UNIVERSITY OF MACEDONIA
DEPARTMENT OF APPLIED INFORMATICS
MSC IN ARTIFICIAL INTELLIGENCE AND DATA ANALYTICS

USING SERIOUS GAMES AND LEARNING ANALYTICS FOR STUDENT PROFILING

A dissertation
by

Athanasios Tsionas

Thessaloniki, September 2023

USING SERIOUS GAMES AND LEARNING ANALYTICS FOR STUDENT PROFILING

Athanasios Tsionas

Bachelor of informatics, AUTH, 2020

Dissertation

Submitted in partial fulfilment of the requirements for
THE MSC IN ARTIFICIAL INTELLIGENCE AND DATA ANALYTICS

Supervisor Professor
Maya Satratzemi

Approved by the three-member Examination Committee on 18/9/2023

Satratzemi Maya	Sakellariou Ilias	Evangelidis Georgios
Professor	Assistant Professor	Professor

Abstract

In recent years with the help of digital games there is an increasing interest in creating Serious Games for learning through play. With the help of machine learning algorithms, an educational serious game can be used, not only to assist the learner in his/her studies, but also help the teacher discover more about the students. In game-based learning we take into account that the student behaves differently according to his/her individual characteristics while learning by playing. The most used method to model a person's personality is using self-report questionnaires. The drawback of this approach is that people may not assess themself correctly or their answers may be biased towards the more socially acceptable responses rather than being truthful. In this paper, we explore the idea of creating an educational serious game with the goals of helping the students to train in an introductory programming lesson and at the same time by capturing the students' in-game actions-data with the utilization of machine learning techniques to predict their personality. A story-based game with gamified educational elements was created to help students to assess their knowledge in the programming language C. The students learn by evaluating code snippets and depending on their response the game would give constructive feedback. After the game's end it is possible to model each student's personality model. Particularly, for modeling the learner's personality we used the Five-Factor Model (OCEAN), a taxonomy of five personality traits (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism), each of which combines many personality characteristics. To evaluate the efficiency of the proposed serious game, we gathered data from 107 first year Computer Science students from the University of Macedonia. The students played the game and filled in the Big Five Inventory (BFI) questionnaire to capture their OCEAN traits. The BFI questionnaire was used as a ground truth regarding the personality of each student. After the data gathering, we used machine learning techniques and also classification algorithms to create our model. We used multiple metrics to assess the prediction of the created models. The results showed that it is effective to model both the extraversion and openness personality dimensions using serious games instead of questionnaires.

Keywords: Personality, Game-based learning, Gaming behaviors, Learning analytics, Data analysis

Περίληψη

Στα σύγχρονα χρόνια με την βοήθεια ηλεκτρονικών παιχνιδιών υπάρχει αύξηση ενδιαφέροντος στην δημιουργία σοβαρών παιχνιδιών για εκπαιδευτικούς σκοπούς. Με την βοήθεια αλγορίθμων μηχανικής μάθησης ένα σοβαρό παιχνίδι εκπαιδευτικού σκοπού μπορεί όχι μόνο να βοηθήσει τους μαθητές στις σπουδές τους αλλά να βοηθήσει και τον καθηγητή στο να ανακαλύψει περισσότερες πληροφορίες για τους μαθητές του. Στη μάθηση με βάση το παιχνίδι παίρνουμε υπόψη μας ότι οι μαθητές συμπεριφέρονται διαφορετικά μέσα στο παιχνίδι με βάση τον προσωπικό χαρακτήρα που έχει κάθε μαθητής. Η πιο διαδεδομένη μέθοδος για να μοντελοποιήσουμε την προσωπικότητα κάποιου είναι οι αυτοαναφορικές δηλώσεις σε μορφή ερωτηματολογίου. Το μειονέκτημα αυτής της μεθόδου είναι ότι οι απαντήσεις που δίνουν μπορεί να είναι με βάση ότι θεωρούν πιο κοινωνικά αποδεκτό και όχι το τι ισχύει για τους ίδιους/ες. Σε αυτή την εργασία ερευνούμε την ιδέα να δημιουργήσουμε ένα σοβαρό παιχνίδι εκπαιδευτικού σκοπού με στόχο να υποστηρίξει τους φοιτητές στο να εξασκηθούν σε ένα εισαγωγικό μάθημα προγραμματισμού και ταυτόχρονα να καταγράφει ενέργειες-δεδομένα που θα μπορέσουν στη συνέχεια να αξιοποιηθούν και με τη χρήση αλγορίθμων μηχανικής μάθησης να προβλέψουν την προσωπικότητα των φοιτητών αυτών. Δημιουργήθηκε ένα παιχνίδι (story based) με παιγνιοποιημένα μαθησιακά στοιχεία για να βοηθήσει τους φοιτητές να μελετήσουν και να αξιολογήσουν τις γνώσεις τους στην γλώσσα προγραμματισμού C. Το παιχνίδι βοηθάει φοιτητές να μάθουν ζητώντας τους να αξιολογήσουν τμήματα κώδικα και αναλόγως με τις απαντήσεις το παιχνίδι τους δίνει εποικοδομητική ανατροφοδότηση. Μετέπειτα μέσο της συμπεριφοράς των μαθητών στο παιχνίδι γίνεται να μοντελοποιηθεί η προσωπικότητα του κάθε μαθητή. Συγκεκριμένα για την μοντελοποίηση των προσωπικοτήτων χρησιμοποιήσαμε το μοντέλο των 5 παραγόντων, γνωστό και ως ‹μεγάλη πεντάδα› (Big-five ή OCEAN), είναι μια ταξινομία από πέντε χαρακτηριστικά προσωπικοτήτων (Διαθεσιμότητα σε εμπειρίες, Ευσυνειδησία, Εξωστρέφεια, Προσήνεια και Νευρωτισμός), όπου ο συνδυασμός αυτών μοντελοποιούν πολλές προσωπικότητες. Για να ελέγξουμε την αποτελεσματικότητα του προτεινόμενου σοβαρού παιχνιδιού διενεργήσαμε μια μελέτη σε 107 πρωτοετείς φοιτητές του τμήματος Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας. Οι φοιτητές έπαιξαν το παιχνίδι και συμπλήρωσαν ένα ευρέως διαδεδομένο ερωτηματολόγιο βασισμένο στην θεωρία των πέντε μεγάλων παραγόντων (Big Five

Inventory - BFI). Οι απαντήσεις από το ερωτηματολόγιο θεωρήθηκαν ως η βασική αλήθεια σε σχέση με την προσωπικότητα του κάθε μαθητή. Μετά την συλλογή δεδομένων χρησιμοποιήσαμε τεχνικές και αλγόριθμους μηχανικής μάθησης για να δημιουργήσαμε τα δικά μας μοντέλα πρόβλεψης. Χρησιμοποιήθηκαν πολλαπλές μετρικές για να αξιολογήσουμε την αποτελεσματικότητα των μοντέλων μας. Τα αποτελέσματα ήταν ενθαρρυντικά καθώς έδειξαν ότι είναι εφικτό να μοντελοποιήσουμε τα χαρακτηριστικά της διαθεσιμότητας σε εμπειρίες και της εξωστρέφειας χρησιμοποιώντας σοβαρά παιχνίδια αντί για ερωτηματολόγια.

Λέξεις κλειδιά: Προσωπικότητα, Μάθηση με βάση το παιχνίδι, συμπεριφορές στο παιχνίδι, αναλυτικά στοιχεία μάθησης, ανάλυση δεδομένων

Acknowledgements

I would like to express my gratitude to Professor Maya Satratzemi, my thesis supervisor for giving me the opportunity to work on such a project. I am also extremely grateful for her patience, valuable feedback and support that she gave me over the course of the thesis. This support helped me to finish my thesis but also pushed me to become a better scientist.

I would like to express my sincere thanks to the professors Evagelidis Georgios and Sakellariou Ilias, my defense committee for generously providing their expertise, knowledge and their insights regarding my thesis.

Finally I would like to thank the PhD Candidate Aristoteli Lazaridi for giving me some advices and suggestions in the course of my thesis.

Table of contents

1 Introduction 1

1.1 Problem – Importance of the topic 1

1.2 Aim – Objectives 2

1.3 Contribution 2

1.4 Structure of the study 2

2 Background – Related work 3

2.1 Digital Games 3

2.1.1 History of digital games 3

2.1.2 Game Genres 4

2.1.3 Game Engines 6

2.1.4 Personality and game's avatar 8

2.2 Personality Modeling 9

2.2.1 Five Factor Model 10

Machine Learning 14

2.2.2 Supervised Learning 14

2.2.3 Unsupervised Learning 19

2.2.4 Reinforcement learning 20

2.3 Related work 21

3 Methodology 23

3.1 Design the game 23

3.1.1 Flow of the game 23

3.1.2 Story of the game 24

3.1.3 Design Choices 26

3.2 Experiment 36

3.2.1 Participants 36

3.2.2 Procedure 37

3.3 Data Analysis 38

3.3.1 Exploring the data 38

3.3.2 Data from the game 38

3.4 Results 43

4 Conclusion 46

4.1 Summary and conclusions 46
4.2 Research limitations 47
4.3 Future extensions 47
Annex A – BFI Questionnaire 53
Annex B - Code questions 55
Annex C - Conference Contribution Paper 69

List of Figures

Figure 2-1: Unity Logo (Unity Technologies, 2023) 7
Figure 2-2: Unity Editor (Unity Technologies, 2023) 7
Figure 2-3: Unreal Logo (Epic Games, 2023) 8
Figure 2-4: Unreal Engine Blueprints (Epic Games, 2023) 8
Figure 2-5: OCEAN Model 10
Figure 2-6: KNN classification of new data point 16
Figure 2-7: Decision Tree 18
Figure 3-1: Player's point of view 27
Figure 3-2: Wrong and correct assessment of the code snippet 27
Figure 3-3: Day 1 Openness Dialogue 30
Figure 3-4: Day 2 Extrovert Dialogue 31
Figure 3-5: Day 3 Extrovert Dialogue 32
Figure 3-6: Day 4 Extrovert & Openness Dialogue 33
Figure 3-7: Day 6 Openness Dialogue 34
Figure 3-8: Day 7 Extrovert Dialogue 35
Figure 3-9: Question ID (Key) - Student response index (value) 39
Figure 3-10: Different conversation paths 40
Figure 3-11: Day 8 selection reason each trait 41
Figure 3-12: List of all IDs and list of used subset 42
Figure 3-13: Naive bayes python code 43
Figure 3-14: Question ID used for each trait 44

List of tables

Table 3-1: Day Event Short 24
Table 3-2: Day design based on their target trait 28
Table 3-3: Day 1 Extrovert Trait 29
Table 3-4: Day 1 Openness Trait 30
Table 3-5: Day 2 Extrovert Trait 30
Table 3-6: Day 2 Openness Trait 31
Table 3-7: Day 3 Extrovert Trait 31
Table 3-8: Day 3 Openness Trait 32
Table 3-9: Day 4 Extrovert Trait 32
Table 3-10: Day 4 Openness Trait 32
Table 3-11: Day 5 Extrovert Trait 33
Table 3-12: Day 5 Openness Trait 33
Table 3-13: Day 6 Extrovert Trait 34
Table 3-14: Day 6 Openness Trait 34
Table 3-15: Day 7 Extrovert Trait 34
Table 3-16: Day 7 Openness Trait 35
Table 3-17: Participant personality distribution 38
Table 3-18: Extrovert models metrics 43
Table 3-19: Openness models metrics 44
Table 3-20: Decision Trees extrovert confusion matrix 45
Table 3-21: Naive Bayes Openness confusion matrix 45

1 Introduction

1.1 Problem – Importance of the topic

The utilization of Serious Games (SGs) as an educational tool is a recent phenomenon, and its full potential is yet to be fully comprehended. In recent years there is an increasing focus on adding SGs to the education field. A lot of studies have compared the effectiveness of traditional teaching methods in comparison to an educational SG (Prensky, 2006). Results have demonstrated that educational SGs have a lot of advantages compared to traditional ones. The traditional environment has become boring and unmotivated. Especially to a generation born with so much technology around (Barron, 2015).

In comparison educational SGs can be more appealing and encourage the learner to try to apply his knowledge. This gives the possibility for the learner to apply or simulate his knowledge, which can be hard to achieve in a classical environment. In a well-designed educational game environment, the learner can have fun while learning, making the process more enjoyable and less stressful (Schunk, 2008).

Although learning through gaming may be appealing, not all learners have the same likings. Several studies show that different personalities tend to prefer different learning styles, game genres and playing strategies (Mariela Pavalache-Ilie, Sorin Cocorada, 2014). The requirements to know each learner's personality have increased in order to better utilize the best-suited teaching model for them. The most used method for modeling the learner’s personality is through self-report questionnaires. Using questionnaires has some limitations. The main limitation is that the learner may not answer the questions truthfully. The learner may choose the more socially acceptable answer or hide information if he believes he won't benefit from responding. Another important concern about self-reports is that the learner may not have the necessary self-awareness needed for the questions. Although a lot of these concerns and weaknesses can be reduced by using various techniques, it is clear that using multiple methods to capture the learner’s personality is beneficial (Zhi Chen,Tao Lin, 2017) (Judd Charles, Mcclelland Gary, 1998).

1.2 Aim – Objectives

This study focuses on creating an educational SG with the ability to model the player's personality type. This can be achieved by capturing the players' in-game behavior. The design of the SG is focused on capturing the Extraversion and Openness personality traits as they are the most impactful dimensions towards a learning system. Additionally the game aims to assist the player in learning the programming language C. The findings in this study can help the advance of using digital games with a serious intention. More specifically to have an alternative method of learning a person's personality model as also to better create interactive and engaging methods to assist learners in their studies.

1.3 Contribution

The idea of using serious games for capturing the player's personality isn't new. Similar studies have been conducted for capturing the player's personality through their in-game behavior. Although the idea isn't new there isn't yet a systematic procedure to accurately achieve the desired result. This study aims to contribute towards this goal. A similar study by Denden et al. (2017) had created a gamified framework and contacted an experiment to measure the framework's accuracy. In comparison this study has a higher number of participants. Additionally in this study multiple machine learning (ML) algorithms were tested for creating an accurate prediction model instead of one.

1.4 Structure of the study

The structure of this thesis is the following:

- Chapter 2 presents what are the game engines and game genres. It also presents the big five personality model. It contains a theoretical background about the different machine learning teaching methods and presents the algorithms of Naïve Bayes, Decision trees and K-nearest neighbors (KNN). Finally, some related studies and their findings are presented.
- Chapter 3 presents the created game for this study. It also presents the experiment we contacted the details about how we analyze the gathered data and the final results.
- Chapter 4 contains a summary and overall conclusion of the study and also mentions potential future directions.

