



Adaptive Game Design Study of a fully adaptive framework

JOSÉ TIAGO DOS SANTOS SUZANO MAGALHÃES

Outubro de 2015

Adaptive Game Design

Study of a fully adaptive framework

José Tiago dos Santos Suzano Magalhães

**Dissertation to obtain Master's degree in Computer Engineering,
Specialization in Graphics Systems and Multimedia**

Supervisor: Filipe de Faria Pacheco, PhD

Jury:

President:

Vowels:

Porto, October 2015

Dedicatory

I want to dedicate this thesis to my mom and my dad, my brother, my sister-in-law and my niece, as well as my whole family who always supported me since I first started my master studies.

I wish to dedicate it to my best friend João Valente who has also been fundamental to my understanding of video games, and has helped me find the best path to developing this project.

And lastly, I want to dedicate it to my best friend and my colleague Joana Osório, who has helped me with both this thesis and throughout my whole college studies in more ways than I can count.

Resumo

Adventure! The Paladin Order foi um projecto ambicioso que começou por ser desenvolvido como um video jogo completo. Tinha como objectivo implementar uma ferramenta diferente que permitisse tornar o jogo completamente adaptativo às decisões do jogador tanto nas interacções e no diálogo com outras personagens, assim como no combate contra os variados inimigos do jogo. Devido à inexperiência do autor uma grande parte do tempo foi passado a estudar e a pesquisar várias possíveis soluções que permitissem criar um ambiente que fosse adaptativo de uma forma simples e interessante, não só para os programadores mas também para qualquer pessoa que fosse responsável por editar o diálogo e a história do jogo. Os resultados foram bastante interessantes, revelando um sistema que depende simultaneamente dos ficheiros de onde é retirado o diálogo, e de um sistema de personalidades que permite definir qual será o comportamento de qualquer objecto do jogo ou, pelo menos, como as outras personagens irão reagir. O produto final é uma ferramenta de bases sólidas que permite uma implementação relativamente simples de um sistema abrangente e adaptativo, com poucas falhas e apenas algumas questões de simplicidade de código.

Palavras-chave: video jogos, design adaptativo, sistema de diálogos baseado em ficheiros, sistema de personalidades

Abstract

This document details the study of an adaptive game design, through the development of a video game who pretends to implement just that. Adventure! The Paladin Order, which was initially planned to be developed as a full video game which had its emphasis on adaptive game design, had its purpose changed along the time of its development to a study and improvement of a framework that permitted a game to be developed as fully adaptive. This happened because of the inexperience of the author when developing video games, a lot of questions were presented when developing a solid base with a good portion of the time being spent on research and study of successful cases to inspire a solution that was easy to use and accessible to edit. Through this study a couple of games were researched for examples of adaptive design with trial and error resulting in the solutions found of the file-based dialogue, as well as the universal personality trait system which can be used for interaction with non-playable characters as well as for the combat system. In the end, despite the project being far from a complete game, it has its engine, its adaptive framework, developed and ready to be used with few sub-optimal issues such as minor bugs, and optimizations.

Keywords: Adaptive game design, dialogue system, personality traits, adaptive combat

Acknowledgements

A very thoughtful thank you goes to Joana Osório for her contribution and collaboration with Adventure! The Paladin Order for the development of social features, as well as the APO's testing team for all their help and dedication in guiding this project towards the most interesting future.

A huge thank you goes towards my teacher and supervisor Filipe Pacheco for always being there and providing countless advices and recommendations towards the better development of this thesis.

And lastly, a thank you to everyone else who in one way or another helped, contributed, and supported this project.

Index

1	Introduction	1
1.1	Objectives	1
1.2	Motivation	2
1.3	Document Structure	2
2	State of the Art	3
2.1	Current state of adaptive game design.....	4
2.2	Adaptive role play.....	4
2.2.1	Strategies used to adapt the story according to a player's story choices.....	4
2.2.2	Advantages and limitations.....	8
2.3	Adaptive game mechanics	9
2.3.1	Strategies used to adapt game mechanics according to a player's actions.....	10
2.3.2	Advantages and limitations.....	12
2.4	The turn-based genre.....	12
2.4.1	Successful cases	15
2.4.2	Adaptive turn-based roleplaying games	18
2.5	Technical tools and frameworks	19
2.5.1	Game Engines	19
2.5.2	3D Modelling and Animations.....	20
3	Adventure! The Paladin Order	23
3.1	Game Design	24
3.1.1	Initial approaches	25
3.1.2	Alpha testing	29
3.1.3	Current approach.....	33
3.2	Implementation.....	36
3.2.1	Dialogue system and personality traits.....	37
3.2.2	Modular implementation	43
3.2.3	Combat system	46
3.2.4	Visuals	50
3.2.5	Future development	58
3.3	Beta evaluation	59
3.3.1	Methodology.....	60
3.3.2	Results	61
3.3.3	Conclusions	64
4	Conclusions	66
4.1	Future work	66

Figure Index

Figure 1 – Dragon Age’s II dialogue wheel displaying an aggressive option.....	5
Figure 2 – Mass Effect’s dialogue wheel with different axis choices (blue Paragon, red Renegade)	6
Figure 3 – Fallout New Vega’s dialogue options displaying a Sneak check and a Speech check. 7	
Figure 4 – Pillars of Eternity dialogue system, detailing multiple dialogue checks that affect disposition and reputation.....	8
Figure 5 – Image generalizing the dialogue in Mass Effect showing how each answers always carries the same meaning	9
Figure 6 – A level in Dishonored with multiple ways to progress (notice how there are different entries to the same building)	10
Figure 7 – A fully built and customized board game for a D&D adventure	13
Figure 8 – D&D’s character sheet which allows specification of all the character’s details	14
Figure 9 – Magic the Gathering board full of creatures and lands	15
Figure 10 – A battle in the Pokémon Omega Ruby/Alpha Sapphire between two Pokémon ...	16
Figure 11 – Final Fantasy 7 combat which defined most of the series future combat systems	Error! Bookmark not defined.
Figure 12 – The world of Adventure! The Paladin Order	23
Figure 13 – Initial conceptualization of a character when attacking (Offensive Flowchart) and another where a character defends (Defensive Flowchart)	28
Figure 14 - Results from the Product Awareness chapter	30
Figure 15 – Results from the various quests presented to the players and the behaviours chosen	31
Figure 16 – Results for the first question in the game design related to game mechanics.....	32
Figure 17 - Results for the second question in the game design related to game mechanics ..	32
Figure 18 – Shop items are stored in a text file with name, item type, quantity and value in semi-colon separated values.....	34
Figure 19 – Main character being prompted to talk to the Knight	39
Figure 20 – Dialogue representation by their nodes	40
Figure 21 – Class diagram demonstrating how dialogue is structured in its classes	41
Figure 22 – Sequence diagram illustrating the interaction process when a character approaches an NPC	42
Figure 23 – Sequence diagram detailing how an event is called, started and ended.....	44
Figure 24 – CMBTPlayer and Enemy scripts inheritance from the Character script.....	46
Figure 25 – Sequence diagram explain the combat flow in a turn-based order.....	49
Figure 26 – Main menu screen of the game	51
Figure 27 – A character choosing the “Avatar” option on the Character Creator menu.....	51
Figure 28 – Overworld HUD presenting various kinds of information.....	52
Figure 29 – Upon clicking the Character button, the player is greeted with a panel of character information (Inventory and Journal panels are also accessible from here)	52

Figure 30 – Combat HUD which allows the player to keep track of various kinds of information as well as attacking the enemy	53
Figure 31 – Early concept of a character model.....	53
Figure 32 – Leonidas the main NPC of the game	54
Figure 33 – Comparison between Adventure! The Paladin Order’s art style (left) and QuickHunt’s art style (right)	54
Figure 34 – Link inspired concept art to be used as a reference sheet for 3D modelling the main character	55
Figure 35 – Main character on the game inspired in Link.....	55
Figure 36 – Concept art for a knight which was used initially as a reference sheet for the design of a character	56
Figure 37 – Leonidas with the main character in the world	56
Figure 38 – Main character with the personality Mages and the Ogre that represents the tutorial combat.....	57
Figure 39 – Overworld model design within the game in the same inspiring art style	57
Figure 40 – The combat scene (without user interface) also made in the same visual art style in a consistent matter	58
Figure 41 – Beta testing phase product awareness results	61
Figure 42 – Results for the questions related to the character creator.....	62
Figure 43 – Average rating of responses (note that numbers on the bottom correspond to each question presented previously).....	62
Figure 44 – Results from the responses of the overworld scenario questions.....	63
Figure 45 – Average results from questions of the same chapter (note that each number on the bottom corresponds to a question).....	63
Figure 46 – Results from the last chapter regarding the adaptive combat system	64
Figure 47 – Average of the results made for each one of the three questions regarding the combat system (note that each number on the bottom of the chart corresponds to one of the questions).....	64

Table Index

Table 1 – Comparison between the Unreal Engine and Unity3D game engines	20
Table 2 – Table describing different classes with their connotations, strengths, weaknesses and requirements to join.....	26

Notations

Notations Index

- AI** Artificial Intelligence
- APO** Adventure! The Paladin Order
- D&D** Dungeons and Dragons
- DA** Dragon Age
- DM** Dungeon Master
- DND** Dungeons and Dragons
- JSON** JavaScript Object Notation
- MTGO** Magic the Gathering: Online
- NPC** Non-playable Character

1 Introduction

Nowadays modern video gaming has become its own culture with millions of people playing and exploring new worlds every day. Thousands of games are being released every year making for a high amount offering of different genre of video games.[1] This way, games have become an important part in many people's lives, with countless ways to gain popularity and make money off of this industry both for companies and players as well with social platforms such as Twitch[2]. A key factor to understand is that when a product garners attention from big companies and starts being "mass produced" it is only natural that some qualities that are kept at heart when first being produced are lost. In this case, the author is referring to how games used to be more focused to the players creativity and what "glued" them to the game, what made them spend so many hours playing it.

Both this manuscript and the project are part of the dissertation of the Master's Degree in Computer Engineering, specialization of Multimedia and Graphic Systems of the Instituto Superior de Engenharia do Porto, and this document in particular serves to report the development process that was the "Adventure! The Paladin Order" (APO) experience. APO was developed as a game that focused as much on the player's decision-making as possible, keeping principles at heart such as the player's enjoyment and the belief that the way a person interacts with the game's world is what is most important. To satisfy these conditions, it is necessary to make the game with an environment that feels like it reacts to the user's decision and isn't just static or repetitive in each play through adventure even if the player takes the same decisions every time.

1.1 Objectives

Both the mission and objectives of this project were very clear, to answer one question which suggested a huge problem: to make an adaptive game which centred on the player's behaviour in a way that was simple and accessible for all parties involved. This requires a very delicate balance between world progression (how the game advances through the story) and granting control over the player. There are games who struggle with this issue, where the player has too

much control and is too strong or, on the other hand, the game seems to provide an experience that is completely unaffected by what the player wants or can do.

To provide a solution, a system that is yet accessible and simple to understand but strikes this delicate balance is the purpose of this dissertation. APO's game design was made to serve the industry with a system that anyone can understand and use, but at the same time provides environments where the player changes the world around him with his decisions and his presence.

1.2 Motivation

This dissertation's idea came from the author's experience with video games, where in some cases they felt like they were telling him a story rather than giving him some control of the situation (which is the case of Japanese role playing games for example), while in other cases the game was too liberal in terms of control and it felt as if there was no consistent story at all. To fill this need for a game that provided an experience that felt like the world progressed on its own but also changed with how the player interacted with it, was what motivated the author to develop this project. The plan was always to have a game that provided a world, a setting, an adventure, an experience that is able to exist on its own but also needs the player's presence, and how that player interacts with the world is how the world will react and change to it.

1.3 Document Structure

This report is divided in to 4 different chapters. Starting with this introduction chapter, follows the state of the art which attempts to explain where the project's system got its inspiration from, followed then by chapter 3 which explains how the game was initial designed, to what it was achieved in reality, and how the implementation was actually developed, while presenting results from both the alpha and beta test phases. The report ends with conclusions and future work.

2 State of the Art

In the current state of the gaming industry there are a lot of video games being produced every year all with different strategies and concepts.[1] Despite this, all games (like any other product) have one aspect in common: they want to harness the consumer's interest in order to increase their sales numbers, they are a business after all. When thinking about it this way, it is possible to argue that all games are user-oriented, because all games are developed to suit what their player base wants to play but, if we analyse these games carefully we come to realize that developers make decisions based on what the players want to buy resulting sometimes in rushed out and less optimal products[3]. This is identifiable in some games that have been criticized as too many[4] and too quick[5] in their series. These companies are, in practice, developing games with the consumer's interest in mind but always in a business perspective, with the main goal of producing something that is what people want to purchase. The games[6] are adaptive and user-oriented but not in a gameplay perspective, they are made to satisfy the player and retain him as a consumer.

Despite this, there are some cases in which developers have tried to create games that are user-oriented and adapt to what the user wants, but these cases usually lead to linear logic when it comes to the role play perspective. The player is presented with a situation and a closed group of decisions, whatever choice he takes will affect that specific situation and nothing (or close to nothing) in the revolving world changes to accommodate the results[7].

To create a game that feels like a living breathing environment, to create a small world that changes and reacts according to the player's decision is the goal for this project. This chapter aims to review the current state of the art in the gaming industry regarding user-oriented game design by analysing what exists in the industry and demonstrating two examples of this kind of game design when it comes to roleplay and mechanics and doing so justify the decisions made throughout the project.

2.1 Current state of adaptive game design

With the current state of the gaming industry there are a lot of examples of games that try to adapt and become as much as user centred as possible, this is noticeable in the open world adventure and the role playing[8] games genres. When it comes to role playing games, there is a clear distinction between western and eastern role playing games. Eastern games have the distinct characteristic in where the player is told a story instead of having control over it, he plays the main character but has no way to change the story of the game. Western games on the other hand, have the particular ability of giving the player a setting and the beginning of a story, but let him take control of how he wants to approach the game[9][10][11]. Games from the Super Nintendo and the Sega Genesis consoles were made in a time where rivalry between companies was extremely intense and thus a great effort was made to captivate the player as much as possible, either through a compelling story or through a challenging game, the player had a variety of games to choose from that allowed him to play the game in different ways.

To make a story of which the player's decision matters, of which the characters respond to certain actions, to make a combat system of which the enemies try their best to defeat the player instead of just repeatedly perform random moves every turn, these are the fundamentals of which this project will try achieve, but in a much smaller scale compared to the average commercial game.

2.2 Adaptive role play

Games such as the Dragon Age[12], Mass Effect[13] and Fallout[14] series use a very similar mechanism to determine the course of the story (albeit Fallout differs a bit more from the other two). The player is given a setting and takes the role of a customizable character, this character interacts with other non-playable characters (NPC) to progress through the story by listening to what these characters tell him and then prompting the player to choose from a set of closed given questions that normally have certain implications associated to it[15][16]. These game's story plot is usually influenced and decided through dialogue choices. Last, but certainly not least, there is the example of Pillars of Eternity,[17] a game which was released in and provides a sublime dialogue and combat inspired decision making, this game's concept is explored at the end of this first sub-chapter.

2.2.1 Strategies used to adapt the story according to a player's story choices

Both Dragon Age and Mass Effect have a moral system: Approval/Disapproval[18] and Paragon/Renegade[19] respectively (see figure 1 and 2). In both games each action has an icon associated to it to represent the "good" choice, the "neutral" choice, and the "bad" choice.



Figure 1 – Dragon Age’s II dialogue wheel displaying an aggressive option (extracted from [20])

In Dragon Age the player has a team that consists of two other members that have their own personalities, when the player takes a decision that goes according to those character’s personality’s traits that partner becomes more attached and friendly, approving their decision. By contrast if the player takes a decision that goes against a character’s personality’s traits that partner becomes less attached and more aggressive, disapproving their decision. This means that some party members, when having enough disapproval, will refuse to aid the player in their quests. But, if the player gains enough approval towards a character he is able to explore a special quest that is often related to that character’s background. Dialogue throughout the game changes and adapts to the player’s situation with their team[21].

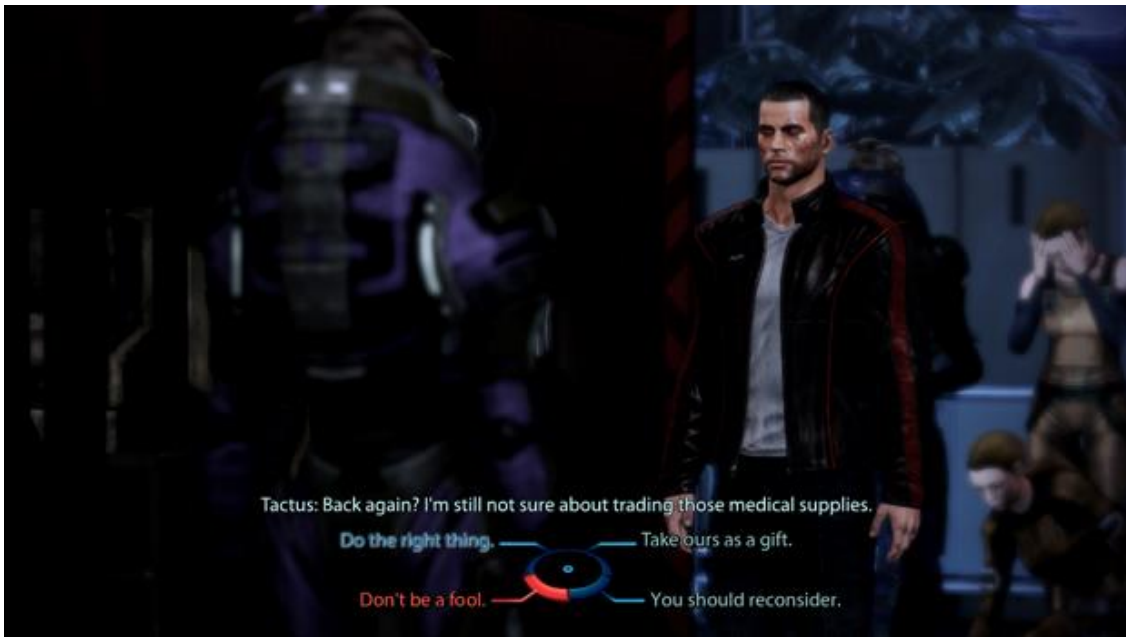


Figure 2 – Mass Effect's dialogue wheel with different axis choices (blue Paragon, red Renegade) (extracted from [22])

Mass Effect simplifies this logic even further, by giving the player a double edged meter that rises to either side depending on the decision the player took. As mentioned before each action has a moral value associated to it: when the player chooses good actions he earns Paragon points, increasing the meter towards the Paragon icon, if the player choose bad actions he earn Renegade points, increasing the meter towards the Renegade icon. This method of dialogue and moral heavily impacts the game by bottle-necking game story in to major plot choices that the player has to take. These dialogue choices vary greatly according to the player's meter, with some good options being available only when the player has a certain amount of Paragon points, and bad options being available only when the player has a certain amount of Renegade points. As with Dragon Age, some missions only become available when the character reaches a certain degree of Paragon or Renegade points. Finally, dialogue choices becomes extended when the character hits a very high value for either side of the moral system[23].

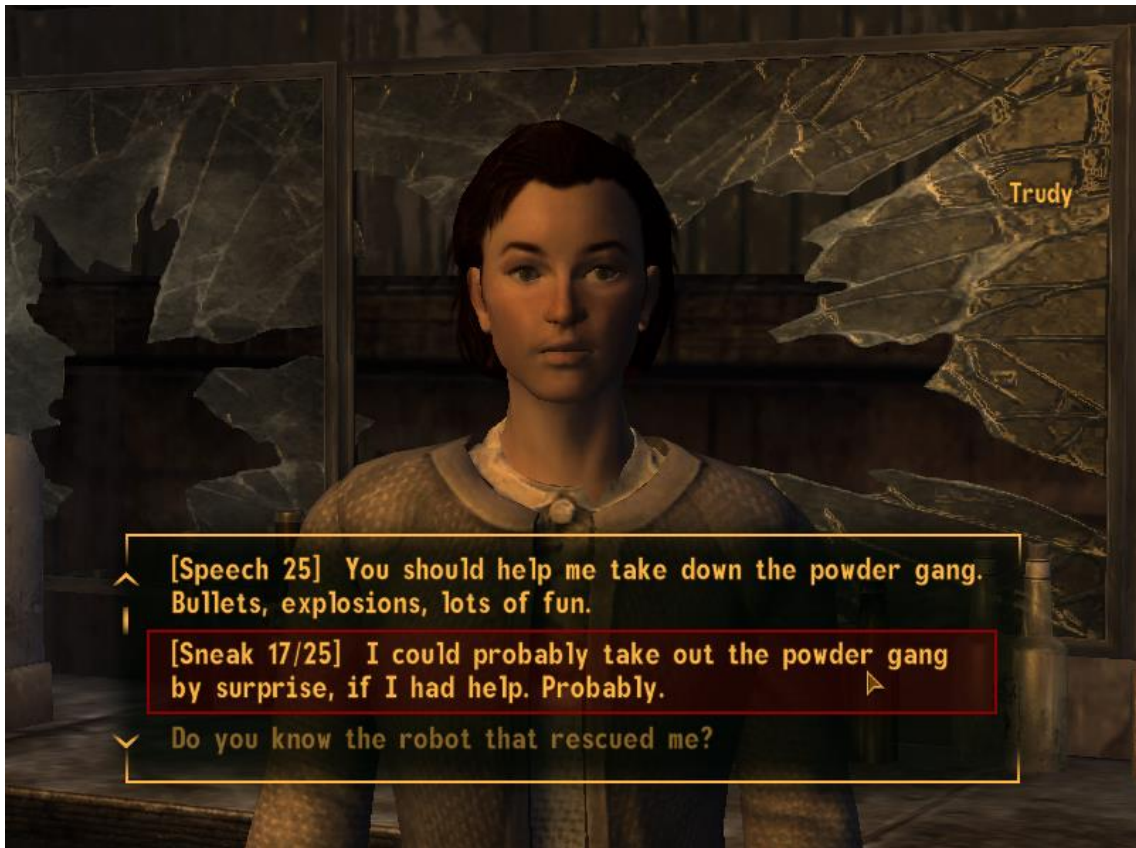


Figure 3 – Fallout New Vegas’s dialogue options displaying a Sneak check and a Speech check (extracted from [24])

In the case of Fallout: New Vegas, game choices centre heavily on the Charisma and Speech stats (see figure 3). As with the previous examples, the game features a dialogue with a set group of answers. These answers change heavily depending on the Charisma and Speech stat that the player’s character has. Perks, traits and Karma also affect dialogue and therefore decisions that the player can take. The simple Karma value represents the moral system, which increases with good actions and decreases with bad actions. Parallel to this approach, there are various factions throughout the game that the player can be part of with each faction having its own reputation, the more jobs the player does for the faction the more reputation he earns, with characters treating him accordingly. Decisions and choices in the game are made through dialogue that varies according to all conditions above but, not all answers work effectively all the time, the greater the Charisma/Speech/Karma/Reputation of the player the higher chance there is for the player to succeed in choosing an answer that is dependent on each one of those conditions, it is worth mentioning that there are answers that can be chosen which do not rely in any condition, similar to the neutral answers that exist in both previous games[25].

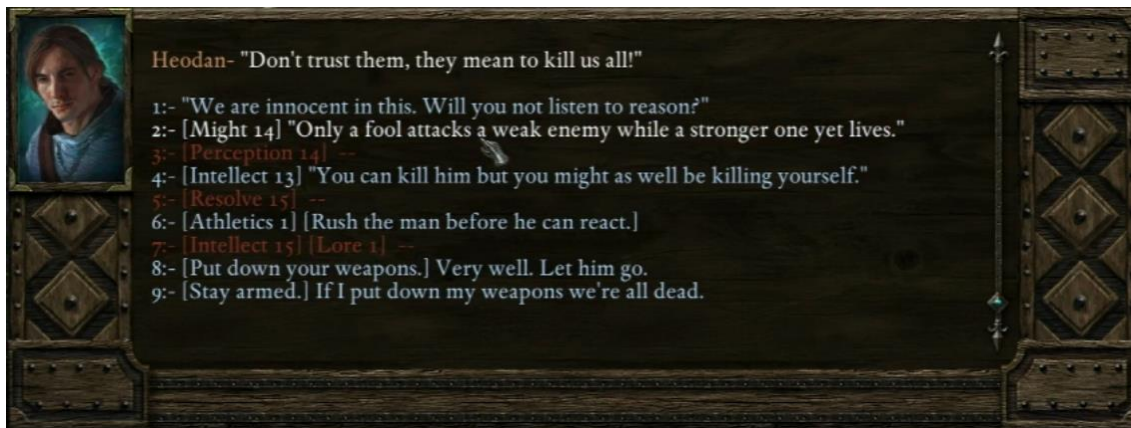


Figure 4 – Pillars of Eternity dialogue system, detailing multiple dialogue checks that affect disposition and reputation (extracted from [26])

Lastly there is the Disposition[27] and Reputation system created in the game Pillars of Eternity (see figure 4). The disposition system serves as kind of reputation system which changes how individual characters perceive and react to the player, based on their personality type. The player's disposition is tracked on a global scale based on their responses or ways of dealing with people. There is no absolute morality or judgement associated to these traits, a well behaved character can be placed in a bad situation just as much as a bad character would, depending on who they interact with at any given time. Personalities are also dynamic in the sense that a player known for being honest is able to develop a reputation of eerie stoicism. The reputations work more with the sense of factions and individual communities. There are many communities, factions and groups in the game's world the player can interact with and develop either positive or negative reputations with. These relationships is what defines how the player is perceived by other characters in the world. This system is conceptually what this project was looking for, the game provides a way both at a macro and micro level to change how the player character is perceived.

