

Computer Games Development CW208

GDD

Year IV

Leo Chen

C00250594

26/04/2023

# 

# 

**DECLARATION**

﻿

**Work submitted for assessment which does not include this declaration will not be assessed.**

* I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.
* I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.
* I have provided a complete bibliography of all works and sources used in the preparation of this submission.
* I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offence.

Student Name: (Printed) Leo Chen

Student Number(s): C00250593

Signature(s): Leo Chen

Date: 26/04/2023

--------------------------------------------------------------------------------------------------------

**Please note:**

1. \*Individual declaration is required by each student for joint projects.
2. Where projects are submitted electronically, students are required to submit their student number.
3. The Institute regulations on plagiarism are set out in Section 10 of Examination and Assessment Regulations published each year in the Student Handbook.

[**Acknowledgements** 3](#_Toc133417261)

[**Game Overview** 4](#_Toc133417262)

[**Feature Set** 4](#_Toc133417263)

[**General Features** 4](#_Toc133417264)

[**Gameplay** 4](#_Toc133417265)

[**The Game World** 4](#_Toc133417266)

[Main Gameplay Loop 5](#_Toc133417267)

[Neural Networks and Unity’s ML-Agents 6](#_Toc133417268)

[Draft Neural Network Architecture 7](#_Toc133417269)

[Units and their behaviours 8](#_Toc133417270)

[Control Points and Resource Points 9](#_Toc133417271)

[Inspiration 11](#_Toc133417272)

[Additional Goals 13](#_Toc133417273)

[References 14](#_Toc133417274)

# 

# **Acknowledgements**

We would like to thank the following people who assisted in completing this project including;

Lei Shi, my supervisor for giving me advice and insights on the project as a whole and my colleagues for their ideas, feedback and input as well as believing in me and my project.

# **Game Overview**

A Machine Learning Real Time Strategy game where the Machine Learning Agent/Opponent learns from every match you play. Both players must capture control points and generate scores from them for 5 minutes while simultaneously managing unit production and coordination.

The main objective of the game is a demonstration of the use of neural networks and machine learning agents as a viable opponent for human players. Unlike traditional real time strategy games with AIs developed to an exact skill level, the machine learning agent would scale based on your performance of the game.

# **Feature Set**

## **General Features**

2D Graphics

Real Time Strategy and unit controls

Basic economy system

Strategic capture points

## **Gameplay**

The main gameplay revolves around commanding groups of units to capture and hold strategic objectives.

Additionally, players are able to purchase units that spawn from their respective spawn points.

## **The Game World**

The game world is of a simplistic design in which there are 3 control points and 4 resource points for the players to capture. The world is in an enclosed area for which the boundaries of the game are set in.