2 Background – Related work

This chapter presents the background necessary for better understanding this study. More specific there is the definition about what is a digital game as also a short history overview about when they started. In the digital game sub chapter it is also presented what are game genres and game sub-genres with deepening in the genres used in this study. Additionally in this sub chapter it is presented the four different relationships that a player can have with his/her in game avatar. In the personality modeling sub chapter it is presented the big five personality model with showcasing the five traits of the model. The machine learning (ML) sub chapter contains a theoretical background about each machine learning teaching school (Supervised Learning, Unsupervised Learning and Reinforced Learning) including common strengths and weaknesses of each one of them. It is also presented how they function as also their common strengths and weaknesses of the supervised learning algorithms of Naïve Bayes, Decision Tress and K-nearest neighbors (KNN). In the related work sub chapter there are presented three similar studies with this one. Each related study propose a game or a gamified system with the purpose of modeling the player's personality based on the Big Five Personality model (OCEAN).

2.1 Digital Games

A digital game is an electronic game that involves interaction with a user. The user can input his actions through an input device such as a keyboard, a mouse, a joystick etc. Digital games include a set of rules and they simulate an environment or a situation with the help of electronic devices (Carenys and Moya,2016).

2.1.1 History of digital games

The first digital games were developed in the 1950s and 1960. Of course, at the time the games were simpler in design and graphics (Ivory, 2015). The first commercial digital game was developed in 1971 "Computer Space" and showed that digital games could reach paying audiences. Until then digital games were mostly used in labs. One of the first successful known digital games is "Pong" released in 1972. From then the developing tools for creating a digital game have vastly improved (MIT Press, 2021). Although the tools to create a game nowadays are more accessible than ever the player's

expectation as also the skill needed to develop a game are high. The more complicated a game is, the more experts need to work on it. The development team needed to create a video game can vary from 1 person for a simple digital game (Will You Snail?, 2022) up to a team of 400 people for a very complex one (Hilliard, 2022).

2.1.2 Game Genres

As of today they have been developed more than 1.000.000 digital games (Scutaru, 2023). These games have different play-styles, set of rules and goals. For this reason the need for a game categorization was created. For the categories of the games, the game genres were created. A game genre shows the game's main playstyle and rules. For example if a game is characterized as a first person shooter (FPS) then the players will expect a game with fast-paced action, playing the game's character in first person (camera and movement) and of course the use of a shooting mechanic in the game. As of today there are more than 30 game genres (Clarke, Lee and Clark, 2017), (Lee et al, 2014), (Crathorne, 2010).

The use of the game genres may help the players understand the main elements of the game but because of the complexity of today's games further categorization was needed. For this reason the game sub-genres were created. Sub-genres are giving information for the secondary elements that are included in the game. For example an FPS game can be characterized as multiplayer if it is played online, tactical if a lot of planning and strategies are involved in the combat and so on. Usually a digital game has a lot of sub-genres. Game sub-genres emphasize a game's mechanic or playstyle (Clearwater, 2011). For this reason their number is always changing because the development of digital games is still evolving and is not uncommon for a new mechanic to be created, meaning a new game sub-genre. A good example for this phenomenon is the Soulslike sub-genre. This sub-genre was created after the release of the successful game Dark Souls and is used from games that were inspired from it (Detering et al, 2022).

For the purpose of this study a digital game "code please" was designed. The domineering genres of this game are: First person, story driven, educational, serious game. In the next paragraphs these genres are shortly presented.

2.1.2.1 First Person

Some digital games are represented by the player's view regarding his/her game avatar. Especially in a game with 3D graphics the player can observe his her avatar and the game's world with two main approaches. As a third person and as a first person. As a third person the player sees the game's avatar with some distance typically from an over-the-shoulder or behind-the-back perspective.

In the first person the player observes his/hers avatar and the game's world in the perspective of the avatar. A first-person game lets the player see exactly what his/her character sees and usually are games telling a story with the element of "I". In most first-person games, the character's hands are shown.

2.1.2.2 Story Driven Game

A story driven game as the name implies is a storytelling game. A lot of games have a story (eg. a fighting game can have a story) but in a story driven game the story is the main aspect of the game. The game is progressing through the story's progression. The player can either observe the story as it unfolds or he/she can affect the story's outcome.

2.1.2.3 Serious Game

A serious game is a game designed for a primary purpose other than pure entertainment. The "serious" adjective in the genre refers to the primary purpose of the game. A lot of industries use serious games to achieve their own purposes. Some of these industries are defense, education, scientific exploration, health care, emergency management, city planning, engineering, politics and art (Ypsilanti et al, 2014).

2.1.2.4 Educational Game

An educational game is a game subgenre under the serious games. An educational game is a digital game with a primary purpose not to entertain the player but to teach or train him/her. An educational game's main purpose can be either drilling a subject matter to the player in a linear manner, while wrapping the game with entertainment aspects or to encourage creative thinking and problem solving, often presented in non-linear experiences (Cole, Parada and Mackenzie, 2023).

Digital games were preferred in education for multiple reasons. First they allow the player to fail without having any "real" penalty. That way the player could try new

things and test theories without the fear of failing. Second, a lot of students have a “sweet spot” for games. Those students might find it more appealing to learn a concept through a digital game instead of a book. Another advantage of an educational digital game is the opportunity for the learner to practice what he knows. A good example of this is a realistic flight simulator game where the trained pilots can safely and accurately practice their knowledge about flying an airplane.

Finally, there are some games that were designed without an educational purpose but found some educational use. Some examples of these kinds of games are some strategy war games that included some historical references like “Total war” and “Age of Empires” (De Aguilera and Mendiz, 2003) (Ritterfeld and Weber, 2006) (Squire, 2003).

2.1.3 Game Engines

Game engines are software development environments for the purpose of creating a digital game. Game engines optimize and simplify the procedure of developing a digital game in a variety of programming languages. A game engine usually includes a 2D or a 3D graphic renderer, functions that simulate real-world physics (also known as physics engine), some artificial intelligence libraries and many more features (Andrade, 2015).

At the moment game engines are easily accessible giving the opportunity to many game developers to create their own digital game. There are a lot of known game engines that are used for creating digital games. The two most popular game engines are Unity and Unreal. In the next paragraphs those engines are shortly presented (Toftedahl and Engström, 2019).

2.1.3.1 Unity

Unity is a 2D and 3D cross-platform game engine developed by Unity Technologies (Figure 2-1). Unity was created in 2005 and since then the game engine has been gradually extended to support a variety of desktop, mobile, console and virtual reality platforms. For the creation of a digital game in Unity, game developers use Unity’s drag and drop functionality as also a scripting API in the programming language C# or Javascript (Unity Technologies, 2023).

Figure 2-1: Unity Logo (Unity Technologies, 2023)

Unity is favorite as a game engine from many developers for multiple reasons. First it is considered to be beginner friendly for it's easy to understand architecture and simple UI (Figure 2-2). Additionally Unity has a huge community making it easy to find Unity specific assets, tutorials or share a problem that someone else might have tackled in the past. Unity's charging plan is also something that a lot of game developers like. If a Unity game has less earnings than $100K per year then Unity is free of charge.

A lot of famous and refined games have been produced by the Unity game engine. Some of the most famous ones are: Hearthstone, Pokemon Go, Among Us, Cuphead (Haas, 2014) (Šmíd, 2017).

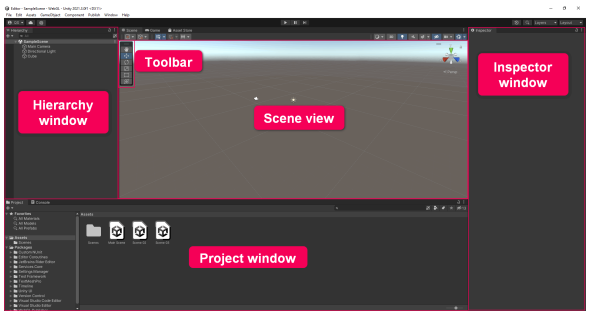

Figure 2-2: Unity Editor (Unity Technologies, 2023)

2.1.3.2 Unreal

Unreal Engine (UE) is a cross-platform game engine mostly designed for producing 3D games. UE was developed by Epic Games. The first game produced by Unreal Engine contributed to the engine's name. The game was showcased in 1998, it was a first person shooter game and its name was Unreal (Figure 2-3).

Figure 2-3: Unreal Logo (Epic Games, 2023)

Unreal Engine uses C++ as a programming language for developing games. Because the programming language C++ is considered a difficult programming language to learn, Unreal Engine also provides a visual scripting tool for developing games known as blueprints (Figure 2-4) (Epic Games, 2023).

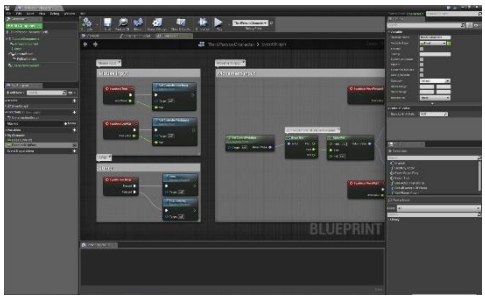

Figure 2-4: Unreal Engine Blueprints (Epic Games, 2023)

Unreal Engine is favorite as a game engine from many developers for multiple reasons. UE is a software with high standards for creating high budget and profile games known also as AAA games. It has a lot of features for supporting high-end graphics and high computational lighting in games. This was amplified on the newest release of the Unreal Engine, the Unreal Engine 5 (UE5) in April 2022. UE5 was especially promoted for its capability of displaying high resolution game graphics (Šmíd, 2017).

A lot of famous and refined games have been produced by the Unreal game engine. Some of the most famous ones are: Fortnite, BioShock, Batman: Arkham City, Deus Ex.

2.1.4 Personality and game's avatar

When a player plays a story based game he is bound to have a "relationship" with the game's character or avatar (Bowman & Schrier, 2018). There are four main relationships between the player and the game's avatar: avatar-as-object, avatar-as-me, avatar-as-symbiote and avatar-as-other. The next four paragraphs describe these four relationships.

Avatar-as-object is the relationship that the player perceives the game's avatar mostly as a tool. Banks and Bowman (2014) found that players with the avatar-as-object orientation were most likely to refer to their avatars as objects, puppets, tools, or toys. Finally these players usually play based if their actions are technically possible within the boundaries of the game and not having in mind that there are consequences to other players or NPCs.

Avatar-as-me is the relationship that the avatar acts as an extension or mirror of the player's personality in the game world. Banks and Bowman (2014) found that players with avatar-as-me orientation mostly refer to their avatars as pieces of themselves or extensions. Additionally such players have a merging experience with their avatar while playing.

Avatar-as-Symbiote is the relationship where the avatar assists the player to create an identity for the game character. The avatar serves the role of a costume allowing the player to add or practice desirable behaviors and characteristics (Banks 2015). In comparison with the relationship avatar-as-me the player sees the avatar as a separate entity and usually the traits that the player adds to the avatar are not his own. These traits can be some idealized or undesired by the player.

Avatar-as-other is the relationship where the player perceives the game avatar as a different entity with its own life. Usually the avatar follows a story based on the game design as also sometimes the players will create new content around the avatar's backstory. Finally Banks and Bowman (2014) found that players with avatar-as-other orientation commonly refer to the character as a person or a partner.

2.2 Personality Modeling

Personality is the enduring configuration of characteristics and behavior that comprises an individual's unique adjustment to life, including major traits, interests, drives, values, self-concept, abilities, and emotional patterns. Personality is generally viewed as a complex, dynamic integration or totality shaped by many forces, including hereditary and constitutional tendencies; physical maturation; early training; identification with significant individuals and groups; culturally conditioned values and roles; and critical experiences and relationships. Various theories explain the structure

and development of personality in different ways, but all agree that personality helps determine behavior (APA, 2023).

2.2.1 Five Factor Model

In order to predict learners' personalities, various personality models have been proposed. One of the most used is the Five-Factor Model (FFM) also known as OCEAN (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism) model (McCrae and John, 1992), (Franić et al, 2014). This model is one that is both compatible with the present view and consistent with everyday observation. The theory behind this model is that each person can be described by using five traits containing two separate, but correlated, aspects reflecting a level of personality below the broad domains. These pairs are Extraversion and introversion, Agreeableness and antagonism, Conscientiousness and lack of direction, Neuroticism and emotional stability and finally Openness and closedness to experience (Figure 2-5).

Figure 2-5: OCEAN Model

Each person has a score in each dimension and through the scores someone can determine the tendencies of that person. To calculate a person's OCEAN score the most common method is through a self-report questionnaire. Each statement in the questionnaire targets one specific pair of traits. The person rates in a Likert scale how much he/she relates with the statement. At the end of the questionnaire after adding the score of each statement the person's personality can be determined. There are some official questionnaires for the FFM. A widely known questionnaire is the Big Five Inventory (BFI) questionnaire (John, Donahue and Kentle, 1991) which has been used widely and is found reliable. It contains 44 questions and each question targets one trait of the OCEAN model. There are many different variations of the BFI questionnaire with different amount of questions for targeting the OCEAN traits. For example there is the

BFI-2 which contains 60 questions or the BFI-2-XS that is an extra small form with 15 questions (Soto and John, 2017).

In the next section each trait of the OCEAN model will be presented. Additionally for each trait they are presented the statements used in BFI to target the presented trait. Statements with an "<R>" in the end of them contribute reverse to that trait's score. It is also important to note that in the BFI questionnaire those statements are mixed and they are not grouped by their targeted traits.

2.2.1.1 Openness

The openness trait focuses on modeling the imagination, insight and creativity of a person. People with high scores on openness tend to be more creative, like abstract thinking and arts, are more open to trying new things and they can easily handle changes. In comparison, people with low scores on openness are considered more logical or down to earth, they don't like the idea of change or trying new things and in general they are considered to have a more traditional approach to life.

Related Question from the BFI: I see Myself as Someone Who...

- Is original, comes up with new ideas
- Is curious about many different things
- Is ingenious, a deep thinker
- Has an active imagination
- Is inventive
- Values artistic, aesthetic experiences
- Prefers work that is routine <R>
- Likes to reflect, play with ideas
- Has few artistic interests <R>
- Is sophisticated in art, music, or literature

2.2.1.2 Conscientiousness

The conscientiousness trait focuses on modeling thoughtfulness, control and goal driven behaviors.

People with high scores on conscientiousness tend to be more focused on finishing important tasks on time, they pay extra attention to detail, like to have a schedule and they spend a lot of time on preparing. In comparison, people with low scores on

conscientiousness don't like the idea of structures and schedule and they tend to procrastinate on important tasks.

