User Assistance for Serious Games Using Hidden Markov Model

-

Design and Implementation of an Adaptive Framework

MASTER'S THESIS

Vivek Yadav

12 September 2020

User Assistance for Serious Games Using Hidden Markov Model

Thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Science

in

Computer Science

by

Vivek Yadav

(Matriculation Number: 11012177)

based on research carried out

under the supervision of

Prof. Dr. Ajinkya Prabhune

and

Dipl.-Inf. Alexander Streicher

12 September 2020

Applied Computer Science

SRH Hochschule Heidelberg

Applied Computer Science
SRH Hochschule Heidelberg

Prof. Dr. Ajinkya Prabhune
Professor, SRH Hochschule Heidelberg
Heidelberg, Germany

Dipl.-Inf. Alexander Streicher
Scientifc Researcher, Computer Scientist, Fraunhofer IOSB
Karlsruhe, Germany

12 September 2020

Supervisors' Certificate

This is to certify that the work presented in the thesis entitled *User Assistance for Serious Games Using Hidden Markov Model* submitted by *Vivek Yadav*, Matriculation Number 11012177, is a record of original research carried out by him under our supervision and guidance in partial fulfillment of the requirements of the degree of *Master of Science* in *Computer Science*. Neither this thesis nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

Prof. Dr. Ajinkya Prabhune	Dipl.-Inf. Alexander Streicher
Professor, SRH Hochschule Heidelberg, Heidelberg, Germany	Scientifc Researcher, Computer Scientist, Fraunhofer IOSB, Karlsruhe, Germany

Declaration of Originality

I, *Vivek Yadav*, Matriculation Number *11012177* hereby declare that this thesis entitled *User Assistance for Serious Games Using Hidden Markov Model* presents my original work carried out as a masters student of SRH Hochschule Heidelberg and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of SRH Hochschule Heidelberg or any other institution. Any contribution made to this research by others, with whom I have worked at SRH Hochschule Heidelberg, Fraunhofer IOSB or elsewhere, is explicitly acknowledged in the thesis. Works of other authors cited in this thesis have been duly acknowledged under the sections “Reference” or “Bibliography”. I have also submitted my original research records to the scrutiny committee for evaluation of my thesis.

I am fully aware that in case of any non-compliance detected in future, the Examination Committee of SRH Hochschule Heidelberg may withdraw the degree awarded to me on the basis of the present thesis.

September 12, 2020
SRH Hochschule Heidelberg

Vivek Yadav

Acknowledgment

I have been part of the journey that started six months ago with an idea and ended with a developed system. During this time, I got the opportunity to discover the excellent work performed by many individuals in eLearning, education, machine learning, controlled systems, and numerous others, which led to the evolution in my thinking and outlook in the respective fields.

I have received constant guidance from my supervisor, professor, and peers, for which I am very thankful to them. I want to announce that, after the completion of this thesis, I want to take this concept and its constituents ahead and perform further research in the future.

(1) SRH Hochschule Heidelberg
(2) Fraunhofer IOSB, Karlsruhe

Vivek Yadav
Matriculation No: 11012177

Abstract

Serious Games, i.e., games not just for pure entertainment and with characterizing goals, are gaining huge popularity for the purpose of education and training. To further increase the learning outcome of serious games, assistance functionalities like adaptive systems observe the users and try to guide them to achieve their learning objectives.

The research question is how to model the user's behavior, their progress, and how to determine the best adaptation strategies to motivate the users and provide assistance whenever required. Using experience-data in a serious game is one approach to develop and train models for adaptivity. This thesis presents SeGaAdapt, an adaptive framework that is based on a Hidden Markov Model (HMM) algorithm for providing dynamic user-assistance and learning analytics for a serious game. For the development and training of the HMM, the system is provided with activity streams or user interaction data gained from an Experience API (xAPI) tracker.

The adaptivity mechanism uses the HMM model to analyze the current state of the user (player) in order to predict the best probable hidden-state or event pertaining to the assistance result. Technical verification of the implementation shows the feasibility of the approach and hints at future research directions. The evaluation concept includes testing with synthetic data and the simulator for serious game.

***Keywords*: *serious games*, *xapi*, *adaptivity*, *hidden markov model*.**

Kurzfassung

Serious Games, d. h. Spiele nicht nur zur reinen Unterhaltung und mit charakteristischen Zielen, gewinnen zum Zweck der allgemeinen und beruflichen Bildung an Popularität. Um das Lernergebnis von Serious Games weiter zu verbessern, beobachten Unterstützungsfunktionen wie adaptive Systeme die Benutzer und versuchen, sie zum Erreichen ihrer Lernziele zu führen.

Die Forschungsfrage lautet, wie das Verhalten des Benutzers und sein Fortschritt modelliert werden können und wie die besten Anpassungsstrategien ermittelt werden können, um die Benutzer zu motivieren und bei Bedarf Unterstützung zu leisten. Die Verwendung von Erfahrungsdaten in einem Serious Game ist ein Ansatz, um Modelle für Anpassungsfähigkeit zu entwickeln und zu trainieren. In dieser Arbeit wird SeGaAdapt vorgestellt, ein adaptives Framework, das auf einem Hidden Markov Model (HMM) -Algorithmus basiert, um dynamische Benutzerunterstützung und Lernanalysen für ein Serious Game bereitzustellen. Für die Entwicklung und Schulung des HMM werden dem System Aktivitätsströme oder Benutzerinteraktionsdaten bereitgestellt, die von einem Experience API (xAPI) -Tracker abgerufen werden.

Der Adaptivitätsmechanismus verwendet das HMM-Modell, um den aktuellen Status des Benutzers (Spielers) zu analysieren. Das Ziel ist den wahrscheinlichsten verborgenen Zustand oder das Ereignis im Zusammenhang mit dem Assistenzergebnis vorherzusagen. Die technische Überprüfung der Implementierung zeigt die Machbarkeit des Ansatzes und gibt Hinweise auf zukünftige Forschungsrichtungen. Das Bewertungskonzept umfasst das Testen mit synthetischen Daten und den Simulator für Serious Game.

***Keywords*: *serious games*, *xapi*, *adaptivity*, *hidden markov model*.**

Contents

Chapter 1

Introduction

The three types of games that are designed with the intent to educate are Commercial off-the-shelf (COTS) video games, Serious Games, and Student Designed Games [Simões, Redondo, and Vilas, 2013]. We are not concerned with student-designed games and will focus on the other two types. COTS video games and Serious Games are developed to deliver education; still, the focus of COTS lies more towards entertainment while serious games concentrate on the performance gap and are designed for specific user groups from the military, corporate, medicine, school, and various other [Becker and Gopin, 2016]. Serious games (SGs), i.e., games not just for pure entertainment and are designed for the purpose of education and training.

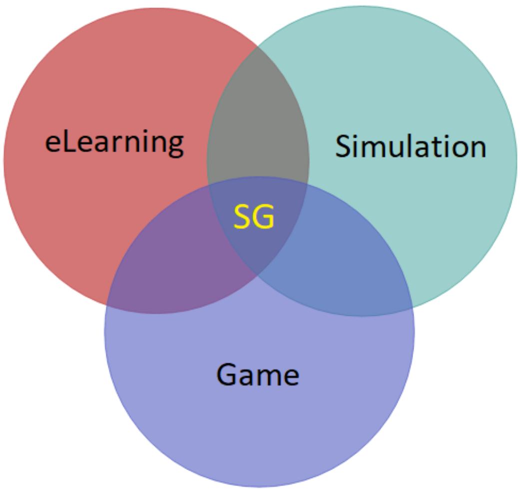

Figure 1.1: Attributes of Serious Games

SGs constitutes attributes such as eLearning, simulation and game, making them a useful tool to educate and training in a fun and interactive way [Papanastasiou, Drigas, and Skianis, 2017]. The importance of serious has widespread in the recent years, but the idea of the serious game as a tool for learning and training has a long history, and it existed even before 2000 when the serious game started becoming more prevalent, for instance, Clark C Abt published a book called ”Serious Games” in 1970 [Clark, 1970], and ”Kriegsspiel” was a very basic wargame with maps and painted blocks resembling an army created in the 18th century to teach the battlefield tactics to the soldiers of the Prussian army [Reisswitz, 1824], and even in the present times, similar kinds of games are used to train the military and defense personals. War games have extensive usage for the assessment of the military recruits in the US army [Allen, 2012].

SGs started becoming very popular, even though slowly, after the end of the previous millennium [Omelina et al., 2012]. With the increasing application of games in a non-gaming environment, game-based learning (GBL) is gaining a lot of prominences, In 2003, it was estimated that the education and training market has a global market share of about $2 trillion [Susi, Johannesson, and Backlund, 2007], which implicitly leads to a rise in GBL. As per the conducted research, there has been an influx in the number of users of SG every year with an estimation of the continuous upward trajectory for the use of SG. It was predicted in the paper 'Serious games: An overview' [Susi, Johannesson, and Backlund, 2007] "by 2008, 40 percent of U.S. companies will adopt serious games in their training efforts" and there have been efforts directed into the area towards identifying the relevance of the SGs in companies and their integration into a corporate environment for the training purpose [Azadegan and J. c. Riedel, 2012], and to support this claim, a survey has been conducted on various companies in UK [Azadegan, J. C. Riedel, and Hauge, 2012].

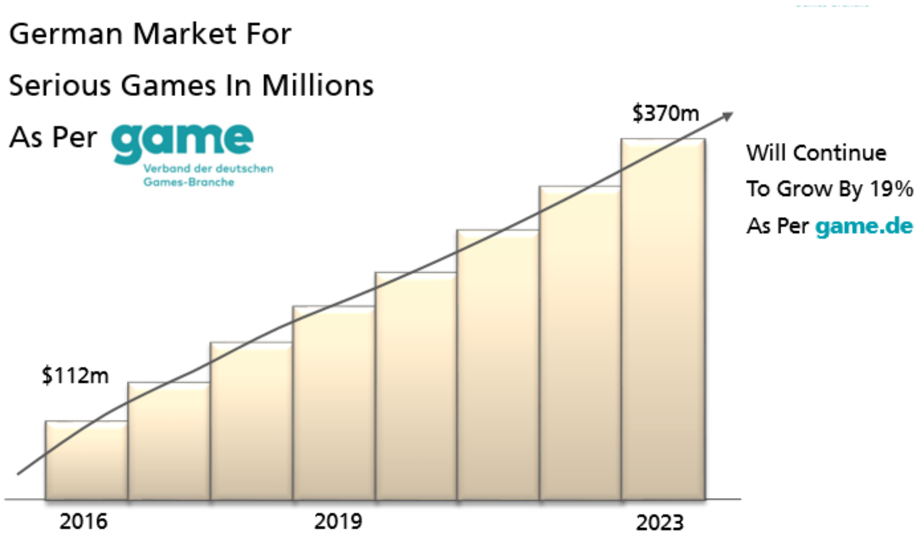

Figure 1.2: German Market for Serious Games in millions as per *game.de* [Hamdorf, 2019]

As per the research conducted by [Hamdorf, 2019] from *game.de* on the German market for SGs, SGs are ascending approximately 19 percent each year and will maintain it's rise until 2023. Preference of SG as a learning tool is more likely to increase based on it's illustration of positive impact and outcome towards the accretion of knowledge and understanding [Connolly et al., 2012].

A good game is the one which comprises of better user experience and user engagement [SERC, 2018]. While designing the SGs the user experience is an imitation of the core principals (Game Elements) with the variations based on the end-users' preferences

[0] Fig 1.3 is modification of graphical data presented by [Hamdorf, 2019] from [game.de, 2020]

and desires [Elaachak, Belahbibe, and Bouhorma, 2015]. Preferences and desires of the users are not easy to understand and draw because it involves a case study of the user but they are mostly static with minimum adaptation requirement, but in contrast, the user engagement with the SGs is quite dynamic, volatile, evolutionary and must adapt continuously as per the user. The consequential severity of the serious games signifies the importance of the gaming activity, therefore there is a lot of emphases devoted towards the betterment of the engagement process. One of the essential parts of the engagement process is assistance that can make the gaming experience more interactive and desirable [Brockmyer et al., 2009]. It is essential to perceive the users' understanding, knowledge, strength, weakness, agility, speed and the use of reflexes (conducted in Exercise therapy which involves hand-eye coordination) to model the state of a user at a given point in time and make use of it in order to assist the respective user whenever they are stuck or lost.

1.1 Motivation

The fields that leverage the use of serious games are education, health care, emergency management, city planning, engineering, defense and various other for the training, teaching, development and simulation purposes [Allbeck, 2018]. Let us consider the example of the 'Kriegsspiel' mentioned earlier (the war-game developed in 18th century), the historical data suggests that game was comprehensible to most of the player and a huge success, as a result, the other armies also created their own war-game for training their soldiers [Wintjes, 2015], but could be a possibility that the game could have been difficult for the group of people who posses a different way of learning and understanding a concept or a system, and due to their divergence to perceive something differently the probability of successful outcome out of the game could be undermined. "Von Altrock notwithstanding, this is not easy to understand as, after all, Georg Leopold von Reisswitz explicitly mentioned Opiz more than once as an important inspiration of his game" [Wintjes, 2015]. This undesired consequence could however be altered through another version of an artifact created after observation, investigation, and experiment conducted on the learning group(s). Despite how plausible this solution sounds, it may address the problem up to some extent at the generic level but at the granular there would still exist issues among most of the learning groups in one or the other way. In order to tackle this problem there must be a degree of dynamic transmutation of the components of serous games (SGs) according to the player. What is required here is 'missing piece of the puzzle' i.e. an adaptive mechanism within the game that amends its components like activities, game elements, difficulty level or contents that corresponds to the state of the player making the life easier for the player in order to accomplish the desired goal of playing the SGs. Being educational, entertaining, and continuously engaging poses great challenges on the design of adaptive SG. One way to tackle this challenge is to acquire an understanding of the person and model the components of SG in order to make the engagement process more suitable as well as educational.

1.1.1 Motivation for Activity Assistance

Activities in the SG are not like levels or stages in entertainment games, wherein the players (users) are driven by the challenges associated with pure entertainment. Rather, the activities are intended to teach, educate, train or guide, without the users losing interest.

Figure 1.3: Activities of Serious Games have Positive Outcomes

The success of SG relies a lot on the activity engagement and later on the outcomes produced from the engagement process. The engagement feature would encourage loyalty if there is element of excitement that can be achieved through adaptation, which would deem for gaining a successful outcome from the game-play [Simões, Redondo, and Vilas, 2013]. The engagement process within SGs involves performing activities which are associated to certain task(s). These activities are performed upon the entities within the SG, for instance the *Carry out search using Alpha Task Force behind enemy lines*. Here, the task of *searching* can be declared as an activity while *Alpha Task Force* and *enemy lines* are entities. Therefore, if the player is facing difficulty to search behind the enemy lines and moving away from the target then an adaptive solution could be: trigger alert or preferably provide the appropriate guidance to the player till some extent which is closer to the target. Adaptive and personalized SGs could lead to better outcomes from the playing exercise [Streicher, Bach, and Roller, 2019] supported by Machine Learning (ML) but there is still lot of room for enhancing the adaptivity through the implementation of AI [Streicher and Smeddinck, 2016].

1.2 Objectives and Goals

Serious games (SGs) have a lot of potential for providing a substantial outcome (2.1.1), but as a whole, an entity can drive its' operation successfully if all its sub-components collaborate with each other and work efficiently to provide significant results at their atomic level. One of the important integral parts is a user engagement, and this thesis is intended to enhance it through the application of Machine Learning. The objective of this thesis is to provide user assistance as an adaptive solution (achieving adaptivity) for the serious games to cater to the need of an enriched and effective learning experience during the activity engagement process customized according to the personal capability. The goal of this thesis is to provide an adaptive framework for the SGs based on Hidden Markov Model (HMM) called SeGaAdapt. The visible and hidden states of gamification are modeled by Hidden Markov Model (HMM) by using the data of the activities performed by the players. HMM is used to analyze the current state of the player and provide the best probable assistance result.

In this thesis, we are aiming to modify the aspects of the SGs that can help the player to achieve successful outcome such as, customization of the activities, or modification of the difficulty, or providing the instructions in form of suggestion/tip according to the players' performances. While performing an activity within the serious game, the outcome of the activity is using as *Experience API* (xAPI) statements [ADL.net, 2013; Lim, 2015], which in turn are used by the HMM to discover if the player is on wrong track. If the player is indeed not on the right track then the assistance for that particular activity is provided by SeGaAdapt. If the player is facing difficulty, suggesting that the player is indeed not on the right track, the upcoming activities are using the adaptive mechanism.

1.3 Overview

The background chapter presents the fundamental knowledge about the topics to understand relevant concepts before proceeding to the chapters with advanced conceptualization. The background chapter consists of information about serious games, Lost Earth Serious Game, Experience API (xAPI), Hidden Markov Model, and adaptivity.

Related work deals with the research and development performed by individuals who have provided their valuable contribution to the concepts relevant to this thesis. It also sheds light towards the state of the art technology concerning the components and operations employed in the thesis's proposed solution such as xAPI, Hidden Markov Model (HMM), and adaptivity.

In the methodology chapter, we will discuss the techniques and methods used by the entities of the proposed framework. You will acquire a general understanding concerning the setup of components and their behavior when they are executed to perform a particular process. The methodology chapter expatiates the following segments in detail:

- Adaptivity and assistance
- Setup and customization of Simulator for Lost Earth Serious Game (SimLESG)
- Model development process
- Working of Compass used for the assistance process
- Adaptivity manifestation using HMM

The implementation chapter comprises the outlining of the leading technology used and how the various parts of the framework interact with one another to behave as a conglomerate entity. The implementation chapter expounds the following segments in detail:

- Implementation of SimLESG
- SeGaAdapt (proposed framework for user assistance) and its components
- Interaction of SimLESG with SeGaAdapt
- e-Learning using Artificial Intelligence (ELAI) Server
- Hidden Markov Model Micro-Service (ELAI-HMM)

The application chapter deals with the presentation of the functional components in action and proof of concept. The application chapter describes the use-cases along with mockups and the working implementation of:

- The Gameplay within SimLESG
- The behavior of models of HMM to identify the misfit activities and to provide assistance
- Adaptivity manifestation with the examples from the SimLESG
- Activity diagram of the functional components
- Role of the Compass in action

Chapter 2

Background

The introduction chapter has already explained about the importance of serious games (SG)s in the field of education and training and how they are becoming an important part in the present times. Therefore, a lot of emphases are directed towards the betterment of SGs, and here in Fraunhofer IOSB, a lot of effort is focused in the direction of improving the user experience. The problem statement and solution of this thesis is aiming for the improvement of user experience during interaction with SGs by providing the user assistance through adaptivity manifestation. Before we dive into theoretical and technical aspects that later come into play, it is essential to comprehend the basics such as SGs, Experience APIs, Hidden Markov Model and Adaptivity.

2.1 Serious Games

Passive learning can get boring sometimes [Bajak, 2014] and not everyone is able to comprehend the valuable information from the process due to the difference of intelligence level and variable in capability of grasping the conveyed knowledge. However, if the the passive learning is catalysed with interactive techniques [Bajak, 2014]. This would catch attention of most of the listeners but still there is no guarantee whether the learning objective has been achieved, due to the fact that only learning is insufficient until its application in the respective field is applicable is doubtful. Therefore, the lectures often take test to measure the scale of acquired knowledge. If the tests are taken right after the lecture, the result would directly proportional to the acquired knowledge but this is always not case. If tests are taken after sometime then it is would be difficult to judge whether the lecture were prudent or not because the self study coming into picture. If the students do not perceive the knowledge during the lectures and rely on studying later to understand the same concept then it is a sheer waste of time and energy, this puts burden on the learner. In case if the person is unable to perceive the concept it may result into loss of confidence which could lead to undesired outcome like failure and sometimes to give up learning the concept [Schäfer et al., 2013].

If case if the person is able to comprehend the concept, due to the tendency of human brains overwriting the unused stored information in our brain with the new information, it is not guaranteed that after sometime the same concept can be demonstrated with the same level of understanding when the concept was understood at first [Aljundi et al., 2018]. Therefore, many of us forget most of things that we learn or understand on daily basis even the important ones, if they are not revised or practices at some interval. There is an alternative to educate or train about the concepts that has more lasting impression in the memory, i.e. to store the information in form of story or an a place like mind palace [Yates, 1992]. This technique is quite well known and it is quite evident that it difficult to forget something which involves a personal experience.

2.1.1 Benefits

Serious games (SG)s can handle the problems mentioned previously. SGs being interactive in nature does not face the same problem which is faced while providing knowledge through the passive learning. The fun nature of SGs does not lead to the same of level boredom and is highly motivating if there is repetition to comprehend the concept as compared the repetition occurred while performing the self study to perceive the same concept [Wernbacher et al., 2012]. Schafer et al. [Schäfer et al., 2013] has displayed the caliber of SGs in order to drive the eagerness of students towards knowledge acquisition, concealed in form of tasks or activities of SGs. Use of SGs as tool to dispense knowledge has been quite effective, with a better cognitive skills as a bonus from the process [Wernbacher et al., 2012].

