ aus dem JSON Format extrahieren.
- Abhängig von dem „Performance Score“ Hinweise für den Nutzer anzeigen.

In der Arbeit von Marlene Dillig werden die Ergebnisse der Mission des Nutzers nicht gespeichert. Deshalb musste sowohl das Konzept als auch die Implementierung von Marlene

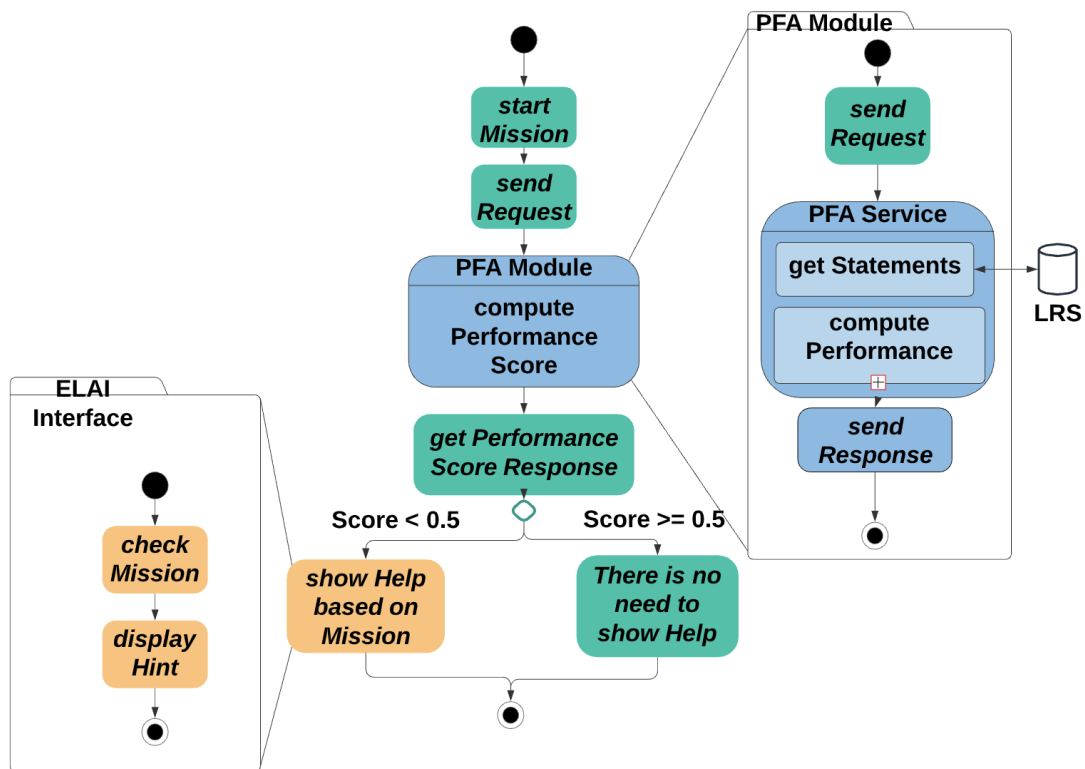


Abbildung 4.2: Aktivitätsdiagramm ab dem Start um dem Nutzer einen Hinweis anzuzeigen

Dillig durch diese Thesis erweitert werden. Im Rahmen dieser Arbeit werden xAPI (Abschnitt 3.4) Statements verwendet, um die Ereignisse des Nutzers festzuhalten zu können. Zuerst wird ein eindeutiger Identifier (Id) benötigt, um die Statements einem Nutzer zuweisen zu können. Dafür wurde bei Start des Spiels eine E-Mail Abfrage erstellt, wie in Abbildung 4.4 zu sehen ist. Anschließend wird bei jeder Antwort des Nutzers, egal ob Erfolg oder Misserfolg, ein xAPI Statement generiert und an den Learning Record Stores (LRS) gesendet.

Durch den Start einer Mission, wird eine Hypertext Transfer Protocol (HTTP) Anfrage mit der Methode GET an das PFA Modul geschickt. Dabei wird die Id mitgesendet, um die xAPI Statements des Nutzers zu erhalten. Danach muss der „Performance Score“ aus dem JSON Format ausgelesen werden. Wie in Abbildung 4.1 zu erkennen ist, greift LISA nur in dem Fall ein, dass der berechnete Wert kleiner als 0.5 ist. Basierend auf dem „Performance Score“ und der Mission in der sich der Nutzer befindet, wählt LISA einen Hinweis aus und zeigt ihn gegebenenfalls an. Wie in Abbildungen 4.5 und 4.6 zu sehen ist, wird hier die selbe Mission gespielt, jedoch wird ein Hinweis nur in einem der beiden Fälle angezeigt.

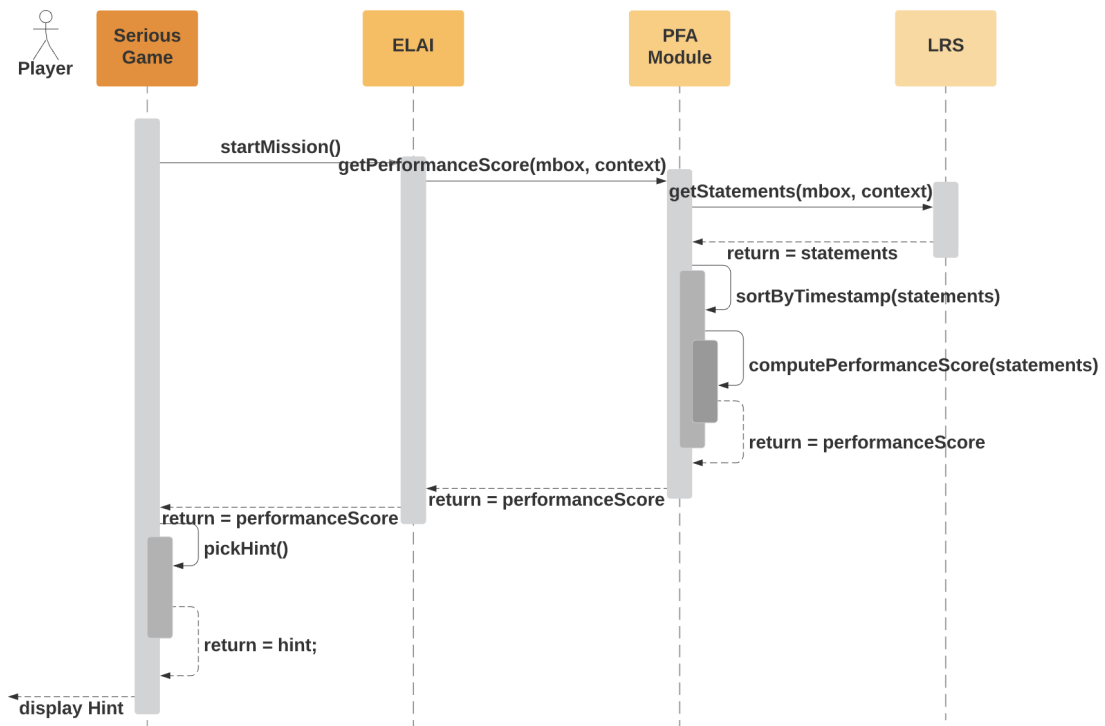


Abbildung 4.3: Sequenzdiagramm ab dem Start der Mission um dem Nutzer einen Hinweis anzuzeigen

## 4.2 Performance Factor Analysis Modellierung

Bevor der Entwurf des PFA-Moduls vorgestellt wird, muss eine Modellierung von PFA dargestellt werden. Wie bereits in Kapitel 3.5.4 erläutert, müssen für die Berechnung von PFA einige Strukturen, wie z.B. WK und Lernobjekte vorliegen. Um diese Strukturen zu modellieren, muss das charakterisierende Ziel des Serious Games (Abschnitt 3.1) vorliegen. Im Fall von SLE ist das charakterisierende Ziel die Wissenslehre im Bereich Bildauswertung. Sowohl für die LFA als auch die PFA muss das charakterisierende Ziel noch weiter in Fähigkeiten („Skills“) unterteilt werden. Solche Fähigkeiten werden auch Lernobjekte genannt. Diese Lernobjekte erhält man, indem man mithilfe von didaktischen Mitteln solche Fähigkeiten ausarbeitet. Meistens ist das die Aufgabe eines Lehrers in der Schule, eines Entwicklers bei einem Spiel oder eines Dozenten bei einer Vorlesung. Ein Beispiel dafür ist in Abbildung 3.14 zu sehen, bei dem die Lernobjekte in Faktoren aufgetrennt sind und ihnen Variablen zugewiesen werden. Ähnlich wie in Abbildung 3.14 wurde das Spiel Lost Earth 2307 in unterschiedliche Faktoren unterteilt, die in dieser Arbeit als Lernobjekte interpretiert werden können. Als Beispiel können wir



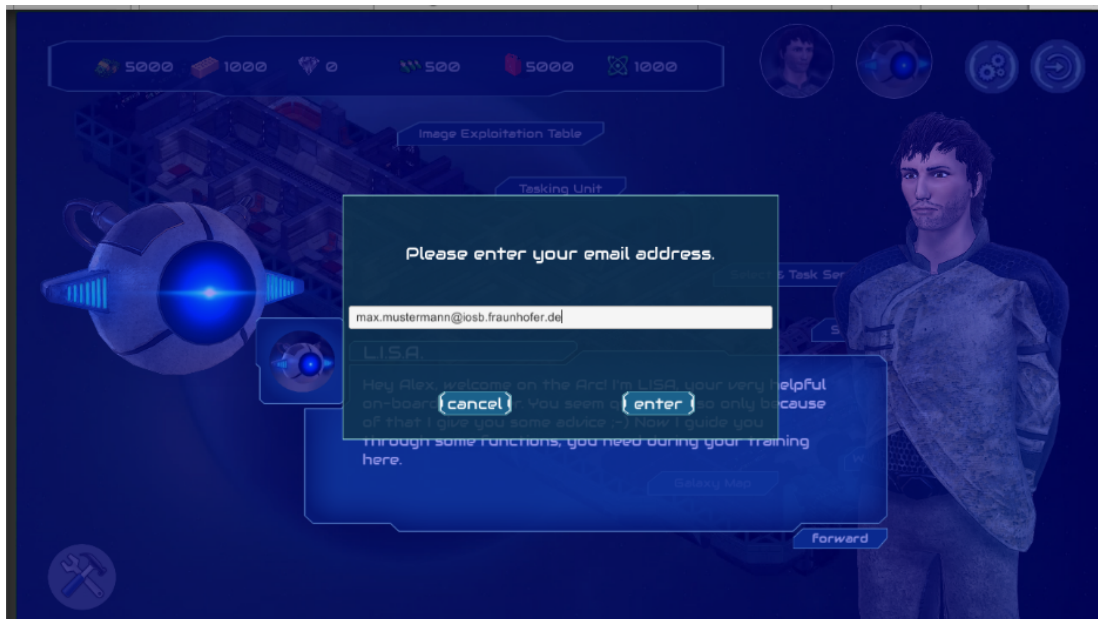


Abbildung 4.4: E-Mail Angabe für den Nutzer, um die Daten im LRS unter dem korrekten Nutzer zu sichern

die Lernobjekte aus Abbildung 4.7 betrachten. Um nun das Modell für die LFA und PFA zu vervollständigen, werden den Lernobjekten WK zugewiesen. Dabei kann ein Lernobjekt eine oder mehrere WK besitzen. Ähnlich wie für die Lernobjekte werden die WK über didaktische Mittel erarbeitet. Ein Beispiel für LE2307 ist ebenfalls in Abbildung 4.7 zu sehen.