Related Question from the BFI: I see Myself as Someone Who…

- Does a thorough job
- Can be somewhat careless <R>
- Is a reliable worker
- Tends to be disorganized <R>
- Tends to be lazy <R>
- Perseveres until the task is finished
- Does things efficiently
- Makes plans and follows through with them
- Is easily distracted <R>

2.2.1.3 Extroversion

The extraversion trait focuses on modeling the willingness of a person having a social interaction with other people. People with high extroversion score tend to be more talkative and they express more their emotions. They usually like to be the center of attention, they enjoy meeting new people and they can easily start conversations. As this trait is useful for social jobs, people with high extraversion are preferred for jobs such as sales, marketing, teaching and politics.

In comparison people with low extraversion score also known as introverts tend to prefer solitude and have less energy in social situations. They don't like small talk and usually they think a lot before speaking.

Related Question from the BFI: I see Myself as Someone Who…

- Is talkative
- Is reserved <R>
- Is full of energy
- Generates a lot of enthusiasm
- Tends to be quiet <R>
- Has an assertive personality
- Is sometimes shy, inhibited
- Is outgoing, sociable

2.2.1.4 Agreeableness

The agreeableness trait focuses on modeling the altruism, kindness and affection of a person.People with a high score on agreeableness tend to have more empathy towards others, they love helping other people and they show interest in others.
In comparison, people with low score on empathy tend to show less interest in other people, they aren't interested in other people's problems and feelings.
Related Question from the BFI:
I see Myself as Someone Who…

- Tends to find fault with others <R>
- Is helpful and unselfish with others
- Starts quarrels with others <R>
- Has a forgiving nature
- Is generally trusting
- Can be cold and aloof <R>
- Is considerate and kind to almost everyone
- Is sometimes rude to others <R>
- Likes to cooperate with others

2.2.1.5 Neuroticism

The neuroticism trait focuses on modeling the sadness and emotional instability of a person. Neuroticism can be described as the person's response to stress and perceived threats in someone's life. People with a high score on neuroticism tend to have mood swings, anxiety and they get upset or dramatic often.
In comparison people with a low score on neuroticism tend to be characterized as emotionally stable, they handle stress well and they rarely feel upset or depressed.
Related Question from the BFI:
I see Myself as Someone Who…

- Is depressed, blue
- Is relaxed, handles stress well <R>
- Can be tense
- Worries a lot
- Is emotionally stable, not easily upset <R>
- Can be moody
- Remains calm in tense situations <R>

- Gets nervous easily

Machine Learning

Machine learning (ML) is a branch of artificial intelligence and computer science. ML is a software application targeted to solve problems for which the development of the solution by human programmers would be cost-prohibitive and time consuming. It is a collection of techniques to help the machine discover its own algorithms for the needed solution. There are three major approaches for teaching the machine how to solve the requested problem. Depending on the problem's nature and limitations a different approach can be more fitted to be used. These three approaches are the "supervised learning", "Unsupervised learning" and the "reinforcement learning".
In the next chapters these approaches are presented. Additionally, there are presented the algorithms used in this study (Rebala et al, 2019).

2.2.2 Supervised Learning

Supervised learning is a machine learning technique that uses a training set to teach the model the desired output. The model measures its accuracy through a loss function and it adjusts itself in each iteration until the error has been sufficiently minimized.

As the model "learns" from the initial training set it is critical the correct implementation of it. There are a lot of factors that must be taken into account when structuring or collecting data for the initial training set. The next paragraphs outline some critical aspects regarding using supervised algorithms.

First it is important the appropriate size for the initial training set. Although the model might get high accuracy using a small training set the final output may not be flexible. For example if we try to train a model to differentiate between images of tulips and roses no matter how high accuracy we reach if the training set has only 10 images then the model won't be able to be used in the real world. The model might be sufficient to categorize images close to the ones in its train set but the small dataset size drastically increases the likelihood of encountering images that are in a completely different angle or lighting having as a result a bad clustering.

Although it is important the big size of the training set it is vital to avoid having redundant dimensions within it. Redundant dimensions can confuse the final model as the model might take into account irrelevant data for its prediction. For example if a

model is trained for the purpose to predict if a person will survive in the Titanic then the person's name is irrelevant information regarding that prediction. In the extreme case that every person named John survived then the model might conclude that if the passenger's name is John then he will survive the Titanic (Liu Bing, 2011).

Another aspect of consideration is the heterogeneity of the training data. In order for the model to have a good distinction between the indicated classes then sufficient training data for each class are needed. A recommended approach is for the initial dataset to have a distribution similar to the real-world distribution as the model usually replicates the training's set distribution to some degree.

2.2.2.1 Naive Bayesian

Naive Bayes is a probabilistic machine learning algorithm based on Bayes Theorem. Its primary use is to solve classification problems. The Bayes Theorem is used to determine the probability of a hypothesis with prior knowledge. The algorithm works under a few assumptions for the data giving the algorithm the title "naive" in its name. The first assumption is that all the features are independent with each other which usually is not the case with real-world data. Additionally, the algorithm assumes that every feature is in the same level of importance meaning that every feature will contribute equally in the model's prediction (Liu Bing, 2011). The formula for Bayes' theorem is shown on (1).

$$P(B) = P(A)P(A)/\,P(B) \quad (1)$$

Where:

- P(A|B) is Posterior probability: Probability of hypothesis A on the ob event B.
- P(B|A) is Likelihood probability: Probability of the evidence given tl probability of a hypothesis is true.
- P(A) is Prior Probability: Probability of hypothesis before observing the evidence.
- P(B) is Marginal Probability: Probability of Evidence.

Naive Bayes has some strengths and weaknesses regarding its application and use. First, the Naive Bayes is straightforward and swift in its implementation, "providing" that way more time to focus on optimizing the algorithm's parameters. It can be used on binary and multi-class classification problems which can be difficult for other classifiers

to manage. Finally, if its assumption about the independence between the features holds true then it requires much less data to produce accurate results in comparison with similar algorithms.

Some of its weaknesses are first that it is subject to zero frequency. Meaning if a feature has zero frequency in the training data then that probability will be also zero. This can be avoided by using a smoothing technique. Finally another weakness of Naive Baes is its unrealistic core assumption. Naive Bayes assumes that all the features are independent with each other which usually isn't the case in real life (Jadhav and Channe, 2016).

2.2.2.2 K-Nearest Neighbors

The k-Nearest Neighbors algorithm, also known as KNN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

For the classification the algorithm uses the distance between data points to determine their class. Data points with small distance between them are considered to belong in the same cluster. KNN predicts the requested point's class in relation with the class of its closest neighbor points (Figure 2-6).

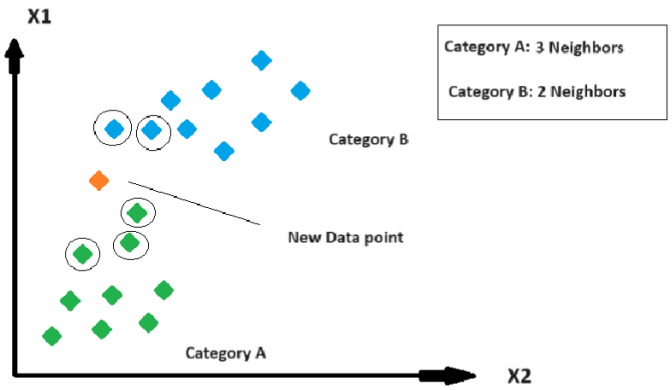

Figure 2-6: KNN classification of new data point

The data point's class is predicted through a "majority vote" of those neighbors. The number of "voting" neighbors as also the distance function used for calculating each data point's distance are vital parameters for this algorithm. For the right number of

neighbors some tries are needed to be done to discover what works best for the used dataset. Although there isn't any specific rule about how many neighbors are best suited there is commonly used an odd number of neighbors to avoid having tie in the "voting" (Liu Bing, 2011). For the distance measurement there are several choices but the most common ones are the Euclidean Distance (2) and the Manhattan Distance (3).

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \quad (2)$$

$$d(x, y) = \left(\sum_{i=1}^{m} \left| x_i - y_i \right| \right) \quad (3)$$

KNN has some strengths and weaknesses regarding its application and use. Some of its strengths are that it is easy to implement. It can also adapt easily when new data samples are added to the training set. Compared to other ML algorithms KNN requires a few hyperparameters to be adjusted. Finally, KNN is more effective if the training set is large.

Despite its strengths KNN also has some weaknesses. The computational cost is high because the algorithm needs to calculate the distance between the data points for all the training samples. In addition to this the algorithm has some issues in scaling. Because it takes more memory and calculating power than other classifiers, the scaling of a problem's dataset can cost time and money (Jadhav and Channe, 2016) (Amra and Maghari, 2017).

2.2.2.3 Decision Trees

Decision tree is a supervised algorithm usually used in data mining. This algorithm can be used for both regression and classification problems. As the name implies the algorithm uses a tree-like model of decisions (Figure 2-7). Decision tree is a popular algorithm in machine learning for its intelligibility, simplicity and easy to comprehend visual representation of the model.

A decision tree is drawn with its root at the top. In each internal node of the tree there is a condition. Usually the condition is represented in the form of a question. Based on the node's condition the tree splits into branches. The end of a branch that doesn't split anymore is a decision leaf. The decision leaves are giving the prediction based on the "answers" the algorithm received in each internal node. In the next paragraphs it is presented the way to construct the tree model (Liu Bing, 2011).

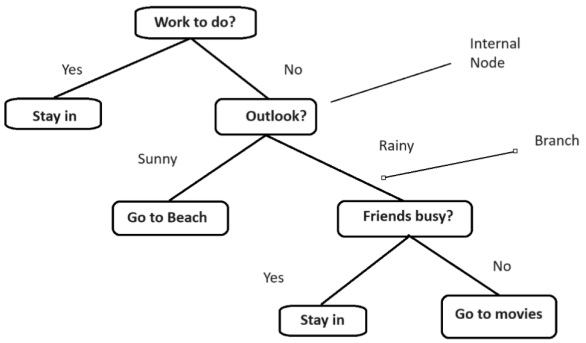

Figure 2-7: Decision Tree

The decision tree algorithm is in the category of greedy algorithms. Each iteration the algorithm tries to identify the best "question" or condition to split the data. There are multiple methods for the algorithm to evaluate the splits in each node and decide what is the best attribute to split the data. The methods are evaluating the quality of each test condition and how well the model can classify the splitted data into a class. The two most popular metrics are the information gain and the Gini impurity (Shannon, 1948).

The information gain measures the reduction of entropy or surprise by splitting the dataset according to a specific variable. The entropy is a metric about the impurity of the selected samples or dataset its values range between 0 and 1. An Entropy of 0 means that all the selected samples belong to the same class and therefore there is 0 surprise in that dataset. On the contrary an entropy of 1 means the selected samples are equally distributed, meaning that we have the maximum surprise. The entropy's formula is the following (4):

$$Entropy(S) = \quad - \sum_{cEC} p(c) log_2 p(c) \tag{4}$$

- S represents the data set that entropy is calculated
- c represents the classes in set, S
- p(c) represents the proportion of data points that belong to class c to the number of total data points in set, S

Information gain calculates the difference in the entropy in two datasets. The formula for the information gain is the following (5):

$$Information\ Gain(S, a) = Entroppy(S) - \sum_{v \in vcalues(a)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{5}$$

- "a" represents a specific attribute or class label

- Entropy(S) is the entropy of dataset, S
- |Sv|/ |S| represents the proportion of the values in Sv to the number of values in dataset, S
- Entropy(Sv) is the entropy of dataset, Sv

Decision trees have some strengths and weaknesses regarding their application and use. Some of the strengths of this algorithm are first how easy it is to understand and implement this algorithm. This advantage is due the ability of the decision trees to handle data of multiple types (either discrete or continuous values) without needing a lot of data pre-processing which isn't usually the case for such algorithms. The nature of decision trees makes it easy to visualise, understand and explain important attributes for making decisions. Decision trees can handle missing values without losing a lot of accuracy from the final model. Finally the algorithm can be applied for both classification and refraction problems.

Despite their strengths, decision trees have some disadvantages. First a small change in the data (eg. adding more observations) can cause a large change in the structure of the decision tree. Decision trees due to their greedy nature they tend to overfit. Additionally, they can be biased towards certain outcomes which can be problematic if the training data are biased. Finally decision trees may not work well with data containing high levels of noise or data with irrelevant attributes (Jadhav and Channe, 2016).

2.2.3 Unsupervised Learning

Unsupervised learning is a type of machine learning that uses artificial intelligence algorithms to identify patterns in the data that are either classified or labeled. In contrast with supervised learning, unsupervised learning models don't need a supervision or an initial dataset to "teach" the model how to label or cluster the data. Unsupervised learning models are used in unstructured data for the purpose of discovering patterns or grouping data without the need of human intervention. This ML teaching model is ideal for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition (Ghahramani, 2003).

The benefits of using unsupervised learning are numerous. It can derive meaning from raw datasets more quickly compared to a human. There is no need to interpret labels for each data point in the dataset as the algorithms don't need them. This can save

a lot of time for the ML and data engineers because usually they have to label each data point by hand. Finally, in a lot of real-world problems we don't always have input data with the corresponding output meaning that unsupervised learning techniques are needed for these kinds of datasets.

Although unsupervised learning algorithms are great for specific problems there come with some challenges. First the results are unpredictable as there are no labels and there is the need for a human to validate the final output. Additionally, identifying hidden patterns in large unclassified datasets can make the whole training process more difficult. Finally unsupervised learning algorithms tend to take longer time to train and produce a final predicting model.

2.2.4 Reinforcement learning

Reinforcement learning (RL) is a type of machine learning where in the software environment agents learn to perform certain actions in a determined environment in order to maximize their rewards in that particular situation. In RL the virtual agents use a trial-and-error approach to learn their given environment and discover through that process what is the optimal way of maximizing their reward. In RL methods there isn't a correct solution in the training data and the model's target is for the agent to learn an optimal or a near-optimal policy that maximizes the reward function. The agents can get a positive reward if they perform a behavior that is desired or they can get a negative reward for the purpose to be discouraged from pursuing an unwanted behavior.

RL models have a lot of advantages regarding their use. First, as the agents learn to create their own policies in the virtual environment, given time they can solve very complex problems that can't be solved with conventional techniques. This can be really useful for passing skills to robots that can be quite tricky on how else to teach those skills especially if we take into account the variety of situations that can change the application of the specific skill.

Although there are a lot of advantages when using RL there are some things to consider regarding the use of RL methods. RL methods usually need a lot of data and a lot of computational power. Additionally, RL methods are highly dependent on the reward function. If a reward function is poorly designed without considering the end result then the agent might not learn the desired behavior. Finally, because there are a lot of environmental variables that can interact in unexpected ways, if the agent develops an

unwanted behavior it can be difficult to diagnose and fix the problem (Garcıa and Fernánde, 2015) (Van, 2011).