2.2.2 Advantages and limitations

All of the games above use a linear dialogue system, the player starts in point A in the story for a quest or a mission and, because he only has a limited group of answers to choose, can only go to point B, C, D or E. The advantage of this is, the player can truly feel immersed in the story as the answers are all hand written and can be adapted to serve each situation as the developers see fit. The more answers they present the player, the more situations, lore and choices they can provide as everything comes as a result to what the player has chosen as it all functions as a structured system. If seen at first sight, it looks to be a good system that provides enough freedom to the player without costing too much regarding game design.[28][29]

The big limitation exists in a form of bottleneaking and illusions: the gamer doesn't really have any freedom because, in the cases of Dragon Age and Mass Effect, only two answers really serve to progress through the story and all others serve to introduce more lore to the player creating

the illusion that we are in control and that we decide the fate of situations. In fact, in the case of Fallout: New Vegas, there is a greater incentive when it comes to spending points every time in Charisma to be able to unlock more dialogue options and access different answers. If we use the example presented before, which is the progression system that Mass Effect and Dragon Age use, there is a situation A with answers B, C, D and E where D and E only serve to serve lore information, B leads to a good decision and C to a bad decision (see figure 5).

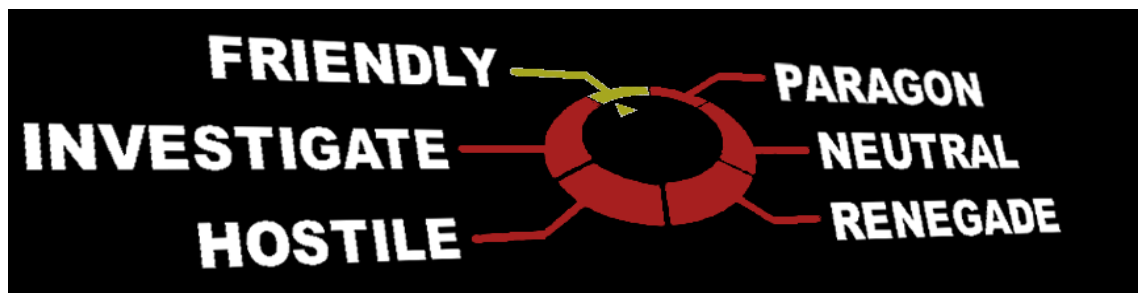


Figure 5 – Image generalizing the dialogue in Mass Effect showing how each answers always carries the same meaning (extracted from [30])

The player chooses either one and a fight ensues, or a destiny of a character is traced, the miner is saved and the criminal is caught, but all it had to be done was to choose the right set of answers to reach a predetermined ending. Mass Effect takes this to the extreme case where the very ending of the game is decided on the very last dialogue choices the player takes no matter what decisions were taken throughout the game.[31]

Although these examples provide a more specific control over what actions the player can perform in an open world genre they are, by the same logic, very limiting and therefore will not be used for this project, since the objective is to have the player presented with situations that he/she can choose to interact with, with the outcome of such situations being determined by choices made using game mechanics. By giving the player control with the game mechanics we are opening a different window of possibilities that allow for a better control of the story the game. The story is no longer limited to the choices the player makes, or to the reputation he upholds with a certain faction or group but instead it changes and adapts to each and every choice the player has taken. With this approach we guarantee that the world breathes and changes according to the player, because every action taken will be seen and will have the characters react around him. The original examples serve as a basis to what should be avoided in a game that provides more control to the player over its story. We are, however, aware that this will make a particular game creation more complex due to additional player alternatives and will do our best to make life easier for designers.

2.3 Adaptive game mechanics

In this chapter we will present the alternative approach to role player games in terms of how the player can interact with video games: their mechanics. Situations in video games where

player is put in control through their mechanics are extremely rare for a number of reasons, first of all it is rather complex to grant control over a situation and allow players to decide the destiny of a story through their actions instead of dialogue. It is possible, but often hard to achieve a balance between the narrative and the game mechanics[32]. On the other hand it is a much controverted topic whether games should focus on storytelling or mechanics.[33] For a game to have multiple decisions through actions the main story either has to be completely independent of the actions of the player, or the different actions he takes somehow lead to a node point in the story, or the game's story must be volatile enough to serve the different plot divergences. Regardless, there are a handful of games that have focused on putting the player in control not through what they say, but through their actions. And to emphasize what was referenced before, in most occasions, these actions do not seem to affect the world around the player no matter what he does.[34]

2.3.1 Strategies used to adapt game mechanics according to a player's actions

Dishonored[35] is a great example of how to create game mechanics that give the player complete freedom but also rely on those actions to advance through the plot. The game provides fixed levels, with fixed objectives but the story varies depending on how the player solves these levels. Each level provides more than one solution to be resolved with the missions, which almost all of the time involve sending the player to kill a character, always provide a choice of either killing or saving a character from their fate.



Figure 6 – A level in Dishonored with multiple ways to progress (notice how there are different entries to the same building) (extracted from [36])

A very good example of this happens in the mission where the player is sent to kill twin brothers of a main character of the plot[37]. To access the brothel in which the brothers reside numerous

entries are provided such as turning in to a rat and going through the front door, or teleporting in through an open window in the second floor (see figure 6). When the player reaches one of the targets he can perform one of two actions: either kill the twin or knock him unconscious and turn him over to a gang that, previous to the mission, has promised to take care of the twin for the player. The twist here consists in the fact that, should the player choose to leave the twin alive and turn him over to the gang, they “will cut out his tongue, shave his head, and lock him in one of his own mines”[38] presenting a moral choice that is solved by the player’s actions. When it comes to the game mechanics, the cherry on top is that the world becomes darker and more infested of disease the more the player kills enemies instead of neutralizing them, with some main characters rejecting the player’s own decisions. This applies to any character in the game. Needless to say, Dishonored provides an excellent balance of options for both situations, there are numerous ways the player is empowered and can pass through a mission undetected or through absolute destruction.[39]

The second example is the very first Fallout of the series[40], the game was released in 1997 and revolves around the player being born in a post-apocalyptic world, and his mission is to save the people from his vault by finding a new water filtration valve. This setting allows for the player to explore a completely new world from scratch: everything has been destroyed, it’s impossible to know what’s out there. The game is brilliant when it comes to mechanics, the genre for the game is an open world tactical role playing game and the gem that shines brightest is the fact that whatever the player does has consequences, which affects the communities that exist around him[41]. The game has too many examples to be called out, but the easiest to encounter is in the second town of the game called Junktown: when the player arrives he/she is prompted to help the community by investigating the activities of a local gang. While the player explores the town, he/she encounters a casino which the owner will ask to kill the leader of the town. Should the player refuse the casino owner proposal, he attacks the player and the natural reaction is to defend yourself and kill the casino owner. If so happens, the leader of the guards of the town banishes the player from Junktown for the act, as he cannot condone murder becoming impossible to return to the town without being attacked by the guards. It is worth mentioning that this isn’t told to the player: it’s a natural consequence that reflects a decision, which should be understood by the player when decides to confront the casino owner[42]. This is the perfect example of mechanics affecting the story and all of the world around the player. The player can choose to help the casino owner and tell the leader of the town about it, or can choose to run away from the fight if the casino owner attacks. There are numerous ways to approach the situation beyond those mentioned, and every character around the player changes his behaviour according to the choices of the player. Another interesting fact is that it is possible to equip and unequip weapons, and every townsfolk usually reproves of having weapons out in the town so if the player does not unequip his weapons, the townsfolk attack the player as a result. This happens to all communities alike creating this feeling of a living, breathing community around the player that reacts to one’s actions.

2.3.2 Advantages and limitations

Both games can be considered prime examples of adaptive game mechanics. Dishonored shines the brightest in the genre of games that have concise closed levels to progress through the game, while Fallout 1 provides the perfect example for an open world setting. As it has been mentioned throughout this thesis, the approaches these games have taken fall in to everything that you should expect from a game that is user-based and therefore adapts to his decisions. Dishonored's approach is much more closed in this sense, because the story is finite and there isn't much the player can do to change it[43][44][45]. The focus relies on each mission having multiple ways of being completed and finished, all depending on the player's course of action. Fallout 1 on the other hand, is perfect in this context. Because of its open world genre each village or community has its own side story that can change in multiple ways all depending on what the player chooses to do. The freedom the game provides is immense and the world vibrates and changes accordingly.

In a way, Dishonored is more limited because of how the game was made by levels, and this is good for people who aren't too keen in exploring and want to focus more in finding new, different, ways to complete a mission. On the other hand, Fallout's ability to provide so many different choices and outcomes for the player can be rather challenging to tackle, with so many ways to complete a given quest he might end up finishing it in a way he didn't intend to by making the wrong decisions at the wrong times. But this shouldn't be considered a limitation, Fallout is perfect when it comes to user-based adaptive gameplay, it is more of an acquired taste situation.

2.4 The turn-based genre

Turn-based games are notably one of the oldest examples in gaming history with games such as chess predating as much as sixth century India[46]. This genre is often called turn-based strategy and can be divided in to different sub genres. In the context of the project the relevant ones are: board games and turn-based tactics.

Many board games are turn-based as they typically involve having each player do different things in their own turn. In the context of this thesis the most noticeable example is, by far, Dungeons and Dragons (D&D) as it set the precedent to what many future virtual games would base their principals on.[47]



Figure 7 – A fully built and customized board game for a D&D adventure (extracted from [48])

D&D is a fantasy board game role-playing game which was first published in 1974 and is characterized by having a very well defined structure but at the same time allowing total freedom (within that structure) to the players. A group of people is often designated as a party and each player controls a character that can be anything between a lawful good hero to a chaotic evil character. This party is then placed in different situations that they themselves must come up with what solutions they wish to employ (see figure 7). To have some control over this situation, another person takes the role of the Dungeon Master (DM) which serves as a narrator but also a moderator and guide of sorts. A typical session of play revolves around a DM providing a storyline to the group and how they fit in it, followed by exchanges between these two parties, re-enacting how their characters would behave in those situations, which can either be solved through combat situations or by finding a mutual agreement through dialogue[49]. The key idea here is that, because it is the player that acts as his character, aspects such as alignments and backgrounds are absolutely customized to whatever the payer wants to do.

Figure 8 – D&D’s character sheet which allows specification of all the character’s details (extracted from [50])

It is also worth mentioning that everything within the game is decided by stats that a character hold, this allows for an even greater level of freedom because the player is able to do whatever he wishes to do but, whether the succeeds or fails at it, actually relies on the type of character he has made as well as in a bit of luck (see figure 8).

The reason why this game became such a big influence in future games of the genre is mostly thanks to the freedom of control it gives to the players. As mentioned before it is highly

structured in the way characters are created and the process of playing is one which allows for a much better approach programming-wise for companies to make a game of. The fact that at the same time the players are given the choice to do what they want allows any game creator to use this very well defined structure to make the basis for a game and provide the player with the story he/she wants. In a way, the game becomes the DM and the player controls a character which is part of a group.

2.4.1 Successful cases

Magic: The Gathering [51], was published back in 1993 and is still going strong nowadays, it features a gameplay surrounding deck building from physical cards that are released periodically through a process that is called expansions.



Figure 9 – Magic the Gathering board full of creatures and lands (extracted from [52])

Each expansion [53] holds its own set of cards, with each card having to be obtained by purchasing boosters (sealed package of cards that usually contains 15 random cards) or, with the upbringing of the internet, buying them online from other players. The game features different formats but the most popular ones involve the player choosing 60 cards or more of his own so he can build a deck and then play said deck against an opponent (see figure 9). A new version of the game was installed to accommodate modern society, a program called Magic: The Gathering Online (MTGO)[54] allows the players to play the very same physical version of the game in a computer with anyone in the world. MTGO became increasingly more important throughout the years.

Possibly one of the biggest gaming franchises and one of Nintendo's most successful games, Pokémon[55] stands as one of the oldest most successful game of all time having generated revenues of ¥4 trillion worldwide (equivalent to €29.4 billion euros) as of 2013[56].



Figure 10 – A battle in the Pokémon SoulSilver/HeartGold between two Pokémon (extracted from [57])

It was originally released in February 27, 1996 and, in its essence has remained very much the same through every iteration of the series. So far there are 24 games that are part of its main series and numerous spin offs. In its core, the world revolves around Pokémon: creatures that exist in the world very much like animals do in ours but, in this case, contain magical powers. The player is given the choice of one of three starters of three elements (fire, water, grass) at the beginning of the game, and the responsibility of catching each and every Pokémon to fill something that is referenced as a Pokédex as well as fighting gym leaders (a gym leader is the strongest trainer of that region) of each region to collect badges and be able to fight the Elite Four and the Champion. When the player fills the Pokédex having caught all of the Pokémon available and defeats the Champion of the game (the strongest trainer of all the game) he/she then becomes Pokémon master. It fits in this style of turn-based games as battles between trainers are done through a typical turn-based system, in which each Pokémon knows 4 different moves that the player chooses to attack the other Pokémon (see figure 10). One of the biggest aspects that has brought such a big success to the game is that each version of a Pokémon game has exclusive Pokémon, which encourages players to trade between themselves so as to have Pokémon that would otherwise be inaccessible to them. The game series has had such a tremendous success that has generated an anime series, a manga, various

animated films, soundtracks as well as creating its own version of a trading card game called Pokémon Trading Card Game.

And finally there is the Final Fantasy series[58], a game which follows different patterns compared to the previously mentioned games but has been great successful nonetheless. Made by Hironobu Sakaguchi its first release was in December 18, 1987, and now has 14 games in its main series with numerous spinoffs (spinoffs being games who are neither cannon, nor usually share the same mechanics), as well as games that feature side stories from the main series.



Figure 11 - Final Fantasy 7 combat which defined most of the series future combat systems (extracted from [59])

It is defined by how almost all of the games in the main series exist separately from one another, Final Fantasy II is not a sequel to Final Fantasy I but rather a whole different game with its own world that merely features identical elements such as, the combat system, items, and some characters (such as the Chocobos) that define the whole franchise. The series goes as far as having game mechanics change from game to game, for example the first Final Fantasy features a traditional turn-based system combat whereas Final Fantasy X has a full active time battle (which translates to a turn-based system combat in which characters wait a period of time to act, instead of waiting for a sequence of characters to perform their actions in their own turn which is the traditional turn-based) while in Final Fantasy 7 (see figure 11) it only has partial full active time battle (battles are still turn-based but some enemies wait for attacks to be performed first before acting). Story-wise the game, in a traditional eastern role playing game style, focus around a main character or a group of characters (commonly known as a party) battling a great evil as well as exploring the characters' internal struggles and relationships. The

series has been a great success so far, with more than 110 million units sold[60] it is considered one of the best-selling video game franchises of all time, having generated original video animations, films as well as an anime series.

2.4.2 Adaptive turn-based roleplaying games

Unfortunately the turn-based role playing games genre[61], which is the genre for the project of this thesis, has close to no examples of adaptive user-based game design. Turn-based games are most common in the eastern countries, and as mentioned at the beginning of this chapter, eastern game developers focus more in telling stories and putting the player in control of the main character, rather than empowering him to change the story as he sees fit. There are no examples of turn based games that give the player control over the story, but there is one situation in a particular game in which a combat, although very different of, comes close to what can be an adaptive turn based mechanic.

In the game Final Fantasy V[62] there is an enemy called the Famed Mimic Gogo[63] which, as the name implies, mimics the players action. In order to defeat him the player should understand that he cannot perform any action so as to mimic the Famed Mimic Gogo himself. If the player attacks with a physical attack, the Gogo will counter attack with a physical attack himself, if he attacks with a magical attack, the Gogo will counter attack with a magical attack himself. This example does not come close to an adaptive user oriented game design, but the fact that the enemy changes his attacks based on what the player does stands for a principle of what can be an adaptive turn-based mechanic.

Nevertheless, as discussed in previous chapters, Fallout 1 is a tactical turn based roleplaying game that provides both dialogue and combat inspired decision making, and even though the project's game will be a standard turn based game, it should be treated as the prime example to follow if the author wishes to accomplish the goal of making an adaptive user oriented game.

This projects mechanics and level design are based off these two games, always keeping in mind that levels should be made giving multiple options to the player but at the same time having those options impact the story and reflect in the world around them. These games will be used as a support to the technical implementations explained in further chapters.

2.5 Technical tools and frameworks

Given the current technological age we are in the era where there are multiple frameworks available for all different kinds of games, from 2D open source engines that are made specifically to be used for browser games[64] to high quality open source software[65]. With so many free to use programs, it is still important to realize which of those work best for each situation and why should that one be used instead of all the others. As such the next sub chapters will talk about the most important programs available, which were the most interesting in the scope of the project which ones were chosen, and why.

2.5.1 Game Engines

To develop this project it is necessary to analyse two different components: what game engine is going to be used to implement the game mechanics as well as what 3D modelling software is going to be for the models and animations. Starting with the game engines, there are a variety of engines for completely different contexts which makes choosing the right engine a difficult task. As such it is important to think what kind of game will the team develop, and then narrow which engines serve that purpose the best. Knowing that the game was meant to be a third-person turn-based roleplaying game, the context of available engines was greatly narrowed. Unreal Engine 4 and Unity3D[66] are both the most popular engines that serve this purpose the best, as well as being free of use which is one of the most important factors. As such, the decision of game engine had to be made between which of the two would provide the easiest solution. To justify which game engine was chosen, the following chapter provides a comparison between the potential engines, and explains which was chosen and why.

2.5.1.1 Unreal Engine 4 vs Unity3D

The Unreal Engine is a game engine developed by Epic Games[56][55][54]. Originally made for the game Unreal, it has been recently updated to a version name 'Unreal Engine 4' that boasts new features as well as a completely free-to-use model (with 5% royalty of sales going towards the owners of the engine).

On the other hand there is Unity3D, a cross-platform game engine developed by Unity Technologies initially announced only for Mac OS at Apple's Worldwide Developers Conference in 2005, available for free in Unity Personal as well as in a paid form name Unity Pro. The Unity Personal alternative is available for individuals or companies with less than USD 100,000 (around EUR 88,000) of annual gross revenue[67].

Both engines have a great set of pros and cons, in light of this table 1 present's comparable features from both engines, not to demonstrate which the best engine is but to show the pros and cons of both in different situations.

Table 1 – Comparison between the Unreal Engine and Unity3D game engines

	Unreal Engine 4	Unity3D
Licensing	Free of use, 5% royalty starts after the first USD 3,000 of revenue per product per quarter.	Free of use or USD 75 per month. Limitations apply to customers who earned or received more than USD 100,000 in revenue or funding in the previous fiscal year.
Current Version	Unreal Engine 4.7	Unity 5
Programming Languages	C++;	C#; JavaScript; Boo;
Compatibility	Runs on Windows, Mac OS, iOS, Android, Xbox One and Playstation 4.	Free version works on Windows, Mac OS and Linux but has some limitations in iOS, Android, BlackBerry 10 and Windows Phone 8.
Graphical Capabilities	Amazing graphical capabilities capable of being on par with games released on next-gen game consoles.	Unity has had some complaints regarding graphical capabilities, so much so that it suffered a graphical updated from version 4 to 5.
Ease of Use	Completely revamped editor with a “Blueprint” system that allows for quick level creation and prototyping.	Very user friendly interface as well as many tutorials to help quickly learn how to master the software.

When analysing both engines one quickly comes to the conclusion that neither is objectively better than the other for the generic situation, the choice of game engine relies then on what game the developers want to build, which programming languages they are the most comfortable with and which software they personally prefer. Given the context of this project, because the game was meant as a proof-of-concept prototype and the developers were most comfortable programming in C# and Javascript, Unity3D was chosen as the game engine. Another definitive feature that helped with the decision was that Unity3D provided the quickest way to setup a base for the game allowing the team to shift their focus on what they need to build for project in the least amount of time possible.

2.5.2 3D Modelling and Animations

The other component necessary for the success of the project was to determine the most optimal software to be used for both modelling and animations. It is important to note that this is not the area of expertise of either of the author which factors in the decision of what program to use. When it comes to 3D modelling and creating animations there are numerous tools available that help perform the task with plenty of tools at the artist’s dispose. The most popular programs[68] are Autodesk 3ds Max[69], Autodesk Maya[70] and Blender[71]. It is important that the piece of software chosen is amongst the most popular because, as it was mentioned

before, the team members lack experience in both modelling and animations so it is vital that the software is backed up by a good amount of tutorials and guides that the author could fall back on to so as to be able to learn how to model the characters necessary for the game. This chapter is dedicated to explore a little bit of each program in an attempt to justify which one was picked over the others and why.

2.5.2.1 Autodesk 3ds Max

Autodesk 3ds Max, previously known as 3D Studio Max, is a professional 3D computer graphics program developed and produced by Autodesk Media and Entertainment. The original version of 3D Studio Max product was created for the DOS platform which provides a lot of history and reputation to this program.

The current version of the program features: a built-in scripting language for automate repetitive tasks, a Character Studio that provides support towards character animation, a Scene Explorer with a hierarchical view of scene data and analysis, importation of DWG files, texture assignment and editing, skinning, skeletons and inverse kinematics and finally an integrated cloth solver. Autodesk 3ds Max is a very professional program with numerous support that is continually added to extend the software's features, unfortunately the licensing for the program comes at a 123 dollars a month or in an one time license purchase of 3,675 dollars. There is also a 30 days free trial that anyone can use. Given the licensing costs for the program and the fact that it would be necessary to use it for an extended period of time, this program could not be chosen for this thesis's project.

2.5.2.2 Autodesk Maya

Autodesk Maya[70] is also a 3D computer graphics software that is able to run on Windows 7, OS X 10.8, RHEL 6.2, Fedora 14 and CentOS 6.2. Originally developed by Alias Systems Corporation it is now currently owned and developed by Autodesk, Inc., it was previously was available in both "Complete" and "Unlimited" editions until August 2008, when it became a single suite.

In Maya users create a virtual workspace (called scene) where they can integrate and edit media of a particular object, these objects are then scene elements which belong to a node graph architecture having its own attributes and customization that the user can take advantage of. In licensing terms, users who are students, teachers can download a full educational version from the Autodesk Education Community albeit these versions can only be licensed for non-commercial use, with some products creating watermarks effects on output renders. Maya boasts several features such as: a fluid liquid simulator, a dynamic cloth simulation tool set, a fur simulator designed for large area coverage of short hairs and hair-like materials, a hair simulator capable of simulating dynamic forces acting on long hair and per-hair collisions, composite support, a camera sequencer which is used to layout multiple camera shots and manage them in one animation sequence and finally an embedded language with a cross-platform scripting language. Even though Autodesk Maya has an educational program, it would require confirmation from ISEP in the first place, which is a problem that takes time to be solved as well as the watermark issue. The licensing itself is 185 dollars per month with a one-time

license of 3,675 dollars. For the same reasons of Autodesk 3ds Max, Autodesk Maya was not chosen because of the licensing prices as well as the problems associated with the educational version.

2.5.2.3 Blender

And finally there is Blender[71] who is also a professional 3D computer graphics software product that works on Windows, Mac OS X and Linux. Originally made by a Dutch animation studio called Neo Geo and Not a Number Technologies on July 18, 2002 in response to the bankruptcy from the studios a “Free Blender” campaign was started which focused on garnering 100,000 euros from the community at the exchange of releasing the program as open source. As an added note, due to the fact that Blender is open source, other programs have tried to repack it and selling cosmetically modified versions of it, examples of these include IllusionMage, 3DMofun and Fluid Designer.

Blender boasts features such as: support for a variety of geometric primitives, an internal render engine with scanline ray tracing, a path tracer render engine called Cycles, integration with a number of external render engines through plugins, key framed animations (including inverse kinematics), soft body dynamics simulation, fluid simulation, smoke simulation, bullet rigid body dynamics, ocean generator with waves, a particle system with support for particle-based hair, the Blender Game Engine (a sub-project which offers interactivity features such as collision detection, dynamics engine and programmable logic), a fully integrated node-based compositor, procedural and node-based textures, and finally real-time control during physics simulation and rendering. One of the key reasons Blender was chosen as the program of election for this project was that it is open source and free of use, with the added fact that there are a multitude of tutorials and guides available to help the process as well as having a very complete and stable set of features that allow for more than what the team needs.