Nachdem ein Modell für Lernobjekte und WK erstellt wurde, müssen diese Entitäten in das SG eingebunden werden, um später beim Nutzer abgefragt werden zu können. Das Ziel dabei ist, dass eine gestellte Frage an den Nutzer ein konkretes Lernobjekt abfragt. Beispielsweise werden in LE2307 die Fragen in Form von Missionen gestellt. Dann wird eine Mission angezeigt, bei der der Nutzer erkennen muss, dass es sich um ein optisches Bild handelt. Wie in Abbildung 4.7 zu sehen ist, hängen mit diesem Lernobjekt vier WK zusammen. Für jede Mission gibt es somit genau ein Lernobjekt, das abgefragt wird. An dieses Lernobjekt sind dann eine oder mehrere WK gekoppelt.

Beim Entwickeln eines Serious Games werden die Missionen mit verschiedenen Schwierigkeitsstufen implementiert. Somit kann man für jede Mission, und damit auch für jedes Lernobjekt, einen Schwierigkeitsgrad festlegen. Durch diese Modellierung erhalten wir für jede WK des Lernobjekts den Parameter  $\beta$  (aus Formel 3.3). Wie in Abbildung 4.8, wird für jede WK ein Verlauf erstellt, um die vorherigen Erfolge und Misserfolge der vorherigen Versuche für die Berechnung des aktuellen „Performance Score“  $m$  zu nutzen. Wie auch in Abschnitt

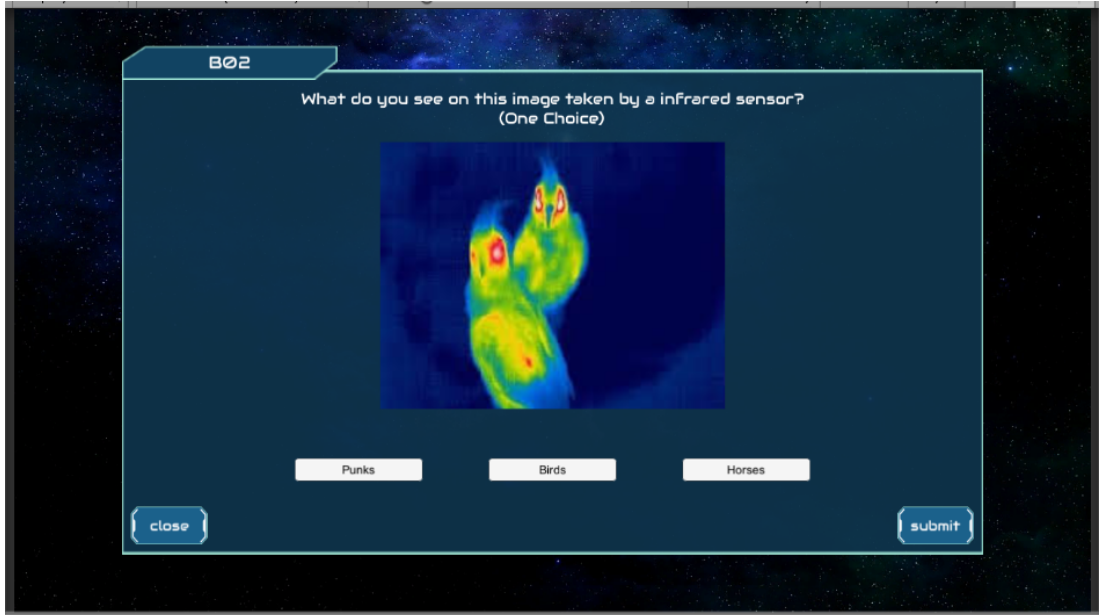


Abbildung 4.5: Mission 2b ohne einen Hinweis von LISA



Abbildung 4.6: Mission 2b mit einem Hinweis von LISA

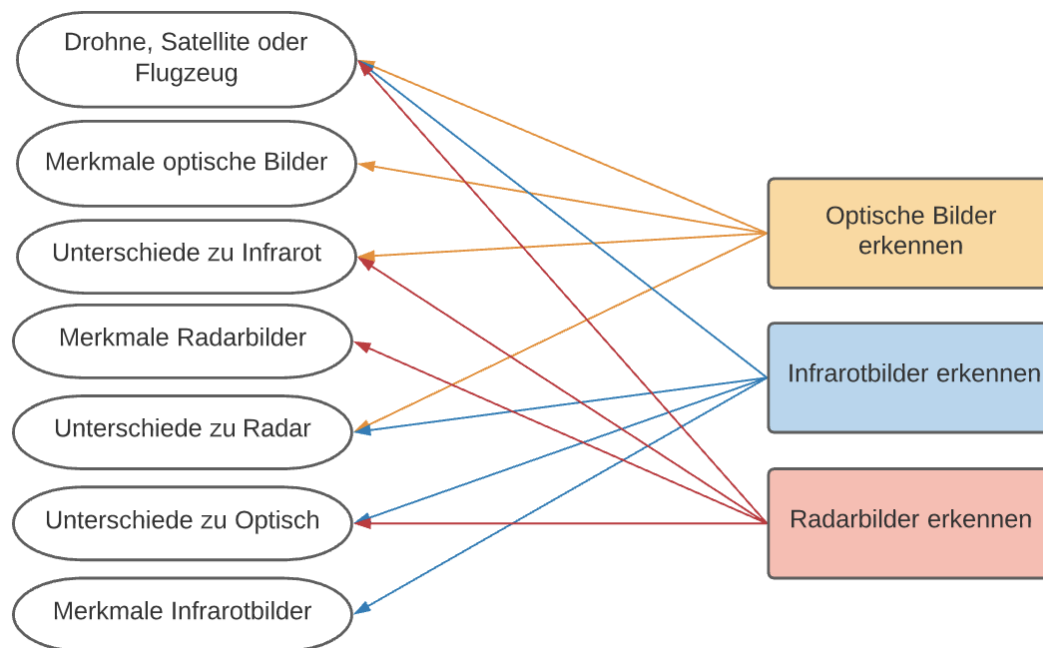


Abbildung 4.7: LE2307 wird in Lernobjekte (eckige Kästen) aufgeteilt und Lernobjekten werden Wissenskomponenten (Ovale) zugewiesen.

3.5.4 erläutert, kann für den Verlauf „Data Aging“  $\delta$  angewendet werden. Die beiden Variablen  $\gamma$  und  $\rho$  (Formel 3.3) sind jeweils Gewichtsfunktionen für den Erfolg und Misserfolg jeder WK. Falls bei einer Modellierung die Information über eine richtige oder falsche Antwort wertvoller ist, können die Funktionen  $\gamma$  oder  $\rho$  die Gewichtung der Antwort regulieren. Im Fall von SG SLE wurden sowohl richtige als auch falsche Antworten gleich gewichtet ( $\gamma = \rho = 1$ ).

Wie im wissenschaftlichen Schreiben von [Gong, Beck, et al. 2010] wurden unterschiedliche Ansätze des PFA Modells evaluiert. Ein schwer zu interpretierendes Ergebnis war, dass das sogenannte „All Skill“ PFA Modell besser als das klassische Modell abgeschnitten hat. Bei der „All Skill“ PFA werden nicht nur die WK des Lernobjekts in der Berechnung für  $m$  verwendet, sondern alle vorhandenen WK. Dieses Ergebnis erfordert jedoch noch weitere Überprüfungen und weitere Versuche, um eine Schlussfolgerung aus diesem Ergebnis ziehen zu können.

### 4.3 Entwurf des PFA-Moduls

Das PFA Modul wird in WebStorm 2020.3.1 als JavaScript (JS) Microservice entwickelt. Das Modul soll anhand von vorhandenen xAPI Statements einen „Performance Score“ mittels Perfor-

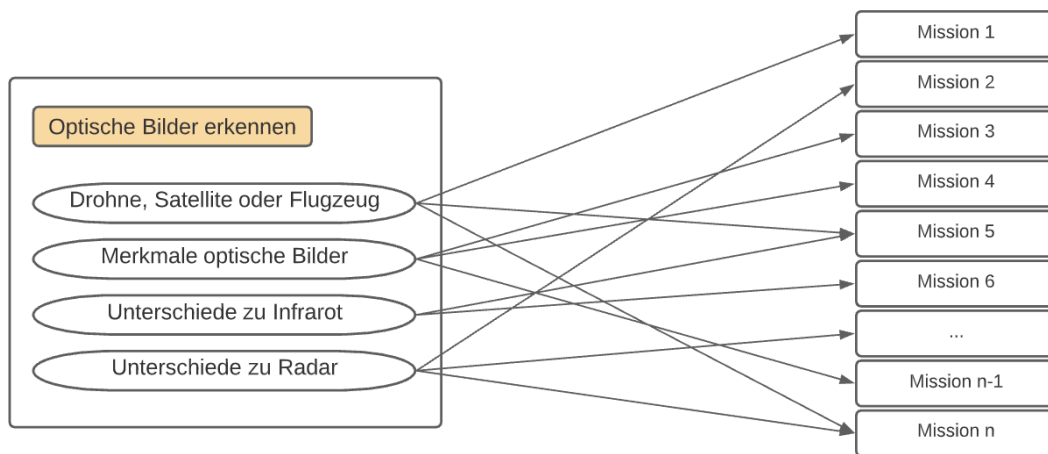


Abbildung 4.8: Die Wissenskomponenten des Lernobjektes „optische Bilder erkennen“ verweisen auf vorherige Missionen mit gleichen WK, um aus dem Verlauf den „Performance Score“  $m$  zu berechnen.

mance Factor Analysis berechnen und diesen, mit zusätzlichen anderen Parametern, in einem JSON Format wiedergeben können. Um das zu erreichen, werden folgende Funktionalitäten benötigt:

- HTTP Anfrage mit benötigten Informationen wie Id und Lernobjekt oder Wissenskomponente erhalten.
- xAPI Statements aus einem Learning Record Stores abrufen und nach Id und Skill filtern.
- Basierend auf den Statements den „Performance Score“ und andere Parameter berechnen.
- Informationen in JSON Format speichern und zurück an das ELAI Interface LISA schicken.

