

Project Proposal

Artificial Intelligence in video games

Supervisor: Dr. Keith Leonard Mannock

**Birkbeck University of London
Computer Science**

**Sylwester Stremlau
Sstrem01**

Disclaimer

I have read and understood the sections of plagiarism in the School Handbook and confirm that the work is my own, with the work of others clearly acknowledged. I give my permission to submit my submission to the plagiarism testing database that the College is using and test it using plagiarism detection software, search engines or meta-searching software.

SYLWESTER STREMLAU

Abstract

Artificial Intelligence is one of the leading topics in the Computer Science industry. It is used to assist with research, extract knowledge out of databases, automate decision making, used in video games and many more. Another big aspect in computer industry is the video game industry, bigger than music and even film industry combined. This project will focus on creation of a basic 2D video game in a Unity Game Engine and making a sophisticated AI that will give players an illusion that they are playing with a real human being. AI will be rule based and use techniques like Hierarchal Pathfinding with smoothing algorithms to give the best experience. It will also, if possible, explore smoothing algorithms and Machine Learning to improve it.

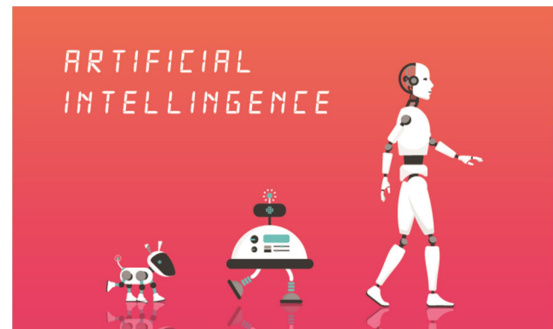
Contents

Chapter 1	4
Introduction	4
Why is it worth tackling?.....	5
Approach	5
A road map of the proposal document.....	7
Chapter 2	8
Chapter 3	10
Analysis, Requirements and Design	10
The Game	10
The view	11
Controls	11
Objectives	11
Language selection, libraries, frameworks, and why.....	13
Chapter 4	14
Chapter 5	15
Chapter 6	16
Glossary.....	16
Any special typography or terminology	16
References	17

Chapter 1

Introduction

This project will focus on creating a basic video game and designing Artificial Intelligence that will behave and challenge players similar to a real human being. The game will be kept very basic; inspired by games like Rocket League, Grand Theft Auto 2 and FIFA it will be a 2D top-down shooter/football game. There will be two teams on a football pitch, one ball and two goals just like in a classic football. Each team will have to work together to shoot the ball by kicking it or by using various weapons like melee, guns etc. to the opponents goal. Each player will have their own health points and the game will be timed, the team with the highest score wins. This type of game has been chosen because it is very basic, should be easy to implement and there is a lot of room to design the AI. Another advantage of doing this kind of game is that the AI will be scalable up or down so if it turns out that the scope of the project was too big, it will be very easy to scale it down by changing the enemy team size.



Making a game can be an enormous project that can take years even for a group of experienced developers to finish. In an interview with The Guardian, Dave Hagewood, CEO of Video Game Company Psyonix said that the process of refinement of Rocket League took over seven years, with a team of 15 people. Therefore this project will not focus so much on refining and polishing the game mechanics and little details – although this will be an important aspect – it will instead focus on designing the game AI as well as possible. As of now, the scope is to make the AI be playable in 1vs1 and 2vs2 game modes. These two game modes should be enough to showcase the AI because their behaviour in these modes will differ greatly.

To keep the scope of the project small, the AI will be rule based because it is a fairly small game with limited rules and situations, therefore it will be possible to use this approach instead of machine learning. Although if possible, it should be interesting to explore machine learning approach and how it differs from the rule based. (Tricentis, 2019; Nagy, 2018)

For the same reason as mentioned before, the graphics of the game will be kept very simple. The ball will most likely be white, grass green and everything will be made out of geometric shapes to not waste too much time making the design – in a game development graphic design is usually done by a separate team.

Aims

- To create an enjoyable game with enemy Artificial Intelligence that is complex enough to be able to challenge a player and make each game exciting and distinct.

Objectives

- To create a working game prototype that is complex enough to support a sophisticated enemy AI.
- To create enjoyable and fun game.
- To create a smooth and intuitive gameplay loop that can be enjoyed for a long period of time.
- To create AI with possibly various difficulties and various team sizes.
- To research ways to make AI more sophisticated and unpredictable.
- To, if possible, use Machine Learning to improve the AI to be more human-like.

Why is it worth tackling?