2.3 Related work

Few works have been conducted which used digital games to model the learner's personality. In their study Denden et al. (2017) it was examined what systems and game elements are preferred based on the student personality. For the purpose of their study they created a gamified Moodle Learning System (LMS). Their study was focus on the extrovert trait of the Big Five personality traits. The study's aim was to discover what game elements (eg. Avatar, leaderboard, badges) like or dislike each characterization (low, balance, high) of the extroversion personality trait. In the experiment they contacted they successfully gathered 57 students of the University of Tunisia. After the students interacted with the LMS they answered a BFI and a game feature assessment questionnaire. In the study they found that personality plays an important part regarding the preference in game elements. For example students with high extroversion traits were more positive in using some games elements like leaderboard or progress bar than students with low extroversion.

Another work from the same research team was used an inspiration for this study. In their study Denden et al. (2018) they created a Computer Architecture Game framework (CAG). The proposed framework is based on an online 2-D role-playing game with the target of helping the students to learn the Computer Architecture subject. The study's aim was with the help of the proposed framework (CAG) to help the students in their studies and also by analysing the students' in-game behavior with the help of Bayesian networks to model their personality. This study was focused only in modeling the Extroversion and Openness trait from the big five personality model. The efficiency of the proposed framework was evaluated with an experiment of 45 participants. The proposed framework had a high level of accuracy with a good and a moderate agreement degree with the BFI.

Another important study contacted from Afroza et al. (2021). In their study they developed a 3D digital game for the purpose of measuring some traits of the player's personality. The target of the study was the creation of an alternative method for capturing someone's personality other than the traditional questionnaires which can be a tedious and intrusive process where users may alter their responses to questions due to

their awareness of the questions asked. The measured traits were based on the OCEAN model and were the Extraversion, Conscientiousness and Neuroticism traits. To measure the study's game effectiveness, they contacted an experiment. For the experiment they gathered 30 participants, having the majority of the participants in the age of 18-24. The participants played the game and answered a BFI questionnaire, then they compared the game's result with the BFI's. The study's results showed a moderate correlation on the Neuroticism trait and small correlation on the Extraversion and Conscientiousness.

3 Methodology

This chapter describes the processes, the methods and procedures used for the creation of the game and for the data analysis.

3.1 Design the game

In the scope of this thesis a serious game ("Code please") was created. The game has two main purposes. The first is to assist students in their activities and the second is to gather data so a machine learning model could be made to predict the students' profile through their in-game behavior. Finally, as a game it should provide a level of entertainment to the player (student).

To cover these targets a story-based game was created. The story-based game genre has as a main engaging mechanic to immerse the student in the story of the game's world and sometimes the story can be influenced through the player's actions. The actions and choices of each player/student in the story can generate relevant data for their inner world. Lastly through the story the students can be tasked to accomplish exercises similar to the ones in their studies and with the help of the game elements the process can be more entertaining.

3.1.1 Flow of the game

The game simulates the passage of time in days for a total of eight days. The game starts with a mini tutorial for the purpose of introducing the story and the controls of the game to the player. The tutorial occurs in the designated student's /player's in-game apartment.

The student plays as a fresh trainee in a small company. The company's main purpose is to evaluate and rank other programmers for their skills. As a trainee the student evaluates some entry level code snippets from some "other" NPC applicants. The trainee can accept or reject the code snippet based on the applicant's assignment. The code is written in the programming language C and its difficulty was set to match the knowledge level of the participants of the experiment (**ANNEX B**). The code exercises were checking the abilities of the students regarding the flow of actions in C, the use of if-else statements either syntactically or logically especially how not to mix subsets of conditional statements, loops, the ability of the students to read and understand the code's functionality and finally some general knowledge in C.

After the compilation of the tutorial the student starts to play the main game and begins his/her work at the company. At the start of the in-game's day the trainee has to go to work and accomplish his/hers "work duties". After the completion of the "work duties" the trainee can leave the office and the day's event occurs. Every day has a different event. In the events the trainee meets his/hers colleagues from work and interacts with them. After the compilation of the day's event the day finishes and the next day in the office starts. The game's main story progresses through the days' event with a final choice needed to be taken from the student on the last day of the game.

3.1.2 Story of the game

The student starts in his/hers in-game room. There are some open boxes indicating that he/she recently moved there. In this area the student is introduced to the game's controls and mechanics. By opening the computer and checking the mails the student learns that he/she has been accepted as a trainee in a company. The company validates and rates the skill level of programmers. The routine of the trainee is first to go and finish his/ hers work duties and after that there is an event each day. In the event the trainee meets the other colleagues from the work, interacts with them and progresses that way the story of the game. The interaction is in the form of a conversation.

During the game the company's director informs the trainee that the company has some financial difficulties. Meaning that the company may need to lay off one of the workers in the near future. Additionally he asks the trainee to observe the other workers indicating that he may ask him/her in later time about who would recommend to lay off. Indeed, at the end of the eighth day the director asks again the trainee which colleague does he/she thinks is better to lay off and why. For the why the student can select either an answer based on the behavior of the chosen NPC-colleague (eg. fooling around too much) or he/she can say that he/she choosed the colleague at random without any particular reason. The introduced colleagues don't have any particular bad characteristics meaning there isn't any obvious answer for which colleague is good to leave for the company. In the next paragraphs each day's events will be shortly presented. Additionally, the Table 3-1 presents shortly the event that took place each day.

Table 3-1: Day Event Short

Day 1	Welcome Party
Day 2	Meet teleworking colleagues

Day 3	Gift idea about colleague
Day 4	Director asks trainee (student) if he wants to accept more difficult tasks
Day 5	Director asks trainee to observe the other colleagues
Day 6	Trip to art exhibition
Day 7	Talk over coffee at work
Day 8	Director asks trainee's opinion about an important matter

On the first day there is a welcome party at the company for the arrival of the trainee (the player) and another newly hired colleague. There the trainee meets most of his/her work colleagues and the company's director. The trainee has the choice to leave the party earlier or talk a bit more with his/her colleagues. If the trainee chooses to stay at the party then he/she can ask some additional questions to the future work colleagues.

On the second day the trainee meets two more colleagues that didn't come to the welcome party because they were tele-working. They also asked the trainee's point of view regarding the ideal teleworking policy.

On the third day a colleague goes to the trainee's office and asks his/ her help for a present regarding another working colleague. The conversation is around gift ideas and colors to use for the wishing card.

On the fourth day the company's director asks the trainee if he/she feels confident to handle more difficult assignments or the trainee prefers to stick to the difficulty level he/she had till now. The game also gives a hint that if the trainee accepts the more difficult tasks then the new assignments will be harder with higher impact on the player's overall score.

On the fifth day The company's director informs the trainee that the company has some financial issues and there is the possibility of needing to lay off one of the hired employees. He asks the trainee if he/she can as a fresh employ observe the other colleagues for any bad influence regarding the work environment because in the future he might want the trainee's opinion regarding the matter. The director also mentions that he won't just blindly follow the trainee's suggestion but he would like the extra input. The trainee can either accept the task, try to convince the director to pass the task to

someone else or question the director regarding why he/she was chosen for a task of that magnitude but in the end the trainee will have to accept the task.

On the sixth day some colleagues organize a trip to an art exhibition. There they play a minigame where each colleague and the trainee try to guess each art piece's price. The art pieces as well as their prices were taken from a real website. In the event they talk a bit about art as also the NPCs ask the trainee if he/she liked the guessing game.

On the seventh day some colleagues go out for a coffee after work. There they talk a bit about the weather. The trainee has the choice to direct the conversation as an outcome to discuss different topics depending on the trainee's choices.

On the eighth day the company's director asks the trainee about his/her opinion for which colleague does he/she thinks is better to lay off. After the trainee chooses a work colleague the director will also ask why did the trainee choose that colleague. Then the trainee can either say something about the colleague's personality or he/she can say that the choice is completely random.

3.1.3 Design Choices

From the four distinct relationships mentioned in the section 2.2.2 between a player and a game's avatar we targeted the creation of the relationship avatar-as-me. As the players' in-game behavior were to be used to model each player's personality the relationship avatar-as-me would minimize the "noise" generated from the avatar's personality. To achieve this relationship the following design choices were made.

The game was created in a first person perspective to simulate how we normally observe the world (Figure 3-1). To avoid having the player spotting differences with the game's avatar breaking the avatar-as-me relationship the player couldn't see his/hers avatar body. Another element used to make the player feel as one with the avatar was the game's scenario. The scenario was realistic with the purpose to make it easier for each player to imagine himself/herself in the avatar's situation. Another element used for enhancing the avatar-as-me relationship are the interactions with the other Non-Playable Characters (NPC). When the player has a conversation with an NPC the NPC will respond according to the player's dialogue choice making the whole conversation feel more interactive and real.

Figure 3-1: Player's point of view

Except the elements targeting the relationship between the player and the game's avatar there were some other elements targeting to motivate the player to continue playing. These elements are focused on making the game more interesting, fun and educational.

First the player could see his/ hers score. The score was influenced by how well the player had performed on his/ hers "work duties". If the player made a mistake in his/ hers work the score would immediately decrease. On the contrary if the player had performed well the score would increase. Another element used to assist the players in their studies is the immediate feedback. After the player submitted the evaluation of a code snippet, then one of the following scenarios would happen. If the evaluation was correct an image of a green check would appear informing the player that he/she did a good job (and his/her score would increase). In comparison if the evaluation was incorrect then an image of a red "X" would appear with a small text below explaining the key element why the other answer is the correct one. With this feedback the players could "fill" their learning gaps and test their knowledge (Figure 3-2).

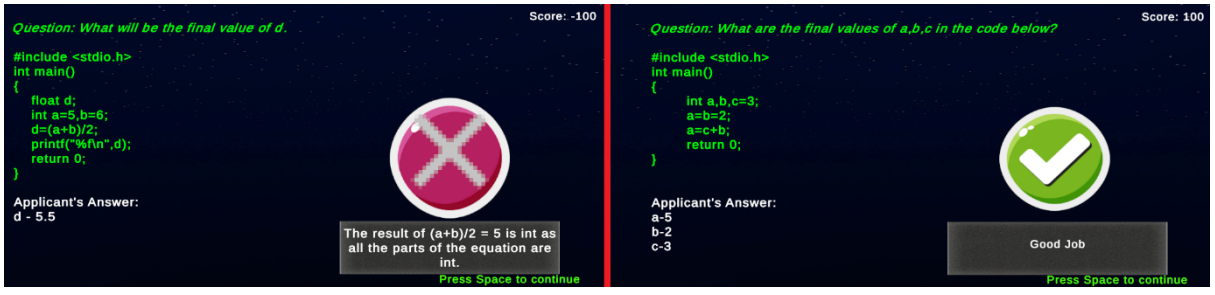

Figure 3-2: Wrong and correct assessment of the code snippet

The code snippets had only logical errors and they were small in size. This approach was used so the players wouldn't get tired by reading a big program or

incorrectly assessing a code snippet because they missed a misspelled variable or by some other syntactical error that can be easily discovered with an interpreter.

For gathering data to model the learner's personality an interactive story was created. In the story the student/ player has some conversations with the NPCs. In the conversations the student can select his/hers response through a variety of answers depending on the context. The conversations were used to advance the story, immerse the student in the game's world and to capture information about the student's inner world.

To accurately capture information about the student's inner world a lot of the conversations had questions similar to the ones from the BFI (Choungourian 1967) (Watson and Clark 1997) and elements from similar works regarding the openness and extrovert trait. Additionally some extra questions were included based on our understanding of the likings and thoughts of the targeted traits aiming to gather additional information.

More specifically for the openness trait the conversations were trying to explore the student's will to try new experiences, the amount of risk the student is willing to take, the student's views about art, the student's color preferences and the tendencies regarding the way the students interacts with the NPCs.

For the extraversion trait the conversations targeted the student's enthusiasm and willingness to communicate with NPCs, the student's color preferences, the risks the student was willing to take and finally the tendencies regarding answering NPCs.

Finally, some of the extra questions had a positive impact on prediction of the final model. Those questions for the openness trait were the student's favorite season, the student's opinion about rain, the final choice the student made on the last day as also the reason the student gave for taking that choice. For the extrovert trait were the student's favorite season, the student's opinion about rain and the student's opinion regarding teleworking.

The days' dialogues were designed with the idea of capturing information about a specific trait. In Table 3-2 the target trait or traits can be easily observed. Although the traits discussed here are based on the BFI and from previous studies some extra capture information outside of these references had a positive impact on the prediction model. A good example of this is the explanation the student gave about his/her final choice on day 8 where the captured information had a positive impact on predicting the openness trait.

Table 3-2: Day design based on their target trait

Days	Conversation Fork	Targets Extroversion	Targets Openness
1	Yes	Yes	Yes
2	No	Yes	No
3	No	Yes	No
4	No	Yes	Yes
5	Yes	No	No
6	No	No	Yes
7	Yes	Yes	Yes
8	No	No	No

In the next paragraphs they are presented the dialogues that were designed each day to target a specific trait. It also shortly described the events of each day. More detailed information about each day's event are presented in 3.1.2 section. It is important to note that not all designed dialogues were used in the final model.

Day 1 event: A small welcome party was prepared for the player and another colleague that started work there the same day. Table 3-3 and Table 3-4 present the targeted traits for this day.

Table 3-3: Day 1 Extrovert Trait

Extrovert trait	Dialogues target
Introverts tend not to like chit-chat and they usually prefer to go straight to the point. For this reason in some dialogues the choices were separated into straight forward answers and some more playful.	● The way student introduce himself/herself ● The way the player introduce himself/herself and the new colleague to others (if he will prefer to speak first or to let the others do the introductions) ● The way the student chose to answer about how long he/she is working for the company. (Say straight forward that it is his/her first day or choose to say something more playful.)

Another capture information was the willingness of the student to interact with other people (NPC).	Dialogues for capturing these information: ● If the student was feeling excited for the party ● The willingness the student showed to meet other colleagues at the beginning of the party ● If the student chose to leave the party early or to stay longer.

Table 3-4: Day 1 Openness Trait

Openness Trait	Dialogues target
People with a high openness score are open to new experiences. They also tend to like more unique tastes.	● Student's likeness about spicy food (Figure 3-3) ● Student's likeness about bitter food

Figure 3-3: Day 1 Openness Dialogue

Day 2 event: On the second day two colleagues introduce themselves and they talk to the player about telework. Table 3-5 and Table 3-6 present the targeted traits for this day.

