City, University of London

BSc (Hons) Computer Science with Games Technology

Individual Project Report

Virtual Laboratories

3D Virtual Science Experiment in Unity

By: Oliver Reekie

Email: oliver.reekie@city.ac.uk

Project Supervisor: Jacob Howe

Proposed by: Oliver Reekie

Report Word Count: 11,914

Abstract

The goal of this project is to recreate the investigative experience of a science practical virtually, to make the experience more accessible to potential users. This project will recreate the experiment 'Determination of g from free fall', educating the user on the scientific knowledge they are testing as well as having them design, conduct and analyse the results from their investigation, facilitating knowledge of the scientific method. Following the experiment, the user will be graded and given feedback on their performance, allowing them to repeat and tune their skills. The software developed in this project will act as a platform for more experiments to be added to.

Table of Contents

Chapter 1: Introduction

1.1 Description of Problem

One of the most important aspects of a science education is the process of conducting investigations and experiments, leading to a greater understanding of the concepts being studied and the process of the scientific method (How important is Practical Science in the Classroom? - Innova Design Group, 2022). However, to undertake many of these experiments, specialist facilities and instruments are required, which prevents a vast amount of people from the benefits of a full scientific education.

This project attempted to remedy this issue, by realistically simulating the laboratory environment and bringing it to users in an accessible, cost-effective manner. Virtual Laboratories is a platform, on which users can play through the process of an experiment through the lens of a first-person videogame. These games are educational experiences that attempt to recreate the learning from experiments as closely as possible. This project focuses on the first experiment added to the platform 'Determination of g From Free Fall'.

This practical sees the user systematically dropping a card through a light gate that measures the time taken for the card to fall through. The user can vary the height at which the card is dropped to give a variety of values that can be analysed through a graph to give an approximation for the value of gravity (g).

The process of the experiment is as follows:

1 The player reads through the informational pages to plan the experiment
2 The player builds the apparatus
3 The player systematically drops the card through the light gate at a variety of heights
4 The player tabulates and graphs their results using a line of best fit
5 The player receives a grade and feedback to educate them on how to improve in future experiments

1.2 Project Objectives

The objective of the project was to recreate a science experiment virtually to allow for an accessible educational experience. This is split into the following sub-objectives:

1 This project shall educate users on the scientific method and how to conduct an experiment
2 This project shall recreate a laboratory virtually with objects that are used realistically
3 The project shall create a table and graph screen that allow the user to store and analyse results
4 The project should provide a grade and feedback to the user to foster improvement
5 The project shall educate me on the use of new software, specifically through Unity and C#

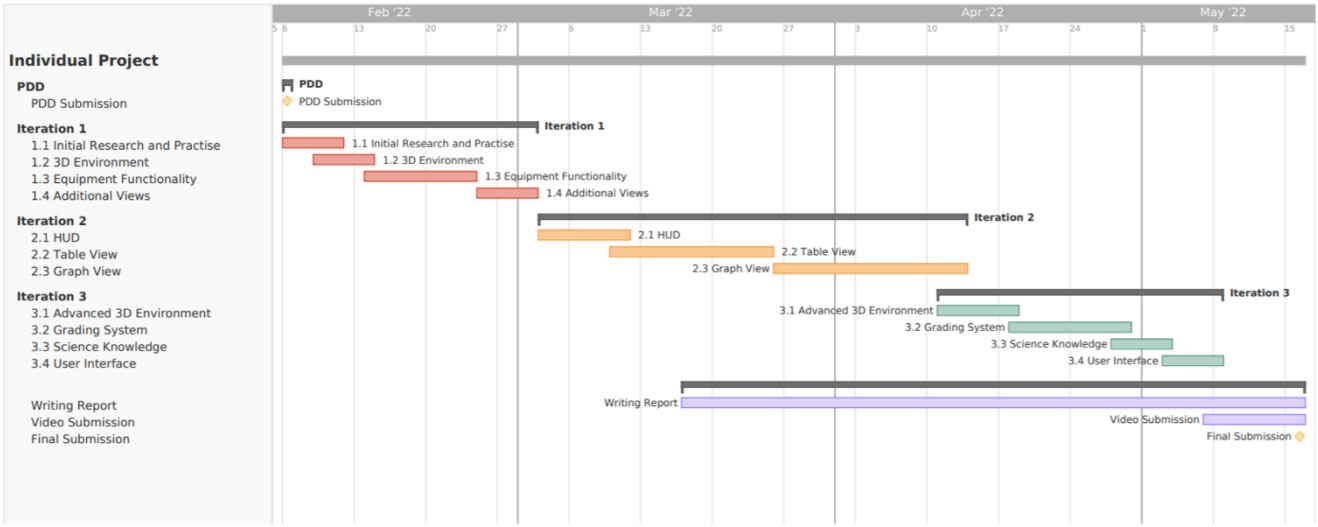
1.3 Project Beneficiaries

The project had four intended beneficiaries. The first and primary beneficiary was the user of the system who will be using this as a supplementary tool to improve their scientific education. The second beneficiary was myself as I was teaching myself to use Unity and C#, developing the range of software I can use and creating a portfolio piece that can demonstrate this ability. The third beneficiary was society as a greater understanding of science produces a generation more able to create advancements that will drive humanity forward. Finally, the last beneficiary was software developers that wish to create their own practical experiment but require a platform to showcase and reach potential users.

1.4 Work Performed

The structure of the project followed the Gannt chart shown in my PDD. Whilst there were changes made to functions within the game, the main stages and timeline were kept consistent, with each of the three development builds matching specific project objectives (1 -3).

The process followed an agile methodology, with work divided into sprints that allowed for testing throughout and changes to components. Initially, the system was designed by planning out requirements and outlining tests to ensure these were met, along with several class diagrams mapping the various variables and functions in the software. UI designs were developed for the various HUD elements. The implementation saw weekly sprints that completed different components of each build outlined in the Gannt chart. The game was regularly tested along with running final tests cases upon the completion of the implementation to ensure the requirements were met. A questionnaire was given out at the end of build two, and a second evaluative questionnaire given upon completion of the software, that allowed me to use user feedback to test and analyse my software, as well as guiding beneficial changes and modifications.



Gannt Chart showing project timeline

1.5 Assumptions and Limitations

This project followed two main assumptions: The first being that the number of online resources was sufficient to educate myself on the core functions of Unity. This ensured that the development process was not delayed by my spending large amounts of time learning the basics. The second was that it would be easy to find art assets and 3d models that I required, or that the models that I could not find were simple enough to be able to be created in blender with no primary experience in 3d modelling.

The project was limited primarily through performance and optimisation. Due to time constraints, and my beginner knowledge of the Unity engine, ensuring that the game ran with minimal performance issues was not a viable project objective. Whilst the game was tested to ensure it ran without issue on multiple devices, maximising efficiency of game processes was not a priority.

The second limitation was the ability to only produce one practical experiment. The software acts as a platform upon which multiple experiments can be added, as well as the complete determination of g practical, however as each experiment requires a vast amount of work to develop, this was the only one created.

Chapter 2: Output Summary

2.1 Unity Project

This is the main output and is the software created. This is a virtual experiment that teaches the user about scientific principles whilst simulating the experiment determination of g from free fall. This is a zip file containing the Unity project, the source code created and the assets it uses. It also includes an executable file that can run the game. The source file includes 36 C# scripts with 3535 lines of code (not including comments) of which 241 lines were reused.

File Type	.zip
Recipient	Players and Examiners
Benefit to Recipient	An educational game that has the user play through a realistic virtual science experiment
Results	5.1.3
Location	Appendix K

2.2 Requirements and Test Cases

A list of the 11 functional and 5 non-functional requirements of the project as well as 13 test cases to ensure these are met.

File Type	.pdf
Recipient	Myself
Benefit to Recipient	Used in software analysis and testing during project development
Results	5.1.2, 5.4.2
Location	Appendix H

2.3 Class Diagrams

Four use case diagrams, created to map out the classes, functions and variables along with how they connect to one another. The diagrams consist of: Build 1, Build 2, Build 2 after User Feedback, Build 3

File Type	.pdf
Recipient	Myself
Benefit to Recipient	Used in software design to guide software development
Results	5.2.1
Location	Appendix I

2.4 UI Designs

Three sets of UI designs for the various HUD elements. The first was created at the start of the project, with the subsequent designs being added to reflect design changes in the system

File Type	.pdf
Recipient	Myself
Benefit to Recipient	Used in software design to guide software development
Results	5.2.2
Location	Appendix E

2.5 User Feedback Questionnaire and Results

This is the consent form, participant information sheet and user feedback questionnaire along with the feedback it received. This was given after build 2 was completed to get feedback on what elements should be changed about build 2 and what should be added to build 3.

File Type	.pdf
Recipient	Myself
Benefit to Recipient	Used to get feedback to make modifications to build 2 and to collect ideas for additions to build 3
Results	5.4.3
Location	Appendix C, Appendix D

2.6 Evaluation Questionnaire and Results

This is the consent form, participant information sheet and evaluation questionnaire along with the feedback it received. This was given after the software was complete to collect the mark and scores the user received after playing the game as well as their previous science experience. This was used to evaluate the success of the software in providing an educational experience.

File Type	.pdf
Recipient	Myself
Benefit to Recipient	Used to collect user scores after playing the game to evaluate the success of the project in creating an educational experience
Results	5.4.3
Location	Appendix F, Appendix G

Chapter 3: Literature Review

3.1 Application Domain

Between a lack of access to specialist facilities and an increased understanding of the benefits of at home learning, the development of software to facilitate this has become increasingly more essential. An article looking into the equipment found in state-funded schools shows that 30-60% of students are missing out on engaging in practical experiments due to limited resources despite schools recognising that a complete science learning experience is "Vital for future prosperity" (Burns, 2022). Completing science practical's is a very important part of learning scientific concepts and the process of an investigation (How important is Practical Science in the Classroom? - Innova Design Group, 2022). Therefore, it is crucial that we make the required experiments more accessible, to allow society to flourish.

A good deal of research has taken place into the effectiveness of virtual science practical's on scientific education. Examination shows that virtual experiments are better at facilitating understanding of complex topics, adapting to specific user requirements, and providing immediate feedback on activity (Heradio et al., 2016).

An investigation into 3D vs 2D experiments shows that 3D realistic environments are far superior at educating the users. With 2D simulations (such as PHET Interactive Simulations) having "weak attractiveness and weak reality", compared to realistic 3D environments with high-definition graphics which are used very effectively already in military training (Liu et al., 2015).

3.2 Similar Software

3.2.1 Labster

Labster (https://www.labster.com/) is a platform that contains over 200 science experiments and practical's that the user can complete to enhance their scientific understanding. Their research and accreditations prove the benefit of virtual experiments to students, something that encouraged my exploration of a similar topic. These practical's are all first person and take place in a lab, recreating a complete experience. However, this software is run on an expensive subscription basis and does not have the practical I am creating. My software therefore fits this gap in a manner that is much more accessible to the general population.

3.2.2 PHET Interactive Simulations

PHET Interactive Simulations (https://phet.colorado.edu/) has a great many science experiments that can be completed online; however, they are not meant to be a realistic practical experience. Instead of a first-person virtual reality experience, they opt for the user moving items in a top down, 2D world. The success and widespread use of this software proves the benefits of completing at home science experiments and has inspired some of the 2D screens that I will be using. However, my project will take a 3D virtual reality approach which should improve the learning experience of the user.

3.2.3 Inch by Inch

Inch by Inch (Dare Looks, 2004) is a videogame developed by LevelXProject and Dare Looks that features involve a first-person character completing racing to complete a science experiment. I played this game during my research as many of the features in this game will translate into my own software. One example of this is the mechanic of building the practical set-up. In the game, the user carries an object, that has physics functionality to increase the realism, and then places this object into another one. These two models are then seamlessly replaced with a model of the two objects combined.

3.3 Algorithms and Architecture

3.3.1 Engine

The three possible game engines I considered using are Unity, Unreal and the City Engine (provided in Advanced Games Technology). Unity is known as the "go-to development environment for indie games" due to its speed and beginner friendly development. Its plan is free to use, there are vast Unity docs to support improvement and a huge asset marketplace, supporting my requirements for 3D models (Dealessandri, 2022). Unity uses C# which meets one of my goals of educating myself with a new coding language. Unreal's emphasis lies in visual design, and whilst can create far superior environments than Unity, this is not an important consideration in my project (Eldad, 2022). Unreal, like the City Engine, uses C++ which I am familiar with and so this would not be an effective tool for my development. The city engine also requires everything to be hard coded which would introduce so much more work that I would be forced to sacrifice the grading system in my own project. I have also read the 'Unity Game Development Cookbook' which details many functions and uses of the editor in creating effective games. (Butterfield-Addison, Manning and Nugent, 2019)

3.3.2 Design patterns

Singletons represent a very useful mechanic for my project, creating a single class that can be easily accessed from any other class. This is very useful for keeping track of the current game attributes. Singletons have the benefit of not being able to be reproduced accidentally by other classes so I can also be sure the variables that are being accessed are the ones I intend to. (Butterfield-Addison, Manning and Nugent, 2019)

Using multiple states are also an essential part of my project, with each set stage of my game needing to be clearly defined and recorded. Using states reduces the complexity and can be easily designed using a finite state machine. The ability to switch between states is also far simpler than using alternative methods. (Kushwah, 2022) (Design Patterns and Refactoring, 2022)

3.3.3 Raycasting and Layers

Raycasting is a very useful tool that involves firing an invisible ray from one point to another, detecting all collisions it causes. In a first-person game this can help easily figure out where the player is looking and what objects they would like to select. I will use this extensively in my game (Glover, 2017). However, raycasting can cause issues by causing unwanted collisions. To alleviate this, I will organise my game objects into layers. "Layers in Unity help indicate functionalities across GameObjects". This will allow me to classify what objects will collide with ray's as well as what objects can interact with users and other objects (Working with the Layer Editor - Unity Learn, 2022).