Der PFA Microservice wurde entwickelt, um dem LISA aus Streamlined Lost Earth bei der Berechnung des „Performance Scores“ zu helfen. Das geschieht, indem die HTTP GET Anfrage mit der Nutzer Id und der abgefragten Fähigkeit an das PFA Modul weitergeleitet wird. Um alle nötigen Informationen für die PFA Berechnung zu erhalten, wurde eine Abbildung von xAPI Statements auf PFA Variablen ausgearbeitet. Diese Abbildung ist in folgender Tabelle zu sehen:

xAPI Parameter	PFA Variablen
Actor	Nutzer $i$
Verb	Ausgang der Aufgabe (Erfolg oder Misserfolg)
Object	Wissenskomponente oder Fähigkeit $j$
Result	Erfolg oder Misserfolg $s_{i,j}, f_{i,j}$
Timestamp	Decay Faktor $\delta$

Aus dem LRS werden alle Statements zum jeweiligen Nutzer abgefragt und sortiert. Eine Sortierung wird nach dem Zeitstempel durchgeführt, um das in Abschnitt 3.5.4 erläuterte Problem mithilfe von „Data Aging“ zu beheben. Anschließend wird mittels PFA der „Performance Score“ berechnet. Zuletzt werden der „Performance Score“ und weitere Parameter, wie z.B. Verfallsfaktor  $\delta$  oder die Wahrscheinlichkeitsverteilung von  $m$ , in dem JSON Format an das ELAI Interface zurückgesendet (Abbildung 4.9).

```
{
  user: 'mailto:test@test.de',
  relativePerformanceScore: 0.3710490011346698,
  performanceScore: '2.5460989291000002',
  probabilityDistribution: '0.9273109996845027',
  constants: { delta: 0.9, statementLimit: '100', fallbackLimit: 100 }
}
```

Abbildung 4.9: Beispiel für eine Antwort im JSON Format



## 5 Implementierung des Regressionsmodells

In diesem Kapitel wird die Implementierung, die im Rahmen dieser Arbeit entstanden ist, vorgestellt. Zu Beginn wurde ein Prototyp für die Performance Factor Analysis (PFA) Berechnung als Webanwendung erstellt. Diese Anwendung war der Proof Of Concept für diese Arbeit und wird in Abschnitt 5.1 erläutert. Anschließend wurde der intelligente adaptive Tutor LISA im SG Streamlined Lost Earth (SLE) um die benötigten Funktionalitäten (Abschnitt 4.1) ergänzt. Das wird in Abschnitt 5.2 dargestellt. Zuletzt musste der Prototyp erweitert und angepasst werden, um eine korrekte PFA Modellierung einzurichten. Das wird in Abschnitt 5.3 ausgeführt.

### 5.1 Erster Prototyp

Für den Proof of Concept dieser Arbeit wurde ein PFA Prototyp entwickelt. Um diesen Prototyp als Microservice aufzurufen, wurde das Node.js-Framework ExpressJS verwendet [*Express - Node.js-Framework von Webanwendungen 2021*]. Die Verwendung der JavaScript Library TinCanJS stellt eine Verbindung zum Learning Record Stores (LRS) sicher [*RusticiSoftware/TinCanJS 2020*]. Mit den erhaltenen Informationen aus diesen beiden Services wurde anschließend ein Performance Score berechnet, der über eine Uniform Resource Locator (URL) mit Query-Strings abgefragt werden konnte.

#### 5.1.1 ExpressJS Web-Framework

Um das PFA-Modul als Microservice einzurichten, wurde die Webanwendung mit dem Node.js Framework ExpressJS entwickelt. Dieses Framework wurde verwendet, da es HTTP-Dienstprogrammmethoden anbietet und somit Anfragen an das Modul erleichtert. Im JS Projekt gibt es zwei Ordner. Der „api“ Ordner dient der Unterstützung der in dem Projekt verwendeten ExpressJS API und der „service“ Ordner ist für die Logik des Projektes, im Fall dieser Arbeit die „Performance Score“ Berechnung, zuständig.

In der „server.js“ wird der Server erstellt und mit dem Code aus Abbildung 5.1 werden die Routen für die URL zugeordnet. Wie in Codeabschnitt 5.2 zu sehen ist, werden hier zwei Endpunkte definiert: Erstens eine „about“ Sektion in der Informationen über das Modul gespeichert werden und zweitens ein „performance“ Endpunkt in dem das JSON Objekt mit

```

const express = require('express')
const app = express();
const port = process.env.PORT || 3000;

const routes = require('./api/routes');
routes(app);

app.listen(port, { callback: function () {
  console.log('Server started on port: ' + port);
}});

```

Abbildung 5.1: Die service.js Datei im PFA-Modul

```

'use strict';

const controller = require('./controller');

module.exports = function (app) {
  app.route('/about')
    .get(controller.about);
  app.route('/performance')
    .get(controller.getPerformance);
};

```

Abbildung 5.2: Die routes.js Datei im PFA-Modul

Berechnungsergebnissen gesichert werden. Mit diesem Aufbau ist es nun möglich, über eine URL mit Query-Strings auf den Microservice zuzugreifen.

### 5.1.2 JavaScript Library für Experience API: TinCanJS

Um den „Performance Score“ mittels Performance Factor Analysis zu berechnen, werden xAPI Statements benötigt. Diese Statements werden in sogenannten Learning Record Stores (LRS) gespeichert (Abschnitt 3.4). Um auf die Statements zugreifen zu können, muss erst eine erfolgreiche Verbindung sichergestellt werden. JavaScript bietet für die Implementierung von Experience API die TinCanJS Bibliothek an [RusticiSoftware/TinCanJS 2020]. In dieser Arbeit wurde die TinCanJS Version 0.50.0 von RusticiSoftware verwendet. Die Verbindung mit dem gewünschten LRS wird über einen TinCan Aufruf hergestellt. Falls die Verbindung fehlschlagen sollte, weil z.B. der Nutzernamen oder das Passwort nicht korrekt sind, wird über die Konsole eine entsprechende Fehlermeldung ausgegeben. Sobald die Verbindung zum LRS garantiert ist, können die gewünschten xAPI Daten abgefragt werden. Hierfür wurde eine extra Methode geschrieben. Beim Abfragen der Statements können die Statements nach bestimmten Eigenschaften gefiltert werden. Im Fall dieser Arbeit wird nach dem Nutzer gefiltert. Als eindeutiger Identifier wird die E-Mail des Nutzers verwendet. Die E-Mail wird hier als Dummy über einen Query-String in der URL mitgeliefert. Weiterhin wird die Anzahl der gehaltenen Statements gefiltert, damit die Abfrage von zu vielen Statements keine Verzögerung des Moduls und des verwendeten Serious Games hervorruft. Zu beachten ist, dass die Methode asynchron aufgerufen wird um mögliche Verzögerungen zu vermeiden. Es kann auch nach anderen Komponenten gefiltert werden, wie z.B. dem Kontext oder Objekten. Bei der Prototyp Implementierung wurde das Filtern nach WK bzw. Lernobjekten noch nicht getestet, da zu diesem Zeitpunkt noch nicht klar war, welche xAPI Statements von dem SG erstellt werden und somit ein Prototyp für die Implementierung von WK noch nicht benötigt wurde.



```
function computePerformanceScore(results) {
  let performanceScore = 0;
  let k = results.length - 1;
  for (let j = 0; j < results.length; j++) {
    // easiness, rho and gamma = 1
    performanceScore += results[j] * Math.pow(DELTA, k);
    k--;
  }
  return performanceScore;
}
```

Abbildung 5.3: Implementierung der PFA Formel

```
const json = {
  user: mbox,
  relativePerformanceScore: relativePerformanceScore,
  performanceScore: performanceScore.toString(),
  probabilityDistribution: probabilityDistribution.toString(),
  constants: {
    delta: DELTA,
    statementLimit: process.env.LRS_STATEMENT_LIMIT,
    fallbackLimit: FALLBACK_LIMIT,
  }
};
res.send(json);
```

Abbildung 5.4: Ein JSON Objekt mit den PFA Parametern und Informationen zum Nutzer wird erstellt

### 5.1.3 Berechnung von Performance Factor Analysis

Um die PFA Berechnung für den Performance Score durchzuführen zu können, mussten die xAPI Statements angepasst werden. Für die Nutzung des Decay Faktors aus Abschnitt 3.5.4, mussten die Statements nach ihrem Zeitstempel sortiert werden. Zwar bietet TinCanJS in der „getStatements“ Methode eine zeitliche Sortierung an, jedoch werden die Statements dann nach der Zeit sortiert, in der sie in das LRS eingegangen sind. Da Statements aber blockweise in das LRS geladen werden, entspricht diese Sortierung nicht der gewünschten Funktion. Daher werden die Statements erst anschließend nach ihrem Zeitstempel sortiert. Um die Berechnung für den „Performance Score“ im Prototypen zu vereinfachen und weil zu diesem Zeitpunkt noch nicht ersichtlich war, wie die Ergebnisse der xAPI Statements aus dem Spiel aussehen werden, wurden die Ergebnisse binär interpretiert. Bei *success = true* war die Mission erfolgreich und 1 wurde als Ergebnis abgespeichert. Als Misserfolg mit dem Ergebnis 0 wurde abgespeichert bei *success = false*. Für die „Performance Score“ Berechnung wurde anschließend die Funktion in Abbildung 5.3 verwendet. Es wird die Wahrscheinlichkeitsverteilung (Sigmoid Funktion, Abschnitt 3.5) berechnet und zuletzt ein JSON aus Berechnungen und Parametern zurückgegeben (Abbildung 5.4).