As a programmer it is important to realize that programming language is just a tool. Therefore it is important to choose the best language for a job and to be able to adapt to new environments all the time. As I only have experience in Java and Python, doing this project will give me an opportunity to learn a completely new environment, which is Unity engine, and a completely new language which is C#. This will allow me to grow as a programmer, learn new techniques and build my portfolio. Unity engine is the industry standard for gaming, John Riccitiello said "It's pretty much half of all games period", about the amount of games made in Unity (Dillet, 2019). And being able to work with the industry standard tools will inspire, challenge and motivate me and on top of that, it will show my potential employers that I am eager to adapt and learn.

Artificial Intelligence is also a very big and important aspect in, not only just gaming, but computer industry overall. Therefore making this project would allow me to dig deeper into the AI, understand its workings and develop a better appreciation for video games. I am also planning to work with Artificial Intelligence in the future either developing it for games or for research. This project would give me the experience and opportunity to learn about it in more detail and maybe even help me with my first job.

Approach

Project approach

At the start of the project, waterfall methodology will be used because it uses a linear sequential approach and planning, designing and analysing will be very important at the start for the project to succeed. However, the outcome of these phases will be used just as a guideline and once the development will begin, agile methodology will be used. Agile will be very useful because requirements and the scope of the project may need to slightly change as the project progresses and it will also help to avoid making it too small or too ambitious that will be impossible to finish. (Edeki, 2015)

Game development approach

In an article for Gamasutra, G.Lopes and R. Kuhnen propose a layered view of the game design which they explain that “each layer corresponds to a generalization or abstraction of the layers below it”. What they propose is the whole process should be broken down into the following layers: Concept, Context, Core (Features and Content), Mechanics and Verbs. Each step can be done sequentially from top-down or bottom up depending on the requirements. These two approaches are very similar to the Top-down/stub and Bottom-up software development techniques. In top-down for example, high level methods and the logic are first designed, dummy methods called stubs are inserted in place of actual methods, the programs is then tested and the whole process is repeated on the lower level.

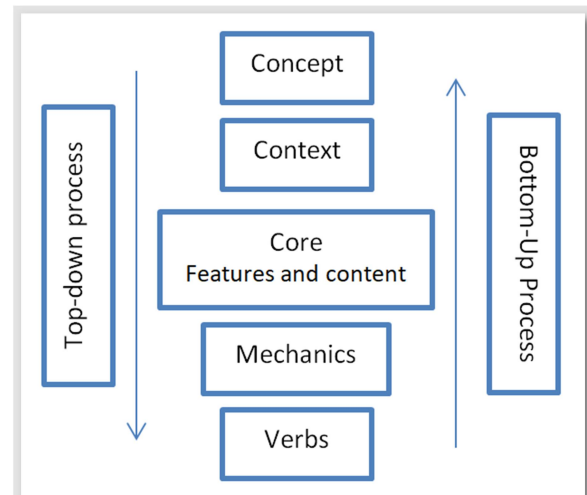


Figure 1. Game development process proposed by G.Lopes and R.Kuhnen

For this project a top-down approach proposed by Lopes and Kuhnen mixed with agile and waterfall methodologies will be used because as E. Bethke said in his book “Game Development and Production”, at its core, game development is just a programming project development with few extra steps.

As shown in figure 2, most of the steps will overlap and will be refined as the project progresses just like in agile methodology. To avoid wasting time, some of these steps like concept and context may get less attention than the other steps like core or mechanics because this project focuses on making AI first and not on making a commercially successful game but a playable one. (Bethke, 2003; Lopes and Kuhnen, 2019)