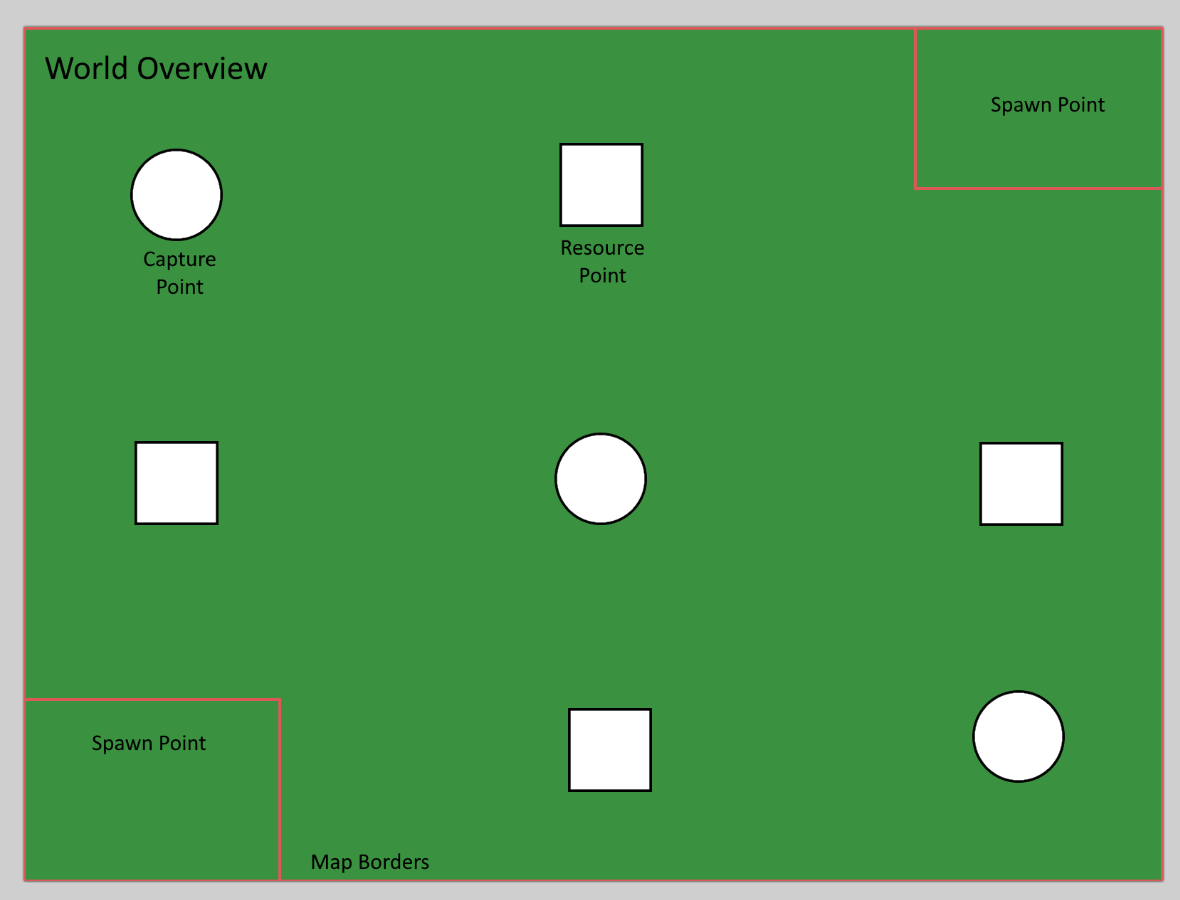


Fig. 1 Mock-up of the environment

## Main Gameplay Loop

The game revolves around holding capture points that generate a score for their respective team. Each game session lasts 3 to 5 minutes and whoever has the highest score once the timer ends would be considered the winner.

In these 3 to 5-minute games, the gameplay is short enough for the agent to have made enough decisions for use in evaluation.

Upon completion of a game, the agent is supposed to evaluate what strategies it employed had worked and which did not work in order to evolve for the next time it played.