2.1.2 Retrospection and Present Day

The popularity of SG as an educational tool begin in the year 1971 when the first version of The Oregon Trail (TOT) SG was released. TOT was the most popular SG in that decade with an immense of user base. During that same period the game designer were just scratching the surface while TOT came on top and addressed the application of core principles of designing an SGs [Rawitsch, 2017]

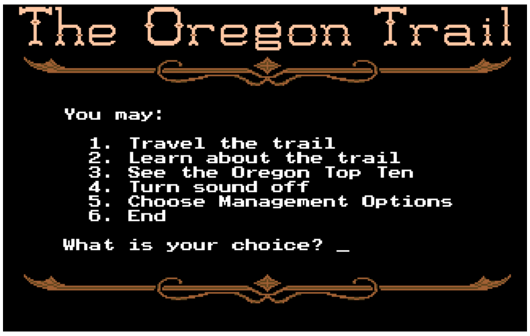

Figure 2.1: The Oregon Trail [RetroGames, 2020]

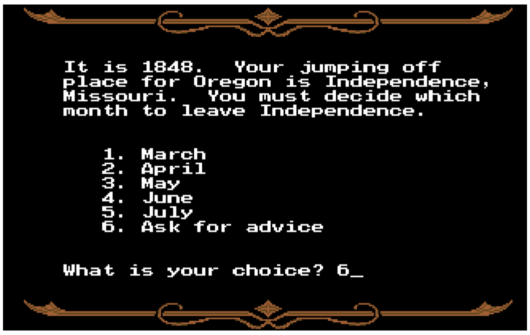

Figure 2.2: Choose travel month or hint [RetroGames, 2020]

TOT delivered the experience as if the player is playing an entertainment game during that time due to it's all over game design and story flow, as you can see in all of the above

Figure 2.3: Buy goods for journey [RetroGames, 2020]

Figure 2.4: A Scenario while player [RetroGames, 2020]

figures. From the above images of the real gaming event we can deduce that the player has to perform analysis, planning and calculation based on the clear instructions, with additional leverage of seeking hints if the player is struggling to reach a decision, as depicted in Fig. 2.2 and Fig. 2.3.

In past few decades, the SGs have evolved significantly making remarkable improvement in all sectors of the user interface, and game design comprising of game elements and game dynamics, as a result the modern SGs appear and work quite notably. Most of the modern SGs are unlike TOT, they are at a whole new level in terms of graphics and user experience with only one thing in common i.e. educational purpose.

2.2 Lost Earth Serious Game (LESG)

Lost Earth Serious Game (LESG) 2307 is developed by Fraunhofer IOSB [Atorf, Kannegieser, and Roller, 2019] under the umbrella of eLearning projects. Lost Earth is a strategy and mission based serious game enriched with pre-eminent graphics and story lines. LESG follows continuous playing format meaning that the player can continue playing the SG as long as he/she desires without an end. LESG offers the player to undertake missions of their choice which consists of tasks to be completed in order to finish the respective mission. In LESG, the player has to perform tasks and activities while playing the round or the mission, or the player can complete them also when not involved in a mission to upgrade features within the game. We will have a closer on these entities in the upcoming sub-sections.

Figure 2.5: Lost Earth Serious Game 2307 [Atorf, 2018]

Fig 2.5 is a screenshot of LESG with the game-elements such as playing options and campaigns.

Time of the Day

Another essential aspect of LESG is the time of the day or the timeline. It comprises of (1) $dawn$, (2) day (3) $dusk$ and (4) $night$. The $timeofday$ is very important from the perspective of the task to be performed and the player must always make a note of the current timeline in progress because most of the tasks in LESG have their feasibility associated with the timeline. However, there are numerous tasks that are independent of the timeline but most the task affect the player's performance rely a lot upon the timeline they are performed in.

Weather Console

The weather console screen displays the current weather conditions of the planet, and it is very essential to keep a check on the ongoing weather condition before planning or proceeding to initialize a given task. Like timeline, weather is also a contributing factor behind a successful completion of certain task that rely on the weather conditions, therefore the task requirement pertaining to the required weather condition must align with the current weather condition. If the weather condition is not feasible according to the task to be performed than the player can switch to another until the desired weather condition arise or the player can simply wait.

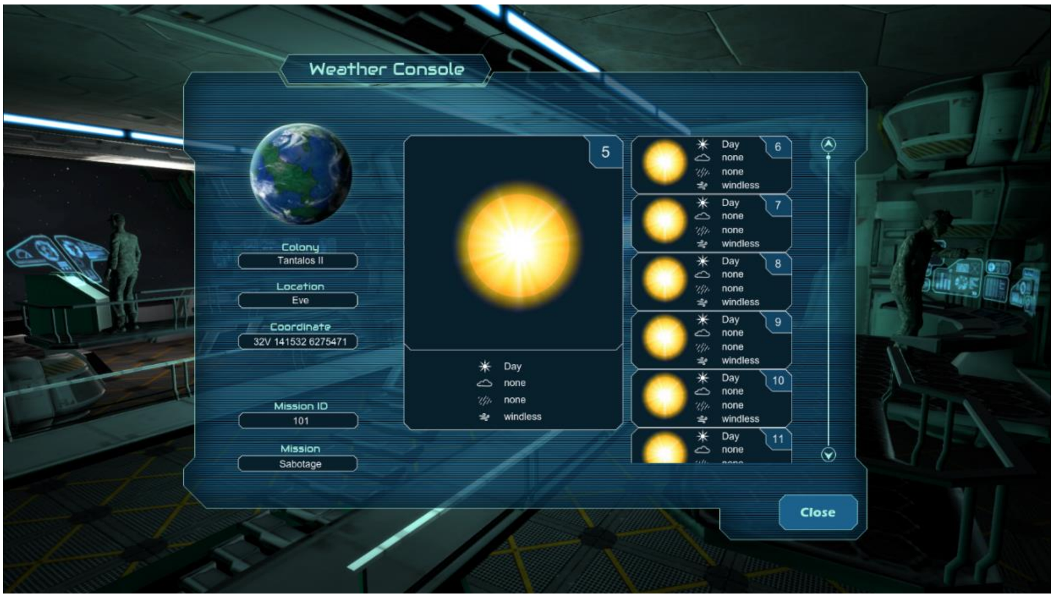

Figure 2.6: Weather Console in Lost Earth [Atorf, 2018]

Fig 2.6 is the weather console screen from LESG.

Missions

Missions and Activities are the most important part of LESG due to their significance towards education and training.

Figure 2.7: Mission Debriefing Screen from Lost Earth [Atorf, 2018]

Player can choose a particular mission in LESG, and upon doing that the mission debriefing screen Fig. 2.7 appears in front of the player. In this screen, the player will be briefed with the activities that are to be performed along with the relevant information with regard to the respective mission.

Player's score is dependent on the performance of the player while enacting the mission. If the player performs very well in the chosen mission by completing all the assignments for the respective mission, then the player is rewarded with the maximum possible score with respect to that mission. If there are assignments that with incomplete or incorrect status in the chosen mission, then the player's score is mutually affected based on the points associated with those assignment.
When the player initially begin the game there are lot of components within the games that are locked (like advanced technology, sensors) during that time. These components are unlocked as the player go on completing the missions.

Activities

Upon choosing a particular mission, the player gets to choose the activities to indulge with, and result of the activities are directly associated with the assignment of the mission. The outcome after conducting an activity decides whether the mission was a success or a failure.

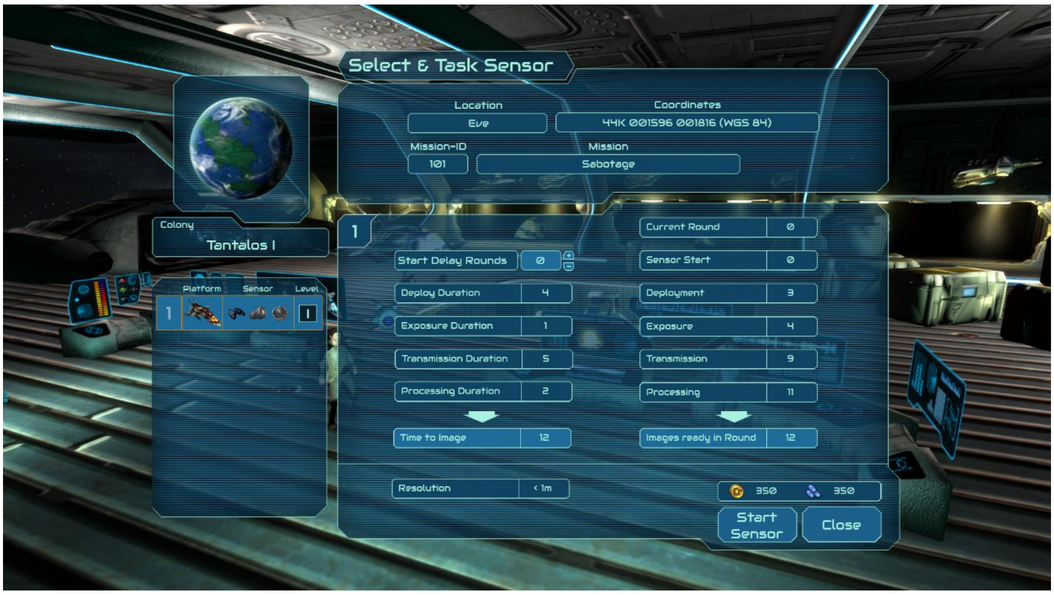

Figure 2.8: Activity Information Screen from Lost Earth [Atorf, 2018]

When the player choose an activity, they encounter the Activity Information screen Fig. 2.8. Fig. 2.8 displays the activity 'Select & Task Sensor' in action. As you can see in the Fig. 2.8 the relevant information of the activity is provided to player.

Sensors

Sensors are an essential part of the LESG that allows the player to oversee the colony's landscape and maneuver performed by the enemy. It can also portray the structures built on the colony's surface by the player or by the enemy. Thus, sensors act as eyes for the player while playing, irrespective of whether they are involved in a mission. Placing the best sensor will provide the best output, but it also cost more resources. A player can score more points from a mission if the maximum outcome is obtained by using a sensor of basic or elementary type. Most of the missions in LESG demand the usage of sensors in some way for a particular task depending on the mission, and various tasks are possibly dependent on the output provided by the sensors. If the player did not make proper use of them, it is highly likely that the player will gain fewer points.

Figure 2.9: Sensors in Lost Earth [Atorf, 2018]

Fig 2.9 displays the list of sensors available in LESG.

List of sensor available in LESG according to their precise influences based on the daytime are as follows:

1. **Electro Optical (EO) Sensor**

   - Night: An EO sensors sees absolutely nothing (automatic fail) unless it is a ground unit standing right next to it. An aircraft EO, however, will see nothing at all during night.

   - Dawn/Dusk: Like night unless you've developed the technology CCD. Then it's like day.

   - Day: EO has the default probabilities.

2. **Infrared (IR) Sensor**

   - Night, Dusk, Dawn, Day: default probabilities.

3. **Radar (SAR) Sensor**

- Night, Dusk, Dawn, Day: default probabilities unless there is a jammer (placed by the enemy).

The player must always keep in mind the cost associated with each sensor, the factors that influence the behavior of each sensor, and the output generated upon the deployment of sensors based on their type. When the player starts playing LESG, the only sensor available for the deployment is optical sensor. In order to have the superior type of sensors, the player has to acquire other types of sensor by investing resources, or the player can upgrade the existing sensor by developing technology.

Apart from portraying of the colony surface, sensors are also viable in depiction of non-ventured region in the space or to keep a watch upon the arrival of an enemy vessels. The area in space revealed by the sensors is displayed in form of hex (black or grey).

Figure 2.10: Hex Tiles in Lost Earth [Atorf, 2018]

Fig 2.10 depicts the hex-tiles (area) from the space that are uncovered by the sensors.

2.3 Experience API - xAPI

eLearning has undergone continuous evolution due to the various technological advances over many years. One of the major technological development is the specification that defines the rules for sharing content and it's structure between the eLearning devices and the remote host or the remote server. Sharable Content Object Reference Model (SCORM) was a popular specification introduced as an initiative from Advanced Distributed Learning (ADL) under the Office of the United States Secretary of Defense around 2004, used for defining the communication specification between the eLearning device and the remote entity for enabling accessibility and interoperability, as a result it became a show stopper within academic and corporate community but it was also welcomed with open arms for it's implementation for educational and training purpose [Bohl et al., 2002; Poltrack et al., 2012]. Because of SCORM, eLearning has acquired even more prominence due its potential to dispense the knowledge without any boundaries like place, time, availability and most important is a wide array of medium. For a decade SCORM was the standard specification for sharing data among different devices and platforms but after a decade it appeared that SCORM is insufficient to fulfill all the needs of the evolved technological environment, as a result, Rustiki developed $TinCanAPI$ in 2013 to fill the limitation gaps of the SCORM [Lim, 2015]; $TinCanAPI$ was later renamed to xAPI. xAPI is a specification for eLearning software including serious games that allows to track the experience of the person during learning. Experience is nothing but an event triggered as result of, the involved person performing an action. The information related to the action is called the experience data which are depicted in a specific form under the guidelines for representing of the xAPI specification.

JSON Object representing an xAPI-Statement:

```
{
  "actor": {
    "name": "john doe",
    "mbox": "mailto:john@example.com"
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/selection",
    "display": { "en-US": "selection" }
  },
  "object": {
    "id": "http://example.com/activities/apple",
    "definition": {
      "type": "http://adlnet.gov/expapi/activities/learn",
      "name": {
        "en-US": "Apple",
        "de-DE": "Apfel"
      },
      "description": {
        "en-US": "An apple is a sweet, edible fruit",
        "de-DE": "Ein Apfel ist eine süße, essbare Frucht"
      },
      "extensions": {
        "http://example.com/kinderTrainingQuestion": "123"
      }
    }
  }
}
```

Above code section represents the structure of the experience data called as xAPI-Statement. xAPI statements constitutes the experience data in $actor - verb - object$ format as per the xAPI specification. The source of event generation, which in this case is the player involved with the SG, is portrayed as an 'actor'. The manoeuvre carried out by the player is termed as the $verb$, and the target of the manoeuvre is represented as an $object$.

The structure of the xAPI statements are evolving periodically based on the emerging requirements to cover the different aspects with regard to the user experience. However, the structure from the code section representing an xAPI statement contains the core components $actor - verb - object$ of the xAPI statement, is expected to undergo extremely finite modifications. The modification expected in the structure of the xAPI statements are, introduction of more components to provide a deeper meaning to the experience, or to the event pertaining to the respective experience.

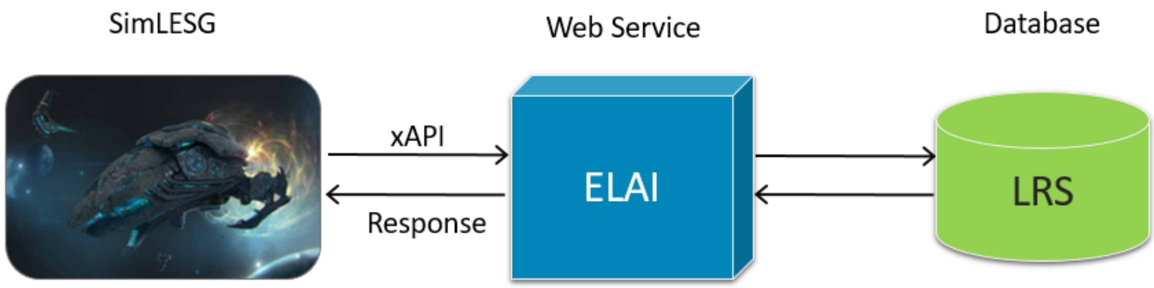

Figure 2.11: Experience API with Web Service

Fig. 2.11 represents an eLearning architecture example that follows the xAPI specification, wherein xAPI acts as a communication layer between serious game and web service (E-Learning A.I. or ELAI: We will learn more about ELAI later in implementation chapter).

2.3.1 xAPI Learning Record Store (LRS)

An eLearning system that incorporates the feature of capturing the experience data, transmits the xAPI statement over the HTTP or HTTPS whenever an activity or an event is performed by user. The eLearning system relies on the entity called as Learning Record Store or the LRS to persist all the incoming xAPI statements. Fig. 5.9 showcases the implementation of an LRS in the architecture that follows xAPI specification. In an inter-operable eLearning environment LRS is one of the most essential components because the other components of the system are to the connected to LRS for the purpose of performing: user personalization, analytics, decision making, statistical solution and machine learning [Streicher and Smeddinck, 2016], in order to provide better insight regarding the user. In our framework LRS is an essentials component, since it used not only for storing maneuvers performed by the player while playing the serious games but it is interconnected with the several component of the system to analyze the player constantly and deliver a better experience.

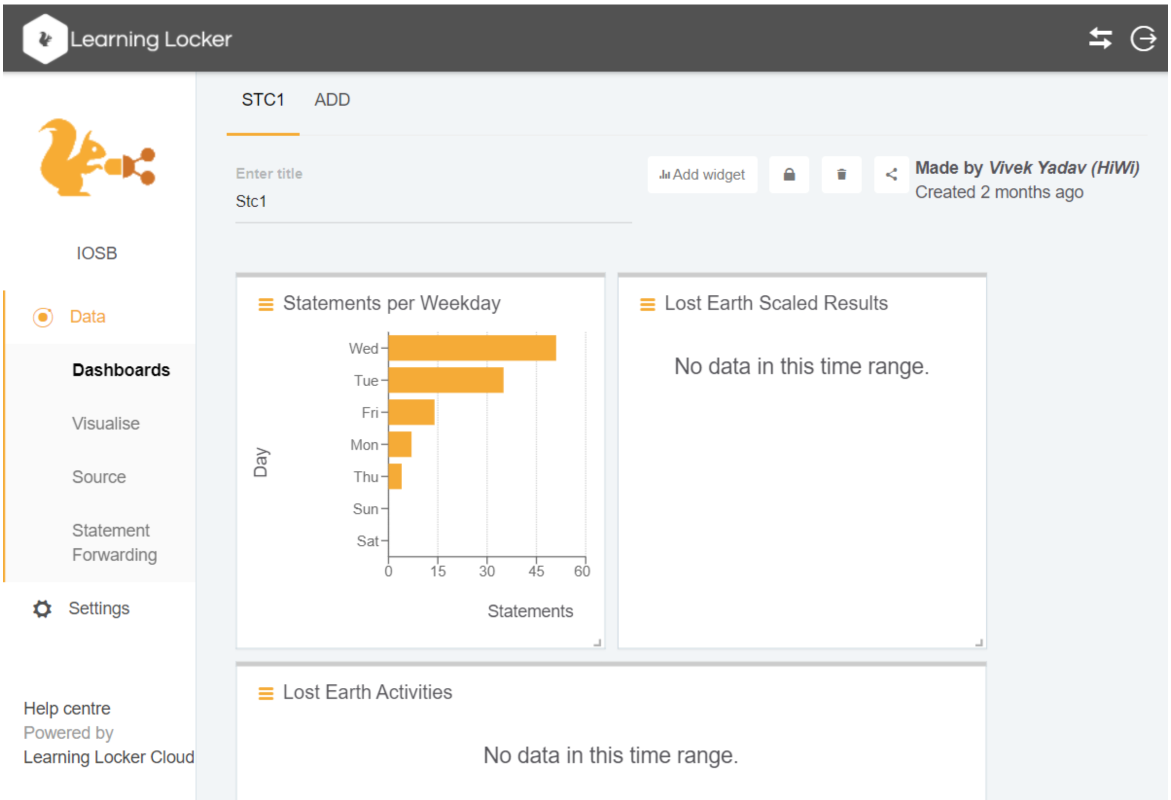

Figure 2.12: Learning Locker - an xAPI Learning Record Store

Fig 2.12 is a snippet of the 'Learning Locker' application that depicts a real player's experience data. HT2 Labs has started the Learning Locker as an open-source project in 2013 [Learning-Locker, 2020]. Learning Locker provides an implementation of LRS with standards defined by xAPI specifications. It serves as a repository to persist the learning activities in the form of xAPI statements. The source of the activities is usually an eLearning device or an inter-operable component that involves user interaction.

2.4 Hidden Markov Model

2.4.1 Markov Chain

If a system consists of a limited number of events or states that occur in a sequence at time steps then the system is assumed to be in a Markov process. At a given time the system can only be in one state i.e. there can be only one event in progress from the set of events. Once that particular event is completed the system transitions to another event, but the event that takes over could be the same event or the different event [Gagniuc, 2017]. A Markov chain describes a temporal stochastic process, where the current state is only dependent on the previous state (first-order Markov chain), and not on all the states before it [Gagniuc, 2017].

2.4.2 What are States

To explain this let's consider an example wherein the daily routine of a person consists of traveling, working, having leisure time and sleeping, and at any given point of the day that person will be in either of these states. The person can be in the sleeping state at a given point in time, and in the next point in time the person can transition into another state like traveling or remain in the same state which is sleeping in this case. This particular phenomenon representing random change in the state over time is called a stochastic process.

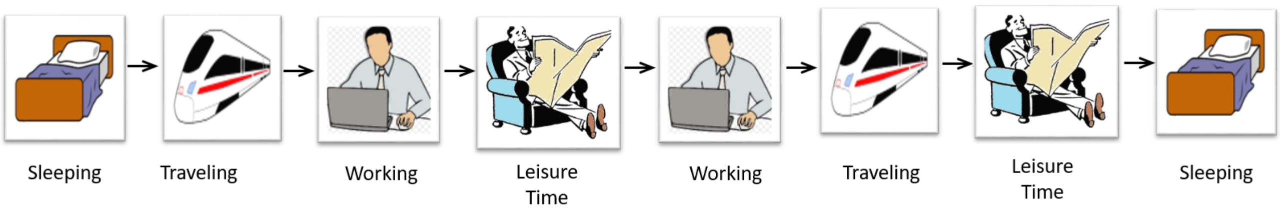

Figure 2.13: Markov Chain based on Synthetic Example

If the events occurs at time steps then the events can be assumed as a chain of events as portrayed in Figure 2.13 comprising of four states or events mentioned previously i.e. traveling, working, having leisure time and sleeping. If the current event or state is only dependent on the previous event, and not on all the event before it then the state sequence is called the Markov Chain [Gagniuc, 2017]. Upon analyzing the chain of events or states, and the state transitions, the prediction about the most probable future state can be derived. To infer the future state, each event that occur within the system are associated with a transition probability value based upon the number of the transition made from that state to the another or the same state. The state which has the highest transition probability from the current to the next state is assumed to the most likely state to occur next. Markov Chain is named after it's creator Andrey Markov.

2.4.3 Transition Probability

The transition probability is the likelihood of moving to another or the same state in the next step.The transition probability is what enables the prediction of the most probable future state to occur. To get the clearer picture, let's consider the previous example of the Markov Chain with states: (1) Sleeping, (2) Traveling, (3) Working, and (4) Leisure Time, with the transition diagram and table mentioned below.

