



Gameplay Mechanics Development

Coursework Report

Gavin George

1501029

# Document Version History

|  |
| --- |
| **Current**  **Version 1.0.0 –**  **April 29th, 2019**  V 0.1.5 –  April 10th, 2019  V 0.1.0 –  March 31st, 2019  V 0.0.9 –  March 24th, 2019  V 0.0.8 –  March 20th, 2019  V 0.0.7 –  March 11th, 2019  V 0.0.6 –  February 27th, 2019  V 0.0.5 –  February 25th, 2019  V 0.0.4 –  February 23rd, 2019  V 0.0.3 –  February 19th, 2019  V 0.0.2 –  February 12th, 2019  V 0.0.1 –  January 30th, 2019 |

Table of Contents

[Document Version History 2](#_Toc7465334)

[Preface 5](#_Toc7465335)

[System Requirements Specification 6](#_Toc7465336)

[1.1 Introduction 6](#_Toc7465337)

[1.1.1 Objective 6](#_Toc7465338)

[1.1.2 Intended Audience 6](#_Toc7465339)

[1.1.3 Project Scope 6](#_Toc7465340)

[1.1.4 Online Demo 6](#_Toc7465341)

[1.2 Product Overview 6](#_Toc7465342)

[1.2.1 Overall Function 6](#_Toc7465343)

[1.2.2 User Guide 7](#_Toc7465344)

[1.2.3 Operating Environment 7](#_Toc7465345)

[1.2.4 Assumptions & Dependencies 7](#_Toc7465346)

[1.3 External Interface Requirements 7](#_Toc7465347)

[1.3.1 User Interface 7](#_Toc7465348)

[1.3.2 Hardware Interface 7](#_Toc7465349)

[1.3.3 Software Interface 7](#_Toc7465350)

[1.4 System Features 7](#_Toc7465351)

[1.4.1 Demo Environment 7](#_Toc7465352)

[1.4.2 Camera Movement 8](#_Toc7465353)

[1.4.3 Mouse Selection 8](#_Toc7465354)

[1.4.4 Worker Units 9](#_Toc7465355)

[1.4.5 Resources 9](#_Toc7465356)

[1.4.6 Construction 9](#_Toc7465357)

[1.5 Other Non-functional Requirements 10](#_Toc7465358)

[1.5.1 Performance 10](#_Toc7465359)

[1.5.2 Design Attributes 10](#_Toc7465360)

[UML Diagrams 11](#_Toc7465361)

[2.1 Use-Case Diagram 11](#_Toc7465362)

[2.2 Class Diagram 11](#_Toc7465363)

[Method 12](#_Toc7465364)

[3.1 Selection Control 12](#_Toc7465365)

[3.1.1 Detailed Description 12](#_Toc7465366)

[3.1.2 Summary of Techniques 12](#_Toc7465367)

[3.2 Worker Control 12](#_Toc7465368)

[3.2.1 Detailed Description 12](#_Toc7465369)

[3.2.2 Summary of Techniques 12](#_Toc7465370)

[3.3 Tasks 12](#_Toc7465371)

[3.3.1 Detailed Description 12](#_Toc7465372)

[3.3.2 Summary of Techniques 12](#_Toc7465373)

[3.4 Buildings 12](#_Toc7465374)

[3.4.1 Detailed Description 12](#_Toc7465375)

[3.4.2 Summary of Techniques 12](#_Toc7465376)

[3.5 Resources 12](#_Toc7465377)

[3.5.1 Detailed Description 12](#_Toc7465378)

[3.5.2 Summary of Techniques 12](#_Toc7465379)

[Development 13](#_Toc7465380)

[4.1 Development Process 13](#_Toc7465381)

[4.2 Concept Design 13](#_Toc7465382)

[4.3 Prototyping 13](#_Toc7465383)

[4.2 Documentation 13](#_Toc7465384)

[Conclusions 14](#_Toc7465385)