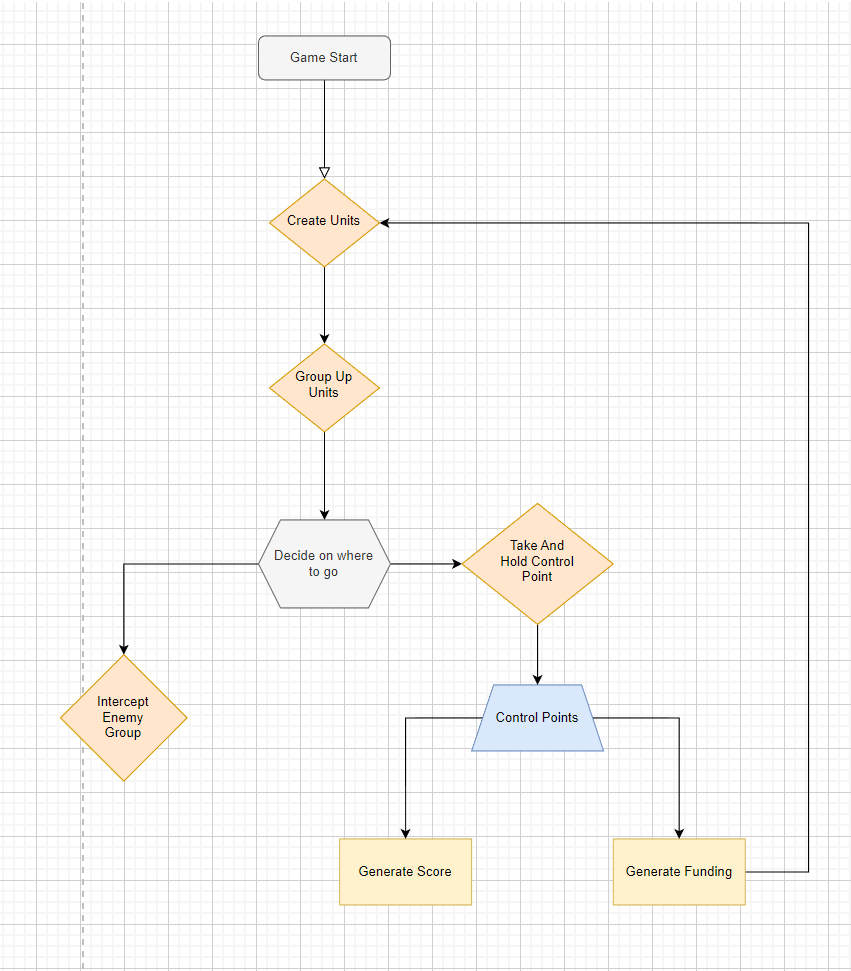


Fig. 2 Basic Game Loop

## Neural Networks and Unity’s ML-Agents

Either through the use of a neural network with backpropagation or with Unity’s ML-Agent’s reinforced learning model, the agent is meant to adapt through two methods of re-evaluation.

The first method is After Action Re-evaluation. The traditional method in which it collects its actions completed and feeds it back through the neural network with the use of back propagation. The other costlier method to implement would-be Real-Time re-evaluation. This proposed method would have the agent constantly revaluate a decision it has made as well as react directly to the environment it is in.