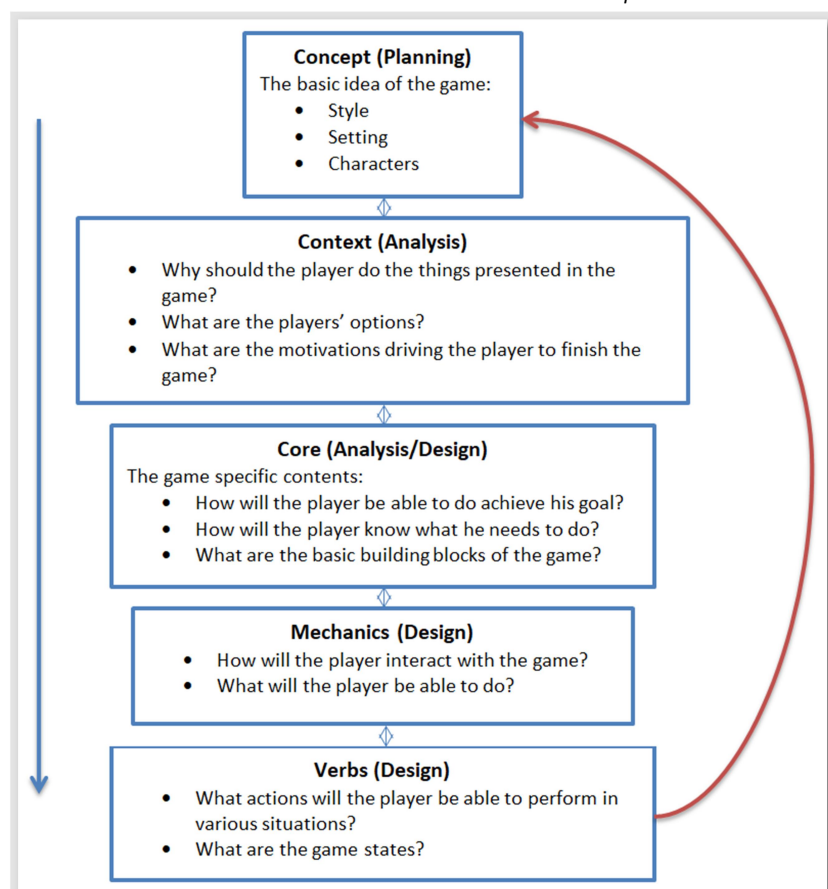


Figure 2. Proposed game development approach for this project.

A road map

In Chapter 2 I will go over the background of the game industry, where it is right now and where it is heading. I will also go through one of the best AI developed for video games like F.E.A.R. and The Sims and will describe what techniques the developers used to make it.

In Chapter 3 I will analyse how the game will need to be made, all its aspects and how I will use the knowledge of other games to design my own. I will then talk about the language, game engine and framework that I intend to use to create this game.

In Chapter 4 I will go through the techniques I will use to test the game and how I will determine whether the project was successful and if it meets the aims and objectives.

In Chapter 5 I will present a Gantt chart which says how I plan to use the time most effectively to make sure the project is finished on time and to a highest standard.

The last chapter includes references and appendices.

Chapter 2

With the emergence of the new technology which allowed for games to be more immersive and complex, video game industry has blew up so much, it surpassed movie and music industry. According to Global Games Market Revenues by Newzoo, video game industry has generated more money than music and movie industry combined for the past 11 years. In 2021 it is estimated that it will make \$180,000m while movie industry will only make around \$51,000m and music industry will make \$22,000m.

One would think the games are getting more and more sophisticated with all the new technology and budget but it may not entirely true. While in some aspects like graphics, storytelling or audio, they do, in other aspects like Artificial Intelligence there may be stagnation. According to numerous articles, one of the best AI is in the game called F.E.A.R. which was released in 17th October 2005, almost fourteen years ago – around the time that video game industry started really blowing up. (Horti, 2019; Orkin, 2006)

The techniques and systems used in F.E.A.R., while not entirely new or original were implemented extremely well and modern FPS' video game AI is usually not much more complicated or sophisticated. There are various reasons for this problem; the obvious one is the lack of funds thus the focus shifts to different aspects of the game like graphics, audio or online gameplay. Another reason may be the focus on the online gameplay – why develop complex and expensive AI that behaves like humans when players can play against each other (real humans)?

Another excellent example of game AI is the EA's life simulation game The Sims, recognized as one of the most influential AI of all time (Bourse, 2012). In this game player controls a family (called Sims), designs their house, gives them orders and does other numerous tasks. Sims can make decisions and survive on their own, however their behaviour is deliberately limited so players have a reason to play and feel like their decisions matter, otherwise the game would just play itself.