Chapter 4 Methods

4.1: Methodology

My project used an agile methodology, an iterative approach, that allowed me to break down a large software product into smaller, more manageable stages. By iterating over these planned builds, I produced sets of functions and improvements, regularly testing them. This is a huge benefit of an agile methodology as "faults are detected earlier and can be fixed before it increases in severity" (Kumar and Bhatia, 2022). This also allowed me to get feedback on each build, giving me improvements that I could make in my next iteration. "As handling change in requirements is the main feature of agile methodology" (Kumar and Bhatia, 2022) I could consistently add new ideas or adapt requirements based on my feedback. This is the biggest motivation for choosing an agile instead of waterfall methodology, the ability to add and changes ideas during development is important to the project's success. As well as the lack of flexibility, "Once the end goal is established, Waterfall does not involve the client" (Hoory and Bottoroff, 2022) which goes against my plan of continually testing based on user feedback to provide improvements.

4.2 Analysis

4.2.1 Research

Due to my lack of experience with Unity, I needed to understand the basics before beginning my development. To do this, I followed a tutorial on YouTube (freeCodeCamp.org, 2021) that allowed me to create a prototype game to get used to the basics of the system. I chose this video due to its high rating and viewership, that it came from an established, proven channel, and that the 7-hour tutorial covered all elements I would need to for my initial builds.

I followed a similar process for my use of blender, a software I had never used before, but needed to as there were multiple required models that did not exist. I followed a series of tutorials on YouTube (Blender Guru, 2019) that taught the basics of 3d modelling, chosen as it was one of the highest rated tutorials on the platform.

4.2.2 Requirements

After breaking the project down into three builds, the requirements were further broken down into several functional and non-functional requirements, with the functional requirements defined as "what the system should do" (ReQtest, 2012) and the non-functional requirements "how the system performs a certain function" (ReQtest, 2012). These are important to set out as they help guide development and to "keep project team going in the right direction" (ReQtest, 2012). I then created test cases for each of the requirements to be run at the end of the development to ensure the requirements were met (see Appendix H).

4.3 Design

4.3.1 Class Diagram

As I knew the requirements of the software was going to change over development, I made each build's class diagram after completing the preceding build. This meant each diagram included whatever new requirements had been added. I created one class diagram for each build, as well as a second for build two reflecting changes from the user feedback questionnaire (see Appendix I).

4.3.2 User Interface

To design the multiple HUD elements that would be displayed on screen, multiple sets of UI designs were created. The initial design was done as part of the design phase, with the second coming during development and the third after the user feedback questionnaire. These changes were made to improve the software and reflect the desires of players. The UI designs only featured the HUD as various menus were designed during implementation. Continued updating of requirements meant that designing the menu's ahead of time was not useful.

4.4 Implementation

4.4.1 Tools

4.4.1.1 Unity

Unity is a free and easy to use game engine which is perfect for beginners due to its massive online support. Along with a vast number of articles and guides, Unity also includes built in demo's and tutorials to teach users how to use the UI as well as a very large asset store that I could use for models required in my project. The main alternative to Unity is Unreal, however, Unreal's focus on art and visual design is not beneficial to my project, who's visual design is very simple. In addition, Unreal uses C++ which did not then meet my objective of teaching myself a new coding language. Unity however uses C# which does meet this objective.

4.4.1.2 Visual Studio

Unity comes with built in visual studio compatibility when writing C# scripts, providing code completion and simple changes to source files. Furthermore, I have experience using Visual studio which removes the learning required.

4.4.1.3 Blender

Blender is a free, 3d modelling software that was used to create several assets within the project. I chose this due to its ease of use. This was very important as I had no experience using any 3D modelling software. Due to blender's popularity, there are a great deal of online tutorials that I was able to use to teach myself to use the program, which meets my objective of learning to use new pieces of software.

4.4.1.4 Visual Paradigm

I used visual paradigm for the creation of design diagrams due to my previous experience using it during the Object Orientated Analysis and Design module as well as the licenses being supplied by the university.

4.4.1.5 Microsoft PowerPoint

PowerPoint was used to develop the 2D user interface assets primarily due to my experience using it. I had developed 2D assets for previous classes using their simple shapes and found this to be the quickest and easiest method of developing the required assets.

4.4.2 Version Control

I used GitHub as a method of version control throughout my project, primarily due to my extensive experience with it in other projects. Using GitHub is easy to understand and use, providing an efficient method of regularly saving work, affording me the ability to go back to previous versions if necessary. This is essential as my project has so many interconnected parts that one small change can have a dramatic effect, something that is easier to remedy with access to previous successful versions. GitHub pushes also allow me an easy way to timetable and track my changes to ensure I am keeping on time with my project plan.

4.4.3 Work Plan

My project has been divided into three main builds: Physics, User Data Interaction and Grading and Scientific Knowledge. The second build (User Data Interaction) saw the project in a minimum viable product stage. It allowed the user to run the experiment and analyse the results. After my second build was completed, the software was play tested by several colleagues, who were then issued with a questionnaire. This allowed me to take on user feedback and inspired several changes and additions, detailed in the relevant sections below.

The final build is based on improvements and additions that will add to the overall user experience. Each of these builds, as well as the tasks within them were all designed to be testable with predefined success conditions as "Breaking down projects into *truly* actionable pieces means

including a clear definition of done" (Georgieff, 2022). This means that I can ensure each iteration is complete before moving to another task, reducing errors made from overlooking problems early on. The specifics of each build were defined in the Gantt chart found in my PDD (Appendix A) and were kept consistent throughout the project, acting as an overall timeline.

4.4.4 Build 1 – Physics

4.4.4.1 Player Control

For the player to operate in the game world a character must be created with given movement functions. This involved creating a simple object to represent the player as well as adding a camera to it which can be controlled with the mouse. A class was added to control player movement with the WASD keys as this is the most common form of movement in videogames. Both the movement and camera controls were made to be the most intuitive to the player to improve the user experience.

4.4.4.2 3d Models

The essential models required in this build are the desks and practical equipment. These were either downloaded (in the case of the desks, ruler, or wires) or created in blender. The involved following online tutorials and producing practice models to teach myself how to use the system before creating the necessary models. Models were created to represent each state of the apparatus and were all loaded into unity.

4.4.4.3 Physics System

All 3d models were given physics functionality (using rigidbodies) and collision detection so the user can correctly interact with them as well as having the objects act realistically (such as falling due to gravity). Unity's RigidBody system allowed me to set the acceleration due to gravity, but after testing, the gravity value is not always perfect and falls in a range. This is perfect for the project as it provided the possibility for anomalous results without it acting as an extra feature that must be programmed. A class was then created that allows the user to pick up items when clicked, moving them in front of the player, and dropping when the player releases the mouse. This is inspired by the work of (Jimmy Vegas, 2018) that demonstrates this mechanic.

To build the apparatus, the user must carry each piece to the build area and drop the item. As the item is added, the existing apparatus was destroyed and replaced with the model that contains the new part. The light gate was given collision detection to record the time taken for objects to fall through it. This data is then output on the timer for the user to see.

4.4.4.4 Ruler and Dropping

When the user drops the weighted card onto the completed apparatus, a ruler canvas appears that allows the user to specify the high that the card will be dropped at. The card will then fall from the specified height and the collisions will be registered. The time taken to fall through the gate is then output on the timer.

4.4.5 Build 2 – User Data Interaction

4.4.5.1 Heads Up Display

The HUD contains an instruction area that holds the next task required of the user, a timer that starts when the game is begun and text that appears in the centre of the screen to display relevant object interactions. The software uses ray casts to determine what is being looked at and then changes the text accordingly. This display also includes a notes page that can be opened at will and used to document the timer values that are output. This was added in after suggestions from the user feedback questionnaires that asked for an easier way to remember the timer values between the experiment and writing up the table.

4.4.5.2 Table View

The table view is a canvas which is activated when the player interacts with the page object on the desk. The canvas contains two sliders, that control the number of columns and rows required. Whilst initially input fields, the controllers for the number of columns and rows were changed to sliders following user feedback. As the user changes the number of sliders, several gridlines are drawn to reflect the values, with input fields instantiated within the drawn boxes for the user to enter values in. The code to draw the gridlines uses the graphics class and was reused from a YouTube video (Game Dev Guide, 2020), which makes a grid with variable rows and columns. Research showed that interreacting directly with the graphics class was the most efficient way to do this. It also allowed for the most customisation as it is the same code that is reused to make the graph paper (see 4.4.5.3). As this was the only code, I was able to find with this mechanic, I decided to use it to solve this problem. Reusing this code was very important as it solved a complex problem that I had no prior experience with.

4.4.5.3 Graph View

Like the table view, the graph view opens when the player interacts with the page and uses the same grid code (with necessary edits) to create a graph paper that would move, and scale with mouse movement. This page also includes moveable input fields so the user can add titles, number grid lines and points that the user can use to plot the graph. Having input fields that drag and drop was inspired by the feedback from user questionnaires. Finally, the page allows the user to draw a line of best fit by creating a line renderer class using code from the same graph paper source (Game Dev Guide, 2020).

4.4.6 Build 3 – Grading and Scientific Knowledge

4.4.6.1 3D Environment

To promote the best learning environment, I created a 3d classroom using models from the Unity Asset store. This was important as studies show "(classroom) design accounts for around 50% of the impact on learning" (Barrett, Davies, Zhang and Barrett, 2015). The pack of assets used created a classroom design matching the principles of naturalisation that the above study references as the most important, as well as being a cost-effective option for my development.

4.4.6.2 Scientific Knowledge Pages

The scientific knowledge pages used are the info page (a page explaining the experiment and what needs to be done) and the equation page which educates the user on how to plot the graph. Both of this are canvas' that are activated by interacting with the pages on the table, however, the equation page acts as a mini game where the user must select the correct equation and drag and drop elements to rearrange into the correct form. The info page has the user input the independent and dependant variables for the experiment. These pages are essential to having the software act as an educational experience.

4.4.6.3 Grading System

The grading system records the users actions and awards marks. A mark scheme was created and when the user has finished the experiment, the system will mark each section. These include correctly labelled graph axis', correct variables planned, and correct value of g determined. All of these were taken from mark schemes of GCSE and A Level practical's. Sections that do not score full marks give feedback for how the user can improve.

4.4.6.4 Menu and Tutorial Screens

The main menu was created as a separate scene that allows the user to select the experiment before starting it. This is to meet the requirement of the software acting as a platform for multiple experiments to be added to. The system was coded so that the software could be seamlessly expanded. Within game, a menu canvas was created to allow the user to quit to the main menu and change any audio settings (see 4.4.6.4). Alongside this, a tutorial system was created that runs when

the experiment begins, explaining how the gameplay mechanics of the system work. Whilst I based the controls off common videogame controls to give the most seamless experience, those less versed in videogames could struggle. These tutorial explanation screens will prevent this problem. Finally, bouncing arrows were added to highlight the next object to interact with.

4.4.6.5 Audio

After extensive research, it was clear the classical music not only aids in the retention of new information (Raypole, 2022) but also reduces anxiety (Sridharan et al., 2007). Consequently, I have included classical background music (which can be toggled on or off), to improve the educational experience.

To increase realism, I have included several sound effects including the rustling of paper or clicks to simulate objects connecting. These are played when relevant events take place in game to create a fully immersive experience.

4.5 Testing

4.5.1 Unit Testing

To ensure there were no faults, the software was tested throughout with each class being tested once written. Given my agile methodology and continuous adaptations, writing prewritten tests would be a waste of time for such low-level problems.

4.5.2 Integration Testing

As I reached the end of the development, I ran the requirement tests that were developed in the design phase. These allowed me to ensure the requirements were met, going back to fix where tests had failed.

4.5.3 User Testing

4.5.3.1 User Feeback Questionnaire

To ensure the software was the best for the user, I completed a prototype of my project up until build two and then had the game play tested by ten users, following which they were given a questionnaire. The responses were used to make changes to the previous builds and to guide how the development of build three would continue.

4.5.3.2 Evaluation Questionnaire

Once the game was finished, it was play tested again with scores collected to allow me to evaluate the success of the project as an educational experience.

Chapter 5: Results

5.1 Analysis

5.1.1 Research

The Unity tutorial I followed led to the creation of a basic game and taught me the basics of Unity. Whilst there was still much more to learn, that I improved and developed upon during my implementation phase, the tutorial taught me sufficient basics on how to operate the Unity IDE and use the various required components.

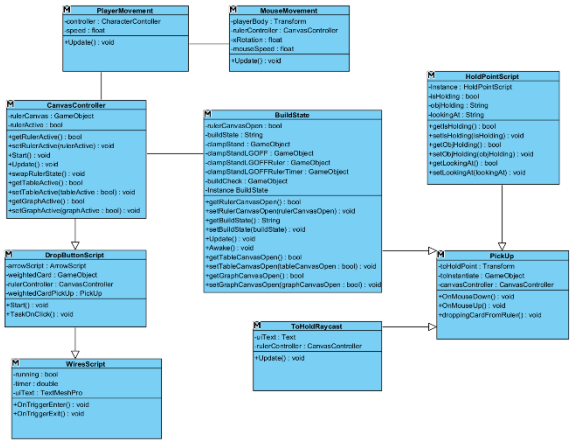
5.1.2 Requirements

I created 11 functional and 5 non-functional requirements to guide my implementation. Following this I created 13 test cases to test each of the requirements, ensuring they were met, to be run following the implementation stage.

5.2 Design

5.2.1 Class Diagram

4 class diagrams were created to map the various classes, variables, and connections between them. These were created before each stage to account for changing requirements. The class diagrams (see Appendix I) consist of: Build 1, Build 2, Build 2 after feedback, Build 3. These were made using visual paradigm.