### Draft Neural Network Architecture

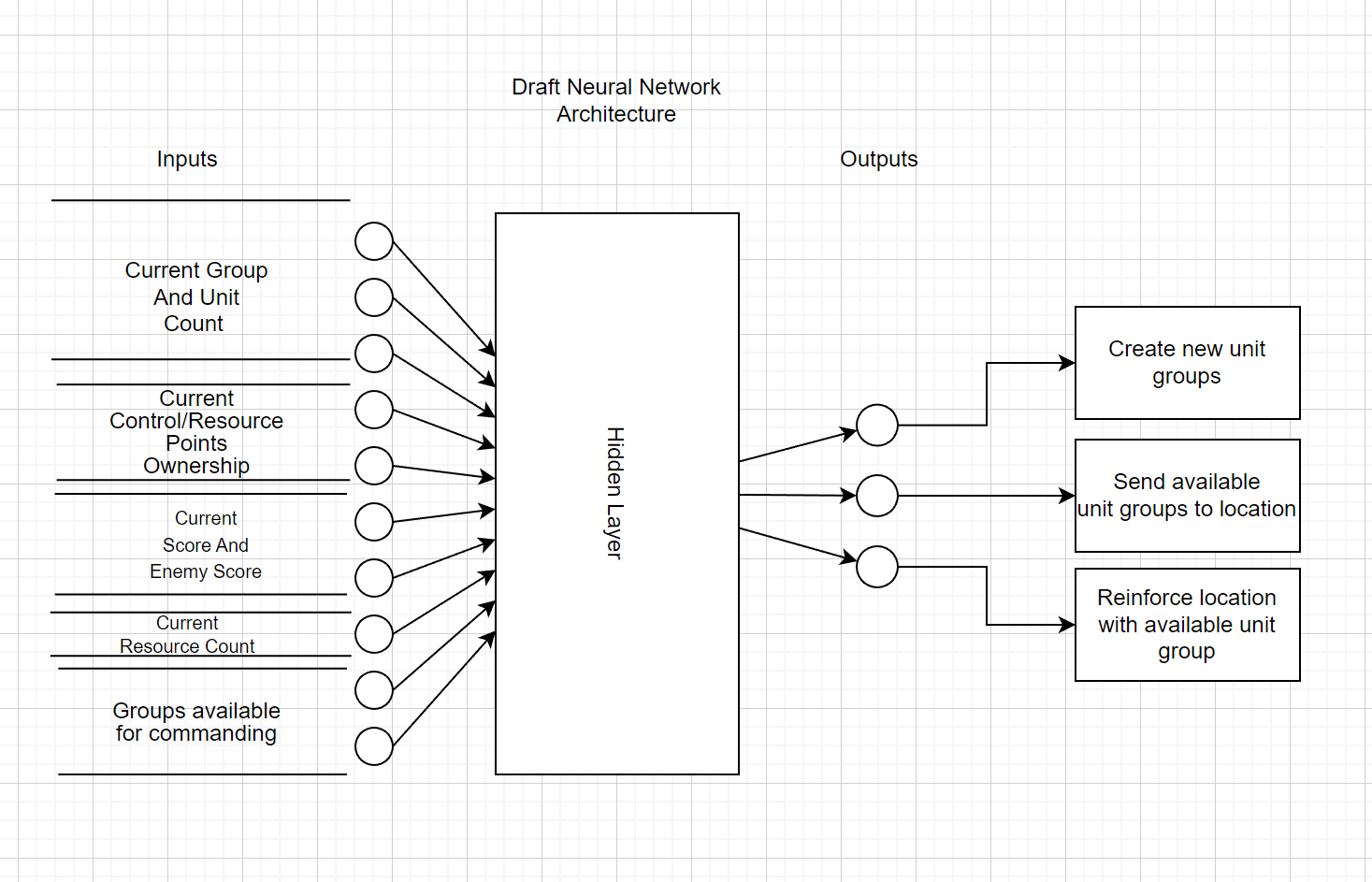


Fig. 3 Proposed Neural Network Architecture

Fig.3 Proposed neural network architecture

The evaluation for what the network would do next will depend on the outcome of the action it took. Detailed in the

## Units and their behaviours

The units are meant to be programmed with two simple behaviours. To scan for targets to shoot at or to move to a location. Their field of view requires it so that they have a clear line of sight towards their target in order to fire.