## 5.2 Implementierung des adaptiven Assistenten

Die in der Arbeit von Marlene Dillig begonnene Erweiterung der Funktionalität von LISA wurde in dieser Arbeit fortgesetzt. Die bisherige Funktionalität ermöglichte dem Nutzer auf eine Spielanleitung zuzugreifen, in welcher der nächste Schritt des Spiels angegeben wurde. Zu dieser Funktionalität wird nun bei jedem Missionsstart eine Anfrage an das PFA-Modul geschickt. Sobald die Antwort ankommt, wird geprüft, ob der aktuelle „Performance Score“

kleiner als 0.5 ist und je nachdem wird dem Nutzer in der Mission ein Hinweis angezeigt. In den folgenden Absätzen werden die neuen Funktionen des intelligenten adaptiven Assistenten LISA vorgestellt:

### **Aufzeichnung von Ergebnissen in Mission**

Mithilfe von TinCan.NET von RusticiSoftware kann, ähnlich wie im vorherigen Kapitel beschrieben, eine Verbindung zum LRS erstellt werden. Anders als zuvor werden hier die xAPI Statements jedoch nicht ausgelesen, sondern erstellt. Hierfür wird ein eindeutiger Identifier benötigt. Im Fall dieser Arbeit wird die E-Mail Adresse verwendet (Abbildung 4.4). Beantwortet der Nutzer eine Mission richtig oder falsch, wird ein xAPI Statement mit der Missionsbezeichnung, dem entsprechenden Verb („passed“ oder „failed“) sowie dem Ergebnis erstellt und an das LRS gesendet. Da die Implementierung der Missionen, zu Beginn dieser Arbeit, nur die Möglichkeit bereitstellten entweder richtig oder falsch zu antworten, wurde das Ergebnis im Statement auch nur als 0 (Falsch) oder 1 (Richtig) abgespeichert. Diese generierten xAPI Statements sind nun die Wissensbasis über den Nutzer, die für die PFA-Berechnung notwendig sind.

### **Verbindung zum PFA-Modul**

Mit RestSharp, der REST API Client Bibliothek für .NET, wird auf den Performance Factor Analysis Microservice zugegriffen [*restsharp/RestSharp 2021*]. Es wird eine HTTP GET Anfrage an das PFA-Modul versendet. Die Antwort enthält dann den benötigten „Performance Score“, anhand dessen beim Start einer Mission geprüft wird, ob der Nutzer einen Hinweis angezeigt bekommt oder nicht.

### **Anzeigen von Hinweisen**

Da in der vorherigen Implementierung noch keine Hinweise für die Missionen erstellt wurden, wurde LISA in dieser Arbeit um Hinweise erweitert. Dafür musste für jede vorhandene Mission ein individueller Hinweis erstellt werden. Dabei wurde ein Panel für den Hinweis erstellt und das LISA Logo mit einem Tipp eingepflegt. Der erstellte Hinweis enthält die richtige Antwort auf die Frage. Abbildungen 5.5 und 5.6 zeigen ein Beispiel für eine Mission in SLE, einmal mit und einmal ohne Hinweis.

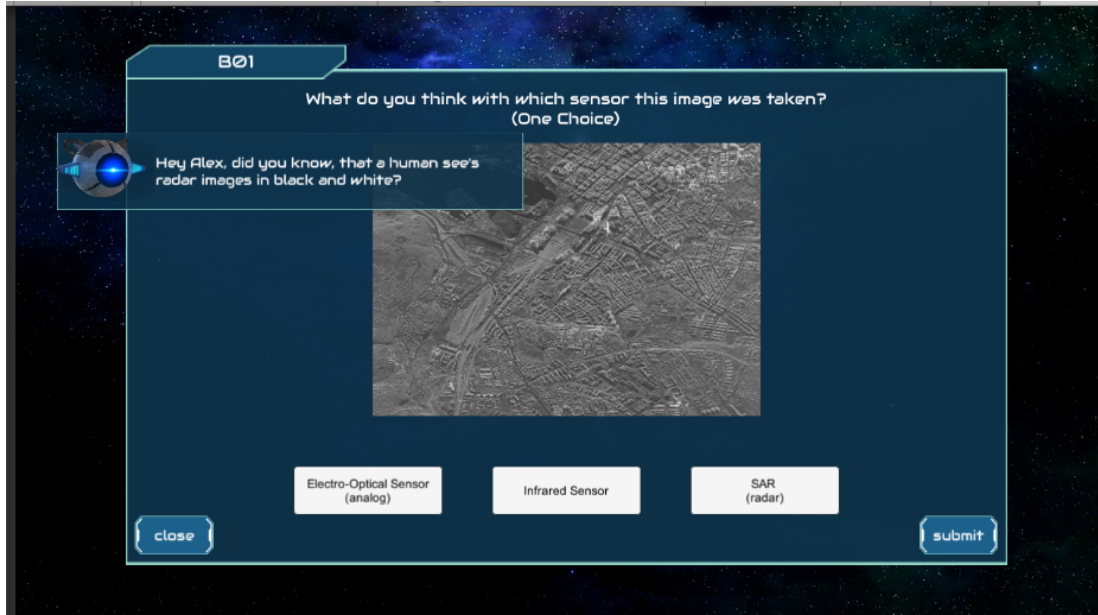


Abbildung 5.5: Eine Sensor Frage mit einem Tipp von LISA

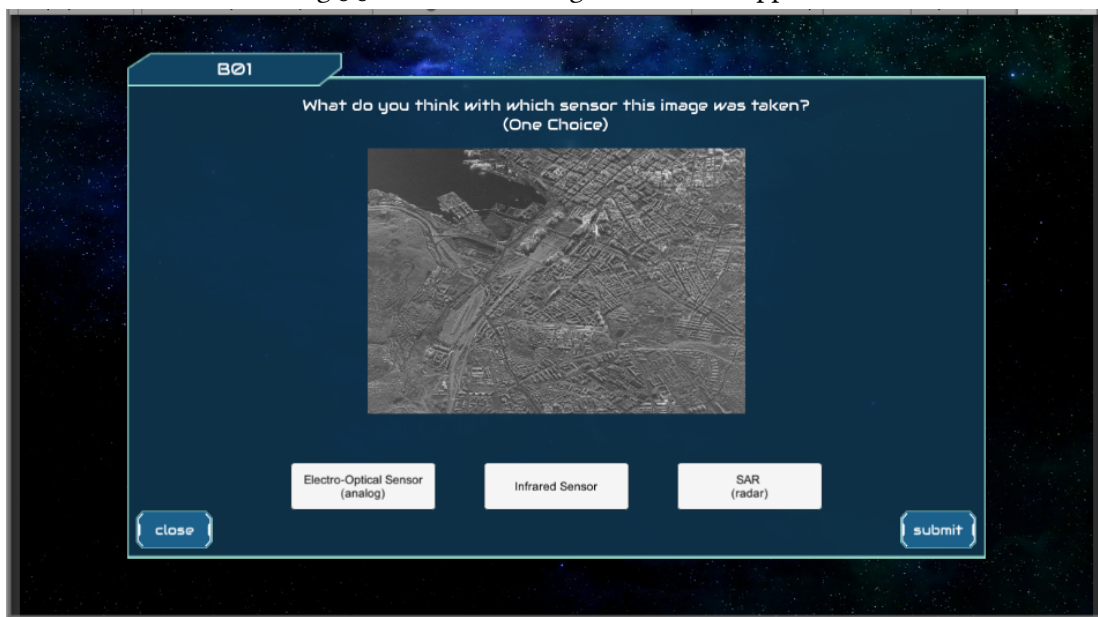


Abbildung 5.6: Eine Sensor Frage ohne einen Tipp von LISA

### Selbsteinschätzung abgeben

Die Nutzerstudie (Abschnitt 6.2) verlangt eine Selbsteinschätzung der Nutzer. Hierfür wurde eine Likert-Skala mit fünf Kategorien implementiert (Abbildung 5.7). Wie in der Abbildung zu sehen, wird der Nutzer nach jeder erfolgreich abgeschlossenen Mission nach seiner Selbsteinschätzung gefragt. Für die Abfrage gilt folgende Skalierung:

Kategorie	Skalierungswert
sehr schlecht	0
schlecht	0.25
mittel	0.5
gut	0.75
sehr gut	1

Mittels dieser Implementierung wird in der Nutzerstudie eine Korrelation zwischen dem „Performance Score“ und der Selbsteinschätzung des Nutzers hergestellt.



Abbildung 5.7: Selbsteinschätzung nach erfolgreichem Abschluss der Mission

## 5.3 Implementierung des PFA-Moduls

Das PFA-Modul wurde implementiert, um für LISA einen „Performance Score“ zu berechnen und diesen mit einigen anderen Informationen, wie z.B. E-Mail des Nutzers und Decay Faktor,

```
const LRS = {  
  ... ENDPOINT: process.env.LRS_ENDPOINT,  
  ... USERNAME: process.env.LRS_USERNAME,  
  ... PASSWORD: process.env.LRS_PASSWORD,  
};  
const FAILED = process.env.FAILED;  
const PASSED = process.env.PASSED;
```

Abbildung 5.8: Systemvariablen werden über Umgebungsvariablen abgerufen

```
function sortByTimestamp(statements) {  
  return statements.sort((curr, next) => {  
    return new Date(curr.timestamp) - new Date(next.timestamp);  
  });  
}
```

Abbildung 5.9: Codeausschnitt, um die geladenen Statements nach ihrem Zeitstempel zu sortieren

in einer JSON zurückzugeben. Dabei bildet der in Abschnitt 5.1 beschriebene Prototyp die Basis für den Performance Factor Analysis Microservice. Das PFA-Modul bietet folgende Funktionalitäten:

### Laden und Sortieren von xAPI Statements

Mit dem PFA-Modul können, genau wie bei dem Prototyp auch, alle xAPI Statements geladen werden. Der LRS muss dabei über Systemvariablen aufgerufen werden. Dafür werden die Laufzeitparameter mittels Umgebungsvariablen übergeben, wie in Abbildung 5.8 zu sehen ist. Dabei wird die URL über LRS\_ENDPOINT, der Nutzernamen über LRS\_USERNAME und das Passwort über LRS\_PASSWORD abgerufen. Identisch zum Prototyp wird ein Limit für die Anzahl an geladenen Statements festgelegt, um so Verzögerungen zu verhindern. Das Limit wurde per Voreinstellung auf 100 gesetzt und die Statements werden ähnlich wie im JS Projekt geladen. Nach dem Laden werden die Statements, wie in Abbildung 5.9 gezeigt, nach ihrem Zeitstempel sortiert.

