

Project Name

XX.XX.20XX

**─**

Your Name

# 

# Changelog

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Changes** |
| 1.0.0 | 08/05/2023 | Initial Setup |
| 1.1.0 | 10/05/2023 | Adding basic information |
| 2.0 | 15/05/2023 | Completed research answers |
|  |  |  |

# Contents

[Changelog 2](#_Toc98447406)

[Contents 3](#_Toc98447407)

[Introduction 5](#_Toc98447408)

[Rationale 5](#_Toc98447409)

[Background 5](#_Toc98447410)

[Terminology 5](#_Toc98447411)

[Non-Goals 5](#_Toc98447412)

[Proposed Design 5](#_Toc98447413)

[Software and Hardware Requirements 5](#_Toc98447414)

[Research 5](#_Toc98447415)

[Procedural Programming 6](#_Toc98447416)

[Object-Oriented Programming 6](#_Toc98447417)

[Functional Programming 6](#_Toc98447418)

[Logical Programming. 6](#_Toc98447419)

[System Architecture 7](#_Toc98447420)

[Architecture Diagram 7](#_Toc98447421)

[Data types 7](#_Toc98447422)

[Data Model 7](#_Toc98447423)

[Interface/API Definitions 7](#_Toc98447424)

[Risks 7](#_Toc98447425)

[Alternatives 7](#_Toc98447426)

[Pseudocode 7](#_Toc98447427)

[System Pseudocode 7](#_Toc98447428)

[UML Diagrams 8](#_Toc98447429)

[UML Class Diagrams 8](#_Toc98447430)

[UML Activity Diagrams 8](#_Toc98447431)

[UML Sequence Diagram 8](#_Toc98447432)

[UML Communication Diagrams 8](#_Toc98447433)

[UML State Diagrams 8](#_Toc98447434)

[Sign Off 8](#_Toc98447435)

[Testing 9](#_Toc98447436)

[Errors and Buggs 9](#_Toc98447437)

[Evaluation 9](#_Toc98447438)

[Reflection 9](#_Toc98447439)

# 

# Introduction

## Rationale

The task is to create a hangman-style game to demonstrate my programming abilities.

## Background

/ Describe any historical context that would be needed to understand the document, including legacy considerations. For example, what is the history of a menu system? What is its purpose? /

## Terminology

/ If the document uses any special words or terms, list them here. For example, what does GUI stand for? What does UI/UX mean? This section is for terms you will use fill this in after you make your document. /

Game Engine

Unity

C#

UI

UX

GUI

Canvas

Software Development Platforms

Sprint

## Proposed Design

/ Start with a brief, high-level description of the project. The following sections will go into more detail. For example, summarize what it is you are needing to make. /

The proposed game is a hangman-style game that has a Main Menu, Game, and Pause Menu and is created within one Unity scene using C# and implementing a Canvas UI.

The game will be made according to the following pseudo-code:

**//MAIN MENU** **//Play** **//Starts new Game** **//Exit** **//Quits to Desktop** **//PAUSE MENU** **//Return to Game** **//Toggles on and off Pause Menu** **//Return to Main Menu** **//Hides Game and shows Main Menu** **//Exit Game** **//Quits the game**

**//GAME** **// Have a collection of words that can be used** **// Choose a word at random and store it in a variable** **// Display the length of the word to the user** **// Have several incorrect guesses that as they are marked off draw a hang man** **// While the user has guesses left and has not guessed the word** **//Prompt the user to guess a letter** **//If the guess is correct** **//Increment correct guesses by 1** **//If the guess is incorrect** **//Increment incorrect guesses by 1** **//Draw the next part of the hangman** **//If the incorrect guesses have all been marked off tell the user** **//They lost and allow user to loop gameplay** **//If correct guess of the word** **//Tell the user they won** **//Prompt to play again**

## Non-Goals

Hangman is a game that everyone has played many times, so it is difficult to make it interesting.

There are non-programming goals that I doubt I will have time to implement, but if I could, I would like to implement:

* Graphics and animations to make the game more interesting.
  + Maybe instead of a hangman, animate another penalty such as someone trapped in a device that will kill them or a bomb that must be defused.
* Sound and voice effects would also have a big impact.

Programming goals:

* High-score system in which the player’s score increases with each word.
* Word score based on scrabble scoring to allow difficulties (see below).
* High-score table – Player enters name on death or quitting with top 10 score.
* Difficulties – Based on online dictionaries for Beginner, Intermediate and Advanced speakers.

## Software and Hardware Requirements

/ A list of all software being used, their versions and costs, as well as the targeted hardware constraints. Considerations should include what platform are you releasing to? /

The software used for this project will be:

* Unity v2021.3.19f – Free with licensing restrictions
* Visual Studio 2022 – Free under a student licence (used at TAFE)
* Rider 2022.3.2 – Free under a student licence (used at home)
* Clip Studio Paint Pro v2.0 - $50USD

The game will made on Windows 10 and will be released on Windows; however, ideally the game will also be released on Android and Linux.

The project will not be released on Mac or iPhone due to content restrictions and a lack of testing hardware available.

# Research

Software Development Platforms

/ What are some Software Development Platforms for Games? Name at least three and give a brief description /

Unity – A free (with restrictions) C#-based game engine with built-in tools for game development. It has been used by indie developers for many years due to it’s generous licensing model, and because of this, has *many* tutorials and forum posts covering almost every subject and potential problem.

Unreal – Another free (with restrictions) game engine with built-in tools for development. Unreal is owned by Epic Games and as such has larger development budget which results in more advanced and cutting-edge features than Unity allowing easier development AAA-style games with advanced graphics. However, Unreal uses C++ instead of C# which has a much larger learning curve and is not as accessible to beginner programmers. Also, it has not been available to indie developers for as long as Unity, so it does not have nearly as many online resources for learning. Although Epic is working to add as many tutorials and online resources as possible to catch up with Unity.

Godot – A completely free and open-source game engine without commercial licensing restrictions that is community supported and not-for-profit. Due to being community made and run, Godot is not as advanced as Unity or Unreal; however, it can use C# or GDScript which is its own light-weight language that may be more accessible to beginners.

All three of the above development platforms have built-in tools for making development easier, support both 2D and 3D games, and are able to build your game on multiple platforms.

Software Development Methodologies

/ What are Software Development Methodologies? Name 10 and give a brief description of each. Compare them and explain which ones are good for Game development. /

1. Agile Development
   * Focuses on the project itself and allows for constant alterations based on feedback and internal changes. The Agile process is free of a rigid framework with the development process divided into short sprints to allow fast results and feedback.
2. Waterfall Development
   * Classic software development model. A rigid framework in which the project stages are done linearly with a stage not started until the last is finished, and the project only moves forward. Easy to stay on-task but very difficult to react to changes.
3. Extreme Programming (XP)
   * XP is an agile methodology which works best for creating software in an unstable environment. It allows for greater flexibility but requires constant meetings with team members and stakeholders. The development team meets to discuss plans and issues rather than writing documentation which makes them more personally committed but can make managing the project and providing estimates very difficult.
4. Lean Development
5. Scrum Development
6. Spiral Development
7. Prototyping Methodology
8. Feature Driven Development
9. Rational Unified Development
10. Joint Application Development

Integrated Development Environment

/ What does Integrated Development Environment or IDE mean? Give an example of an IDE used for Game development. /

S.O.L.I.D Design Principles

/ What does S.O.L.I.D stand for in reference to Object Oriented Design Principles? Explain all 5 principles within S.O.L.I.D. /