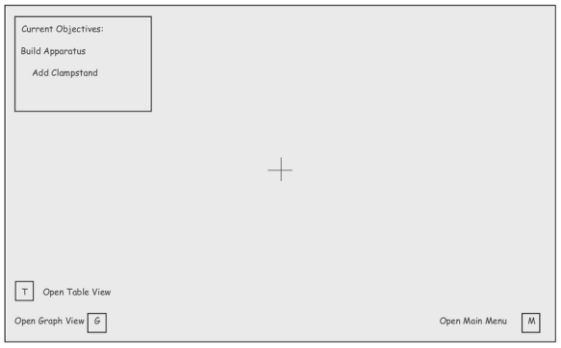


Build 1 Class Diagram

5.2.2 User Interface

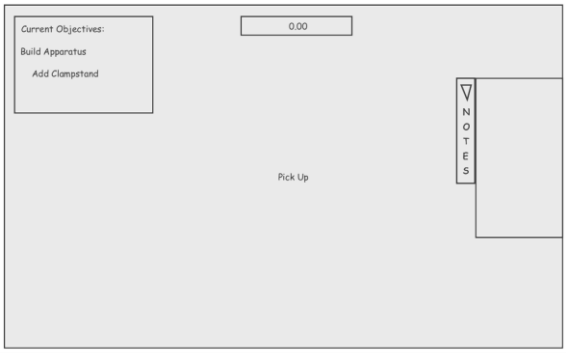
The difference between the first and last UI designs is substantial, with the first including cross hairs (which were removed after user feedback deemed them unrealistic) and buttons for opening various menus. These were replaced with books that the user can interact with to open the menus.

The latest UI design also includes a timer and notes page which were both added as features after the user feedback questionnaire. In all UI designs, the HUD features are at the edges of the screen, so they do not affect the game play.



Initial UI Design

In the game these assets use a blue and white colour scheme. This colour scheme was tested on users and received positive feedback. The motivation behind the colours chosen, was to "calm the mind and aid concentration" (Psychological Properties of Colours - Colour Affects, 2021). To ensure the software is accessible to all players, I tested the assets in the Coblis Simulator (Coblindor, n.d.). This is a website that allows you to view assets through the lens of different forms of colour blindness, allowing me to ensure everyone will be able to read and understand all elements of the software.



Final UI Design

5.3 Implementation

5.3.1 Build 1 – Physics

5.3.1.1 Player Control

To create the player, I created a simple cylinder with the renderer removed to prevent shadows appearing. To this player I childed a camera. As the player character is moved or rotated, the same

transformations will apply to any childed object, meaning I only must set controls for the player, with the camera moving accordingly.

To move my character, I added a character controller to the player. This is a built-in unity system for simplifying the movement of a character. "Its function is to move the player according to the environment" and "provides basic collider responses without any physics" (IronEqual, 2022). The benefit of this over a RigidBody (Unity's physics class) is that I can code the specific movement functions, ensuring that the motion is framerate dependant. I did this by factoring 'Time.deltaTime' into my speed calculation. Therefore, I chose to use 'Move' in place of 'SimpleMove' as the move function allows me the ability to account for framerate and program my own gravity. A second benefit to using this class is that I don't need to handle inputs, as by default the system will move the character when using WASD.

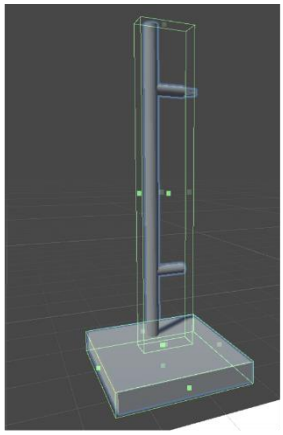
I then created a class to control the first-person camera angle. Like the player movement, this needed to be independent of frame rate to ensure the game plays the same on all devices, meeting the non-functional requirement and allowing for maximum accessibility. After testing I found the constant float of 100.f gave the best user experience. To ensure the realism is maintained, I then clamped the rotation meaning that the player cannot look beyond directly up or directly down, matching human motion. I then applied the rotation to the player and camera before locking the mouse cursor to the screen. This ensures the mouse is invisible as having a visible mouse in the screen will remove the realism. In addition to this, by locking the mouse, the cursor is always set to the centre of the screen, meaning I can still use the mouse to select objects that are placed in the screen centre.

5.3.1.2 3d Models

In the game, several core 3d models are required. The first I added were the tables, wires, and ruler, these were downloaded (see asset listings) and the required scaling and positioning was applied. However, as the remaining models I needed were extremely specific, I was unable to find relevant assets to download and use. Instead, I elected to produce my own assets following the research and practise into blender modelling that I completed in my research section. I then produced the individual pieces required for the experiment based off existing equipment I had used in the past. These were: The clamp stand, light gate, and timer. After research showed that attaching multiple 3d physics objects would be very challenging, I decided to create 3d models representing each apparatus state in turn. These were again completed in blender. Finally, these objects were all loaded into the games and saved as prefabs, unity's system for prebuilt game objects. This allowed me to save all their physical properties without needing to set them when making any game changes. All the models were scaled to look correct given the scale and height of the camera.

5.3.1.3 Physics System

In the game the objects need to be picked up, fall with gravity, and collide with one another. To add this, I used box colliders to act as bounding boxes around my objects. For more complex shapes I created stacked models which add "multiple bounding volumes as needed" (Child, 2021). Having collision detection for every vertex is very computationally expensive and could cause performance issues in game so these stacked models comprised of box colliders are far more efficient. Most of these colliders are non-trigger colliders, meaning they act as physics objects but don't call methods on collisions, thus improving the efficiency. The exception, however, is the collision along the light gate's laser. This is a set as a trigger collider so we can measure all collisions associated with it.



Box Colliders on Clamp Stand

In addition to the box colliders, many of the objects (namely the apparatus elements) require gravity to act on them. This was added using a RigidBody, unity's built-in physics system. This means that objects will naturally fall under the effect of gravity, as well as giving the ability for

me to vary physics attributes such as mass and drag. In this RigidBody system I locked the rotation for the objects that needed to be picked up to stop them from freely spinning when getting knocked by another object.

To create the ability to pick up objects, I added an empty game object called 'HoldPoint' that was childed in front of the player. I then created a class called 'PickUp' based on (Jimmy Vegas, 2018). This initial code deactivates the object's gravity and transforms it's position to the HoldPoint position when the mouse is pressed over it. When the mouse is released, the gravity is returned, and the object falls. The PickUp script was then be applied to all objects capable of being picked up. It uses unity's built-in input system that calls the function when the mouse is clicked over an object.

During the game, different functions must take place depending on what the user is holding. As such a script was created to record what item is being held. After research done in my literature review, I found that the best way to do this is to use a singleton. This is a class that exists in one single instance in the project and so can be globally accessed without defining the instance in every class that it is used in. To protect against multiple instances of the class, any new additions are destroyed instantly.

This class is called 'HoldPointScript' and contains getters and setters for the string variable 'objHolding'. In my pickup script I reference the HoldPointScript and set what item is being held, for the duration that the mouse button is down. I then created another class called 'ToHoldRaycast' which uses ray casting to determine, in every frame, what object the user is looking at. The object's name is then held in the singleton 'HoldPointScript' so that every class can access whatever the user is looking at in any given moment.

Finally, to build the apparatus, the user must be holding the next object in the build order (see 5.3.2.1). My initial plan was to have each of the apparatus stages saved as prefabs but not instantiated in the scene, however, as each prefab will have to have scripts attached and must reference game objects that would not exist at the time of instantiating them, this would not work. Instead, I loaded all apparatus stage game objects beforehand and moved their position to outside the classroom area.

To know that the user is looking at the build area, a cube object called 'BuildObjCheck' was created. The mesh filter was disabled, rendering it invisible to the player, however it was given collision properties, so it interacted with the raycast system. This means when the user is looking at it, 'HoldPointScript' records this. When the player is holding the next part required to build and releases the object when looking at the build area, the current apparatus and object being held are destroyed, and the next apparatus stage game object is transformed to the correct position, creating the illusion that the piece has attached onto the apparatus. For example, when the apparatus shows the clamp stand and light gate, and the player is holding the ruler, the clamp stand, light gate and ruler are all destroyed, replaced with the object 'Clampstand, lightgate ruler'. The apparatus objects do not have pick up scripts attached, and so cannot be picked up by the user accidentally. A script called 'BuildState' was created to keep track of the current state of the apparatus as well as the stages of the game. This record means the apparatus must be built in a specific order, preventing errors. As this class is referenced in so many places, it is a singleton to increase the efficiency and ease of use.

Apparatus stage of ClampStand,LGOFF with the player holding the ruler

Apparatus stage of ClampStand,LGOFF,Ruler

5.3.1.4 Ruler and Dropping

When the apparatus is fully built, the user can then pick up the card and carry it to the build area, releasing and dropping the card. When this happens, in the pickup class, the ruler canvas is opened, and the player movement is suspended. The camera is locked in place and the mouse appears on the screen. This canvas has an image of the ruler with an arrow that can be dragged along it, locked so it cannot move past the edges of the ruler. A text component outputs the current distance in cm where the arrow is placed along the ruler. The user can then click to drop at the specified height. When this happens, the ruler's position is transformed to the specified height along the ruler where it falls and accelerates due to gravity.

As the light gate has a collision trigger on the laser, I have used the functions OnTriggerEnter and OnTriggerExit to time, considering frame rate, the time taken for the card to fall through the gate. This time is then rounded and reduced to 3 decimal places, before being output on the timer. Initial prototypes had this value also output on the side of the screen in the HUD, but this was changed due to user feedback to keep the experience as realistic as possible.

5.3.2 Build 2 – User Data Interaction

5.3.2.1 Heads Up Display

The first part of the HUD to add was the text in the centre of the screen that describes the next action to take place. In initial prototypes of the game, crosshairs were used to highlight the centre of the screen. However, these were removed after questionnaire feedback showed that it took away from the realism of the experience.

To add this text, I furthered the 'ToHoldRaycast' class to give a different command depending on the name of the object being looked at. This meant that when the user hovers over an object that can be picked up, 'Pick Up' is displayed on the screen, consequently when the user is holding an object and looks at the build area, 'Build' will be displayed etc. As these ray casts collide based on physics objects, I used layers to prevent unnecessary calls, reducing efficiency issues. This meant taking all non-interactable objects and adding them to an 'Ignore Raycast Layer'.

A timer was created and placed at the top of the HUD to allow users to better organise their time. The timer is controlled by the class 'OverallTimerScript' that increments the timer every frame, setting to show minutes and seconds, and considering framerate so it works seamlessly on all device speeds.

The instructions section of the HUD is a box on the top left of the screen that contains text objects. References to these are held in the 'BuildState' class. Each stage of the game has a pre-set instruction

that is output on the screen when the user reaches the preceding stage, forcing the user to continue the game in the set order.

Following user feedback, I added a notes page that allows the user to record values from the timer before putting them in the table. This involved creating a notes canvas that was toggled on and off using the tab key. When toggled on, the notes page had an input field that allowed the user to enter any text they required.

5.3.2.2 Table View

The table view is a canvas that opens when the player interacts with the page object on the desk. It contains two sliders representing the number of columns and rows that the user would like to have, as well as a number above each that shows the value of the slider. These sliders are locked with maximums to ensure the number of rows and columns don't get too large. Initially I built this using both input fields and sliders to test the best mechanic, with user feedback questionnaires preferring the sliders.

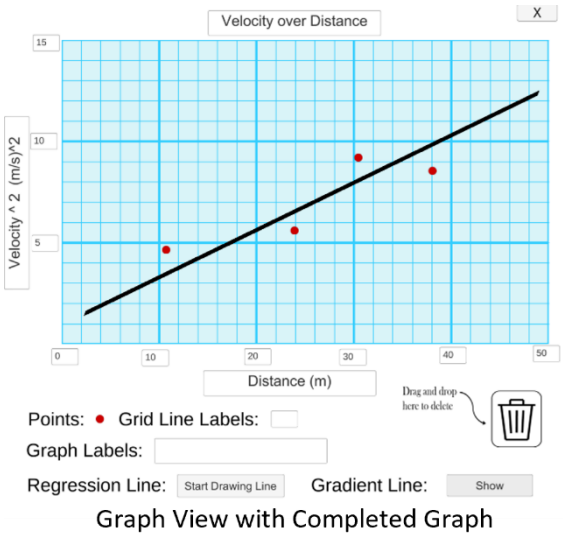
To create the grid lines for the table, I adapted code from (Game Dev Guide, 2020) into a class called 'GridScript'. This code interacts with the graphics class to build primitive triangles on the canvas. The method that I used is called 'createSquare' and, given a width and height, draws 8 triangles, 2 for each line, to create a square. Given the set size of the table, I then take the slider values for the number of rows and columns to calculate how many squares are needed and of what dimensions. These are then drawn with the 'createSquare' method and rendered on the screen using a canvas renderer. When the slider values are changed, the lines are deleted and redrawn with the new values. This produces a dynamically table that can be freely edited by the user.

To allow the user to write in the table, input fields needed to be instantiated to fit in all the boxes. To do this, within 'GridScript' I added a method that instantiates input fields into the coordinates of every square in the table, adjusting their size so they fit perfectly. I then keep a reference to each input field in a list. When the slider values are changed, I delete all the input fields in the list and empty the list, before instantiating the new input fields. This ensures that there are always perfectly positioned and sized input fields filling the table.

5.3.2.3 Graph View

The graph view is a canvas that opens when the player interacts with the page object on the desk. This page allows the user to create a graph of their results, label the graph, and use a line of best fit to analyse their data.

To create the graph paper, I used the same 'createSquare' function as the table view from (Game Dev Guide, 2020). However, I run this code twice, to create the larger grid squares, and then again to create the smaller squares within. I did this to replicate real graph paper which has multiple square sizes to help plot axis'. I plot the number of squares proportionally based on the width and height of the paper. To allow the user to customise the graph paper for their data, I used the mouse scroll wheel to zoom in and out of the grid. To do this, I increase and decrease the number of squares that are rendered as the mouse scroll data changes. This creates the illusion that the graph is zooming in and out, when the user is simply changing the number of rows and columns.