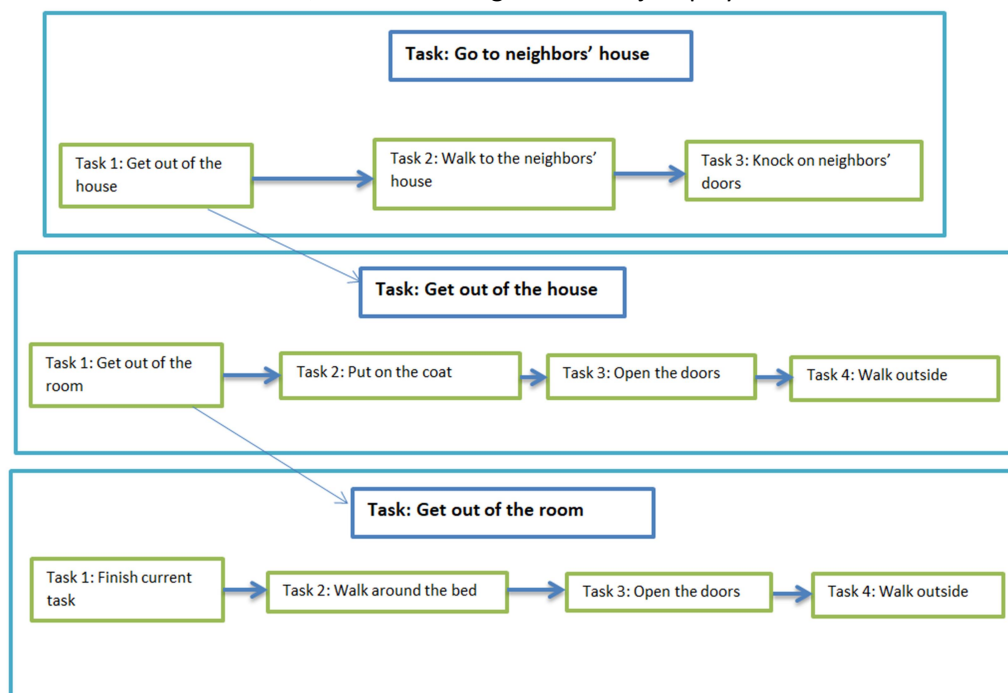


Figure 3. The Sims Hierarchical Pathfinding AI example.

The Sims use Hierarchical Pathfinding which means that for example, when the Sim decides to go to the neighbour, this step is broken down into smaller and smaller pieces, first he needs to go from his house to the neighbour's house, this means that first he needs to get out from his house, which means he needs to get out of the room that he's in and put on a coat, to put on a coat he needs to take the coat out and so on. This is mixed with smoothing algorithms to offer a very smooth path and make the Sims more realistic. HP is a great way to make AI feel realistic and intelligent, can be applied in a fairly easy way and is a great way to program AI to make seemingly complex decisions by breaking them down. (Botea, Muller and Schaffer, 2019)

Chapter 3

Analysis, Requirements and Design

The Game

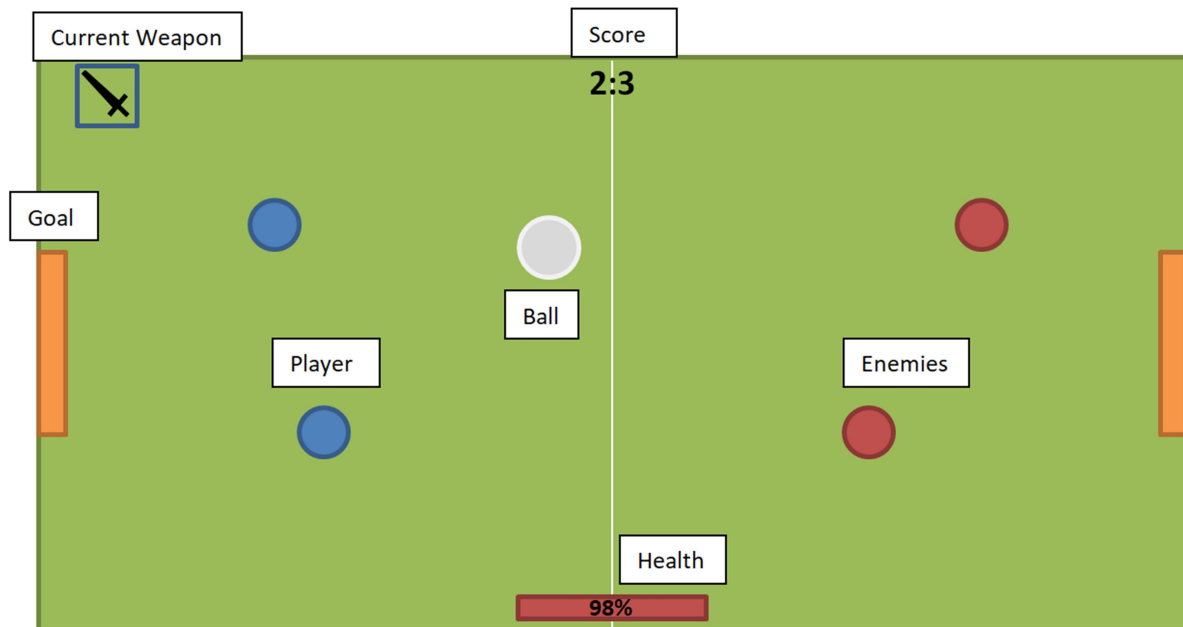


Figure 4. Basic design of the game project.

The game will be a topdown shooter/fighting/football game similar to Rocket League, there will be two goals, one ball and at least one player on each team. Each player will try to shoot the ball into the opponent's goal to earn a point. Team with the highest amount of points at the end of the game, wins. Since there are already successful football games like EA's FIFA and PES' Pro Evolution Soccer, this game will be slightly different and will not exactly follow the classic football rules. For example, there will be no offside; the ball will bounce off the sides and there will be no penalties etc.