3 Adventure! The Paladin Order

Adventure! The Paladin Order (see figure 12) began as a project whose purpose was to explore a different aspect in the RPG genre and to provide enlightenment to the writer on how to approach game development in an adaptive way. The whole project started as a game to be completed in approximately over half a month, providing a different experience compared to other video games in the sense that the whole game was to be made completely focused on adaptability and depending on the actions of the player. Taking from the principal of, every action has a reaction, using real life as a comparison since actions in real life have almost ways some repercussion too, the game was to provide an experience where every action that was given possibility for the player to take would affect and change the whole environment that surrounded the player.



Figure 12 – The world of Adventure! The Paladin Order

As such, this chapter is dedicated to explain the evolutionary process that went from designing a video game that was highly focused on adaptiveness to studying and developing a framework that allows for a generic implementation of an adaptive overworld and even a combat system.

The first sub-chapter serves to present the initial ideas that were chosen for the game, the initial game design concept, how the first alpha test phase with the tester groups was done based on this initial concept, what were the results and how they changed the concept, finishing with how the game design was theorized initially and how different testing and experiences changed the whole perspective of the issue.

The second sub-chapter details the implementation of the actual project as it stands, explaining in detail how the dialogue system was developed emphasizing its modular implementation and how it sets the base for the rest of the game. The modular implementation is also expanded in the next chapter, while other decisions in different concepts such as the 3D models, the art and the combat system are explained as well.

3.1 Game Design

The game design of this project was initially thought of as approaching a real completed video game (or at the very least a demo), that could be used and referenced as an example of what an adaptive game design is. To even begin conceptualizing such a project it would be necessary to specify what an adaptive game design consists of as different interpretations of it can be understood. As such, an adaptive game design by the interpretation of the author is defined as: a game design focused on the player's decision-making and behaviour, a design that allows him/her to take actions which create reactions by the surrounding environment.

To achieve this effect, the initial concept was to develop a 3D video game that created a small environment of what could possibly be a fully adaptive experience for any kind of player whether casual or hard core. The best genre that allows for this effect would be the role playing game genre, with an open world to provide a more immerse experience to the player. With this in mind, due to the inexperience of the author when developing video games, to time constraints and to a lot of trial and error experiences, the project's focus was changed. So instead of developing a fully playable 3D video game the objective shifted to studying what is the best possible architecture for a fully adaptive experience as well as how to implement such architecture in a way that it becomes easy for developers to implement in their own video games. Through a big learning process that spanned over 8 months, a lot of trial-and-error and experimenting, an architectural framework for an adaptive game design was developed allowing a game to feel like a breathing environment that surrounds the player.

This chapter explains in detail what was initially thought of for the completely playable video game experience, how the alpha tests were made with this ideal in mind and then expands on the concept of the adaptive design that was reached after changing the focus on the project from a video game, to a study of an architectural foundation.

3.1.1 Initial approaches

Initially the project consisted of a 3D video game in an open world perspective inspired by games such as Fallout, The Elder Scrolls[72] series, the Final Fantasy series and many others. Taking these as an example the idea consisted of making a game that provided a short scenario in which the player could interact with various non-playable characters (NPC) and experience different changes in the surrounding world driven by his actions.

The story concept for the game was of a paladin order, a group of warriors from various parts of the world that serve a greater good and upheld justice, which holds a castle in a valley and is one of the most important trading centres in this world. This meant that the castle grounds held a lot of different characters of varied backgrounds such as merchants that had come to perform their trades and mercenaries that came to provide their services for whoever needed. This paladin order was in charge of maintaining order of the surrounding world which also meant that a lot of people, whether villagers or kings, would come there to ask for the assistance of the paladins. This kind of setting provided countless interactions for the player in numerous ways, allowing for a very expanding creative when it comes to creating adventures for the player. The plot twist in the game's story was that the paladin's order leader was corrupt which the player would come to realize more and more the more he/she progressed through the story by certain giveaways such as characters being punished unnecessarily or other characters hinting at how the leader was performing shady activities with evil creatures. A critical point in the game story was to never provide a situation where the player had no control over whatsoever (for example, a character would be killed in the same way regardless of what the player would choose to do, at the very least in this situation he/she should be able to control how the character would die and that could have some future repercussions). The player would then be inserted in this world as a vagabond, an adventurer who was found and retrieved in the surrounding forest with no clue to what their background was or what kind of person he/she is. To spare this human, the paladin order decided to test his abilities and should he/she succeed in the quests provided, the order would introduce him in their ranks.

The game design itself had some concept design that was envisioned to compliment both this story setting as well as to fit the adaptive game design. Some concepts consisted of:

- The player should not have too much health and fights should be relatively short to emphasize the importance of the player's actions in combat;
- The result of the combat should somehow affect the world out of it (for example provide a way to not kill the enemy, or if the character was killed in a certain way (more brutal, less brutal) different characters would react in different ways);
- The combat itself should provide a wide array of options to the player and these options would then be reflected in the enemy's retribution;

Allied to these generic concepts, both a levelling system and a class specialization were conceptualized to create another level of depth to the experience. Initially the character was an Adventurer, a generic class with no specialization in any kind of fighting, this made it so which quests the player initially decided to take, and how he would fulfil them, would shape their

character and create a path for their specialization. For example, there was a merchant who was in need for the player to fetch certain items from the nearby forest, multiple ways would be provided to solve this quest namely: kill the merchant and take his money, fetch the items from the forest as intended, or buy counterfeit of less quality items from another cheaper merchant in the grounds. These 3 solutions would shape what the character would be: the first one is a more aggressive solution, while the second one is more just and benevolent, whereas the third one is more cunning and selfish, each solution has a connotation to it and could be associated to a class specialization, such as: violent posture then warrior class, just posture then healer class, cunning posture then rogue class.

Of course, class specialization should be something that would be chosen after a long process of shaping the character, if by performing one action the player’s class is immediately defined than the purpose of shaping the character’s personality is defeated. This specialization would be given and trained by a NPC (or group of NPCs) that had such a connotation associated to that specialization, after the player had met the necessary requirements. Table 2 describes the available classes in the game, what was the connotation associated to them, their strengths and weaknesses as well as what was the requirements necessary to be able to achieve them. Requirements refers to what posture or behaviour the player character should have in any given quest that would fulfil these requirements and then accept such specialization.

Table 2 – Table describing different classes with their connotations, strengths, weaknesses and requirements to join (part 1)

Name	Connotation	Strengths	Weaknesses	Requirements
Warrior	Represents strength and physical force. The leader would most likely be a mercenary indoctrinated in to the paladin order.	Increases the strength of the player and the use heavier armour and weapons.	Makes the player weigh more, less suitable for more nimble athletic situations.	Strength, superiority or any other show of force.
Mage	Represents intelligence, smart thinking, and patience. The leader would most likely be a smart wizard that served as a counsellor for the order’s leader.	Increases the magical prowess of the player, allowing him/her to wield magical weapons, items and equipment more effectively.	Very weak class, not suitable for taking too much or close quarter combats with regular weaponry.	Intelligence, smarts, magical tricks, and an abundant use of magical items or spells.

Table 3 – Table describing different classes with their connotations, strengths, weaknesses and requirements to join (part 2)

Name	Connotation	Strengths	Weaknesses	Requirements
Rogue	Represents cunning actions, light selfishness, social behaviour, gain and improvement of one self through respectable actions within the society. The leader would be a socially known character that performed either side jobs or public jobs for the order.	Allows the player to perform more social options, increases the success of how social activities happen, more dialogue focused gameplay with some dexterity based combat.	Less combat, more dialogue, combat experience is stifled not so emphasized and more focused on what the dialogue options were taken previously.	Banter, social skills, solving problems through dialogue and passive conversation.
Assassin	Represents death, combat, always in a way that the person in question is never found of its deed, pure selfishness and improvement of ones situations regardless the cost. The leader would be a mysterious dark entity only accessible by some under extreme circumstances.	Great focus on combat, extremely focused on producing high outputs of damage and defeating (most of the time killing) enemies as effectively as possible.	Less focus on dialogue, more focused on following orders and granting a more incredible, complex, combat experience.	Killing characters, selfishness, greed.

Associated to the levelling and class system was the questing system which, in a very simplified manner consisted of providing quests which impacted nearby characters depending on how the character would solve them. These quests could either impact the characters locally, globally or specific characters in specific ways. This would be done by modifying the personality values of the different characters depending on the localization of the quest for example, if the quest was local a list of characters would be associated to this “local” connotation and when the player completed such quest violently the characters would behave in a way that would suit this violent behaviour.

Finally, the combat system was conceptualized in a different way that ended up being implemented in its reality. There were three different aspects that influenced how combat would be determined. The equipment the characters had would define which attacks they could perform (for example, spears would allow to thrust while swords allowed for swing types of attacks), each character had resistances both to types of attack (referred before as thrusts and swings) as well as elemental resistances to magical attacks that would be determined either by

the armour they wore or their nature (or both) and their character specialization, which would allow for a different use of skills and spells according to it. The combat was initially thought of as a turn based combat with no active turns, the player would choose three actions that would represent three different actions his turn and the enemy would choose another group of three actions to react to such actions. The same way worked the other way around, the enemy would choose actions based on the players behaviour and personality and the player would choose actions to contradict this attack without actually knowing what the enemy would perform (see figure 13). This raised a big problem because how was the player supposed to know which actions to use as defence when he couldn't possibly know what the enemy was going to choose?

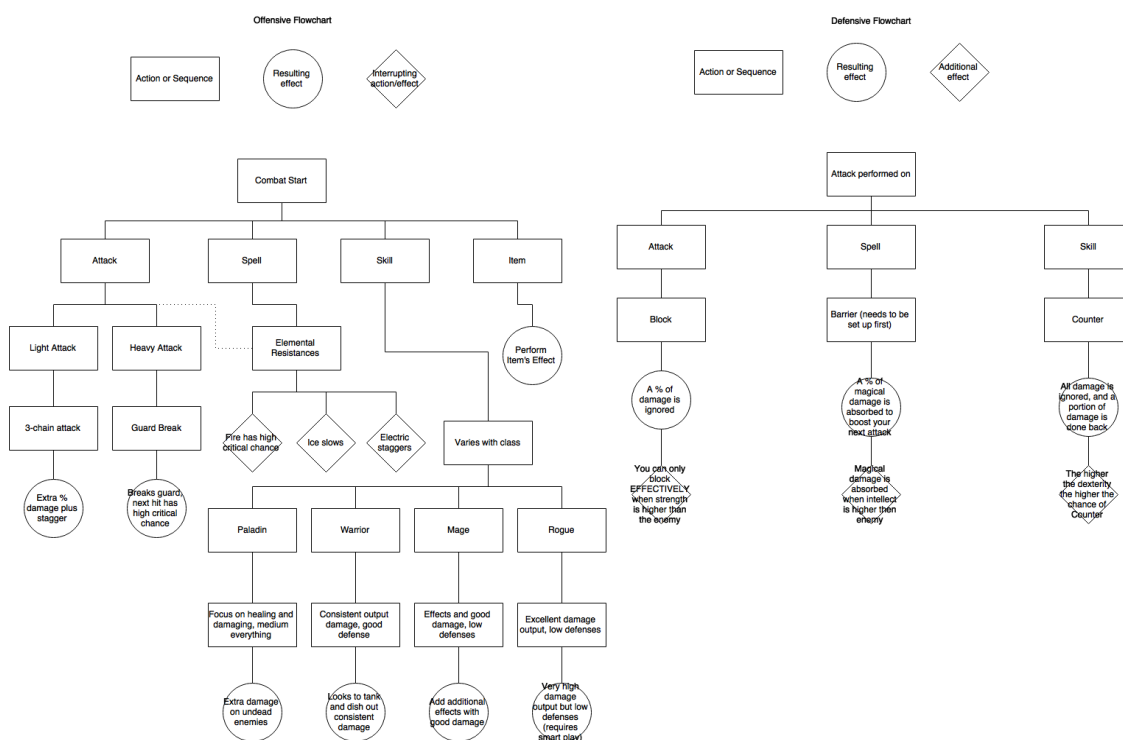


Figure 13 – Initial conceptualization of a character when attacking (Offensive Flowchart) and another where a character defends (Defensive Flowchart)

Such problem was left unsolved as the combat system was later shifted in to a different concept. The enemy's artificial intelligence (AI), however, would work as for each time the player character would take an action the enemy would record this action and "learn" more about what types of attacks the player performs and how often it does. Based off this learning process, each passing turn the enemy would more and more chose actions that would more efficiently predict what the player was going to use afterwards. To emphasize this aspect each skill the player had could somehow perform some combination of power with other skills (for example if the player choose that his first action is an attack that breaks the enemy's arm or, the next attack would do bonus damage, creating a predictable pattern). Another aspect the AI could predict attacks from was: if the enemy had a particular weakness associated to it then, more often than not, the enemy would try to protect itself from such weakness.

3.1.2 Alpha testing

The alpha testing phase for the project was done during the time where the author was still studying the best way to develop an adaptive game design as such, the testing phase was used to provide game design scenarios and try to understand which direction the game should take. The process began by writing a pdf file in form version (Annex A) that any person could receive, fill and send to the teams' personal email address. This process resulted in gathering a "testing team" of roughly ten testers, between the ages of 19 and 25. After having the team gathered, the alpha testing process began by mass emailing the team a Google Form survey (Annex B) whose purpose was to determine which sense and direction should the game design of the game take, which one would be the most appealing to the general audience as well as gathering some product awareness and feedback for future work.

Regarding game design the purpose of the test was to assure given the context and story of the game which direction quests in the world should take, what actions players would like to take, and what results should they bring. Combat was also explored by presenting ideas to the players and ask them which scenario they would rather be in. In general, the testing process was used to provide ideas to the players, and then use them to filter which ones were good and feasible and which ones were not entirely interesting.

The form began by talking about product awareness to measure how interesting was the concept of the project for the testers team (see figure 14). The first question "Was the document clear enough to understand everything about the game?" had the objective of determining if the initial form (the one who was used to perform the invitations) was clear enough to understand the whole purpose and concept of the project as a whole. The second question "How likely are you to recommend this game to your friends or colleagues?" was to determine if the idea was interesting to the point it could be a reliable game that people would be eager to share. The third and last question of product awareness "Do you think the game is distinguishable and different enough from all the other games in the market?" meant to measure how innovative the project's concept was.

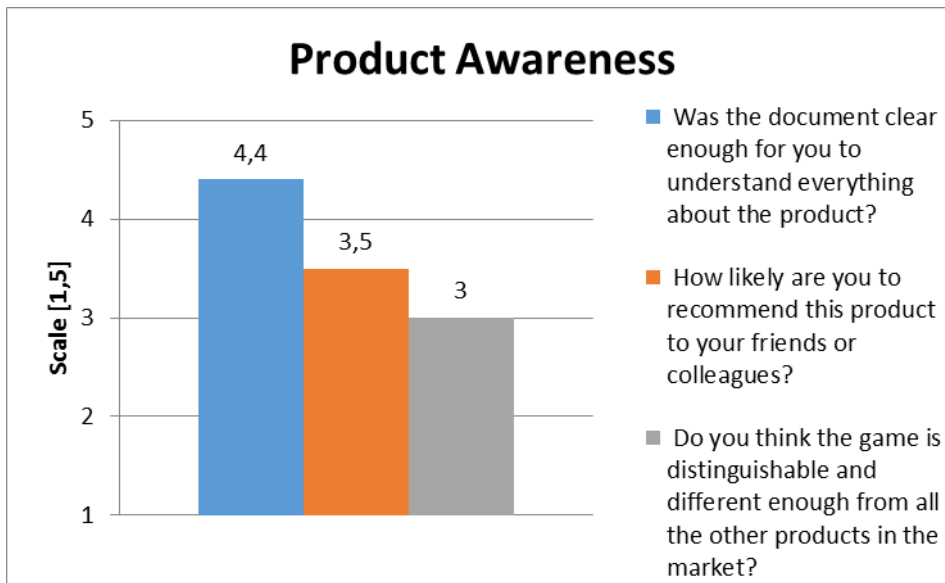


Figure 14 - Results from the Product Awareness chapter

As the graphic displays most users completely understood what the projects concept was (as the invitational form explained everything about it), demonstrating that the invitational form was clear and simple to understand. All testers provided above average responses when it came to sharing the game with their friends, from this we can conclude that the game holds some potential as a popular project. Finally, the last question which meant to determine how innovative the project's idea was had mixed responses with an average rating in the end, with this it is possible to conclude that some ideas regarding the project had to be revised if one of the key focus would be to deliver a very innovative and different idea in to the video gaming market.

Leaving the product awareness behind the form enters the game design chapter related to the story, in this chapter several quests are presented to the team with the objective of determining which solutions they would most like to see implemented in the game (see figure 15). Every answer can be grouped in to different behaviours these being: violent, cunning, diplomatic and explorative. The results were gathered as such:

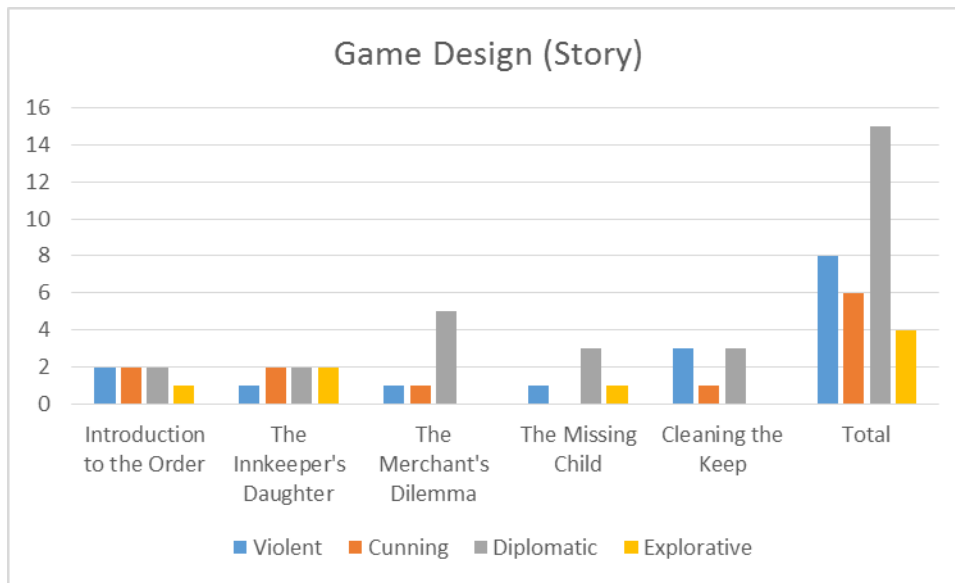


Figure 15 – Results from the various quests presented to the players and the behaviours chosen

Each quest was treated as an open question allowing the testers to describe how they would handle each situation individually. For the sake of this thesis and to simplify the results, each question could be translated to a behaviour, the four behaviours displayed on the graphic encompass all solution presented. Each point in violent corresponds to each time a tester chosen an aggressive option such as fighting a character, each point in cunning corresponds to each time a tester chose a less obvious way to handle a situation by stealing from a character or sneaking without being detected, each point in diplomatic corresponds to each time a tester decided to talk to a character or enter a discussion to solve a given problem without the use of any other means, each point in explorative corresponds to each time a tester chose to explore all available (or multiple ways) to solve a given quest before deciding how to solve a given problem. From the results gathered it's possible to understand that each quest had different responses from the user but in a general way the diplomatic solution was the one who was most preferred from the testing group. It is then possible to conclude that the testing group was more inclined to find a passive, nonviolent solution to each quest given. This meant that the project in itself should focus more on providing solutions that depended more on discussions and diplomatic interactions, followed by a wide range of violent solutions, and finally compliment the rest with cunning and explorative solutions.

The final chapter in the form was related to how the game mechanics should be made. No question went in to any specific detail of which skills should be implemented or anything technical, they focused primarily in raising questions such as: the team wants to make a game which the combat is focused on the player's decisions, of these possible implemented scenarios which ones would you like to see the most in the game? This is the key aspect of this group of questions that the testers took the liberty to reply with their own opinions. The first question attempted to understand that, given the context of the project, would the testers be more

interested in facing one difficult enemy that adapts to their choices or multiple enemies that share a weakness to be discovered? Figure 16 displays the results of the question:

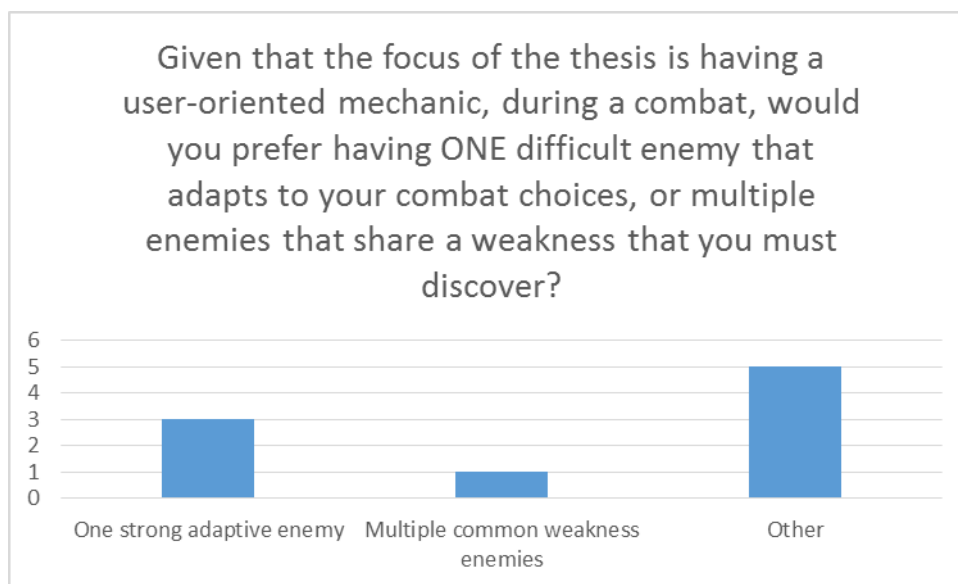


Figure 16 – Results for the first question in the game design related to game mechanics

Looking at the graphic it is possible to tell most of the testers decided for an “Other” this option in all cases referred it was used to specify that they wished for both options to exist, one tester went even further to explain the bosses should be one strong adaptive enemy and lesser enemies should be multiple with common weaknesses. Taking this in to account, in the ideal scenario a game would have both kinds of enemies, as the tester exemplified, the bosses being strong and adaptive while the lesser minions being multiple common weaknesses provided the best solution to how the game’s enemy AI should be developed in a way that would be adaptive to every player. Figure 17 displays the answers obtained:

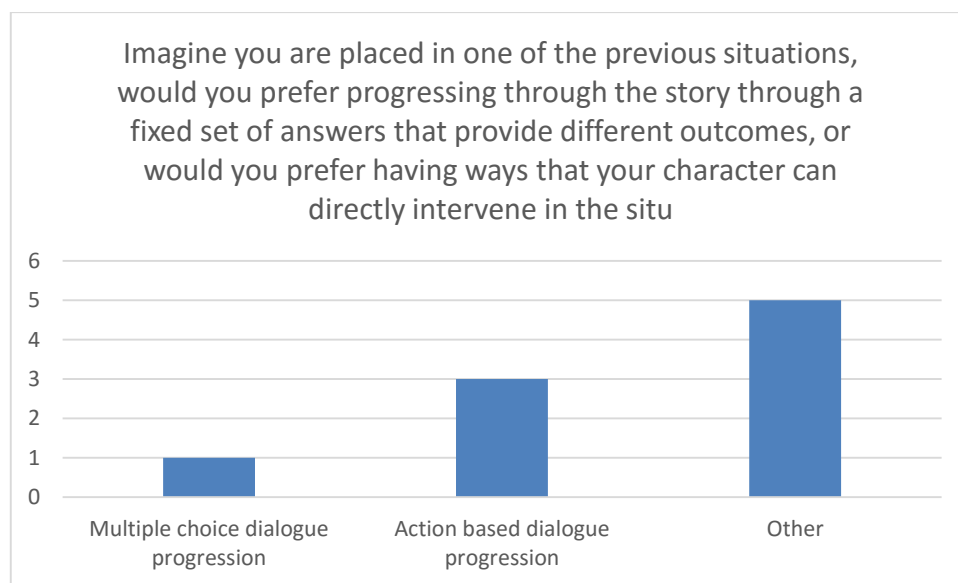


Figure 17 - Results for the second question in the game design related to game mechanics

The second question, which attempted to understand how the story progression should be done in the game when it came to dialogue interaction, had similar results. Only one tester chose to have multiple choice dialogue, three testers chose action based dialogue, and every other tester chose the “Other” suggesting that both solutions would be the preferred solution. Knowing this it would be possible to provide such a world where choosing options in a dialogue would affect the outcomes of solutions but, at the same time, allowing for interaction with world objects would result in different outcomes as well, this would be the ideal solution.

