

**National College of Ireland**

2020/2021

Shane Mulrooney, x19454016, x19454016@student.ncirl.ie

Sean Fulton, x19518013, x19518013@student.ncirl.ie

Cree Gunning, x19302733, x19302733@student.ncirl.ie

Plague Panic

Technical Report

**Table of Contents**

[Executive Summary 2](#_Toc51756265)

[1.0 Introduction 2](#_Toc51756266)

[1.1. Background 2](#_Toc51756267)

[1.2. Aims 2](#_Toc51756268)

[1.3. Technology 2](#_Toc51756269)

[1.4. Structure 2](#_Toc51756270)

[2.0 System 2](#_Toc51756271)

[2.1. Requirements 2](#_Toc51756272)

[2.1.1. Functional Requirements 2](#_Toc51756273)

[2.1.1.1. Use Case Diagram 2](#_Toc51756274)

[2.1.1.2. Requirement 1 <Name of requirement in a few words> 2](#_Toc51756275)

[2.1.1.3. Description & Priority 2](#_Toc51756276)

[2.1.1.4. Use Case 3](#_Toc51756277)

[2.1.2. Data Requirements 4](#_Toc51756278)

[2.1.3. User Requirements 4](#_Toc51756279)

[2.1.4. Environmental Requirements 4](#_Toc51756280)

[2.1.5. Usability Requirements 4](#_Toc51756281)

[2.2. Design & Architecture 4](#_Toc51756282)

[2.3. Implementation 4](#_Toc51756283)

[2.4. Graphical User Interface (GUI) 4](#_Toc51756284)

[2.5. Testing 4](#_Toc51756285)

[2.6. Evaluation 4](#_Toc51756286)

[3.0 Conclusions 4](#_Toc51756287)

[4.0 Further Development or Research 4](#_Toc51756288)

[5.0 References 5](#_Toc51756289)

[6.0 Appendices 5](#_Toc51756290)

[6.1. Project Plan 5](#_Toc51756291)

[6.1. Ethics Approval Application (only if required) 5](#_Toc51756292)

[6.2. Reflective Journals 5](#_Toc51756293)

[6.3. Other materials used 5](#_Toc51756294)

# Executive Summary

Max 300 words. Summarise the key points of the report. Restate the purpose of the report, highlight the major points of the report, and describe any results, conclusions, or recommendations from the report.

# **Introduction**

## Background

Being gamers ourselves, we wanted a challenge to develop a game of our own and found our Team Project module to be a great opportunity to do so. Small development teams who make intricate and complex games has always fascinated us, so we sought to understand the difficulties and complexities of developing a game with a small group of people.

## Aims

The main goal for our project is to develop a game that is both fully functional and a fun experience for the player. Combining different elements from various genres of games has allowed us to create something uniquely fun for players who like to challenge themselves.

With the limited time we had to complete this project, we decided to create a game that does not necessarily have a main objective, but to instead challenge the player to better their score every time they play.

## Technology

At the beginning of the development process for our game, we set out to use any sufficient game development tool that would allow us to achieve the game we had envisioned. This led us to Pygame. Pygame is a game development module, as part of the Python programming language. Pygame games are developed without a UI, so the entire game is created using code. We ran into trouble early on with Pygame, finding it increasingly difficult to visualise the development of our game. At this point we decided that a game engine would be preferable for our development process. This is when we chose Godot as our main development technology.

Godot is a free to use game development engine which makes use of scenes and nodes, which act as classes, in a hierarchal, tree-like structure. Godot’s UI is easy to learn and great for beginners of game development. The scripting language is GDScript, which is very similar to Python, with some minor differences. All of these factors made choosing Godot as our main development engine very easy for us.

Godot allowed us to achieve elements of our game that we initially thought would be extremely difficult or even impossible to implement with our experience with game development, however, with sufficient research and perseverance we were able to achieve the goals we stated in the planning phase of our game.

## Structure

Provide a brief overview of the structure of the document and what is addressed in each section.

# **System**

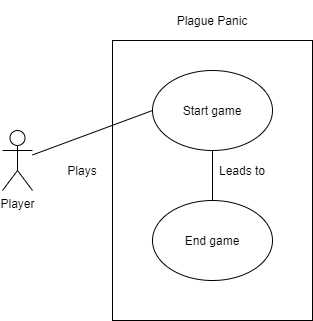
## Requirements

All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

## Functional Requirements

1. Functional Requirement: Start Game
2. Functional Requirement: End Game

## Use Case Diagram



## Requirement 1: Start Game

## Description & Priority

This requirement is of highest priority as it grants the player the ability to access the functions of the game.