[0]Animated image courtesy of [ClipArtsFree, 2020]

Table 2.1: Transition Probability of Fig. 2.14

	Sleeping	Traveling	Working	Leisure Time
Sleeping	0.1	0.6	0	0.3
Traveling	0.2	0.1	0.3	0.4
Working	0	0.4	0.3	0.3
Leisure Time	0.6	0	0.2	0.2

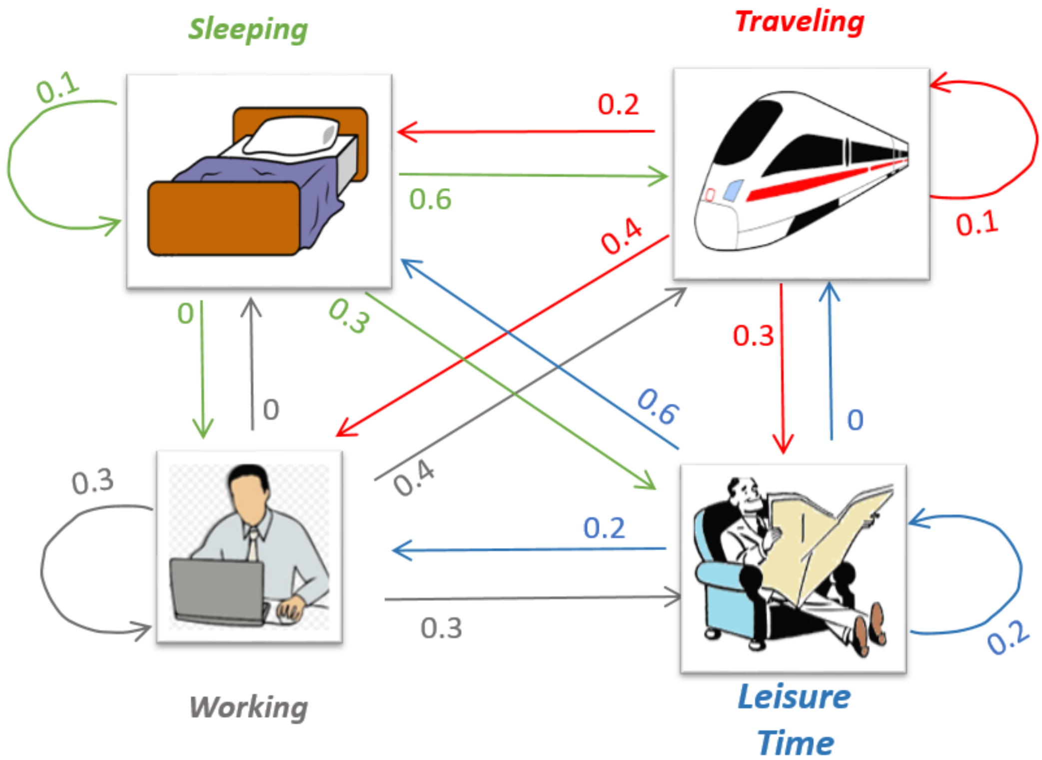

Figure 2.14: Transition Diagram (Synthetic) of Markov Chain

Table 2.1 and Fig. 2.14 represents the transition of states of the synthetic system assumed to be a Markov Chain. The values from the Table 2.1 and weights of edges from the Fig. 2.14 are the transition probability, and by using this particular information, the event or the state with the highest probability from the current position can be assumed as the most likely future event or state to follow after completion of the current event. Let's assume that the person is currently sleeping right now, and based on the information provided by 2.1 and Fig. 2.14, the best guess of the upcoming event is the traveling.

2.4.4 About Hidden Markov Model

The HMM consists of a Markov chain, but it cannot be observed directly, so the states in the chain are called hidden states, and the observables are used to identify the underlying states (hidden) [Rabiner and Juang, 1986]. The states or hidden states are the latent variables

[0] Animated image courtesy of [ClipArtsFree, 2020]

associated with the observables, which means that the purpose of the observables, in a system assumed to be HMM is, it should provide the information of the states of the system. The observables can be used to deduce the states only if there is some connection between them.

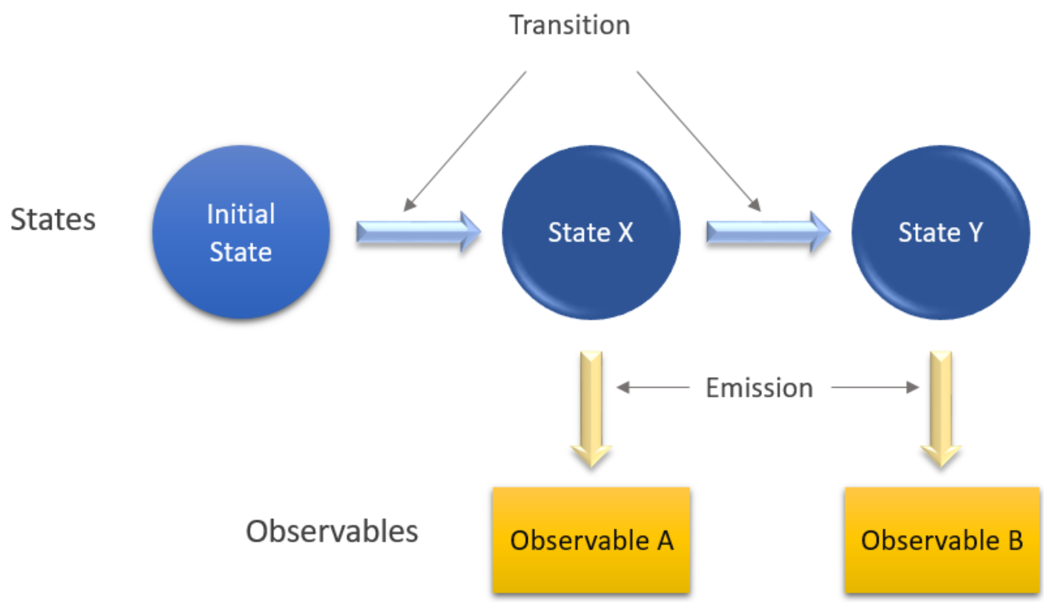

Figure 2.15: States (Hidden States) emit Observables (Visible States) in HMM

Fig 2.15 represents the stochastic states and observables. The HMM is stochastic in nature, which means at each time step, there is always a state in progress within the system assumed to be an HMM.

HMM belongs to the family of Bayesian networks where the States have the property of transitioning from one state to another, and they also emit observables unidirectionally from them [Ghahramani, 2001]. In the upcoming sub-sections, we will gain more insight into the constituents and working of HMM.

2.4.5 What are States and Observables

To understand more clearly lets consider that there are 2 friends Jack and Jill, who have different daily routines. On any working day, Jack can be in either of the following states at a given time of day: (1) traveling, (2) working, and (3) leisure time. Since Jill cannot see Jack directly, it would be difficult for Jill to tell precisely which state Jack might be in. Since they are very good friends, they are connected with each other through online web sites and applications. The only thing visible (observable) to Jill from her position is the type of web-site or application on which Jack is currently active on related to (1) music, (2) productivity, (3) social network. Jill could extract information about Jack concerning the website or application (observable) Jack is using. She also has a few instances related to the state in which Jack is in when he is visiting a website or the application. Using the respective information regarding the states and the observables, Jill can derive a connection between the state and the observation. Thus Jill can estimate the state in which Jack would most likely fall at a given point based on the website or the application Jack uses.

Table 2.2: States (Hidden States) and Observables (Visible State)

States ->	Traveling	Working	Leisure Time
Obervables ->	Music	Productivity	Social Network

2.4.6 Probabilities in HMM

The HMM consists of state transitions and emissions. We have already discussed the transition or transition of state in sub-section 2.4.3, however, the emission has not been discussed so far, and we are going to gain more insight regarding it in this sub-section. It works the same in case of HMM as well with only one difference; the states are not directly visible like Markov Chain. For discovering the states of HMM, the observables come into play with the ’emission’ concept. The states or the events, of the HMM that are hidden, emit a specific behavior or another visible state, making a connection between the observables and the underlying states (hidden) due to dependency of the observables on the underlying states. This particular behavior of the visible states, associated with the system’s underlying states, is called emissions, and the probability of their occurrence when the underlying state is in progress is called emission probability.

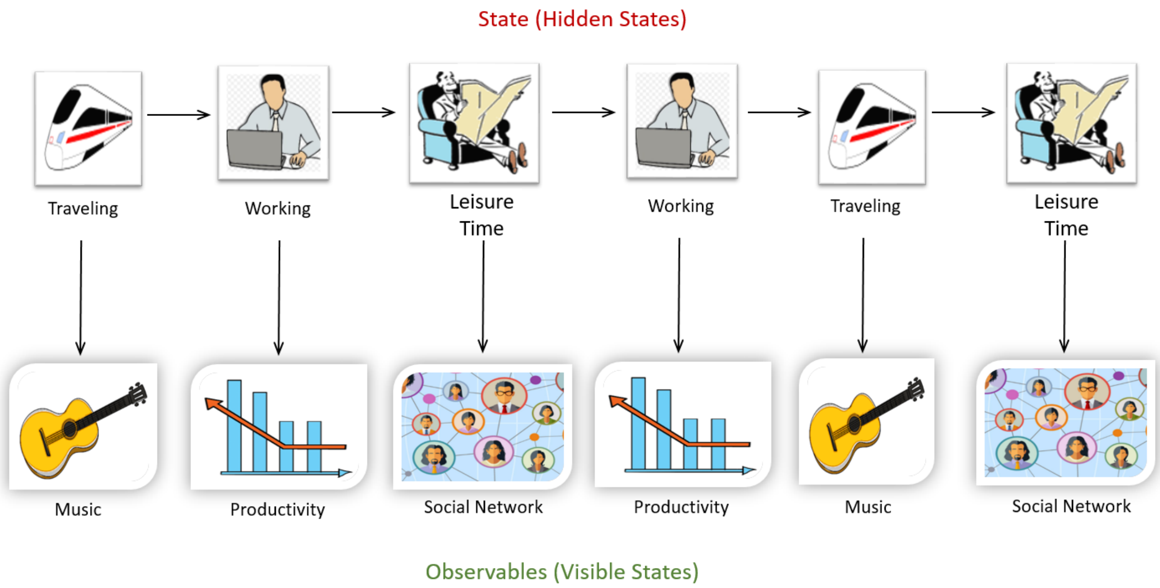

Figure 2.16: Example of Emission Probability in Hidden Markov Model

Consider the example presented in sub-section 2.4.5 with a few states and a few observables defined in Table 2.2. Let’s assume that Jill can gather information on Jack’s whereabouts containing a few instances of observation (observable or visible state) and the

[0]Animated image courtesy of [ClipArtsFree, 2020]

underlying states (hidden). Using that information, she can form a connection between the observables and the state as depicted in Fig. 2.16.

Jill could form a connection between these two with the help of time overlap: at what time the observations and the states were similar, e.g., during travel, work, and leisure time which apps were used by Jack. Based on the data collected, she can derive the emission probability, as shown in Table 2.3, of the states with the observations. She would be able to predict that in which state Jack falls when a particular observable presents itself [Rabiner and Juang, 1986]. The emission probability is derived in Table 2.3 based on the synthetic example.

Table 2.3: Emission Probability (Synthetic) of Fig. 2.16

	Music	Productivity	Social Network
Traveling	0.6	0.1	0.3
Working	0.2	0.7	0.1
Leisure Time	0.3	0.0	0.7

If we have a sequence of observables, then using the emission probability, like the one in Table 2.3, it is possible to derive the sequence of states. The derived sequence of states represents the transition of states [Rabiner and Juang, 1986]. Using state's transition as shown in Fig 2.16, it is become fairly easy to create the transition probability of the states from the given system; it is explained in 2.4.3.

Table 2.4: Transition Probability (Synthetic) of Fig. 2.16

	Traveling	Working	Leisure Time
Traveling	0.1	0.4	0.5
Working	0.4	0.1	0.5
Leisure Time	0.2	0.5	0.3

The transition of states is similar to transition probability discussed in sub-section 2.4.3, but in this case, the observables are used to derive the current state using emission probability and guess the future state using transition probabilities, like one in Table 2.4.

Based on the current observation, prediction of future state can be made using emission probability and transition probabilities alone; let's consider an example. Suppose that Jack is on a social networking website, then according to the emission probability values from Table 2.3, we can derive that he is in the relax mode ($leisure - activity$) right now. Since the current state is finalized i.e., $leisure - activity$, we could look up to Table 2.4 for the transition probabilities and check which state has the maximum likelihood of occurring after state $leisure - activity$. Well, the $working$ state has a probability of 0.5, which highest among all the states transitioning from $leisure - activity$ state. Hence, we can state that the most likely upcoming future state would be $working$.

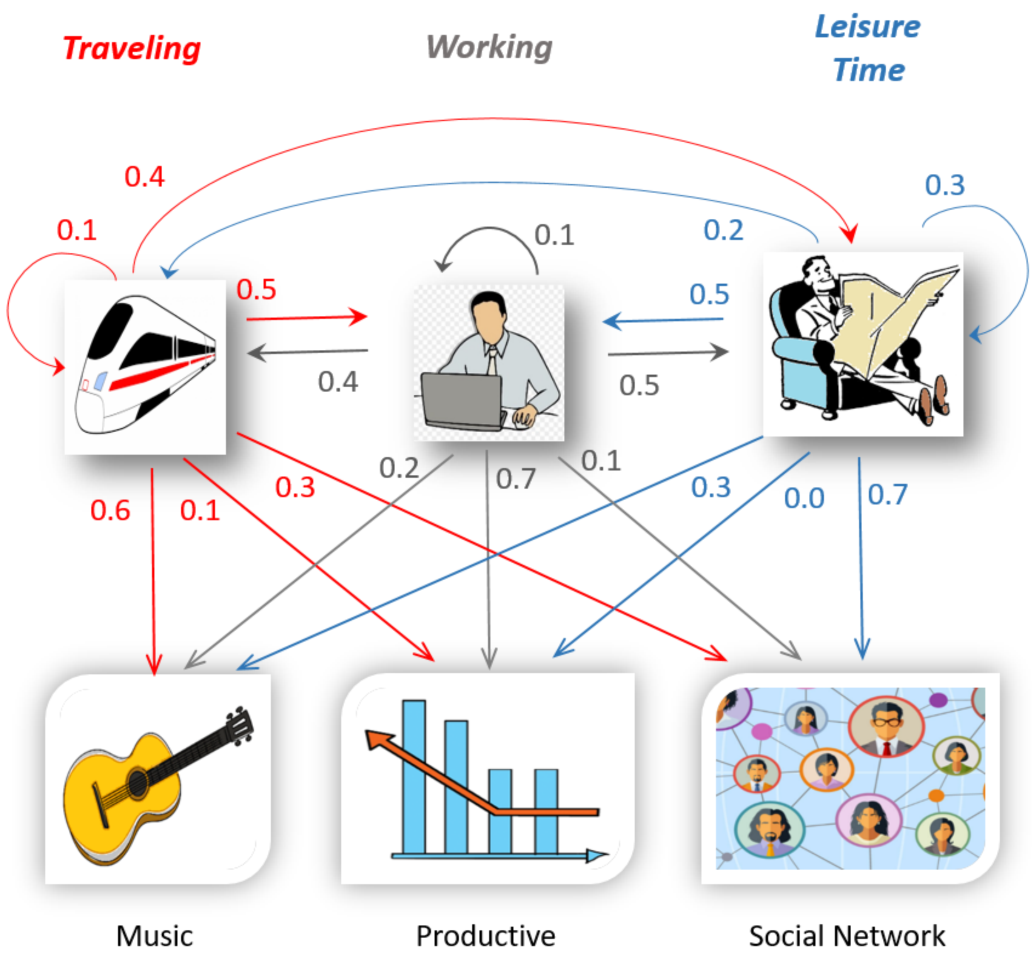

Figure 2.17: Transition Diagram (Synthetic) of Hidden Markov Model

Fig 2.17 is the diagrammatic representation to portray a holistic view of HMM with the emission probabilities and transition probabilities from Table 2.3 and Table 2.4.

2.5 Adaptivity

As mentioned in the introduction chapter, serious game (SG) has many potentials to deliver the education and training, but if we ask ourselves, will an SG be able to achieve the same outcome for each individual involved in the learning process? The answer to this question is no. What hinders an SG from providing a successful outcome with all the individuals involved. To understand this, let's take a scorecard containing the result of all the students that belong to a particular course. Even if we remove the differentiating factors like age group and location, we would still find variations in the scorecard. The reason behind this variation is that people are different from each other; this also includes their way of perceiving information. If SG can adapt based on how individuals perceive information,

[0]Animated image courtesy of [ClipArtsFree, 2020]

then the outcome ratio can most likely be higher than the SG that does not have an adaptive mechanism.

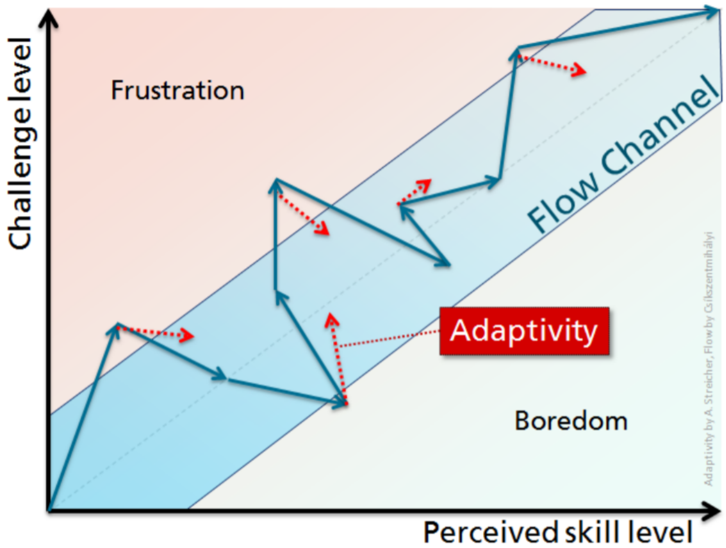

Figure 2.18: Adaptivity Flow Channel [Streicher and Smeddinck, 2016]

Fig 2.18 portrays the Adaptivity with learning graph, based on factors such as frustration and boredom. The goal is to keep the learning graph within the flow channel region so that the interest of the individual is not affected. The primarily responsible factors behind the genesis of frustration and boredom are pace, content, and difficulty level. Let's assume that an SG is involved in the learning process, and if the SG can possess power over the responsible factors like in a controlled system, then it would be possible to keep the learning curve within the flow channel.

This thesis features the development and usage of a controlled system that uses the Hidden Markov Model to manifest the adaptivity in the form of assistance while an individual plays an SG. The framework in the discussion here is called SeGaAdapt, which comprises an application server and a micro-service application that embeds the Hidden Markov Model. The micro-service application continuously monitors the player's activities and implements adaptation strategies based on the player's maneuver.

Chapter 3

Related Work

This chapter will discuss the published content and state of the art related to the components of the proposed solution (framework).

3.1 Experience API (xAPI)

The simulator for the serious game developed for this thesis to be integrated with the proposed framework follows xAPI specification; therefore, it is essential to know the contribution bestowed for the implementation and techniques concerning xAPI and why it is the best choice for developing the interoperable architecture for the serious games.

Sharable Content Object Reference Model (SCORM) was very instrumental in creating an inter-operable ecosystem for eLearning, but it proved to be a deficit in a few cases; therefore, xAPI was introduced to fill its shoes and cover the deficiencies of SCORM. xAPI was an Advanced Distributed Learning (ADL) Initiative program [ODUSD, 1999] and Rustici Software introduced xAPI in 2013 [Software, 2020], it was vehemently accepted everywhere concerned with eLearning, and it still is the foremost choice for creating an inter-operable ecosystem for eLearning.

Upon creation of xAPI, it has become one of the useful tools to perform communication [Streicher and Smeddinck, 2016] concerned with serious games (SGs) to address goals such as adaptivity [Streicher, Bach, and Roller, 2019] and analytics [Serrano-Laguna et al., 2017]. [Streicher and Roller, 2015] have showcased the role of xAPI to communicate SG with the server to perform 'adaptive inter-operable tutoring agent'. Report formulated for xAPI [Kevan and Ryan, 2016] specifies xAPI as a critical ingredient in generating a positive learning environment. EU H2020 RAGE and BEACONING are serious game projects [eUCM, 2020] with extensive research and implementation performed towards usage and effectiveness of SGs; one of the projects exhibits the research and analytics towards a location-based serious game with xAPI [Perez-Colado et al., 2018] to record the player movement and location-based interactions.

3.2 Machine Learning

The usage of machine learning is prominent in almost all IT sectors. Serious games (SGs) are no exception from using machine learning for user assistance, prediction, user model, and many others. [Streicher and Smeddinck, 2016] mentioned the role of machine learning to create dynamic, adaptive, and personalized games. It is essential to understand which machine learning algorithms would effectively carry out which particular task. For instance [Bellotti et al., 2009] used genetic computation and reinforcement learning on the player data to perform task modeling; [Slimani et al., 2018] used k-means and expectation-maximization algorithm for educational data mining.

3.2.1 Markov Model

Even before when SGs were not so prevalent for educational purposes, Markov Models were used for various objectives for other education and training later tasks. Researchers have been dealing with the daunting task of solving problems like pattern recognition and analytics using students' and players' data from eLearning devices and other sources. Theoretical technique for performing knowledge assessment of student is presented by [Falmagne and Doignon, 1988] using Markovian property of Markov Models. Markov Chain is used by [Liu, Lin, and Chen, 2010] upon the student's algebra tests to derive transition probability of the student learning state and form a connection with the teaching so that improvement can be reflected towards enhancement of teaching methods.

Objective of "A generic and efficient emotion-driven approach toward personalized assessment and adaptation in serious games" [Mostefai, Balla, and Trigano, 2019] is to perform the Assessment of the user's personality type and playing style for the adaptation. To solve this problem the Personalized Adaptation method is used driven by emotions based on Markov modeling with dependency between serious game events and change in player's emotional state using the adaptation technique.

