

William Cook

C010703m@student.staffs.ac.uk

**[Lego Star Wars]**

Technical Design Document

Contents

[Project Introduction 2](#_Toc190733266)

[Project Goals 2](#_Toc190733267)

[Challenges and Risks 2](#_Toc190733268)

[Hardware Requirements 2](#_Toc190733269)

[Platforms 3](#_Toc190733270)

[Target Platform 3](#_Toc190733271)

[Engine Specific Specifications and Limitations 3](#_Toc190733272)

[Engine Summary 3](#_Toc190733273)

[Systems and Diagrams 3](#_Toc190733274)

[System 1 Main UI 3](#_Toc190733275)

[System 2 Stud 4](#_Toc190733276)

[System 3 Character 4](#_Toc190733277)

[System 4 Building 5](#_Toc190733278)

[Optimisation and Profiling 5](#_Toc190733279)

[Profiling Systems 5](#_Toc190733280)

[Profiling Graphics 5](#_Toc190733281)

[Coding Standards 5](#_Toc190733282)

[Programming Standards 5](#_Toc190733283)

[Style Guide 6](#_Toc190733284)

[Commenting Rules 6](#_Toc190733285)

[Code Review Procedures 6](#_Toc190733286)

[Production Overview 6](#_Toc190733287)

[Moscow 6](#_Toc190733288)

[Timeline 6](#_Toc190733289)

[Budgeting 6](#_Toc190733290)

# Project Introduction

The Project aims to develop a LEGO-inspired game in Unreal Engine using Blueprints. The game will capture the satisfying gameplay elements in classic LEGO titles such as LEGO Star Wars: The Complete Saga, LEGO Batman, and LEGO DC Super-Villains. The project will bring back the nostalgia that made these games so memorable and loved, while aiming to modernise the experience using the up-to date engine to create high-detail and realistic materials.

The game is designed to provide a fun and engaging experience, focusing on the pick-up and play experience, using intuitive UI to show button prompts, sounds and animations to make it clear when a player picks up studs or interacts with objects.

## Project Goals

The goal of this project is to develop and create a LEGO-Inspired game within Unreal Engine using Blueprints. The game will feature core mechanics such as:

* Building System – Players can use connect bricks together to create a structure.
* Swappable Characters – A character-switching system allowing players to swap between different characters
* Stud Collection – Players can collect studs as an in-game currency
* Additional Mechanics(Interacting, AI, Attacking)

## Challenges and Risks

* Swappable Characters – Implementing this system to allow both characters to be able to interact with the environment, with one character being played by an AI. The challenge lies in ensuring smooth transitions between characters, where the game doesn’t break or cause issues when switching.
* Building System – Ensuring a smooth and seamless system, where bricks transition and snap together, with animations, satisfying sound effects without any bugs that can break immersion.
* AI Behaviour – Designing AI that feels engaging and useful. It’s important to ensure that friendly AI can interact with the environment, without getting stuck. The AI must be able to follow the players movement including jumping to places, attack and interact without causing frustration and breaking immersion for the player.
* Lego Feel – Recreating the satisfying feel and nostalgia when playing classic Lego games. Creating polished mechanics which include sound effects and animations.

## Hardware Requirements

Minimum System Requirements

* **Operating System:** Windows 10
* **Processor:** Intel Core i5-4460 / AMD Ryzen 3 1200
* **Memory:** 16 GB RAM
* **Graphics Card:** NVIDIA GTX 1660 Ti / AMD Radeon RX 590
* **Storage:** 256 GB SSD

# Platforms

## Target Platform

The game will be developed for PC. It will use Mouse and Keyboard controls and will not support controller input.

## Engine Specific Specifications and Limitations

* Disk Space – Unreal Engine projects can become very large due to assets. The game will be approximately 1.8GB of Disk space, which may increase with further development.
* Performance Optimisation – The game aims to be optimised for performance, especially when handling a lot of moving parts such as the bricks. Ensuring that mechanics don’t cause major FPS drops or even crashes.

