



Gameplay Mechanics Development

Coursework Report

Gavin George

1501029

# Document Version History

|  |
| --- |
| **Current**  **Version 0.0.5 –**  **February 25rd, 2019**  Version 0.0.4 –  February 23rd, 2019  Version 0.0.3 –  February 19th, 2019  Version 0.0.2 –  February 12th, 2019  Version 0.0.1 –  January 30th, 2019 |

Table of Contents

[Document Version History 2](#_Toc2115367)

[Preface 5](#_Toc2115368)

[Requirements Specification 6](#_Toc2115369)

[1.1 Introduction 6](#_Toc2115370)

[1.1.1 Objective 6](#_Toc2115371)

[1.1.2 Intended Audience 6](#_Toc2115372)

[1.1.3 Project Scope 6](#_Toc2115373)

[1.2 Product Overview 6](#_Toc2115374)

[1.2.1 Overall Function 6](#_Toc2115375)

[1.2.2 User Guide 6](#_Toc2115376)

[1.2.3 Technical Requirements 6](#_Toc2115377)

[1.3 External Interface Requirements 6](#_Toc2115378)

[1.3.1 User Interface 6](#_Toc2115379)

[1.3.2 Hardware Interface 7](#_Toc2115380)

[1.3.3 Software Interface 7](#_Toc2115381)

[1.4 Project Features 7](#_Toc2115382)

[1.4.1 Demo Environment 7](#_Toc2115383)

[1.4.2 Camera Movement 7](#_Toc2115384)

[1.4.3 Mouse Selection 7](#_Toc2115385)

[1.4.4 Worker Units 8](#_Toc2115386)

[1.4.5 Resources 8](#_Toc2115387)

[1.4.6 Construction 8](#_Toc2115388)

[1.5 Non-functional Requirements 8](#_Toc2115389)

[1.5.1 Performance 8](#_Toc2115390)

[1.5.2 Design Attributes 8](#_Toc2115391)

[UML Diagrams 9](#_Toc2115392)

[Method 10](#_Toc2115393)

[3.1 Summary of Techniques 10](#_Toc2115394)

[3.2 Detailed Descriptions 10](#_Toc2115395)

[Development 11](#_Toc2115396)

[4.1 Development Process 11](#_Toc2115397)

[4.2 Concept Design 11](#_Toc2115398)

[4.3 Prototyping 11](#_Toc2115399)

[4.2 Documentation 11](#_Toc2115400)

[Conclusions 12](#_Toc2115401)

[5.1 Shortcomings 12](#_Toc2115402)

[5.2 Areas for Improvement 12](#_Toc2115403)

[5.3 Possible Solutions 12](#_Toc2115404)

[5.4 Extending the Application 12](#_Toc2115405)

[5.5 What I have Learned 12](#_Toc2115406)

[References 13](#_Toc2115407)

[6.1 Brief 13](#_Toc2115408)

[6.2 Techniques 13](#_Toc2115409)

[6.2.1 C++ 13](#_Toc2115410)

[6.2.2 Blueprints 13](#_Toc2115411)

[6.3 Research 13](#_Toc2115412)

[6.4 Resources 13](#_Toc2115413)

# Preface

This document is a comprehensive report intended to accompany the submitted coursework project, “CMP302 Gameplay Mechanics Development” by the author, Gavin George. Included within this report is a detailed specification of each facet of the chosen mechanic. A summary description of the mechanic is as follows: the mechanic features RTS style camera controls, selectable units, base-building and resource management. These features are typical of a real time strategy game, which was the intended objective; to implement RTS base building mechanics. The system allows the user to direct units to construct buildings and harvest resources, navigating and viewing the environment using a scrollable/panning/rotatable spring arm camera. The aim of this report is to deliver an in-depth explanation of the requirements and specifications of the system, to delineate the technical aspects and techniques used to achieve the implementation and finally to explain the development process of the project, providing UML diagrams as a visual aid and an online video demo to present the system.

