University of Plymouth

School of Engineering, Computing and Mathematics



COMP3000

Final Stage Computing Project

2021/2022

*(Defend & Capture)*

James Yarnold

*(10621871)*

BSc (Hons) Computing & Games Development

# Acknowledgements

I would like to thank friends and family; especially my girlfriend Leah Humphries for always being there for me through each year of this course especially when times were low and giving me the strength to keep going and my parents at home who always provided me with support and guidance.

I would also like to thank Mr James Hayter and Mr Paul Watson for their efforts on teaching Unity and project semantics thought out the key stages of the course.

# Abstract

Table of Contents

[Acknowledgements 2](#_Toc101902352)

[Abstract 2](#_Toc101902353)

[Word Count 3](#_Toc101902354)

[Code 3](#_Toc101902355)

[Introduction 3](#_Toc101902356)

[Project Definition 3](#_Toc101902357)

[Purpose 3](#_Toc101902358)

[Objectives & Functional Requirements 4](#_Toc101902359)

[Minimum viable product 4](#_Toc101902360)

[Minimum awesome product 4](#_Toc101902361)

[Method of Approach 5](#_Toc101902362)

[Project Management 5](#_Toc101902363)

[Microsoft Teams 5](#_Toc101902364)

[Version Control 5](#_Toc101902365)

[Language and software used 5](#_Toc101902366)

[Legal, Social, Ethical and Professional 5](#_Toc101902367)

[Implementation 5](#_Toc101902368)

[Sprints 0 - 12 5](#_Toc101902369)

[End of Project Report 6](#_Toc101902370)

[Project Post-Mortem 6](#_Toc101902371)

[Conclusion 6](#_Toc101902372)

[Bibliography 6](#_Toc101902373)

[Appendices 6](#_Toc101902374)

# Word Count

# Code

(*GitHub link*)

# Introduction

This report details my experience in taking on a large solo project that attempts to showcase my skills and understanding of Unity’s libraries as well as incorporating many of the aspects that were taught to me over the duration of the course, specifically, I drew inspiration from COMP2007 where I built a 3D game and 3D scene for the first time as well as Comp3013 where I was project manager and spent more time with project development strategies and became more familiar with how to operate a large scale project from a planning level. I intended to utilise the experience id gained in these modules to plan and enact a project of appropriate size and scale that can be used as an important stand out portfolio piece that includes hall marks of industry standard criteria that employers would like to see. With significant focus on Unity and partner software with unity such as blender, I hoped to link this to the current COMP3014 module to achieve this portfolio piece as well as to feel fulfilled that my career at university lead to this, and I can be proud of it.

## Project Definition

Defend & Capture is a 3D single player, artificial intelligence driven, real time strategy game with a casual simulation twist on the combat. It is built on the Unity game engine, Specifically, built primarily using Unity’s navmesh AI system, raycast networks and lightweight custom built sorting algorithms. The game features some custom-built assets using Blender and free assets imported from the Unity asset store that creates an overarching ‘toy soldiers’ aesthetic to the game realising a casual theme that aims to welcomes a younger audience to the RTS; several RTS focused features and several important AI mechanics designed for engaging gameplay. The gameplay loop is designed around a rock paper scissors-based unit creation where unit types of counter other unit types, the player must manage their bases supplies and buildings while trying to stay on top of the unit counters the enemy AI is creating. The main objective for the player is to get to the AI base and destroy it to win.

## Purpose

on a personal level, the purpose of the game was to provide a creative platform for me to push my skills and knowledge of Unity in a way that’s challenging and introduces new concepts to my skill base to ultimately show to employers. On a gameplay level, Defend and capture hopes to stretch these skills in trying to produce a valuable, comprehensive AI using Unity’s libraries. Making the AI challenging and synergistic was an important aim to make the journey of playing an experience centred around determination so players can feel they did well even if they lost resulting in rewarding gameplay.