## Engine Summary

* Unreal Engine 5.4.4
* Unreal Engines Physics System

No use of downloadable plugins, keeping the projects size to a minimum while avoiding any compatibility issues. Only using built-in features and tools provided by the engine.

# Systems and Diagrams

## System 1 Main UI

The main UI acts as the canvas panel where all other widgets will be put into and initialised from. Widgets that are created dynamically will not be created from the main UI and instead from the blueprint that requires a widget. The game mode will initialise the main UI and will have a reference to it.



**IA Plan**



## System 2 Stud

Player will pick up studs by walking into them, these studs will then move to the players stud count within the main UI. Studs can also be spawned which will impulse into the air and bounce.



**Class Diagram**



**IA Plan**



## System 3 Character

**Diagram**



**Class Diagram**



**IA Plan**



## System 4 Building

**Mechanic Diagram**



**Class Diagram**



**IA Plan**



# Optimisation and Profiling

## Profiling Systems

To ensure the game functions as intended, self-playtesting was done throughout development, to quickly find common bugs and issues. Additionally, a class playtest was conducted at week 5 to gather feedback from multiple players, identifying areas of improvement and bugs. Testing helped refine mechanics, UI, and overall player experience.

## Profiling Graphics

Graphics performance was tested by turning on the frame rate, memory usage and monitoring this during gameplay, to identify any frame drops and any other performance issues.

# Coding Standards

## Programming Standards

Throughout development, I have used unreal Engine’s naming conventions, ensuring consistency
This includes:

* Blueprints(BP\_)
* Sounds(Snd\_)
* Widgets(WBP\_)
* Textures(T\_)

To maintain performance and efficiency:

* **Casting has been minimised** – Using Blueprint Interfaces where possible to improve modularity and memory.
* **Avoiding “Get All Actors”** – Using overlap events to prevent unnecessary performance drops.

## Style Guide

Maintaining clean space:

* **Nodes are properly aligned** – Connections are kept straight using reroute nodes for better readability and avoid spaghetti.
* **Node Spacing** – Nodes are spaced enough a part to prevent overlap that may make it harder to read.
* **Organised Content Folder**
	+ **Animations** – Has all animations, montages
	+ **Blueprints** – Contains gameplay elements
	+ **Sounds** – All sounds and Sound Cues
	+ **Materials** – Most Materials, some are stored with their own character
	+ **UI** – All widget blueprints

## Commenting Rules

* **Key functionality is commented** – Complex parts of a blueprint is commented to help explain how it works.
* **Events and Functions** – These have been renamed appropriately

## Code Review Procedures

* **Iterative refinement** – Mechanics are often revisited to find areas of improvement.
	+ **Class-Based Structure** – Creating a Base Character class, with Base Player and Base Enemy as child classes, allowing for modular character management.

# Production Overview

## Moscow

|  |  |  |  |
| --- | --- | --- | --- |
| Must | Should | Could | Won’t |
| Playable Character | Swappable Characters | Selectable Characters | Multiple Levels |
| Camera | Interactable Levers | Polished Level | Co-op Multiplayer |
| User Interface | Building Pieces | Puzzles | Saving |
| Pick up and play | True Percentage | AI interact with lever | Dialogue |
| Collectables(Studs) | AI Enemies | Unique Characters | Climbing |
| Sounds | Spatial UI | Cutscenes |  |
|  | Attacking | Minimap |  |
|  |  | Destruction |  |

## Timeline

[PFTD Gant Chart](https://staffsuniversity-my.sharepoint.com/personal/c010703m_student_staffs_ac_uk/Documents/Prototyping%20For%20Technical%20Designers/PFTD%20Gant%20Chart.xlsx?web=1)

## Budgeting

**Development Time**

* The project was developed over 8 weeks, with final submission on Tuesday of Week 9.
* Time was allocated for research, design, prototyping, iterating and testing.

**Financial Costs**

* **Software** – Unreal Engine is free
* **Assets** – All assets were self-created or sourced for free from websites.