3.2.2 Hidden Markov Model (HMM)

Hidden Markov Model (HMM) is adopted as a useful technique for understanding the player's performance and behavior, implementing motivational strategies, classification, and analytics integrated with serious games (SGs).

European project PlayMancer deals with affect recognition in SG for supporting patient with behavioral and mental disorder treatments and chronic pain rehabilitation, and the approach used for training the probabilistic models by [Moussa and Magnenat-Thalmann, 2009] utilize Hidden Markov Models, Bayesian networks, and Dynamic Bayesian networks for affect recognition and later provide the basis for studying fusion models. Similarly, the research report [Fu, Zapata, and Mavronikolas, 2014] highlights many statistical methods used in SGs for assessment, it mentions the use of HMM [Jeong et al., 2010] utilized for the classification of students into high and low performing students based on the student's phase selection in a learning cycle while playing an SG.

[Derbali, Ghali, and Frasson, 2013] describe the approach for investigating new ways to improve learner's performance by using physiological sensors. Motivational strategies are thoroughly studied and then implemented to promote learners' performance and motivation using HeapMotiv serious game. They identified physiological patterns associated with motivational strategies by building HMMs, which uses Keller's ARCS model concerning motivation and electrophysiological data based on human reflex actions.

Use of HMM to model individual differences is elaborated by [Bunian et al., 2017]. The modeling approach is developed using data collected from players playing a Role-Playing Game (RPG) and thus proposing an approach that presents a HMM of player in-game behaviors to model individual differences, and using the output of the HMM to generate behavioral features used to classify real world players' characteristics, including game expertise and the big five personality traits. The research emphasizes on the differentiation of the players based on the individual behaviors, characteristics and personality traits using Hidden Markov Model.

The implementation of HMM in the proposed framework for adaptivity manifestation in SGs is widely open but is some cases, HMM is used for adaptivity manifestation is other kinds of games. [Caserman et al., 2018] has used HMM within Exergames to assist patients by identifying their body movements, and the same technique can be implemented in some way within SGs as well to help the players.

3.3 Adaptivity and Personalization

Serious game (SG) customization techniques pertaining to adaptivity and personalization can be quite useful to maintain the player's interest and motivate the player to play further. Customization can be achieved through the existing player data, which is quite popular and widely used approach. Another approach can be the customization through player data from an external source such as social media to adjust the SG according to the player's profile [Konert, Göbel, and Steinmetz, 2014].

[Bellotti et al., 2009] have well illustrated the mission's modeling based on player data such as user profile to maximize the learning objective through task customization to maintain the game flow; dynamic adaptive techniques described in the paper are the most eye-catching concepts concerning this thesis. Importance of adaptivity and personalization is well illustrated by [Streicher and Smeddinck, 2016] and [Göbel and Wendel, 2016]. [Streicher and Smeddinck, 2016] showcased very well the motivation behind the manifestation of adaptivity in SGs; they also expiated several methods and techniques to implement adaptive and personalized for fun and informative eco-system for SGs that derives the successful outcome from the gaming process. [Hussaan, Sehaba, and Mille, 2011] have focused on the system development that considers the player information and data of the domain, pedagogy, and SG to generate adaptive scenarios.

Development of learning adaption for serious games (SGs) is one of the key components of the SGAPID framework presented by [Nguyen, Gardner, and Sheridan, 2018].

3.4 User Assistance and Game Analytics

Equipping serious games (SGs) with a guidance and assistance system that assists the learners and help instructors to improve the learning process and teaching methods is presented by [Elaachak, Belahbibe, and Bouhorma, 2015]; which introduced new system for assistance, guidance and learning analytics based on a multi agent system that will work in tandem with a serious game emphasizing on the assistance system and learning analytics.

Analytics in e-learning has become a hot topic to provide a better understanding of motivation, strategies, and component that effect during the learning process. The literature review related to data analytics from the serious game presented by [Alonso-Fernández et al., 2019], show applications of data science techniques on the game data. [Nguyen, Gardner, and Sheridan, 2018] proposed the integration of learning analytics in SGs synchronized with realtime data for intellectually challenged audiences. [Kostoulas et al., 2012] proved the effects of speech recognition in a user performance through analytics by collecting the dataset for evaluation and performed speech recognition experiments to demonstrate a connection between performance and models of emotional states. [Slimani et al., 2018] used machine learning algorithms for learning analytics through SGs to present the player experience data analysis.

Chapter 4

Methodology

In this chapter we emphasis on the model, techniques, and phases that are used in this thesis to develop the proposed solution to address the research question. We take a closer look at how a concept in the form of method, logic, or feature works. The prime lies in explaining the role and behavior of techniques and components integrated with the proposed system. Essential topics such as assistance, game simulator, model training, and adaptivity manifestation are expatiated to familiarize the proposed framework's internal process. Since the game simulator plays a vital role in proof of concept, therefore, much emphasis is devoted towards its customization and entities that mimic entities of Lost Earth Serious Game.

4.1 Adaptivity Using Assistance

Adaptivity is the quality or capability of a system to adapt according to the given condition, and it can be achieved in different ways.

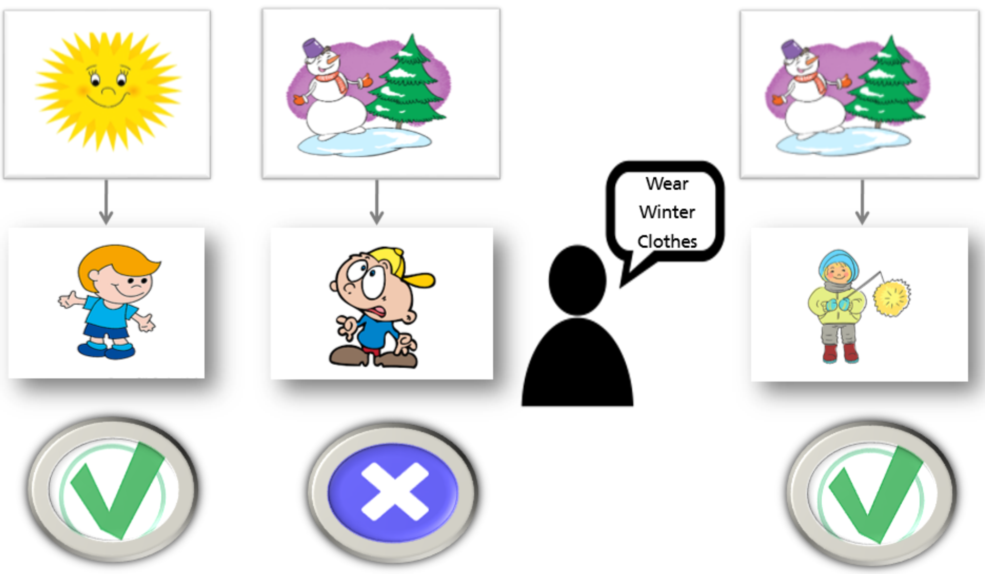

Figure 4.1: Adaptivity Using Assistance

[0]Animated image courtesy of [ClipArtsFree, 2020]

Since we are so concerned with the positive outcome for the gaming process without providing any room for the negative aspects such as frustration and boring and positive aspects such as entertaining, we need a system that can be adaptive when required. It is essential to assist whenever the player cannot overcome a particular task and would be able to complete the task only if the player receives help from an external source. The external sources can be an individual such as friends, parent, teacher, or it can be an informative electronic medium. In contrast, if a player can complete tasks flawlessly or with a bit of ups and downs, then it signifies that the respective player comprehends the situation within the game effectively and channel that understanding towards completing the given task. For such an individual, the assistance feature would be fruitless. It would become imperative for the system to keep the game as competitive as possible to cope with the player's level to maintain the drive to play further so that the player does not feel bored. Fig 2.18 from Chapter-Introduction is a graphical representation of both of the challenges mentioned earlier and attributes necessary to maintain the flow. The motivation of this work lies towards providing assistance to the player who is facing difficulty and not towards the latter scenario of maintaining the challenge to evade the boredom effect.

4.2 Assistance Implementation

There are 2 ways to implement the dynamic user assistance through HMM, i.e. within the game, or at server end that interacts with the game.

(i) In built implementation: The assistance mechanism can be integrated within the game itself. The metadata within the game can contain the observations along with observable and hidden states required for prediction purpose. The game application can store the data in the heap whereas the web based game implementation uses cookies to store the data.

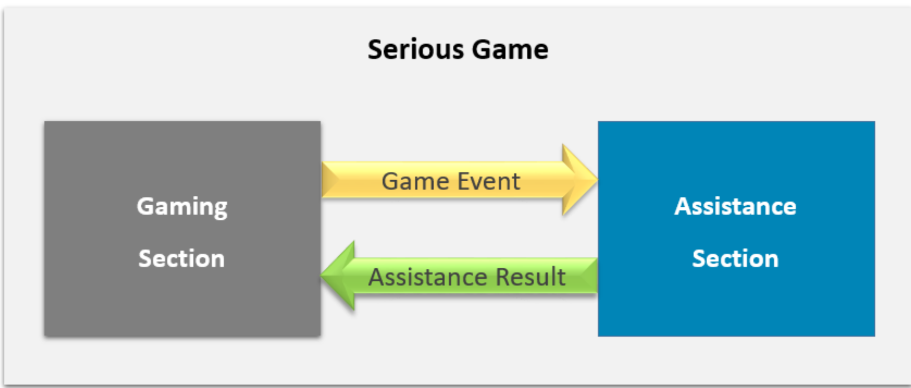

Figure 4.2: Assistance mechanism implementation within serious game

(ii) Remote implementation: Implementation of the functional component on the remote server that interacts with the serious games (SGs) is a widely used strategy. The remote server implementation is adopted by [Andrade et al., 2013] to connect SG with a control system, whereas [Hassan et al., 2012] has used multiple servers that interact with each other, and one of them hosts a cloud-based SG while the remaining servers are configured to execute specific tasks.

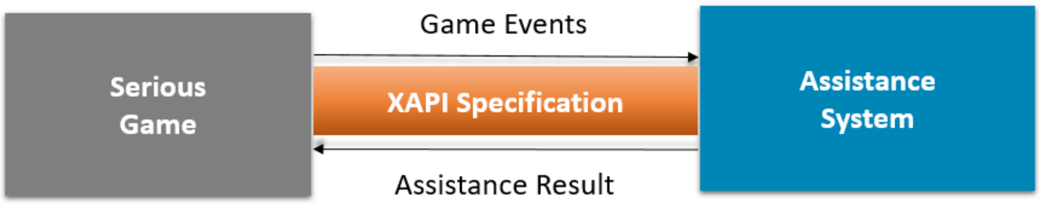

Figure 4.3: Assistance mechanism implementation at remote system

Fig 4.3 represents the proposed assistance system at a remote location, acting as an application server. The application server monitors the user experience and continuously serves the incoming requests from the SG. One of the server endpoints acts as the xAPI statement receiver that accepts the player input from the game and responds to the game.

4.3 Simulator for Lost Earth Serious Game

Lost Earth is a strategy and mission based serious game (SG) developed by Fraunhofer IOSB enriched with pre-eminent graphics using ’Unity game engine’. As a project, Lost Earth Serious Game (LESG) is a sophisticated project that is quite significant in size. To reflect the adaptivity manifestation within LESG coined in this thesis is an enormous challenge to accomplish in 6 months. The solution approach towards implementation of the concept introduced in this could be to develop an alternative game that simulates LESG or parts of LESG. Therefore, an idea to develop a project called Simulator for Lost Earth Serious Game (SimLESG) is a benevolent approach for providing adaptivity manifestation. The game simulator within this project mimics a few sections of the game-play scenario from LESG in an elementary form in the form of a command line and/or web interface. The benefit of using this simulator is to smoothly address the challenge of adaptivity manifestation through the integration of simulator with the system that implements user assistance by using the Hidden Markov Model. SimLESG mimics the strategy of employing the missions and activities to provide the gaming scenario like LESG.

4.3.1 Customization and Setup of Simulation

SimLESG is developed as a simulator project for LESG in Java programming language with a straightforward and generic configuration to set up the game environment, and then proceed to play. It contains a programming console and a web interface for the player to interact with or play the simulator game; it is up to the player to choose and proceed with the gameplay. SimLESG provides a simple mechanism to customize the interface of simulation through the JSON mapping file. Due to the simple customization nature of SimLESG, it can take the form of the LESG scenarios and any other strategy based SG effortlessly, that employs missions and activities. The instances of missions and activities are mapped from the content of the JSON Mapping file object-mapping.json.

Configuration code containing details of activities:

```
"activities" : {
  "Deploy Sensor" : {
    "name": "Deploy Sensor",
    "comment" : "Activity to deploy sensor",
    "value": ""
  },
  "Image Interpretation" : {
    "name": "Image Interpretation",
    "comment" : "Activity to initiate the image interpretation",
    "value": ""
  },
  "Check Weather" : {
    "name": "Check Weather",
    "comment" : "Activity to check the weather condition",
    "value": ""
  },
  "Check Image" : {
    "name": "Check Image",
    "comment" : "Activity to check the image received from interpretation",
    "value": ""
  },
  "Allocate Recruits" : {
    "name": "Allocate Recruits",
    "comment" : "Activity to the recruits for the mission",
    "value": ""
  },
  "Allocate Workers" : {
    "name": "Allocate Workers",
    "comment" : "Activity to the workers for the mission",
    "value": ""
```

```
    }
  }
```

Configuration code containing details of missions:

```
  "missions" : [
    {
      "name": "Exploration",
      "activities" : [
        "Deploy Sensor",
        "Image Interpretation",
        "Check Image",
        "Allocate Recruits"
      ]
    },
    {
      "name": "Build Colony",
      "activities" : [
        "Deploy Sensor",
        "Image Interpretation",
        "Check Image",
        "Check Weather",
        "Allocate Recruits",
        "Allocate Workers"
      ]
    }
  ]
```

There is a set of instructions to process the input configuration $JSON$ file and provide objects that represent $Missions$ and $Activities$.

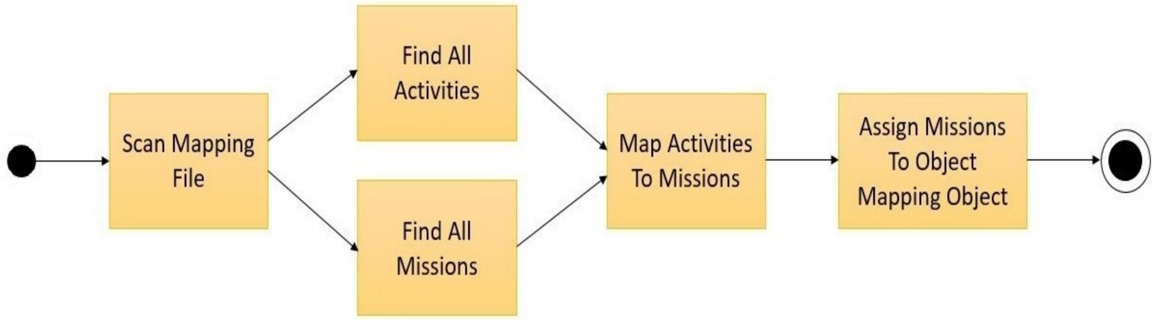

Figure 4.4: Object Mapping Process

Fig 4.4 depicts the UML of the steps performed by the object mapping function within SimLESG for mapping missions and activities. The Java object mapper scans the

JSON simulation configuration file to find all the missions and activities available. The configuration file contains details about the activities: name, comment, and default value; It also contains details about the missions: name and the mapped activities. By utilizing the details of the activities from the mapping file, instances of activities are created by SimLESG, and similarly, using the details of the missions, instances of missions are created. The code template given below contains content from the JSON mapping file.

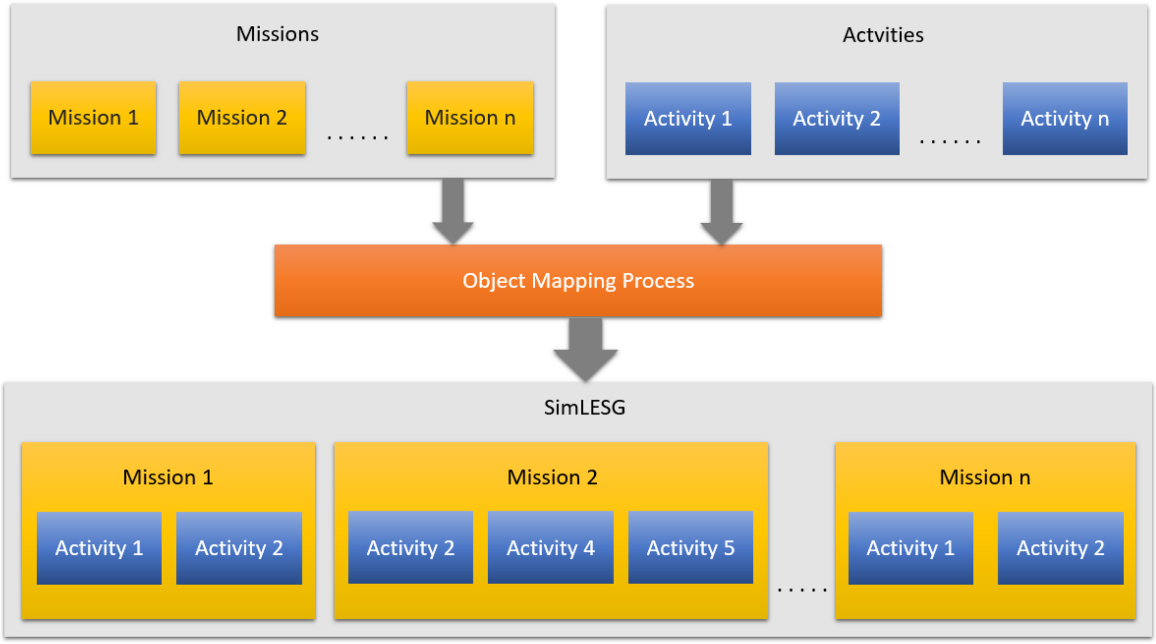

Figure 4.5: Mapped Mission and Activities

4.3.2 Missions and Activities in SimLESG

Like LESG, SimLESG contains missions which user can opt to play. The missions are associated with specified goals, and to get the highest score, the player must complete all the goals from that respective mission successfully. If a goal is incomplete or performed incorrectly, then the player will score comparatively less in that respective mission. After choosing a mission, the game displays the list of activities specific to the selected mission. The configuration for mapping the activities with the missions is defined in JSON configuration file; respective mapping configuration is outlined in the code verbatim part 4.3.1 of subsection 4.3.1. These activities will provide a means to achieve all the goals of the chosen mission. Players can perform only one activity at a time from the list of available activities. To perform these activities, the player has to select the activity and perform an action based on the chosen activity.

There are two types of activities based on the input type: (1) System-defined input and (2) User-defined input. The activities that fall under system-defined input category, are basically meant for transmitting information to the user; for instance, if the player selects an activity like $CheckWeather$, the game will return the information about the current weather condition, and the player need not have to provide input. Most of the activities fall under the user-defined input category, and to perform an activity of this type, the player is expected to provide input. For instance, if the player chooses the activity like $DeploySensor$, the respective activity expects the player to provide an input like which sensor to deploy.

4.3.3 Time Of Day

Time of day is an essential aspect of the Lost Earth; therefore, it is equally important to incorporate it with SimLESG. There exist four kinds of time of day, and they are dawn, day, dusk, and night. The dawn is the morning period; day follows when the dawn ends; dusk comes after day during the evening period, but there is the presence of natural light; dusk is followed by night when there is no natural light present. We will encounter time of day in the upcoming chapters. Time of day is a crucial factor in completing several tasks successfully; therefore, the player needs to be aware of the current time of day in process and which actions are suitable to which time of day while performing an activity.

4.4 Model Development Process

Model development is the most critical process of machine learning because it forms the basis in the decision-making process. The models are trained using various methods applied to the historical data. In many cases, the models are updated dynamically based when personalization is concerned. For developing the prototype framework, this thesis is restricted to the training HMM using historical data. Still, future work involves the dynamic modeling of HMM to provide a personalized experience.

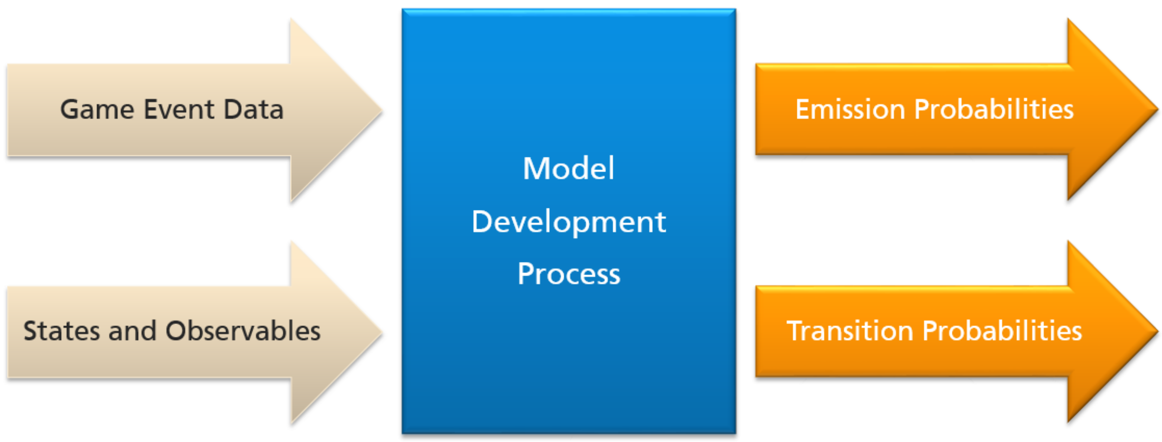

Figure 4.6: Model Development Process

Fig 5.12 is a schematic representation of the model development process conducted on the game event data with the output as the probabilities. We train the HMM by using the game event data (from multiple players) of serious game (SG) comprising of activities performed while playing. Game event data can be previously recorded xAPI statements or activity records in a database. To construct the most feasible strategy, the activities with the highest score or point gain from the player data is used for training HMM.