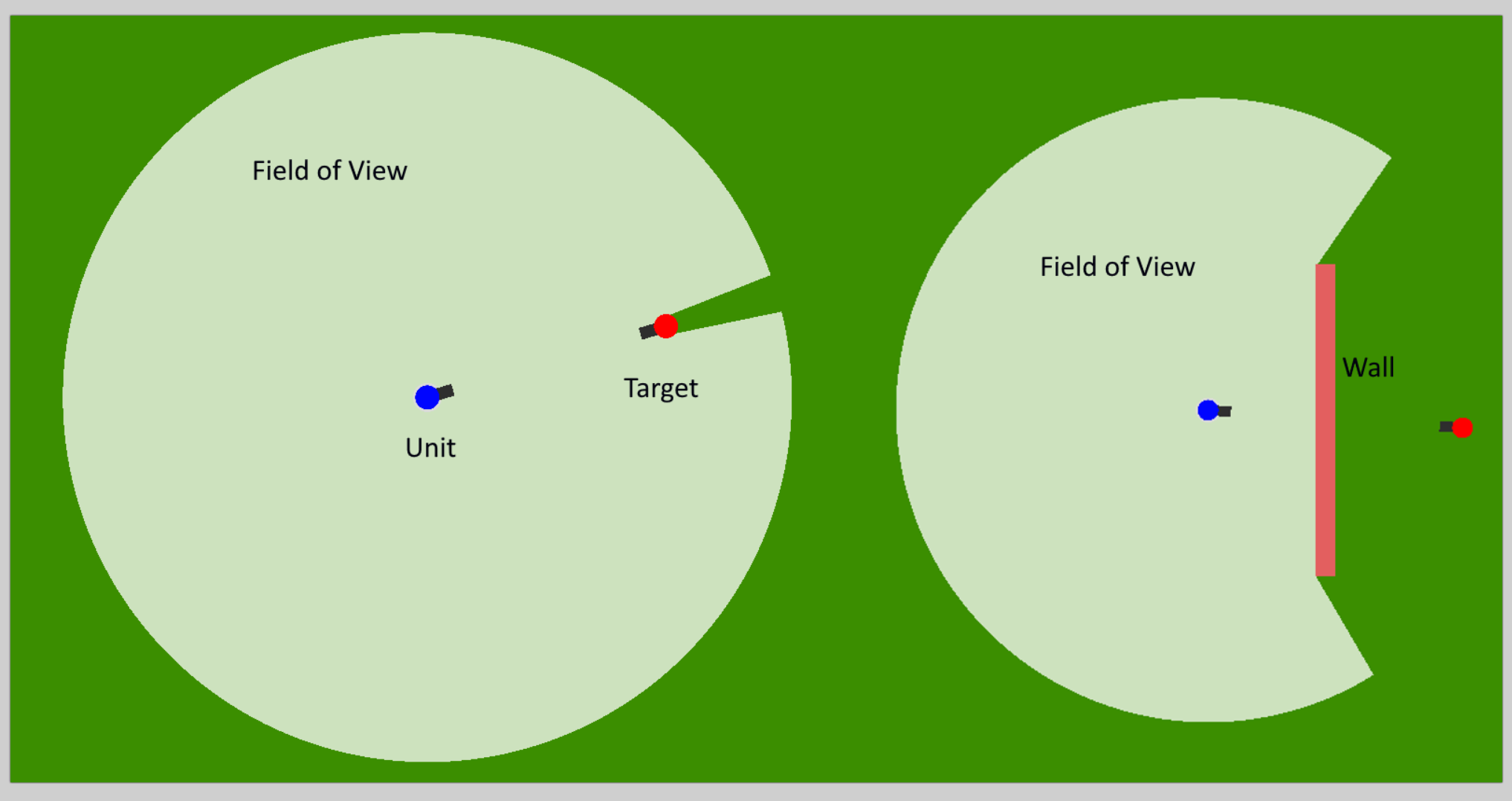


Fig. 4 Field of view example

The simplistic behaviour will be the basis of the units the players can command. Without needing to over complicate their design. Though lessons will be taken from Artificial Intelligence in Video Games (Ian Millington, 2019) in designing simple but effective unit AI.

Formations

Part of the unit design was to allow for the creation of formations. These allow for a tactical advantage when sending groups of units to battle where said formation is needed.

**Line Formation:**

A simple line formation that provides the best frontal fire power.

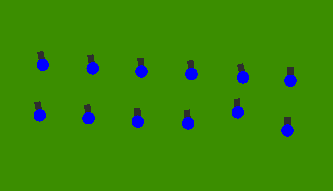


Fig. 5 Line Formation

**Box Formation:**

A hollow formation that provides the best defensive advantage.

**Chevron Formation:**

An arrow head like formation which provides a wide arc of fire.

## Control Points and Resource Points

The two main points of contention outlined in the introduction are control points and resource points.

Control Points

There are areas in which once captured will generate a score for their respective team.

Identified on the map as a circle, they will be the main areas of contention for them

