

# Project Proposal

---

**Artificial Intelligence in video games**

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

## Acknowledgements

## Contents

Chapter 1 .....	6
Introduction .....	6
State the problem you are trying to solve .....	6
Why is it worth tackling?.....	6
Approach .....	6
Any knowledge you presume of the reader to understand the proposal .....	8
Any special typography or terminology .....	8
A road map of the proposal document .....	8
Chapter 2 .....	9
Chapter 3 .....	10
Analysis, Requirements and Design .....	10
The Game .....	10
The view .....	11
Controls .....	11
Objectives .....	11
Characters .....	11
The game loop .....	12
Language selection, libraries, frameworks, and why .....	12
Prototype.....	12
Chapter 4 .....	12
Chapter 5 .....	12
Chapter 6 .....	12

# Chapter 1

## Introduction

### State the problem you are trying to solve

This project will focus on creating a video game Artificial Intelligence that will behave and challenge players similarly to a real human being.

### Aims

- To create a 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 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?

*(References will be added at the 'reference' page at the end)*

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 lots of different environments. 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.

Artificial Intelligence is also a very big and important aspect in, not only just gaming, but computer industry as well. Therefore making this project would allow me to dig deeper into the AI, understand its workings and develop a better appreciation for video games.

*(Need to add more)*

### Approach

#### Waterfall vs Agile

Waterfall is a type of Software Development Life Cycle (SDLC) which uses linear sequential approach to creating projects. There are few phases; Requirements Analysis, System Design, Implementation,

Testing, Deployment and Maintenance. Each phase has to be finished one by one, before the next phase can be completed. It is a very basic and easy to understand approach which is still used by (NUMBER) of companies/projects. It has few disadvantages, for example, there can be a very long period of time between the Requirements Analysis and Implementation, during which the requirements of the project can change.

[https://www.tutorialspoint.com/sdlc/sdlc\\_waterfall\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm)

## Agile

Agile is a type of SDLC which uses iterative and incremental approach. There are similar phases to Waterfall methodology, Planning, Requirements Analysis, Design, Coding, Unit Testing and User Acceptance Testing. Each phase is done one by one, but the process is divided into time boxed tasks. For example each phase would be completed in one iteration, in a very short time – one iteration may take two to three months. At the end of each iteration there should be a new deliverable like new features, GUI etc. then the process starts again. This allows the projects to be able to adapt to new and changing requirements.

## WRITE WHICH ONE I CHOSE

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.

## Unity

The first choice that may be the most popular is the Unity engine, a cross platform game engines that is used in half of the world's games, developed by Unity Technologies. Unity uses scripting API in C# programming language, and gives users a way to make 2D or 3D games. It supports various APIs like Direct3D, OpenGL or WebGL and is supported on various platforms like Android, Windows, PlayStation and others.

<https://unity.com/>

Another choice is the GameMaker studio developed by YoYo games. It provides a video game engine supported on various platforms like Windows, MacOS and Ubuntu among others. It has a very basic Drag and Drop (DnD) scripting tool that can be used by novice creators but is complex enough to make complex and sophisticated games.

[https://en.wikipedia.org/wiki/GameMaker\\_Studio](https://en.wikipedia.org/wiki/GameMaker_Studio)

The third option considered is the open-source library written for the Python programming language called PyGame. It provides support mainly for Windows, Linux, MacOS but it does not provide support for consoles. It is not a very complicated or popular tool (anecdotally it is usually used as a learning tool) but can be very effective for very small application.

For this project, Unity engine seems like the best fit, it has 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. The first basic prototype has

been written in PyGame to test and work on game physics since Unity takes more time to learn and basic draft has to be made and planned before the work on the actual game begins.

### **Any knowledge you presume of the reader to understand the proposal**

#### **Artificial Intelligence**

<https://searchenterpriseai.techtarget.com/definition/AI-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.