The technique applied by [Gong et al., 2017] on the data in time series is most suitable for developing the models in the case of player data in the form of xAPI statements. There are two way to train the HMM:

1. Using pre-existing data

2. Dynamically using live event data

This thesis is resorted to training the HMM using the first way i.e., training using the player's data. Several methods are executed on the 'game event data' to discover the states and observables with respect to the components like mission, activity or within the game. The values pertaining to the game event are applied to Forward Tracking or Baum-Welch Algorithm to discover the probabilities like initial values, transition probabilities and emission probabilities. A trained HMM comprises (a) transition probabilities of the states and (b) emission probabilities of the observables from the states. Transition probabilities and emission probabilities are previously explained in the section 2.4. In our case, the HMM acts alike, as depicted in the section 2.4.

After training the HMM, the framework can start receiving the xAPI statements in the 'actor-verb-object' format which contains information related to the activities that take place within the SG.

4.5 Adaptivity Manifestation using HMM

There are two ways to discover the need for adaptivity. The first way involves the request generated by the player when the player is facing difficulty, which is relatively easy, and the system does not have to play any role in the detection of adaptivity. The second way requires monitoring the events generated by the player through the xAPI tracker and scanning for the aspects such as if the activities are performed incorrectly, or check if the performance parameters (or scores) are lower than the expected performance. Given the second case, if the system detects the need for adaptivity, it triggers the signal to initiate the adaptive mechanism. The system responds to the serious game (SG) with the adaptive strategy to help the player, so that the desired outcome is achieved from the gameplay process.

For developing the appropriate strategy, the system utilizes the models of the Hidden Markov Model (HMM). Since HMM is a probabilistic model[Xuan, Zhang, and Chai, 2001], a trained HMM can predict the feasible adaption strategy with maximum probability. There are three ways to manifest adaptivity to provide a more feasible exercise with the higher probability towards the positive outcome:

1. Provide hints to player while playing
2. Customization of game elements
3. Tune-in the parameters of SG

4.5.1 Provide Hints

Hints can be a useful element to help someone come up with the given answer to a question or the solution to a problem in real life. There are two policies when it comes to presenting hints; (1) one must comprehend that "in a given situation what needs to be done" (2) do not provide an answer directly but rather offer something which leads to the solution. These policies apply in real life to someone who has acquired information or experience related to the given problem. The case is the same when there is a computer involved that dispenses hints. The computer system relies on historical data related to the given problem to satisfy both the policies to develop and provide hints [Price, Dong, and Barnes, 2016].

HMM's emission probability is the perfect candidate to fulfill the first policy, i.e., deduction of the appropriate solution. Hints can be developed by excavating the data that leads to the discovered solution. [Price, Dong, and Barnes, 2016].

4.5.2 Customize Game Elements

Another way to assist a player is by customizing the game elements so that it becomes easier for the player to perform activities successfully. The upcoming task or exercise can be modified by altering the conditions/attributes to provide a more feasible task or exercise with a higher probability of achieving a positive outcome.

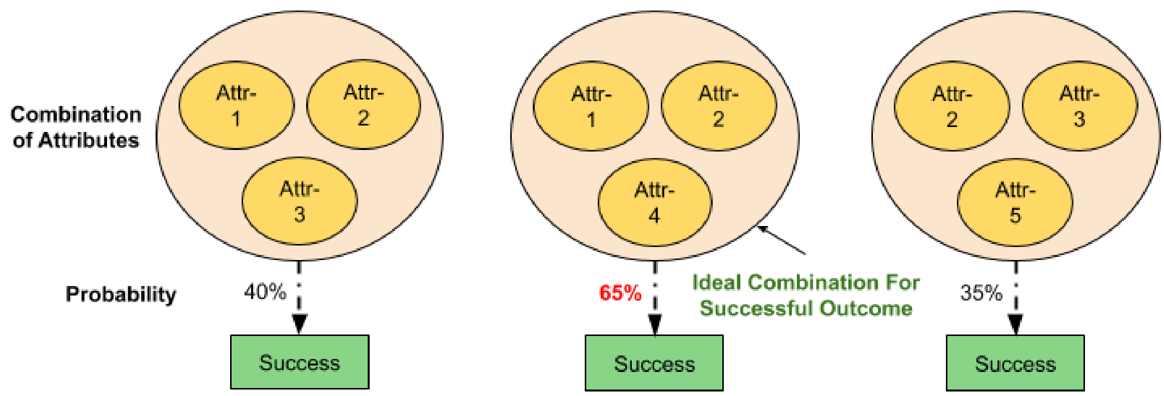

Figure 4.7: Combination of activities attributes with the outcome probability

Fig 4.7 represents the customization of attributes in the upcoming tasks to provide the feasible activity. Evaluation of the attributes is conducted by combining different attributes as one and estimating the one with the highest success probability. The resultant combination of attributes with the highest evaluation result is fused into the upcoming task. This process continues until the player needs assistance.

A lot of player data and game data are required to evaluate the combination of attributes and manifest an exact adaptation of this kind. Unfortunately, due to various limitations, it is not possible to generate data that can pave the way towards an exact implementation. Preferably, the basic implementation is conducted successfully to visualize the notion.

4.5.3 Tune Game Parameters

Game tuning could be achieved by enforcing the modification at the game level rather than the granular level. Game tuning involves changing the game's parameters, like difficulty level, game environment, and themes. The rule for this form of adaptivity is straightforward. Monitor the performance of the player and if the player's performance requires improvement, then adjust the difficulty level of the game, or change the game level user interface such environment and theme.

This thesis does not cover adaptivity manifestation imposed by tuning the game parameter, but it has been reversed for future work.

Chapter 5

Implementation

There have been enough details provided so far to understand the implementation of this thesis. The implementation part includes presenting an understanding of how the architecture looks and behaves at the abstract level. To address the research question entirely, we require functional components such as serious game (SG), and the framework that interacts with the respective SG and provides assistance. The candidate serious, as we have already discussed in the methodology chapter, is Simulator for Lost Earth Serious Game (SimLESG). The framework that assists an SG is called SeGaAdapt. From outside, SeGaAdapt appears to be a single entity, but it is a conglomerate. It is a combination of the sub-components that work at the atomic level and collaborate with each sub-components to provide the implementation of the proposed frameworks successfully. Now let's dig into SimLESG and SeGaAdapt.

5.1 Simulation for Lost Earth Serious Game

We have already been through about the customization and setup of SimLESG in the Methodology chapter, as well as the essential entities such as missions, activities and time of day. In order to interact with SimLESG, the player can either use the programming console or the web interface.

5.1.1 Programming Console

The programming console of SimLESG is contrived for interaction with SimLESG in Java programming language based on simple textual input and output. The player needs to read the information carefully and provide input only via keypress on the keyboard to interact with the simulator. In order to exit the programming console or to quit the mission, the player could insert character 'q'. The benefit of using the programming console is that it provides information in the most detailed form.

As you can see in Fig 5.2, the player is displayed the total number of available missions, and each mission is assigned with a unique number. The player can insert the number assigned to the mission to play that respective mission.

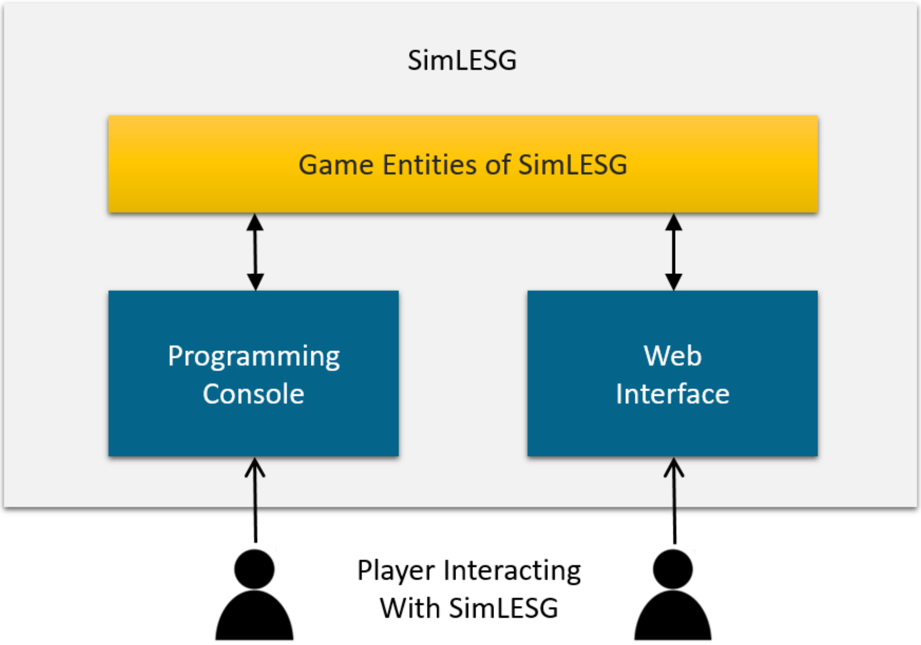

Figure 5.1: Player's Interaction with SimLESG

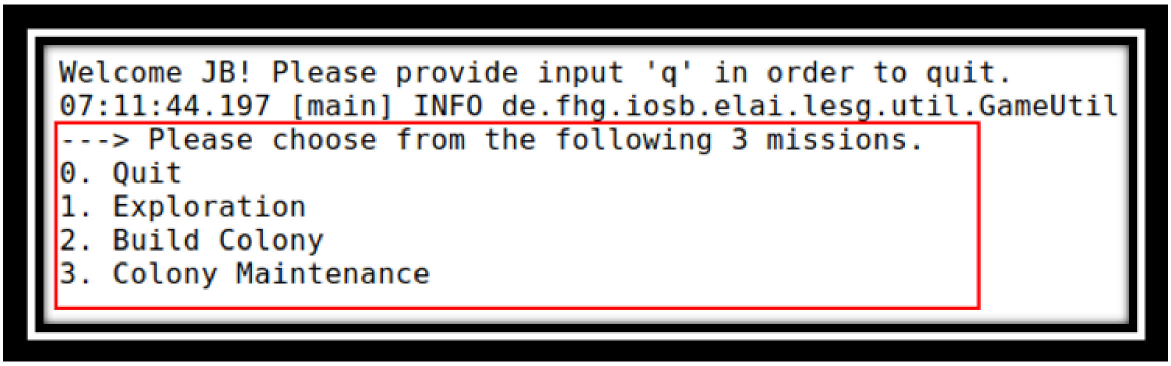

Figure 5.2: Screenshot of programming console from SimLESG

5.1.2 Web Interface

The web interface of SimLESG is developed in VueJS and HTML5. The web interface uses Java for configuration of game components used for simulation, and VueJS is used to bestow the dynamic nature, and the HTML components are used for representing game components. The web interface's internal behavior is the same as the Programming Console, with the only difference that it is provided with UI for interaction replacing Java instances with HTML components.

All missions are displayed by default, and upon selecting a particular mission, the underlying activities appear themselves. When the player starts the simulation game, there is always a $timeofday$ active, which can be changed anytime using the $'Next'$ button on the screen's top right section. The capability to change the $timeofday$ is provided so the player can change the time of day feasible for performing a particular. The $timeofday$ of change

Figure 5.3: Screenshot of web interface from SimLESG

in sequential order only, which is: $Dawn, Day, Dusk, Night$

5.1.3 xAPI Sender

The xAPI sender, embedded within the SG (like SimLESG), is assigned with the task of posting the xAPI statements regularly to the remote server (SeGaAdapt).

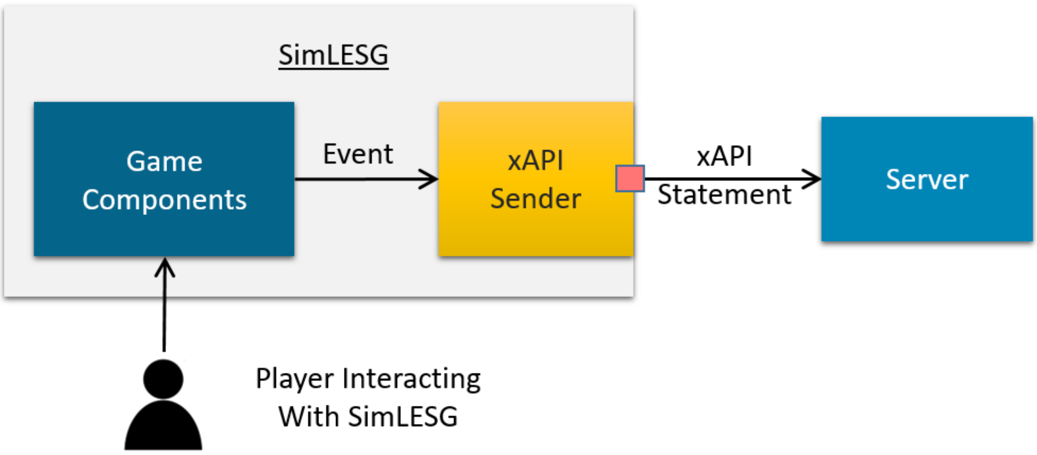

Figure 5.4: xAPI Sender in Simulator for Lost Earth Serious Game

The information about the player's actions is added to the $verb$. The $Object$ part of the xAPI statement expatiates detail concerning the entity for which the player acted. The $actor$

is part of the xAPI statement contains the name of the player.

Example of an xAPI statement sent by SimLESG:


```
{
 "actor": {
  "objectType": "player",
  "name": "jb",
  "mbox": "jb@mail_test.de"
 },
 "verb": {
  "id": "http://www.iosb.fraunhofer.de/simlesg/mission/started",
  "display": {
   "en-US": "started"
  }
 },
 "object": {
  "objectType": "mission",
  "id": "http://www.iosb.fraunhofer.de/schema/xapi/simlesg/mission/ColonyMaint
         enance",
  "definition": {
   "name": "Colony Maintenance",
   "description": {
    "en-US": "You are engaged in mission: Colony Maintenance"
   }
  }
 }
}
```


The above code section contains the xAPI statement in $actor - verb - object$ format of the most basic form, wherein the actor begins $ColonyMaintenance$ mission.

5.2 SeGaAdapt

SeGaAdapt is the adaptive framework located remotely and communicates with serious games (SGs) through xAPI. It consists of various components that work with each other to provide user assistance for SGs.

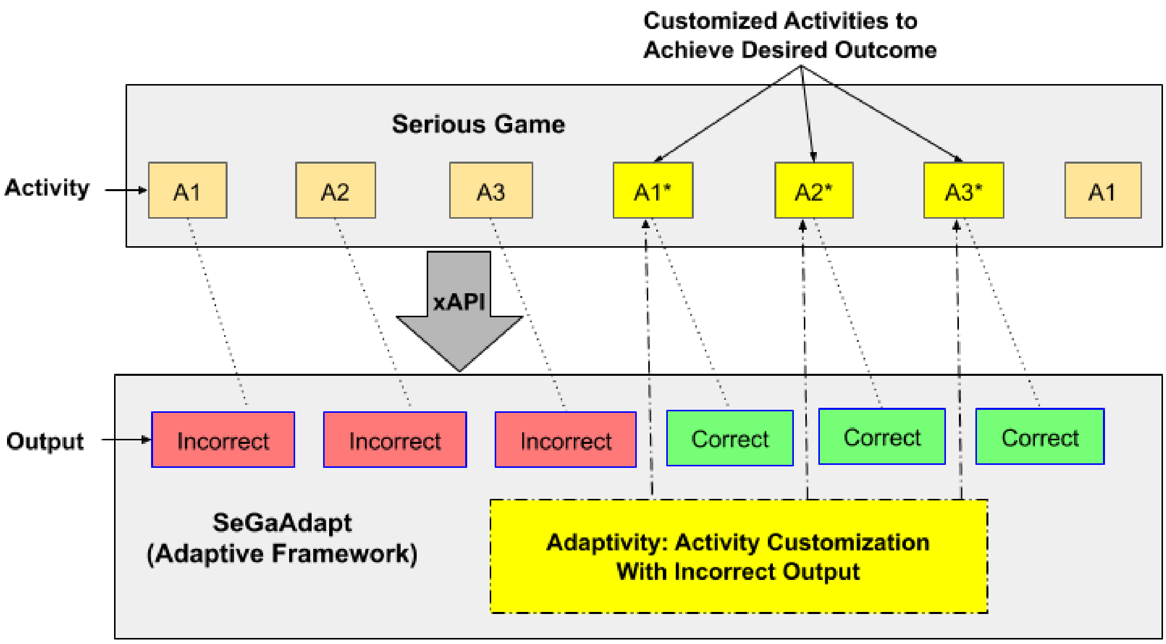

Figure 5.5: Activity Customization of Serious Game by SeGaAdapt [Yadav, Streicher, and Prabhune, 2020]

Fig. 5.5, depicts the implementation of the SG with SeGaAdapt framework. The player undergoes education and training through interaction with the SG by undertaking the tasks or exercises within SG. The player interaction leads to the generation of events. These events are activities and moves performed by the player, used for the composition of the xAPI statements by SG, and then sends it to SeGaAdapt. SeGaAdapt framework receives the xAPI statements from SG and starts performing several processes over the xAPI statements to make sense of the data. If the SeGaAdapt framework deducts that the player requires assistance, it responds to the SG with the adaptive strategy.

As you can in Fig. 5.5, activities A_i are non-assisted activities, and there is no involvement of the SeGaAdapt. SeGaAdapt only receives xAPI statements of the events generated while performing these activities. The activities A_i^* in Fig. 5.5 have undergone adaptive process within SG. SeGaAdapt provides the instructions of the adaptivity to the SG. The goal of producing adaptive instruction is to assist the player when the player is not doing well.

5.2.1 xAPI Tracker

Trackers are used for collecting the data of the player's interaction with SGs [Serrano-Laguna et al., 2017], and then the collection of data can be used for any required purpose. The received data can be persisted and may be used later for evaluation purposes, e.g., study the development, follow the progress/downfall, or the data can be processed dynamically and provided with output response in a synchronized fashion. It is debatable that the tracker lies on the client-side or the server-side. The arguments presented in favor of both the approach are quite reasonable, but it depends on where the data is getting sent or persisted. In our case, the data is recorded at the server-side; therefore, the tracker implementation lies at the server-side. When the tracker receives the data in the form of xAPI statements, the tracker is denoted as an xAPI tracker [Berg et al., 2016].

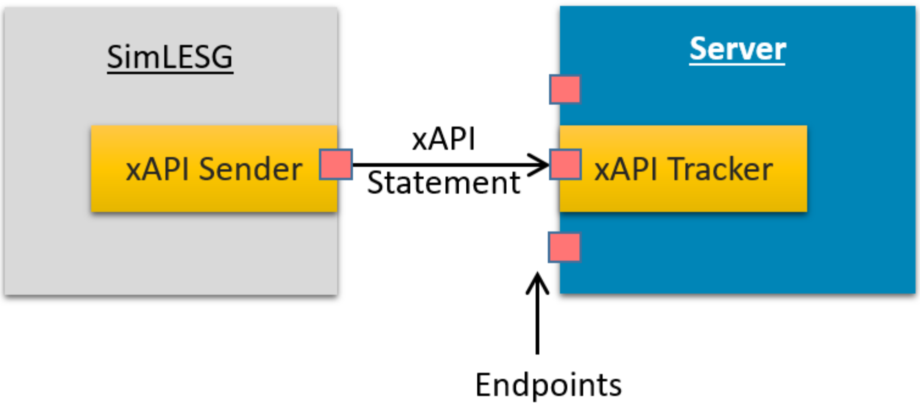

Figure 5.6: xAPI Tracker with Endpoints in Server

The xAPI statements sent by the xAPI sender, within the SG, is received by the xAPI tracker at the remote server [Berg et al., 2016]. The xAPI tracker is embedded within SeGaAdapt to constantly receive xAPI statement sent by an SG. In this way, the SeGaAdapt framework can track the player's performance and respond with an appropriate adaptive step whenever the performance dips below a specific threshold value. As you can see in Fig. 5.5, SeGaAdapt monitors the activities that have a successful outcome and an unsuccessful outcome using the xAPI communication channel. SeGaAdapt framework consists of the compass, which consistently keeps an eye on the activities. The compass does not step in if the player is doing well; instead, it's role is to guide the player when there is a rise in the number of unsuccessful outcomes.

5.2.2 Activity Flagging

Before we proceed further, let's understand the connection between activities and states. Suppose that the climate condition is sunny with a scorching temperature. There is a boy who has to go outside; the boy applies sunscreen and wears a cap before going out. Here the state is weather conditions outside, and the activity is the preparation before stepping out.

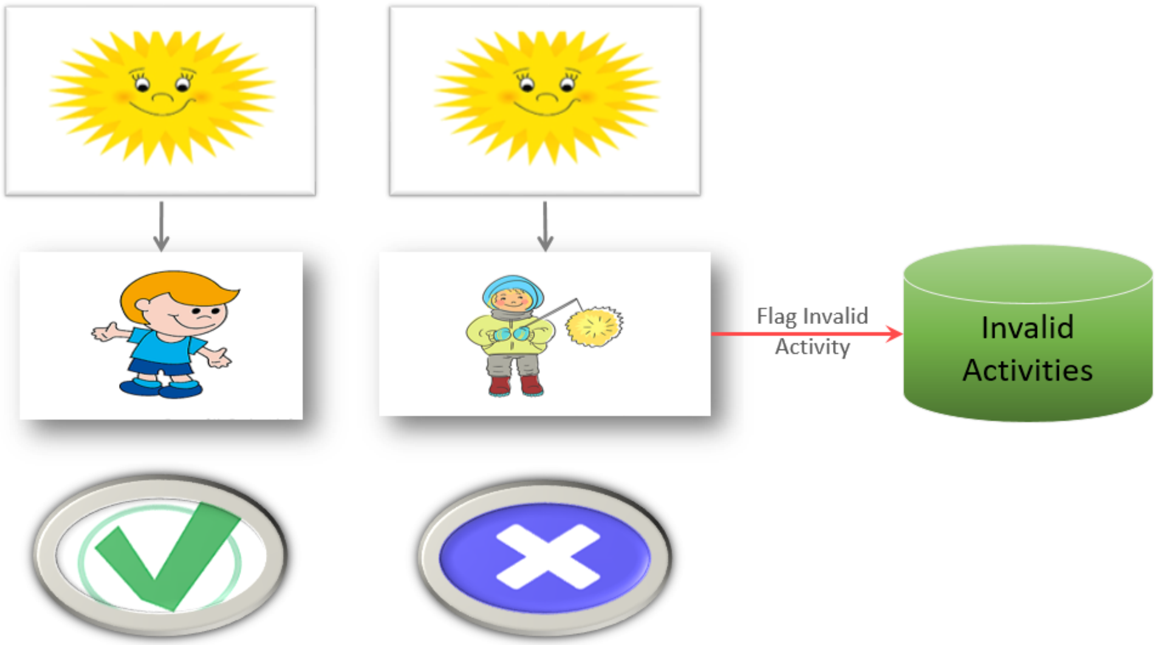

Figure 5.7: Relation between State and Activity and Activity Flagging

Activities are performed by keeping the state's perspective in mind, so we can say that the states affect the activity. The framework is connected to the database that contains the records of the activities with appropriate input for a given state; the activities with the invalid input for a given can be determined using those records. The activities provided with invalid input by the player are unsuccessful outcomes, and the compass flags these unsuccessful outcomes. When the player is about to undertake the task or exercise that are flagged (had incorrect result), the framework sends the adaptivity signal to the SG. Using the received response, the SG can take appropriate measures to get a successful outcome.