Table 3-5: Day 2 Extrovert Trait

Extrovert trait	Dialogues target
Usually people with high extraversion score prefer to meet new people so we expect players with a high extraversion score to be less supportive about teleworking policies.	● What kind of teleworking policies should the companies implement (Figure 3-4). The player could select: ○ Possibility for entirely remote job

	○ Hybrid system ○ Or only after request ● The player was also asked if he/she is bother by having meetings through distance or he/she prefers for the meetings to be face-to-face

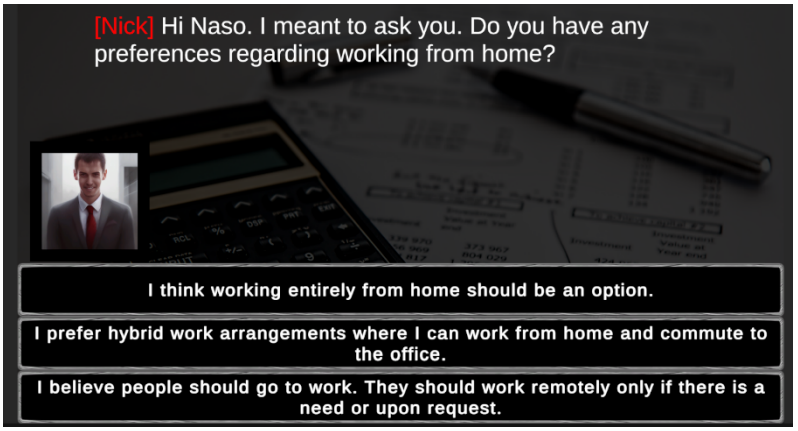

Figure 3-4: Day 2 Extrovert Dialogue

Table 3-6: Day 2 Openness Trait

Openness Trait	Dialogues target
No Openness traits were targeted	No Openness traits were targeted

Day 3 event: On the third day a colleague asks the student to help her about a gift she is thinking of giving to another work colleague. Table 3-7 and Table 3-8 present the targeted traits for this day.

Table 3-7: Day 3 Extrovert Trait

Extrovert trait	Dialogues target
The initial dialogue captures some information about how willing the student is to help her with the given task. The choices have both straightforward answers and playful ones.	The colleague asks the student what gift does he/she suggests. On the first day the birthday colleague had mentioned his preference for football (for the student to learn this the student should have stayed at the party). This question is designed to further capture if the student stayed at the

	party and paid attention to the names /dialogues of that day.
Studies have shown that people with high extraversion scores prefer warm colors more than cold ones.	The colleague asked the student what color he/she suggests for the envelope containing the wishing card (Figure 3-5).

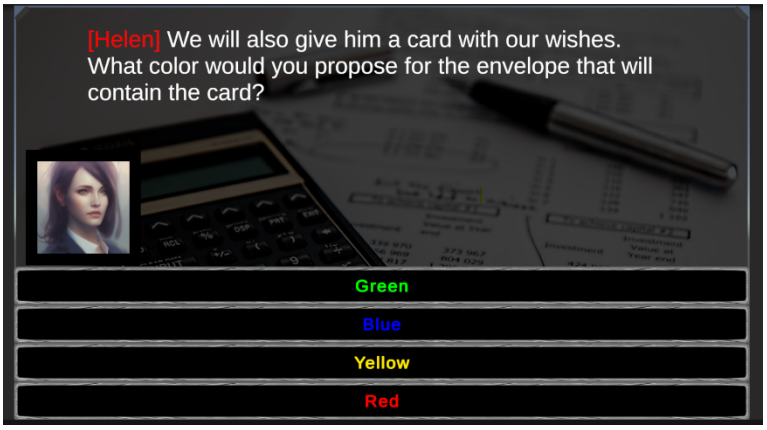

Figure 3-5: Day 3 Extrovert Dialogue

Table 3-8: Day 3 Openness Trait

Openness Trait	Dialogues target
No Openness traits were targeted	No Openness traits were targeted

Day 4 event: On this day the agency's director asks the student if he/she is willing to take on more hard tasks. The game informs the student that by accepting he/she will have more difficult tasks but the reward for successfully completing the work as well as the punishment from failing to do so will be greater (Figure 3-6). Table 3-9 and Table 3-10 present the targeted traits for this day.

Table 3-9: Day 4 Extrovert Trait

Extrovert trait	Dialogues target
People with high scores on extrovert and on openness they tend to take more risks.	Agency's director asks the student if he/she is willing to take on more hard tasks.

Table 3-10: Day 4 Openness Trait

Openness Trait	Dialogues target

People with high scores on extrovert and on openness they tend to take more risks.	Agency's director asks the student if he/she is willing to take on more hard tasks.

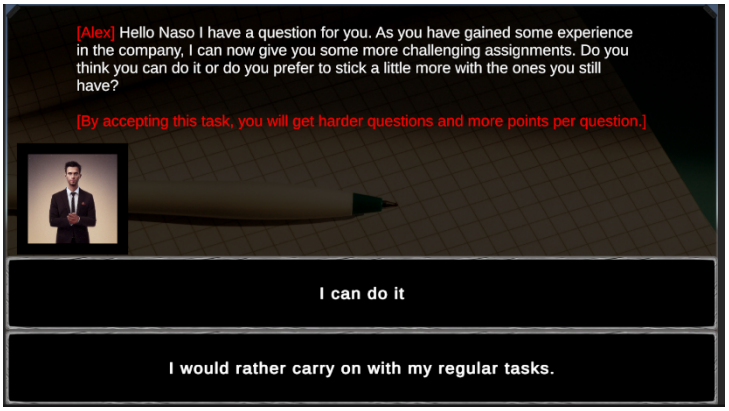

Figure 3-6: Day 4 Extrovert & Openness Dialogue

Day 5 event: On this day the agency's director asks the student to observe the other colleagues. The dialogues for this day are mostly focused on progressing the story by giving some stakes and some responsibilities to the student. Table 3-11 and Table 3-12 present the targeted traits for this day. Although the dialogues aren't based on a specific research a variety of different choices are given to the student.

Response to the task: What was the initial response to this task?

- The student say he/she will try his/her best
- Say that he /she will do it but without any pleasure
- Say that if it is necessary he/she will do it

What question did the student choose to ask the director. The choices are.

- To give the task to someone else
- To ask a specification about the task (what is the students looking out for)
- To not ask any question at all

Table 3-11: Day 5 Extrovert Trait

Extrovert trait	Dialogues target
No Extrovert traits were targeted	No Extrovert traits were targeted

Table 3-12: Day 5 Openness Trait

Openness Trait	Dialogues target
No Openness traits were targeted	No Openness traits were targeted

Day 6 event: On this day some colleagues organize to go to an art exhibition with the student. At the exhibition they play a game where the student and the work colleagues are trying to guess the selling value of the art piece. First the colleagues are trying to guess the cost of the art and then after hearing everyone's choices the student could also guess what is the right price (from four choices). Table 3-13 and Table 3-14 present the targeted traits for this day.

Table 3-13: Day 6 Extrovert Trait

Extrovert trait	Dialogues target
No Extrovert traits were targeted	No Extrovert traits were targeted

Table 3-14: Day 6 Openness Trait

Openness Trait	Dialogues target
People with high openness score tend to like art and usually they have a variety of artistic interests (BFI questions, Annex A)	• Expected people with high openness to be more familiar with art and they could have more accurate guesses than people with low openness (Figure 3-7) • At the end of the mini-game the work colleagues ask the student if he/she likes abstract art as also if he liked the mini-game

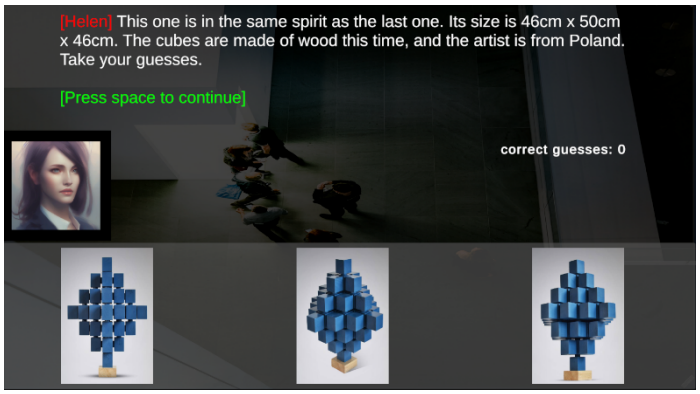

Figure 3-7: Day 6 Openness Dialogue

Day 7 event: On this day the student goes out for a coffee with some work colleagues. At the coffee the student can direct the conversation. Table 3-15 and Table 3-16 present the targeted traits for this day.

Table 3-15: Day 7 Extrovert Trait

Extrovert trait	Dialogues target
The student can choose to ask the group a question and direct the conversion. People with high extraversion tend to be more social so it is expected that students with high extraversion scores would choose a more interesting or playful subject (Figure 3-8).	The possible of the student choices are: ● Ask them about a wild story. ● Ask them about their weekend plans. ● Ask them if they like to work at the agency.
A person's feelings and actions towards rain, can be influenced from both openness and extraversion traits. For example people with low extroversion would prefer to stay home in contrast with people with high extraversion score.	The student's choices regarding their thought and actions about rain are: ● Prefer to stay home ● Like to go out but avoid getting wet ● Walk in the rain ● Doesn't have any major negative or positive feeling about rain

Table 3-16: Day 7 Openness Trait

Openness Trait	Dialogues target
A person's feelings and actions towards rain, can be influenced from both openness and extraversion traits. People with high openness score might want to take the opportunity and experience something more unique (eg. walking in the rain).	The student's choices regarding their thought and actions about rain are: ● Prefer to stay home ● Like to go out but avoid getting wet ● Walk in the rain ● Doesn't have any major negative or positive feeling about rain

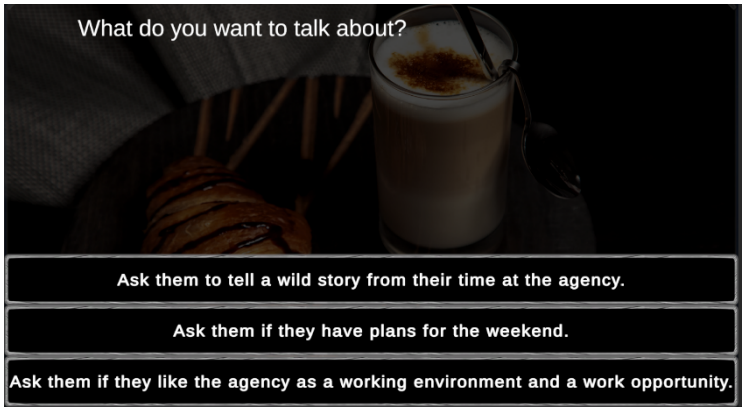

Figure 3-8: Day 7 Extrovert Dialogue

Day 8 event: On this day the student has to declare to the company's director a colleague that he/she believes can be laid off. The day is structured with the main purpose to build the story, make the game more engaging and close the game with an ending. Regarding that the student's work colleagues are designed by having in mind to "represent" a trait from the targeted traits.

- There is a colleague with a high extraversion score, always ready to say something and join almost every event.
- There is a colleague with a low extraversion score. He doesn't join many events and he doesn't participate a lot in any small talks.
- There is a colleague with a high openness score. She likes art and exploring weird conversation topics.
- There is a colleague with balanced scores to openness and extroversion, but she is new to the work (started same day as the students)
- There is a colleague with balanced scores in both extrovert and introvert traits.

The student could also justify his/her choice regarding the colleague he/she chose based on a characteristic of the chosen colleague or to just say he/she picked someone at random.

It is expected that people with similar personality traits make similar choices. As the presented colleagues don't have any particular good reason to be laid off the student's choice should be influenced by the student's preference regarding a character. For example a student with a high openness score might like the other colleagues with a high openness score and not choose them.

3.2 Experiment

After the design and creation of the game there was the need to gather data from the learners. For this step an experiment was conducted at the University of Macedonia.

3.2.1 Participants

The participants for this experiment were the students of the Department of Applied Informatics, of the University of Macedonia studying the first-year curriculum about the programming language C. As mentioned before the game was designed to match the student knowledge on this curriculum. For the students who completed the participation in the experiment a small bonus was given. The criteria to participate in the experiment was first to study the curriculum to ensure the necessary skill level of the programming language C and for the students to have a sufficient level of reading and understanding English. In total 152 students were shown interest in participating in the experiment. After removing participants that either didn't show up, had technical issues or not sufficient enough knowledge of English, the final participants were 107 students. From those 107 students 72 were male and 35 were female.

3.2.2 Procedure

Because the total number of the participants was way larger than the computers we had at the university lab it was decided for the participants to play the game on their own computer and to only use the lab computers in case a participant couldn't play the game on his/hers home computer. For playing the game we set a day and a time. To ensure that it would be the first time the participants would play the game we shared the game's link at the scheduled time. In addition to further avoid possible problems we had an open zoom call with the participants in order to easily explain the game's controls, guide them if they were stuck and troubleshoot any problems. Because the created digital game for this study is a story game the data from the first playthrough are extremely important as students in the second playthrough might try to take different paths and choices.

The students started by playing the game. The game needed around 40 minutes for its compilation. Before the start of the game the students were introduced to the game's controls and story. At the beginning of the game there was a mini tutorial area to teach the students the controls, the user interface and the game's flow. After the completion of the game the learners had to send us the game's data and to fill a BFI

questionnaire. A deadline for the questionnaire was given not later than the end of the day. The deadline for the questionnaire was set this way for multiple reasons. We wanted the participants (students) to fill the questionnaire near the time that they played the game so they have a similar mindset and train of thought. Additionally, we didn't force the participants to immediately take the questionnaire after the game because they might have been tired from the game or the other procedures and we wanted to avoid the possibility of them answering the questionnaire hastily.

3.3 Data Analysis

For the data analysis the questions from the BFI were used as a ground truth for the personality of each learner. As the game was designed to capture only the Openness and Extraversion trait the given BFI questionnaire (**ANNEX A**) had only the questions related for these two traits. More specifically 8 questions were for the extraversion and 10 for the openness trait. As for the trait description there are three distinct categories. Low, balanced and high. To determine each category the mean of the learner's score was used. The mean can be between 1 and 5. The mean-scores in the range 1 to 2.4 were labeled as low, 2.5 to 3.4 as balanced and 3.5 to 5 as high.

3.3.1 Exploring the data

First it was essential to explore and validate that we had sufficient data for each category regarding every trait. For the extraversion trait the distribution was: 18 students had low scores, 50 had balanced scores and 39 had high scores. For the openness trait the distribution was: 2 learners had low scores, 46 had balanced scores and 59 had high scores. Because of the low number of learners with low scores the prediction of low openness was excluded. After further investigation the learners with low score on openness had a mean score of 2.4 which is borderline for them to be characterized as balanced. For this reason, it was decided to keep their metrics but label them as balanced (Table 3-17).