### Berechnung des „Performance Scores“ mittels Performance Factor Analysis

Wie bereits in Abschnitt 5.2 erläutert, wurden die Mission und die dazugehörigen xAPI Statements so konzipiert, dass die Mission entweder ein Erfolg oder Misserfolg ist. Das bedeutet, dass im Rahmen dieser Arbeit Ergebnisse nur binär gewertet werden können. Außerdem sind die Missionen alle unabhängig voneinander und zeigen keine gemeinsam benötigten Fähigkeiten oder Wissenskomponenten auf. Daher ist die Implementierung des Prototyps (Abbildung 5.3) für einen „Performance Score“ ausreichend. Um einen passenden Decay Faktor zu bestimmen, wurde für diese Arbeit die Information aus dem wissenschaftlichen Schreiben von [Gong, Beck, et al. 2010] verwendet. Dort wird ein Decay Faktor von  $\delta = 0.9$  ausgewählt, weil  $\delta = 1$  klassische Performance Factor Analysis bedeuten würde und ein niedrigerer Faktor ( $\delta < 0.9$ ) ältere Antworten als deutlich weniger wichtig interpretieren würde. Da der berechnete „Performance

Score“ einen Wert  $m$  im Intervall  $m \in [0, \infty]$  ausgibt, aber ein Wert  $0 \leq m \leq 1$  erwartet wird, wird ein relativer „Performance Score“ wie in Abbildung 5.10 berechnet.

```
function computeMapping(performanceScore, resultLength) {  
  ... let sum = 0;  
  ... for (let k = 0; k < resultLength; k++) {  
    ... sum += Math.pow(DELTA, k);  
  }  
  ... return performanceScore / sum;  
}
```

Abbildung 5.10: Der „Performance Score“ wird normalisiert.

## 5.4 Diskussion

In diesem Kapitel wird darüber diskutiert, inwieweit der Entwurf aus Kapitel 4 in der Implementierung dieser Arbeit realisiert wurde. Ähnlich wie in den vorherigen Kapiteln wird zuerst LISA als Komponente und anschließend das PFA-Modul besprochen.

### Intelligenter adaptiver Assistent

Nach der Implementierung dieser Arbeit ist LISA in der Lage Ereignisse, mittels xAPI Statements aufzuzeichnen und in einem LRS zu speichern. Es werden alle Ergebnisse der Missionen aufgezeichnet, was es für zukünftige Arbeiten ermöglicht, diese Daten zu verwenden. Die Komponente ist durch die bereits vorhandene Anbindung an den LRS leicht durch weitere Experience API Aufzeichnungen, wie z.B. benötigte Zeit für eine Mission, erweiterbar. LISA kann ebenfalls erfolgreich auf das PFA-Modul zugreifen und den „Performance Score“ verwenden, um das SG Streamlined Lost Earth adaptiv zu gestalten. Diese Funktion kann im Hauptmenü durch einen Knopfdruck auch ausgeschaltet werden.

Das Anzeigen von Hinweisen wird auch von LISA gewährleistet. Für jede Mission existiert nun ein Hinweis. Zwar wurde für jede Mission nur ein Hinweis implementiert, es kann jedoch durch die Anpassung des Ereignisses einfach erweitert werden. Somit ist es vertretbar, dass in dieser Arbeit nur ein Hinweis als Demonstration implementiert wurde. Zusätzlich wurde die Aufzeichnung der Selbsteinschätzung ermöglicht, was von großem Interesse für andere, darauf aufbauende Themen oder die Weiterführung dieser Arbeit ist.

### **PFA-Modul**

Der PFA Microservice kann eine Anfrage von LISA entgegennehmen und mit Hilfe der weitergeleiteten Informationen einen „Performance Score“ berechnen. Dafür müssen xAPI Statements geladen und sortiert werden. Anhand der Ergebnisse dieser Statements kann das Modul den Score mittels Performance Factor Analysis berechnen und diesen, zusätzlich mit anderen Parametern, als JSON Objekt zurücksenden. Somit wurden alle Funktionalitäten aus Abschnitt 4.3 implementiert.

Die Implementierung der PFA Berechnung beruht nicht ganz auf der State of the Art-Methodik und kann daher verbessert werden. Damit das Modul sowohl für das Serious Games als auch die Nutzerstudie funktioniert, wurde die Implementierung einfach gehalten. Die fehlenden Parameter und Faktoren (z.B. Wissenskomponente) sollten, sobald das Spiel Streamlined Lost Earth um einige Missionen und Zusammenhänge erweitert wurde, in zukünftigen Arbeiten erweitert werden. Dennoch ist die Erkenntnis aus dem wissenschaftlichen Schreiben von [Gong, Beck, et al. 2010], dass das klassische PFA schlechtere Ergebnisse liefert als das „All-Skill“-PFA, welches alle WK und Fähigkeiten verwendet. Aufgrund dieses Ansatzes wurde in dieser Arbeit nicht nach WK gefiltert, sondern alle Fähigkeiten wurden für die Analyse verwendet.





## 6 Evaluation

In diesem Kapitel wird gezeigt, dass die gewünschte Adaptivität des Lernspiels Streamlined Lost Earth (SLE) mithilfe des adaptiven Assistenten LISA erzielt wurde. Im ersten Abschnitt wird ein Anwendungsszenario aus dem Spiel SLE vorgestellt und die einzelnen Adaptivitätsmechanismen werden aufgezeigt. Im darauf folgenden Abschnitt wird die im Rahmen dieser Arbeit entstandene Nutzerstudie vorgestellt. Jeder Nutzer spielt sich dabei durch das Anwendungsszenario. Daraufhin wird der berechnete „Performance Score“ mit der Selbsteinschätzung des Nutzers verglichen, um eine relative Güte zwischen den Werten herstellen zu können.

### 6.1 Anwendungsszenario

In diesem Anwendungsszenario spielt Max Mustermann einen Probedurchlauf von Streamlined Lost Earth (SLE). Max startet das Spiel und hat direkt im Hauptmenü die Möglichkeit das Spiel entweder mit PFA Berechnung und Hinweisen zu spielen oder diese Zusatzfunktion auszuschalten, wie in Abbildung 6.1 zu sehen ist.

Max lässt die Zusatzfunktion aktiv und startet das Spiel. Direkt beim Start gibt der Nutzer seine E-Mail Adresse an (siehe Abbildung 4.4). Nachdem Max seine E-Mail Adresse angegeben hat, findet er sich in der Raumschiffansicht (Abbildung 3.5) wieder. Hier erklärt der Tutoring Assistent LISA, was das Spielziel ist und wie Max Missionen starten kann. Max klickt nach den Hinweisen auf die Galaxy Map (Abbildung 6.2), um dort seine erste Mission auf Magento zu starten.

Nach einer kurzen Einweisung von Katrina startet Max die Mission auf Magento. Beim Start der Mission wird der aktuelle Performance Score von Max berechnet. Falls dieser unter einem Schwellenwert liegt (hier 0.5), wird dem Nutzer ein Hinweis bei der Mission angezeigt. Max folgt weiterhin den Anweisungen von LISA und kehrt zur Raumschiffübersicht zurück. Hier klickt Max nun auf die Select und Task Sensor Konsole (siehe Abbildung 6.3), um geeignete Bilder für die Mission zu erhalten.

Einen kurzen Moment später erscheint ein Hinweis im Image Exploitation Table, dass die Aufnahmen eingetroffen sind. Max klickt auf den Image Exploitation Table. Wie in Abbildung 6.4 zu sehen, erscheint eine Multiple Choice Frage mit einem zusätzlichen Hinweis von LISA.



Abbildung 6.1: Der Startbildschirm von SLE

Alternativ dazu ist in Abbildung 6.5 die Aufgabe ohne Hinweis zu sehen.

Der Hinweis bei der ersten Mission erscheint aufgrund des Kaltstartproblems. Das Kaltstartproblem taucht auf, weil noch keine Nutzerdaten über Max vorliegen. Falls Max die Frage falsch beantwortet, wird als xAPI Statement eine inkorrekte Antwort abgespeichert. Falls Max die Frage richtig beantwortet, wird die korrekte Antwort im Learning Record Stores (LRS) gespeichert. Nachdem Max Mustermann die Mission erfolgreich abgeschlossen hat, erhält er eine Belohnung.

Im Anschluss an die Mission wird Max nach seiner Selbsteinschätzung zur abgeschlossenen Mission gefragt (Abbildung 5.7). Die Selbsteinschätzung von Max wird ebenfalls als xAPI Statement im LRS gespeichert.

Um die nächste Mission auf Verdan zu starten, muss Max erst im Shop ein weiteres Transportmittel kaufen. Dabei hilft LISA weiter mit einigen Hinweisen. Sobald sich Max Mustermann auf Verdan befindet, werden ihm die nächsten drei Missionen von Thalos erklärt (Abbildung 6.6).

Beim Start der zweiten Mission wird erneut der aktuelle Performance Score berechnet, mit dem Unterschied, dass nun bereits Daten von der ersten Mission vorliegen. Anhand des berechneten Wertes entscheidet sich, ob Max Mustermann für den zweiten Auftrag ebenfalls einen Hinweis erhält. Max klickt auf die zweite Mission und es erscheint die Aufgabe, die in den Abbildungen 5.5 und 5.6 zu sehen ist. Die Ergebnisse von Max Antworten werden ebenfalls



Abbildung 6.2: Karte auf der alle Planeten angezeigt sind, die der Nutzer erreichen kann

als xAPI Statements im Learning Record Stores gespeichert. Nach erfolgreichem Abschluss der Mission folgt erneut die Abfrage nach der Selbsteinschätzung. Dieser Vorgang wiederholt sich ebenfalls für die darauf folgenden Missionen auf Verdän (Abbildungen 4.6, 4.5, 6.7, 6.8). Nach Abschluss der letzten Mission auf Verdän sowie nach Angabe der Selbsteinschätzung ist Max Mustermann fertig mit dem Probedurchlauf.