To make the game more interesting and unique, players will have health points, stamina points and weapons like swords or guns with which they will be able to attack other players to gain additional points, score and to for example shoot them as they are about to kick the ball in to the goal. The default positions of the players should be researched during the testing to make sure that the flow of the game is not broken.

It is considered that the ball should also have health points, although much higher than the players to avoid breaking the flow of the game. This would create unique situations and add tension to the game. This would also add variation because players would have to ask themselves if they want to use swords that can send the ball really far but are not able to hit the ball from a far or if they want to use guns that are really good from the far but are not as good in a close range.

The view



Figure 5. GTA 1 video game 1997

The 'camera' will be pointed from above like in games like Grand Theft Auto 1, Hotline Miami, Overcooked, Undertale and many others. While this Point of View (POV) may seem out-dated, the gaming community and market for this type of games is still very big. For example, Undertale, released in September 2015, was one of the best-selling games on the digital game market Steam, with 530,343 copies sold by the end of 2015 (Galyonkin, 2019). Undertale uses top down view and retro graphics in its gameplay and it does not stop people from buying and enjoying it.

Controls

The game will be focused on gamepad controls so the players will be able to control the character in 360 degrees. Player will tilt the analog stick in the direction they want to move and the character will follow their lead. Alternative controls for keyboard and mouse controls would be to use arrow keys (or WASD) for walking and mouse for pointing in the direction to move. This should be researched, tested and compared with other games with the goal to make the controls intuitive, simple and easy to use.

Objectives

The objective of this game will be to shoot the ball into the opponents' goal while trying to stay alive and not letting the other team score the goal. Each player will have its own set of weapons and power-ups. These will have to be designed, tested and adjusted during the project but will be kept fairly simple to avoid overcomplicating the design of the AI, which this project will focus on.

Artificial Intelligence

Artificial Intelligence will be written in C# as this is the language that Unity uses so there is no need to use a different one. The AI will use decision trees based on rules to make their move and Hierarchal Pathfinding as explained in Chapter 2 to some extent; these will have to be kept simple as AI will have to make decisions very quickly and efficiently in real time. Although a lag or delay to the decisions may be added deliberately to create various difficulties to fit various types of players. The best reaction time for the AI should be researched and planned during the design process of the AI and during the Quality Assurance step.

The design and programming of the AI will have to be different for different team sizes, for example, AI would have to play in team in a 2vs2 game and on their own in a 1vs1 game. This will mean that their decision trees will have to be different because a team aspect will have to be added. Most of the time the AI's goal will be to score a goal but sometimes that may change, for example a new power up will be available, and so their new short term goal would be to get the power up but they may be on the other side of the pitch so first they would have to kick the ball to the other side of the pitch then fight the enemy and sprint to the power up, this would be when the Hierarchal Pathfinding would come in. Figure 6 shows a very basic example of how it would make its decisions in real time. This would most likely be much more complicated since it would have smoothing algorithms and other calculations on top. The best way that enemy AI should move should be researched and planned in the planning and design parts of the project.