Table 3-17: Participant personality distribution

Extrovert-Openness	Extrovert	Openness
Low	17	0
Balanced	51	49
High	39	58

3.3.2 Data from the game

The data gathered from the game were in the form of a dictionary. The NPC's question ID was used as the key and the student's selected answer was used as the value. Each question's ID was structured in a way to provide some information about the context of the question (instead of using hard to interpret numbers). The student's (student's) conversation choices were presented in the same order for every individual game (they weren't shuffled) so only the student's answer index was kept (Figure 3-9).

```
event_day1_Qid_mt_collegues,0
event_day1_Qid_sayMyName,2
event_day1_Qid_whenStarted,3
event_day1_Qid_MeetNewPeople,2
event_day1_Qid_Initial Greeding,1
event_day1_Qid_Pass time at parrty,2
event_day1_Qid_Ready for questions,2
event_day1_Qid_Spicy food,0
event_day1_Qid_Bitter food,1
event_Event2_Qid_sayMyNameD2,1
event_Event2_Qid_Teleworking,1
event_Event2_Qid_Teleworking2,1
event_Event2_Qid_AgreeOnTelework,0
event_Event2_Qid_beyD2,2
event_Event2_Qid_giftQ,2
event_Event2_Qid_GiftChoice,0
event_Event2_Qid_ColorChoice,1
event_Event2_Qid_riskTaking,0
event_Event2_Qid_Alex_task_response,1
event_Event2_Qid_Alex_task_question,1
event_Event2_Qid_like_abstract_art,1
```

Figure 3-9: Question ID (Key) - Student response index (value)

As it was mentioned in the design choices section the NPC's were reacting to the student's choices. Meaning that on some occasions the students could follow a different conversation path based on their answers. Although the paths were designed to reconnect to the main conversation path they still generated some unique key-value pairs for some students. In order to create homogeneous data the data gathered from forked conversation paths were discarded. Of course, the initial choice which conversation path the student chose was kept in the training.

For example, the student on the first day he had the choice to leave the welcome party earlier or to stick around. If the student stayed at the party then he/she would had some extra interactions with the NPCs (Figure 3-10).

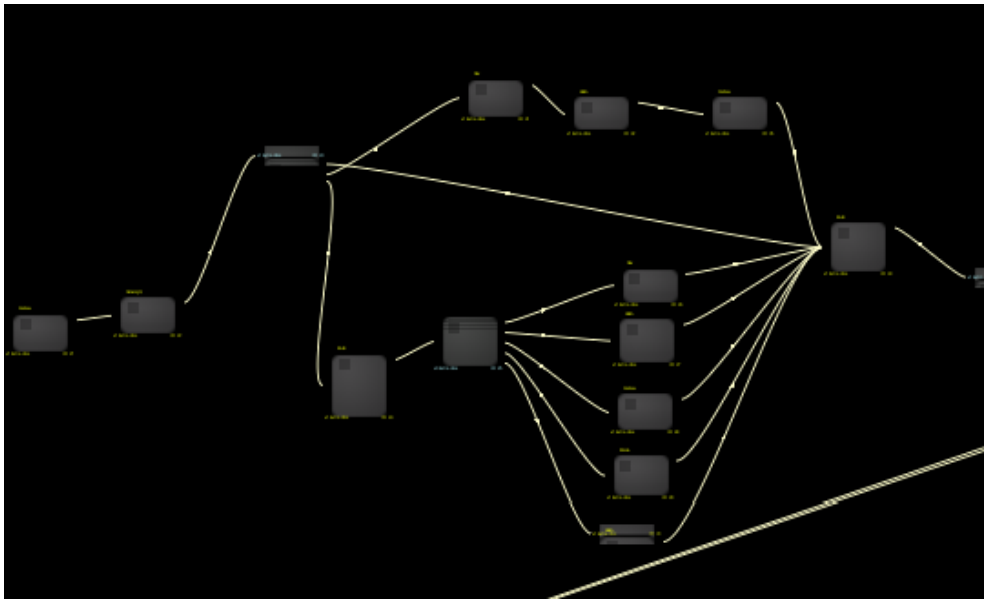

Figure 3-10: Different conversation paths

The student's score and success in his/hers work "duties" weren't taken into account for training the model. These two metrics weren't used in the analysis because they mostly represent the stundent's knowledge for the programming language C and not his/ her personality. This approach was used to avoid overpopulating the training data with low relevance data.

Before the use of any machine learning algorithms, first we tried to see any pattern in the data using only data visualization.student We visualized the preference of all students regarding their dialogued choices. For example in Figure 3-11 we can see the students preferences regarding the reason of the selected colleague on day 8. Almost 44% of the students chose to say that their final answer on day 8 was random and the other 56% gave a reason based on the selected NPC personality. We can also see that there isn't an obvious preference for either answer, neither for the extrovert trait nor for the openness trait. From the 51 balanced extrovert 25 selected the first answer (they selected a random colleague) and 26 selected the second (they they selected the colleague based on his/her personality).Similar was the case for the other trait characteristics. Although it is presented that the machine learning algorithms found this information in combination with others useful. After using a similar examination for each player choice in the dataset we didn't observe any clear preference between any player's choice and the person's personality model.

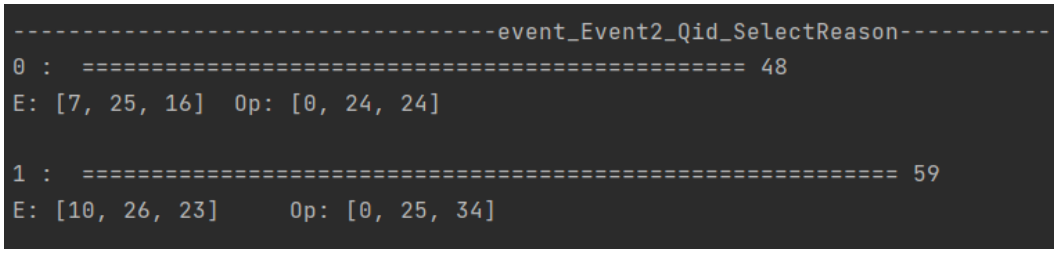

Figure 3-11: Day 8 selection reason each trait

For the data analysis 3 algorithms were used: KNN, Decision Trees and Naive bayes. To train the algorithms different subsets from the main conversation roots were used. Questions and replies regarding each trait were kept in the corresponding subset. For example the question if the player likes art which is very similar to a question from the BFI regarding the openness was kept in the subset for the openness trait and not in the extrovert's.

Except the conversations inspired by the BFI there were some question-answer pairs created with the use of results from similar studies regarding the student's in game behavior and his/her personality. Additionally some extra dialogues were created either based on our understanding of the personality traits or for the sake of having more natural dialogues.

To determine which key-pair variable subset was optimal for each model there are multiple methods that could be used. In this study we selected to evaluate each subset performance through a metric (eg. F1-score, accuracy, ROC curve etc.). Because of the size of the study's dataset's dimensions (22 in total) we couldn't all the possible subsets. For this reason for each trait (Extraversion, Openness) we started by including all the dimensions (variables) and then we started to "eliminate" bad for the model variables. The next paragraph has in detail the "elimination" process we used

First the model was trained with the initial variable set (in the beginning we started with all the variables) and then a second model was trained with a subset of the first model's variables. Usually for the second model we removed one or two variables from the first model. The selected test-to-remove variables were based on the relevance the variables had with the predicted trait. If the second model had better metrics then the removed variables were eliminated. We didn't test to add later in the process previews eliminated variables but variables that "survived" a testing round could be reselected to test if they still contribute positively to the model's overall performance. The procedure stopped and the subset was finalized when we reached the point where by removing any

single or pair of variables the model's performance was getting worse. This can be seen in Figure 3-12. The list "commonListAll" has all the overlapping IDs that were present in every player's dataset. In comparison the list "keptList" is the tested subset at the time, in this case it is also the final subset for the extraversion trait.

```
commonListAll = ['event_day1_Qid_mt_collegues', 'event_day1_Qid_sayMyName', 'event_day1_Qid_whenStarted',
                 'event_day1_Qid_MeetNewPeople', 'event_day1_Qid_Initial Greeding',
                 'event_day1_Qid_Pass time at parrty', 'event_day1_Qid_Ready for questions',
                 'event_day1_Qid_Spicy food', 'event_day1_Qid_Bitter food', 'event_Event2_Qid_sayMyNameD2',
                 'event_Event2_Qid_Teleworking', 'event_Event2_Qid_Teleworking2', 'event_Event2_Qid_AgreeOnTelework',
                 'event_Event2_Qid_beyD2', 'event_Event2_Qid_giftQ', 'event_Event2_Qid_GiftChoice',
                 'event_Event2_Qid_ColorChoice', 'event_Event2_Qid_riskTaking', 'event_Event2_Qid_Alex_task_response',
                 'event_Event2_Qid_Alex_task_question', 'event_Event2_Qid_like_abstract_art',
                 'event_Event2_Qid_GuessArt1', 'event_Event2_Qid_GuessArt2', 'event_Event2_Qid_ArtReady',
                 'event_Event2_Qid_GuessArt3', 'event_Event2_Qid_LikeArtGame', 'event_Event2_Qid_Ask them at coffee',
                 'event_Event2_Qid_rain', 'event_Event2_Qid_FavSeason', 'event_Event2_Qid_FinalChoice',
                 'event_Event2_Qid_SelectReason']


keptList = [
                 'event_Event2_Qid_giftQ',
                 'event_Event2_Qid_ColorChoice', 'event_Event2_Qid_Alex_task_response',
                 'event_Event2_Qid_Alex_task_question', 'event_Event2_Qid_like_abstract_art',
                  'event_Event2_Qid_ArtReady',
                 'event_Event2_Qid_GuessArt3', 'event_Event2_Qid_LikeArtGame', 'event_Event2_Qid_Ask them at coffee',
                 'event_Event2_Qid_rain', 'event_Event2_Qid_FavSeason', 'event_Event2_Qid_FinalChoice',
                 'event_Event2_Qid_SelectReason']
```

Figure 3-12: List of all IDs and list of used subset

For the comparison of the models various steps were used. The models were measured based on their accuracy, f1-score, Kappa, ROC/AUC score. To find the best model of each tested subset the model was trained 10 times with different seeds. For each seed the model was also trained on all the possible PCA combinations (from reducing the training dimensions to one to completely not use PCA). Because of the high number of different trained models we first set a value as the main comparison metric and the programme presented the full metrics of the best model regarding this value. We used the AUC/ROC score as a main value. If the other metrics were low then for the specific iteration a different value would be used.

The data analysis was contacted with the use of the programming language python. The algorithms (Naive Bayes, Decision Trees, KNN) used in this study as also the metrics used to measure each algorithm's performance were created and measured by using standard methods from the sklearn library. The Naive bayes was implemented from sklearn.naive_bayes import GaussianNB, the decision trees where implemented from sklearn.tree import DecisionTreeClassifier and the KNN was implemented from sklearn.neighbors import KNeighborsClassifier.

For each algorithm a different function was created (Figure 3-13). That approach was used in order to be able to call the function multiple times with different subsets. The function has as an input the data-frame, the name of the y_column, if it has to use PCA and if yes how many dimensions do we want to keep at the end. At the end the function

returned all the metrics mentioned above. The main py file that calls the function keeps all the metrics from the best performance combination of PCA and seed for each model.

```
def bayes_unscaled(df, usePCA=False, PCAdim=2, y_column = "ExtrClass", multiclass = False):
    if usePCA:
        X = myPCA(df.loc[:, df.columns != y_column], dim=PCAdim)
    else:
        X = df.loc[:, df.columns != y_column]
    y = df[y_column].values
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=testSize, random_state=randomSeed)
    nbModel = GaussianNB()
    nbModel.fit(X_train, y_train)
    y_pred = nbModel.predict(X_test)
    cf = confusion_matrix(y_test, y_pred)
    cv = cross_val_score(nbModel, X_train, y_train, cv = 2)

    if multiclass:
        y_pred_proba = nbModel.predict_proba(X_test)
        Roc = roc_auc_score(y_test, y_pred_proba, multi_class="ovr")
    else:
        Roc = roc_auc_score(y_train, nbModel.predict_proba(X_train)[:, 1])
    kappa = cohen_kappa_score(y_test,y_pred)
    return f1_score(y_test, y_pred, average="weighted"), metrics.accuracy_score(y_test, y_pred), cf, cv, Roc, kappa
```

Figure 3-13: Naive bayes python code

3.4 Results

The results showed good scores for all the used metrics. For each trait a different algorithm, seed and PCA was best suited. For the extrovert trait the most suited algorithm was the decision trees. The final model is able to predict the player's extrovert characteristic (low, balanced, high) based on their in-game behavior with accuracy 72%. Additionally the f1-score was at 0.736 and the AUC/ROC score was at 0.741 showing that the model is reliable. To further determine the agreement degree between the model and the BFI questionnaire the Cohen's kappa coefficient was calculated. Cohen (1960) suggested the kappa result be interpreted as follows: values $\leq$ 0 as indicating no agreement and 0.01–0.20 as none to slight, 0.21–0.40 as fair, 0.41– 0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1.00 as almost perfect agreement. The decision trees had a score of kappa = 0.54 (Table 3-18).

Table 3-18: Extrovert models metrics

Extrovert	Accuracy	F1-score	Kappa	ROC AUC	PCA
Naïve Bayes	0.5454	0.566	0.276	0.627	12
Decision Trees	0.727	0.736	0.541	0.744	6
KNN	0.59	0.578	0.292	0.559	8

For the openness trait higher results were achieved. The most suited algorithm was Naïve Bayes. The model had accuracy of 81% to correctly determine the learner's openness trait (balanced, high). Also, the f1-score of 0.811 and the AUC/ROC score of 0.75 support the model's results (Table 3-19). Lastly the Cohen kappa score of the algorithm was 0.636 indicating a good agreement with the BFI questionnaire (Table 5.4). For the openness it was found the best performing model was the Naive bayes with initial 13 dimensions and then reducing them to 12 with PCA

Table 3-19: Openness models metrics

Openness	Accuracy	F1-score	Kappa	ROC AUC	PCA
Naïve Bayes	0.818	0.811	0.636	0.75	12
Decision Trees	0.545	0.541	0.09	0.998	4
KNN	0.636	0.633	0.272	0.617	12

All the models showed better results when the dimensions were reduced. From the initial 31 dimension (one dimension per question-answer pair) the algorithms had more accurate predictions when used around 12-13 dimensions. Some question-answer pairs had a positive impact for predicting both the characteristics of the extrovert and openness trait (Figure 3-14).