## 6.2 Nutzerstudie

In diesem Abschnitt wird die im Rahmen dieser Arbeit entstandene Nutzerstudie näher betrachtet. Zuerst wird die Hypothese der Studie erläutert. Anschließend wird der Aufbau der Studie dargelegt und zum Abschluss werden die Ergebnisse der Studie vorgestellt.

### 6.2.1 Hypothese

Um die Hypothese zu formulieren, werden eine Annahme und eine These benötigt. Wie bereits in Abschnitt 3.1.2 erläutert, hat Adaptivität eines Serious Games (SG) Einfluss auf den Lernerfolg eines Nutzers. Aufbauend auf dieser Annahme wurde ein „Performance Score“ berechnet. Dieser „Performance Score“ wurde mittels PFA (Abschnitt 3.5.4) bestimmt. Die These dazu lautet, dass der Gebrauch dieses berechneten „Performance Scores“ die Adaptivität



Abbildung 6.3: Select und Task Sensor Konsole zum Auswählen der Sensoren

an den Nutzer verbessert. Aus diesen beiden Formulierungen lässt sich folgende Hypothese herleiten:

Die Selbsteinschätzung des Nutzer korreliert mit dem „Performance Score“.

Für diese Studie wird die ihm Rahmen dieser Arbeit eingebaute Selbsteinschätzung verwendet. Dabei wird die Korrelation mittels mathematischer Methoden bestimmt. Konkret wird die Bravais Pearson Korrelation verwendet [Pearson Korrelation 2021]. Für die Berechnung der Pearson Korrelation wird die Formel 6.1 verwendet.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \quad (6.1)$$

### 6.2.2 Aufbau

Für den Aufbau der Nutzerstudie wird das SG Streamlined Lost Earth (SLE) verwendet. Die Studie fand im Januar 2021 statt. Auf Grund der, zu diesem Zeitpunkt vorliegenden COVID-19 Situation findet die Studie über Online-Meetings statt. Um den Prozess zu erleichtern und nicht jedem Nutzer die benötigte Software ausliefern zu müssen, handelt es sich um eine geführte online Studie. Für die Durchführung wird die Programmierumgebung Unity mit



Abbildung 6.4: Eine Multiple Choice Frage mit einem Tipp von LISA

der Version 19.2.21 verwendet. Dabei spielt der Leiter der Studie genau einen Probedurchlauf des Anwendungsszenarios aus Abschnitt 6.1 und gibt dem Nutzer bei jeder Mission und bei jeder Selbsteinschätzung über Bildschirmfreigabe die Möglichkeit seine Antworten selbst zu wählen. Es gibt insgesamt 10 Nutzer. Alle Nutzer sind in der Altersgruppe 20-45 Jahre und aus unterschiedlichen Arbeitsbereichen wie beispielsweise Studenten, wissenschaftliche Mitarbeiter, Lehrer, Ingenieure usw. Der „Performance Score“ wird, wie bereits erklärt, mittels PFA berechnet. Nach jeder erfolgreich abgeschlossenen Mission wird der Nutzer nach seiner Selbsteinschätzung gefragt. Die Selbsteinschätzung wird in fünf gleichverteilte Kategorien ( $[0;1]$ ) eingeordnet, wie in Abbildung 5.7 zu sehen. Dieses Konzept basiert auf dem Verfahren zur Messung von persönlichen Einstellungen, der sogenannten Likert-Skala [Likert-Skala 2021]. In dieser Arbeit wurden die in [Guy et al. 1977] und [Garland 1991] empfohlenen fünf Kategorien ausgewählt. Die Anzahl an Kategorien wurde ungerade gewählt, weil es dem Nutzer die Möglichkeit auf eine neutrale Position gibt und somit zu keiner Tendenz zwingt, so wie es bei der geraden Anzahl an Kategorien der Fall wäre. Die Nutzerdaten werden bereits über xAPI Statements im Spiel generiert. Nach jeder Mission wird die Korrelation zwischen aktuell berechnetem „Performance Score“ und Selbsteinschätzungen mittels Bravais Pearson Korrelation berechnet.





Abbildung 6.5: Eine Multiple Choice Frage ohne einen Tipp von LISA

### 6.2.3 Ergebnisse

In diesem Abschnitt werden die Ergebnisse der Nutzerstudie vorgestellt und anschließend mögliche Fehler und Verbesserungen diskutiert.

In Abbildung 6.9 sind die Ergebnisse der der Nutzerstudie zu sehen. Das Anwendungsszenario wurde mit insgesamt 10 Probanden durchgeführt. Auf der X-Achse sind die Probanden aufgezählt. Jedem Probanden sind zwei Balken zugewiesen. Auf der Y-Achse wird der Performance Wert im Intervall  $[0,1]$  angegeben. Der blaue Balken zeigt den mittels PFA berechneten „Performance Score“ und der orangene Balken die Selbsteinschätzung des Nutzers an. Der Nutzer hat insgesamt vier Missionen gespielt und die Werte in Mission  $i$  entsprechen dem „Performance Score“ und der Selbsteinschätzung, welche bis zur  $i$ -ten Mission akkumuliert wurden.

Nach der Durchführung des Szenarios mit allen Probanden, wurde zwischen dem „Performance Score“ und der Selbsteinschätzung die Pearson Korrelation ermittelt. In der folgenden Tabelle sieht man für jede Mission die dazugehörige Korrelation:



Abbildung 6.6: Die Missionsübersicht auf Verdan

Mission	Pearson Korrelation
1	0.35
2	0.80
3	0.84
4	0.81

Es ist eine positive Korrelation zu erkennen. Dabei nimmt die Korrelation mit jeder Mission zu, flacht jedoch nach Mission vier ab.

Es war zu erwarten, dass Mission eins die schlechteste Korrelation aufweist, da für die PFA Berechnung anfangs keine Daten über den Nutzer vorliegen und sich die erste Antwort damit sehr stark auf die Berechnung auswirkt. Dieses Problem ist in der Informatik als das Kaltstartproblem bekannt. Zusätzlich gibt es bei Mission eins Proband 7 einen Messfehler. Der Proband ist farbenblind und hat durch diese Störung der Farbwahrnehmung die Mission nur zufällig richtig beantwortet, seine Leistung jedoch als sehr schlecht eingeschätzt. Seine Begründung war, dass seine Antwort nicht auf seinem Wissen sondern auf Zufall basierte und er damit sehr schlecht in dieser Mission abgeschnitten hat. In den nächsten Missionen steigt die Pearson Korrelation auf 0.8 und mehr an. Das weist darauf hin, dass es einen linearen Zusammenhang zwischen der Selbsteinschätzung und dem „Performance Score“ gibt. Im folgenden Absatz werden noch mögliche Verbesserungen und Erweiterungen der Evaluation



Abbildung 6.7: Ein Wimmelbild mit einem Tipp von LISA

vorge stellt.

## Diskussion

Im Rahmen der Evaluation wurde deutlich, dass Mission vier besser untersucht werden kann. Nach der Implementierung von Marlene Dillig, die das Spiel SLE entwickelt hat, ist Mission 4 das Wimmelbild (Abbildung 6.8). Das Ergebnis der Mission ist entweder Erfolg oder Misserfolg. So gut wie alle Probanden haben die Mission korrekt gelöst, aber viele Probanden haben unterschiedlich lange gebraucht, um die Gegenstände auf dem Bild zu erkennen. Ein Verbesserungsvorschlag wäre eine Messung der Zeit für diese oder auch andere Missionen durchzuführen und in die Berechnung miteinzubeziehen. Damit könnte man nicht nur Erfolg und Misserfolg trennen, sondern auch wie lange eine Person für eine Mission gebraucht hat. Ein weiterer Vorschlag für die Evaluation der Ergebnisse ist eine andere Methode der Auswertung. In dieser Arbeit wurden die Ergebnisse der Missionen immer aufeinander akkumuliert und dann verglichen. Es wäre interessant zu sehen, wie sich die Korrelation ändert, wenn man die Missionen individuell betrachtet. Zusätzlich wäre es interessant zu sehen, wie sich die Selbsteinschätzung ändert und damit ggf. die Korrelation, wenn anstelle einer fünf Kategorie Likert-Skala eine sechs oder sieben Kategorie Likert-Skala eingesetzt wird. Denn eine Likert-Skala mit gerader Anzahl an Kategorien würde den Nutzer zu einer Tendenz zwingen. Durch





Abbildung 6.8: Ein Wimmelbild ohne einen Tipp von LISA

die höhere Anzahl an Kategorien könnte auch die Selbsteinschätzung genauer werden.