Graph View with Completed Graph

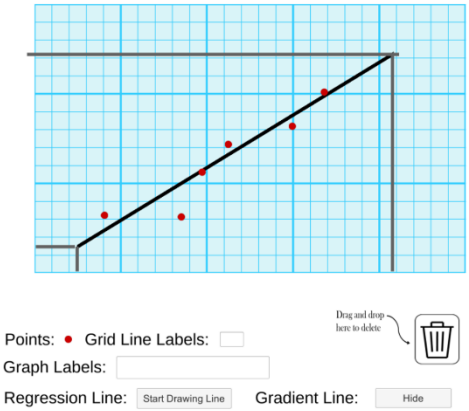
To create the graph points, I used a circle image along with a class that detects when the user presses and releases the mouse over the image. This allows me to move the image whilst the mouse is down, creating a drag and drop mechanic. When the mouse clicks the image at it's spawn point, a new circle image is instantiated with the same properties, so the user can place as many points as required.

As a required part of practical experiments, the user must label the graoh title, axis' and have a scale of labels on grid lines. To add this feature I have added two types of inputfields of different sizes, the graph labels and grid line labels. These both use the same drag and drop mechanic as the graph points as well as being instantiated in the same way when the user intially drags them. As the y axis label needs to be rotated and vertical, the class that moves it will detect when it passes the edge of the graph and will rotate the field accordingly. Following user feedback, the consensus found is that fields that drag and drop would provide a better learning experience, rather than fields that instantiate at the ends of grid lines.

As the user can create as many points or labels as required, there can get to the point when there are too many. To combat this, I have created a delete area modelled off the windows 'Trash' button for simplicity. When the user drops a button or label in this area, the system will delete the game object. This keeps points from cluttering the graph and the deletion ensures the system will not slow with an increasing number of gameobjects.

Unity contains a built in system for rendering lines called 'Line Renderer' however, these lines are 3d and not made to be used in two dimensions on a canvas. To create a 2d line renderer I adapted code from (Game Dev Guide, 2020). This code takes points and renders two triangles to create a line between them. To improve this code to work with my project, I took the mouse input when the user clicks to create the first point, and have the second point track the mouse position until the user releases the mouse, plotting the point. This allows the user to drag a line of best fit and rotate it before finalising it. When the user clicks again, the previous line is deleted and a new line begins drawing. Finally, so the user can switch between drawing the line and editing points, I added the button 'Start Drawing Line'. When activated the user can draw the line, after which locking the button will stop drawing so the user can continue to move the points.

To analsyse the gradient of the line, which is required in our calculations, the user needs to be able to determine the change in the x and y value of the line of best fit. To do this, I have added the gradient line button. This holds the positions of the start and end of the line of best fit, and when toggled on, draws lines from the points to each axis. To do this, rectangular plain images are saved which resize and reposition to ensure they draw lines between the line of best fit end points, and the axis. This allows the user to read the difference in values to use for the calculation. This button can also be toggled off so the user can make graph changes without the lines being in the way.



Graph View with Completed Graph and Gradient Lines to Calculate X and Y Differences

5.3.3 Build 3 – Grading and Scientific Knowledge

5.3.3.1 3D Environment

Using a pack of classroom assets, I created a 3d environment resembling a classroom as much as possible to facilitate learning (see 4.4.6.1). This was done in the Unity IDE, with each object given box colliders so the player would interact with them as if real.

5.3.3.2 Scientific Knowledge Pages

The scientific knowledge pages consist of the info, equation, and final pages. These are all canvas' that are opened when the user clicks the relevant page object on the desk. To create these, I used cubes that were resized to resemble paper, and textured with images of the information on the

canvas. When opened, the info page provides text and diagrams explaining the process of the experiment and contains two input fields for the user to enter the independent and dependant variables.

The equation page acts as a small minigame, and unlike the other two pages, does not have a close button until the page is complete. First, the user is displayed 5 possible equations and they must select the correct one. These are all created as buttons, that when pressed either activate a red cross or a green tick when the user selects them based on whether they are correct. Once the correct equation is selected, the user must rearrange the equation with a drag and drop game. Using a class called 'Equation Mover' the user can select and drag each component of the equation into boxes. The boxes are images that contain a locator to determine if the equation component is dropped within the box. If it is, the component is locked in place. Once the equation is completely rearranged, the close button appears along with text explaining how to graph your results.

The final page contains information for the user on how to analyse the graph, as well as two input fields for the user to enter their change in y and x values from the graph. When pressing the enter button, the program divides the numbers outputting the value of g that has been determined. The user then can leave and make changes or submit their work. When submitting their work, the text and fields that have been shown are disabled and the grading fields are activated. Pressing the submit button runs the grading system (see 5.3.3.2). All the marks are output on the screen by editing text elements. When the submit button is pressed the timer also freezes, with its value being shown on the final page for the user to see.

5.3.3.3 Grading System

The grading system is run when the user submits their value for g and measures the user's activity throughout the experiment. The sections that are marked, and how their mark is calculated is as follows:

Gravity Value – The correct value of g is 9.81 so to grade the user, the system takes the calculated value from the final page and awards 5 marks if the user calculated the value between 9 and 10. For every integer value difference between the correct and calculated answer, the mark awarded reduces by 1.

Independent and Dependent Variable – The variables are entered by the user in input fields in the info page. These are read by the grading system class and marks are divided based on the word entered. For the independent variable, if the user enters a variation (with the letter symbol or the word with beginning with an upper or lowercase letter) of distance, 5 marks are added, if the user enters height, 3 marks are given. For the dependent variable, if the user enters velocity, 5 marks are added, if speed is entered 3 marks are added. The higher marks are awarded for more specific words.

Graph Labels – The three graph labels that should be added to the graph are a title, y axis label and x axis label. In the grading system 2 marks are awarded for each of these. As each title label is created, a reference to it is stored in a list which is accessed by the grading system. The graph title label is found by comparing the y position with locators to see which label is situated above the graph. As the user has a lot of freedom in what to call the graph, the two marks are awarded if this is present and non-empty. The Y axis label is found by comparing positions with locators to see whether it is on the left of the graph, with marks awarded if it contains the word speed (1 mark) or velocity (2). The X axis label is found by checking which label is below the graph against locators with marks awarded if the label contains height (1 mark) or distance (2 marks). To account for different ways of phrasing this answer, the system checks for if the word begins with an uppercase or lowercase letter, or if single symbol letters (such as v for velocity) are used. Misspelled answers are not awarded marks as spelling is an important part of preparing these materials.

Grid Line Labels – As the graph axis must be labelled with a consistent scale, it is important to check this as part of the grading system. As each grid line label is instantiated, a reference to it is held in a list. This is done in the grid line label class. This list is accessed in the grading system class so there is a reference to every grid line label on the graph. First, we remove any empty labels from the list. Accessing the X and Y positions of the labels, we can then check whether each label is on the left or right side of the y axis, indicating which axis the labels are for. The labels are then added to two different lists based on the axis they are on. The system will then count how many labels are on axis and will award 1 mark per axis where at least 3 non-empty labels exist. This means the user is given marks for having some labels on each axis. To then mark the consistency of the scale, the labels are ordered by their values, giving each label in an increasing order. Using a loop, we can then check the current value against the next value, recording the difference, and checking if there is a consistent linear or exponential increase in the scale. If this is consistent a further 2 marks are awarded per axis. Therefore, if both axis' are scaled correctly with at least 3 labels each, the user will then be awarded 6 marks.

Table Units – When making the table, the first row that contains the table headers must contain correct units for the data in the column they replicate. The table that the user makes should follow the guide of the table shown in the info page. This contains six columns. When the table instantiates input fields, if they are in the first row, they are added to a list. The grading system then checks if the input field's text contains the necessary units such as (s) or (m/s). For every correct title, the user is awarded a mark, up to a total of 6.

No Blanks in Table – This marking point is to ensure the user has filled each field in the table. Initially 5 marks are available to the user and for every blank field, this mark is reduced by 1 until no marks are awarded. To check this, when each input field is instantiated, a reference to it is added to a list that is read by the grading system. The system then checks if the input fields are empty, removing the marks if they are.

Line of Best Fit Gradient – This marks the user on the accuracy of their line of best fit compared to the plots on the graph. Each plot is saved into a list when instantiated, which is accessed by the grading class. The grading class removes any deleted plots and any plots outside the graph paper. The plot's coordinates are then analysed using the least squares method, summing values fed into the equation shown. This calculates the optimum gradient for the line of best fit, which is then compared to the gradient of the actual line of best fit drawn by the user. Marks are awarded based on the size of the discrepancy between these values.

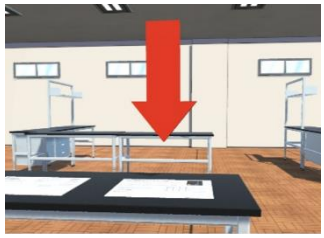
$$m = \frac{\sum_{i=1}^{n}(x_i-\overline{X})(y_i-\overline{Y})}{\sum_{i=1}^{n}(x_i-\overline{X})^2}$$

(Line of Best Fit (Least Square Method), 2022)

Once the system has awarded marks, text objects are used to print out the marks for each section to the player, along with an image showing a letter grade from A – E based on the total mark. For each of the marking points there is a feedback sentence that describes what the user must do to achieve full marks. For each section the user did not receive full marks in, the relevant sentence is shown on the screen as a list of points for improvement.

5.3.3.4 Menu and Tutorial Screens

To create the tutorial system, I created a canvas called welcome canvas that uses a series of images and text to describe the various gameplay controls from movement to the interaction with objects. This tutorial is divided into 5 different steps that are activated in turn as the user moves through them by clicking a forward and backwards button. A skip button is also present on the screen so that experienced players can skip to the end of the tutorial.



Bouncing Arrow

To make the process of the game clearer to the player, I have added large 3D arrows that bounce above the next area that the user must go to. This matches the instructions shown on the top left of the HUD. These are initially set to inactive and are

activated when the relevant stage is reached. To bounce the arrows, the Y position of the arrow GameObject is increased and decreased between a min and max value.

In game I created a menu that is activated when the user presses the 'ESC' key. This was created on a canvas called 'Menu Canvas' that includes buttons for each of the functions controlled through the class 'CloseButtonController'. These functions are: Continue, which closes the menu and unfreezes the game, Tutorial, which replays the tutorial process, Settings, which opens a settings page allowing the user to toggle audio on and off, and finally Quit Game, which closes the experiment and returns the game to the main menu.

To meet the objective of having my software act as platform upon which multiple experiments can be added, I needed a menu that the user can us to navigate between different experiments. To do this I created a new scene called 'Main_Menu_Scene'. This is a whole new 3D environment with no connection to the existing 'Determination_of_g_Scene'. On a canvas, I added multiple buttons along the screen to represent each practical, with each showing unavailable as the practical has yet to be added. The available practical is determination of g, initially showing as greyed out, with an image that becomes coloured when selected. There are then two buttons that allow the player to quit the game or play the selected practical. When the practical is selected a text box is shown that details the practical and the skills it teaches. To add new experiments, all that is needed is to change the image to show the new practical and add the line of code to the button that runs the new scene, this means the software is seamlessly able to add new experiments and act as a larger educational platform. To design the menu, I used the same 3d environment as the practical, however with a greater variety of science models on the desks (such as beakers and circuits) representing the other possible experiments. The camera then moves back and forth along the room showing the game title and science models. This is done in a class that adds and takes away from the x transform of the camera between a minimum and maximum position.

Main Menu Scene

5.3.3.5 Audio

To add audio in Unity, we require an Audio Source and an Audio Listener. An audio listener is a class attached to the camera that acts as a microphone for audio events in the game, with an audio source acting as the speaker which can be added to Game Objects to have sound appear to come from different locations. However, due to the small-scale nature of the game, with each of the audio events occurring when the player is directly in front of the object making the sound, I instead created game objects with the audio source attached and childed them to the player. Using audio event tracks found online, I created all the required audio sources, these are: The whoosh of the card falling, the click of the apparatus being build, the rustle of the paper being picked up and the background piano audio. The piano music was used based on research that shows classical music helps students study.

A class called Audio Controller was then created that holds references to all the audio sources. I created functions to play and pause each of the tracks. In the class pick up, I added a reference to audio controller. When each stage of the apparatus is changed, the click track is played. When the card is released and falls, the whoosh is played and when each page is opened, the paper rustle is played. The piano track automatically plays, but both the music and special events tracks can be prevented in the settings menu (see 5.3.3.3).

5.4 Testing

5.4.1 Unit Testing

5.4.2 Integration Testing

As part of the design phase I wrote test cases for each of the functional and non-functional requirements. Due to the agile nature of my project, I intended the requirements to change over the life of the project. This meant that the test cases were adapted to fit the changing requirements. At the end of the project there were 13 test cases to be run that checked all the testable requirements (Appendix H). All these test cases passed successfully except for test 13 which tested the system on multiple screen sizes. Whilst unit testing had shown no issues running the game within the Unity IDE, when running an executable file on a new screen size, the coordinate functions I had been using for some of the games functions (such as the bin in the graph page) did not work. This is because the coordinates are calculated using pixel numbers so the size of the screen affects the world coordinate system. This led to my addition of locators; small blank images with the render removed so they are invisible to the user. Instead of using a global coordinate system, these locators are used to mark positions in space. Distance to these can then be calculated to detmine if functions are run. The places that locators are used are described in the implentation section of this report (5.3). Following this change, all 13 test cases were successfully passed proving the requirements had been met.