# Requirements Specification

## 

## Introduction

### 1.1.1 Objective

The overall objective of this project was to provide a solution to a concept that the author synthesized in accordance with the coursework brief [6.1]. The concept of real time strategy base building was popularised through games such as Warcraft [6.3, 1] & Age of Empires [6.3, 2]. The RTS genre has bloomed over the last ten years with the same mechanics at its core and this project exists to fulfil and demonstrate said mechanics.

### 1.1.2 Intended Audience

Users of the application are expected to use this report as a guide to understand the applications functionality. Assessors will use the document to critically evaluate the project as part of the coursework submission. Furthermore, the project is intended as an academic resource for any who wish to use it for educational purposes.

### 1.1.3 Project Scope

The scope of this project limits it to an application with a single scene to act as a demo environment, with the intention of keeping the project compact and minimising un-necessary features. The scene will contain all the necessary elements to thoroughly present and demonstrate the full range of features available in the system. This approach is not that of a regular game project but more that of a specific system prototype that can be used for demonstration purposes.

The essential elements of the project are the five core features of the system, forming the base building mechanic. Stretch goals for this projected included: building and unit stats, simple AI other than pathfinding and a small variety of different building types, resource types and unit types. Graphics was the stretch goal with the least priority.

## Product Overview

### 1.2.1 Overall Function

### 1.2.2 User Guide

### 1.2.3 Technical Requirements

## External Interface Requirements

### 1.3.1 User Interface

* Project files are interfaced via the Unreal Engine
* The demo environment via the level editor & inspector
* Editable class attributes via hooks in the BP editor
* System interaction via gameplay using input peripherals

### 1.3.2 Hardware Interface

* The application is developed for the Windows PC platform
* Input requires keyboard and mouse

### 1.3.3 Software Interface

* The application was built in Unreal Engine 4 v\_4.19
* Blueprints were created in the Unreal Engine Blueprint Editor
* IDE used for programming was Microsoft Visual Studio 2017

## Project Features

### 1.4.1 Demo Environment

#### 1.4.1.1 Description

#### 1.4.1.2 Functional Requirements

**(Req.1) Level Design:**

The scene will consist of a landscape terrain. The terrain will not have obstacles and will be a flat plane. Resource patches will be placed around the map.

**(Req.2) User Interface:**

Include a menu system for selecting a building to construct, which is visible only when a worker is selected. Also, must include a display for showing selected units/buildings and their stats. Resource numbers will be visible on a bar at the top of the screen. The mouse position will have a cursor icon.

### 1.4.2 Camera Movement

#### 1.4.2.1 Description

#### 1.4.2.2 Functional Requirements

**(Req.1) WASD and Edge Scroll:**

The camera must be transformed left, right, up and down using the keyboard. The camera must also move if the mouse is moved past the related screen edge.

**(Req.2) Pan and Rotate:**

The camera must be tilted up and down and rotated using the mouse. The pan value will have a reset button.

**(Req.3) Zoom:**

The camera must be zoomed in and out using the mouse wheel. The zoom level will have a reset button.

**(Req.4) Editable Settings:**

The camera settings for move speed and maximum & minimum zoom must be alterable in the blueprint editor for the use of artists or designers.

### 1.4.3 Mouse Selection

#### 1.4.3.1 Description

#### 1.4.3.2 Functional Requirements

**(Req.1) Click Select:**

When the mouse cursor is hovering over a unit or building left click will select and highlight the unit or building. The cursor icon will change when hovering over a selectable object.

**(Req.2) Drag Select Box:**

Pressing and holding the select button will draw a box between the first click point and the mouse cursor current position. Releasing the select button will cause all units within the box to be selected. Buildings cannot be selected with the selection box.

### 1.4.4 Worker Units

#### 1.4.4.1 Description

#### 1.4.4.2 Functional Requirements

**(Req.1) Move to Position:**

Units will move to the position of the mouse when the user right clicks an area of empty ground on the terrain. Units must be selected to be moved. Units will move around structures. Units will space themselves apart to prevent clipping.

**(Req.2) Build Structure:**

Selected workers will automatically move to construct a placed building. Workers not building can be sent to construct a building by right clicking on the unfinished structure.