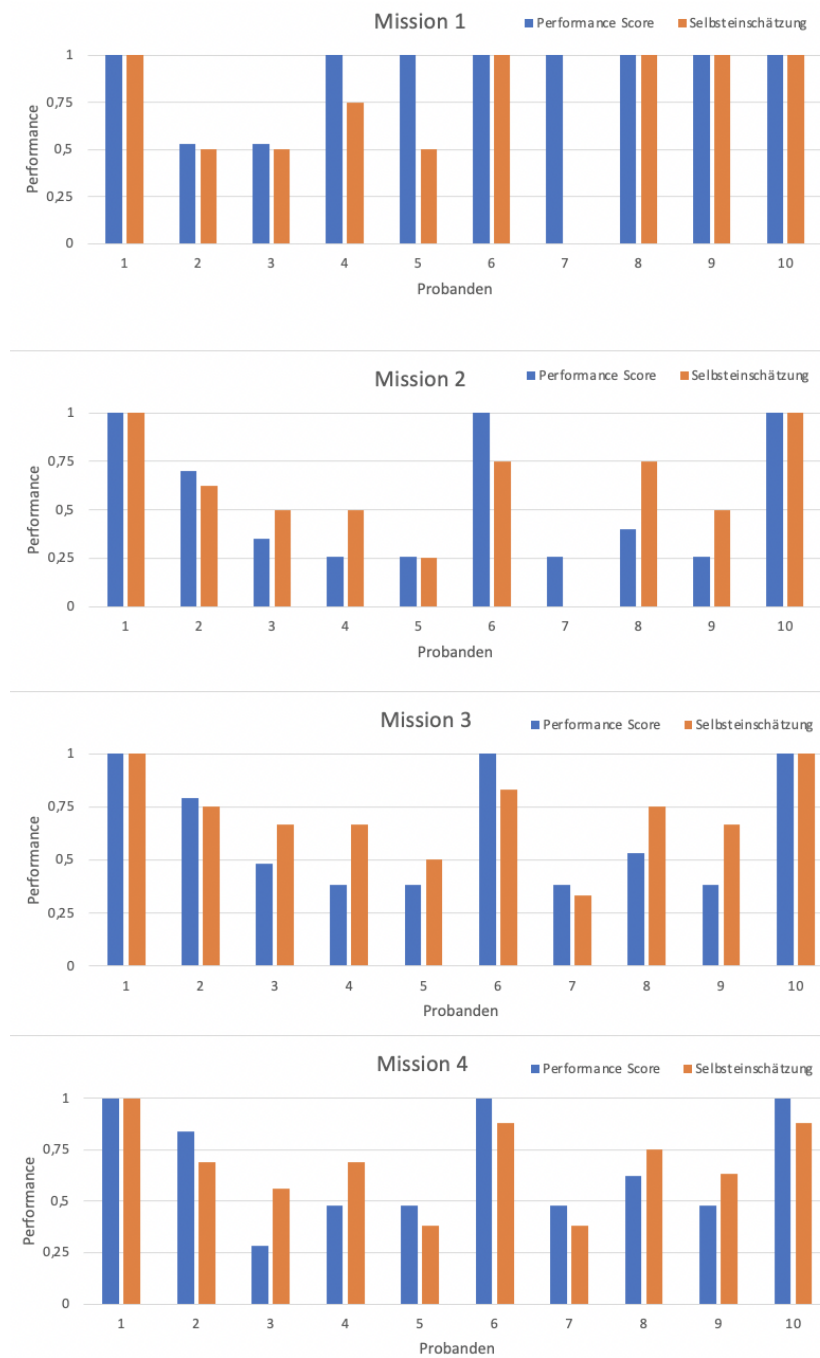


Abbildung 6.9: Die Ergebnisse der Nutzerstudie: in blau der berechnete „Performance Score“ und in orange die Selbstschätzung des Nutzers

## 7 Fazit und Ausblick

Das übergeordnete Ziel dieser Arbeit war es, die Adaptivität des Serious Games zu verbessern. Konkreter war der Plan in dieser Thesis einen „Performance Score“ für einen Nutzer in einem SG zu bestimmen, um diesen für die Lernfortschrittoptimierung zu nutzen. Hierfür wurden drei unterschiedliche Methoden der Datenanalyse vorgestellt: Knowledge Tracing, Learning Factor Analysis und Performance Factor Analysis. Diese Modelle wurden miteinander verglichen und im Rahmen dieser Arbeit daraufhin festgelegt, das PFA Modell für die Konzeption und Implementierung zu verwenden. Dieser Entschluss wurde gefasst aufgrund der Vorteile von PFA gegenüber LFA und KT. Im Gegensatz zu KT kann PFA auch einen „Performance Score“ für komplexe Fragen mit mehreren Wissenskomponenten berechnen kann. LFA kann das zwar auch, aber dafür kann LFA den „Performance Score“ nicht für einen konkreten Nutzer berechnen, so wie es in PFA der Fall ist.

Anschließend musste eine Modellierung von PFA auf SG allgemein ausgearbeitet werden. Darüber hinaus sollte eine praktische Implementierung des PFA Modells realisiert werden, die dem vorgestellten Konzept folgt und eine Evaluation ermöglicht. Um das zu realisieren, musste zusätzlich das ELAI-Framework des Spiels Streamlined Lost Earth erweitert werden. Dafür musste das ELAI-Framework um die Datenerfassung des Nutzerverhaltens durch xAPI Statements erweitert werden. Zusätzlich wurde das Interface um Hinweise für die Missionen ergänzt, um dem Nutzer bei einem niedrigen „Performance Score“ Hilfestellung zu geben. Für die Nutzerstudie wurde noch eine Abfrage nach der Selbsteinschätzung implementiert.

Um einem einheitlichen Standard bei der Modellierung von Nutzerverhalten zu folgen, wurde das xAPI Protokoll vorgestellt und verwendet. Die notwendigen Funktionen für den intelligenten Tutor Assistenten LISA, einschließlich der Aufnahme von Nutzerverhalten und der Überprüfung des „Performance Scores“, wurden alle umgesetzt. Zusätzlich wurde eine Abfrage über die Selbsteinschätzung des Nutzers nach jeder Mission implementiert.

Das PFA Modul wurde als Microservice in ExpressJS entwickelt. Bei der Implementierung wurde das klassische PFA Modell um weitere Features ergänzt. Bei diesen Features handelt es sich um den Decay Factor, der zusätzlich Datenalterung berücksichtigt und die „All Skill“-PFA, bei der alle Wissenskomponenten des Spiele berücksichtigt werden. Dennoch mussten einige Anpassungen an der Modellierung getroffen werden, da z.B. das SG SLE nicht durch konkrete

WK erweitert werden konnte.

Abschließend wurde eine Nutzerstudie durchgeführt, bei der jeder Nutzer ein Anwendungsszenario gespielt hat. Mit Hilfe der Nutzerstudie konnte eine positive Korrelation zwischen Selbsteinschätzung und dem „Performance Score“ berechnet werden. Die Korrelationen für die vier Missionen aus dem Anwendungsszenario waren (0.35, 0.80, 0.84, 0.81). Die niedrige positive Korrelation aus Mission 1 konnte mit dem Kaltstart des Spieles begründet werden. Damit konnten alle gesetzten Ziele und Anforderungen innerhalb des vorgegebenen Zeitraums erfüllt werden.

In späteren Arbeiten gäbe es die Möglichkeit, den adaptiven Assistenten LISA zu verbessern, indem die Datenerfassung des Nutzerverhaltens erweitert und optimiert wird. Interessant wäre zu sehen, ob sich aus einer größeren Datenmenge ein präziserer „Performance Score“ berechnen lässt. Ein Problem dabei könnte die Annahme sein, dass bestimmte Parameter, wie z.B. Zeit, einen Einfluss haben, der aber nicht vorhanden ist und deswegen das Vorhersagemodell überangepasst wird. Weiterhin könnte das ELAI Interface durch zusätzliche Adaptivitätsantworten erweitert werden. Im Rahmen dieser Arbeit wird dem Nutzer, falls der „Performance Score“ entsprechend niedrig ist, ein Hinweis angezeigt. Möglicherweise lässt sich der Lernfortschritt durch mehrere Hinweismöglichkeiten optimieren. Mit dem berechneten „Performance Score“ würde, anhand von unterschiedlichen Grenzwerten, ein anderer Hinweis angezeigt werden.

Darüber hinaus könnten im PFA Modell Erweiterungen realisiert werden. Um eine Korrelation für den „Performance Score“ zu berechnen, wurde die Selbsteinschätzung verwendet. Hier wäre es interessant zu sehen, ob es eine Möglichkeit gibt, die Selbsteinschätzung für die PFA Berechnung zu verwenden. Beispielsweise könnte sich eine Abfrage nach der Selbsteinschätzung am Anfang des Spiels positiv auf das Kaltstartproblem auswirken. Damit würde die Selbsteinschätzung als initialer Wissenstand fungieren. Weiterhin könnte die Selbsteinschätzung als ein Hinweis für Berechnungsfehler, wie z.B. der „Guess“ oder „Slip“ Parameter bei KT, dienen. Daraus könnte eine weitere Nutzerstudie entstehen, in der das PFA Modell dieser Arbeit mit dem PFA mit Selbsteinschätzung verglichen wird, um zu erfahren, ob sich das Einbinden der Selbsteinschätzung positiv auf die Korrelation auswirkt.

## Literatur

- Andersen, Erik (2012). „Optimizing Adaptivity in Educational Games.“ In: *Proceedings of the International Conference on the Foundations of Digital Games*. FDG '12. Raleigh, North Carolina: Association for Computing Machinery, pp. 279–281. URL: <https://doi.org/10.1145/2282338.2282398>.
- Baker, Ryan SJ d et al. (2008). „More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing.“ In: *International conference on intelligent tutoring systems*. Springer, pp. 406–415.
- Beck, Joseph E. et al. (2007). „Identifiability: A Fundamental Problem of Student Modeling.“ In: *User Modeling 2007*. Ed. by Cristina Conati et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 137–146.
- Becker, Torsten et al. (2016). *Stochastische Risikomodellierung und statistische Methoden*. Springer.
- Cen, Hao, Kenneth Koedinger, et al. (2006). „Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement.“ In: *Intelligent Tutoring Systems*. Ed. by David Hutchison et al. Vol. 4053. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 164–175.
- (2008). „Comparing two IRT models for conjunctive skills.“ In: *International Conference on Intelligent Tutoring Systems*. Springer, pp. 796–798.
- Cen, Hao, Kenneth R. Koedinger, et al. (2007). „Is Over Practice Necessary? –Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining.“ In: *Proceedings of the 2007 Conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*. NLD: IOS Press, pp. 511–518.
- Chatterjee, Samprit et al. (2015). *Regression analysis by example*. John Wiley & Sons.
- Chen, Jenova (2007). „Flow in games (and everything else).“ In: *Communications of the ACM* 50.4, pp. 31–34.
- Corbett, Albert T et al. (1994). „Knowledge tracing: Modeling the acquisition of procedural knowledge.“ In: *User modeling and user-adapted interaction* 4.4, pp. 253–278.
- Costello, AB et al. (Jan. 2005). „Best Practices in Exploratory Factor Analysis: Four Recommendations for Getting the Most From Your Analysis.“ In: 10, pp. 1–9.