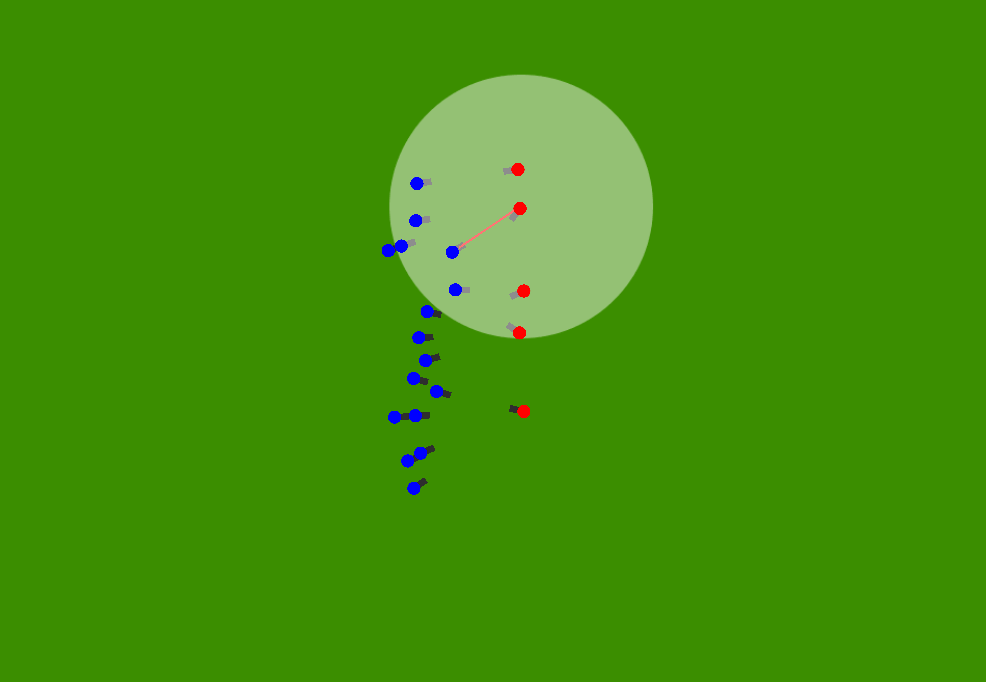
****

Fig. 6 Red and Blue contesting a control point

Resource Points

There are areas in which once captured will generate funding for their respective team. Secondary to the main control points but they allow for the generation of additional funding. This allows for more unit production. They are identified on the map as small squares.

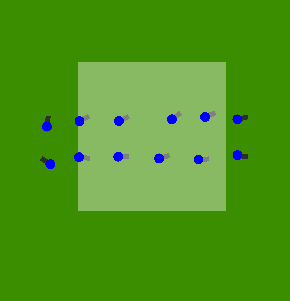


Fig. 7 Blue team holding a resource point

To capture any point requires at least one unit to enter their area and to hold it for a certain amount of time until it is fully captured to your team’s side. Once done it will passively generate the points to the team’s side.

# Inspiration

Command and Conquer (Electronic Arts, Westwood Studios, 1995)



Fig. 8 Command and Conquer Remastered

The father of all real time strategy games. The complex nature of the game with unit, economy and strategy management was a heavy inspiration for the project.

The gameplay, while more advanced than this project, takes some of its mechanical elements from the game.

Though the art style for my project is made up of very simplistic shapes, the underlying mechanisms that make up the game are somewhat present.

Company of Heroes (Relic Entertainment, 2006)



Fig. 9 Company of Heroes Screenshot

Evolving into a more historical setting for real time strategy games. Company of Heroes focuses more on the units and engagements rather than in a more traditional game of managing resource acquisition. Base building and resource management is more simplified here. Unit production is still tied to specific buildings that need to be built but resource management is now a passive accumulation of certain resources. Said resources are tied to control nodes around the map which must be captured. The more of a certain resource node you have, the more resources you get passively.

In a more traditional game, worker units would be needed to extract resources to bring back to their base or resource depot.

In Company of Heroes, resources are generated passively and are adjusted with the resource nodes they control rather than a resource gathering mechanic in other real time strategy games.

## 

## 

## Additional Goals

**Additional Units**

For additional content I would like to put in would be vehicles for the players to utilise. Two come to mind, tanks and jeeps.

Both of which would have their advantages and disadvantages to their use.

Tanks being the power but cumbersome and expensive unit.

And jeeps being fast but relatively weak units for hit and run action.

**Base Building**

Instead of having units spawned from one set location, I’d have it evolve into a more traditional RTS with buildings that produce those units. Worker units to fill the traditional construction of production buildings and with the potential of constructing defensive structures.

**Higher Map Roster**

Though the simplistic map for the game served its purpose as a simple test bed. Compared to real time strategy games like StarCraft (Activision-Blizzard Entertainment, 2010) or Command and Conquer (Electronic Arts, Westwood Studios, 1995), their maps are more expansive allowing for more flexible strategy involved in it.

# References

Activision-Blizzard Entertainment, 2010. *StartCraft 2.* s.l.:Activision-Blizzard Entertainment.

Electronic Arts, Westwood Studios, 1995. *Command and Conquer.* s.l.:Electronic Arts.

Ian Millington, J. F., 2019. *Artificial Intelligence for Games.* s.l.:s.n.

Relic Entertainment, 2006. *Company of Heroes.* s.l.:s.n.