5.2.3 Assistance as Adaptivity

In Fig. 5.5, activities A_i are the ones that are performed by the player while playing SG. The output of these activities is not affected by the external component, like the SeGaAdapt framework in this case. As you can see in Fig. 5.5, there are many activities A_i that are not performed correctly by the player. This phenomenon implies that the player is not doing well, and there is a conclusion drawn by the framework that the output of future activities would likely follow the same faith. As a result, the framework comes into the picture so that it can assist those activities that fall under the category of 'not done correctly' in the future. In Fig. 5.5, activities A_i^* are the ones that are performed by the user, which have undergone

[0]Animated image courtesy of [ClipArtsFree, 2020]

some kind of adaptivity orchestrated by the SeGaAdapt framework, thus, likely changing the consequence of the output as compared to earlier. The prime focus lies in the outputs A_i^*, because they are associated with the proposed solution and merits of this thesis; as per the proposed theory, they alter the course of learning and provide the desired learning outcome.

5.2.4 Components of SeGaAdapt

The SeGaAdapt framework comprises various components that collaborate with each other to process the player's event data and provide assistance.

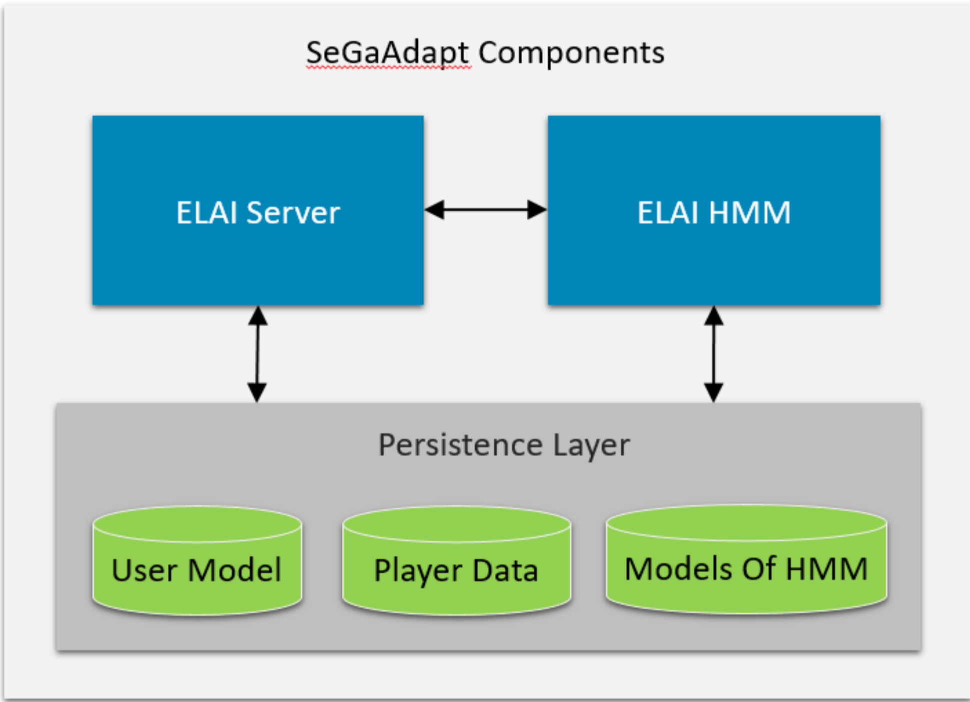

Figure 5.8: Architecture with SeGaAdapt with eLearning Artifical Intelligence (ELAI) Server and ELAI Hidden Markov Model (HMM) Micro-Service

SeGaAdapt framework comprises of three components are:

1. e-Learning using Artificial Intelligence (ELAI) Server
2. Hidden Markov Model Micro-Service (ELAI-HMM)
3. Persistence Layer

From these three component two of them fall under category of applications: ELAI Server and ELAI-HMM. We will learn in detail about the mentioned applications in the upcoming sections. Persistence player contains user model, player data or xAPI statements (events produced during gameplay), and trained models of HMM. The entities that are responsible for persisting player data and models, and player data are: (1)Learning Record

Store (LRS) and (2) Database (Neo4j). Player data consists of game events, scores, achievements, and user profiles of every player. Neo4j graph database is the most feasible candidate for implementing the player model, data, and analytics in graph format.

5.3 e-Learning using Artificial Intelligence (ELAI) Server

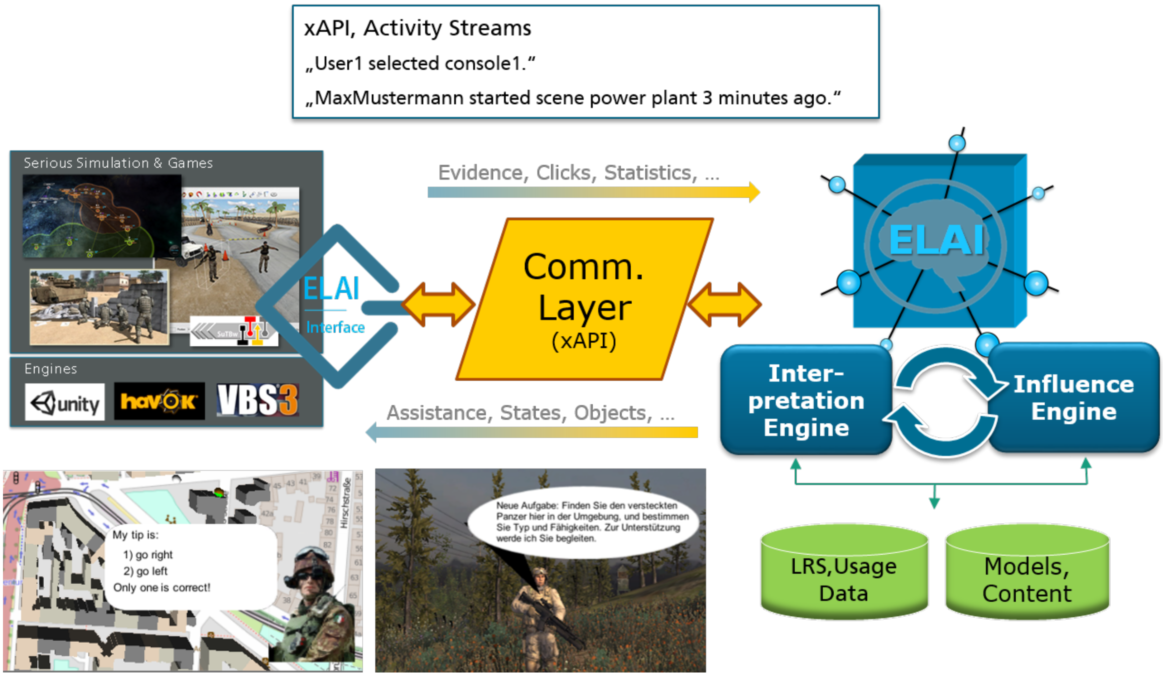

Figure 5.9: Architecture with Experience API [Streicher, 2018]

Fig 5.9 outlines an architecture of the adaptive framework that follows the xAPI specification called e-Learning using Artificial Intelligence (ELAI). ELAI project was started in 2019 by the Interoperability department at Fraunhofer IOSB, and it is developed as a web application as well as a web service. The ELAI project aims to provide a smart intelligence-based adaptive feature for a better e-Learning experience for entities such as e-learning devices and serious games (SGs). Since, the ELAI follows xAPI specification, which means 2 things:

1. ELAI server interacts with SGs and e-learning devices through xAPI as a communication channel
2. ELAI server uses Learning Record Store (LRS) to persist the data.

The communication channel receives the xAPI statements comprising actions and events triggered by the player within the SG and e-learning devices. These xAPI statements are in JSON format, and they are stored as raw JSON data in the LRS. Apart from communication and persistence, various engines also exist within ELAI, such as interpretation engine and influence. Interpretation engines are responsible for rendering the received events and inputs to scores and various other entities to make more sense of the data. Influence engines are responsible for altering the performance of the learning process. Interpretation engines and influence engines work hand in hand for manifesting adaptivity efficiently.

5.4 Hidden Markov Model Micro-Service (ELAI-HMM)

The proposed solution that provides assistance using HMM is implemented on the micro-service application called ELAI-HMM.

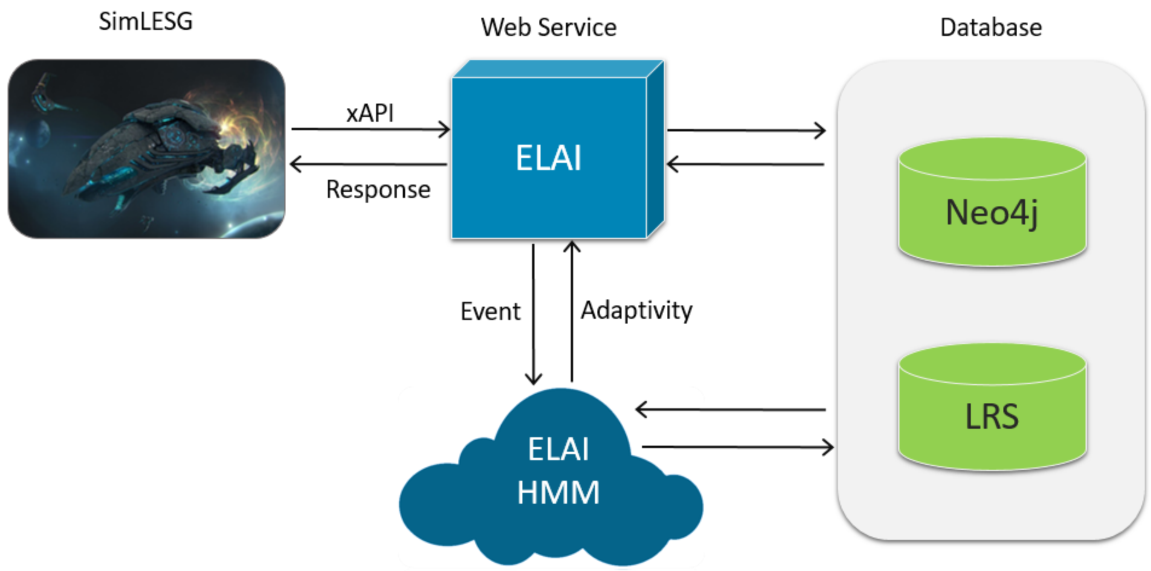

Figure 5.10: Architecture of SeGaAdapt

Fig 5.10 represents the architecture of all the prime components of SeGaAdapt interacting with each other. In this architecture, Simulator for Lost Earth Serious Game (SimLESG) communicates through xAPI with e-Learning using Artificial Intelligence (ELAI) server that acts as a web service/application. ELAI server receives the xAPI statement containing information about the event performed by the player. The event data is wrapped within the Actor-Verb-Object sections of the xAPI statement. ELAI server executes the process of its own, using event data. After performing its task, the ELAI server then sends the event data further to the ELAI-HMM microservice using the HTTP protocol, and it waits for the response. ELAI-HMM communicates with ELAI server accepting player's event data and providing assistance ELAI-HMM analyzes the event data to check whether the player requires assistance or not. If it concludes that there is no need to assist, then the 'All Okay' signal is sent to the ELAI server; otherwise, it sends the response containing assistance information back to the ELAI server.

5.4.1 Components of ELAI HMM

Hidden Markov Model (HMM) can provide better assistance only in case if the model is well trained and the player's moves and performance is tracked. To monitor and provide the assistance, we need an approach that involves an entity that acts like a compass specialized in observing each player's events and adapting if given the player needs assistance.

ELAI-HMM comprises of a compass and trained HMM. HMM is trained using the player and the programming objects that represent the states and observables. Fig **??** represents the components of ELAI-HMM and communication among its internal components, and also external entities.

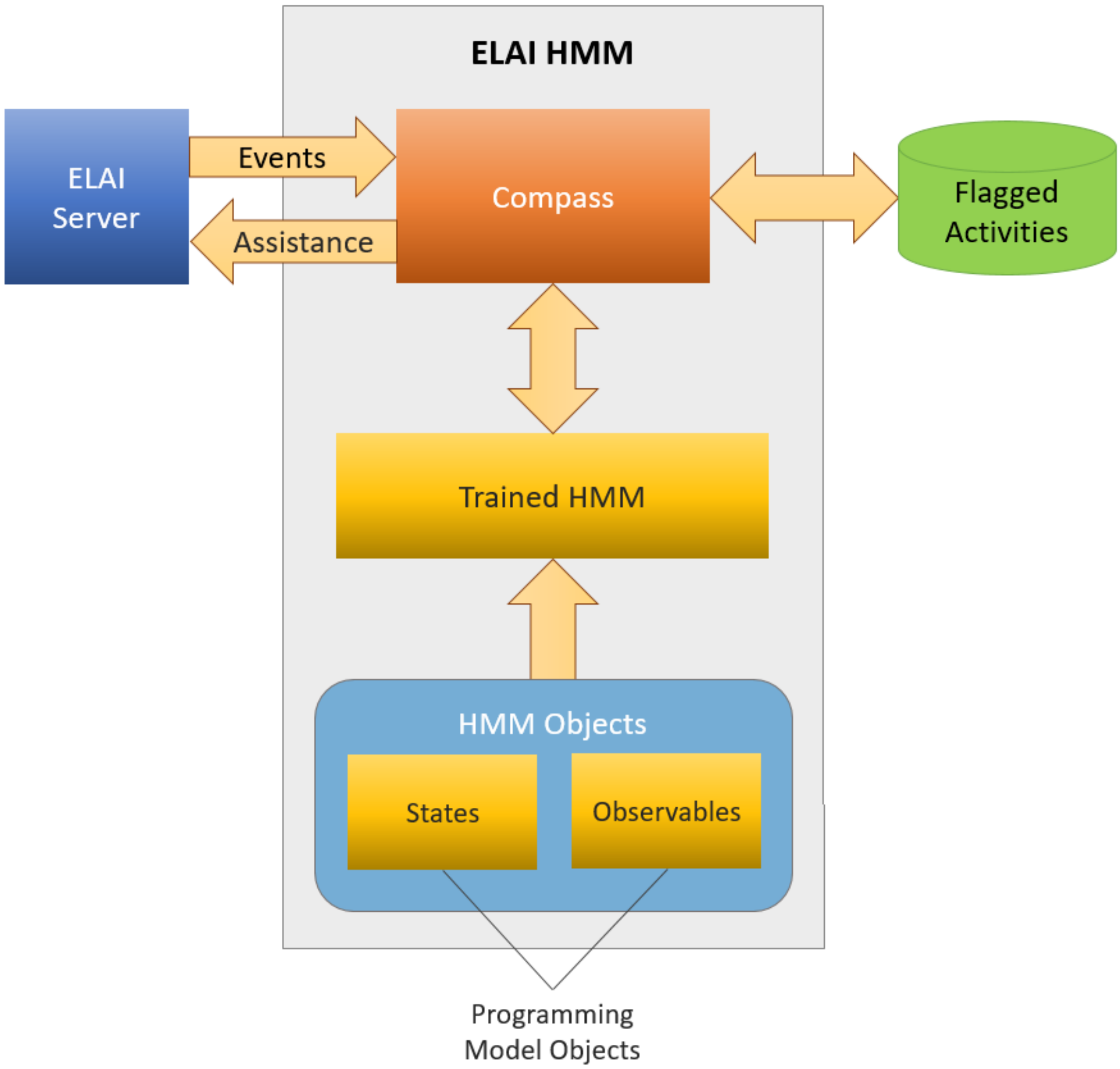

Figure 5.11: Components of ELAI HMM

The compass plays the main role in ELAI HMM to process the input data and provide assistance in response. It continually monitors the input player data coming from the ELAI server. The compass processes the input data received from the ELAI server to check if the player is on the right track. To conclude, this compass works with the trained HMM by plugging the event data to the states and the HMM observables. The transition and emission probabilities from the trained HMM are used to evaluate whether the player has performed the activities appropriately. If the player is doing the activities in the desired manner, then 'All Okay' signal returned by ELAI-HMM to the ELAI server. In case if the activity is not performed correctly, then the ELAI-HMM persists the players' wrong moves and the state in which it was carried out into the database. These persisted values are used along with the probability values from the trained HMM to assess the feasible activity; consequently, discovered feasible activity is used for the developing assistance response.

5.5 Compass

Using the compass, ELAI-HMM can take necessary measures to maintain the player's performance by monitoring the activities and performance parameters such as the player's score. The compass triggers the adaptivity requirement signal (like a distress signal) whenever the player's score reaches the minimum threshold. If the current score is comparatively lower than the average score, then the adaptivity can be enforced for the upcoming tasks by providing hints or customizing the conditions/attributes.

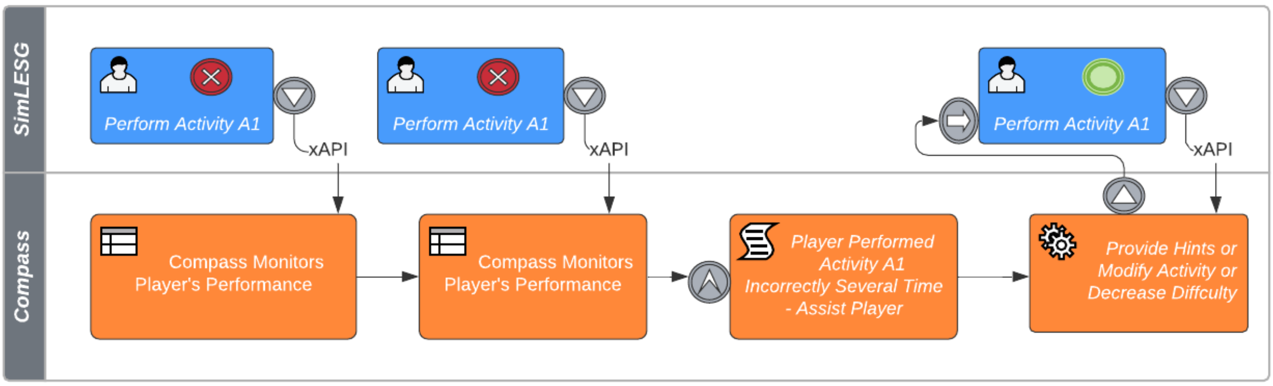

Figure 5.12: Working of Compass

In the proposed solution, the serious game communicates with the ELAI server, and then the ELAI server takes the help of ELAI HMM to provide assistance. But in Fig 5.12, the interaction between the serious game and the compass is portrayed directly to avoid complexity. Fig 5.12 represents the compass in action by monitoring the activity A1 using event data and assisting the player in performing activity A1 correctly.

Chapter 6

Application

This chapter focuses on applying the proposed solution using the Simulator for Lost Earth Serious Game (SimLESG) and all components of the SeGaAdapt framework. The topics, concepts, logic, and entities mentioned in the previous chapters are applied in this chapter. This chapter highlights that the proposed solution approach works and addresses the problem statement by showing the SimLESG and units of the SeGaAdapt framework in action. The activity diagrams, UMLs, use cases, mock-ups presented in this chapter will implement a better understanding of the idea and workflow for you.

6.1 GamePlay In SimLESG

The methodology and implementation chapter has already covered a great deal of understanding regarding Simulator for Lost Earth Serious Game (SimLESG). Therefore, this section begins with some overview and the application of SimLESG for gameplay. SimLESG mimics the behavior of Lost Earth Serious Game (LESG) of the most fundamental forms and scenarios. There are two ways to interact with SimLESG (1) programming console and (2) web interface. For simplicity of display and testing purpose, interaction with SimLESG through web interface is considered.

SimLESG communicates with the framework prototype for technical verification as a proof of concept. SimLESG contains xAPI formatter and xAPI sender to communicate with the framework. Each interaction of the player with SimLESG is transmuted into xAPI statement by xAPI formatter and the resultant xAPI statement is forwarded to xAPI sender. The xAPI sender dispatches the xAPI statement to the endpoint(s) of the SeGaAdapt configured within SimLESG. For application purposes, the two aspects from LESG are considered: (1) time of day (states) and (2) sensor (observables).