## Use Case: Start Game

**Scope**

The scope of this use case is to allow the Player to start the game and access all the functions the game has to offer.

**Description**

This use case describes the Player starting the game.

**Flow Description**

**Precondition**

The system is awaiting to be launched by the user.

**Activation**

This use case starts when the Player runs the application.

**Main flow**

1. The Player runs Plague Panic application.
2. The Player selects Start Game from the Main Menu.
3. The Player plays the game.

**Alternate flow**

1. : Player quits game from Main Menu.
2. The Player opens Plague Panic.
3. The Player selects Exit Game from the Main Menu.
4. The game closes.

**Termination**

Upon termination this Use Case presents the End Game requirement.

**Post condition**

The Player has the ability to play the game.

## Requirement 2: End Game

## Description & Priority

This requirement enables the game to end when the character the Player chooses to exit the game.

## Use Case: End Game

**Scope**

The scope of this use case is to enable the game application to close when the Player decides to quit the game.

**Description**

This use case describes the Player exiting the game.

**Flow Description**

**Precondition**

The game is running.

**Activation**

This use case starts when the Player chooses the quit game option from the main menu or game over screen.

**Main flow**

1. The game is running.
2. The Player’s character dies.
3. The Player exits the game.

**Alternate flow**

1. : The Player quits the game.
2. The game is running.
3. The Player exits the game from main menu.

**Termination**

Upon termination the game application closes and the system ends.

**Post condition**

The system is finished.

## Non Functional-Requirements

Adequate Framerate:

## Design & Architecture

Describe the design, system architecture and components used. Describe the main algorithms used in the project. (Note use standard mathematical notations if applicable).

An architecture diagram may be useful. In case of a distributed system, it may be useful to describe functions and/or data structures in each component separately.

## Implementation

Describe the main algorithms/classes/functions used in the code. Consider to show and explain interesting code snippets where appropriate.

## Graphical User Interface (GUI)

Provide screenshots of key screens and explain what can be seen in each one.

## Testing

Player Movement:

Expected Outcome: User presses any of the directional keys (WASD), the Player character moves in the respective direction.

Actual Outcome: The user pressed the WASD keys and the Player character moved in the respective direction.

Result: PASS

Player Shooting:

Expected Outcome: When the user clicks the left mouse button a bullet should propel from the Player character and travel towards the direction of the mouse.

Actual Outcome: The user clicked the left mouse button and the bullet continuously travelled in opposite direction to the mouse position.

Result: FAIL

Player Shooting:

Expected Outcome: When the user clicks the left mouse button a bullet should propel from the Player character and travel towards the direction of the mouse.

Actual Outcome: The user clicked the left mouse button and the bullet travelled from the Player character’ sprite towards the position of the mouse.

Result: PASS

Zombie Spawns:

Expected Outcome: When each round starts a certain amount of zombies should spawn (Depending on the current round number.) in the set spawn points around the map.

Actual Outcome: The main level scene loaded and the correct amount of zombies spawned in their respective spawn points.

Result: PASS

Round System:

Expected Outcome: When the player character defeats the required amount of zombies to progress to next round the current round will be updated to the next round.

Actual Outcome: The player defeated the last zombie during the current round, the game progressed to the next round spawning more zombies for the player character to defeat.

Result: PASS

Start Game Button (Main Menu):

Expected Outcome: Once the user clicks the Start Game button on the main menu of the application the application will change to the main level scene.

Actual Outcome: The user clicks the Start Game button on the main menu and the application changes to the main level scene.

Result: PASS

Quit Game Button (Main Menu):

Expected Outcome: Once the user clicks the Quit Game button the application window will close

Actual Outcome: The user clicks the Quit Game button and the application closes.

Result: PASS

Play Again Button (Game Over Menu):

Expected Outcome: When the user presses the Play Again button is pressed the level scene restarts.

Actual Outcome: The user pressed the Play Again button and the intro cutscene played again.

Result: FAIL

Play Again Button(Game Over Menu):

Expected Outcome: When the user presses the Play Again button is pressed the level scene restarts.

Actual Outcome: The user pressed the Play Again button and the game restarted by loading the level scene.

Result: PASS

# **Conclusions**

Describe the advantages/disadvantages, strengths and limitations of the project

# **Further Development**

With additional time and resources, which direction would this project take?

# **References**

Please include references throughout your document where appropriate. See [here](https://libguides.ncirl.ie/referencingandavoidingplagiarism) for a guide on referencing from the NCI library.

# **Appendices**

This section should contain information that is supplementary to the main body of the report.

## Project Plan

## Collaboration Summary

Player Character Git Log:

