

**National College of Ireland**

2020/2021

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

Sean Fulton,<Student Number>,<Student Email>

Cree Gunning,<Student Number>,<Student Email>

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

Why did you undertake this project?

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

What does the project aim to achieve?

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

What technology will you use to achieve what you have set out to do and how will you use it?

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

This section lists the functional requirements in **ranked order**. Functional requirements describe the possible effects of a software system, in other words, what the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe how the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

Short, imperative sentence stating highest ranked functional requirement.

## Use Case Diagram

## Requirement 1 <Name of requirement in a few words>

The heading of this section should read, e.g., “Requirement 1: User registration” or “Requirements 1: Participant takes test”

## Description & Priority

A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

## Use Case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

**Scope**

The scope of this use case is to …….

**Description**

This use case describes the ………..

**Use Case Diagram**

Diagram should highlight actors and uses cases……..

**Flow Description**

**Precondition**

The system is in initialisation mode……..

**Activation**

This use case starts when an <Actor>…………

**Main flow**

1. The system identifies the ………….
2. The <Actor> …………...(See A1)
3. The system …………..(See E1)
4. The <Actor> ………….

**Alternate flow**

A1 : <title of A1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 3 of the main flow

**Exceptional flow**

E1 : <title of E1>

1. The system …………..
2. The <Actor> ………….
3. The use case continues at position 4 of the main flow

**Termination**

The system presents the next ……….

**Post condition**

The system goes into a wait state

**List further functional requirements here, using the same structure as for Requirement1.**

## Data Requirements

## User Requirements

## Environmental Requirements

## Usability Requirements

## 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

Describe any testing tools, test plans and test specifications used in the project.

# **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

Summarise and provide evidence of collaboration throughout the project. Evidence may take the form of Trello boards, Chat logs, Git Logs etc.