## Objectives & Functional Requirements

Upon project conception the functional requirements were established to give a good foundation to the backlog and actualise concepts into preliminary planning stages. The objectives that aligned with my existential goals were:

* Use a Kanban style project management board than parallels with key agile project management manifesto core values.
* Deliver an error/ bug free experience as to promote this in future works.
* Explore the ways of production of a custom-built AI in the unity game engine
* Use the knowledge of previous modules to amalgamate a gaming experience worthy of a portfolio showcase.

Core manifesto values reinforce an idea of not getting bogged down in extensive and exhaustive documentation but streamlining these into user stories as to begin the project sooner and without delay from the documents; they also stress the use of ‘customer collaboration’ which brings the potential customers into the development process to give developers useful insight as its being built and then accepting that changes might be helpful to the projects solution.(Wrike, n.d.).

My functional requirements were split into my minimum viable product (MVP) and my Maximum Awesome Product (MAP)

### Minimum viable product

* Player can build buildings and each of them effect the game in their specific way.
* Players can control troops using the single click system
* Player units and enemy units fight and can hurt each other
* Player can destroy the enemy base and win, and Players can have their base destroyed and lose (Conquest)
* Enemy Ai moves around the map randomly
* Game has no errors and functions on mid to low spec PCs

### Minimum awesome product

* Players can control the troops using single click, group select, select all of one type of units and select all units
* Player can upgrade existing units using the building upgrades system
* Units fight each other and rock paper scissors system is completely integrated
  + Flanking troops (if a troop is shot in the back) they take extra damage
* Players can decide which units fight which unit by clicking on the unit then the enemy unit
* Enemy AI uses tactics to fight for example, flanking and counter flanking, distractions, counter units, rushes/ turtling, scouting, directly fighting troops weak to the unit, or holding strategic areas of the map.
* More than one map
* More than one game type, conquest, and dominion (hold areas to gain points, first to x points wins)

Full context of these can be seen in the games design document at… [insert appendix number]

# Method of Approach

My main goal with my project management was to display a level of knowledge around agile project management that would be satisfactory or exceptional to the level of industry standards. To adhere to an Agile strategy, my project would be iteratively assessed and dynamically adapted to achieve functionality targets, with the addition of deliverable reviews on a biweekly or triweekly basis depending on task sizes, importance of depth and discovery of problematic errors. These ‘Sprints’ would dictate the flow of progress and task lists for that section. Adjusting the length of them dynamically was important given some tasks of the project were bigger and more important to complete than others. Following these with a review would portray what was achieved in the sprint and plans for the next sprint could be assessed with the information of the last sprint in plain text, helping the natural flow between the end of one sprint and the start of another. Sprints as they are on teams can be seen at [insert appendix number]

Research was an important step in creating the project Systems, they were written and broken down with the use of diagrams and charts (which are centralised into the GDD as per manifesto guidelines) which derived the projects backlog and allowed the project to be separated down in MVPs and MAPs. This was important as understanding the aims around creating the different parts of the game would set out the paths and avenues that would guide future sprints, therefore planning around revision and research times would be more effectively achieved with more specific conceptual points being brought to the forefront.

To achieve these methodologies, I needed a viable understanding of platforms in which to launch these project management strategies. There was a number of applications and packages I used for project management, but my aim was to keep the amount of these down to centralise my progression and development into apps that specialised in the project management and development field as well as having connections to my past experience at university, so time spent adjusting to new software and APIs wasn’t needed. I also needed a version control system with analogous criteria for project redundancy.

## Project Management

### Microsoft Teams

Teams was used for several different reasons

## Version Control

## Language and software used

# Legal, Social, Ethical and Professional

Credited assets

Appropriate licences

PEGI 16

# Implementation

## Sprints 0 - 12

# End of Project Report

# Project Post-Mortem

# Conclusion

# Bibliography

Wrike, n.d. *What Is the Agile Manifesto?.* [Online]   
Available at: https://www.wrike.com/agile-guide/agile-manifesto/#the-four-agile-manifesto-values  
[Accessed 23 April 2022].

# Appendices