Test ID:	**11**
Requirement to Test:	Functional Requirement 11
Test Case Description:	Grading system successfully grades player
Test Steps:	1. Enter 'height' as independent variable 2. Enter 'Velocity' as dependent variable 3. Enter a change in y of 14 and a change in x of 2 4. Add a graph label above the graph with the text 'Velocity and Distance', add a graph label on the left of the graph with the text '(V)' and add a graph label underneath the graph with no text 5. Add two grid line labels on the left of the graph with the text '5' and '10' in ascending order. Add three grid line labels underneath the graph with the text '10', '20', '30' in ascending order 6. Create a table with three columns. In the first field in column one add the text '(s)', in the first field of column two add the text '(s)' and in the first field of column three add the text '(m/s)' 7. Create a table with 4 rows and 4 columns. Enter a random string of text in every box
Expected Outcome:	1. Gets 3/5 for independent variable 2. Gets 5/5 for dependent variable 3. Gets 3/5 for gravity value 4. Gets 4/6 for graph 5. Gets 3/6 for grid line labels 6. Gets 3/6 for table units 7. Gets 5/5 for no blanks in table Player receives mark of E
Pass/Fail:	Pass

Test Case for Functional Requirement 11

5.4.3 User Testing

5.4.3.1 User Feedback Questionnaire

The questionnaire I created was designed to improve the existing build one and two as well as provide guidance on what to include in build three. To provide a comparison, in some cases multiple options were created so the user could pick the best, after which the others were deleted. For example, the table view was made with both input fields and sliders, with the input fields removed after user testing.

Questionnaire

Question 1: Playing the game both with and without crosshairs, which provides a better experience of aiming and realism?

Question 2: Is the text that appears when hovering over an object clear and easy to read? How could this be improved e.g., size and colour.

Question 3: Is the UI colour easy to read? Is this a pleasing colour? What other colours could provide a better experience?

Question 4: The game includes a timer that begins when the game is started, should this be shown to the user? If so, how?

Question 5: When dropping the card through the light gate, the value for the time taken for the card to fall is output on the screen (HUD) as well as shown on the light gate, is this a useful feature?

Question 6: When adding and removing rows to the table view, what is the best way to achieve this? Playing the game with input fields and sliders, is one of these a better mechanic? Is there another suggested method?

Question 7: When using the graph view, what would be a better mechanic for creating the squares on graph paper, a static image, moveable with sliders or scrollable with scroll wheel?

Question 8: When using the graph view, what would provide a better experience for labelling axis? A drag and drop input field mechanic or input fields that generate with each graph line?

Question 9: Is the game enjoyable?

Question 10: Are there any other comments or suggestions you would like to make that could improve the game?

User Feedback Questionnaire Questions

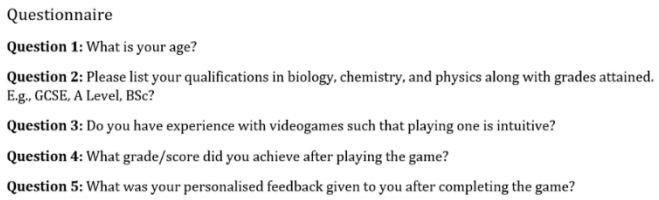
The changes made based on the questionnaire results are:

- The crosshairs were removed to increase realism
- The pickup text was made larger and bolder
- The timer was placed at the top of the screen in constant view
- The timer values are not output on the screen, only on the timer object
- The table view uses sliders not input fields
- In the graph view, the paper is made zoomable with the mouse scroll wheel
- In the graph view, the axis' are labelled using a drag and drop mechanic

- A notes page was added so the user doesn't have to remember the value before opening the table view

5.4.3.2 Evaluation Questionnaire

This questionnaire involved 10 users playing the game and giving the mark and feedback received. It also collected their previous experience to determine how much they had learned.


Questionnaire

Question 1: What is your age?

Question 2: Please list your qualifications in biology, chemistry, and physics along with grades attained. E.g., GCSE, A Level, BSc?

Question 3: Do you have experience with videogames such that playing one is intuitive?

Question 4: What grade/score did you achieve after playing the game?

Question 5: What was your personalised feedback given to you after completing the game?

Evaluation Questionnaire Questions

Examining the results, we can see that no grade was lower than a 'B' with six of the ten results being an A. However, the previous experience of players was very ranged including multiple levels and grades of qualification. This proves that the system was capable of successfully educating players, meeting objective one of the projects.

Chapter 6: Conclusions and Discussion

6.1 Objectives

6.1.1 This project shall educate users on the scientific method and how to conduct an experiment

The success of this objective can be best analysed through my evaluation questionnaire (see 5.4.3.2) which saw ten participants play the game and achieve very good scores (with no grade being lower than a B) despite coming from a large variety of scientific backgrounds. This demonstrates not only the effectiveness of my software as an educational experience, but that virtual science experiments are a very effective way of educating students, adding to the work of studies that demonstrate the effectiveness of virtual experiments over in person laboratories (Heradio et al., 2016).

6.1.2 This project shall recreate a laboratory virtually with objects that are used realistically

The virtual environment created definitely satisfies this objective, with all objects in the game having physics functionality, behaving realistically, and all designs based on what would facilitate learning in students (see 4.4.6.1). In combination with the success of the above objective (6.1.1), this project serves to act as greater evidence for the conclusions drawn from sources analysed in my literature review, showing the benefits full 3D environments have on learning over 2D games (Liu et al., 2015). This should promote the benefit of fully virtual, first-person experiments to facilitate learning and can therefore act as a point for further research in future projects.

6.1.3 The project shall create a table and graph screen that allow the user to store and analyse results

The table and graph screens created within the project function very successfully, enabling the user to create any table or scatter plot required. The graph functionality fits this objective perfectly, with the only downside to the table view being that when the number of columns or rows are edited, the input fields clear. Whilst the solution was planned to fix this, recording the values, and then refilling the input fields in the newly calculated locations, the timeline of the project did not allow this to be completed. However, as this does not affect the useability of the table, I consider this objective a success.

6.1.4 The project should provide a grade and feedback to the user to foster improvement

The software is successfully able to grade the user on metrics typically used to evaluate performance in practical experiments. Whilst the grading system covered most of these points, there was a limitation in that the system was designed to give the user as much freedom as possible which restricted my grading abilities. I had wanted to grade the user on the removal of anomalous values in the table, however, as the user could design the table anyway, they wished, it would be very difficult to have the system figure out what values related to what variables before deciding if they were anomalous. If the user filled in a pre-designed table, this would not be difficult, however, doing so would take away from a large portion of the learning experience of designing the table and all its components. This introduces an interesting area of future research as different AI models could be used to analyse student performance, giving interesting insights on learning, and providing a very personalised experience. This is something that my project can act as a point of research for. Studies that I researched (Heradio et al., 2016) spoke of accessibility benefits and immediate feedback, but lacked research into the personalised nature of virtual experiments that my software demonstrates. The combination of my software and AI models could further this area of research massively.

6.1.5 The project shall educate me on the use of new software, specifically through Unity and C#

The extent to which this objective was met is clearly demonstrated in the amount of time each section of the game took to develop. Before beginning this project, I had no Unity experience whatsoever, and upon starting my implementation only had the little skills developed through my research. As such, the longest build to develop was the first build, not due to its complexity, but due

to the time taken for me to develop the system whilst learning it myself. This greatly changed over the project with each subsequent build being quicker to develop (despite increasing difficulty) as I became more comfortable with the Unity development process. After completing the project, I feel very confident that I could create future Unity projects with ease.

6.2 Project Management

The agile method followed in my development was integral to my success. Over the course of the project, my requirements changed regularly, whether by choice or by user feedback. Due to my lack of experience with Unity, designing the system entirely beforehand was not an option, with weekly sprints allowing me the freedom to try new ideas on isolated functions without applying them to the larger system.

This project was hugely beneficial to me and developed my skills massively. Given my now experience with Unity, I would be able to recreate this software much more quickly and efficiently. This is very important as I plan to continue to develop experiments to add to my software, hosting it online so that users can develop their education.

Most of the challenges faced came from my lack of Unity experience, with some functions being delayed extensively whilst I researched ways to solve the problem. This however was primarily an issue with build one and two as over the course of the development I became comfortable enough with the system to develop the more complex functions without issue.

6.3 Reuse

In my project, I reused Code from two sources, adapting their methods to suit my requirements. Their use is documented in my report. They were essential to the completion of my software as they allowed me to complete complex functions that I had no prior experience in. Something that would have greatly delayed by timeline should I have to research and solve the problem alone.

6.4 Conclusion

In conclusion, I believe this project to be very successful, shown by the successful evaluation questionnaires and the completion of all planned objectives, including my personal development. This has led to the creation of a portfolio piece that I can use to showcase my abilities as well as a software that I can continue to develop and improve.

Chapter 7: References

Barrett, P., Davies, F., Zhang, Y. and Barrett, L., 2015. The impact of classroom design on pupils' learning: Final results of a holistic, multi-level analysis. *Building and Environment*, [online] 89, pp.118-133. Available at: <https://www.sciencedirect.com/science/article/pii/S0360132315000700?via%3Dihub> [Accessed 9 April 2022].

Blender Guru (2019). Blender 2.8 Beginner Tutorial - Part 1. Available at: https://www.youtube.com/watch?v=TPrnSACiTJ4 [Accessed 20th January 2022]

Burns, J., 2022. *School science: Lack of kit 'means pupils miss out'*. [online] BBC News. Available at: <https://www.bbc.co.uk/news/education-22369573> [Accessed 15 February 2022].

Butterfield-Addison, P., Manning, J. and Nugent, T., 2019. *Unity game development cookbook*. United States: O'Reilly Media Inc.

Butterfield-Addison, P., Manning, J. and Nugent, T., 2019. *Unity game development cookbook*. United States: O'Reilly Media Inc, pp.35 - 38.

Child, C. (2021). Session 7: Collision Detection and Introduction to AI

Coblindor, n.d. Color Blindness Simulator [online resource]. Available at: https://www.color-blindness.com/coblis-color-blindness-simulator/ [Accessed 12 February 2022]

Colour-affects.co.uk. 2021. Psychological Properties Of Colours - Colour Affects. [online] Available at: <http://www.colour-affects.co.uk/psychological-properties-of-colours> [Accessed 11 March 2022].

Dare Looks (2018). Inch by Inch [Computer Game]. Available at https://store.steampowered.com/app/992120/Inch_by_Inch/ [Downloaded 18th December 2021]

Dealessandri, M., 2022. *What is the best game engine: is Unity right for you?*. [online] GamesIndustry.biz. Available at: <https://www.gamesindustry.biz/articles/2020-01-16-what-is-the-best-game-engine-is-unity-the-right-game-engine-for-you> [Accessed 15 February 2022].

Eldad, A., 2022. *Unity vs Unreal, What Kind of Game Dev Are You? - Incredibuild*. [online] Incredibuild. Available at: <https://www.incredibuild.com/blog/unity-vs-unreal-what-kind-of-game-dev-are-you> [Accessed 14 February 2022].

freeCodeCamp.org (2021). freeCodeCamp.org: Everything You Need to Know About Singletons in Unity. Available at: https://www.youtube.com/watch?v=gB1F9G0JXOo [Accessed 30th January 2022]

Game Dev Guide (2020). Creating a UI Line Renderer in Unity. Available at: https://www.youtube.com/watch?v=--LB7URk60A&t=221s [Accessed 15rd March 2022]

Georgieff, M., 2022. *Breaking Down Large Projects Into Tasks in an Agile Process*. [online] Agilesherpas.com. Available at: <https://www.agilesherpas.com/blog/agile-work-breakdown> [Accessed 4 March 2022].

Glover, J., 2017. *Learn and Understand Raycasting in Unity3D*. [online] GameDev Academy. Available at: <https://gamedevacademy.org/learn-and-understand-raycasting-in-unity3d/> [Accessed 17 February 2022].

Heradio, R., de la Torre, L., Galan, D., Cabrerizo, F., Herrera-Viedma, E. and Dormido, S., 2016. Virtual and remote labs in education: A bibliometric analysis. *Computers & Education*, [online] 98, pp.14-

38. Available at: <https://www.sciencedirect.com/science/article/pii/S0360131516300677> [Accessed 15 February 2022].

Hoory, L. and Bottoroff, C., 2022. *Agile vs. Waterfall: Which Project Management Methodology Should I Use?*. [online] Forbes.com. Available at: <https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/#:~:text=Flexible.,with%20a%20defined%20end%20goal.> [Accessed 3 March 2022].

Infallible Code (2020). Infallible Code: Learn Unity – Beginner's Game Development Tutorial. Available at: https://www.youtube.com/watch?v=mpM0C6quQjs [Accessed 14th February 2022]

Innova Design Group. 2022. *How important is Practical Science in the Classroom? - Innova Design Group*. [online] Available at: <https://www.innovadesigngroup.co.uk/news/practical-science-in-the-classroom/#:~:text=In%20conclusion,broad%20understanding%20of%20scientific%20concepts> [Accessed 15 February 2022].

IronEqual, 2022. *Unity: CHARACTER CONTROLLER vs RIGIDBODY*. [online] Medium. Available at: <https://medium.com/ironequal/unity-character-controller-vs-rigidbody-a1e243591483> [Accessed 11 April 2022].

Jimmy Vegas (2018). How To Pick Up And Place / Throw Objects With C# - Mini Unity Tutorial – Easy & Fun. Available at: https://www.youtube.com/watch?v=IEV64CLZra8&t=484s [Accessed 3rd February 2022]

Kumar, G. and Bhatia, P., 2022. Impact of Agile Methodology on Software Development Process. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, [online] 2(4), pp.3-4. Available at: <https://www.researchgate.net/profile/Gaurav-Kumar-175/publication/255707851_Impact_of_Agile_Methodology_on_Software_Development_Process/links/00b49520489442e12d000000/Impact-of-Agile-Methodology-on-Software-Development-Process.pdf> [Accessed 4 March 2022].

Kushwah, D., 2022. *Advantages Of State Design Pattern With Example | Complete Guide 2022*. [online] Beetechnical. Available at: <https://beetechnical.com/tech-tutorial/state-design-patterns-when-to-use-pros-and-cons/> [Accessed 14 February 2022].

Liu, D., Valdiviezo-Díaz, P., Riofrio, G., Sun, Y. and Barba, R., 2015. Integration of Virtual Labs into Science E-learning. *Procedia Computer Science*, [online] 75, pp.95-102. Available at: <https://www.sciencedirect.com/science/article/pii/S1877050915036856> [Accessed 15 February 2022].

Raypole, C., 2022. *Here's How Music Can Help You Concentrate*. [online] Healthline. Available at: <https://www.healthline.com/health/does-music-help-you-study#negative-effects> [Accessed 2 May 2022].