The alpha testing process was extremely important because it allowed the author to theorize how to design the story of the game, how to provide interactions between the players and the characters, how to develop enemy characters and their artificial intelligence, setting the base for how the game would be. Initially, the project as being developed as a game based off these initial approaches and the results received from the alpha tests, this resulted in the game have numerous features that were planned to exist in future iterations but never got developed in the final project. The reason for this was that the author lacked experience and enough time to answer the colossal answer that is, how to develop a game which its design is fully adaptive to what the player decides in all aspects? But this problem will be answered to a higher degree on the next chapter.

3.1.3 Current approach

The main key focus of an adaptive game design for this video game had to clearly be the interaction of the player with the characters in the rest of the world. This is how a game becomes adaptive, when the player talks to the environment and decides what to do, with the environment responding according to these actions. For this it would be fundamental to develop an interaction where the player was able to take choices based off which situation is presented. This raised a problem to the author, how to develop a system where the player could make choices from, and how would that system make the world change accordingly?

Based off the initial approaches it was difficult for the author to come up with a solution, for someone who did not have any experience in this field and with a limited time period, the solution couldn't be something too complex to develop. This led to a lot of experimenting and trial and error, because there was no clear solution, the project began by being programmed to be a fully playable demo that is why there are features such as a third-person camera, NPC shops (which were not added in the final project), as well as 3D models hand made by the author inspired in an art style. A lot of the time was spent learning how to do these features and then attempt to actually implement them, with many times failing at doing so and having to rewrite the whole code, and leaning a lot in the process.

3.1.3.1 Shop and dialog data

The solution to the question presented earlier materialized when making the “shop functionality” coupled with the dialogue system. To have a shop, an NPC had to have persistent information. This could be made directly in the scripts but that would leave a static integration

of how the shop was made and it was not very user friendly neither to the programmers (because scripts would have to be written for each store) neither for non-programmers. To overcome this and provide persistence whenever the player left a game text files were used (see figure 18). This allows for creation of “save files” (for the complete game status).

```
Bomb;Consumable;5;150  
Potion;Consumable;2;100
```

Figure 18 – Shop items are stored in a text file with name, item type, quantity and value in semi-colon separated values

After implementing the shop, the dialogue system was the next feature because it also needed to be persistent in case the game had a save game feature in the future. This prompted the author to develop a dialogue system feature which also loaded from files. Because lines of text were too limiting these dialogue files were made in JSON so they could have a more easily readable and accessible structure. This differs from the reason the items of a store were made in a text file because, in the case of dialogue, each character can have (and will probably have if the game gains dimension) huge amounts of dialogue, not to mention that each dialogue line can be big in of itself. It becomes a lot easier to access a JSON file and copy paste JSON objects, rather than have cryptic coded scripts that need to have their variables changed every time a dialogue needs to be added, edited or removed. This system where the dialogue changes the world was one solution that appeared when studying what would be the best approach to this concept. With this solution regardless of what happened on the game’s world when a player talks to a character and goes through its dialogue messages, then a group of events will always be triggered as they are part of the dialogue and independent of the rest of the game.

3.1.3.2 Personality traits

While developing this dialogue system the biggest problem that emerged was how is the player going to take decisions based on the messages provided? There were various solutions in mind: loading files would only load dialogue messages with the events to be generated would have to exist on a separate script that dealt with that character’s dialogue – this was an okay idea and had some potential since it made it easier for characters in the game to change the types of events generated; there could be dialogue messages that prompted multiple answers to the user with each answer representing an action but it felt too confusing and complicated to implement with the time and resources available; and it was then that the answers provided in the alpha testing form gave the solution: personality traits.

During the alpha testing phase, testers were presented with different quests that provided different settings. Initially, the quests were nothing more than a group of adventures that were meant to be implemented in the game at some point, with the alpha tests helping the author understand what kind of answers the general audience would like to see in that scenario. It was interesting to realize that each answer the testers gave, even though specific in its own way, could be simplified in to a personality behaviour. Some testers wanted to face the enemies head on and fight with them (aggressive, violent behaviour), some others wanted to sneak and find an alternate route that would serve their own purpose (cunning, stealth behaviour), the vast

majority of the testers wanted a more peaceful solution trying to find ways they could talk to the characters and convince them of a common solution (pacifist, diplomatic behaviour), and few wanted to explore the surrounding and every option they can take before deciding what to do (explorative behaviour). The idea then became making so characters would have a group of personalities of their own. If a character has a group of personalities it is then possible for the NPCs to look at them, analyse them, and react to them properly.

Inspired by games such as Pillars of Eternity, DnD and other medieval fiction, the whole concept of personality traits comes from, picturing a scenario where the player has the power to imagine whatever character he wants to create, and then actually pours such imagination in to a game. At first this concept seemed daunting, how could a game ever encompass all possibilities within the prism of what a human being can imagine? The answer came in to developing scenarios which were provided by a personality group system. Having a game design centred on a group of personality traits that reflect the behaviour of the player's character allows for a bigger range of possibilities (within the game's limits) for any kind of character to be created. In an ideal scenario, the player would have the option of creating his/hers character whether as a blank canvas (no specification of what kind of character it is and rather let his actions within the game shape his character) or specify fully within the games boundaries what kind of character he wanted. Right now the project employs a system which is a mix of both, the player is able to choose from a pre-determined group of alignments that set a basis for their character's personality trait but their actions within the game also change their personality.

3.1.3.3 Personality traits for NPCs

To explain the workings of the non-playable characters or the combat system it is necessary to detail this personality trait system. In the game there are eight numeric values from 0 to 100 that represent personality traits: rational, caring, violent, pacifist, courageous, reluctant, diplomatic and cunning (later shortened to four traits for testing purposes). These traits vary according to player actions creating the idea that each action the player takes helps shape up the character. One important aspect when this personality system was developed was to make it so it wasn't binding. If the player created a character with a violent background alignment then that character should still be able to make pacifist (opposite personality trait) choices with lesser frequency or even become a pacifist in the process albeit less quickly.

In the case of the project this reflection is not transparent, as it implies being able to make multiple actions along a period of time. Because such was not possible in the time given, personality traits were designed only for the player character but for any object in the game, to be able to accommodate this possibility in the future.

The main idea is that any object can have specific personality traits associated with it. Imagine the case where character A has 20 (out of 100) points in the violent trait. The player character cannot interact with character A until it has at least 20 points in the violent trait. Thus, personality it's a requirement: the player is not violent enough to interact.

Any sort of combination can be used for this purpose, another example is making the requirement if overall the player character is at least 25% cunning. Because both the player

character and any other character or even object hold a group of personalities their own it is then possible to make the game vary and behave differently by, comparing the player character's personality and matching a set of requirements which are typically based on the target character's personality as well. This is what provides a living breathing experience to the game, these requirements change the NPCs behaviour independently because they rely on the player character's personality which in turn rely on the actions he/she takes. This lead us to the second point, when questing throughout the world the player character is often prompted to perform actions as responses to certain situations, in the case of the project when prompted to confront the Ogre the player can take one of our actions (Rational, Violent, Courageous and Diplomatic), when one of these actions is selected it changes the player character's personality according to what action he decided to take, this means that each action will fluctuate the character's personality and allow him to achieve the requirements set by the NPCs. At the same time, although not described in the project itself, it's possible to change the NPCs behaviour through given key actions in a quest, imagine a scenario where the player chooses between saving a princess or defending a village, whatever decision will change the other character's behaviour not by the player's personality but by what decision he decided to take in that scenario. This happens because how each NPC behaves is decided by its dialogue tree, when a key action happens it becomes as easy as simply changing the current dialogue tree of the target NPCs to whatever tree corresponds to their reaction to such an event. This is explained in more detail in the further chapters of implementation but, essentially, events in the world can also be triggered through dialogue, that's why it generates reaction.

It is important to realize the potential of this system, although simple in concept it creates a lot of potential since the only requirements to build reaction to actions or personalities lay solely in attributing a personality group to a given object in the game (anything can hold a personality trait group), and then setting its requirements for whatever interaction is to be had in a given moment in the game's plot. This is what cause reaction, the player has a certain behaviour strength to it and other characters will react according to it, the player takes a meaningful action in the plot and the world around him/her reverberates with consequences.

3.2 Implementation

In reality many of the aspects previously mentioned on the initial approaches chapter were disregarded and changed since the focus of the project went from instead of making a video game to study the best way to implement an adaptive game design. Because of the initial idea to develop a complete demo, the author began by developing an NPC structure that would allow a character to interact with the player through dialogue in three different actions: talk to the player, with optionally open a shop (sell items), and generate events. To create these functionalities it was taken in to account how it would be necessary for them to be made in a way that were accessible for people who were not used to programming or didn't knew anything about it, as such, they were made through reading information from a file, they could've been done through scripts but that would make it less generic, less accessible as not anyone can read through code. This "reading information from a file" refers to the dialogue

system, which is the first chapter of this sub-chapter, it explains in detail how the dialogue system coupled with the personality traits technically set the base of the project and all its adaptable game design, as well as making it quite accessible. The modular implementation chapter pretends to explain how everything in the game was made as most generic as possible so that everything can be seen as pieces of a puzzle which can be removed or added or multiplied as the developer sees fit, in most cases being as easy as dragging scripts to game objects in Unity with minimal setting up requirements. The visuals chapter explains how the models of the game were designed, as well as what art style was the game inspired in. The combat system chapter technically specifies how even the combat system was inspired in the personality trait system and how each action the character takes changes the enemy's personality and subsequently its behaviour. And finally, the future development explains how technically the project is still in a very early phase and how most of the present features could be refined, and how there are many other features that could be added in the future to compliment and solidify this concept.

3.2.1 Dialogue system and personality traits

The dialogue system is intricately connected to the personality trait system as one relies on the other. The process of determining personality begins in the character creator, to demonstrate that behaviour changes according to actions it was easier for the author to develop a character creator that would explain how the system work, would set the initial stats for the character, as well as different personality values. The character creator also served to give some form and shape to the player character, allowing for a name to be assigned, a background to be chosen that changes the stats of the character, and finally an alignment that sets some values for the personality traits of the character. The last panel gives a simplified explanation of how personality traits affect the world and change it depending on the situation. Going in to the game the player is then able to interact with only one character, as all other refuse to talk to him/her because of the lack of personality values. Dialogue in this case happens after the player approaches a character, is prompted by message, and presses a certain key. All non-playable characters in the world have a state structured which exists within their own NPC script's update function (update function is called once per frame) allowing the developers to change the interaction state of the character when they so desire.

There is a custom enumerable variable created in a global accessible document that holds different states such as default, prompt, talking, shopping, open shop and clear scene. There is then a switch in the update function that reads the value of a variable that is of this enumerable type and usually is initiated as default – default corresponds to passive activity, the NPC does nothing awaiting interaction (see code 1).

```

// Update is called multiple times a frame
protected virtual void Update()
{
    switch (CurrentInteractionState)
    {
        case ResourceManager.InteractionState.Prompt:
            // Check if the player presses the assigned key when prompted
            if (Input.GetKeyDown(KeyCode.E) && InBounds)
            {
                // Check personality trait requirements
                if (!CheckPersonalityTraitRequirements())
                {
                    return;
                }

                // Look at the player (rotate according to Y axis)
                TurnTowardsThePlayer();

                // Turn off the prompt panel
                PanelManager.SetPromptPanelState(false);

                // Ask our manager for the next message for this NPC
                RefDialogueManager.GetNextMessage(this);

                // If we are prompting the player to talk to we set our new interaction state to talking
                CurrentInteractionState = ResourceManager.InteractionState.Talking;

                // And finally set the dialogue panel to true
                PanelManager.SetDialoguePanelState(true);
            }
            break;
        case ResourceManager.InteractionState.Talking:
            // Check if the player presses the assigned key when prompted
            if (Input.GetKeyDown(KeyCode.E) && InBounds)
            {
                // Ask our manager for the next message for this NPC
                // If it returns true it means we reached the end of our dialogue
                if (RefDialogueManager.GetNextMessage(this))
                {
                    if (isShop && !gameObject.GetComponent<Shop>().IsEmpty())
                    {
                        // We do this by setting our state to shopping
                        CurrentInteractionState = ResourceManager.InteractionState.Shopping;
                    }
                    else
                    {
                        // Otherwise set to default as we are no longer talking
                        CurrentInteractionState = ResourceManager.InteractionState.Default;
                    }
                }
            }

            // Look at the player (rotate according to Y axis)
            TurnTowardsThePlayer();
        }
        break;
        case ResourceManager.InteractionState.Shopping:
            // Here we look for the Shop component associated with this game object
            // and change the interaction state to shopping, making it draw the shop
            // and behave accordingly (we just do a safe check for the component in case we forget about it)
            GetComponent<Shop>().currentInteractionState = ResourceManager.InteractionState.OpenShop;

            // This script returns to default
            CurrentInteractionState = ResourceManager.InteractionState.Default;
            break;
        case ResourceManager.InteractionState.Default:
            break;
    }
}

```

Code 1 - NPC script Update function detailing the different states that vary according to its current interaction state variable

When the player approaches the character and enters a sphere collider, that collider will then change the state of the NPC to prompt, making the switch enter the prompt field and then, effectively, prompt a window to the user so he/she is able to interact with that character (see figure 19). When the player hits the designated key “E” the character rotates towards the player, clears the prompt window, fetches the next message and changes its state to talking.



Figure 19 – Main character being prompted to talk to the Knight

Every time the player hits “E” from then on, the NPC keeps fetching the next messaging and updating the display accordingly. When it reaches the final message in its dialogue tree, then it clears the message window and goes back to default state. If a message corresponds to a shop prompt, than that message line should have an event that changes the NPC’s state to shopping, effectively triggering the associated Shop script of the game object and opening a shop window for the player to buy items from. The NPC script is able to fetch these messages because all NPC dialogue exist in a well-structured JSON file outside of the game, making it accessible and easily editable by anyone (see code 2).

```

{
  "activeDialogueTree":0,
  "dialogue":[{
    "dialogueTree":[
      {
        "message":0,
        "text":"Hello world!"
      }
    ],
    "dialogueTreeIndex":0
  },
  {
    "dialogueTree":[
      {
        "message":0,
        "text":"Hello world 2!",
        "choices":
        {
          "Yes": {
            "index":3
          },
          "No": {
            "index":4
          }
        }
      }
    ],
    "message":1,
    "text":"Event message!",
    "event":1
  }
  ],
  "dialogueTreeIndex":1
}

```

Code 2 - Example structure of an NPC’s dialogue JSON file

Each NPC’s dialogue can be seen as an independent graph, with various dialogue trees, which in turn contain dialogue messages. Each JSON has an object that identifies which is the currently

active dialogue tree, this was to allow save games in the future of the current dialogue tree the NPC was in. This is followed by a dialogue object that holds several dialogue trees with each dialogue tree being the object that holds the messages itself (see figure 20). In short, the dialogue object is the whole dialogue universe of the character, with each dialogue tree being a node in the world that hold the messages corresponding to that tree.

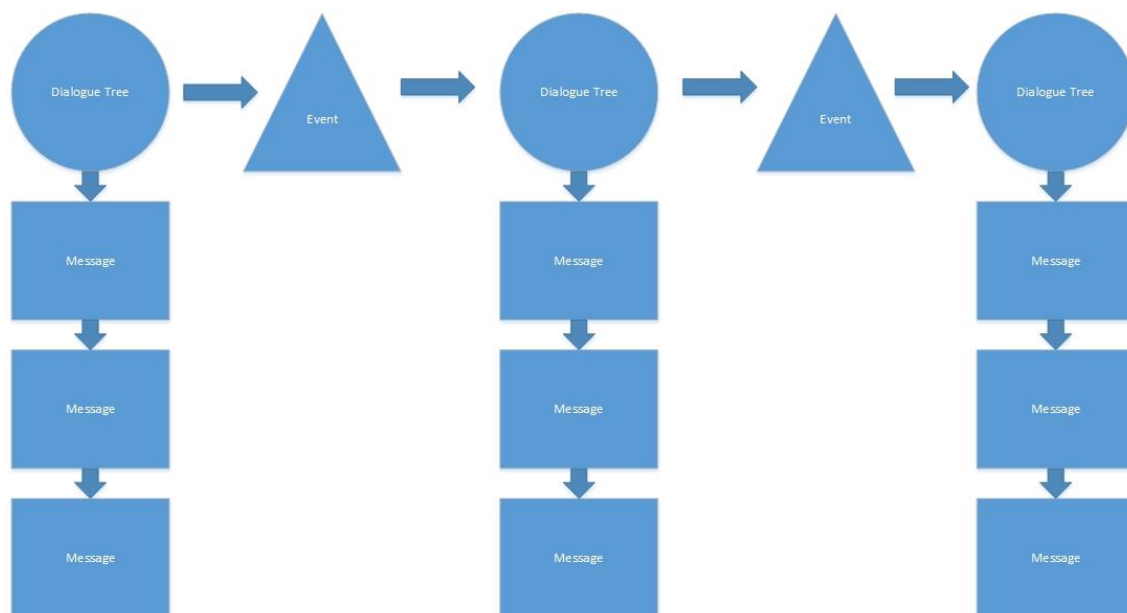


Figure 20 – Dialogue representation by their nodes

Each message object itself has: an ID that identifies the message as well as a text object that holds the actual text message; optionally can also have a choices object (which will prompt in game several options buttons that can be clicked) that holds the text of the options as well as which index they will move towards when the option is clicked; finally it can also have an event object that corresponds to which event is that message going to trigger, this object is also optional.

The one important NPC script loads all of the dialogue through a helper class called FileManager, what this helper class does is, it looks for a file in the StreamingAssets folder (a global Unity folder that is compiled along with the executable) which has a name of CharacterName_dialogue.json (where character name is the actual name of the character that is trying to load its dialogue), and will then proceed to load the JSON objects and convert them to their respective values. To explain how dialogue is loaded from a file it is easier to explain the structure of each object first. In the JSON file for the character’s dialogue, we have dialogue trees which are essentially groups of dialogue lines, in the game’s code there is a class called DialogueTree that serves the exact same purpose, holding a list of Dialogue which is essentially all the dialogue lines that correspond to this dialogue tree, as well as an integer variable to hold the dialogue tree index (as described on the class diagram in figure 21). This Dialogue object that composites the dialogue tree, is what is seen on the file as the one who holds the message index, the message text, with optional objects for choices and events to be triggered. Each NPC

script in turn has a list of dialogue tree objects effectively making for its full dialogue and the respective consequences.

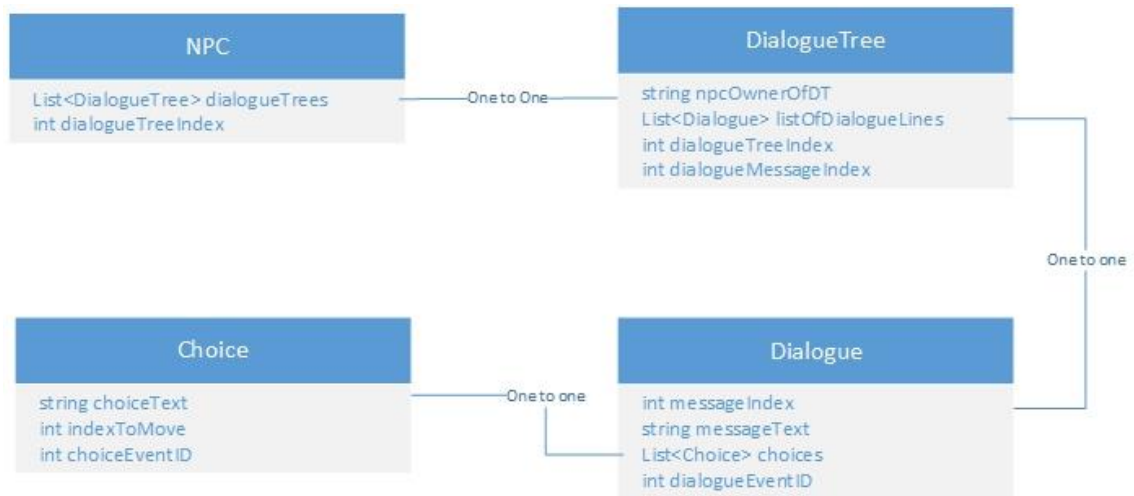


Figure 21 – Class diagram demonstrating how dialogue is structured in its classes

The script begins by trying to load the file as explained above. In “LoadDialogueFromFile” function in the global FileManager script, the file is first loaded to a string and then deserialized as one big JSON object. Through the use of the Facebook’s JSON library¹, it’s possible to scan the JSON object for the keys corresponding to the information we want. The function then starts by loading the current active dialogue tree and setting it in the NPC’s script, it then proceeds to fill a dialogue tree object with dialogue objects, and setting the dialogue tree index. When a dialogue tree is properly filled it is then added to the list of dialogue trees that was created initially and returned to the NPC script who stores it.

When a player interacts with the NPC and requests a new message the DialogueManager (another globally accessible script for support), receives the NPC script of which the player is talking to and begins by retrieving the active DialogueTree object, checking if we are in fact at the end of our dialogue, because if so it returns true and the NPC ensures that windows are closed and everything dialogue related is reset in that current dialogue tree, it then checks if such dialogue message has choices, in the case it happens it fetches the choices of that message and displays them to the user along with the message, if not it checks for events that can be associated to the message and calls them while displaying the message to the user at the same time and incrementing the current message index. Figure 22 attempts to explain this concept better through a sequence diagram:

¹ courtesy of the colleague also working on this project

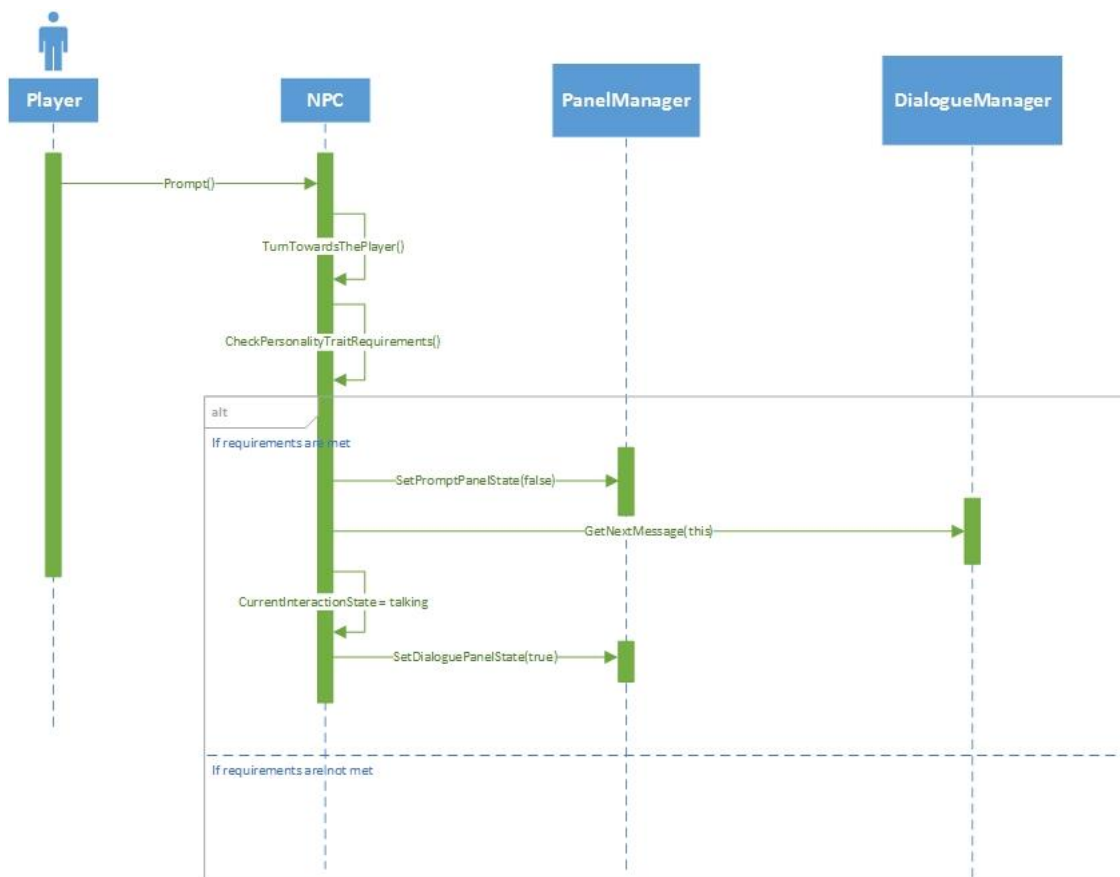


Figure 22 – Sequence diagram illustrating the interaction process when a character approaches an NPC

One thing which is of extreme importance is to note that, since both choices and events can be triggered through dialogue (events can also be called independently through code), it provides a new layer of simplicity to the system. If one writer wanted to implement the dialogue of character it would be as simple to fill the JSON with the corresponding dialogue messages and then explain to the developers how a certain event was to be triggered in key dialogue messages.