	Question ID	Openness	Extrovert
2	event_day1_Qid_mt_collegues		
3	event_day1_Qid_sayMyName		
4	event_day1_Qid_whenStarted		
5	event_day1_Qid_MeetNewPeople		
6	event_day1_Qid_Initial Greeding		
7	event_day1_Qid_Pass time at parrty		1
8	event_day1_Qid_Ready for questions		1
9	event_day1_Qid_Spicy food		
10	event_day1_Qid_Bitter food		
11	event_Event2_Qid_sayMyNameD2		
12	event_Event2_Qid_Teleworking		1
13	event_Event2_Qid_Teleworking2		1
14	event_Event2_Qid_AgreeOnTelework		
15	event_Event2_Qid_beyD2		
16	event_Event2_Qid_giftQ	1	1
17	event_Event2_Qid_GiftChoice		1
18	event_Event2_Qid_ColorChoice	1	1
19	event_Event2_Qid_riskTaking		1
20	event_Event2_Qid_Alex_task_response	1	1
21	event_Event2_Qid_Alex_task_question	1	
22	event_Event2_Qid_like_abstract_art		
23	event_Event2_Qid_GuessArt1		
24	event_Event2_Qid_GuessArt2		
25	event_Event2_Qid_ArtReady	1	
26	event_Event2_Qid_GuessArt3	1	
27	event_Event2_Qid_LikeArtGame	1	
28	event_Event2_Qid_Ask them at coffee	1	1
29	event_Event2_Qid_rain	1	1
30	event_Event2_Qid_FavSeason	1	1
31	event_Event2_Qid_FinalChoice	1	
32	event_Event2_Qid_SelectReason	1	

Figure 3-14: Question ID used for each trait

After observing the confusion matrix of the best performing model for the extrovert trait (decision trees) from the test set (data not used for training) we can see that the model can confidently determine the balanced and high extroverts but have some difficulties with the low extroverts. This can be explained by the training set's data distribution as the low extroverts were significantly lesser in number than the other two categories. Meaning the algorithm is expecting to identify more often balanced and high extroverts (Table 3-20).

Table 3-20: Decision Trees extrovert confusion matrix

Decision Trees - Extrovert	High	Balanced	Low
High	1	1	0
Balanced	1	8	1
Low	1	2	7

After observing the confusion matrix from the test set from the best performing model for the openness trait (Naive Bayes) we can determine that the model is good at predicting the balanced openness and had some difficulties by correctly identifying the high openness (Table 3-21).

Table 3-21: Naive Bayes Openness confusion matrix

Naïve Bayes - Openness	High	Balanced
High	7	4
Balanced	0	11

4 Conclusion

4.1 Summary and conclusions

Our study aimed to design an educational digital serious game with two main purposes. The first purpose of the study is for the game to assist the players to learn the programming language C. Gamified systems and educational games can be very beneficial for educational purposes. For many students digital games are more appealing than studying. Additionally, such environments can inspire students to apply and experiment on their knowledge without the fear of falling.

The second purpose of the study is to capture and model the player's profile through his/her in game behavior. A variety of data can be gathered by observing someone's in-game behavior. These data can be utilized to discover information about the player. Modeling someone's personality profile other than the commonly used self-report can be extremely beneficial. Self-report questionnaires may be subjective to the person's view of himself/herself which can be influenced by social or even personal factors. Additionally, the knowledge of a student's personality model can be beneficial as there are multiple studies regarding what are some preferred techniques for learning, studying and teaching based on the student's personality.

In this study we created a digital game with the above-mentioned purposes. The game was targeted for first year university students and exercises from the university's material was used as an inspiration for some gamified elements in the game. Additionally in the study's scope an experiment was contacted. In the experiment we gathered 107 students studying the first year curriculum for the programming language C and asked them first to to play the study's designed game and then to fill a BFI questionnaire. After the compilation of those tasks we analyzed the students' in-game behavior and we compared it with their BFI scores. With the help of ML algorithms and techniques we trained some models that could predict with high accuracy the student's personality using only the game data. For the selection of the final algorithms various data analytics techniques and algorithms were tested and evaluated. More specific we tested to analyze the data with the ML algorithms of Naïve Bayes, Decision Tress and KNN. We also observed that with the use of PCA techniques the final result were more accurate. Different algorithms achieved better results for each trait. For the Extrovert trait the Decision Tres algorithm had higher score than the others achieving of accuracy 0.727,

F1-score 0.736, Kappa 0.541 and AUC/ROC score of 0.744. For the Openness trait the Naïve bayes algorithm had higher scores than the others achieving of accuracy 0.818, F1-score 0.811, Kappa 0.636 and AUC/ROC 0.75.

This study's results are promising and actions were taken to further advance previews related work. In her work (Denden et al, 2017) she created an educational framework (CAG) to capture the students' in game behavior and then create their personality model. In her study they had forty five participants which were less in number than the participants of this study. Additionally for the creation of the prediction model they used only one ML algorithm (Bayesian networks) without testing the accuracy of other ML algorithms. Finally for the measurement of the final model's efficiency they used the accuracy and the Kappa metrics. Those two metrics alone may not show the full picture.

4.2 Research limitations

The first limitation for this study is the number of participants in the experiment. Although compared to other studies the number is not small it still needs more participants for a more statistically accurate result. Second, the participants were from the same university and the same study field. A bigger variate range could give more accurate results. Another limitation was that in this study we focused only on two traits (Openness, Extraversion) of the Big Five personality model. Last but not least the game was designed only by studying relative works and without the help of a provisional psychologist, meaning that more accurate and personality specific dialogues could be achieved.

4.3 Future extensions

Some suggestions for a future study are first the creation of a game that can be played by more applicants. Although the personality traits focused on this study are considered the most impactful regarding the player's in-game behavior, the modeling of the remaining personality dimensions can be useful. The utilization of additional established personality models or questionnaires can be used for further comparing results. Finally further research can be done on finding the optimal way for grouping students regarding their personality model.

References

Afroza, A., Murray, K., Wünsche, B.C. and Denny, P. (2021) “Who am I?-Development and Analysis of an Interactive 3D Game for Psychometric Testing”, In 2021 Australasian Computer Science Week Multiconference, February, pp. 1-10.

Amra, I.A.A. and Maghari, A.Y., 2017, May. Students performance prediction using KNN and Naïve Bayesian. In 2017 8th international conference on information technology (ICIT) (pp. 909-913). IEEE.

Andrade, A., 2015. Game engines: a survey. EAI Endorsed Transactions on Serious Games, 2(6).

APA Dictionary of Psychology (2023) Available at: https://dictionary.apa.org/personality (Accessed: 1 March 2023)

Barron, J.L., (2015) Comparison of a video game based learning environment and a traditional learning environment, (Doctoral dissertation, Duquesne University).

Bowman, S.L. and Schrier, K., 2018. Players and their characters in role-playing games. In Role-Playing Game Studies (pp. 395-410). Routledge.

Banks, Jaime and Nicholas David Bowman. 2014. “From Toy and Tool to Partner and Person: Phenomenal Convergence/Divergence Among Game Avatar Metaphors.” Proceedings from Papers of Internet Research 16: The 16th Annual Meeting of the Association of Internet Researchers. Phoenix, AZ, October 21-24.

Banks, J., 2015. Object, me, symbiote, other: A social typology of player-avatar relationships. First Monday.

Carenys, J. and Moya, S., 2016. Digital game-based learning in accounting and business education. Accounting Education, 25(6), pp.598-651.

Chen, Z., and Lin, T. (2017) “Automatic personality identification using writing behaviours: an exploratory study”, Behaviour & information technology, Vol 36, No.8, pp.839-845.

Choungourian, A., (1967) “Introversion—extraversion and color preferences”, Journal of Projective Techniques and Personality Assessment, Vol 31, No. 4, pp.92-94.

Clarke, R.I., Lee, J.H. and Clark, N., 2017. Why video game genres fail: A classificatory analysis. Games and Culture, 12(5), pp.445-465.

Clearwater, D., 2011. What defines video game genre? Thinking about genre study after the great divide. Loading..., 5(8).

Cohen, J., (1960) “A coefficient of agreement for nominal scales”, Educational and psychological measurement, Vol 20, No. 1, pp.37-46.

Cole, C., Parada, R.H. and Mackenzie, E., 2023. Why and How to Define Educational Video Games?. Games and Culture, p.15554120231183495.

Crathorne, P.J., 2010. Video game genres and their music (Doctoral dissertation, Stellenbosch: University of Stellenbosch).

Denden, M., Tlili, A., Essalmi, F. and Jemni, M., (2017) "Educational gamification based on personality", In 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), IEEE, October, pp. 1399-1405.

Denden, M., Tlili, A., Essalmi, F. and Jemni, M., 2018. Implicit modeling of learners' personalities in a game-based learning environment using their gaming behaviors. Smart Learning Environments, 5(1), pp.1-19.

Deterding, S., Andersen, M.M., Kiverstein, J. and Miller, M., 2022. Mastering uncertainty: A predictive processing account of enjoying uncertain success in video game play. Frontiers in psychology, 13, p.924953.

Epic Games. The most powerful real-time 3D creation tool, Unreal Engine. Available at: https://www.unrealengine.com/en-US/license (Accessed: 02 September 2023).

Franić, S., Borsboom, D., Dolan, C.V. and Boomsma, D.I., 2014. The big five personality traits: psychological entities or statistical constructs?. Behavior genetics, 44, pp.591-604.

Garcıa, J. and Fernández, F., 2015. A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research, 16(1), pp.1437-1480.

Ghahramani, Z., 2003. Unsupervised learning. In Summer school on machine learning (pp. 72-112). Berlin, Heidelberg: Springer Berlin Heidelberg.

Haas, J.K., 2014. A history of the unity game engine. Diss. Worcester Polytechnic Institute, 483(2014), p.484.

Hilliard, K. (n.d.). God Of War Ragnarök's Director Discovered He Had Aphantasia During Development. [online] Game Informer. Available at: https://www.gameinformer.com/interview/2022/11/09/god-of-war-ragnaroks-director-discovered-he-had-aphantasia-during-development [Accessed 31 Aug. 2023].

Ivory, J.D., 2015. A brief history of video games. The video game debate: Unravelling the physical, social, and psychological effects of digital games, pp.1-21.

Jadhav, S.D. and Channe, H.P., 2016. Comparative study of K-NN, naive Bayes and decision tree classification techniques. International Journal of Science and Research (IJSR), 5(1), pp.1842-1845.

John, O.P., Donahue, E.M. and Kentle, R.L., 1991. Big five inventory. Journal of Personality and Social Psychology.

Judd C. M. and McClelland, G. H. (1998) “Measurement”, The handbook of social psychology, Vol 1, pp. 180- 232.

Lee, J.H., Karlova, N., Clarke, R.I., Thornton, K. and Perti, A., 2014. Facet analysis of video game genres. IConference 2014 Proceedings.

Liu, B. and Liu, B., 2011. Supervised learning. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, pp.63-132.

Pavalache-Ilie, M. and Cocorada, S. (2014) “Interactions of Students’ Personality in the Online Learning Environment”, Procedia-Social and Behavioral Sciences, Vol 128, pp. 117-122.

McCrae, R. R., and John, O. P. (1992) “An introduction to the five-factor model and its applications”, Journal of personality, Vol 60, No. 2, pp. 175-215.

MIT Press. (2021) Before Pong, There Was Computer Space available at: https://thereader.mitpress.mit.edu/before-pong-there-was-computer-space/ (accessed at: 1 Aug 2023)

Prensky, M. (2006) Don't bother me, Mom, I'm learning!: How computer and video games are preparing your kids for 21st century success and how you can help!, St. Paul, MN: Paragon house

Rebala, G., Ravi, A., Churiwala, S., Rebala, G., Ravi, A. and Churiwala, S., 2019. Machine learning definition and basics. An introduction to machine learning, pp.1-17.

Ritterfeld, U. and Weber, R., 2006. Video games for entertainment and education. Playing video games: Motives, responses, and consequences, pp.399-413.

Shannon, C.E., 1948. A mathematical theory of communication. The Bell system technical journal, 27(3), pp.379-423.

Schunk, D. H., Pintrich, P. R., & Meece, J. L. (2008) Motivation in education: theory, research, and applications, (3rd ed.), Upper Saddle River, N.J.: Pearson/Merrill Prentice Hall.

Šmíd, A., 2017. Comparison of unity and unreal engine. Czech Technical University in Prague, pp.41-61.

Soto, C.J. and John, O.P., 2017. Short and extra-short forms of the Big Five Inventory–2: The BFI-2-S and BFI-2-XS. Journal of Research in Personality, 68, pp.69-81.

Squire, K., 2003. Video games in education. Int. J. Intell. Games & Simulation, 2(1), pp.49-62.

Toftedahl, M. and Engström, H., 2019. A taxonomy of game engines and the tools that drive the industry. In DiGRA 2019, The 12th Digital Games Research Association Conference, Kyoto, Japan, August, 6-10, 2019. Digital Games Research Association (DiGRA).

Unity Technologies Imaginecreategrowdo Moreimaginewith Unity, Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine. Available at: https://unity.com/pricing#plans-student-and-hobbyist (Accessed: 02 September 2023).

Van Hasselt, H.P., 2011. Insights in reinforcement learning. Hado van Hasselt.

Will You Snail?. (2022). PC [Game]. Steam: Jonas Tyroller

Watson, D. and Clark, L.A., (1997) “Extraversion and its positive emotional core”, In Handbook of personality psychology, Academic Press, pp. 767-793

Ypsilanti, A., Vivas, A.B., Räisänen, T., Viitala, M., Ijäs, T. and Ropes, D., 2014. Are serious video games something more than a game? A review on the effectiveness of serious games to facilitate intergenerational learning. Education and Information Technologies, 19, pp.515-529.

Annex A – BFI Questionnaire

BFI questions used. The questions used a Likert system [1-5]. The final score was the mean of all the answers. Questions with a "R" next to them have a reverse scoring. Meaning if the applicant choose rates the question with 1 then for the scoring counting we count the choice as a 5 and vice versa. The same applies for the air 2 and 4.

The questions start with the statement:

I see Myself as Someone Who:

BFI questions used for Openness:

- Is original, comes up with new ideas
- Is curious about many things
- Is ingenious, a deep thinker
- Has an active imagination
- Is inventive
- Values artistic, aesthetic expressions
- <R> Prefers work that is routine
- Likes to reflect, play with ideas
- <R> Has few artistic interests
- Is sophisticated in art, music, or literature

BFI questions used for Extrovert:

- Is talkative
- <R> Is reversed
- Is full of energy
- Generates a lot of enthusiasm
- <R> Tends to be quiet
- Has assertive personality
- <R> Is sometimes sh, inhibited
- Is outgoing, sociable

BFI questions used for Openness (Translated in Greek):

Βλέπω τον εαυτό μου ως κάποιον που (Greek translation):

- Είναι πρωτότυπος, επινοεί νέες ιδέες
- Είναι περίεργος για πολλά διαφορετικά πράγματα
- Είναι πολυμήχανος, βαθυστόχαστος
- Έχει ζωηρή φαντασία
- Είναι εφευρετικός
- Εκτιμά τις καλλιτεχνικές, καλαίσθητες εμπειρίες
- <R> Προτιμά δουλειά που είναι ρουτίνα
- Του αρέσει να αναλογίζεται/συλλογίζεται, να παίζει με ιδέες
- <R> Έχει λίγα καλλιτεχνικά ενδιαφέροντα
- Διαθέτει εκλεπτυσμένο γούστο στην τέχνη, τη μουσική ή τη λογοτεχνία

BFI questions used for Extrovert (Translated in Greek):

- Είναι ομιλητικός
- <R> Είναι επιφυλακτικός
- Είναι γεμάτος ενέργεια.
- Δημιουργεί πολύ ενθουσιασμό
- <R> Τείνει να είναι ήσυχος/πράος.
- Διαθέτει ισχυρή προσωπικότητα
- <R> Είναι μερικές φορές ντροπαλός/συνεσταλμένος, διστακτικός
- Είναι εξωστρεφής, κοινωνικός

Annex B - Code questions

In this part we have listed all the coding questions used in the game to test and support the students in learning the programming language C. The code questions are composed in three parts:

- First is the **question** that the "applicant" (NPC) had to answer.
- Second is the code that was given to the "applicant" (NPC) to see and answer based on the given question
- Third is the "applicant's" (NPC) answer.