ReQtest, 2012. *Functional vs Non-Functional Requirements - Understand the Difference*. [online] ReQtest. Available at: <https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/> [Accessed 12 March 2022].

Sourcemaking.com. 2022. *Design Patterns and Refactoring*. [online] Available at: <https://sourcemaking.com/design_patterns/state> [Accessed 15 February 2022].

Sridharan, D., Levitin, D., Chafe, C., Berger, J. and Menon, V., 2007. Neural Dynamics of Event Segmentation in Music: Converging Evidence for Dissociable Ventral and Dorsal Networks. *Neuron*, [online] 55(3), pp.521-532. Available at:

<https://www.sciencedirect.com/science/article/pii/S0896627307005004> [Accessed 2 May 2022].

Unity Learn. 2022. *Working with the Layer Editor - Unity Learn*. [online] Available at: <https://learn.unity.com/tutorial/working-with-the-layer-editor#> [Accessed 16 February 2022].

Varsitytutors.com. 2022. *Line of Best Fit (Least Square Method)*. [online] Available at: <https://www.varsitytutors.com/hotmath/hotmath_help/topics/line-of-best-fit> [Accessed 10 May 2022].

Chapter 8: Glossary

Crosshairs – A cross in the centre of the screen used for aiming

Rigidbodies - Components added to game objects that allow for the addition of physics functionality

Box Colliders – Boxes that mark the edges of game objects for use in collision detection

Input fields – UI Boxes that allow the user to enter a string of text

Ray casting – The process of firing a ray from the camera along the direction of the camera view to record any collisions

Layers – A method of categorising game objects, limiting functions to only specific sets of objects

HUD – Information presented to the user on the screen in fixed positions, independent of camera movement

Singleton – A design pattern creating a globally accessible class of which only a single instance can exist

GameObject – A class of an object within unity. Initially empty and can have any component attached, including scripts, 3d models and 2d assets

Transform – The component of a GameObject containing the position and rotation in world space

Prefab – A GameObject with components and alterations made so that it can be easily instantiated

Scene – An environment containing all GameObjects

Appendix A, Project Definition Document and Ethics Form

Course: BSc (Hons) Computer Science with Games Technology

Title: Virtual Science Experiment using Unity

Name: Oliver Reekie

Email: oliver.reekie@city.ac.uk

Consultant: Jacob Howe

Consultant Email: J.M.Howe@city.ac.uk

Proposed by: Oliver Reekie

Project Description:

The goal of this project is to recreate the investigative experience of a science practical virtually, making the experience more accessible to potential users. This project will recreate the experiment 'Determination of g from free fall', educating the user on the scientific knowledge they are testing as well as having them design, conduct and analyse the results from their investigation, facilitating knowledge of the scientific method. Following the experiment, the user will be graded and given feedback on their performance, allowing them to repeat and tune their skills. The software developed in this project will also act as a platform for more experiments to be added to.

Proprietary Interests:

This project will be created using the Personal Edition of the Unity game engine in accordance with Unity's licensing policies. I will also being using externally created assets which will all be credited with their individual copyright requirements.

Word Count: 1358

Contents

Problem to be solved

Practical science activities are essential to learning and understanding the process of scientific investigation and concepts (How important is Practical Science in the Classroom? - Innova Design Group, 2022). Therefore, bringing the ability for those studying science to engage in experimentation is an invaluable experience that many miss out on. Without access to science facilities a great many people are unable to take part in such activities, this is what my project attempts to solve. I will be using a game engine to replicate not just a science practical in a realistic virtual environment, but to recreate the entire investigative experience in a more accessible form.

This virtual laboratory experience allows users to interact with experiments in a safe and cost-effective manner, providing specialist equipment (7 Benefits of Using Virtual Labs in K-12 Education, 2022) that can be studied in the comfort of one's own home. This allows the user to work at their own pace, repeating sections that were difficult to understand and relieving the pressure of an in-person experience (Staff, 2022). These benefits will ensure that the material is learnt in an effective way, providing the best experience to those developing their scientific understanding.

Virtual science laboratories as a method of learning do exist with the most prominent being Labster (http://www.labster.com/). However, most of these platforms are not built to be realistic and are more simple interpretations made to demonstrate concepts, not recreate a genuine experiment (https://phet.colorado.edu/en/simulations/browse). The platforms that do create these experiences such as Labster are run on subscription models and are aimed at use with a VR or AR headset. Whilst I considered producing my project in VR as it would increase the realism of the experience, this would also massively reduce the accessibility of this software, which contradicts one of the primary reasons for my application, to bring practical experiments to everyone.

As this project's primary goal is to develop the user's understanding of the scientific method, my software will follow the main stages that it espouses. This will begin with a question that is to be investigated with supporting background research to explain the science behind it and a stating of an initial hypothesis. We will then design the experiment in conjunction with the user, having them assemble the equipment in a 3-dimensional, first-person environment. The user will then perform the experiment, collecting data and then analysing it by graphing the data. Finally, the software will grade the user on their performance and analysis with targeted feedback. This grading, combined with the infinite replayability, means the user can continuously learn and improve their experimentation skills, something that would be impossible in an in-person setting. My software therefore follows the main stages of the scientific method, which is one of the primary subjects that my software is made to educate the user in (ProjectPlace, 2022).

The second goal of the software is to educate the user on the scientific concepts that the experiment directly relates to. For this project I will be re-creating the experiment "Determination of g from free fall". This is a staple in practical experimentation as it does not require complex scientific understanding to comprehend, whilst still requiring graphing and the rearrangement of equations to fully complete. This experiment also works with gravity which, unlike many other experiments, is relatable and demonstrates a concept that we experience constantly. This relatability is extremely beneficial in facilitating learning (Grafwallner, 2022).

The mechanics of my project will be inspired by games such as ‘Inch by Inch’ (https://store.steampowered.com/app/992120/Inch_by_Inch/). This is a first-person game where the player completes science experiments by carrying items in front of them and placing them in set machines, whilst also acting as bodies that can be placed wherever the player wants. These are mechanics that will be used in my software.

Project Objectives

This project shall use physics to create a virtual simulation of a science practical. This will involve the creation of a 3d environment with 3d models that use physics functionality that allow the user to accurately complete the experiment.

This project shall allow users to store and analyse data using separate screens that can be used to correctly measure the acceleration due to gravity.

The project shall educate users on the scientific method, principles and how to correctly organise and analyse their data.

The project shall grade the users on their ability to record and analyse the results of their work. The software shall also give feedback to allow them to improve.

Project Beneficiaries

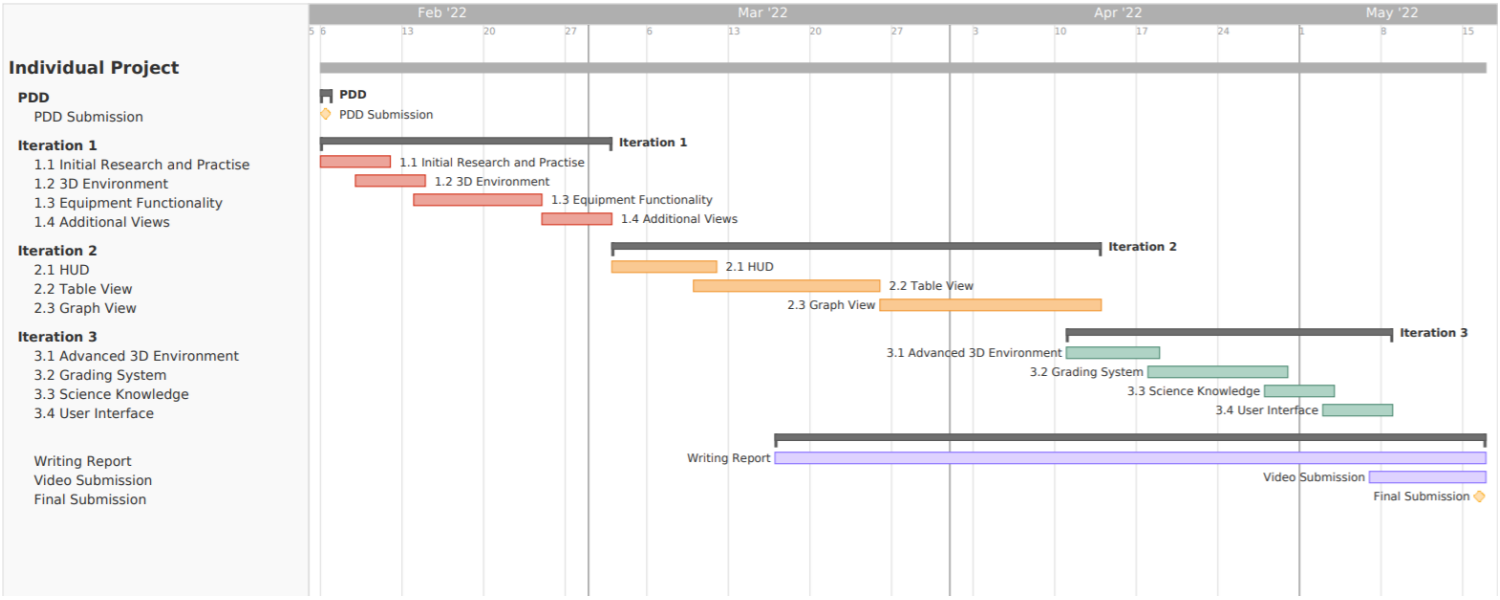
The primary beneficiary for this project is those attempting to learn more about the scientific method and to develop practical ability. This will therefore be useful for anyone learning or using this as a revision tool to supplement their learning.

A secondary beneficiary will be myself as this project will develop my skills and knowledge of C# and the Unity engine. This project will also be used as a portfolio piece, demonstrating my skills for future employers.

The third beneficiary will be societal benefits that come from a greater understanding of science. As science is the driving force for advancement my project, which not only teaches one experiment but acts as a platform to continually improve the use of the scientific method, will create a society that is more capable of creating the processes required to improve the world around us.

Finally, as I plan to create the interface that starts the practical experiment, the software will be able to be built upon by software developers, acting as a base platform for multiple other virtual experiments to be added to.

Work Plan



Project Risks

Objective	Probability /5	Severity /5	Risk	Prevention	Mitigation
1.1 Initial Research	1	2	It takes me longer to understand C# and unity concepts that prevents starting my development	Practise C# and Unity with online courses before the PDD approval to reduce learning time during project	Allocate time in the initial week to work on learning so it doesn't push back my timetable
1.2 3d Environment	2	2	Difficulty in finding and creating assets	Find assets before the project begins	Use blender tutorials to practise making my own assets to compensate for the lack of available ones
1.3 Equipment Functionality	3	4	Issues with physics and light gate detection	Use the same physics system as 1.2 so I should have good experience when creating it	Seek advise from project team to help understand physics issues
1.4 Additional Views	4	5	Difficulty in creating new views	Use Unity courses and tutorials to prepare	Continue using Unity courses to create the views
2.1 HUD	2	4	Unable to correlate HUD information	Use materials from Advanced Games Tech to help add HUD display elements	Discuss with project team how best to implement HUD
2.1 Table View	4	5	I am unable to have user created tables work properly	I will set aside lots of time for this section to allow for lots of practise and testing	Discuss with supervisor how best to store ad work with user input

2.2 Graph View	5	5	Issues with the user graphing and averaging the results	Conduct lots of prior research into similar applications to take ideas for my own. Dedicate a lot of time to this section	Test multiple types of graphing ability and use user feedback to test simpler applications
3.1 Advanced 3d Environment	2	1	Unable to find pre-made assets	Research assets before this stage and practise blender	Use blender to make my own assets
3.2 Grading System	4	3	Difficulty in grading different areas of work	Discuss with supervisor to produce models for comparison	This is a key objective however can be simplified if time is a limitation to remove detailed feedback
3.3 Science Knowledge	1	4	I am unable to complete this due to time limitations	Consolidate required research materials beforehand to reduce research time	Simplify to the core requirements and prioritise equation solving
3.4 User Interface	3	2	I am unable to complete this interface with time limitations	The research conducted in 1.4 should prevent issues	This is an extra feature that does not affect the project's goals so it can be left out if necessary

References

Grafwallner, P., 2022. *Keeping Learning Real, Relevant, and Relatable*. [online] Edutopia. Available at: <https://www.edutopia.org/article/keeping-learning-real-relevant-and-relatable> [Accessed 5 February 2022].

Hurix Digital. 2022. *7 Benefits of Using Virtual Labs in K-12 Education*. [online] Available at: <https://www.hurix.com/benefits-of-using-virtual-labs-in-k-12-education/> [Accessed 4 February 2022].

Innova Design Group. 2022. *How important is Practical Science in the Classroom? - Innova Design Group*. [online] Available at: <https://www.innovadesigngroup.co.uk/news/practical-science-in-the-classroom/#:~:text=In%20conclusion,broad%20understanding%20of%20scientific%20concepts> [Accessed 4 February 2022].

ProjectPlace, 2022. *How to Conduct Experiments Using the Scientific Method*. [online] Instructables.com. Available at: <https://www.instructables.com/How-to-Conduct-Experiments-Using-the-Scientific-Me/> [Accessed 4 February 2022].

Staff, U., 2022. *Top Advantages and Disadvantages of Studying at Home - UrbanMatter*. [online] UrbanMatter. Available at: <https://urbanmatter.com/top-advantages-and-disadvantages-of-studying-at-home/> [Accessed 4 February 2022].

Research Ethics Review Form: BSc, MSc and MA Projects

Computer Science Research Ethics Committee (CSREC)

http://www.city.ac.uk/department-computer-science/research-ethics

Undergraduate and postgraduate students undertaking their final project in the Department of Computer Science are required to consider the ethics of their project work and to ensure that it complies with research ethics guidelines. In some cases, a project will need approval from an ethics committee before it can proceed. Usually, but not always, this will be because the student is involving other people ("participants") in the project.

