How To Fail Your Research Degree Digital Version

Welcome to the how to fail your research degree digital version project documentation. The goal of this document is to offer a comprehensive overview of the project and its numerous components. It applies to the project's developers, project managers, and other stakeholders. A complete project overview, technical requirements, coding standards, development plans, and a user manual are all included in this paper. We would like to thank the whole team for their efforts, which enabled us to create this paper.

Contents

1. Project Overview:

In recent years, more and more students are worried that they will not succeed in graduating, so daisy has created a card game to help postgraduate students get their degrees. However, the card game is expensive and not easily transportable, so this project has digitised the card game 'How to fail your research degree' to help more schools and students.Success Criteria: 80% student and teacher satisfaction.

2. Design Document:

a). Asset
- Audio (Game music)
- CardPrefabs(Prefabs for cards)
- Resources (Picture)
- UI Elements(Picture for UI production)

b).Material(Materials generated by shader)
- Scenes (Scenes of the game)
- Scriptes (Stored C# scripts)
    - AnimationScenes
    - HelpScenes
    - HelpScenesTest
    - Editor
    - MainScenes
    - MainScenesTest
    - WaitingRoomScenes
    - WaitingRoomScenesTest
    - ResultScenes
    - ResultScenesTest
    - OtherTest

c).Shader (Shader)

d).Package(Location of storage for various libraries)

3. Technical Specifications

1. Unity Engine Core
    Scene and game object management
    Component system
    Asset manager
    Prefab system
2.Rendering
    Materials and shaders
    Texture mapping
3.Audio
    Audio effects and filters
4.User Interface (UI)
    UI elements and layout
    UI event system
5.Scripting and Programming
    C# scripting
    Mono/.NET runtime
    Visual Studio integration
    Editor scripting

4. Coding Standards

1.Naming conventions:
    - Use PascalCase for naming classes and structs: MyCustomClass
    - Use camelCase for naming methods, properties, local variables, and parameters: myCustomMethod
    - Use _camelCase for naming private variables: _privateVariable
    - Use all uppercase letters and underscores for naming constants: CONSTANT_VALUE
    - Use descriptive names and avoid abbreviations and single-character names
2.File structure:
    - Each class, struct, and interface should have a separate file with the file name matching the name of the class, struct, or interface
    - Organize folders and namespaces according to functionality and logical structure
3.Code formatting:
    - Use 4 spaces for indentation, not tabs
    - Use spaces around operators and after commas
    - Place the opening brace { on the same line as the definition of the class, method, property, struct, and other code blocks, and align the closing brace } with the last line of the code block
    - Use empty lines to separate different code blocks, methods, and logical sections
4.Comments:
    - Use single-line and multi-line comments to provide detailed explanations of the purpose and logic of the code
5.Code organization:
    - Follow the Single Responsibility Principle, ensuring that each class and method is responsible for a single functionality
    - Use properties to encapsulate fields within a class, protecting the internal state and providing an interface for external access
    - Place reusable code in separate methods or extension methods
6.Error handling:

- Use try-catch statements to handle code that may raise exceptions
- For expected error situations, use conditional statements to check and take appropriate measures

7.Performance optimization:

- Avoid using performance-expensive methods like GetComponent or FindObjectOfType in Update methods
- Use object pooling to reuse objects and avoid frequent creation and destruction of objects

These coding guidelines examples can be adjusted according to project and team requirements.

5. Development Plan

Team members

- Ben O'Hara 2456365o@student.gla.ac.uk
- Jiacheng Zhu 2584876z@student.gla.ac.uk
- Kai Wang 2539930w@student.gla.ac.uk
- Xinuo Zhou 2550838z@student.gla.ac.uk
- Pok Chung 2512232c@student.gla.ac.uk

~/09/2022---09/11/2022

All team members: Design prototype in Figma

09/11/2022---29/11/2022

Kai: Write part of the game code

All other members: Prepare second customer meeting

02/01/2023---17/01/2023

Ben: Finished the help scene and part of timer

Xinuo: Worklate tile

Pok: card import and build card attributes

Jiacheng: Part of waiting room

Kai: Result scene, win checker and active card arrow check

17/01/2023---13/02/2023

Ben: Timer, web socket and server

Jiacheng: waiting scene, newbie orientation scene, profile and event card

Xinuo: setting panel

Pok: unit test

Kai: fix game bugs and event card implementation

13/02/2023---15/03/2023

Ben: web socket and server

Jiacheng: web socket and multiple player

Xinuo: merge forks and prepare dissertation

Pok: unit test

Kai: fix game bugs and CI/CD develop

15/03/2023---22/03/2023

Demostration:

Ben: Meeting chair & Project Motivation

Jiacheng: Main Achievement &Technical Challenge

Xinuo: Technical Decision & Reason

Kai: Game demo

Pok: Questions

Project allocation:

Ben: dissertation

Jiacheng: dissertation

Xinuo: Game Guide

Kai: Process mark

Pok: Product mark

6.User Manual

Mac,Windows

If your computer is Windows system skip this session

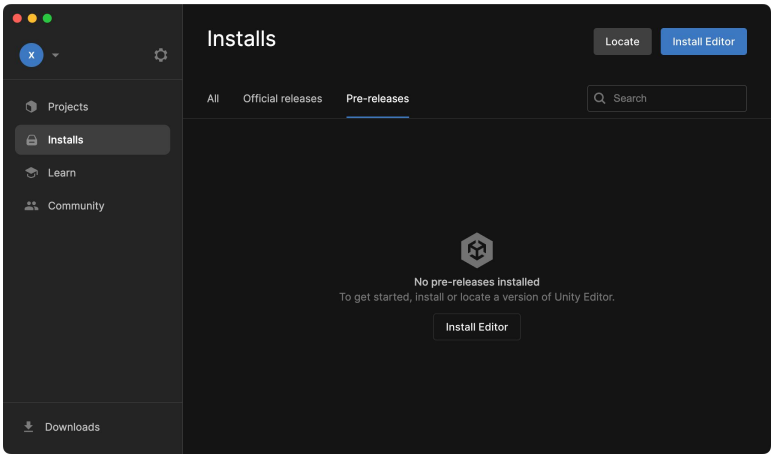
Application Installs

Install applications if you want to edit and create the code base yourself.
1. Install Unity - Project Built with Version 2021.3.11f1
2. Setup Git
3. Install Visual Studio - an editor that allows you to create the server executable.

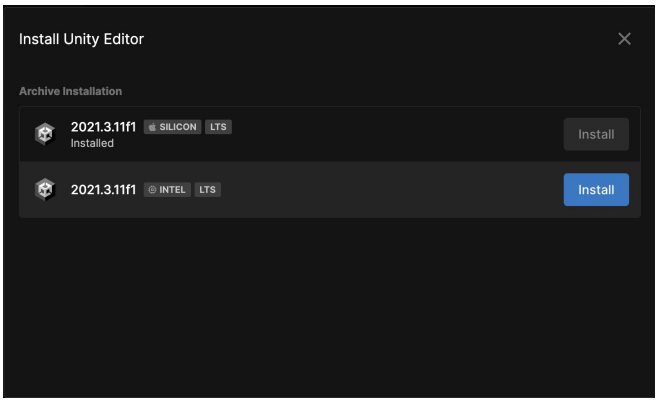
Install Unity

Visit https://unity.com/ and create a Unity ID account or you can use a google account, apple account, or facebook account to log in directly. We only need to get the free version here. On this website https://unity.com/download there are three steps, first select your computer system and download the unity hub via the link, then find and download the version according to the one mentioned before, here you can download it directly from https://unity.com/releases/editor/archive and finally you are ready to start the project.

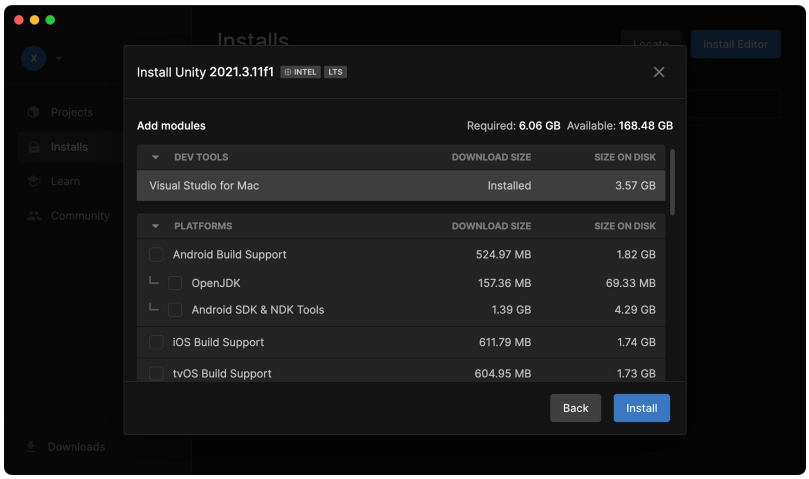
1). Of course, you may download Unityhub, open it, install it in the settings on the left, there is an install Editor, find the previous version in the archive, and if not, it offers a link to the website.