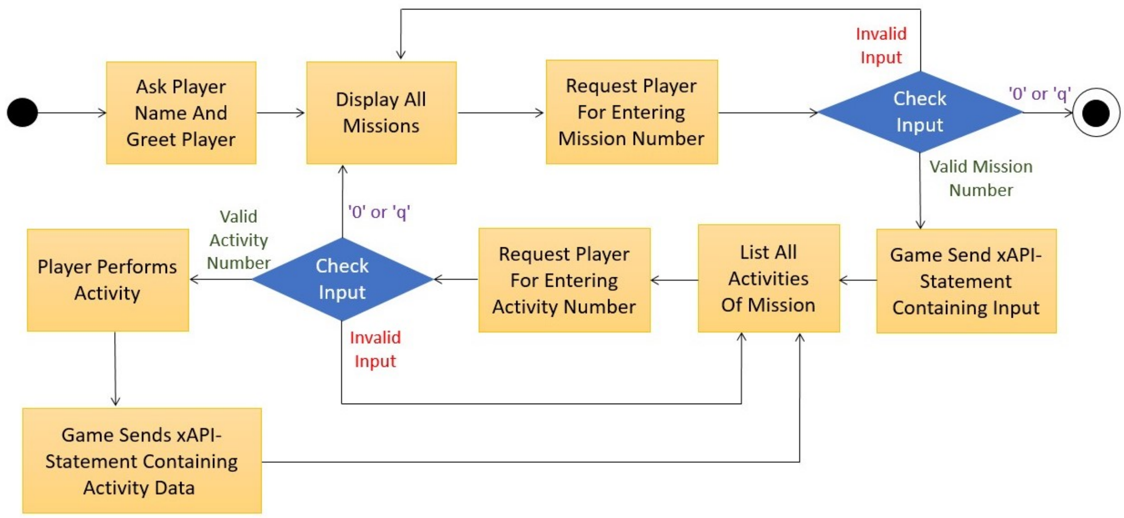

Figure 6.1: UML - GamePlay workflow while playing SimLESG

Fig 6.1 is a UML of the gameplay in programming console but works in the same way for the web interface. As you can see, SimLESG sends events performed by the player, such as the choice of mission and performing of activities.

6.2 Application of HMM

HMM: The HMM is $\lambda = (A, B, \pi)$. A is transition probability of states (hidden states), B is emission probabilities of observables (visible states) from states (hidden states), and π is the initial probability.

Let's discuss them in detail concerning the given states in the application and the probability value.

6.2.1 Emission Probability

The emission probability of observables from the states represents the likelihood of encountering the observable for the given state [Rabiner and Juang, 1986]. The sensors and time-of-day are ideal candidates for representing observables and states within the game because there is a connection between sensors and time of day (see 2.2). Imagery sensors are one of the game's producible artifacts which are deployed to collect image data. Each sensor has similarities and differences, which makes choosing the type of sensors crucial for performing the activities correctly. The sensor deployment activity is performed to illustrate the visual interpretation of the environment from the aerial view.

Deployment of a sensor that is not suitable to the given state (like time of day) portrays that the sensor deployment activity is not executed properly, resulting in loss of points and mission failure in Lost Earth Serious Game. The 3 available sensor types are (1) electro-optical (EO), cheapest but efficient only during the day; (2) radar (SAR), costly and efficient in all times of day unless distorted; (3) infrared (IR), costly and efficient in all times of day.

Table 6.1: B: Emission Probability

	EO	IR	SAR
Dawn	0.2	0.5	0.3
Day	0.7	0.2	0.1
Dusk	0.15	0.4	0.45
Night	0.1	0.4	0.5

Any sensor can be deployed at any given time of day, but the type of sensor must be carefully selected due to atmospheric variations and deployment costs. Table 6.1 is emission probability of sensors (observables) from the states (time of day) representing the sensor's effectiveness for the given state.

6.2.2 Transition Probability

The transition probability is the likelihood of occurring the next state after the current in the future [Rabiner and Juang, 1986]. Time of day (see 2.2 can undoubtedly represent the state that is in progress at the given time. It denotes the ongoing time of day within the game, which keeps on changing. The available time of day is dawn, day, dusk, and night.

Table 6.2: A: Transition Probability

	Dawn	Day	Dusk	Night
Dawn	0.0	1.0	0.0	0.0
Day	0.0	0.0	1.0	0.0
Dusk	0.0	0.0	0.0	1.0
Night	1.0	0.0	0.0	0.0

Table 6.2 represents the values of transition probability of the number of state, and in our case, it is the time of day. The probability values from Table 6.2 signify the possibility of occurring the upcoming time of day after the current time of day is over.

6.2.3 Initial Probability

The initial probability of the state in HMM provides the ability to estimate the most initial state's plausibility when the system (assumed to be HMM) begins [Rabiner and Juang, 1986]. It is calculated using the time series data accumulated of the state.

Table 6.3: π: Initial Probability

	Dawn	Day	Dusk	Night
Probability	0.25	0.25	0.25	0.25

Table 6.3 holds the values of the initial probability of time of day. Since the π initial probability of all the states is equal, each state (time of day) has a possibility of 25% to occur when the system begins.

6.2.4 Holistic View of HMM

For adaptivity manifestation, the system provides assistance using the entities of HMM that we discussed in the prior sub-sections. A well trained HMM consists of values defined for emission probability, transition probability, and initial probability.

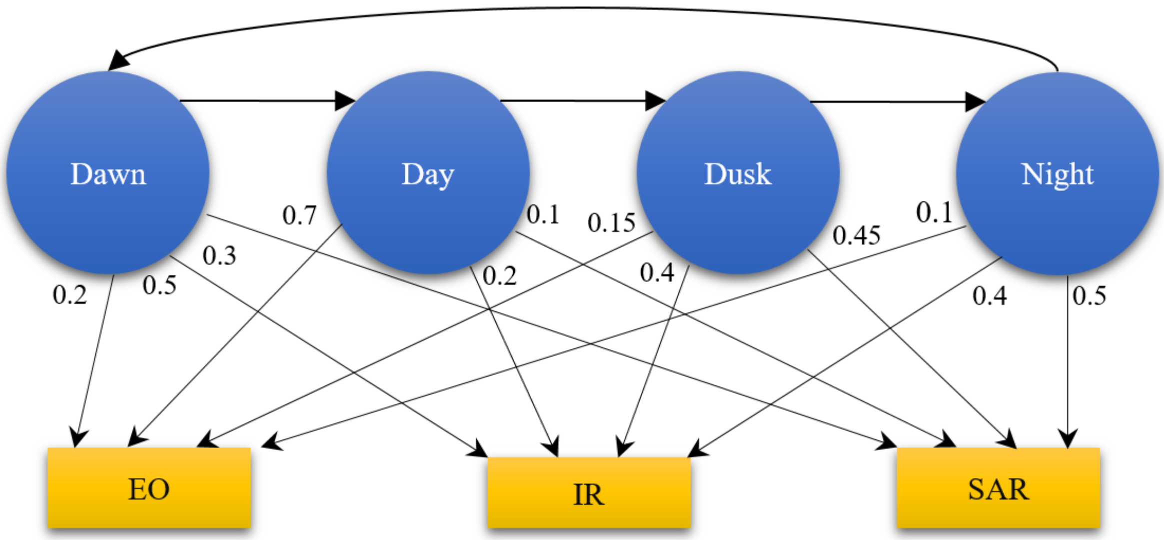

Figure 6.2: HMM probabilities with time of day and sensor

Since the HMM is a form of Bayesian network [Ghahramani, 2001], it can be viewed as a graphical representation as portrayed in Fig 6.2. Fig 6.2 provides a holistic view is developed using values from Table 6.1 and Table 6.2.

6.3 Workflow

Fig 6.3 is the activity diagram representing the workflow and component interaction. It describes the operation of Simulator for Lost Earth Serious Game with the SeGaAdapt framework. Fig 6.3 portrays the functional components within the SeGaAdapt framework, such as performance check, compass, and HMM is incorporated along with the intercommunication between them at the granular level.

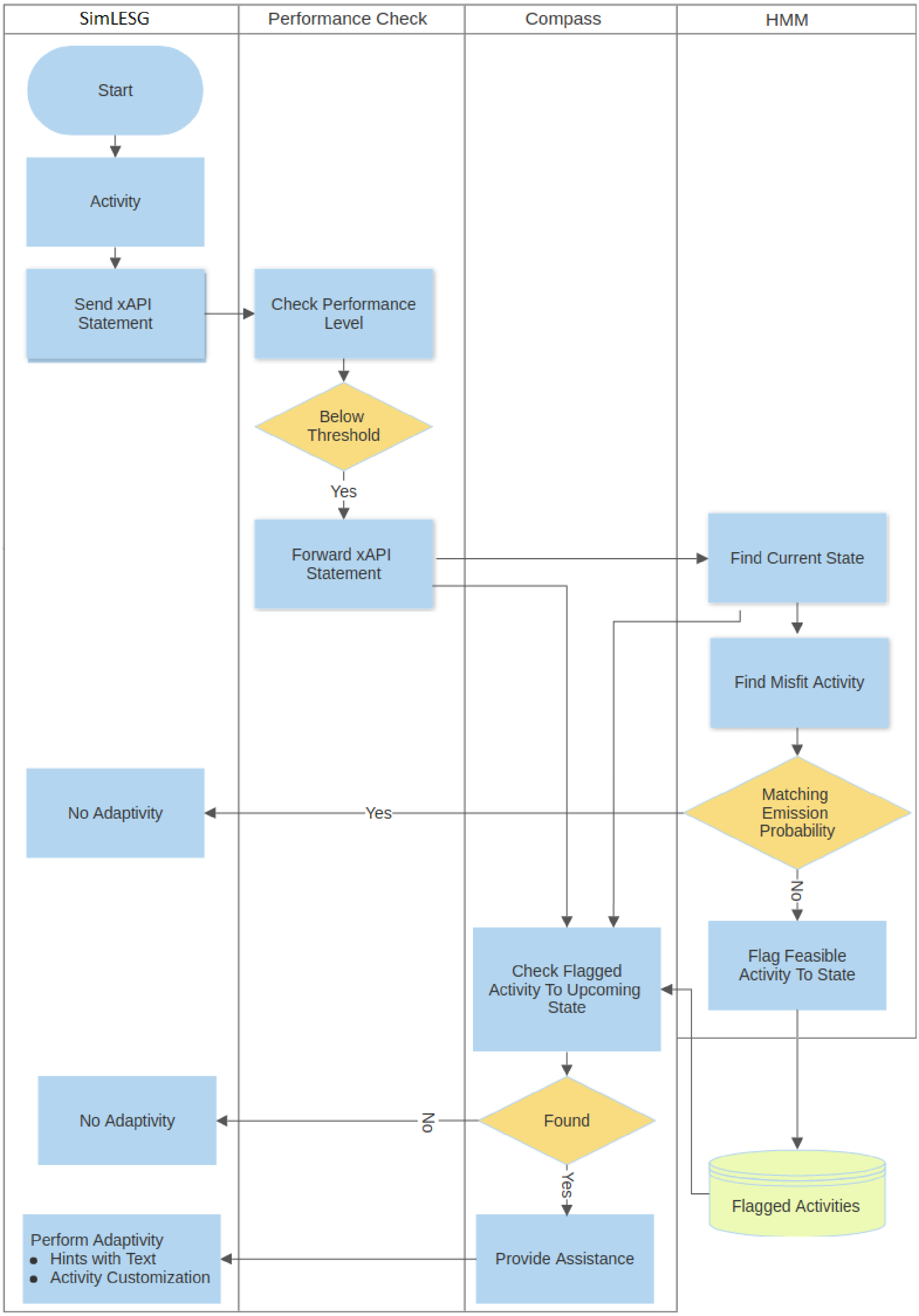

Figure 6.3: Activity Diagram Of Framework

6.4 Adaptivity Manifestation

When the game starts, the information about the initial state is sent to the framework. Using the initial state and the transition probability from Table 1, the system keeps track of the current state and is able to predict the future state using Markov chain, and meanwhile the HMM checks for the incorrect activity. The two adaptivity manifestions to assist player has been implemented for the prototype: (1) provide hints and (2) customization of task.

6.4.1 Assistance using Task Customization

Assistance can be provided through customization by modifying the attributes of the tasks. In this way, the player will have fewer chances of committing mistakes.

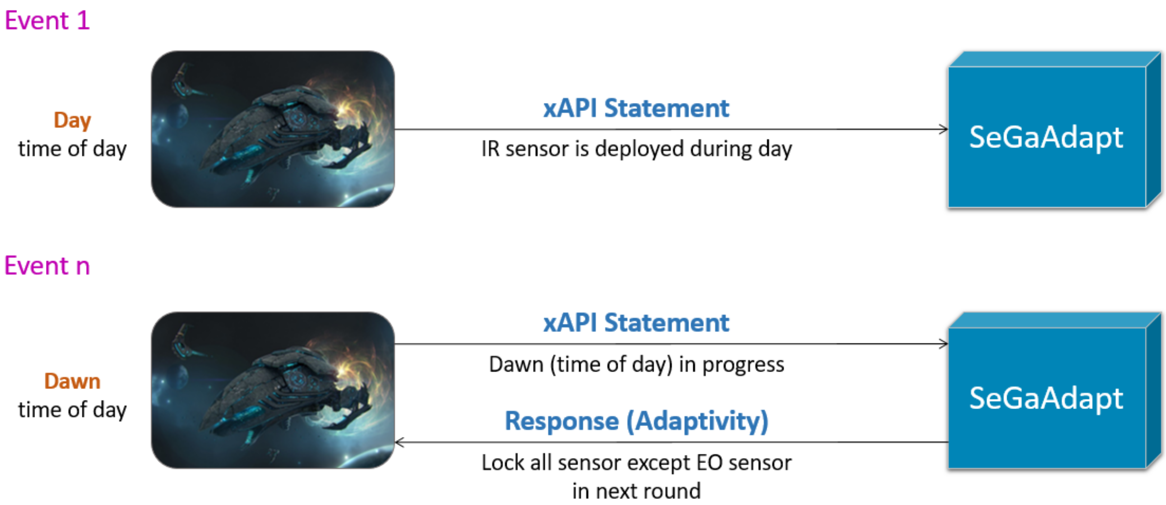

Figure 6.4: Adaptivity - Sent instruction to lock all all sensors except EO sensor during dawn

The SG and framework behave exactly same as Instance 1 with one difference: the SG locks the non-feasible game element, hence no chance of its utilization. The player used the IR sensor during day. As per the emission probability from Table 2 the EO sensor is much more feasible for day, therefore during day all sensor types except EO are being disabled and enabled again only after completion of a "day"-round.

6.4.2 Assistance using Hints

Adaptivity manifestation using hints to provide assistance is applied successfully, and it behaves as per the expectation.

While performing the sensor deployment activity in round x, the EO sensor is deployed during dusk. As per the emission probabilities from Table 2, EO sensor is not feasible for dusk and the most feasible one is IR sensor, thus IR sensor is flagged to 'day'. Using transition probabilities (Table 1), a deduction can be made that the round $x + 3$ will be day

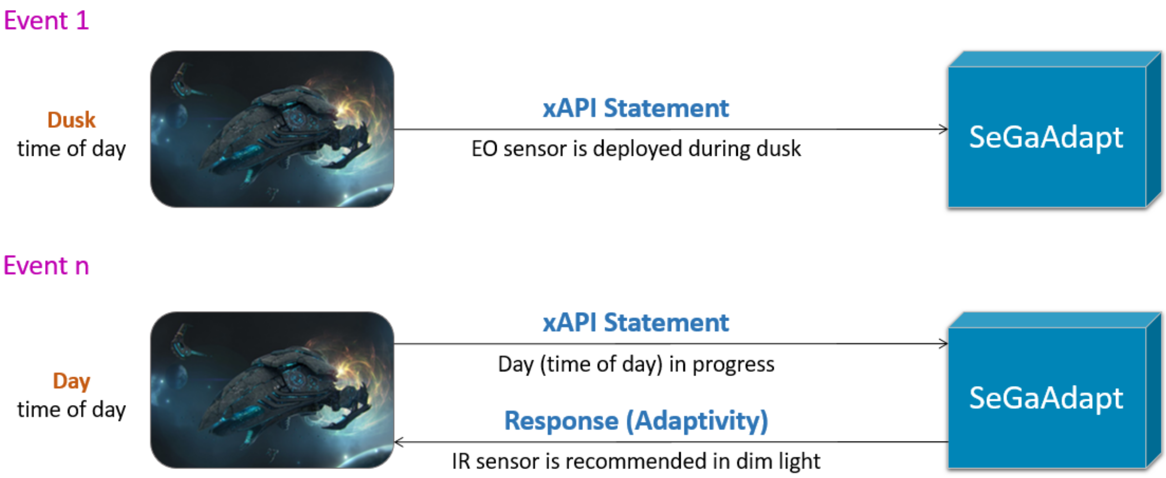

Figure 6.5: Adaptivity - Hint to deploy feasible sensor during dusk

and the round $x + 4$ will be dusk. Upon reaching the round $x + 3$ the compass checks for the flagged sensor to the upcoming state i.e., day and sends message to SG. The SG displays message stating ”IR sensor would be more feasible for dusk”. This gives the chance to the player to deploy IR sensor in the next round.

6.5 Assistance Imitation in SimLESG using SeGaAdapt

To provide the actual result for the proposed solution, the Simulator for Lost Earth Serious Game (SimLESG) is connected to the SeGaAdapt framework in realtime, and the player poorly executes the sensor deployment activity.

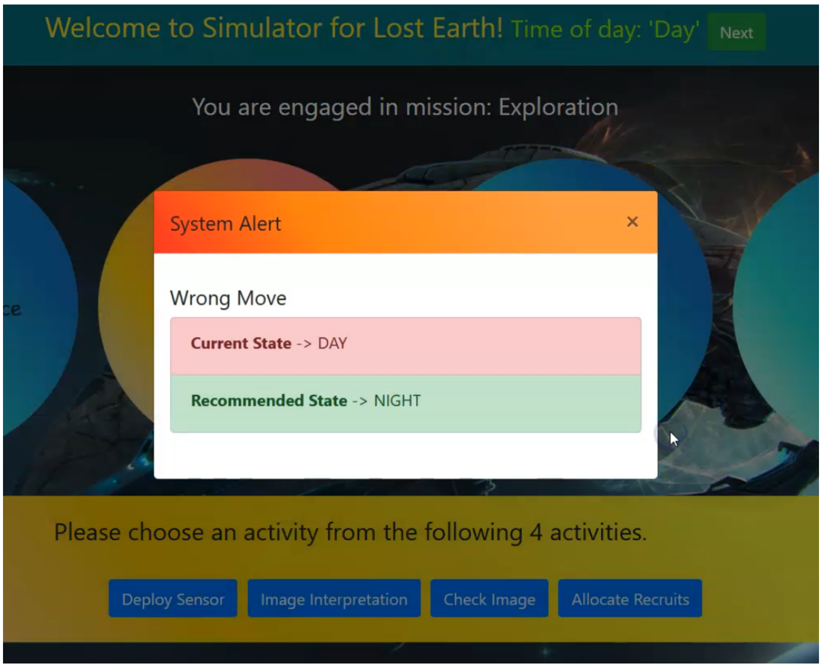

Figure 6.6: Wrong move detection in SimLESG

The player deploys an IR sensor when the time of day in progress is day. Fig 6.6 is the screenshot from SimLESG displaying information sent from the SeGaAdapt framework. The SeGaAdapt framework was able to deduce that the sensor deployment activity is not completed correctly using trained HMM (see Fig 6.2). The SeGaAdapt persists this wrong move and the state to assist the player with the sensor deployment activity in the future.

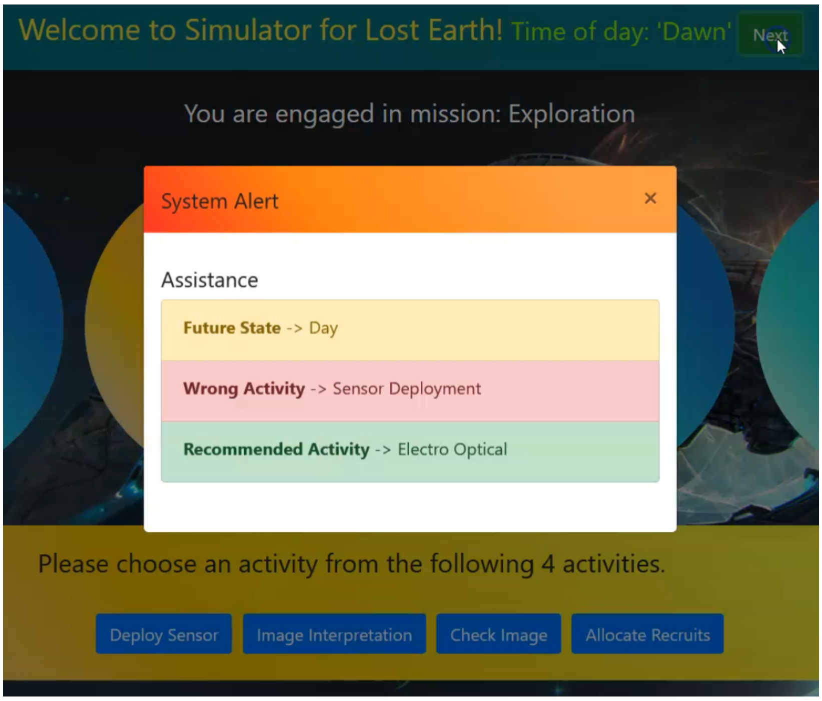

Figure 6.7: Assistance in SimLESG

The player carries on with the gameplay, but as soon as the player reaches to $dawn$ time of day, SeGaAdapt can predict that the upcoming time of day after $dawn$ would be day (see Fig 6.2). As you can see in Fig6.7, the SimLESG presents the response sent by SeGaAdapt when $dawn$ time of day is progress. It sends the recommendation to conduct the sensor deployment activity appropriately in the upcoming state, which is day in this case.

6.5.1 Role of Compass

There are various scenarios in which the SeGaAdapt framework can prove to be quite useful. Fig 6.9 portrays one of the use cases with regard to providing assistance while performing the sensor deployment task. Let's assume that the player deployed the sensor at an inappropriate time of the day (or timeline); therefore, that particular activity flagged incorrect sensor deployment. When the player is about to perform the same task, the system behaves as a guidance mechanism and asks the user to check the time of day before beginning with the sensor deployment task.

This entire process leads to the sensor's deployment at an appropriate time of day, which further leads to a successful outcome. The desired result is possible through the learning process by assisting the player at the right time.