In order to ensure that appropriate consideration is given to ethical issues, all students must complete this form and attach it to their project proposal document. There are two parts:

PART A: Ethics Checklist. All students must complete this part.
The checklist identifies whether the project requires ethical approval and, if so, where to apply for approval.

PART B: Ethics Proportionate Review Form. Students who have answered "no" to all questions in A1, A2 and A3 and "yes" to question 4 in A4 in the ethics checklist must complete this part. The project supervisor has delegated authority to provide approval in such cases that are considered to involve MINIMAL risk. The approval may be ***provisional*** – *identifying the planned research as* likely to involve MINIMAL RISK. In such cases you must additionally seek ***full approval*** from the supervisor as the project progresses and details are established. ***Full approval*** must be acquired in writing, before beginning the planned research.

A.1 If you answer YES to any of the questions in this block, you must apply to an appropriate external ethics committee for approval and log this approval as an External Application through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*
1.1	Does your research require approval from the National Research Ethics Service (NRES)? *e.g. because you are recruiting current NHS patients or staff?* *If you are unsure try - https://www.hra.nhs.uk/approvals-amendments/what-approvals-do-i-need/*	**NO**
1.2	Will you recruit participants who fall under the auspices of the Mental Capacity Act? *Such research needs to be approved by an external ethics committee such as NRES or the Social Care Research Ethics Committee - http://www.scie.org.uk/research/ethics-committee/*	**NO**
1.3	Will you recruit any participants who are currently under the auspices of the Criminal Justice System, for example, but not limited to, people on remand, prisoners and those on probation?	**NO**

	Such research needs to be authorised by the ethics approval system of the National Offender Management Service.	
A.2 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee, you must apply for approval from the Senate Research Ethics Committee (SREC) through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*
2.1	Does your research involve participants who are unable to give informed consent? *For example, but not limited to, people who may have a degree of learning disability or mental health problem, that means they are unable to make an informed decision on their own behalf.*	**NO**
2.2	Is there a risk that your research might lead to disclosures from participants concerning their involvement in illegal activities?	**NO**
2.3	Is there a risk that obscene and or illegal material may need to be accessed for your research study (including online content and other material)?	**NO**
2.4	Does your project involve participants disclosing information about special category or sensitive subjects? *For example, but not limited to: racial or ethnic origin; political opinions; religious beliefs; trade union membership; physical or mental health; sexual life; criminal offences and proceedings*	**NO**
2.5	Does your research involve you travelling to another country outside of the UK, where the Foreign & Commonwealth Office has issued a travel warning that affects the area in which you will study? *Please check the latest guidance from the FCO - http://www.fco.gov.uk/en/*	**NO**
2.6	Does your research involve invasive or intrusive procedures? *These may include, but are not limited to, electrical stimulation, heat, cold or bruising.*	**NO**
2.7	Does your research involve animals?	**NO**
2.8	Does your research involve the administration of drugs, placebos or other substances to study participants?	**NO**
A.3 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee or the SREC, you must apply for approval from the Computer Science Research Ethics Committee (CSREC) through Research Ethics Online - https://ethics.city.ac.uk/		*Delete as appropriate*

Depending on the level of risk associated with your application, it may be referred to the Senate Research Ethics Committee.		
3.1	Does your research involve participants who are under the age of 18?	**NO**
3.2	Does your research involve adults who are vulnerable because of their social, psychological or medical circumstances (vulnerable adults)? *This includes adults with cognitive and / or learning disabilities, adults with physical disabilities and older people.*	**NO**
3.3	Are participants recruited because they are staff or students of City, University of London? *For example, students studying on a particular course or module.* *If yes, then approval is also required from the Head of Department or Programme Director.*	**NO**
3.4	Does your research involve intentional deception of participants?	**NO**
3.5	Does your research involve participants taking part without their informed consent?	**NO**
3.5	Is the risk posed to participants greater than that in normal working life?	**NO**
3.7	Is the risk posed to you, the researcher(s), greater than that in normal working life?	**NO**
A.4 If you answer YES to the following question and your answers to all other questions in sections A1, A2 and A3 are NO, then your project is deemed to be of MINIMAL RISK. **If this is the case, then you can apply for approval through your supervisor under PROPORTIONATE REVIEW. You do so by completing PART B of this form.** **If you have answered NO to all questions on this form, then your project does not require ethical approval. You should submit and retain this form as evidence of this.**		*Delete as appropriate*
4	Does your project involve human participants or their identifiable personal data? *For example, as interviewees, respondents to a survey or participants in testing.*	**YES**

PART B: Ethics Proportionate Review Form

If you answered YES to question 4 and NO to all other questions in sections A1, A2 and A3 in PART A of this form, then you may use PART B of this form to submit an application for a proportionate ethics review of your project. Your project supervisor has delegated authority to review and approve this application under proportionate review. You must receive final approval from your supervisor in writing before beginning the planned research.

However, if you cannot provide all the required attachments (see B.3) with your project proposal (e.g. because you have not yet written the consent forms, interview schedules etc), the approval from your supervisor will be ***provisional***. You **must** submit the missing items to your supervisor for approval prior to commencing these parts of your project. Once again, you must receive written confirmation from your supervisor that any provisional approval has been superseded by with ***full approval*** of the planned activity as detailed in the full documents. **Failure to follow this procedure and demonstrate that final approval has been achieved may result in you failing the project module.**

Your supervisor may ask you to submit a full ethics application through Research Ethics Online, for instance if they are unable to approve your application, if the level of risks associated with your project change, or if you need an approval letter from the CSREC for an external organisation.

B.1 The following questions must be answered fully. All grey instructions must be removed.		*Delete as appropriate*
1.1.	Will you ensure that participants taking part in your project are fully informed about the purpose of the research?	**YES**
1.2	Will you ensure that participants taking part in your project are fully informed about the procedures affecting them or affecting any information collected about them, including information about how the data will be used, to whom it will be disclosed, and how long it will be kept?	**YES**
1.3	When people agree to participate in your project, will it be made clear to them that they may withdraw (i.e. not participate) at any time without any penalty?	**YES**
1.4	Will consent be obtained from the participants in your project? Consent from participants will be necessary if you plan to involve them in your project or if you plan to use identifiable personal data from existing records. "Identifiable personal data" means data relating to a living person who might be identifiable if the record includes their name, username, student id, DNA, fingerprint, address, etc.	**YES**
1.5	Have you made arrangements to ensure that material and/or private information obtained from or about the participating individuals will remain confidential?	**YES**

B.2 If the answer to the following question (B2) is YES, you must provide details		*Delete as appropriate*
2	Will the research be conducted in the participant's home or other non-University location? *If **YES**, you must provide details of how your safety will be ensured.*	**NO**

B.3 Attachments **ALL of the following documents MUST be provided to supervisors if applicable.** **All must be considered prior to final approval by supervisors.** **A written record of final approval must be provided and retained.**	***YES***	***NO***	***Not Applicable***
Details on how safety will be assured in any non-University location, including risk assessment if required (see B2)			
Details of arrangements to ensure that material and/or private information obtained from or about the participating individuals will remain confidential (see B1.5) *Any personal data must be acquired, stored and made accessible in ways that are GDPR compliant.*			
Full protocol for any workshops or interviews**			
Participant information sheet(s)**			
Consent form(s)**			
Questionnaire(s)** *sharing a Qualtrics survey with your supervisor is recommended.*			
Topic guide(s) for interviews and focus groups**			
Permission from external organisations or Head of Department** *e.g. for recruitment of participants*			

***If these items are not available at the time of submitting your project proposal, then **provisional approval** can still be given, under the condition that you must submit the final versions of all items to your supervisor for approval at a later date. **All** such items **must** be*

*seen and approved by your supervisor before the activity for which they are needed begins. Written evidence of **final approval** of your planned activity must be acquired from your supervisor before you commence.*

Changes

If your plans change and any aspects of your research that are documented in the approval process change as a consequence, then any approval acquired is invalid. If issues addressed in Part A (the checklist) are affected, then you must complete the approval process again and establish the kind of approval that is required. If issues addressed in Part B are affected, then you must forward updated documentation to your supervisor and have received written confirmation of approval of the revised activity before proceeding.

Templates for Consent and Information

You must use the templates provided by the University as the basis for your participant information sheets and consent forms. You **must** adapt them according to the needs of your project before you submit them for consideration.

Participant Information Sheets, Consent Forms and Protocols must be consistent. Please ensure that this is the case prior to seeking approval. Failure to do so will slow down the approval process.

We strongly recommend using Qualtrics to produce digital information sheets and consent forms.

Further Information

http://www.city.ac.uk/department-computer-science/research-ethics

https://www.city.ac.uk/research/ethics/how-to-apply/participant-recruitment

https://www.city.ac.uk/research/ethics

Appendix B, Reuse Summary

Downloaded Assets

Name	Date of Download	License	URL
Free Laboratory Pack	20/04/2022	Standard Unity Asset Store EULA	https://assetstore.unity.com/packages/3d/props/tools/free-laboratory-pack-123782#description
Anime Classroom	20/04/2022	Standard Unity Asset Store EULA	https://assetstore.unity.com/packages/3d/props/interior/anime-classroom-44135#description
Clipboard	5/04/2022	Standard Unity Asset Store EULA	https://assetstore.unity.com/packages/3d/props/clipboard-137662#publisher
Physics Lab Props	13/02/2022	Standard Unity Asset Store EULA	https://assetstore.unity.com/packages/3d/props/tools/physics-lab-props-63942#publisher
Lab Environment Set	18/01/2022	Standard Unity Asset Store EULA	https://assetstore.unity.com/packages/3d/props/furniture/lab-environment-set-89958
Metal Meter Stick	10/02/2022	Royalty Free License	https://www.cgtrader.com/3d-models/industrial/tool/metal-meter-stick?utm_source=Iterable&utm_medium=email&utm_campaign=campaign_2566120

Code Reuse

In the software four classes reuse code. This code is modified to fit my project. In all cases the code is clearly marked in the relevant files. The files that contained reused code and the source are:

File	Origin URL	Methods Reused
GraphPaper.cs	https://www.youtube.com/watch?v=--LB7URk60A&t=221s	createSquare() OnPopulateMesh()
GridScript.cs	https://www.youtube.com/watch?v=--LB7URk60A&t=221s	createSquare() OnPopulateMesh()
UILineRenderer.cs	https://www.youtube.com/watch?v=--LB7URk60A&t=221s	DrawVerticiesForPoint() OnPopulateMesh() GetAngle()
PickUp.cs	https://www.youtube.com/watch?v=IEV64CLZra8&t=484s	The lines: this.transform.parent = null; GetComponent<Rigidbody>().useGravity = true;

Appendix C, Questionnaire, Participant Information Sheet and Consent Form

Participant Information Sheet

What is the purpose of this project?

The purpose of this project is to create a virtual science experiment to provide the same at-home science learning, to individuals unable to access a laboratory environment. This questionnaire will give important feedback to improve this software to provide the best experience for the user.

Do I have to take part?

Participation in this questionnaire is entirely voluntary. You are permitted to withdraw at any point without incurring any penalties. Should you agree to take part you will be asked to sign a participant consent form

What will happen if I take part?

If you agree to take part in this questionnaire you will play the game whilst being asked a series of questions based on features that are currently implemented. You will be asked for your opinions and in some cases shown multiple versions of features to allow you to draw a comparison.

Is my questionnaire confidential?

Yes, no personal information will be collected, and all responses are entirely anonymous.

What will happen with the results of the questionnaire?

All user feedback will be collected and used to improve the current build of the game to ensure the best experience is created for the end user.

Participant Consent Form

Please tick the following boxes to record consent

1	I have had the project explained to me and have read the participant information sheet	
2	I understand that participation in this questionnaire is voluntary and anonymous	
3	I agree to the recording of my statements for use in project development	
4	I consent to take part in this questionnaire	

Questionnaire

Question 1: Playing the game both with and without crosshairs, which provides a better experience of aiming and realism?

Question 2: Is the text that appears when hovering over an object clear and easy to read? How could this be improved e.g., size and colour.

Question 3: Is the UI colour easy to read? Is this a pleasing colour? What other colours could provide a better experience?

Question 4: The game includes a timer that begins when the game is started, should this be shown to the user? If so, how?

Question 5: When dropping the card through the light gate, the value for the time taken for the card to fall is output on the screen (HUD) as well as shown on the light gate, is this a useful feature?

Question 6: When adding and removing rows to the table view, what is the best way to achieve this? Playing the game with input fields and sliders, is one of these a better mechanic? Is there another suggested method?

Question 7: When using the graph view, what would be a better mechanic for creating the squares on graph paper, a static image, moveable with sliders or scrollable with scroll wheel?

Question 8: When using the graph view, what would provide a better experience for labelling axis? A drag and drop input field mechanic or input fields that generate with each graph line?

Question 9: Is the game enjoyable?

Question 10: Are there any other comments or suggestions you would like to make that could improve the game?

Appendix D, Questionnaire Results

Questionnaire 1

Question 1: Playing the game both with and without crosshairs, which provides a better experience of aiming and realism?

The crosshairs are better as it is easier to see what you are pointing at

Question 2: Is the text that appears when hovering over an object clear and easy to read? How could this be improved e.g., size and colour.

The text could be a darker colour, so it stands out more

Question 3: Is the UI colour easy to read? Is this a pleasing colour? What other colours could provide a better experience?

Yes, the colours are very easy to read

Question 4: The game includes a timer that begins when the game is started, should this be shown to the user? If so, how?

Yes, the timer should be shown the whole time clearly on the screen

Question 5: When dropping the card through the light gate, the value for the time taken for the card to fall is output on the screen (HUD) as well as shown on the light gate, is this a useful addition?

Yes, but it takes away from the realism of the experience

Question 6: When adding and removing rows to the table view, what is the best way to achieve this? Playing the game with input fields and sliders, is one of these a better mechanic? Is there another suggested method?

The input fields are better as they are more specific

Question 7: When using the graph view, what would be a better mechanic for creating the squares on graph paper, a static image, moveable with sliders or scrollable with scroll wheel?