This becomes even more important when referring to the personality traits of a character. As of now personality traits are treated as enumerables, stored in a dictionary of personality trait and an integer (corresponding to the value of that personality trait), both to NPCs and the player character. When a player character interacts with an NPC, one of the requirements to be able to talk to such character is checking if the highest NPC personality trait is lower than that of the player character, if so, then the NPC will talk otherwise it will refuse interaction. This can either be done this, statically through function checks before displaying a message, or it can be done with the system explained early, talking to the NPC provides a generic initial message that triggers an event which checks for requirements or, for example, the NPC script can hold an event object which corresponds to its requirements so as to whenever the player attempts interaction the NPC calls the event associated to it and checks for requirements. In the grand perspective of things this means that, whether through dialogue, or through scripts, the NPC's behaviour can change as anything can access it to either change its own dialogue tree index –

making the NPC display different messages – or to change the event associated which checks for requirements when talking to that character. The key aspect here is simplicity and accessibility, this is what provides that aspect of living breathing environment, when changing the characters in the world becomes as simple as modifying the value of an integer or changing an object associated to that NPC script.

3.2.2 Modular implementation

The modular implementation attempts to explain how most concepts in the game were made to be as generic and easy to implement as possible, similar to how a puzzle piece can be exchanged or replaced in a different place of the puzzle. Creating an NPC in the game becomes as easy as attaching the NPC script to a game object, typing the name of the NPC in the script, creating the dialogue JSON file and optionally create a shop text file in the case that the NPC is a shop. After that the whole NPC script takes care to load all of the necessary information and to make the game object behave as a character, this happens because each character behaviours is the same between them the only thing that actually changes is how the NPC checks for requirements which, as mentioned before, can be done through events or exterior game objects.

So far the word event has been thrown around a lot but it has never been explained to detail, this is because the events were also made as a modular implementation. Events are objects in the game which, just as the name implies, change the game in significant ways when a critical point is achieved. Because events are absolutely generic and can be called by anything or anyone, a class EventDatabase was made. This class serves to do nothing more but interface between the game world and the event that was called, it holds a list of the game's events as well as functions who search for the event by its ID and either trigger or end it. Anything in the game that wishes to trigger an event has to have this EventDatabase, and then call its respective functions. The event itself inherits from another class called QuestEvent because each event always has an ID to identify it, a target NPC (which is absolutely optional, it exists to simplify access of characters in the world without using the unity engine statically), an integer which corresponds to the new dialogue tree index of the target NPC (also a simplification since most events will or should almost always change the dialogue tree of target NPCs), and a boolean to determine whether the event is currently active or not. When it comes to functions there are only two inherited functions necessary, StartEvent() and EndEvent(). By inheriting from such a generic class, any event is made generic, for developers they have to assign an event ID and their correspondent target NPCs with new target dialogue indexes while implementing the start event and end event functions for whatever the event is supposed to do, while the only thing anyone responsible for the dialogue has to do is assign the event ID to the corresponding and it immediately gets triggered by the NPC script through the use of the previously mentioned EventDatabase class. Figure 23 explains through a sequence diagram how an event is fetched and triggered through the dialogue manager (responsible for interacting with the UI).

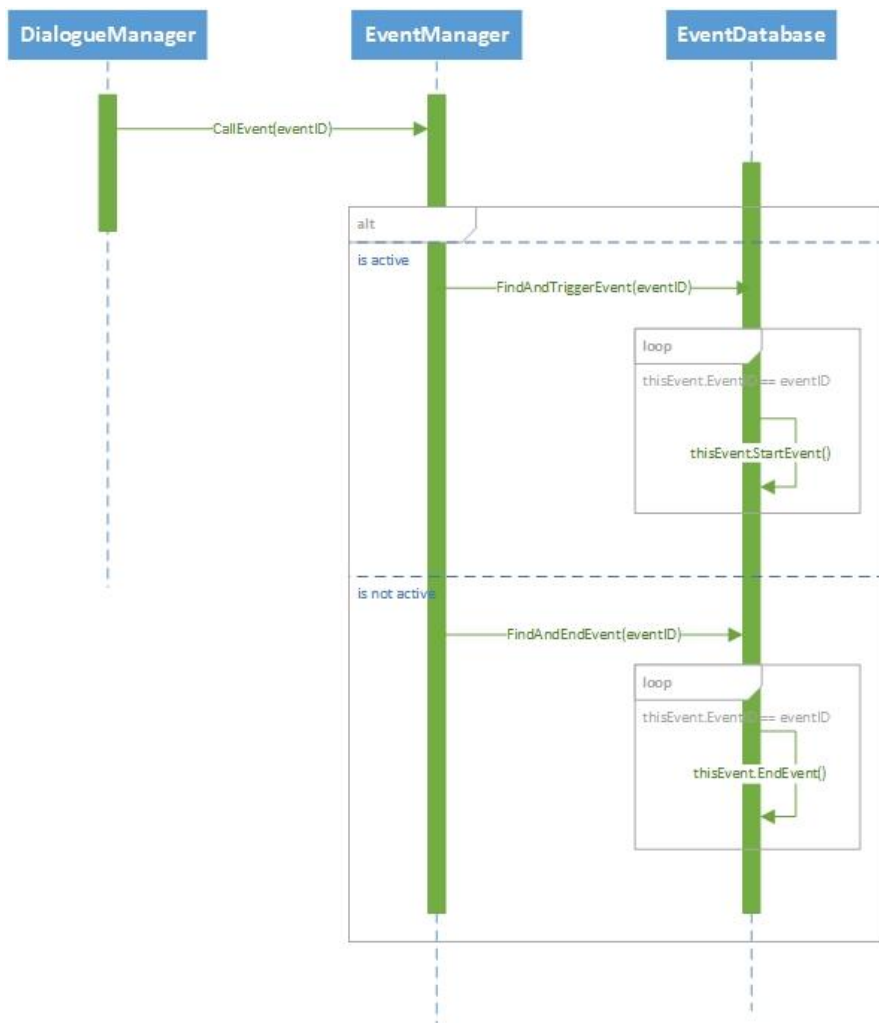


Figure 23 – Sequence diagram detailing how an event is called, started and ended

It is extremely important to emphasize the potential that this event system holds. Because events can either be triggered through dialogue messages or through coded scripts and they can effectively, change anything in the world they hold a tremendous amount of potential in a modular way. The personality trait system as of now works by comparing the highest trait the character holds with that of the player character, if the player character has a lower trait then the requirement is failed. This can easily be changed by associating an event as a variable of the NPC script. What this means is that, if we associate an event, than anything can be the interaction requirement of that character. When the player character tries to interact with that NPC the event is called performing whatever checks necessary. Code 3 displays one of these cases where the player interacts with an item that provides gold.

```

public class Event001 : QuestEvent {

    // This particular event requires a reference to our inventory manager
    private InventoryManager inventoryManager;

    public InventoryManager InventoryManager
    {
        get { return inventoryManager; }
        set { inventoryManager = value; }
    }

    // Constructor for Event001
    public Event001()
    {
        // Fill the base information but for this specific event
        base.EventID = 1;
        base.TargetNPC = "Player";
        base.NewDialogueTreeIndexOfTargetNPC = 0;

        // Get the reference of our inventory manager
        InventoryManager = InventoryManager.GetInstance();
    }

    // Override version of super class StartEvent
    public override void StartEvent()
    {
        // Calls item manager to handle what to do with the item found
        InventoryManager.TreatItem(2, 1, true);
    }
}

```

Code 3 – Example event for when a given item is found, in this case when Gold is found

Unity inherently holds a group of functions that allow access to any game objects in the scene. The `GameObject` library has two functions: `Find` and `FindGameObjectWithTag`. The first function `Find`, looks for a game object with the given name, this function is more static since if two game objects happen to hold the same name or if the name is changed then the function will not achieve the purpose desired. The second function works with the tag feature, in Unity game objects may optionally hold a tag value, this is important because it allows the developers to fetch either any amount of game objects as long as they hold that tag value. With these two functions an event holds access to anything in the game.

By making events these generic classes that can be called either externally (through files) or internally (through scripts), and thanks to the Unity functions to access the world, they can effectively create any sort of requirements, or actions, whatever that's necessary. They are ambiguous enough to be used in any case scenario and powerful enough that they can verify or change other game objects, allowing to be used by characters as their own personality requirement checks, or to change the world dynamically, or even the behaviour of any other characters.

Both the player, the enemy and the whole turn-based combat system were made in a modular perspective. Characters inherit from a generic `Character` class that holds common traits for both

characters. This is even more important in the combat system because, given that the system is turn-based it is of vital importance to make any character behave accordingly in the combat system's queue. Figure 24 details how each class is structured and how other classes inherit their structure and implement variance of their own.

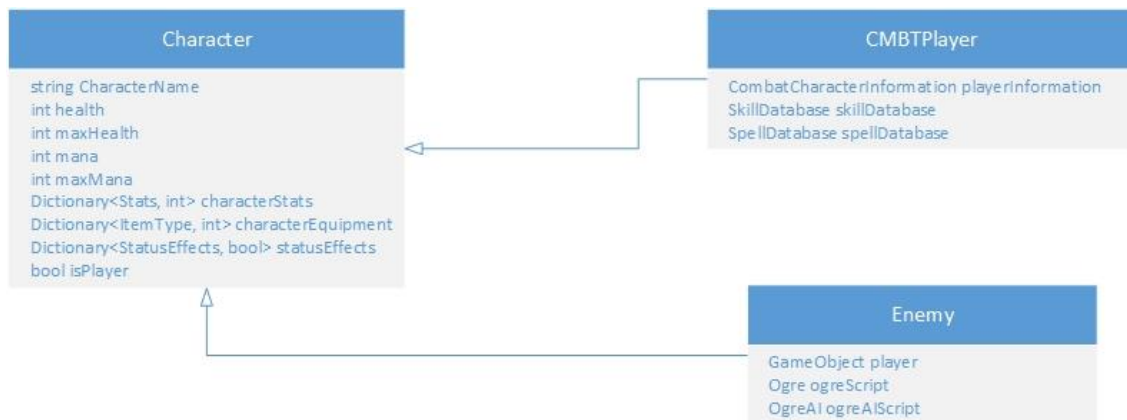


Figure 24 – CMBTPlayer and Enemy scripts inheritance from the Character script

The generic **Character** class holds a character name, the amount of health the character currently has, the maximum health of the character, the amount of mana the character currently has, the maximum mana of the character, the character stats, equipment currently affected status effects and if the character is the player or not. In regards to functions there are only two common inherited functions: `ActivateCharacter()` which corresponds to what triggers a character to act in its combat turn, and `ApplyStatusEffects()` a class which, as the name implies, applies a certain status effect to the player it is generic because status effects are applied the same way both for enemies and player characters alike. This way the combat itself can activate characters generically without worrying for what kind of enemy or player character it is that we are referring to. Even the combat itself is modular, characters in a fashion very similar to how they behave in the overworld interacting with NPCs, also have states albeit in this case the states correspond to turn phases, when a character is activated by the **CombatState** script it activates it start turn and from there performs whatever actions necessary until it reaches the end turn state where it tells the **CombatState** script that its turn is over. Implementing characters this way means that no matter what character or what enemy is developed all the **CombatState** has to do is manage the state of the combat by, checking which character is supposed to act, trigger that character's turn, wait for it to finish, calculate the next character to act and trigger that character repeating the cycle no matter which characters are involved.

3.2.3 Combat system

The type of combat system chosen for the game was a turn-based combat system, this type of system was chosen both for being rather simple to implement and because it also allowed for a deep complexity of artificial intelligence behaviour. This was important since the whole focus of the project was to have an AI that changed its behaviour according to the player's actions.

Initially the combat system was developed to have several enemies with one player character facing them, in which case there is a visual queue that is sorted and update at the end of each turn. It was made this way because initially the author expected to have a game with a bigger environment with more combats with multiple enemies. The project fundamentally uses this combat system except that there is no visual queue on display and the sorting of characters is simplified.

The turn-based combat core is the CombatState script, which is in charge of managing the start of the combat, how the turn queue is managed, which character's turn to go active, as well as resorting the characters at the end of each turn and ending the combat. Figure 25 details this combat system further in a sequence diagram. This script holds a list of Characters (the generic class) which consists of all the characters within the game. Upon initialization of the combat scene, the script finds all the characters in that scene which have either a tag "Enemy" or "Player" finding the corresponding game objects that are going to take turns and partake in the combat itself adding them to that list of Characters. After that it sorts the characters based off their own agility stat and re-added to the list of Characters. Finally the combat state calls ActivateCharacter() for the character of which is at the top of the list.

Before explaining what activating the character means for both the player character and an enemy, it is worth explaining the stats in the game. Stats are affected by the player background, different background provide bigger or lesser values to different stats, whereas each stat influence the game (mostly the combat) in different ways. There are three main stats (strength, agility and intellect) and three lesser stats (vitality, reflexes and knowledge).

Strength defines how much defence a character has, as well as its health pool. The formula for strength is as follows:

$$\text{Health Pool} = (\text{Strength} / \text{Max Strength}) * \text{Max Health}$$

Formula 1 – Health pool formula

Where the formula for defence is (Armor defence is optional):

$$\text{Defence} = (\text{Damage} / \text{Max Strength}) * (\text{Strength} + \text{Armor Defence})$$

Formula 2 – Defence formula

Vitality, the secondary stat of strength, represents a resistance to status effects and magical attacks, the formula for vitality is as follows:

$$\text{Resistance} = \text{Magic Damage} * (\text{Vitality} / \text{Max Vitality})$$

Formula 3 – Resistance formula

Agility determines how much damage an attack is going to hit the opposing character for, the value is managed by the player (in some ways) and as such the formula cannot rely too much on its value:

$$\text{Damage} = \text{Weapon Damage} + \text{Weapon Damage} * (\text{Agility} / \text{Max Agility})$$

Formula 4 – Damage formula

Reflexes, the secondary stat of agility, serve to solely determine the critical chance of each attack performed. The formula for reflex is:

$$\text{Critical Hit\%} = \text{Max Reflex} - \text{Reflex} \text{ (Where Max Reflex is 100\%)}$$

Formula 5 – Critical hit percentage formula

Intellect, similar to strength in ways, defines how much damage a character does with magical attacks. Intellect's formula is:

$$\text{Mana Pool} = (\text{Intellect} / \text{Max Intellect}) * \text{Max Mana Pool}$$

Formula 6 – Mana pool formula

And finally Knowledge, the secondary stat of intellect, provides another boost to magical attacks, amplifying it further depending on the enemy resistances. The formula for knowledge is:

$$\text{Bonus Damage} = (\text{Knowledge} / \text{Max Knowledge}) * 100$$

Formula 7 – Bonus damage formula

With this stat explanation it's now possible to explain that each character is ordered in the list of Characters in order of their Agility, when the CombatState activates a character it will activate the fastest character as the first character to attack. Each character, similar to overworld behavior, has a state of its own, dividing the process in to: StartTurn, Moving, Attacking, Returning and EndTurn, with slight changes between enemy characters and player characters.

When a character is activated, it changes its state to StartTurn if it's an enemy, otherwise it has its state in default until the player chooses an attack. In StartTurn the character will then check if it has been destroyed, if so it prompts a winning or losing message (depending if it's the player that has been destroyed or the enemy), then proceeds to check if status effects have been applied, ending with changing the state to moving.

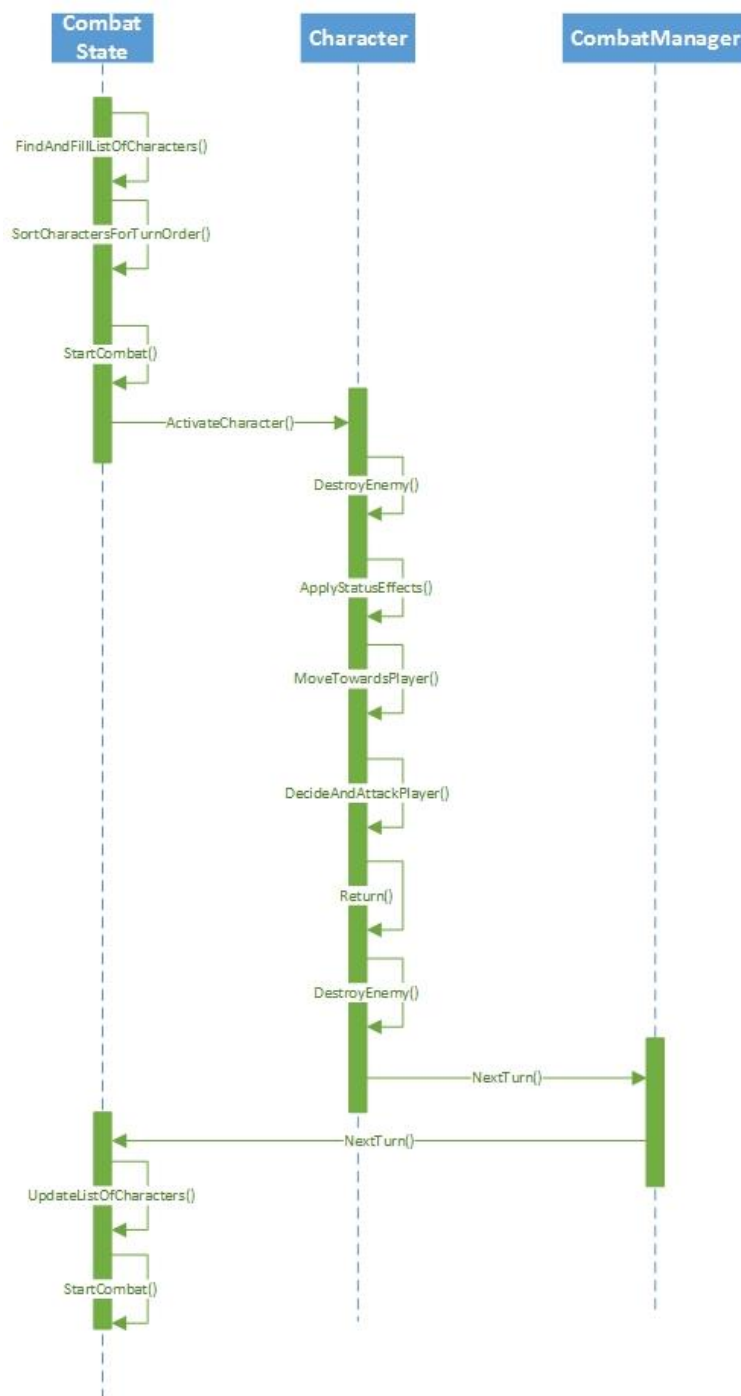


Figure 25 – Sequence diagram explain the combat flow of an enemy in a turn-based order

The moving state changes the position of the character and moves it toward the enemy, when the character has reached its target it changes the state to attacking. In the case of attacking, if it's a player character it performs the attack chosen, in the case of the enemy it first has to decide which attack to perform this is where it gets interesting. In both cases, the type of attack that is performed varies according to the weapon equipped, this can be seen in the project by taking a look at the status effect tab on the enemy's status bar, each weapon that is given to

the player performs a different status effect, because each weapon has its own specialty and form of attacking. The enemy AI should have a varying script (one for each enemy) that determines how the AI is going to behave, in the case of the project, the Ogre checks for its highest personality trait and then chooses the attack based off of its own personality. Each attack the player does towards the Ogre changes its personality, more aggressive attacks increase its violent trait where more sneaky attacks increase its cunning trait, and the Ogre chooses an attack based off of which one is higher. After an attack has been chosen and performed the character changes its state to returning, which consists of moving back to the original order, and once arrived at its destination it changes its own state to EndTurn, which once again checks if the character has been destroyed, to be able to end the combat on its own turn, in case it wasn't it calls the CombatManager (a script whose only job is to provide an interface between the characters and manager scripts) and tells it, it has finished its turn it's time to get the next turn and proceed with the battle. This cycle is repeatedly infinitely until one of the characters has fallen ending the battle. It's important to note that the Ogre's combat is entirely decided on his own personality and if the player takes diplomatic decisions it's possible to even avoid this entire fight all together.

3.2.4 Visuals

3D Modelling is an art form with numerous different styles that produce varied results. Of all the different techniques, the most interesting ones were digital sculpting, edge modelling and box modelling.

Digital sculpting is a modelling technique which implies sculpting 3D models from mesh-based geometry. Much like in real life but with tablets, an artist begins by using a mesh-based geometry object and then sculpts it until it reaches the desired 3D look. This technique was discarded by the author as it implies an immense knowledge of sculpting and drawing techniques. Edge modelling is a technique similar to box modelling but instead builds a model piece by piece by placing loops of polygonal faces along contours, filling any gaps in between. This technique is excellent when it comes to modelling objects that are inherently round and edge-based, such as eyes, eye sockets and face of 3D models in general. Once again, this art technique was discarded as too complicated for the author to develop as it implies knowing how to manipulate edges to achieve the desired look. The modelling technique chosen for the game was box modelling. Box modelling was chosen because it is a simple technique, one of the easiest to pick up and learn since it implies designing a character from a primitive shape (in the case of the project it was from cubes), where the primitive is stretched, subdivided and expanded to normally fit a concept image. The resulting look is a simple, yet cute and adorable look that is still able to inspire excitement.

When it comes to the visual aspect the game begins by presenting a main menu screen (see figure 26) which was prepared to have different features implemented in the future (such as options and load game feature).



Figure 26 – Main menu screen of the game

Upon clicking on the “New Game” button, the player is then led to the Character Creator screen (see figure 27), a screen which was prepared specifically to build and ascertain different options and characteristics of the player such as: their name, alignment, background, and an explanation of how personality traits work.

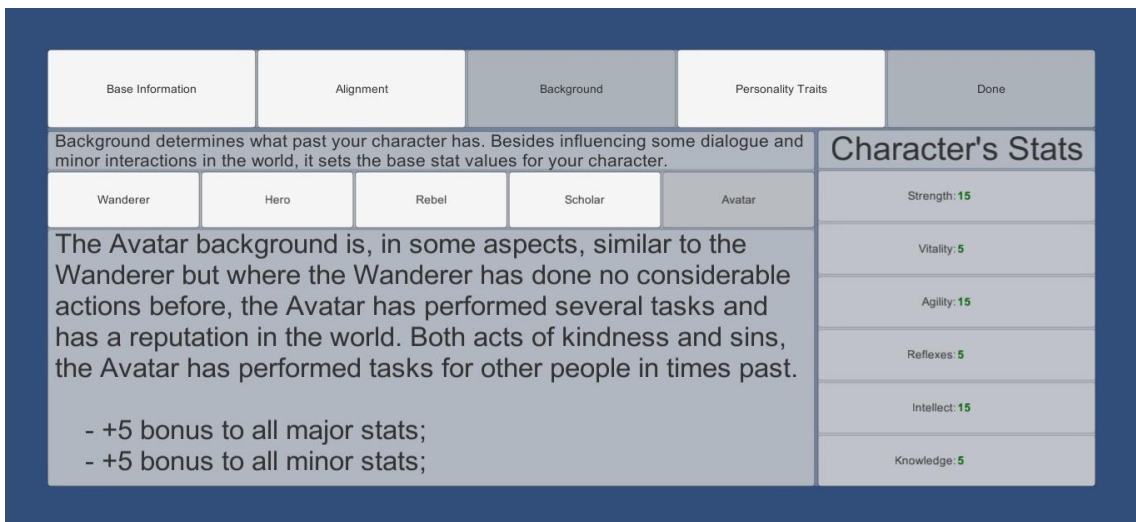


Figure 27 – A character choosing the “Avatar” option on the Character Creator menu

Once a name, an alignment and a background have been chosen for the character, by clicking the “Done” button we enter the overworld scene itself which contains a HUD (see figure 28) that presents different options.

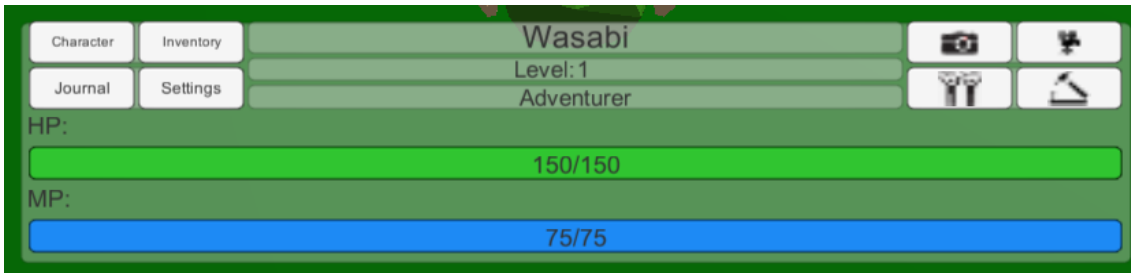


Figure 28 – Overworld HUD presenting various kinds of information

In the HUD itself a Character information, an Inventory, a Journal and a Settings button are presented so the player is able to, respectively, access their character’s information (see figure 29), check their Inventory for items and equipment, check their Journal for their current quest information, and finally open the settings menu to quit their game.