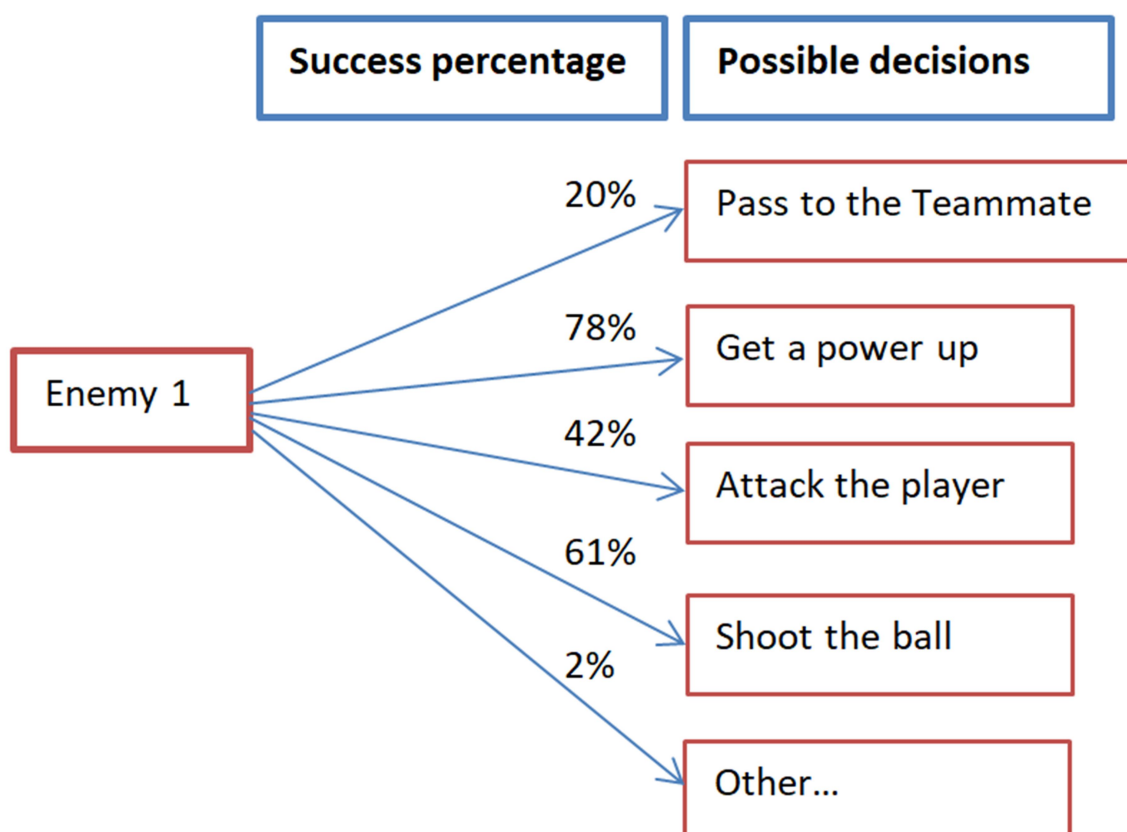


Figure 6. Example decision tree.

Language selection, libraries, frameworks, and why

There are various approaches available to create games and to write the Artificial Intelligence. Some of them are more popular and robust and others are smaller but still very popular. Three choices that fit the projects requirements have been taken into consideration these are Unity, GameMaker and PyGame.

For this project, Unity engine seems like the best fit because it is industry standard so it has numerous options like 2D game support, lots of tutorial material, is very popular and is widely supported. It also has a very easy interface which will allow finishing the main gameplay in a timely manner so more time can be spent perfecting the AI. Unity is also free which means that no additional costs will be needed.

Unity uses API written in C# which is a multi-paradigm programming language developed by Microsoft in 2000s, with current version being 7.3.

Python Pygame library has been chosen to use as a help during the design process of the project to help make better decisions and to let easily test various features. PyGame is a Python Programming Language library used to create and develop small applications and games in 2D. It is not usually used by developers since there are many other, better alternatives but it can still be very powerful.

Chapter 4

As in agile methodologies, testing is done after each iteration, which is after each function has been added. Since this project will be done using agile methodologies, each function will be tested for bugs and unforeseen behaviour after it is implemented. This is important because the project can get very big and this will ensure that it works as well as it should at all times. (Edeki, 2015)

The testing of the game will be broken down into two steps; first step will be after the basic game mechanics have been implemented and the game will be fully playable. This will involve playing the game against other people, letting other play the game against each other and doing what is called video game quality assurance (QA) by sending out free copies, letting friends play or playing with family. This will involve playing the game over and over again finding bugs, imbalances or other issues, then fixing or changing them and doing the whole process again. It will also involve playing the game casually to find how fun the game really is, since the whole point of making video games is to create something that can be enjoyed. The outcome of this should be detailed reports of each aspect of the game like mechanics or bugs, reviews or short notes written by anyone playing the game.

The next step will be similar to the first but this time the game AI will be tested. It will also involve playing the game over and over again to find situations where the game AI will not be able to make a decision, find any bugs or limitations and to find out how intelligent and realistic AI really is. The outcome of this process will be the same as in the first step.

When there are too many bugs and every new addition to the game will keep breaking it, it will indicate that the game has been a failure because when there are too many bugs in the game, the game can become unplayable, not fun and even frustrating for the players. The indication that the game has failed will also be when players will not see a reason to play the game, it will be unintuitive or the game will take more time to set up than to actually play. Another indication that the project did not meet the aims and objectives will be when the AI will be too easy or completely impossible to win with i.e. unfair.

A good indication that the game is successful will be when various types of players – those that for example play with mouse and keyboard or gamepad, or those that prefer to use melee instead of range weapons etc. – will be able to easily get into the game and use most of the possible functions intuitively. Another indication would be that the game does not have too many bugs or limitations, players do not have to spend too much time debugging the game and it is enjoyable at least to some extent. This will mean that most of the objectives of this project have been met. A biggest indication however that the game is successful is when the game AI will be challenging, balanced, fair (at least at lower difficulties), unpredictable, immersive, and behave realistically or at least give an impression that the player is playing with a real human and not a computer controlled software. This will mean that the project has met its main aim and all the objectives.