2). Once you've located version 2021.3.11, if you are mac system decide whether to install silicon or intel based on the chip in your machine.Windows can be downloaded directly

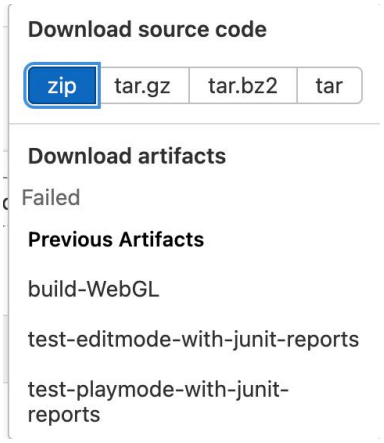


3). Click on install; if you already have Visual Studio installed, you don't need to install it; if not, kindly pick it, then choose mac build support in platforms(Windows: choose Windows build support), your language, and lastly documentations, then install it, and your first step is complete!



Download Source Code

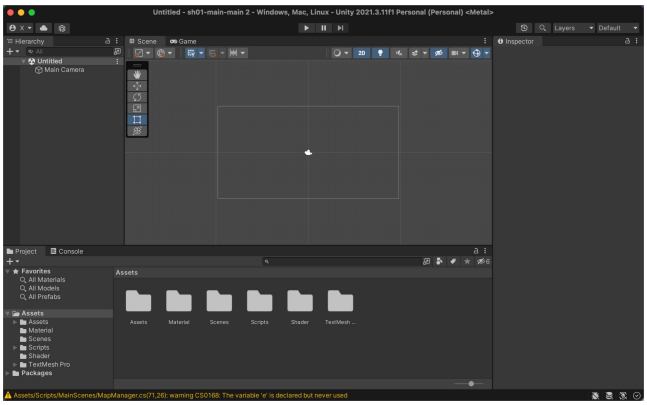
There are two main ways to download code. You can download the code directly from GitLab . Go to the website https://stgit.dcs.gla.ac.uk/team-project-h/2022/sh01/sh01-main to download the source code directly



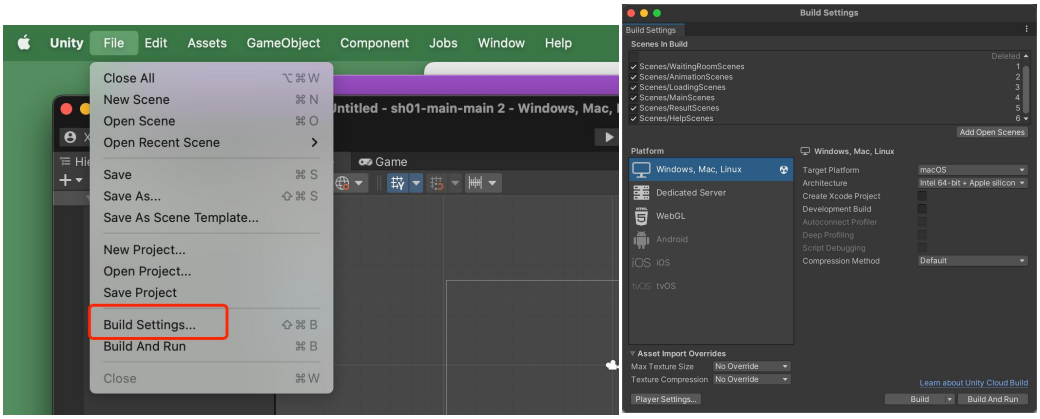
Unzip the archive into a folder of your choice. Alternatively, you can use Git to clone the repository, however, you must have Git installed to use this method. Go into a folder with cd in the terminal then you can clone the repository by typing "git clone https://stgit.dcs.gla.ac.uk/team-project-h/2022/sh01/sh01-main.git".

Build and run

Click open on the unity hub, then choose the source code you just acquired and open it. When the image is opened, it appears like this.



Don't worry if it doesn't appear to be ready to play; in the upper left corner of File, pick Build Settings.



Once inside, check all the scenes and finally click build and run

Then store it where you want it to be. The perfect scene appears, turn on the sound and start your game