Figure 29 – Upon clicking the Character button, the player is greeted with a panel of character information (Inventory and Journal panels are also accessible from here)

Lastly, the combat scene has a HUD of its own which was inspired in other games (see figure 30) allowing the player to keep track of his character’s health, mana, the enemy’s health, what effects are affecting his character and what effects is he affecting the enemy and, most importantly, perform actions in the combat such as: Attack; Skill; Spell; use Item.



Figure 30 – Combat HUD which allows the player to keep track of various kinds of information as well as attacking the enemy

Initially the modelling process of the characters was quite difficult and frustrating because the author was designing models without reference sheets and more of a block and cubical style, leading to some very questionable model quality (see figure 31).

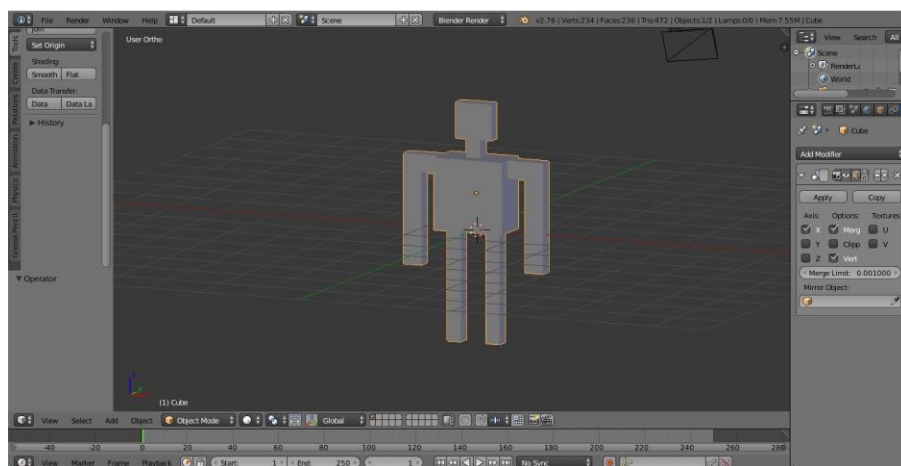


Figure 31 – Early concept of a character model

After some thorough experimentation and investigation, modelling from reference sheets resulted in much more inspiring products that, even though they were not close to the final result, had a much better visual look (see figure 32).

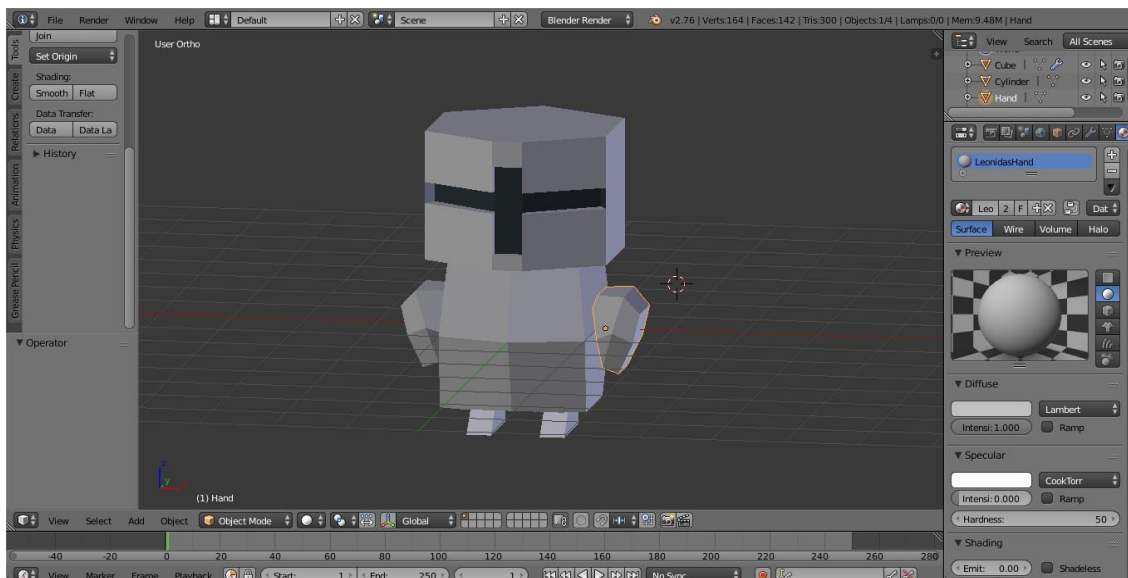


Figure 32 – Leonidas the main NPC of the game

The visual art style was chosen for being one of the most common art styles that follows box modelling, a combination of low poly models with the flat shading lighting technique. Low poly means low polygon, it refers to when a model is built and shaped with a very low polygon count, meaning the resulting model will be very less demanding and more efficient when compared to high poly models. Because box modelling usually results in more block-shaped models (as well as low poly models), the flat shading lighting technique fits perfectly. Flat shading is a lighting technique that involves shading each polygon of an object based on its normal and the direction of the light. By abandoning any attempt at smoothing and emphasizing the reduced low polygon count and more blocky art style, it's possible to produce some very interesting colourful results. The idea for this art form was inspired in an indie game called QuickHunt[73] as seen on figure 33.



Figure 33 – Comparison between Adventure! The Paladin Order's art style (left) and QuickHunt's art style (right)

Where QuickHunt had some more striking bright colours with a bigger emphasize on bright lighting, Adventure! The Paladin Order went with a darker colour inspired art style with a less

strong more soothing lighting. Several concept art obtained from the internet inspired this model style as well as the visual art style, and provided a much better appealing look, as seen on figure 34 and 36. A Link[74] inspired concept art was used to develop the main character of the game as seen in figure 35.



Figure 34 – Link inspired concept art to be used as a reference sheet for 3D modelling the main character



Figure 35 – Main character on the game inspired in Link

Whereas a knight reference sheet was used to make one of the more important NPC characters in the game, as seen in figure 37.



Figure 36 – Concept art for a knight which was used initially as a reference sheet for the design of a character

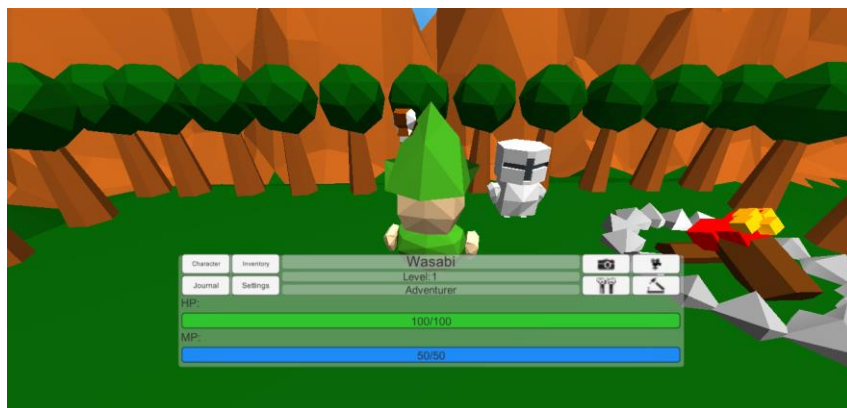


Figure 37 – Leonidas with the main character in the world

The Mages which were a mix of both the main character and the knight's concept idea, with changes to colouring and accessories, while the Ogre was a complete overhaul and scaling of the other characters with more defining colours and expressions, as seen on figure 38.



Figure 38 – Main character with the personality Mages and the Ogre that represents the tutorial combat

The whole overworld space and combat scenario were created in this style same style for visual consistency, as seen on figure 39 and 40.



Figure 39 – Overworld model design within the game in the same inspiring art style



Figure 40 – The combat scene (without user interface) also made in the same visual art style in a consistent matter

3.2.5 Future development

Technically speaking the project itself is lacking some improvements. In regards to the character creator, the concept should change in the future from choosing from a group of pre-determined alignments to allowing the player to write his own type of alignment and then give him a specific amount of points he could assign to personality traits creating his own personally customized alignment. This could even be further expanded by integrating a levelling system where each time a character levelled up, the player could ever so slightly boost or prioritize a personality trait of his own liking, allowing him/her to effectively start the game with a blank canvas if he/she chose to not assign any points at the time of creating the character.

The dialogue system has some few bugs and tweaks that could be fixed: right now there are some situations where if the player clicks multiple times in the talk key while there are options displaying then multiple options are generated in the same place. Some code could be optimized and cleaned up to make a clearer understanding of how the dialogue process interacts with the interface, and right now some functions have “work-arounds” to solve specific problems. For example the “multiple options” bug was solved by having a work-around that checks the interface to see if there are any options already displayed – this should be changed to checking if the options of that message have been displayed yet or not. Complementary, the event system can be expanded to accommodate the idea discussed earlier where a character holds an event instead of just comparing personality traits to check if they can interact or not. An alternate solution to this problem would be having a requisite event on the dialogue file itself so as to make it easier to configure which events the player character has to meet the requirements for. Having an event on the file is a great idea because it is very flexible but doesn’t limit the game’s ability to modify the event in any case that it is necessary to do so.

The visual aspect of the game has some limitations, a rework of the interface needs to be done with additional art integrated within the game, loading screens are still using external art and some points of the game, for example when loading a combat scene, don't even have a loading screen to make the transition between the overworld area and the combat area. A bug in the game engine originated that the interface had to be made smaller since objects in the limits of the visual area are impossible to click on some situations, this issue was investigated and this seems to be a recurrent bug from the Unity's new UI.

Right now, the combat only changes the attacks according to its highest personality trait. This lacks originality as the behaviours only change according to the higher personality ignoring variations in other personality traits, this was done in order to simplify programming in this starting phase. Also in the current version each enemy needs to have its own enemy script and respective AI script associated to it and this does not scale up nicely. In terms of programming this could be optimized by having a generic EnemyInformation script that holds the enemy information (health, mana, stats, equipment, attacks, etc...) as well as a generic EnemyAI script. Then, to create a new enemy, it would be as simple as adding a copy of the Enemy script (already available in the current project), and then associate it to the respective EnemyInformation script and respective EnemyAI script. Through this method, implementing AI becomes independent of whatever the developers decide to do and not only limited to personality behaviours.

The third-person camera in the game, even though not explored in this report, has some issues when it comes to having models in front of it. The camera was supposed to move forwards towards the player in the event that a model was in front of it but, for some unknown reason to the author, such does not happen and should be fixed in future developments of the project. The camera also raises some really unexplainable exceptions that, from what the author was able to investigate, is also a known recurring bug in Unity when trying to instantiate objects that are all initialized at the same time.

Of all the points enumerated earlier the most important part for the project is to simplify the way the game has been coded in a way that makes dialogue simple, understandable and highly scalable as a script, as well as finishing preparing the basis for features such as classes and levelling systems which right now only exist as interface text.

3.3 Beta evaluation

The beta phase, in the context of the project, served to ascertain if the resulting project had the desired impact or not. This time the testing group was of 10 members, which were submitted to test the project as it is right now, with questions being specifically designed towards understanding whether the group understood both the personality traits role in the game, as well as how the player decisions impacted the game.

The following chapters explain the methodology of the beta test more extensively and what the results and conclusions were from this testing phase.

3.3.1 Methodology

Both the alpha and beta tests started with a product awareness chapter whose objective was to measure what was the opinion of the testers from the concept to what was in fact developed. This was done initially to realize how interesting the conceptual idea was, while in the beta phase the purpose was to understand how captivating the resulting project and then perform a comparison between the tests to realize which parts failed or were successful and why.

Regarding the game design part itself, the beta test had three distinct chapters: character creator; overworld scenario; and combat system. The character creator asked questions about: the amount of alignments and background available to understand whether predetermined alignments and backgrounds were too restraining or if the problem lied in the amount of alignments and backgrounds offered, as well as to hear the tester's personal opinion; the explanation of the personality trait system to understand if they in fact understood what the personality system was all about so they were able to go in to the game understanding the concepts they were about to be presented with; and how captivating the whole feature seemed to validate if the initial presentation (from the alpha testing) compared to this solution (character creator) was interesting or not. There was also an open answer question available if any tester felt the need to contribute more information or comments to the whole experience with the creator.

The overworld scenario chapter explored: how easy to use were the interface controls to validate if there were any issues in the interface between the player and the user when testing for the game; if the testing group understood why some characters interacted with their own character but some didn't, this was to implicitly understand if the group understood how some characters have personality requirements or not; if they felt that their decisions impacted the characters around, to comprehend whether or not the reactions were impactful or if they needed more relevance and emphasis; and finally, whether the outcomes, the reactions, they got for their character if they fit their behaviour, because it's not only important to understand if the reactions and responses from characters around the world are impactful, but also if they are meaningful in the context of the character the testers have. Ending this chapter there is an open answer question, in the case that the testers felt like they needed to add more information to how their experience or for any comment they would like to do.

The last chapter investigated the combat system in an attempt to understand if what was developed was on the right step to what we feel should be an adaptive combat system. The first question focused on how entertaining the combat was: yes it's important to have a combat that is adaptive but most of all it's necessary to ensure that it's fun for the player regardless of how adaptive it is. The second question tackles how adaptive the fight was, as obvious as it seems it's also necessary to ensure if the enemy did, in fact adapt and change with the passing turns to the player's actions. Lastly, it was also important to realize if the players understood that their decisions in the world outside of the combat did impact the combat itself, as this was also one of the key points when developing the system. The remaining question was meant to

garner more suggestions and recommendations as to what course should the combat take in the future.

Except for the open answer questions, all other questions had answers of ratings from 1 to 5 where 1 represents that they don't agree with the questions or dislike a situation and 5 that they are absolutely satisfied with a situation.

3.3.2 Results

The first results in the beta form are the ones relative to the product awareness of the project as presented in figure 41:

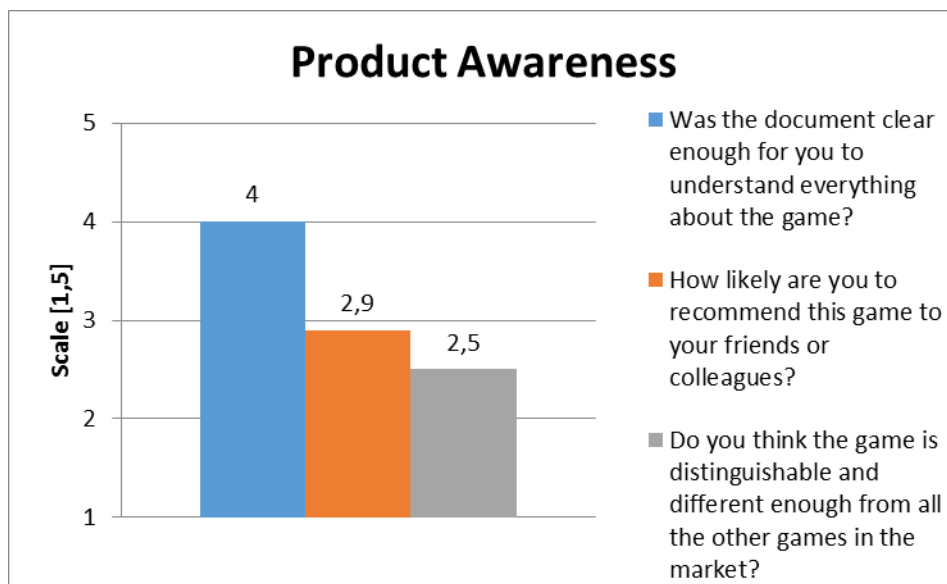


Figure 41 – Beta testing phase product awareness results

Compared to the alpha testing phase it is noticeable that there was a slight decrease in the every question. The clarity of the document had a very slight decrease from a 4,4 to a 4, recommendation of project fell from a 3,5 to a 2,9 and whether the project was distinguishable or not went from a 3 to 2,5.

In terms of the character creator results were gathered as presented on figures 42 and 43:

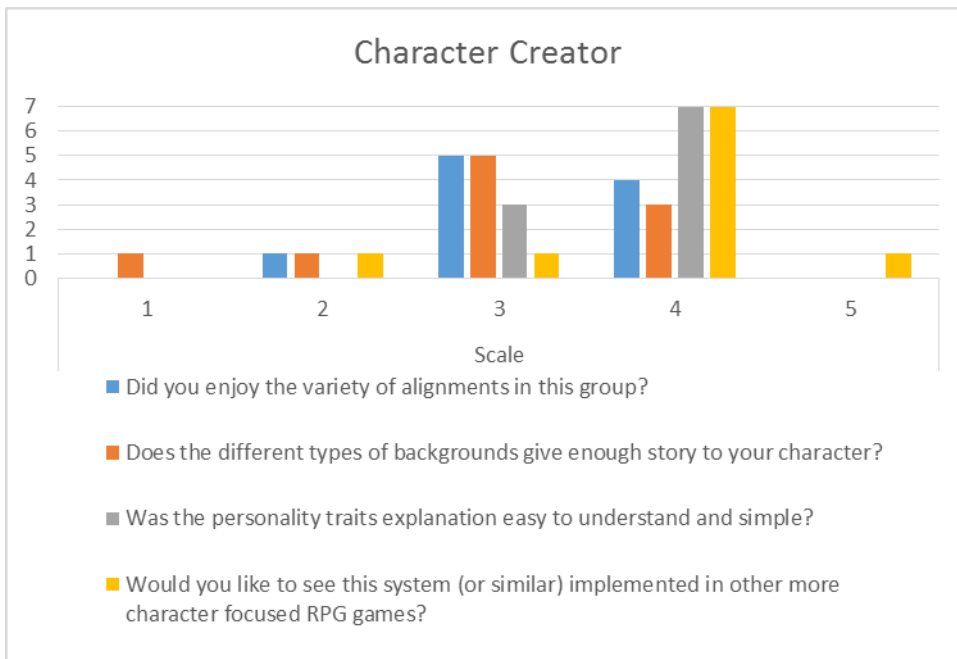


Figure 42 – Results for the questions related to the character creator

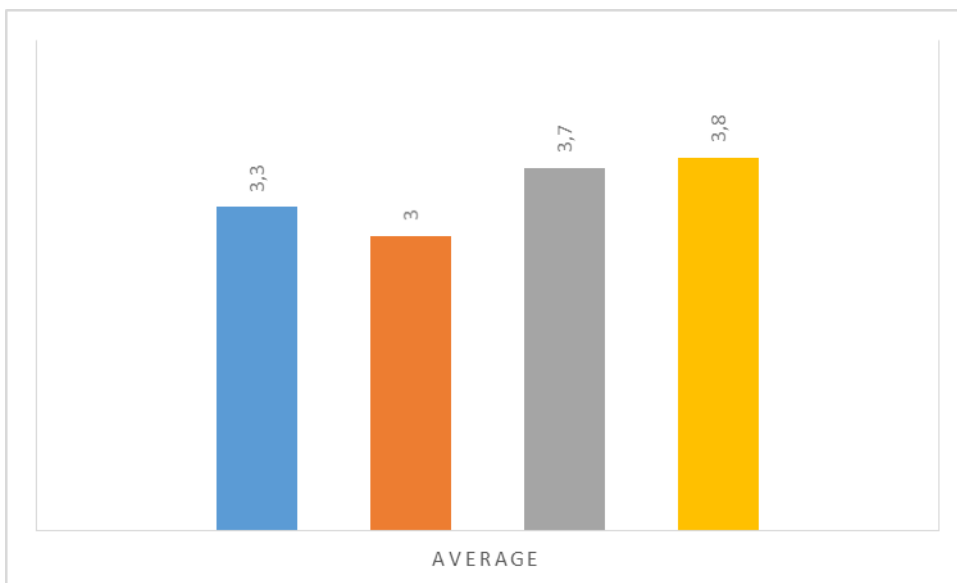


Figure 43 – Average rating of responses (scale goes from 0 to 5)

On average most questions garnered an average positive result. Testers were satisfied with the variety of alignments in the game, but didn't feel that there were enough backgrounds to create the character they pretended to envision, they also thought the personality trait system was easy to understand and simple, with most testers being interested in seeing this system in other RPG games.

The overworld scenario had a not so positive review from the testing group, as figures 44 and 45 can ascertain:

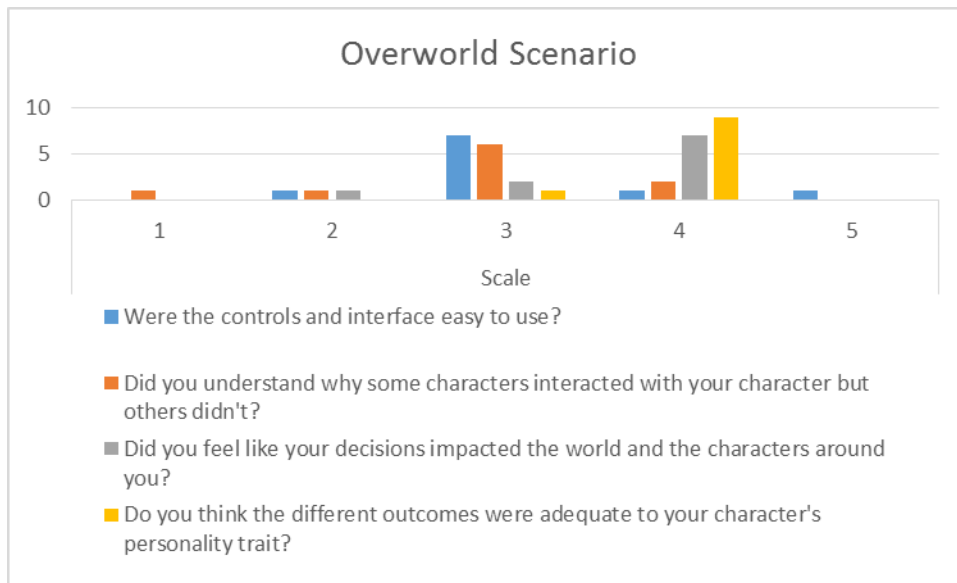


Figure 44 – Results from the responses of the overworld scenario questions

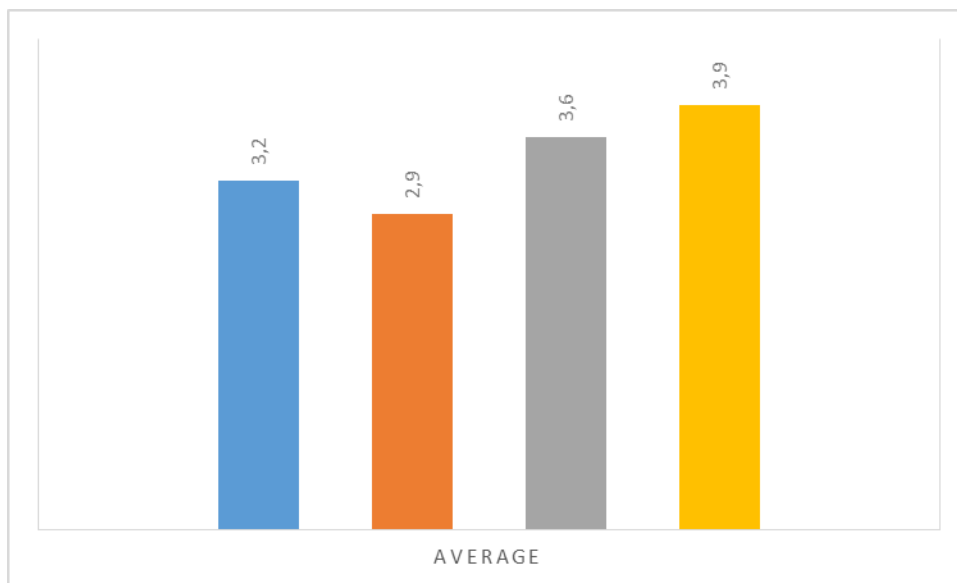


Figure 45 – Average results from questions of the same chapter (scale goes from 0 to 5)

Testers were slightly less satisfied with the result with the overworld scenario with some questions gathering some 1 and 2 ratings, overall there was a positive attitude towards the scenario and complaints were more of the visual nature.