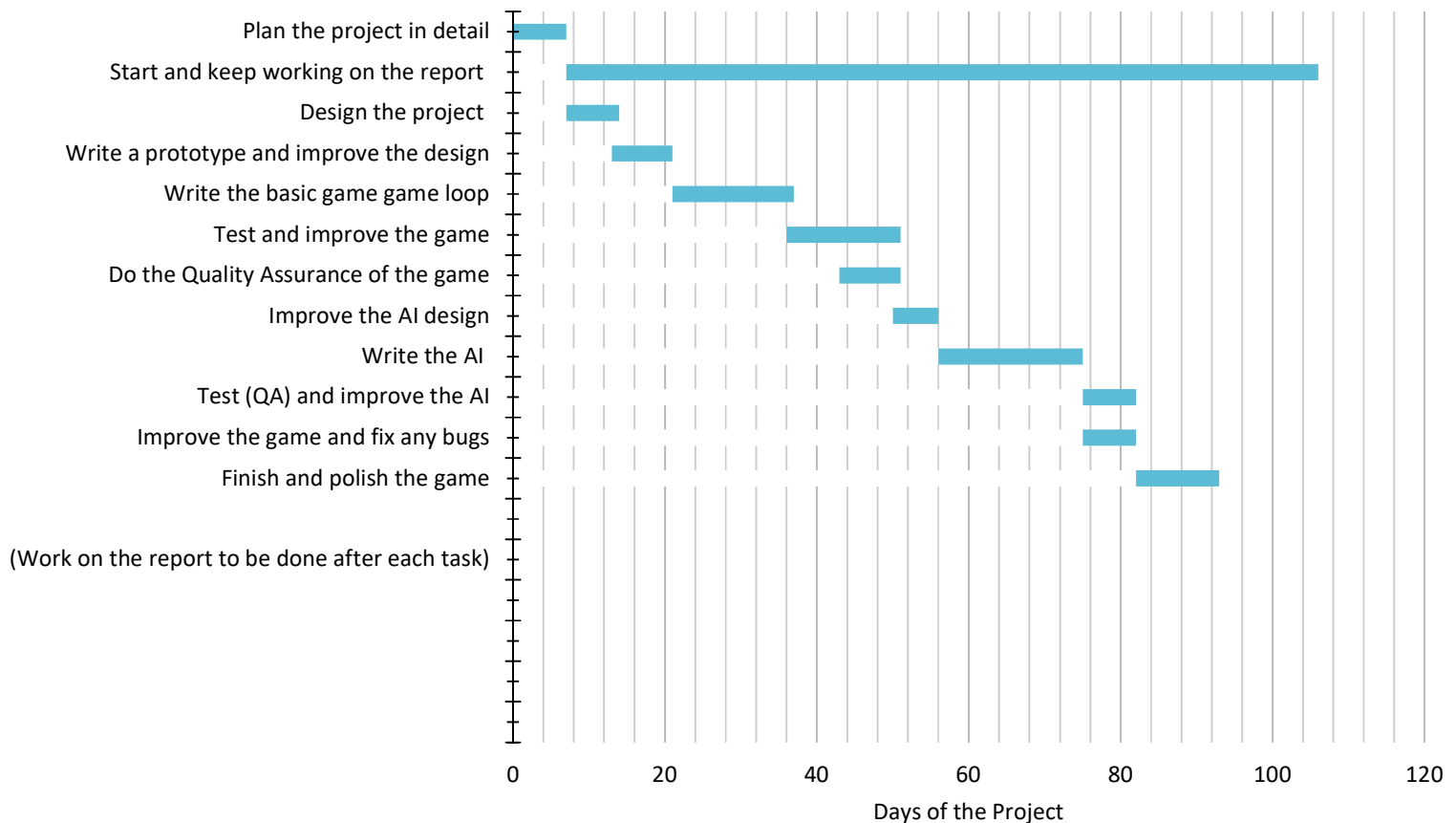
Chapter 5

GANTT CHART

* = an automatically calculated cell

	TASK NAME	START DATE	END DATE	START ON DAY*	DURATION* (WORK DAYS)
First Sample Project					
	Plan the project in detail	6/1	6/7	0	7
	Start and keep working on the report	6/8	9/14	7	99
	Design the project	6/8	6/14	7	7
	Write a prototype and improve the design	6/14	6/21	13	8
	Write the basic game game loop	6/22	7/7	21	16
	Test and improve the game	7/7	7/21	36	15
	Do the Quality Assurance of the game	7/14	7/21	43	8
	Improve the AI design	7/21	7/26	50	6
	Write the AI	7/27	8/14	56	19
	Test (QA) and improve the AI	8/15	8/21	75	7
	Improve the game and fix any bugs	8/15	8/21	75	7
	Finish and polish the game	8/22	9/1	82	11

(Work on the report to be done after each task)



Chapter 6

Glossary

Artificial Intelligence

Artificial Intelligence is the computer simulation of human behaviour and intelligence. It includes various processes like learning, adapting and thinking to act like a real human. There are various types of AI, from weak (also known as narrow) designed to do a certain task, to more complex, like strong AI that acts and thinks more like a real human being. (Rouse, 2019)

Rule Based AI

Rule based AI is a way to program AI to make decisions based on rules created and modified by a human expert.