- Csikszentmihalyi, Mihaly (2009). *Flow*. HarperCollins.
- De Sande, Brett van (2013). „Properties of the Bayesian Knowledge Tracing Model.“ In: *Journal of Educational Data Mining* 5.2, pp. 1–10.
- Defining serious games - The primary objective is not entertainment* (Mar. 2016). en-US. URL: <https://flowleadership.org/serious-games/> (visited on 01/20/2021).
- Dörner, Ralf et al., eds. (2016). *Serious games: Foundations, Concepts and Practice*. Cham: Springer.
- Doroudi, Shayan et al. (2019). „Fairer but Not Fair Enough On the Equitability of Knowledge Tracing.“ In: LAK19. Tempe, AZ, USA: Association for Computing Machinery, pp. 335–339. URL: <https://doi.org/10.1145/3303772.3303838>.
- Express - Node.js-Framework von Webanwendungen* (2021). de. URL: <https://expressjs.com/de/> (visited on 01/23/2021).
- Falakmasir, Mohammad Hassan et al. (2013). „A Spectral Learning Approach to Knowledge Tracing.“ In: *EDM*, pp. 28–34.
- Frutos-Pascual, Maite et al. (2017). „Review of the Use of AI Techniques in Serious Games: Decision Making and Machine Learning.“ In: *IEEE Transactions on Computational Intelligence and AI in Games* 9.2, pp. 133–152.
- Garland, Ron (1991). „The mid-point on a rating scale: Is it desirable.“ In: *Marketing Bulletin*, pp. 66–70.
- Gong, Yue, Joseph E. Beck, et al. (2010). „How to Construct More Accurate Student Models: Comparing and Optimizing Knowledge Tracing and Performance Factor Analysis.“ In: 2010.
- Gong, Yue, Dovan Rai, et al. (Jan. 2009). „Does Self-Discipline impact students’ knowledge and learning?.“ In: pp. 61–70.
- Guy, Rebecca F. et al. (1977). „The Neutral Point on a Likert Scale.“ In: *The Journal of Psychology* 95.2, pp. 199–204.
- Introduction to PCA and EFA* (2021). URL: [https://rstudio-pubs-static.s3.amazonaws.com/363499\\_73a1c1a94da148b6ad81e6eb8dc1b771.html](https://rstudio-pubs-static.s3.amazonaws.com/363499_73a1c1a94da148b6ad81e6eb8dc1b771.html) (visited on 01/28/2021).
- Kevan, Jonathan M et al. (2016). „Experience API: Flexible, decentralized and activity-centric data collection.“ In: *Technology, knowledge and learning* 21.1, pp. 143–149.
- Kiili, Kristian (2005). „Digital game-based learning: Towards an experiential gaming model.“ In: *The Internet and higher education* 8.1, pp. 13–24.
- Koedinger, Kenneth R. et al. (2012). „The knowledge-learning-instruction framework: bridging the science-practice chasm to enhance robust student learning.“ In: *Cognitive science* 36.5, pp. 757–798.
- Lane, H. Chad (2012). „Cognitive Models of Learning.“ In: *Encyclopedia of the Sciences of Learning*. Ed. by Norbert M. Seel. Boston, MA: Springer US, pp. 608–610. URL: [https://doi.org/10.1007/978-1-4419-1428-6\\_241](https://doi.org/10.1007/978-1-4419-1428-6_241).

- Leszczenski, James M et al. (2007). „What’s in a word? Extending learning factors analysis to model reading transfer.“ In: *13th International Conference on Artificial Intelligence in Education, Educational Data Mining Workshop*.
- Liegle, Jens O et al. (2000). „Developing adaptive intelligent tutoring systems: a general framework and its implementations.“ In: *Proceedings of the ISECON Conference, Philadelphia*. Citeseer.
- Likert-Skala (2021). *Likert-Skala: Definition, Beispiel und Vorteile*. de. URL: <https://www.qualtrics.com/de/erlebnismanagement/marktforschung/likert-skala/> (visited on 01/27/2021).
- Lobo, Desmond et al. (Mar. 2010). „A New Procedure to Help System/Network Administrators Identify Multiple Rootkit Infections.“ In: pp. 124–128.
- Lost Earth 2307 - Fraunhofer IOSB (2021). de. URL: <https://www.iosb.fraunhofer.de/de/projekte-produkte/lost-earth-2307.html> (visited on 01/22/2021).
- Moon, T.K. (Dec. 1996). „The expectation-maximization algorithm.“ In: *Signal Processing Magazine, IEEE* 13, pp. 47–60.
- Pardos, Zachary et al. (June 2010). „T.: Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing.“ In: pp. 255–266.
- Pavlik Jr, Philip I. et al. (2009). „Performance Factors Analysis – A New Alternative to Knowledge Tracing.“ In: 2009.
- Pearson Korrelation (2021). *Pearson Korrelation: Berechnung und Interpretation*. de. URL: <https://studyflix.de/statistik/pearson-korrelation-1051> (visited on 01/27/2021).
- Perttula, Arttu et al. (2017). „Flow experience in game based learning—a systematic literature review.“ In: *International Journal of Serious Games* 4.1, pp. 57–72.
- Radha, R et al. (2020). „E-Learning during lockdown of Covid-19 pandemic: A global perspective.“ In: *International journal of control and automation* 13.4, pp. 1088–1099.
- restsharp/RestSharp (Jan. 2021). original-date: 2009-11-16T02:01:45Z. URL: <https://github.com/restsharp/RestSharp> (visited on 01/23/2021).
- Romero, C. et al. (2007). „Educational data mining: A survey from 1995 to 2005.“ In: *Expert Systems with Applications* 33.1, pp. 135–146. URL: <http://www.sciencedirect.com/science/article/pii/S0957417406001266>.
- Romero, Cristobal et al. (2020). „Educational data mining and learning analytics: An updated survey.“ In: *WIREs Data Mining and Knowledge Discovery* 10.3, e1355. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1355>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1355>.
- RusticiSoftware/TinCanJS (Dec. 2020). original-date: 2012-08-17T13:41:20Z. URL: <https://github.com/RusticiSoftware/TinCanJS> (visited on 01/23/2021).

- Sampayo-Vargas, Sandra et al. (2013). „The effectiveness of adaptive difficulty adjustments on students’ motivation and learning in an educational computer game.“ In: *Computers & Education* 69, pp. 452–462.
- Schultz, Sarah E et al. (2014). „Expanding Knowledge Tracing to Prediction of Gaming Behaviors.“ In: *EDM (Workshops)*.
- Shute, Valerie J. et al. (2012). „Adaptive Educational Systems.“ In: *Adaptive Technologies for Training and Education*. Ed. by Paula J. Durlach et al. Cambridge: Cambridge University Press, pp. 7–27.
- Streamlined Lost Earth* (2021). URL: <https://gitlab-ext.iosb.fraunhofer.de/seriousgames/streamlinedlosteearth/streamlinedlosteearth> (visited on 01/22/2021).
- Streicher, Alexander und Wolfgang Roller (2015). *Towards an Interoperable Adaptive Tutoring Agent for Simulations and Serious Games*.
- (2017). „Interoperable Adaptivity and Learning Analytics for Serious Games in Image Interpretation.“ In: *Data Driven Approaches in Digital Education*. Ed. by Élise Lavoué et al. Vol. 10474. Lecture Notes in Computer Science. Cham: Springer International Publishing, Imprint, and Springer, pp. 598–601.
- Streicher, Alexander, Wolfgang Roller, und Christian Biegemeier (2017). „Application of Adaptive Game-Based Learning in Image Interpretation.“ In: *European Conference on Games Based Learning*. Academic Conferences International Limited, pp. 975–978.
- Streicher, Alexander und Jan D. Smeddinck (2016). „Personalized and Adaptive Serious Games.“ In: *Entertainment Computing and Serious Games*. Ed. by Ralf Dörner et al. Vol. 9970. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 332–377.
- What is xAPI aka the Experience API or Tin Can API* (2021). en-US. URL: <https://xapi.com/overview/> (visited on 01/20/2021).
- Yadav, Vivek et al. (2020). „User Assistance for Serious Games Using Hidden Markov Model.“ In: *Addressing Global Challenges and Quality Education*. Lecture Notes in Computer Science 12315. Ed. by Carlos Alario-Hoyos et al., pp. 380–385. URL: [http://link.springer.com/10.1007/978-3-030-57717-9\\_31](http://link.springer.com/10.1007/978-3-030-57717-9_31) (visited on 09/23/2020).



---

## Glossar

*BKT* Bayesian Knowledge Tracing. 26, 27, 29,

*ELAI* E-Learning Artificial Intelligence. v, vii, 2-4, 15, 17, 20-22, 24, 39, 46, 47, 69, 70,

*HMM* Hidden Markov Modell. 26-29,

*HTTP* Hypertext Transfer Protocol. 41, 46, 49, 52,

*JS* JavaScript. 45, 49, 51, 55, 69,

*JSON* JavaScript Object Notation. 24, 25, 39-41, 46, 47, 49, 51, 55, 57,

*KT* Knowledge Tracing. v, vii, 2-4, 8-10, 12, 15, 22, 26-29, 32, 33, 69, 70,

*LE2307* Lost Earth 2307. 3, 7, 11, 20, 27, 39, 42, 43, 45,

*LFA* Learning Factor Analysis. v, vii, 2-4, 9-12, 15, 22, 25, 28-30, 32-35, 42, 43, 69,

*LRS* Learning Record Stores. 22, 23, 37, 39, 41, 43, 46, 47, 49-52, 55, 56, 60, 61,

*PFA* Performance Factor Analysis. v, vii, 2-5, 9, 11, 12, 15, 20, 22, 25, 26, 28, 29, 33-36, 39-43,  
45-47, 49-52, 54-57, 59, 61, 63-65, 69, 70,

*SG* Serious Games. v, vii, 2-4, 7-9, 12, 15-24, 27, 36, 39, 40, 42, 43, 45, 49, 50, 56, 57, 61, 62, 69,

*SLE* Streamlined Lost Earth. v, vii, 3, 5, 20, 27, 36, 39, 40, 42, 45, 46, 49, 52, 56, 57, 59, 60, 62, 66,  
69,

*URL* Uniform Resource Locator. 49, 50, 55,

*WK* Wissenskomponente. 3, 11, 12, 27, 29-35, 42, 43, 45-47, 50, 55, 57, 69, 70,

*xAPI* Experience API. 4, 21, 23-25, 37, 39, 41, 45-47, 50-52, 55-57, 60, 61, 63, 69,