Last but not least is the evaluation of the combat system had mixed results that can be seen on figures 46 and 47:

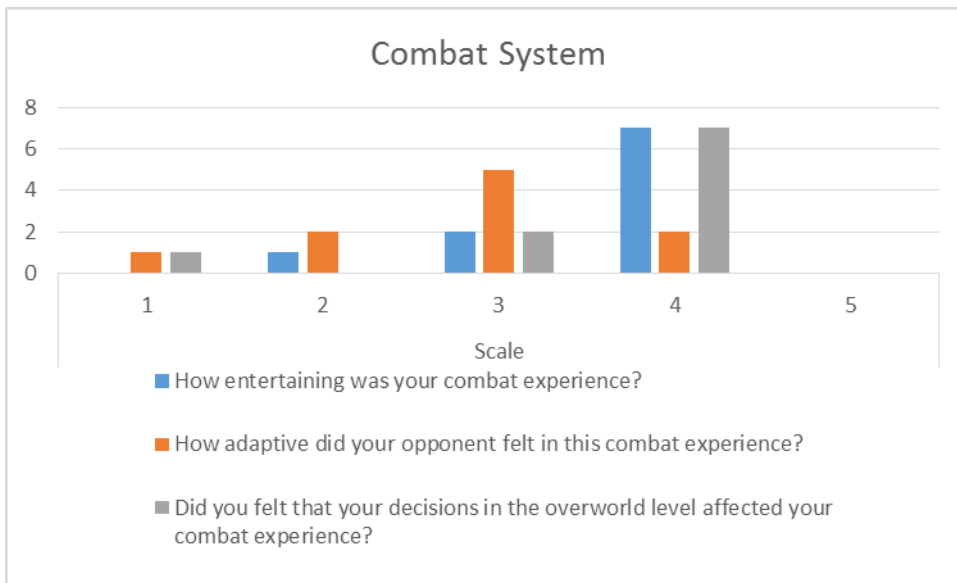


Figure 46 – Results from the last chapter regarding the adaptive combat system

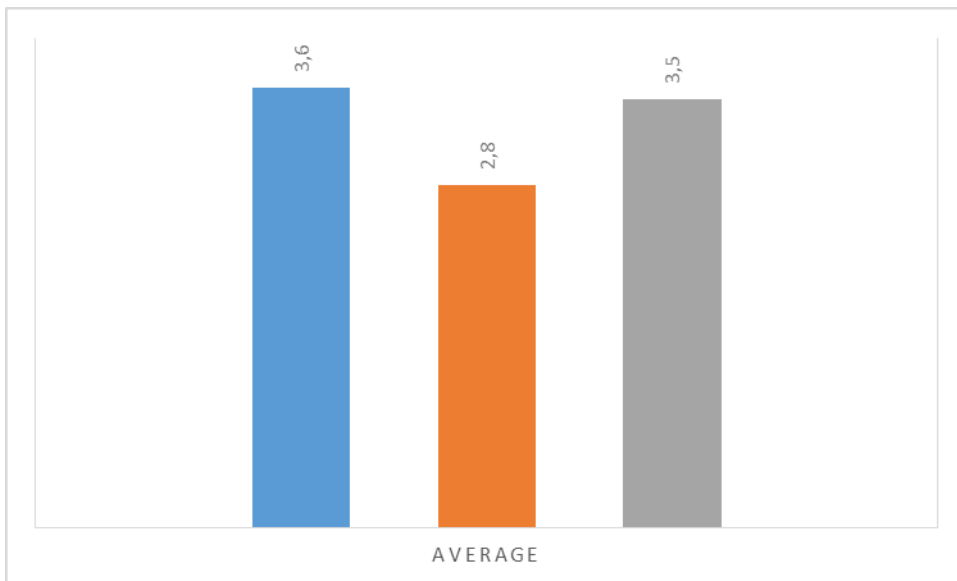


Figure 47 – Average of the results made for each one of the three questions regarding the combat system (scale goes from 0 to 5)

The combat system had the worst responses so far, the average tester felt that they were entertained by the battle, that they opponent felt adaptive, and that their decisions outside the combat matter but, by far, this group was the one who received the most less positive ratings as well as the most critics.

3.3.3 Conclusions

In conclusion, the average amount of responses was neither extremely positive nor absurdly negative (the average being a mild rating of 3 out of 5). From the analysis of the results of the

product awareness it's possible to conclude that people had a certain ideal of what the game was going to be, and were somewhat disappointed with what they ended up receiving, or the project was at least not up to their expectations.

Regarding the specific chapters themselves, the main critique of the whole project was, without a doubt, the visuals of the game. The fact that the game is using bland, default Unity UI was a big critique to the game in general, whereas in the character creator the main fault fell to the point that there weren't enough alignments for the testers to choose from, nor enough backgrounds, in general a deeper level of customization was demanded.

In an interesting twist suggestion, two testers suggested a point-and-click point of view for the game allowing for a more personal and close interaction with either the characters or objects in the world, this would be a great feature to implement in the future because it allows the players to take specific decisions that are not limited only to dialogue. All other complaints can be summarized as the lack of visual feedback in interactions, some players did not understand very well why some characters refused to talk to them, and didn't also understand fully how situations varied from one another, because there was a lack of visual feedback from the game, both dialogue and animations wise.

The combat system itself also garnered some critique in the visual department, some testers did not notice that the enemy performed different attacks according to their personality and their actions because, once again, of the visual feedback from the game was basic (text in boxes). Some testers suggested that weapons should have behaviours or maybe even personalities of their own, which is interesting in the sense that the game is prepared to be developed this way. Other groups of testers wanted different skills according to the character's behaviour as well, this has also been planned but was not implemented in the end.

In conclusion, it's possible to affirm that, given the circumstances, the project itself was not rejected by the testers group yet it wasn't worshipped either however this was not unexpected since the most criticized part for the group was the animations, interactions, story and more real game development and not the focus of this thesis that was the adaptive design and the framework itself. The feeling the author gets after reading such criticism is that, with proper visuals and presentation of the quest, the results would've been much more positive and the group would have had a much easier time to understand the project's concept. It's possible to conclude that the game is an interesting project that players are willing to play and explore more, if it is to be done by experienced developers that can provide a coherent game with more detailed visuals evolved story, and improved game mechanics.

4 Conclusions

In the end it is safe to assume the project has achieved quite an interesting prototype phase with a framework that has potential to become so much more. Granted that from what was thought of initially, the whole concept of the project changed drastically but, it's also true that the project as it is now, has a lot of consistency and basis for real-world game development. With a bit more of refinement of the game's adaptive personality and dialogue system, it is ready to enter a development phase where the actual game content and appearance are the main focus. The personality trait complemented with the dialogue system allows for a customization of a character's behaviour and the in-game decision dialogues enables characters that can be shaped throughout an actual adventure which always remained as the original points of interest for this dissertation. What remains now is then to actually develop the game's content.

4.1 Future work

When talking about future work it becomes easier to start talking about features from the whole experience since of the beginning of the game until the end of it. From the main menu there was meant to be an "Options" button where the player could change several settings such as the screen's resolution, anti-aliasing, sound settings, keyboard hotkeys, and others. Because Unity already has a graphics and input editor on game startup this feature was discarded. While still in the main menu screen, the game was also meant to have a "Load Game" button that provided a feature where the player could load his last saved state in the world and come back to the game to the whole experience he was in, this feature would also exist inside the game itself coupled with a "Save Game" feature.

In the character creator itself as studied from both the alpha and beta version, the ideal scenario would be where the player could choose from a predetermined group of alignments with a broader combination (lawful good, neutral good, chaotic good, lawful neutral, true neutral,

chaotic neutral, lawful evil, neutral evil, chaotic evil) that would give a predetermined behaviour to his/her character. Additionally, there would be a customization feature which meant that the player would have access to a limited amount of personality points he could spend in each personality trait, creating his own alignment (and therefore character behaviour) as he/she see fit.

The backgrounds would also suffer the same remodelling as the alignment issue, where there were a group of backgrounds that boosted different stats but there would be an option where the player could have an unspecified background that provided no stat boost and the player could then choose himself where he would like to spend his stats. This could either be done by giving him a limited amount of points to spend or by setting the character's level lower for when choosing the unspecified background, this would mean that the character levels up a lot faster giving more flexibility to where to spend points in.

Within the game itself there are many features who were left unimplemented because they did not directly contribute to the development of the adaptive framework. A levelling system where the player could learn new skills with experience points gained from defeating monsters and completing quests was lightly designed, but not developed. The abilities the character learns and how he progresses through the system did not depend on the character's personality but instead on its class. Speaking of which, a class system was also planned for the game that would give character specialization. Classes were unlocked to a character based on their personality (a pacifist character has access to the Healer class where a violent character does not) and he/she would only be able to wield certain weapons and shields if he belonged to a certain class. To be part of a class a character must fulfil the classes respective personality requirements, and even then would have to undergo that classes specific questline to further solidify its personality.

The questing system had some features left unattended such as: providing rewards on completion; a completed quest history; a real-time objective updates; and both globalized and localized personality questing. The plan here would be to provide features that players have come to expect from these kinds of games, except for the case of the globalized and localized personality questing which refers to grouping character in an area according to the quest's localization. This is a feature specific to this project and it would be useful in a scope where the game would take bigger proportions, providing a way for the developers to specify whether a quest was local or global. Local quests mean that only characters from a given location are affected by how the quest is done and finished, while a global quest means every character in the world has to change and react to how the quest was done. This doesn't happen in the project itself, all because the solution the project uses to reveal how adaptive the experience is shifted from rewarding a completed quest to, have characters provide rewards as a result of the character's behaviour. As it has been suggested by the tester teams, adding a point-and-click point of view to the game would give a much more detailed and personalized experience to the game. By having point-and-click the developers are able to add a new layer of specification to the game, actions are more specific since players are allowed to deliberately click and interact with specific items the way they want. This poses another issue because it can generate a lot of complexity, interaction-wise, but with time and investigation there is surely a

simple and efficient way to implement this functionality, without adding too much complexity to the behaviour of the characters, especially if the interactions generate the events already developed in the project.

There is a lot of art missing for the game. The main menu requires a background art, as well as for buttons and their background panels, in the main menu, inside the game and in the combat. Almost every UI related aspect was left unattended as it implied designing visual art that did not directly contribute to the project's objective of adaptive design. The art style has been defined as low-poly box modelling so it is important to develop all these key visuals (buttons, panels, etc.) in the same style to maintain integrity between the visual aspects of the game.

In the combat system itself, after the whole experienced was developed, the plan was to have the character start with only one or two attack and have a simpler weaker enemy he could fight with. The initial idea was to remain where the player could not be too strong, have too much health or deal too much damage to enemies of his level range, because if so it would overpower the AI or make the experience too repetitive, effectively eliminating the adaptive experience of the combat system itself. Skills and spells would rely on the type of class the character belonged to, and he would be able to unlock them through the levelling system explained earlier and use them in the combat itself. As of now the game has attack type resistances, which was planned and developed to demonstrate how different each weapon is and how they provide a different combat experience.

In a similar fashion, magical resistances were also planned for the game, these were not only going to be associated with skills and spells but also with the weapon themselves. Skills would vary in damage depending on the weapon equipped, it's elemental status (if it had one), and the elemental resistances of the enemy AI. There were 8 elemental resistances planned: Fire, Water, Ice, Earth, Lightning, Light, Dark, and Neutral. The AI was to be explored further, where the AI would have three (or more) different phases. The first phase implied the AI read the behaviour the player character brought in to the fight and then would change its state to it, for example if the player was a pacifist and went in to the fight with that as his/hers defining personality, then the AI would start the fight would less threatening moves if the AI was also of a pacifist nature. If the opposite happened, the player character is majorly violent and the enemy AI is of a pacifist nature, then it would begin the fight with the most aggressive moves it knew. The remaining phases are complementary to this behaviour, the AI would favour different types of attacks depending on how the player would proceed in the fight, for example if the player heals the enemy, and the enemy is of a pacifist nature, there would be a meter where if the player healed the enemy for a good enough amount of health the battle would end with the enemy refusing to battle the player or some other way to progress the quest, this would happen regardless of the player's major defining personality trait as it refers to a battle specific behaviour – although a violent player against a pacifist enemy would have a harder requirement to meet in order to finish the fight this way. This was not added in the final project because it meant implementing a too complex behaviour, albeit interesting, due to time constraints and lack of development skills.

Last but not least, actual implementation of game content is what is missing. Comparing to the initial approaches it is true that there were a lot of features left unattended, mostly because of knowledge limitations or time. The game's content is also severely lacking. According to the initial plan, the game is meant to have a paladin order keep with tutorial that both set the stage for the story as well as introducing the game to its adaptive design. Several other quests, which were presented on the alpha testing and planned for development were left unattended as the story was not implemented as well.

Despite all these flaws pointed earlier, it is a fact how this project remains a true testament on how to develop a system who is at the same time simple to use but very highly scalable with a support for big open world adventures in a simple, well structured, way. For the author himself, this project was a huge experience, having never dwelled in a serious game development adventure, it provided a fantastic experience of which the author has learned much from and will certain carry it with him towards his future works.

Bibliography

- [1] “On average, how many video games are released each year, by platform? - Quora.” [Online]. Available: <http://www.quora.com/On-average-how-many-video-games-are-released-each-year-by-platform>. [Accessed: 26-Mar-2015].
- [2] “Twitch,” *Twitch*. [Online]. Available: <http://www.twitch.tv/>. [Accessed: 19-Oct-2015].
- [2] “When Games Are Rushed To Release Who Is To Blame? Hint: It’s Not The Testers,” *Forbes*. [Online]. Available: <http://www.forbes.com/sites/olliebarder/2014/11/17/when-games-are-rushed-to-release-who-is-to-blame-hint-its-not-the-testers/>. [Accessed: 05-Oct-2015].
- [4] “How Many Nights at Freddy’s Are Too Many? | Hardcore Gamer.” .
- [5] “There Are WAY Too Many Assassin’s Creed Games,” *Entertainment Fuse*. .
- [6] “The Biggest Video Game Delays Of 2014 - Forbes.” [Online]. Available: <http://www.forbes.com/sites/erikkain/2014/06/07/the-biggest-video-game-delays-of-2014/>. [Accessed: 26-Mar-2015].
- [7] “Illusive choice: Mass Effect takes the ‘role play’ out of RPG,” *VG247.com*. .
- [8] “Role-playing game - Wikipedia, the free encyclopedia.” [Online]. Available: http://en.wikipedia.org/wiki/Role-playing_game. [Accessed: 26-Mar-2015].
- [9] “The History of Computer Role-Playing Games Part 1: The Early Years (1980-1983).” [Online]. Available: http://www.gamasutra.com/view/feature/3623/the_history_of_computer_.php. [Accessed: 05-Oct-2015].
- [10] “The History of Computer Role-Playing Games Part 2: The Golden Age (1985-1993).” [Online]. Available: http://www.gamasutra.com/view/feature/1706/the_history_of_computer_.php. [Accessed: 05-Oct-2015].
- [11] “The History of Computer Role-Playing Games Part III: The Platinum and Modern Ages (1994-2004).” [Online]. Available: http://www.gamasutra.com/view/feature/1571/the_history_of_computer_.php. [Accessed: 05-Oct-2015].
- [12] “BioWare | Dragon Age.” [Online]. Available: <http://dragonage.bioware.com/agegate/>. [Accessed: 05-Oct-2015].
- [13] “Mass Effect.” [Online]. Available: <http://masseffect.bioware.com/agegate/?url=%2F>. [Accessed: 05-Oct-2015].
- [14] “Fallout.” [Online]. Available: <https://www.fallout4.com/age-gate>. [Accessed: 05-Oct-2015].
- [15] “Gamasutra: Gerald Belman’s Blog - On the stupidity of BioWare’s dialogue system.” [Online]. Available: http://www.gamasutra.com/blogs/GeraldBelman/20120913/177677/On_the_stupidity_of_BioWares_dialogue_system.php. [Accessed: 05-Oct-2015].
- [16] “The dialogue wheel: Why I hate it with a passion - Dragon Age II - Giant Bomb.” [Online]. Available: <http://www.giantbomb.com/dragon-age-ii/3030-30995/forums/the-dialogue-wheel-why-i-hate-it-with-a-passion-486305/>. [Accessed: 05-Oct-2015].
- [17] “Pillars of Eternity.” [Online]. Available: <http://eternity.obsidian.net/>. [Accessed: 28-May-2015].

- [18] “Approval,” *Dragon Age Wiki*. [Online]. Available: <http://dragonage.wikia.com/wiki/Approval>. [Accessed: 20-Feb-2015].
- [19] “Morality,” *Mass Effect Wiki*. [Online]. Available: <http://masseffect.wikia.com/wiki/Morality>. [Accessed: 20-Feb-2015].
- [20] “Janel Comeau: 5 Reasons Why Dragon Age: Origins is Still the Best Fantasy Game of All Time.” .
- [21] B. Crecente, “Dragon Age: Origins Preview: Violence, Lust and Betrayal,” *Kotaku*. [Online]. Available: <http://kotaku.com/5250623/dragon-age-origins-preview-violence-lust-and-betrayal>. [Accessed: 15-Oct-2015].
- [22] “Remind me again why Bioware dropped their excellently balanced Dialogue wheel from ME2? - General Discussion,” *The BioWare Forum*. [Online]. Available: <http://forum.bioware.com/topic/518601-remind-me-again-why-bioware-dropped-their-excellently-balanced-dialogue-wheel-from-me2/>. [Accessed: 24-Oct-2015].
- [23] “Dialogue Options | Dragon Age: Inquisition Wiki.” [Online]. Available: <http://dragonage3.wiki.fextralife.com/Dialogue+Options>. [Accessed: 05-Oct-2015].
- [24] “Let’s go gambling in New Vegas! | Page 2 | rpg codex > It’s finally Thursday.” [Online]. Available: <http://www.rpgcodex.net/forums/index.php?threads/lets-go-gambling-in-new-vegas.50824/page-2>. [Accessed: 24-Oct-2015].
- [25] *Fallout: New Vegas - Video Interview - IGN Video*. .
- [26] “Pillars of Eternity |OT| Go For The Eyes Again!,” *NeoGAF*. [Online]. Available: <http://www.neogaf.com/forum/showthread.php?t=1017619&page=6>. [Accessed: 24-Oct-2015].
- [27] “Disposition - Official Pillars of Eternity Wiki.” [Online]. Available: <http://pillarsofeternity.gamepedia.com/Disposition>. [Accessed: 04-Jun-2015].
- [28] “Mass Effect, Dialogue Trees, and Romance — Medium.” [Online]. Available: <https://medium.com/@videodante/mass-effect-dialogue-trees-and-romance-f332de52aa68>. [Accessed: 26-Mar-2015].
- [29] “Rude awakening: Why Fallout: New Vegas felt incomplete | flaming mac.” [Online]. Available: <http://flamingmac.com/2011/09/why-fallout-new-vegas-felt-incomplete/>. [Accessed: 26-Mar-2015].
- [30] “Could a simple mod fix a major problem people have with Mass Effect?,” *NeoGAF*. [Online]. Available: <http://www.neogaf.com/forum/showthread.php?t=506810>. [Accessed: 24-Oct-2015].
- [31] “Mass Effect 3’s Ending Disrespects Its Most Invested Players.” [Online]. Available: <http://kotaku.com/5898743/mass-effect-3s-ending-disrespects-its-most-invested-players>. [Accessed: 26-Mar-2015].
- [32] C. Pruett, “Balancing Game Mechanics and Narrative,” *Interface J. Educ. Community Values*, vol. 4, no. 6, 2004.
- [33] “Video Games and Storytelling - YouTube.” [Online]. Available: <https://www.youtube.com/watch?v=1jdG2LHair0>. [Accessed: 26-Mar-2015].
- [34] “Play Mechanics: Adapting Video Games For The Screen - ScreenCraft.” [Online]. Available: <http://screencraft.org/2015/03/02/mechanics-play-adapting-video-games-screen/>. [Accessed: 26-Mar-2015].
- [35] “Dishonored.” [Online]. Available: <http://www.dishonored.com/>. [Accessed: 26-Mar-2015].

- [36] “Dishonored Review | STFUandPlay.” [Online]. Available: <http://www.stfuandplay.com/story/content/dishonored-review#.ViuTiPmrRqM>. [Accessed: 24-Oct-2015].
- [37] “The Golden Cat,” *Dishonored Wiki*. [Online]. Available: http://dishonored.wikia.com/wiki/The_Golden_Cat. [Accessed: 05-Oct-2015].
- [38] “Morgan Pendleton,” *Dishonored Wiki*. [Online]. Available: http://dishonored.wikia.com/wiki/Morgan_Pendleton. [Accessed: 24-Feb-2015].
- [39] “Before You Start... Tips For Playing Dishonored The Best Way.” [Online]. Available: <http://kotaku.com/5950016/before-you-start-tips-for-playing-dishonored-the-best-way>. [Accessed: 26-Mar-2015].
- [40] “Fallout: A Post Nuclear Role Playing Game.” [Online]. Available: http://www.nma-fallout.com/fallout1/official_site/. [Accessed: 05-Oct-2015].
- [41] *Computer Gaming World*. Golden Empire Publications, 1998.
- [42] P. Lai, “Fallout: Rebirth Through Nuclear Holocaust.”
- [43] A. G. on O. 08 and 2012 at 12:00a, “Dishonored review: Crime and punishment,” *Polygon*. [Online]. Available: <http://www.polygon.com/2012/10/8/3440792/Dishonored-review>. [Accessed: 05-Oct-2015].
- [44] D. W. P. 11/10/2013 V. tested X. 360, “Dishonored review,” *Eurogamer.net*. [Online]. Available: <http://www.eurogamer.net/articles/2012-10-08-dishonored-review>. [Accessed: 05-Oct-2015].
- [45] “Dishonored Review,” *Giant Bomb*. [Online]. Available: <http://www.giantbomb.com/reviews/dishonored-review/1900-529/>. [Accessed: 05-Oct-2015].
- [46] “Hindi and the origins of chess,” *Chess News*. [Online]. Available: <http://en.chessbase.com/post/hindi-and-the-origins-of-chess>. [Accessed: 05-Oct-2015].
- [47] M. Barton, *Dungeons and Desktops: The History of Computer Role-Playing Games*. Taylor & Francis, 2008.
- [48] “Across the Board Games | Reviews, Kickstarter Updates, Community Discussion.” .
- [49] “Basic Rules for Dungeons & Dragons | Dungeons & Dragons.” [Online]. Available: <https://dnd.wizards.com/articles/features/basicrules>. [Accessed: 05-Oct-2015].
- [50] “Official D&D 5E Adventurer’s League Character Sheet and Log - Cropped.” [Online]. Available: <http://www.enworld.org/forum/rpgdownloads.php?do=download&downloadid=1052>. [Accessed: 24-Oct-2015].
- [51] “MAGIC: THE GATHERING,” *MAGIC: THE GATHERING*. [Online]. Available: <http://magic.wizards.com/en>. [Accessed: 05-Oct-2015].
- [52] “Quarterfinals: Running the Tables with Paul Rietzl : Daily MTG : Magic: The Gathering.” [Online]. Available: <http://archive.wizards.com/Magic/Magazine/Article.aspx?x=mtg/daily/eventcoverage/ptpar11/qf1>. [Accessed: 24-Oct-2015].
- [53] “CARD SET ARCHIVE - PRODUCTS - GAME INFO,” *MAGIC: THE GATHERING*. [Online]. Available: <http://magic.wizards.com/en/game-info/products/card-set-archive>. [Accessed: 05-Oct-2015].

- [54] “MAGIC ONLINE - PRODUCTS - GAME INFO,” *MAGIC: THE GATHERING*. [Online]. Available: <http://magic.wizards.com/en/content/magic-online-products-game-info>. [Accessed: 05-Oct-2015].
- [55] “The Official Pokémon Website | Pokemon.com | Explore the World of Pokémon.” [Online]. Available: <http://www.pokemon.com/us/>. [Accessed: 05-Oct-2015].
- [56] “【速報】『ポケットモンスター X・Y』は 2013 年 10 月 12 日 (土) に世界同時発売！【Nintendo Direct @ E3 2013】,” *ファミ通.com*. [Online]. Available: <http://www.famitsu.com/news/201306/11034770.html>. [Accessed: 05-Oct-2015].
- [57] “Battles,” *Pokémon Essentials Wiki*. [Online]. Available: <http://pokemonesentials.wikia.com/wiki/Battles>. [Accessed: 24-Oct-2015].
- [58] “FINAL FANTASY XIV.” [Online]. Available: <http://www.finalfantasyxiv.com/>. [Accessed: 05-Oct-2015].
- [59] “Final Fantasy 7 Playstation Used,” *playstation 3 spiele ab 3*. [Online]. Available: <http://playstation-3-spiele-ab-3.weebly.com/1/post/2014/10/final-fantasy-7-playstation-used.html>. [Accessed: 24-Oct-2015].
- [60] “Businesses | SQUARE ENIX HOLDINGS CO., LTD.” [Online]. Available: <http://www.hd.square-enix.com/eng/group/index.html>. [Accessed: 21-Oct-2015].
- [61] “Turn-based strategy - Wikipedia, the free encyclopedia.” [Online]. Available: http://en.wikipedia.org/wiki/Turn-based_strategy. [Accessed: 26-Mar-2015].
- [62] “Final Fantasy V - Wikipedia, the free encyclopedia.” [Online]. Available: http://en.wikipedia.org/wiki/Final_Fantasy_V. [Accessed: 26-Mar-2015].
- [63] “Famed Mimic Gogo,” *Final Fantasy Wiki*. [Online]. Available: http://finalfantasy.wikia.com/wiki/Famed_Mimic_Gogo. [Accessed: 24-Feb-2015].
- [64] “Quintus JavaScript HTML5 Game Engine.” [Online]. Available: <http://www.html5quintus.com/>. [Accessed: 05-Oct-2015].
- [65] “What is Unreal Engine 4.” [Online]. Available: <https://www.unrealengine.com/what-is-unreal-engine-4>. [Accessed: 05-Oct-2015].
- [66] “Unity - Game Engine.” [Online]. Available: <http://unity3d.com/pt/>. [Accessed: 05-Oct-2015].
- [67] “Unity - Store.” [Online]. Available: <https://store.unity3d.com/pt-BR/products>. [Accessed: 05-Oct-2015].
- [68] “8 awesome options for 3D modeling software,” *The Creative Edge*. [Online]. Available: <https://99designs.com/designer-blog/2014/10/14/3d-modeling-software-guide/>. [Accessed: 21-Oct-2015].
- [69] “3d Modeling & Rendering Software | 3ds Max 2016 | Autodesk.” [Online]. Available: <http://www.autodesk.com/products/3ds-max/overview>. [Accessed: 05-Oct-2015].
- [70] “3D Animation And Modeling Software | Maya | Autodesk.” [Online]. Available: <http://www.autodesk.com/products/maya/overview>. [Accessed: 05-Oct-2015].
- [71] “blender.org - Home of the Blender project - Free and Open 3D Creation Software,” *blender.org*. [Online]. Available: <https://www.blender.org>. [Accessed: 05-Oct-2015].
- [72] “The Elder Scrolls Official Site.” [Online]. Available: <http://www.elderscrolls.com/>. [Accessed: 21-Oct-2015].