The student had to examine these three aspects of the question and accept or reject the "applicant's" (NPC's) answer. The students didn't have to explain the reason they accepted or rejected the answer. If the student evaluated wrong the "applicant's" answer then a pop up text would explain why the applicant's answer is right or wrong.

The questions are grouped in days. Additionally next to the test number it is also labeled if the test is Correct (the student should accept the test) or wrong (the student should reject the test). Each day had a focus on testing a particular subject or category of the programming language C.

On the same day the student had to evaluate multiple tests.

Day 1

Focus test fields: {General knowledge of C, use of #define, setting value on variables}

1. ***Test 1.1 - Correct***

Question: What are the final values of a,b,c in the code below?

```
#include <stdio.h>
int main()
{
	int a,b,c=3;
	a=b=2;
	a=c+b;
	return 0;
}
```

Applicants Answer:

a-5

b-2

c-3

Error Message: Try read the values more careful

2. ***Test 1.2 - Correct***

Question: Is the code below correct?

```
#include <stdio.h>
#define MM 23
int main()
{
      int a,b;
      a=4+(b=2);
      b=c+b+MM;
      return 0;
}
```

Applicant's Answer:

Yes

Error Message: #include and #define don't need ";" at the end

3. ***Test 1.3 - Wrong***

Question: Is the code below correct?

```
#include <stdio.h>
#define MM 23;
#define c 3
int main()
{
    int a,b;
    a=2;
    float d;
    d=4.3;
    a=4+(b=2);
    c=c+b+MM;
    return 0;
}
```

Applicant's Answer

Yes

Error Message: The variable c is defined we can't redeclare it

4. ***Test 1.4 - Wrong***

Question: Is the code below correct?

```
#include <stdio.h>
#define ONE 12
#define TWO 78
main()
{
	int c=3,a,b;
	float c=5.6;
	b=ONE+TWO;
	a=printf("Hello word");
	printf("End");
	return 0;
}
```

Applicant's Answer

Yes

Error Message: The variable c is already declared as an int we can't change it to float

Day 2

Focus test fields: {order of actions in the same line, if, Math operations with float and int}

5. ***Test 2.1 - Correct***

Question: What will be the final values of a,b,aa,bb?

```
#include <stdio.h>
int main()
{
    int a,b,aa,bb,x,y;
    x=y=100;
    a=++x;
    b=y++;
    aa=++x;
    bb=y++;
    printf("The value of a is %d\n",a);
    printf("The value of b is %d\n",b);
    printf("The value of aa is %d\n",aa);
    printf("The value of bb is %d\n",bb);
    return 0;
```

```
}
```

Applicant's Answer:

a - 101

b - 100

aa - 102

bb - 101

Error Message: a = ++x means: first we add 1 to x and then we set a=x. So the value of a is 101. The opposite applies for b=y++

6. ***Test 2.2 - Wrong***

Question: What will be the printed values of the code below?

```
#include <stdio.h>
int main()
{
char ch,b='A';
      ch='A';
if (ch==b)
      printf("Yes-1 ");
else
      printf("No-1 ");
if ("A"== 'A')
      printf("Yes-2 ");
else
      printf("No-2 ");
      return 0;
}
```

Applicant's Answer

Yes-1 Yes-2

Error Message: The operator "A"=='A' is false [comparison pointer with integer]. Also, some 'C' compilers may not even run the program.

7. ***Test 2.3 - Wrong***

Question: What will be the final value of d.

```
#include <stdio.h>
int main()
{
    float d;
```

```
    int a=5,b=6;
    d=(a+b)/2;
    printf("%f\n",d);
    return 0;
}
```

Applicant's Answer

d - 5.5

Error Message: The result of (a+b)/2 = 5 is int as all the parts of the equation are int.

8. ***Test 2.4 - Correct***

Question: what will be the final values of f,d?

```
#include <stdio.h>
int main()
{
    float f,d;
    int k=5,l=6;
    f=5/2;
    d=5/10*100;
    printf("%f %f \n",f,d);
    return 0;
}
```

Applicant's Answer

f - 2

d - 0

Error Message: d=5/10*1000 -> 0*1000 -> 0 [because all the parts of the equation are int]

Day 3 - Applicant had to submit a code based on the question

Focus test fields: {For loops, If and subsets, understanding of problem and solution}

9. ***Test 3.1 - Wrong***

Question: Write a code to calculate the total price of the printed papers.

The price is 0.01$ per paper if the papers are < 10

The price is 0.2$ per paper if the papers are < 50

The price is 0.3$ per paper if the papers are < 150

and after 150 papers the price is 0.4$ per paper

Applicant's Answer:

```
#include <stdio.h>
#include <stdlib.h>
#include "simpio.h"
int main()
{
    int papers;
    float finalPay;
    papers =  GetInteger();
    if (papers<=10)
        finalPay = papers * 0.01;
    else if (papers<=150)
        finalPay = papers * 0.3;
    else if (papers<=50)
        finalPay = papers * 0.2;
    else
        finalPay = papers * 0.4;
    printf("The final cost is %f\n",finalPay);
    return 0;
}
```

Error Message: Be careful of the "if else". The condition if else(papers<=50) will never run.

10. ***Test 3.2 - Correct***

Question: Write a program to add all the numbers from 0-1000 (including 0 and 1000).

Applicant's Answer

```
#include <stdio.h>
int main()
{
    int a,sum;
    sum=0;
    for (a=0; a<=1000; a++)
        sum=sum+a;
    printf("The sum is %d\n",sum);
    return 0;
}
```

Error Message: The final condition for the "for loop" (a<=1000) is including "1000" to the sum

11. ***Test 3.3 - Wrong***

Question: Write a program to add all the numbers from 0-1000 (including 0 and 1000).

Applicant's Answer

```
#include <stdio.h>
int main()
{
    int a,sum;
    a=sum=0;
    while (a<=1000)
    {
        a++;
        sum=sum+a;
    }
    printf("The sum is %d\n",sum);
    return 0;
}
```

Error Message: Be careful we increase the "a" at the BEGINNING of the loop. This loop is the sum of all the number [1,1001]

Day 4

Focus test fields: {if - else statement, understanding of subsets, MAth operations with float and int }

12. ***Test 4.1 - Wrong***

Question: Write a code to calculate the booked score of a theater.

If the theater is 0-30% booked is "Low on bookings"

If the theater is 30-70% booked is "Normal on bookings"

if the theater is 70-100% booked is "High on bookings"

Applicant's Answer:

```
#include <stdio.h>
#include "simpio.h"
int main()
{
    int theater_size, bookings;
    float percentage;
```

```
    theater_size = GetInteger();
    bookings = GetInteger();
    percentage = bookings/ theater_size;
    printf("%f \n", percentage);
    if (percentage <= 0.30)
        printf("Low on bookings\n");
    else if (percentage <= 0.70)
        printf("Normal on bookings\n");
    else
        printf("High on bookings\n");
    return 0;
}
```

Error Message: The variables “theater_size” and “bookings” are int so percentage will always be ‘0’ or ‘1’ (because all part of the equation are int)

13. ***Test 4.2 - Correct***

Question: Write a code to calculate the booked score of a theater.

If the theater is 0-30% booked is “Low on bookings”

If the theater is 30-70% booked is “Normal on bookings”

if the theater is 70-100% booked is “High on bookings”

Applicant’s Answer:

```
#include <stdio.h>
#include "simpio.h"
int main()
{
    int theater_size, bookings;
    float percentage;
    theater_size = GetInteger();
    bookings = GetInteger();
    percentage = (float)bookings/ (float)theater_size;
    printf("%f \n", percentage);
    if (percentage <= 0.30)
        printf("Low on bookings\n");
    else if (percentage <= 0.70)
        printf("Normal on bookings\n");
    else
        printf("High on bookings\n");
    return 0;
```

```
}
```

Error Message: With the typecast “(float)” the numbers are considered as floats. So the result of the division “(float)bookings/ (float)theater_size;” will also be float

14. ***Test 4.3 - Wrong***

Question: Write a code to calculate the booked score of a theater.

If the theater is 0-30% booked is “Low on bookings”

If the theater is 30-70% booked is “Normal on bookings”

if the theater is 70-100% booked is “High on bookings”

Applicant’s Answer:

```
#include <stdio.h>
#include "simpio.h"
int main()
{
    int theater_size, bookings;
    float percentage;
    theater_size = GetInteger();
    bookings = GetInteger();
    percentage = (float)bookings/ (float)theater_size;
    printf("%f \n", percentage);
    if (percentage <= 0.30)
        printf("Low on bookings\n");
    else
        printf("High on bookings\n");
    else if (percentage <= 0.70)
        printf("Normal on bookings\n");
    return 0;
}
```

Error Message: You can't have "else if" after "else" in the same “if” structure

Day 5

Focus test fields: {understanding coding, char in C }

15. ***Test 5.1 - Wrong***

Question: What is the function of the code below?

```
#include <stdio.h>
```

```
int main()
{
    char ch;
    int a,fl=0;
    a=0;
    while (fl<2)
    {
        ch=getchar();
        if (ch=='*') fl++;
        if (fl==1 && ch!='*') ++a;
    }
    printf("%d Total characters: \n",a);
}
```

Applicant's Answer:

The program counts and print the total number of all the characters they where input excluding "*". The program will end at the second "*" character.

Error Message: The program counts the inputs AFTER the first "*".

16. ***Test 5.2 - Wrong***

Question: What is the final value of k?

```
#include <stdio.h>
int main()
{
    int i,j,k=0;
    for (i=0;i<=5;i=i+2)
    {
        k++;
        for (j=1;j<2;++j)
            k=k+2;
    }
    printf("k=%d\n",k);
    return 0;
}
```

Applicant's Answer:

I don't know, but please, can you let me pass? I know loops are my weak point, but I really need to do well on the test. I need this job.

Aristotelis

Error Message: You may have a big heart, but the applicant didn't answer the question correctly.

17. ***Test 5.3 - Correct***

Question: Is the code below correct?

```
#include <stdio.h>
#include <math.h>
int main()
{
    double fx,x;
    for (x=0.0;x<=1;x=x+0.05)
    {
        fx= pow(x,2);
        printf("The square of x=%f is fx=%f\n",x,fx);
    }
    return 0;
}
```

Applicant's Answer:

Yes

Error Message: You can use a float as a step in a for loop!

Day 6

Focus test fields: {General knowledge in C}

18. ***Test 6.1 - Correct***

Question: Is the following statement correct?

x = 5/2;

The value of x is 2.

Applicant's Answer:

Yes

Error Message: It is true as all the members of the division are int.

19. ***Test 6.2 - Wrong***

Question: Is the following statement correct?

The syntax "i++" increase the variable i by 1 that's not the case with "++i"

Applicant’s Answer

Yes

Error Message: The syntax “++i” also increase the variable i by 1 but in a different priority than “i++”

20. ***Test 6.3 - Correct***

Question: Is the following statement correct?

The size of one char variable is 1 byte.

Applicant’s Answer:

Yes

Error Message: The size of a char variable is 1 byte (or 8 bit)!

Day 7

Focus test fields: {General knowledge in C}

21. ***Test 7.1 - Wrong***

Question: Is the following statement correct?

The operators “++” and “--” can’t be used on variables type float.

Applicant’s Answer:

Yes

Error Message: The operators "++" and "--" can't be used on floats because they try to increase the number by "1" and not by "1.0"

22. ***Test 7.2 - Wrong***

Question: Is the following statement correct?

The result of the syntax <x= 1 + 1.0> is type double

Applicant’s Answer:

No, because we have an int in the syntax so the final result will be int.

Error Message: The applicant got it wrong. The final result will be double as there is a double number “1.0” in the syntax

23. ***Test 7.3 - Correct***

Question: Is the following statement correct?

In C there is no operator to calculate the power of a number.

Applicant’s Answer:

Yes

Error Message: C doesn’t have an operator for using power. Instead we use the function “pow(x,y)”

24. ***Test 7.4 - Wrong***

Question: Is the following statement correct?

The syntax x==5 is equal to x=5

Applicant’s Answer:

Yes as both have the same outcome.

Error Message: In one case (x==5) we check if the variable x is equal to 5 in the other case (x=5) we assign the value 5 to the variable x

Day 8

Focus test fields: {General knowledge in C}

25. ***Test 8.1 - Wrong***

Question: Is the following statement correct?

The loop while-do will always run at least one time.

Applicant’s Answer

Yes, and if the condition isn't met, it won't run a second time.

Error Message: The while-do first CHECKS the condition and then runs the code in the loop

26. ***Test 8.2 - Correct***

Question: Is the following statement correct?

The main function runs automatically when you run the program

Applicant’s Answer

Yes.

Error Message: You don't have to manually call the main

27. ***Test 8.3 - Wrong***

Question: Is the following statement correct?

You have to declare before main functions you define after main in order to use them.

Applicant’s Answer

No, because the functions are defined we don't need to declare them a second time.

Error Message: The applicant is wrong. You have to declare the functions before main for the program to be able to use them. This is called a "forward declaration."

28. ***Test 8.4 - Correct***

Question: Is the following statement correct?

The loop do-while will always run at least one time.

Applicant’s Answer

Yes, and if the condition isn't met it won't run a second time.

Error Message: The applicant is right. In do-while first we run the loop code once and then we check the condition.

Annex C - Conference Contribution Paper

The following paper has been accepted for presentation at the 17th European Conference on Games Based Learning (ECGBL) 2023 and it will be included in the conference proceedings.

Athanasios Tsionas, Maya Satratzemi, “Educational Story-Based Game for Capturing the Learner’s Personality”, 17th European Conference on Games Based Learning (ECGBL) 2023, 5 - 6 October 2023, Enschede, Netherlands.