Programming Methodologies

/ Describe the following Programming Methodologies/

### Procedural Programming

### Object-Oriented Programming

### Functional Programming

### Logical Programming

Open-Source Development tools

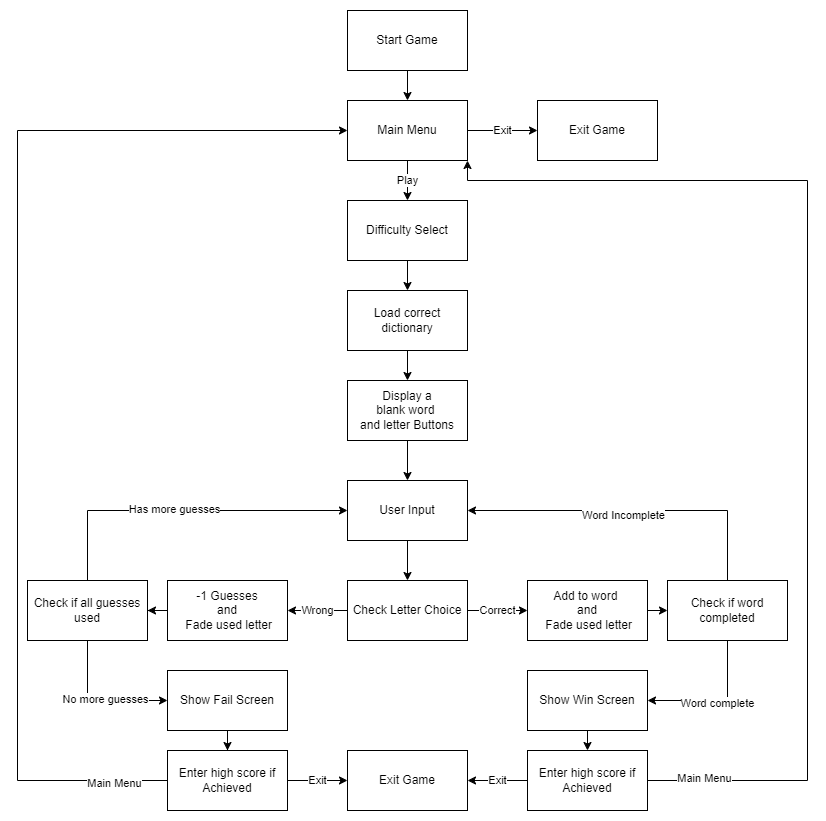
/ What are some Open-Source Development tools that can be used by Game Programmers? What does Open-Source mean? /

### 

# System Architecture

## Architecture Diagram

/Diagram of the planned system – This is like a mind map or flow chart of what you are going to make or what you need to do/



## Data types

/ Describe the main data types you will be using and how they work. /

## Data Model

/ Describe how you will use that data and how it is stored and used. /

## Interface/API Definitions

/ Describe the various components and libraries you will be using that are inbuilt into unity. For example, GameObject, Image, SceneManagement, UnityEngine…etc. Link the Unity API Manual to show where to find information on those elements. /

## Risks

/ If there are any risks or unknowns, list them here. Also, if there is additional research to be done, mention that as well. /

## Alternatives

/ If there are other potential solutions which were considered and rejected, list them here, as well as the reason why they were not chosen. /

# Pseudocode

## System Pseudocode

/ Written plan of the code and how you are going to write it in English. This is where you expand from what you were given in the brief. /

# UML Diagrams

## UML Class Diagrams

/ Class diagram is a graphical notation used to construct and visualize object-oriented systems. A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the systems. /

## UML Activity Diagrams

/ Sequence diagram to describe dynamic aspects of the system. /

## UML Sequence Diagram

/ Interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. /

## UML Communication Diagrams

/ Communication diagram is an extension of object diagram that shows the objects along with the messages that travel from one to another*.* /

## UML State Diagrams

/ State diagram shows the different states of an entity. /

# Sign Off

Name

James-Rae

Role

Lead Programmer

Signature

Date

[Click/tap to select date]

# Testing

## Errors and Bugs

Outline the test classes used. Add rows to table as required.

|  |  |  |  |
| --- | --- | --- | --- |
| **Class Name** | **Description of Error** | **Screenshots of testing** | **Solution** |
| [Name of Class or Object that the error is connected to] | [Description of the error/error message] | [Add and resize relevant screen shots] | [Explain solution/fix to error] |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Evaluation

## Reflection

/Provide a self-reflection on your performance. /