- [73] “Twirlbound | QuickHunt.” [Online]. Available: <http://www.twirlbound.com/game/QuickHunt>. [Accessed: 14-Oct-2015].
- [74] “Link.” [Online]. Available: <http://play.nintendo.com/themes/friends/link/>. [Accessed: 14-Oct-2015].

Annex A

Testers Invitational Form

Adventure! The Paladin Order

Alpha Phase: Conceptual Tests

Game Summary

You are a villager, a nobody stowed away by society. Without a family or anyone to support you, you then decided to throw away your life in to the Paladin Order of the Sun an order dedicated to cleanse the world of evil forces. With this video game we wish to provide the players community with a geeky, funny and visually cute experience, satisfying the more casual players, but also reaching out for the more experienced fans who miss the days of Final Fantasy 7 and all of the other turn based games.

About Us

We are a pair of students from Portugal who are currently creating a video game for their master's thesis on Computer Sciences in the specialization of Multimedia and Graphical Systems at the Instituto Superior de Engenharia do Porto.

Game Design

With every passing day there is an increasing number of people playing video games, either due to the evolution of technology towards mobile systems or thanks to the fact that technology is becoming gradually more accessible to everyone. With an ever increasing and differentiated group of players, it becomes more and more important to develop video games that are accessible, challenging, discernible and modular independently of the group of players and the environment in question. Therefore, with this project, we wish to develop a video game whose game design is focused on the decisions of the player. By play testing the game thoroughly throughout its different phases and by presenting different scenarios to the testers we can gauge whether the situation satisfies the conditions we established before or if there is something that needs to be changed. It is vital for the success of this thesis that the test group contains players from different social backgrounds with different expectations, as it is not possible to have a game who is accessible and challenging if we just test a group of highly experienced gamers, disregarding the other social groups. Game design wise, the game is an old school turn-based game with elements of exploration in between.

Social Features

Nowadays, social media contents and the large number of users in social networks are one of the most promising sources for the personalization of game play experiences. Furthermore, the invitation of social network friends to influence the game experience with new contents and participation in game plays can increase the awareness of players' activities outside the game and consequently raise the curiosity of the game to users that are not currently playing. This fact can also contribute to the creation of collaborative environments to improve the game quality of every participant. However,

social media integration can quickly evolve into a simple promotion system for the players who connect their games to social networks accounts, so developers need to carefully consider the benefits and disadvantages of social media as an opt-in experience.

Our players can choose not to use the social media plugin, but if they use, they may benefit with that. We do not force the users to connect to any social network to play and we also do not push users to share information in order to progress in the game. Our goal is to make players look and ask for social integration. Unlike most social sharing systems designed for games our share menu interacts with user in non-intrusive way allowing them to choose and share only the best game moments. With this menu players can share screenshots, videos and even game stories with friends using social platforms. Players can also take advantage of their social communities to improve their game experiences. They can get better game equipment using social craft system or create small parties to help them fighting in the combats with social party system.

Final product

At the end of the development stage we expect to have a playable demo for the game. This demo will contain the very first main quest with a dungeon to explore, and several side quests that heavily impact the world in which the players are in.

Game so far

In the following chapter we will talk a little bit about the state of the project so far. It is worth noting that the project is in a very structural phase, all you see here will not reflect the game in its final stage.

Overworld

The game will have a third-person perspective that the players can use to investigate the world around them. Mechanics wise, there is a smart camera in place, gravity and slide implemented as well as NPC interaction and treasure finding so far.

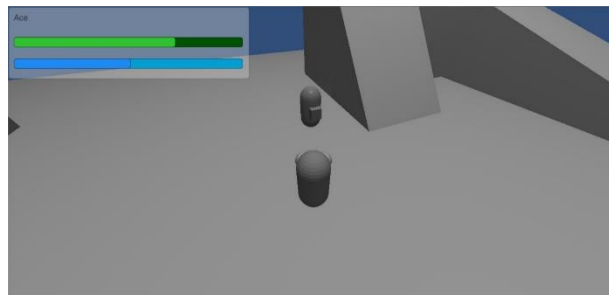


Figure 1 Character in Overworld space

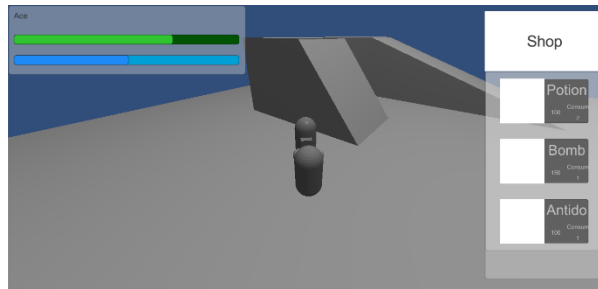


Figure 2 Character interacting with a Shop NPC

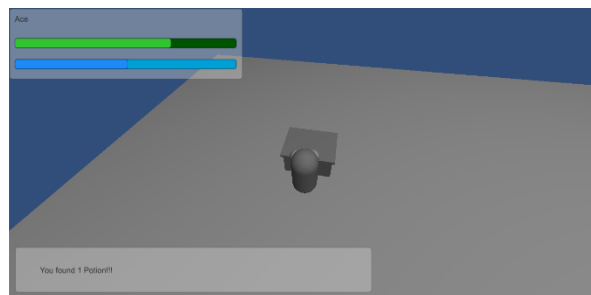


Figure 3 Character found an item!

Combat

Combat is made in a turn-base fashion with a highly intelligent AI. The enemies learn with the players' actions that are made each turn and adapt to defeat the player as quickly as possible.



Figure 4 Early representation of combat

Share Menu

To share videos, screenshots and game history information, all the player needs is a social network account. The players will have access to a menu (by clicking in a share button or share shortcut) similar to the following:

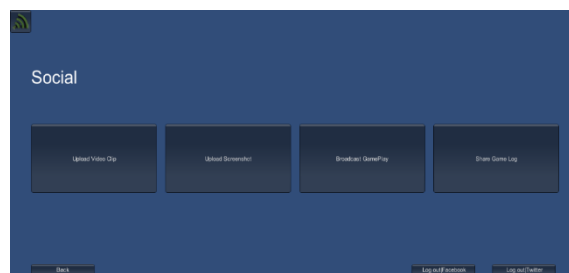


Figure 5 Draft of Share Menu

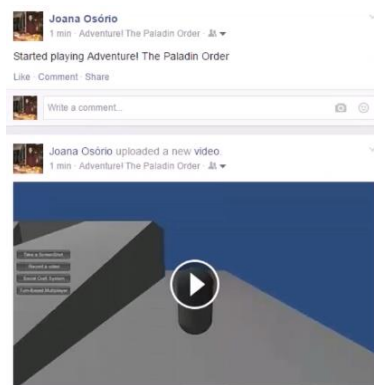


Figure 6 Share experiences on Facebook

Screenshots

The players can take screenshots while playing a game and upload them to social networks. To take a screenshot players have to press and hold Screenshot button (or screenshot shortcut) for at least 1 second.

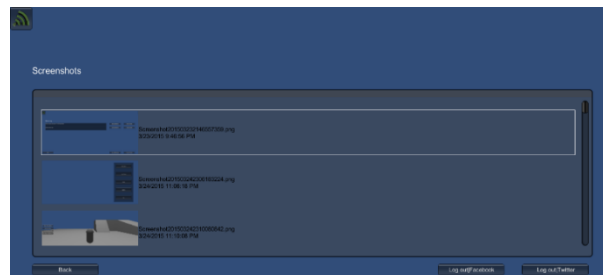


Figure 7 Draft of Screenshot Menu

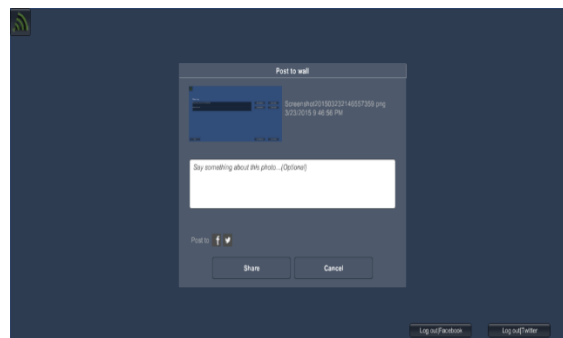


Figure 8 Draft of Share Screenshot Dialog

Videos

The players can record short video clips (1~15 minutes) while playing a game and upload them to social networks. To record players have to press and hold the record button (or record shortcut) for at least 1 second.

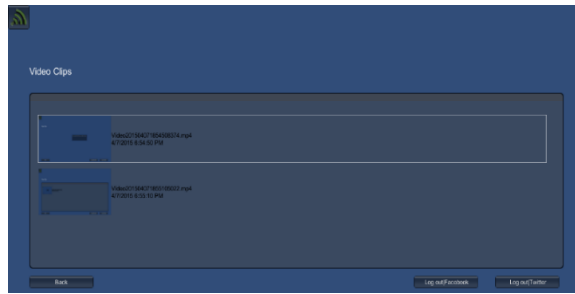


Figure 9 Draft of Video Menu

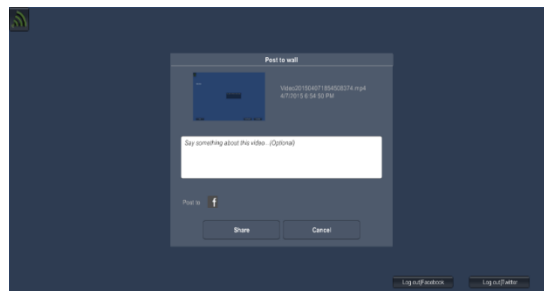


Figure 10 Draft of Share Video Dialog

Game Log

Players can share information about trophies and game events on social networks. This information is presented in a history board similar to the following:



Figure 11 Draft of Game Log

What is expected from you

When performing tests, these are divided in to two separate groups:

- Extensive tests;
- Short tests.

Extensive tests consist of testing big chunks of the game that normally require more time and a more in-depth feedback. Here are some examples:

- Newly implemented combat mechanics;
- Complex social features;
- Different engine quirks;
- Story and quest related adventures.

Short tests consist of testing short features of the game that are spontaneously added as a reflect to something that has been forgotten or fixed, or in the case of something that was suggested by one of the testers

and thus was modified or added after it was already implemented. Here are some examples:

- Bug fixes;
- Functionality corrections;
- Small improvements made based off testing feedback.

Furthermore we expect to receive as much feedback as possible via honest constructive criticism as well as reporting any bugs, errors or quirks that should be attended.

Functions to perform

As a tester, you are expected to participate in all of the different testing phases through which the game is going to pass through in its development. However it is fine if you find yourself in a position where you cannot perform a certain test and thus have to skip it.

This next paragraph is very important. Each testing phase will work differently from one another, some will only consist of filling questions from a form to help plan future features, while others we will ask you to perform a skype call with you (screen sharing is all we need) so we can assist, see and analyze your gameplay, your reactions and probably complains. This is what it's called a testing session, where you play and test a specific part of the game and we analyze and decide how to fix or improve it based on your experience.

In the case of which you don't feel comfortable in performing a testing session with us, we ask you to record a playing session of yourself testing and report (along with the video) where you had the most difficulty, what are your complains, as well as other aspects.

Form

If you have reached this chapter and you are still interested in becoming a tester for our game, please fill the form bellow with the information we ask you so you can be added to our testers list and we can contact you in the future.

Leave this field in blank if you want to keep as anonymous, this field is only required for university formal purposes.

Personal Name:

This next field is obligatory as we require some form of public identity to identify you.

Nickname*: _____

Provide us with some form of contact, the most accessible method we ask for is your email.

Your email*:

In this field please describe what your availability to perform these tests is, not only in a time perspective but also in any foreseen problems. We expect the development of the game to finish sometime around October so if there will be any instances in which you are lacking time or aren't available for some reason let us know so we can plan ahead.

Availability*:

If you have any additional comments you would like to add, please write them in this section.

Additional comments:

Contacts

After filling the form in the previous chapter, please email this document back to us so we can receive your information and add you to our list of testers. If you have any questions you'd like to ask before sending us this document, don't hesitate in emailing us.

Contact email: contactclockworkinc@gmail.com

Personal website: <http://tdmei.azurewebsites.net/>

Annex B

Alpha testing survey – Adaptive Game Design sections

Alpha Test Vo.01 - Conceptual Analysis

*Required

B. About the Story

The focus of this project is to develop a game that is highly adaptable of the choices that the player makes. This involves the story as well.

For this section we ask that you see each of the questions presented as a quest for the game, and then, imagine that your character is going through each and one of the situations and how YOU would choose to approach the situation.

1. Quest 1: Introduction to the Order

After being picked up on the streets, your character is adopted in to a highly militarized order known as the Order of the Sun. This order is responsible to maintain peace and justice around the world and houses many adventurers that wish to partake in the purging of evil. As the new member you are ordered to attend the initiation ceremony. You have heard stories about the initiation but you really don't know which rumors are true or false only that a select few are able to finish it. Now when the initiations begins all participants are placed in an open field, the Headmaster address the participants and greets everyone that wishes to join the order but, he will not acknowledge anyone for only the strongest are worthy to enter. With this in mind, he lets everyone know that the initiation is nothing more than a battle royale.

1.1. If, while you were playing the game, you were confronted with this situation what would you and your character choose to do? *

2. Quest 2: The Innkeeper's Daughter

As you walk through the keep's market, you glance over the various shops in display in town. You notice a different sound between the crowds, a rising argument between two people, a father, and a young daughter. The girl seems to be close to crying and the father seems to be highly intoxicated. You know that if you approach the situation, things will heat up between whatever party you decide to be involved with.

2.1. If, while you were playing the game, you were confronted with this situation what would you and your character choose to do? *

3. Quest 3: The Merchant's Dilemma

Returning from a mission you were sent to, you find two merchants arguing in the market area. One accuses the other of stealing him of clients, while the second one is sure the other has been bad mouthing him to people as to have clients not go to his store. As you listen to the discussion, both merchants get progressively more and more aggressive towards one another and you fear the situation might escalate in to physical assaults. You think that if there is ever a situation to intervene it should be now.

3.1. If, while you were playing the game, you were confronted with this situation what would you and your character choose to do? *

4. Quest 4: The Missing Child

You are sent to explore a cave and clear out what monsters lurk inside as they have been assaulting nearby farms. While in the cave, you notice a ghostly girl running around, in what seems to be guiding you along the way. Half-way through you find yourself in a seemingly open chamber with a small pool of water in the middle, the child sits on the edge with her legs within the water making splashes. It interacts with you and tells you the story of how she became trapped in the cave, her parents left her because they were unable to raise her due to being poor and locked her inside unbeknownst that the cave was a shrine to a goddess. She looks at you, and fears that you have come to rid of her, a battle ensues between monsters she summons and your character. After exploring the rest of the dungeon you reach a shrine of a goddess that vaguely reminds you of the child you just met and talked to. The spirit comes within the statue, still a child, and presents herself as the daughter of the goddess of the shrine. She tells you the story of how this shrine is her mother's resting place, and how her human parents abandoned her in this cave for fear of her godly powers. She then proceeds to explain how the people of the keep have forgotten about this place, and how this has affected the community itself.

4.1. If, while you were playing the game, you were confronted with this situation what would you and your character choose to do? *

5. Quest 5: Cleaning the Keep

The high order soldier who found you discusses with the headmaster how you are worthy of participating the initiation ritual, the headmaster does not seem convinced and decides to send you on a quest to clear the lower level of the keep of the recent infestation they have. This serves both as a test to your skills and to show that you are strong enough to enter the ritual. While exploring the dungeon you found out it's been infested by clockwork rats that seemed to be magically alive! When you reach the end you found the clockwork rat boss that wants to stop you from killing his brothers and begins a fight with you. When you win, the boss begs and implores you to not kill him as he and his brothers were created by the headmaster to be his pets up until one day he tossed them in to the catacombs leaving them to rot.

5.1. If, while you were playing the game, you were confronted with this situation what would you and your character choose to do? *

Alpha Test Vo.01 - Conceptual Analysis

*Required

C. Mechanics

It is equally important to study, analyze and develop solid game mechanics as they are the foundation of a game.

It is worth noting that the focus of this project is to develop a game that is highly adaptable of the choices that the player makes. This involves game mechanics as well.

1. Given that the focus of the thesis is having a user-oriented mechanic, during a combat, would you prefer having ONE difficult enemy that adapts to your combat choices, or multiple enemies that share a weakness that you must discover? *

- One strong adaptive enemy
- Multiple common weakness enemies
- Other:

2. Imagine you are placed in one of the previous situations, would you prefer progressing through the story through a fixed set of answers that provide different outcomes, or would you prefer having ways that your character can directly intervene in the situation without having to resort to dialogue? *

- Multiple choice dialogue progression
- Action based dialogue progression
- Other:

Alpha Test V0.01 - Conceptual Analysis

E. Suggestions and Feedback Ideas

We are always open to all kinds of feedback, in fact we encourage you to give us all of your ideas so we can further improve our project! Use this section to provide us with any new ideas we could implement or if you think we ought to do something in a different way.

1. What would you change?

From what you know of the game, what features would you change or do differently.

2. Are our future ideas appealing?

Given what was presented in this form that is not yet implemented in the game, would you suggest any changes or if you find these ideas unappealing how could we make them more interesting.

3. What would you like to see added to the game in the future?

If you have any ideas of your own that you would like to see added to the game, write to us so we can consider adding them in a future version of the game.

Annex C

Beta testing survey – Adaptive Game Design sections

Beta Test

This questionnaire takes approximately 20 minutes (with the visualization of videos included) and involves a 15 minutes' interaction with the game. It consists on 7 parts: A- Product Awareness, B- Character Creator, C- Overworld scenario, D- Combat system, E- Social Overlay - share menu, F- Social overlay - Party system and friends skills and G- Suggestions and Feedback Ideas.

To carry out this survey you need to:

- 1- Download our game: <http://tdmei.azurewebsites.net/Content/Game/APO.zip> (This zip contains a .exe with the game and a folder with game data. We recommend you to run game with resolution 1024x600.)
- 2- Have an account on Twitter or Facebook and optionally on Youtube (Google).
- 3- When requested in the tests (inside the game), temporarily accept our applications permissions in social networks.
- 4- When requested in the tests (inside the game), temporarily share some game contents in social networks.
- 5- Follow <https://twitter.com/Joana129> or <https://twitter.com/APOClockworkInc> on twitter OR add <https://www.facebook.com/loana.s.osorio> as facebook friend.
- 6- (Optional) If you have doubts about new features, bugs or limitations of social overlay, please download the following notes: http://tdmei.azurewebsites.net/Content/Docs/S.O.S_SocialOverlay.txt

It is worth noting that the focus of the game design aspect of the project (chapters, B,C and D) has shifted from a real game oriented experience to more of a study of an adaptive game design. With this, keep in mind that some aspects (such as the user interface) are less emphasized.

Thank you for your interest in being tester of our game and for filling out the questionnaire. Your answers will be confidential and very important for an appropriate framework of our master thesis.

*Required

A. Product Awareness

This section focus on how interesting and well built the ideas and fundamentals of this project were made.

1. Was the document clear enough for you to understand everything about the game? *

Rate how easy it was to understand what the game is about based on the document given.

1 2 3 4 5

Didn't understand Understood everything

2. How likely are you to recommend this game to your friends or colleagues? *

Consider that the game was on a retail stage and free to play.

1 2 3 4 5

Wouldn't recommend Highly recommend

3. Do you think the game is distinguishable and different enough from all the other games in the market? *

Given the current market of video games would you consider this game to be unique enough to be different, or too similar to all other games.

1 2 3 4 5

Unoriginal, not unique Very unique

Beta Test

*Required

B. Character Creator

In this game, actions are all dictated through a group of determined personality traits. These traits are what define the posture and attitude of your character toward the game. The character creator serves for you, the player, to set a base of aspects for your character emphasizing a unique personality system.

With this in mind we would ask you to think of a character's personality you would like to build and then answer the following questions.

Please watch this 30 second trailer about the character creator before testing the application.

Additional notes:

- 1- Click new game;
- 2- Type in the name Jack (this name is mandatory; any other cause a known error);
- 3- Read all the information in each panel;
- 4- Choose alignment, background, read carefully the personality traits system description;
- 5- Click done button.



1. Did you enjoy the variety of alignments in this group? *

Were you able to choose the proper alignment for the character you pictured?

1 2 3 4 5

No, not at all Yes, liked a lot

2. Does the different types of backgrounds give enough story to your character? *

Were you able to choose the right background for the character you imagined?

1 2 3 4 5

No, they don't Yes, they do

3.1 Was the personality traits explanation easy to understand and simple? *

Were you able to understand the implications and impacts of the different personality traits in the game?

1 2 3 4 5

No, they weren't Yes, they were

3.2 Would you like to see this system (or similar) implemented in other more character focused RPG games? *

1 2 3 4 5

No, not at all Yes, a lot

4. How was your experience with the character creator? *

What aspects would you change, remove or add to the creator?

Beta Test

*Required

C. Overworld scenario

Welcome to the overworld of the game. You are now in control of your character and you seem trapped in a forest of sorts. Please do keep in mind that the focus will be in how your character's personality traits affect the game.

If necessary, try talking to every character multiple times (specially after taking a decision).

We would also ask you to repeat this scene multiple times with different characters of varying different traits, and try different answers, so you can feel the difference with each outcome.

Please watch this 30 second trailer about the overworld level before testing the application.

Additional notes:

- 1- Experiment with the different buttons available (I opens inventory, C opens the character panel, J opens the journal);
- 2- Attempt to interact with everything (including Ogre) before talking to the white knight in front of you;
- 2- Talk to the white knight;
- 3- After accepting the quest and taking a decision check your Journal panel and your Character panel;
- 4- Interact with everything again (leave the Ogre for the last interaction and make sure to equip any items given to you before talking to it).

Note: If you would like to leave the game, press the Options button in the bottom panel or press the Escape key then Quit.



1. Were the controls and interface easy to use? *

1 2 3 4 5

Very hard Very easy

2. Did you understand why some characters interacted with your character but others didn't? *

1 2 3 4 5

Didn't understand Understood

3. Did you feel like your decisions impacted the world and the characters around you? *

If it felt like the world reacted and changed to your approach to a problem.

1 2 3 4 5

Decisions didn't matter Decisions matter

4. Do you think the different outcomes were adequate to your character's personality trait? *

If the world adapted and changed to your decisions according to how your character is.

1 2 3 4 5

Not adequate Very adequate

5. How was your experience with this small world?

Did you feel like the world changed and adapted to your decisions inside the game? What would you change, remove or add to this level?

Beta Test

*Required

D. Combat system

The last piece of game design related experience in the combat system. Once more the focus here is how the enemy adapts to your decision making and if it feels like your decisions, while in combat, make for a different combat experience.

1. How entertaining was your combat experience? *

If it was interesting to fight this particular opponent.

1 2 3 4 5

Not entertaining Very entertaining

2. How adaptive did your opponent felt in this combat experience? *

1 2 3 4 5

Not very adaptive Very adaptive

3. Did you felt that your decisions in the overworld level affected your combat experience? *

1 2 3 4 5

Didn't affect Affected very much

4. What would you change in this combat system to make it more dependent on your character's behavior?