**(Req.3) Harvest Resource:**

Selected units will move to harvest a resource patch by right clicking on the resource. Units will bring gathered resources back to a collection building to deposit. When carrying a resource, the unit will have a visual effect, depending on the resource. If carrying a resource of ‘type 1’ and ordered to harvest a resource of ‘type 2’, the carried ‘type 1’ resource will be replaced with ‘type 2’. When ordered to harvest the worker will automatically continue harvesting and depositing from the same resource.

### 1.4.5 Resources

#### 1.4.5.1 Description

#### 1.4.5.2 Functional Requirements

**(Req.1) Resource Depletion:**

As the resources are harvested from the patch the total number of resources in the patch will reduce. If a patch has zero resources, the patch stops being available to harvest.

**(Req.2) Maximum Workers:**

Each patch of resources has a maximum number of concurrent workers. Assigned workers will wait until there is a free space for harvesting.

### 1.4.6 Construction

#### 1.4.6.1 Description

#### 1.4.6.2 Functional Requirements

**(Req.1) Placing a Building:**

Selecting a building from the menu spawns a ‘ghost’ that will follow the mouse cursor. Left clicking on an empty space will place the building. Buildings cannot be placed on top of other buildings. Units will move away from the building placement location.

**(Req.2) Build Cost:**

Buildings will take a set number of resources and time to construct. Resources will be removed when the building is placed. Unfinished buildings can be cancelled, returning the resources used.

**(Req.3) Multiple Builders:**

Assigning more workers to build will decrease the build time for a building under construction (increase the build speed).

## Non-functional Requirements

### 1.5.1 Performance

### 1.5.2 Design Attributes

# UML Diagrams

# Method

## 3.1 Summary of Techniques

## 3.2 Detailed Descriptions

# Development

## 4.1 Development Process

## 4.2 Concept Design

## 4.3 Prototyping

## 4.2 Documentation

# Conclusions

## 5.1 Shortcomings

## 5.2 Areas for Improvement

## 5.3 Possible Solutions

## 5.4 Extending the Application

## 5.5 What I have Learned

# References

## 6.1 Brief

*Bett, M. (2019). [online] Blackboard.abertay.ac.uk. Available at:* [*https://blackboard.abertay.ac.uk/webapps/blackboard/content/listContent.jsp?course\_id=\_8571\_1&content\_id=\_524873\_1&mode=reset*](https://blackboard.abertay.ac.uk/webapps/blackboard/content/listContent.jsp?course_id=_8571_1&content_id=_524873_1&mode=reset%20) *[Accessed 30 Jan. 2019].*

## 6.2 Techniques

### 6.2.1 C++

### 6.2.2 Blueprints

*YouTube. (2015). Unreal Engine 4: RTS Tutorial Series. [online] Available at:* [*https://www.youtube.com/watch?v=FZK5T-vAVFA&list=PLDnygpcOYwFW2XtNyiandrLDG\_\_OAZs7Q*](https://www.youtube.com/watch?v=FZK5T-vAVFA&list=PLDnygpcOYwFW2XtNyiandrLDG__OAZs7Q) *[Accessed 23 Feb. 2019].*

## 6.3 Research

[1] *En.wikipedia.org. (2019). Warcraft: Orcs & Humans (1994). [online] Available at:* [*https://en.wikipedia.org/wiki/Warcraft:\_Orcs\_%26\_Humans*](https://en.wikipedia.org/wiki/Warcraft:_Orcs_%26_Humans) *[Accessed 19 Feb. 2019].*

[2] *En.wikipedia.org. (2019). Age of Empires (video game) (1997). [online] Available at:* [*https://en.wikipedia.org/wiki/Age\_of\_Empires\_(video\_game)*](https://en.wikipedia.org/wiki/Age_of_Empires_(video_game)%20) *[Accessed 19 Feb. 2019].*

## 6.4 Resources

*Docs.unrealengine.com. (2004). Unreal Engine 4 Documentation. [online] Available at:* [*https://docs.unrealengine.com/en-us*](https://docs.unrealengine.com/en-us%20) *[Accessed 24 Feb. 2019]. -> Unreal Engine 4 -> Starter Content*