#### **Machine Learning**

<https://www.expertsystem.com/machine-learning-definition/>

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. For example, engineer would feed AI with a number of pictures of dogs, it would look for patterns and train itself, so when the engineer would then 'show' it the picture, it would be able to predict if it's a dog or not.

#### **Video game engine**

[https://en.wikipedia.org/wiki/Game\\_engine](https://en.wikipedia.org/wiki/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.

#### **Game Loop/Gameplay**

<http://www.informit.com/articles/article.aspx?p=2167437&seqNum=2>

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.

#### **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.

### **Any special typography or terminology**

AI – Artificial Intelligence

ML – Machine Learning

GUI – Graphical User Interface

API – Application Programming Interface

SDLC – Software development Life Cycle

### **A road map of the proposal document**



## Chapter 2

### Background

#### Literature review?

With the emergence of the new technology which allowed for games to be more immersive and complex experience, video game industry has blew up so much, it surpassed movie and music industry. According to (<https://lpesports.com/e-sports-news/the-video-games-industry-is-bigger-than-hollywood>) 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 \$51,000 and music industry will make \$22,000.

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 some articles, one of the best AI is in the game called F.E.A.R. which was released in 17<sup>th</sup> October 2005, almost fourteen years ago – around the time video game industry started really blowing up.

<https://news.ycombinator.com/item?id=14028842>

<https://www.rockpapershotgun.com/2017/04/03/why-fears-ai-is-still-the-best-in-first-person-shooters/>

[http://alumni.media.mit.edu/~jorkin/gdc2006\\_orkin\\_jeff\\_fear.pdf](http://alumni.media.mit.edu/~jorkin/gdc2006_orkin_jeff_fear.pdf)

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)?

Talk a little bit more about AI in games like The Sims or Rocket League and how they affect the industry and gameplay

## Chapter 3

### Analysis, Requirements and Design

What is the project?

What it will use?

How will it work?

#### The Game

#### **ADD DESIGN DIAGRAMS**

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. Once the

players' health goes down to zero, he would be reset to either the middle of the playing field or in some other position. The position that the player should reset to 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 to the game 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

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([http://www.gamasutra.com/view/news/262548/GTA\\_5\\_leads\\_Steam\\_Spys\\_list\\_of\\_bestselling\\_2015\\_Steam\\_games.php](http://www.gamasutra.com/view/news/262548/GTA_5_leads_Steam_Spys_list_of_bestselling_2015_Steam_games.php)

). Undertale uses top down view and retro graphics in its gameplay and it does not stop people from buying and enjoying it.

In this game players will most likely not be able to move the camera because the game window should be big enough that they can see what is going on at all times. This will also help to free players' fingers so they press the action buttons faster.

### 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 to find the best way that players should move because game mechanics are the fundamentals of the game and are what makes the game fun.

<https://www.pluralsight.com/blog/film-games/character-controls-camera-3cs-game-development>

[https://en.wikipedia.org/wiki/Game\\_mechanics](https://en.wikipedia.org/wiki/Game_mechanics)

### 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 like fast running, guns, swords and other. 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.

### Characters

## The game loop

## Language selection, libraries, frameworks, and why

Used Unity engine with C# scripting API

Unity engine will be used as the main framework for the GUI which uses scripting API for the C# programming language. Unity has been chosen because it supports 2D and 3D games which is very helpful because it will be important to – while designing the project – choose which computer graphics is the best fit. There is also a lot of tutorial material and a big community, which is very important because making games can become very complicated very fast.

C# is a multi-paradigm developed by Microsoft in 2000s, current version is 7.3.

Testing/prototype

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.

What will AI use?

Etc. etc.

Python for the prototype and testing

PyGame

## Prototype

The prototype has been written in PyGame library

## Chapter 4

Experimentation and evaluation

How I will show that the project meets the original aims and objectives

## Chapter 5

Timescale

Grantt chart

## Chapter 6

References etc.