[5.1 Shortcomings 14](#_Toc7465386)

[5.2 Known Issues / Areas for Improvement 14](#_Toc7465387)

[5.3 Possible Solutions 14](#_Toc7465388)

[5.4 Extending the Application 14](#_Toc7465389)

[5.5 What I have Learned 14](#_Toc7465390)

[References 15](#_Toc7465391)

[6.1 Brief 15](#_Toc7465392)

[6.2 Techniques 15](#_Toc7465393)

[6.2.1 C++ 15](#_Toc7465394)

[6.2.2 Blueprints 15](#_Toc7465395)

[6.3 Research 15](#_Toc7465396)

[6.4 Resources 15](#_Toc7465397)

[Appendices 16](#_Toc7465398)

[Appendix A: User Guide 16](#_Toc7465399)

[Appendix B: UML Diagrams 17](#_Toc7465400)

# 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.

# 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.

### 1.1.4 Online Demo

An online demonstration of this product was uploaded to YouTube, as per the coursework brief for this submission.

## Overall Description

### 1.2.1 Product Functions

The product must allow the user to perform a number of functions, these functions are listed thusly:

* The product must let the user control the camera dynamically
* The product must let the user select and interact with the units and buildings in the scene
* The product must let the user assign tasks to individual AI workers units
* The product must perform with a high level of robustness, reliability and efficiency

### 1.2.2 Product Perspective

The project is a new, self-contained product representing a gameplay component of an RTS style game. The system demonstrates the building and resource management mechanics of this genre of game.

### 1.2.3 User Guide

For information regarding the setup and gameplay control of the product consult the User Guide [See Appendix A] which is included within this documentation.

### 1.2.4 Operating Environment

The system shall operate through the Unreal Engine 4 editor on a Windows machine. The minimum hardware specifications for this are as follows, sourced from the Epic Wiki [6.3, 3].

* Desktop PC or Mac
* Windows 7 64-bit or Mac OS X 10.9.2 or later
* Quad-core Intel or AMD processor, 2.5 GHz or faster
* NVIDIA GeForce 470 GTX or AMD Radeon 6870 HD series card or higher
* 8 GB RAM

### 1.2.5 Assumptions & Dependencies

The system depends on the correct setup of the Unreal Project used by the assessor. The makeup of the demo level and project settings could affect the project if incorrect. The contents of the System Requirements Specification was produced upon the assumption that the project had been setup as instructed in the User Guide [See Appendix A].

### 1.2.6 Design & Implementation Constraints

Constraints regarding the design and implementation of the product were stipulated in the coursework brief [6.1].

## External Interface Requirements

### 1.3.1 User Interface

* Project files interfaced via the Unreal Engine
* Custom HUD class acts as interface between player and game via UI elements
* 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 shall be 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\_4.22
* Blueprints were created in the Unreal Engine Blueprint Editor
* C++ Classes were implemented using the UE4 C++ class constructer
* IDE used for programming was Microsoft Visual Studio 2017

## System Features

### 1.4.1 Demo Environment

#### 1.4.1.1 Description & Priority

#### 1.4.1.2 Stimulus/Response Sequences

#### 1.4.1.3 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 & Priority

#### 1.4.2.2 Stimulus/Response Sequences

#### 1.4.2.3 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 & Priority

#### 1.4.3.2 Stimulus/Response Sequences

#### 1.4.3.3 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

#### 1.4.3.1 Description & Priority

#### 1.4.3.2 Stimulus/Response Sequences

#### 1.4.3.3 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

#### 1.4.4.1 Description & Priority

#### 1.4.4.2 Stimulus/Response Sequences

#### 1.4.4.3 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

#### 1.4.5.1 Description & Priority

#### 1.4.5.2 Stimulus/Response Sequences

#### 1.4.5.3 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).

## Other Non-functional Requirements

### 1.5.1 Performance

### 1.5.2 Design Attributes

# UML Diagrams

## 2.1 Use-Case Diagram

The Use-Case Diagram [See Appendix B, Figure 1] for the system was used to visualise the system from the user’s point of view.