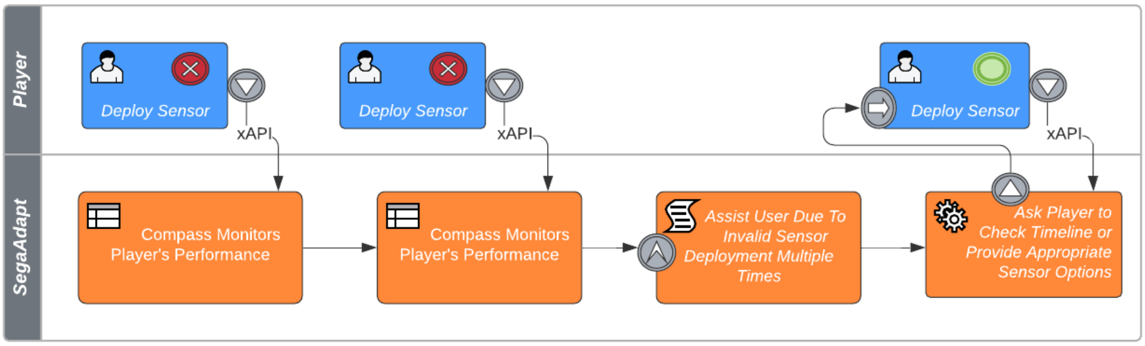

Figure 6.8: BPMN of adaptivity for sensor deployment task

As you can see in Fig 6.8, when the player deploys the sensor which is not feasible for the state in progress, then that particular sensor deployment task is persisted by the system. When the user is about to approach the state where the mistake was made, the system does not intervene, but rather it waits for the player's performance to drop until a certain extent is configured within the system. When the performance reaches the configured threshold value, the system decides to indulge and perform the assistance process.

6.6 Use Case in Lost Earth

The main aim of the thesis is to provide adaptivity for serious games (SGs). Most like candidate SG was Lost Earth Serious Game (LESG), but it was not possible to provide the implementation in such a short time; therefore, the alternative approach of having a simulator got finalized to support the application of proposed solution. Implementing the user assistance functionality within LESG requires constant communication with SeGaAdapt and the ability to manifest adaptive strategy suggested by SeGaAdapt. For constant communication, the LESG must be equipped with an xAPI statement sender to transmit the event data to the remote server. To manifest the adaptive strategy suggested by SeGaAdapt, LESG must be able to show hints using the alert dialog or screen and must be able to, customize the attributes of the future task and adjust parameters at the game level. Fig 6.9 is the mock-up that represents the implementation of the user assistance feature within LESG using SeGaAdapt in a nutshell.

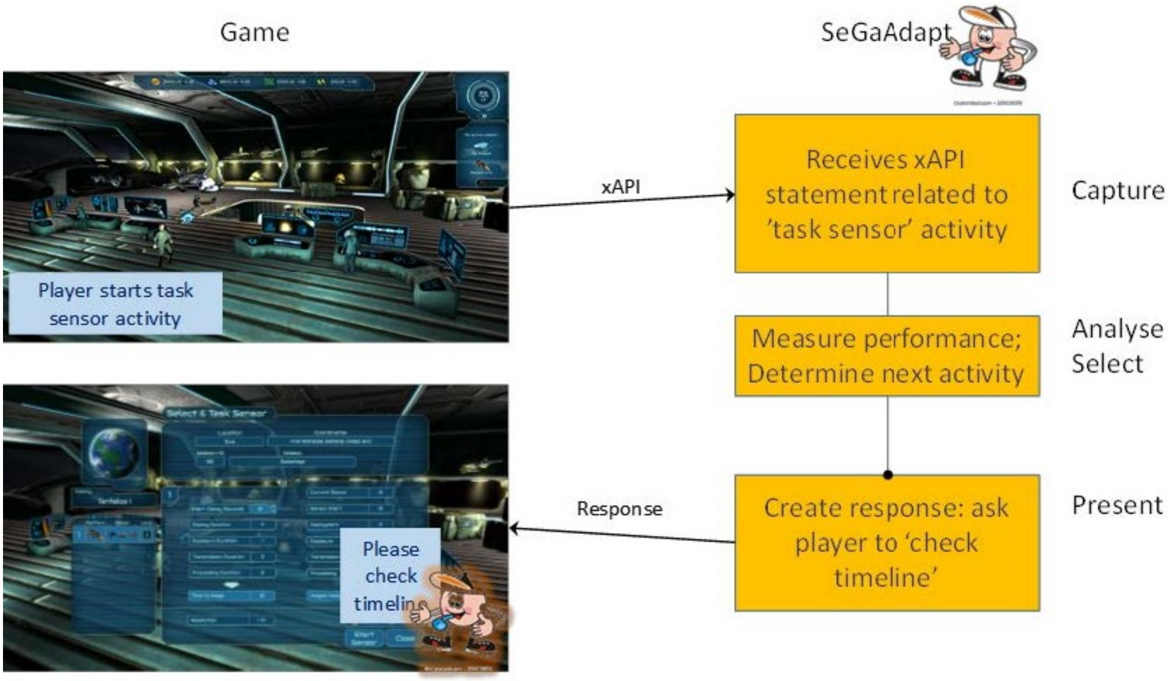

Figure 6.9: Use case of adaptivity for sensor deployment task

Fig 6.9 portrays the desired implementation of the adaptivity when the user makes a wrong move, and LESG is able to express the response provided by the SeGaAdapt framework.

Chapter 7

Conclusion and Outlook

The applications conducted using the Simulator for Lost Earth Serious Game (SimLESG) and SeGaAdapt the adpativie framework addressed the research question efficiently. The entities falling under xAPI specification, like xAPI statements, served as a perfect tool to monitor the events generated by the player and analyze at the time to understand the player's stance.

SimLESG covered most of the conditions, specifications, and behaviors expected from a serious game (SG), especially from Lost Earth Serious Game (LESG), such as mimic the missions, activities, and most importantly time of day. New missions and activities can be added to SimLESG by changing values in the JSON configuration file; likewise, modification can also be made by tweaking values from the same JSON configuration file. SimLESG continuously sends each and every event triggered by the player within SimLESG to the SeGaAdapt framework as an xAPI statement.

The assistance mechanism based on Hidden Markov Model (HMM) was implemented successfully within the micro-service application, which consistently received input concerning the player's actions and provided adaptive techniques as a response whenever the players performed dipped below. Multiple-use case experiments carried out during the application phase has favorably supported the claim regarding the competence of the proposed solution as a useful tool for user assistance. HMM was able to distinguish the misfit activities for a given state (missions and time of day) in the game and also provides the most feasible activity. This realization enforced the application to assist the player by not letting the player perform the same mistake twice. Feedback from the framework provided the player with the chance to be well prepared in advance and gather resources or the right tool to perform the upcoming activity more efficiently.

SimLESG manifested the adaptivity responses provided by the SeGaAdapt framework pretty efficiently, like displaying the hints and decreasing the game's difficulty level. The decrease in difficulty level allows the game to modify activities that are easier to perform as compared to earlier while displaying the hints permitted the player to act on the assistance response provided by the SeGaAdapt framework.

Scope for Further Research

This is not the end of the road, but rather a beginning of the next chapter. The future work includes:

- Manifest adaptivity in the actual Lost Earth Serious Game.
- Perform evaluation using different players.
- Conduct survey of related to the effectiveness of assistance.
- Investigate adaptivity feasibility with other serious games.

Bibliography

ADL.net (2013). *xAPI Specification 1.0.1*. Publication Title: 2014. URL: https://github.com/adlnet/xAPI-Spec/blob/a752217060b83a2e15dfab69f8c257cd86a888e6/xAPI.md (visited on 08/26/2016) (cit. on p. 5).

Aljundi, Rahaf et al. (2018). "Memory aware synapses: Learning what (not) to forget". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154 (cit. on p. 8).

Allbeck, Jan (2018). *Serious Games*. URL: https://cs.gmu.edu/~gaia/SeriousGames/index.html (cit. on p. 3).

Allen, Robertson Lee (2012). "War Games at Work: Networks of Power in the US Army Video Game Project". PhD thesis (cit. on p. 2).

Alonso-Fernández, Cristina et al. (2019). "Applications of data science to game learning analytics data: A systematic literature review". In: *Computers & Education* 141, p. 103612 (cit. on p. 28).

Andrade, Kleber De O et al. (2013). "Rehabilitation robotics and serious games: An initial architecture for simultaneous players". In: *2013 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)*. IEEE, pp. 1–6 (cit. on p. 31).

Atorf, Daniel (2018). *Lost Earth 2307*. URL: https://www.iosb.fraunhofer.de/servlet/is/58015/ (cit. on pp. 10–14).

Atorf, Daniel, Ehm Kannegieser, and Wolfgang Roller (2019). "Balancing Realism and Engagement for a Serious Game in the Domain of Remote Sensing". In: *Games and Learning Alliance*. Ed. by Allegra M Sobke H. Gentile M. Vol. 11385. Cham: Springer International Publishing, pp. 146–156. ISBN: 978-3-030-11548-7. DOI: 10.1007/978-3-030-11548-7_14 (cit. on p. 9).

Azadegan, Aida, Johann CKH Riedel, and Jannicke Baalsrud Hauge (2012). "Serious games adoption in corporate training". In: *International conference on serious games development and applications*. Springer, pp. 74–85 (cit. on p. 2).

Azadegan, Aida and Johann ckh Riedel (2012). "Serious games integration in companies: a research and application framework". In: *2012 IEEE 12th International Conference on Advanced Learning Technologies*. IEEE, pp. 485–487 (cit. on p. 2).

Bajak, Aleszu (2014). “Lectures aren’t just boring, they’re ineffective, too, study finds”. In: *Science* 12 (cit. on p. 7).

Becker, Katrin and L. Gopin (Jan. 2016). “Selection Criteria for Using Commercial Off the Shelf Games (COTS) for Learning”. In: (cit. on p. 1).

Bellotti, Francesco et al. (2009). “Adaptive experience engine for serious games”. In: *IEEE Transactions on Computational Intelligence and AI in Games* 1.4, pp. 264–280 (cit. on pp. 26, 27).

Berg, Alan et al. (2016). “Dutch cooking with xAPI recipes: The good, the bad, and the consistent”. In: *2016 IEEE 16th International Conference on Advanced Learning Technologies (ICALT)*. IEEE, pp. 234–236 (cit. on p. 44).

Bohl, Oliver et al. (2002). “The sharable content object reference model (SCORM)-a critical review”. In: *International Conference on Computers in Education, 2002. Proceedings.* IEEE, pp. 950–951 (cit. on p. 15).

Brockmyer, Jeanne H et al. (2009). “The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing”. In: *Journal of Experimental Social Psychology* 45.4, pp. 624–634 (cit. on p. 3).

Bunian, Sara et al. (2017). “Modeling individual differences in game behavior using HMM”. In: *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference* (cit. on p. 27).

Caserman, Polona et al. (2018). “Recognition of Full-Body Movements in VR-Based Exergames Using Hidden Markov Models”. In: *Joint International Conference on Serious Games*. Springer, pp. 191–203 (cit. on p. 27).

Clark, C Abt (1970). “Serious games”. In: *Nueva York: Viking* (cit. on p. 1).

ClipArtsFree (2020). *ClipArtsFree*. URL: `https://www.clipartsfree.de/` (visited on 08/30/2020) (cit. on pp. 18, 19, 21, 23, 29, 45).

Connolly, Thomas M et al. (2012). “A systematic literature review of empirical evidence on computer games and serious games”. In: *Computers & education* 59.2, pp. 661–686 (cit. on p. 2).

Derbali, Lotfi, Ramla Ghali, and Claude Frasson (2013). “Assessing motivational strategies in serious games using hidden Markov models”. In: *The Twenty-Sixth International FLAIRS Conference* (cit. on p. 27).

Elaachak, Lotfi, Amine Belahbibe, and Mohammed Bouhorma (2015). “Towards a system of guidance, assistance and learning analytics based on multi agent system applied on serious games”. In: *International Journal of Electrical and Computer Engineering* 5.2, p. 344 (cit. on pp. 3, 28).

eUCM (2020). *eUCM Development Community*. URL: `https://www.e-ucm.es/` (visited on 08/31/2020) (cit. on p. 25).

Falmagne, J-Cl and J-P Doignon (1988). “A Markovian procedure for assessing the state of a system”. In: *Journal of Mathematical Psychology* 32.3, pp. 232–258 (cit. on p. 26).

Fu, Jianbin, Diego Zapata, and Elia Mavronikolas (2014). "Statistical methods for assessments in simulations and serious games". In: *ETS Research Report Series* 2014.2, pp. 1–17 (cit. on p. 26).

Gagniuc, Paul A (2017). *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons (cit. on pp. 17, 18).

game.de (2020). *game – Verband der deutschen Games-Branche e.V.* URL: https://www.game.de/ (cit. on p. 2).

Ghahramani, Zoubin (2001). "An introduction to hidden Markov models and Bayesian networks". In: *Hidden Markov models: applications in computer vision*. World Scientific, pp. 9–41 (cit. on pp. 20, 54).

Göbel, Stefan and Viktor Wendel (2016). "Personalization and adaptation". In: *Serious Games*. Springer, pp. 161–210 (cit. on p. 27).

Gong, Wenbing et al. (2017). "Using a hidden Markov model for improving the spatial-temporal consistency of time series land cover classification". In: *ISPRS International Journal of Geo-Information* 6.10, p. 292 (cit. on p. 36).

Hamdorf, Thorsten (2019). *Deutscher Markt für Serious Games soll durchschnittlich um 19 Prozent pro Jahr wachsen*. URL: https://www.game.de/marktdaten/deutscher-markt-fuer-serious-games-soll-durchschnittlich-um-19-prozent-pro-jahr-wachsen/ (cit. on p. 2).

Hassan, Mohammad Mehedi et al. (2012). "A cloud-based serious games framework for obesity". In: *Proceedings of the 1st ACM multimedia international workshop on Cloud-based multimedia applications and services for e-health*, pp. 15–20 (cit. on p. 31).

Hussaan, Aarij Mahmood, Karim Sehaba, and Alain Mille (2011). "Tailoring serious games with adaptive pedagogical scenarios: A serious game for persons with cognitive disabilities". In: *2011 IEEE 11th International Conference on Advanced Learning Technologies*. IEEE, pp. 486–490 (cit. on p. 28).

Jeong, Hogyeong et al. (2010). "Analysis of productive learning behaviors in a structured inquiry cycle using hidden Markov models". In: *Educational Data Mining 2010*. Citeseer (cit. on p. 27).

Kevan, Jonathan M and Paul R Ryan (2016). "Experience API: Flexible, decentralized and activity-centric data collection". In: *Technology, knowledge and learning* 21.1, pp. 143–149 (cit. on p. 25).

Konert, Johannes, Stefan Göbel, and Ralf Steinmetz (2014). "Video game personalization via social media participation". In: *International Conference on Serious Games*. Springer, pp. 35–46 (cit. on p. 27).

Kostoulas, Theodoros et al. (2012). "Affective speech interface in serious games for supporting therapy of mental disorders". In: *Expert Systems with Applications* 39.12, pp. 11072–11079 (cit. on p. 28).

Learning-Locker (2020). *Learning Locker - Learning Record Store (LRS) implementing the xAPI (Tin Can API)*. URL: https://docs.learninglocker.net/overview-architecture/ (cit. on p. 17).

Lim, Kin Chew (2015). "Case studies of xAPI applications to e-Learning". In: *The Twelfth International Conference on eLearning for Knowledge-Based Society*, pp. 3–1 (cit. on pp. 5, 15).

Liu, Yun-zhong, Ya-ping Lin, and Zhi-ping Chen (2010). "Text information extraction based on hidden Markov model [J]". In: *Acta Simulata Systematica Sinica* 3, p. 038 (cit. on p. 26).

Mostefai, Belkacem, Amar Balla, and Philippe Trigano (2019). "A generic and efficient emotion-driven approach toward personalized assessment and adaptation in serious games". In: *Cognitive Systems Research* 56, pp. 82–106 (cit. on p. 26).

Moussa, Maher Ben and Nadia Magnenat-Thalmann (2009). "Applying affect recognition in serious games: The playmancer project". In: *International Workshop on Motion in Games*. Springer, pp. 53–62 (cit. on p. 26).

Nguyen, Andy, Lesley A Gardner, and Don Sheridan (2018). "A framework for applying learning analytics in serious games for people with intellectual disabilities". In: *British Journal of Educational Technology* 49.4, pp. 673–689 (cit. on p. 28).

ODUSD (1999). *Office of the Deputy Under Secretary of Defense for Readiness (ODUSD(R)), Readiness and Training (April 30, 1999)*. URL: https://prhome.defense.gov/portals/52/documents/rfm/readiness/docs/adl_stratplan.pdf (cit. on p. 25).

Omelina, Lubos et al. (2012). "Serious games for physical rehabilitation: designing highly configurable and adaptable games". In: *Proceedings of the 9th International Conference on Disability, Virtual Reality & Associated Technologies*, pp. 195–201 (cit. on p. 2).

Papanastasiou, George, Athanasios Drigas, and Charalabos Skianis (2017). "Serious games in preschool and primary education: benefits and impacts on curriculum course syllabus". In: *International Journal of Emerging Technologies in Learning (iJET)* 12.01, pp. 44–56 (cit. on p. 1).

Perez-Colado, Victor M et al. (2018). "Learning analytics for location-based serious games". In: *2018 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, pp. 1192–1200 (cit. on p. 25).

Poltrack, Jonathan et al. (2012). "The next generation of scorm: Innovation for the global force". In: *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*. Vol. 2012. 1. National Training System Association Orlando (cit. on p. 15).

Price, Thomas W, Yihuan Dong, and Tiffany Barnes (2016). "Generating Data-Driven Hints for Open-Ended Programming." In: *International Educational Data Mining Society* (cit. on p. 37).

Rabiner, Lawrence and B Juang (1986). “An introduction to hidden Markov models”. In: *IEEE ASSP magazine* 3.1, pp. 4–16 (cit. on pp. 19, 22, 52, 53).

Rawitsch, Don (2017). “Classic Game Postmortem:“Oregon Trail.”” In: *Game Developers Conference-GDC 2017* (cit. on p. 8).

Reisswitz, B. Von (1824). *Kriegsspiel* (cit. on p. 1).

RetroGames (2020). *RetroGames*. URL: https://www.retrogames.cz/play_687-DOS.php?language=EN (visited on 07/15/2020) (cit. on pp. 8, 9).

Schäfer, Andreas et al. (2013). “From boring to scoring–a collaborative serious game for learning and practicing mathematical logic for computer science education”. In: *Computer Science Education* 23.2, pp. 87–111 (cit. on pp. 7, 8).

SERC (May 2018). *What Makes a Good Game?* URL: https://serc.carleton.edu/introgeo/games/goodgame.html (cit. on p. 2).

Serrano-Laguna, Ángel et al. (2017). “Applying standards to systematize learning analytics in serious games”. In: *Computer Standards & Interfaces* 50, pp. 116–123 (cit. on pp. 25, 44).

Simões, Jorge, Rebeca DíAz Redondo, and Ana FernáNdez Vilas (2013). “A social gamification framework for a K-6 learning platform”. In: *Computers in Human Behavior* 29.2, pp. 345–353 (cit. on pp. 1, 4).

Slimani, Abdelali et al. (2018). “Learning analytics through serious games: Data mining algorithms for performance measurement and improvement purposes”. In: *International Journal of Emerging Technologies in Learning (iJET)* 13.1, pp. 46–64 (cit. on pp. 26, 28).

Software, Rustici (2020). *xAPI Adopters*. URL: https://xapi.com/adopters/ (cit. on p. 25).

Streicher, Alexander (2018). *E-Learning A.I. (ELAI)*. URL: https://www.iosb.fraunhofer.de/servlet/is/77629/ (cit. on p. 47).

Streicher, Alexander, Lukas Bach, and Wolfgang Roller (2019). “Usage Simulation and Testing with xAPI for Adaptive E-Learning”. In: *European Conference on Technology Enhanced Learning*. Springer, pp. 692–695 (cit. on pp. 4, 25).

Streicher, Alexander and Wolfgang Roller (2015). “Towards an interoperable adaptive tutoring agent for simulations and serious games”. In: *International Conference on Theory and Practice in Modern Computing, MCCSIS*, pp. 194–197 (cit. on p. 25).

Streicher, Alexander and Jan D Smeddinck (2016). “Personalized and adaptive serious games”. In: *Entertainment Computing and Serious Games*. Springer, pp. 332–377 (cit. on pp. 4, 16, 24–27).

Susi, Tarja, Mikael Johannesson, and Per Backlund (2007). *Serious games: An overview* (cit. on p. 2).

Wernbacher, Thomas et al. (2012). “Learning by playing: can serious games be fun”. In: *Proceedings of 6th European Conference on Games Based Learning*, pp. 534–541 (cit. on p. 8).

Wintjes, Jorit (2015). “Europe’s Earliest Kriegsspiel? Book Seven of Reinhard Graf zu Solms’ Kriegsregierung and the ‘Prehistory’of Professional War Gaming”. In: *British Journal for Military History* 2.1 (cit. on p. 3).

Xuan, Guorong, Wei Zhang, and Peiqi Chai (2001). “EM algorithms of Gaussian mixture model and hidden Markov model”. In: *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*. Vol. 1. IEEE, pp. 145–148 (cit. on p. 36).

Yadav, Vivek, Alexander Streicher, and Prabhune Prabhune (2020). “User Assistance for Serious Game using Hidden Markov Model”. In: *European Conference on Technology Enhanced Learning*. Springer (cit. on p. 43).

Yates, Frances Amelia (1992). *The art of memory*. Vol. 64. Random House (cit. on p. 8).

List of Figures

List of Tables

Dissemination

Internationally indexed journals (*Web of Science, SCI, Scopus, etc.*)[1]

1. User Assistance for Serious Games Using Hidden Markov Model
   Vivek Yadav, Alexander Streicher, Ajinkya Prabhune
   15th European Conference on Technology Enhanced Learning, EC-TEL 2020, Heidelberg, Germany, September 14–18, 2020, Proceedings

[1] Articles already published, in press, or formally accepted for publication.