Machine Learning

Machine Learning is an implementation of Artificial Intelligence to learn and train itself using data. Machine Learning usually needs training data fed and supervised by an engineer, to look for patterns and similarities so it can make more complex and informed decisions on new data. (Nagy, 2018)

Video game engine

A video game engine is a software environment that provides functionality to allow creating and developing games more easily. It provides stuff like rendering engine, physics engine, sound, scripting, animation and other essential functions needed to make a game. (Ward, 2019)

Game Loop/Gameplay

Game loop is a very basic loop that controls the flow of the game. A typical game loop is: process inputs, update game and render objects on the screen. (Bethke, 2003)

Game physics

Game physics are a simulation of physics that follow more or less how objects behave in real life. Games do not have to follow real life physics but they usually have their own rules that are consistent throughout the game. (Bethke, 2003)

Any special typography or terminology

AI – Artificial Intelligence

ML – Machine Learning

GUI – Graphical User Interface

API – Application Programming Interface

SDLC – Software development Life Cycle

References

- Bethke, E. (2003). *Game development and production*. Plano, Tex.: Wordware Pub., pp.3 - 36.
- Botea, A., Muller, M. and Schaeffer, J. (2019). Near Optimal Hierarchical Path-Finding. *Department of Computing Science, University of Alberta*.
- Bourse, Y. (2012). Artificial Intelligence in The Sims series.
- Dillet, R. (2019). *Unity CEO says half of all games are built on Unity*. [online] TechCrunch. Available at: https://techcrunch.com/2018/09/05/unity-ceo-says-half-of-all-games-are-built-on-unity/?guccounter=1&guce_referrer_us=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_cs=Pg_iiAtjpWlznUbkoBpuPA [Accessed 4 Apr. 2019].
- Edeki, C. (2015). AGILE SOFTWARE DEVELOPMENT METHODOLOGY. *European Journal of Mathematics and Computer Science*, 2(1).
- Galyonkin, S. (2019). *Preliminary results for Steam sales in 2015*. [online] Steam Spy. Available at: <https://galyonk.in/preliminary-results-for-steam-sales-in-2015-43bb49a767bd> [Accessed 4 Apr. 2019].
- Horti, S. (2019). *Why F.E.A.R.'s AI is still the best in first-person shooters*. [online] Rock Paper Shotgun. Available at: <https://www.rockpapershotgun.com/2017/04/03/why-fears-ai-is-still-the-best-in-first-person-shooters/> [Accessed 4 Apr. 2019].
- Lopes, G. and Kuhnen, R. (2019). *Game Design Cognition: The Bottom-Up And Top-Down Approaches*. [online] Gamasutra.com. Available at: http://www.gamasutra.com/view/feature/130542/game_design_cognition_the_.php [Accessed 4 Apr. 2019].
- Nagy, Z. (2018). *Artificial Intelligence and Machine Learning Fundamentals*. Birmingham: Packt Publishing Ltd.
- Orkin, J. (2006). Three States and a Plan: The A.I. of F.E.A.R. *Game Developers Conference*.
- Rouse, M. (2019). *What is AI (artificial intelligence)? - Definition from WhatIs.com*. [online] SearchEnterpriseAI. Available at: <https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence> [Accessed 4 Apr. 2019].
- Tricentis. (2019). *AI Approaches Compared: Rule-Based Testing vs. Learning - Tricentis*. [online] Available at: <https://www.tricentis.com/artificial-intelligence-software-testing/ai-approaches-rule-based-testing-vs-learning/#> [Accessed 4 Apr. 2019].
- Ward, J. (2019). *What is a Game Engine?- GameCareerGuide.com*. [online] Gamecareerguide.com. Available at: http://www.gamecareerguide.com/features/529/what_is_a_game_.php [Accessed 4 Apr. 2019].
- Wijman, T. (2019). *Global Games Market Revenues 2018 | Per Region & Segment | Newzoo*. [online] Newzoo. Available at: <https://newzoo.com/insights/articles/global-games-market-reaches-137-9-billion-in-2018-mobile-games-take-half/> [Accessed 4 Apr. 2019].