## 2.2 Class Diagram

The Class Diagram [See Appendix B, Figure 2] was created to model the static structure of the system and its classes, attributes operations and relationships.

# Method

## 3.1 Selection Control

### 3.1.1 Detailed Description

The primary method of control for the player was through mouse selection input. As specified in the system features functional requirements it was necessary to be able to drag a selection box

### 3.1.2 Summary of Techniques

## 3.2 Worker Control

### 3.2.1 Detailed Description

### 3.2.2 Summary of Techniques

## 3.3 Tasks

### 3.3.1 Detailed Description

### 3.3.2 Summary of Techniques

## 3.4 Buildings

### 3.4.1 Detailed Description

### 3.4.2 Summary of Techniques

## 3.5 Resources

### 3.5.1 Detailed Description

### 3.5.2 Summary of Techniques

# Development

## 4.1 Development Process

## 4.2 Concept Design

## 4.3 Prototyping

## 4.2 Documentation

# Conclusions

## 5.1 Shortcomings

## 5.2 Known Issues / 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++

*Wiki.unrealengine.com. (2015). HUD, Canvas, Code Sample of 800+ Lines, Create Buttons & Draw Materials - Epic Wiki. [online] Available at:* [*https://wiki.unrealengine.com/HUD,\_Canvas,\_Code\_Sample\_of\_800%2B\_Lines,\_Create\_Buttons\_%26\_Draw\_Materials*](https://wiki.unrealengine.com/HUD,_Canvas,_Code_Sample_of_800%2B_Lines,_Create_Buttons_%26_Draw_Materials%20) *[Accessed 20 Mar. 2019].*

*YouTube. (2017). [UE4 C++] RTS Select and Move Units. [online] Available at:* [*https://www.youtube.com/watch?v=TLp4XyX6y4o*](https://www.youtube.com/watch?v=TLp4XyX6y4o%20) *[Accessed 20 Mar. 2019].*

*Docs.unrealengine.com. (2004). Programming Guide. [online] Available at:*[*https://docs.unrealengine.com/en-us/Programming*](https://docs.unrealengine.com/en-us/Programming%20) *[Accessed 11 Mar. 2019].*

*Api.unrealengine.com. (2004). Unreal Engine Programming and Scripting Reference. [online] Available at:* [*http://api.unrealengine.com/INT/index.html*](http://api.unrealengine.com/INT/index.html%20) *[Accessed 11 Mar. 2019].*

### 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].*

[3] *Wiki.unrealengine.com. (2004). Recommended Hardware - Epic Wiki. [online] Available at:* [*https://wiki.unrealengine.com/Recommended\_Hardware*](https://wiki.unrealengine.com/Recommended_Hardware%20) *[Accessed 10 Apr. 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*

*Visual Studio. (2017). Visual Studio IDE, Code Editor, Azure DevOps, & App Centre - Visual Studio. [online] Available at:* [*https://visualstudio.microsoft.com/*](https://visualstudio.microsoft.com/%20) *[Accessed 24 Feb. 2019].*

*Cite This For Me. (2019). Save Time and Improve your Marks with CiteThisForMe, The No. 1 Citation Tool. [online] Available at:* [*http://www.citethisforme.com/*](http://www.citethisforme.com/%20) *[Accessed 12 Feb. 2019].*

*Wiegers, K. (1999). [online] Web.cs.dal.ca. Available at:*[*https://web.cs.dal.ca/~hawkey/3130/srs\_template-ieee.doc*](https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc%20) *[Accessed 30 Jan. 2019].*

# Appendices

## Appendix A: User Guide

## Appendix B: UML Diagrams

|  |
| --- |
| Figure B.1, Use-Case Diagram |
| Figure B.2, Class Diagram |