Scroll wheel

Question 8: When using the graph view, what would provide a better experience for labelling axis? A drag and drop input field mechanic or input fields that generate with each graph line?

A drag and drop would be better

Question 9: Is the game enjoyable?

Yes, it is very enjoyable to play

Question 10: Are there any other comments or suggestions you would like to make that could improve the game?

No

Questionnaire 2

Question 1: Playing the game both with and without crosshairs, which provides a better experience of aiming and realism?

The crosshairs take away from the realism but improve the aiming experience

Question 2: Is the text that appears when hovering over an object clear and easy to read? How could this be improved e.g., size and colour.

The text is clear and easy to read

Question 3: Is the UI colour easy to read? Is this a pleasing colour? What other colours could provide a better experience?

The colours work well and are very pleasing

Question 4: The game includes a timer that begins when the game is started, should this be shown to the user? If so, how?

I think showing the timer is important to keep people on track

Question 5: When dropping the card through the light gate, the value for the time taken for the card to fall is output on the screen (HUD) as well as shown on the light gate, is this a useful addition?

No, I think it is unrealistic and should just be on the light gate

Question 6: When adding and removing rows to the table view, what is the best way to achieve this? Playing the game with input fields and sliders, is one of these a better mechanic? Is there another suggested method?

I think sliders are the more user friendly of the two

Question 7: When using the graph view, what would be a better mechanic for creating the squares on graph paper, a static image, moveable with sliders or scrollable with scroll wheel?

Using the mouse wheel to scroll is the best option

Question 8: When using the graph view, what would provide a better experience for labelling axis? A drag and drop input field mechanic or input fields that generate with each graph line?

Dynamic input fields are more impressive and fits with scrolling

Question 9: Is the game enjoyable?

Yes

Question 10: Are there any other comments or suggestions you would like to make that could improve the game?

No

Questionnaire 3

Question 1: Playing the game both with and without crosshairs, which provides a better experience of aiming and realism?

It is easy to see where you are looking so the crosshairs aren't important

Question 2: Is the text that appears when hovering over an object clear and easy to read? How could this be improved e.g., size and colour.

The text is very small to see, especially if your eyesight is poor

Question 3: Is the UI colour easy to read? Is this a pleasing colour? What other colours could provide a better experience?

The white works well but the blue colour could be darker

Question 4: The game includes a timer that begins when the game is started, should this be shown to the user? If so, how?

Showing the timer could be daunting but it is useful, perhaps showing it should be optional

Question 5: When dropping the card through the light gate, the value for the time taken for the card to fall is output on the screen (HUD) as well as shown on the light gate, is this a useful addition?

This does make the value easier to see

Question 6: When adding and removing rows to the table view, what is the best way to achieve this? Playing the game with input fields and sliders, is one of these a better mechanic? Is there another suggested method?

Sliders are a better mechanic

Question 7: When using the graph view, what would be a better mechanic for creating the squares on graph paper, a static image, moveable with sliders or scrollable with scroll wheel?

A slider for zooming on each axis is better as the user is used to using them with the table

Question 8: When using the graph view, what would provide a better experience for labelling axis? A drag and drop input field mechanic or input fields that generate with each graph line?

Dragging and dropping is better because it forces the user to label the axis rather than being provided a box

Question 9: Is the game enjoyable?

It is very fun

Question 10: Are there any other comments or suggestions you would like to make that could improve the game?

The last timer value should stay on the screen so that the user doesn't have to remember it

Questionnaire 4

Question 1: Playing the game both with and without crosshairs, which provides a better experience of aiming and realism?

Since the middle of the screen is where you are looking, crosshairs are not necessary

Question 2: Is the text that appears when hovering over an object clear and easy to read? How could this be improved e.g., size and colour.

The text should be bigger to make it more clear

Question 3: Is the UI colour easy to read? Is this a pleasing colour? What other colours could provide a better experience?

The colours work well but the user could be given options

Question 4: The game includes a timer that begins when the game is started, should this be shown to the user? If so, how?

The timer is good but should start from when the user first interacts with the pieces to give people time to get prepare

Question 5: When dropping the card through the light gate, the value for the time taken for the card to fall is output on the screen (HUD) as well as shown on the light gate, is this a useful addition?

This is not useful as the timer is clear to read anyway

Question 6: When adding and removing rows to the table view, what is the best way to achieve this? Playing the game with input fields and sliders, is one of these a better mechanic? Is there another suggested method?

If the user can press enter to add a new row this could be very intuitive

Question 7: When using the graph view, what would be a better mechanic for creating the squares on graph paper, a static image, moveable with sliders or scrollable with scroll wheel?

A scrolling zoom would be a better mechanic

Question 8: When using the graph view, what would provide a better experience for labelling axis? A drag and drop input field mechanic or input fields that generate with each graph line?

Dragging and dropping

Question 9: Is the game enjoyable?

Yes, it is enjoyable

Question 10: Are there any other comments or suggestions you would like to make that could improve the game?

Not really

Questionnaire 5

Question 1: Playing the game both with and without crosshairs, which provides a better experience of aiming and realism?

The crosshairs make it easy for the player to see where they are looking, but the realism is affected

Question 2: Is the text that appears when hovering over an object clear and easy to read? How could this be improved e.g., size and colour.

A more vibrant colour could make it clearer

Question 3: Is the UI colour easy to read? Is this a pleasing colour? What other colours could provide a better experience?

It is very easy to read and clear, the colours now are good as the blue is quite seamless

Question 4: The game includes a timer that begins when the game is started, should this be shown to the user? If so, how?

The timer is good and is realistic as a classroom would also have a timer

Question 5: When dropping the card through the light gate, the value for the time taken for the card to fall is output on the screen (HUD) as well as shown on the light gate, is this a useful addition?

The output on the screen makes this seem like a game rather than an actual experiment

Question 6: When adding and removing rows to the table view, what is the best way to achieve this? Playing the game with input fields and sliders, is one of these a better mechanic? Is there another suggested method?

I think input fields are better as they allow the user to pick an exact number of rows and columns

Question 7: When using the graph view, what would be a better mechanic for creating the squares on graph paper, a static image, moveable with sliders or scrollable with scroll wheel?

Sliders would be intuitive but wouldn't be as good of an experience for the user

Question 8: When using the graph view, what would provide a better experience for labelling axis? A drag and drop input field mechanic or input fields that generate with each graph line?

Generating the input fields is easier for the user

Question 9: Is the game enjoyable?

Yes, very fun

Question 10: Are there any other comments or suggestions you would like to make that could improve the game?

Not at the moment

Questionnaire 6

Question 1: Playing the game both with and without crosshairs, which provides a better experience of aiming and realism?

The crosshairs are useful but would be better if they were optional and could be toggled on and off

Question 2: Is the text that appears when hovering over an object clear and easy to read? How could this be improved e.g., size and colour.

The text is easy to read so it doesn't need to be changed

Question 3: Is the UI colour easy to read? Is this a pleasing colour? What other colours could provide a better experience?

The colours are nice and calming but a green colour could be good with the white text

Question 4: The game includes a timer that begins when the game is started, should this be shown to the user? If so, how?

The timer should be shown optionally, perhaps in a second screen that is opened optionally?

Question 5: When dropping the card through the light gate, the value for the time taken for the card to fall is output on the screen (HUD) as well as shown on the light gate, is this a useful addition?

The feature is useful but when considering the effects on how the experience is affected, it is not worth it

Question 6: When adding and removing rows to the table view, what is the best way to achieve this? Playing the game with input fields and sliders, is one of these a better mechanic? Is there another suggested method?

Sliders are better than input fields

Question 7: When using the graph view, what would be a better mechanic for creating the squares on graph paper, a static image, moveable with sliders or scrollable with scroll wheel?

Scrolling and sliders would be good, but I don't think a static image would work

Question 8: When using the graph view, what would provide a better experience for labelling axis? A drag and drop input field mechanic or input fields that generate with each graph line?

As the game is educational, dragging and dropping is better

Question 9: Is the game enjoyable?

It could be more fun

Question 10: Are there any other comments or suggestions you would like to make that could improve the game?

A jumping mechanic would make the gameplay more realistic

Questionnaire 7

Question 1: Playing the game both with and without crosshairs, which provides a better experience of aiming and realism?

The crosshairs are not needed as the game is intuitive enough

Question 2: Is the text that appears when hovering over an object clear and easy to read? How could this be improved e.g., size and colour.

The text would be clearer if it were made larger, the colour is suitable though

Question 3: Is the UI colour easy to read? Is this a pleasing colour? What other colours could provide a better experience?

Yes, the background colour is very good but black would be a better text colour

Question 4: The game includes a timer that begins when the game is started, should this be shown to the user? If so, how?

Having the timer as an object in the game would be more realistic

Question 5: When dropping the card through the light gate, the value for the time taken for the card to fall is output on the screen (HUD) as well as shown on the light gate, is this a useful addition?

I don't think this is a useful feature, the light gate is sufficient

Question 6: When adding and removing rows to the table view, what is the best way to achieve this? Playing the game with input fields and sliders, is one of these a better mechanic? Is there another suggested method?

Sliders are more dynamic which allow the user to adjust more easily

Question 7: When using the graph view, what would be a better mechanic for creating the squares on graph paper, a static image, moveable with sliders or scrollable with scroll wheel?

A scrolling mechanic is similar to other programs which means it's probably easier for the user

Question 8: When using the graph view, what would provide a better experience for labelling axis? A drag and drop input field mechanic or input fields that generate with each graph line?

Generating input fields with each line

Question 9: Is the game enjoyable?

Yes, it is good

Question 10: Are there any other comments or suggestions you would like to make that could improve the game?

No

Questionnaire 8

Question 1: Playing the game both with and without crosshairs, which provides a better experience of aiming and realism?

The crosshairs make the experience less realistic, so the game is better without it

Question 2: Is the text that appears when hovering over an object clear and easy to read? How could this be improved e.g., size and colour.

The text could be a darker colour, so it stands out more

Question 3: Is the UI colour easy to read? Is this a pleasing colour? What other colours could provide a better experience?

The text can be difficult to see but this could be due to the text size or font

Question 4: The game includes a timer that begins when the game is started, should this be shown to the user? If so, how?

Having the timer clear on the screen is good

Question 5: When dropping the card through the light gate, the value for the time taken for the card to fall is output on the screen (HUD) as well as shown on the light gate, is this a useful addition?

No, this is not a required feature

Question 6: When adding and removing rows to the table view, what is the best way to achieve this? Playing the game with input fields and sliders, is one of these a better mechanic? Is there another suggested method?

Sliders are a more visually pleasing method for the user

Question 7: When using the graph view, what would be a better mechanic for creating the squares on graph paper, a static image, moveable with sliders or scrollable with scroll wheel?

Scroll wheel is the best

Question 8: When using the graph view, what would provide a better experience for labelling axis? A drag and drop input field mechanic or input fields that generate with each graph line?

Drag and drop

Question 9: Is the game enjoyable?

Yes

Question 10: Are there any other comments or suggestions you would like to make that could improve the game?

Could the table view stay open whilst doing the experiment, so the user doesn't have to remember the values from the timer

Questionnaire 9

Question 1: Playing the game both with and without crosshairs, which provides a better experience of aiming and realism?

The crosshairs are not required as the game is easy to use

Question 2: Is the text that appears when hovering over an object clear and easy to read? How could this be improved e.g., size and colour.

The size is good but the text could be a darker colour that stands out more

Question 3: Is the UI colour easy to read? Is this a pleasing colour? What other colours could provide a better experience?

The colour scheme seems very natural and isn't in the way

Question 4: The game includes a timer that begins when the game is started, should this be shown to the user? If so, how?

Yes, it should be, the middle of the screen is a good place

Question 5: When dropping the card through the light gate, the value for the time taken for the card to fall is output on the screen (HUD) as well as shown on the light gate, is this a useful addition?

No, this is not very useful

Question 6: When adding and removing rows to the table view, what is the best way to achieve this? Playing the game with input fields and sliders, is one of these a better mechanic? Is there another suggested method?

Input fields are a better mechanic

Question 7: When using the graph view, what would be a better mechanic for creating the squares on graph paper, a static image, moveable with sliders or scrollable with scroll wheel?

Sliders

Question 8: When using the graph view, what would provide a better experience for labelling axis? A drag and drop input field mechanic or input fields that generate with each graph line?

Drag and drop is better for teaching the user how to make a proper graph

Question 9: Is the game enjoyable?

Since the game is educational, it doesn't need to be fun

Question 10: Are there any other comments or suggestions you would like to make that could improve the game?

The labels for the axis names and graph title should be drag and drop too. This forces the user to choose what to label

Questionnaire 10

Question 1: Playing the game both with and without crosshairs, which provides a better experience of aiming and realism?

The game does not require crosshairs, it is easy enough to use anyway

Question 2: Is the text that appears when hovering over an object clear and easy to read? How could this be improved e.g., size and colour.

The text size could be variable based on eyesight. But the colour works well

Question 3: Is the UI colour easy to read? Is this a pleasing colour? What other colours could provide a better experience?

A good colour scheme blends in but looks good, I think the blue and white fits this

Question 4: The game includes a timer that begins when the game is started, should this be shown to the user? If so, how?

I don't think the timer should be shown as it is an extra pressure on the player

Question 5: When dropping the card through the light gate, the value for the time taken for the card to fall is output on the screen (HUD) as well as shown on the light gate, is this a useful addition?

Outputting on the screen feels like cheating and taking away step so I don't think it should be included

Question 6: When adding and removing rows to the table view, what is the best way to achieve this? Playing the game with input fields and sliders, is one of these a better mechanic? Is there another suggested method?

Scroll wheels are the best over the others

Question 7: When using the graph view, what would be a better mechanic for creating the squares on graph paper, a static image, moveable with sliders or scrollable with scroll wheel?

Scrolling to zoom in and out is the most intuitive for the player

Question 8: When using the graph view, what would provide a better experience for labelling axis? A drag and drop input field mechanic or input fields that generate with each graph line?

Drag and drop

Question 9: Is the game enjoyable?

Yes

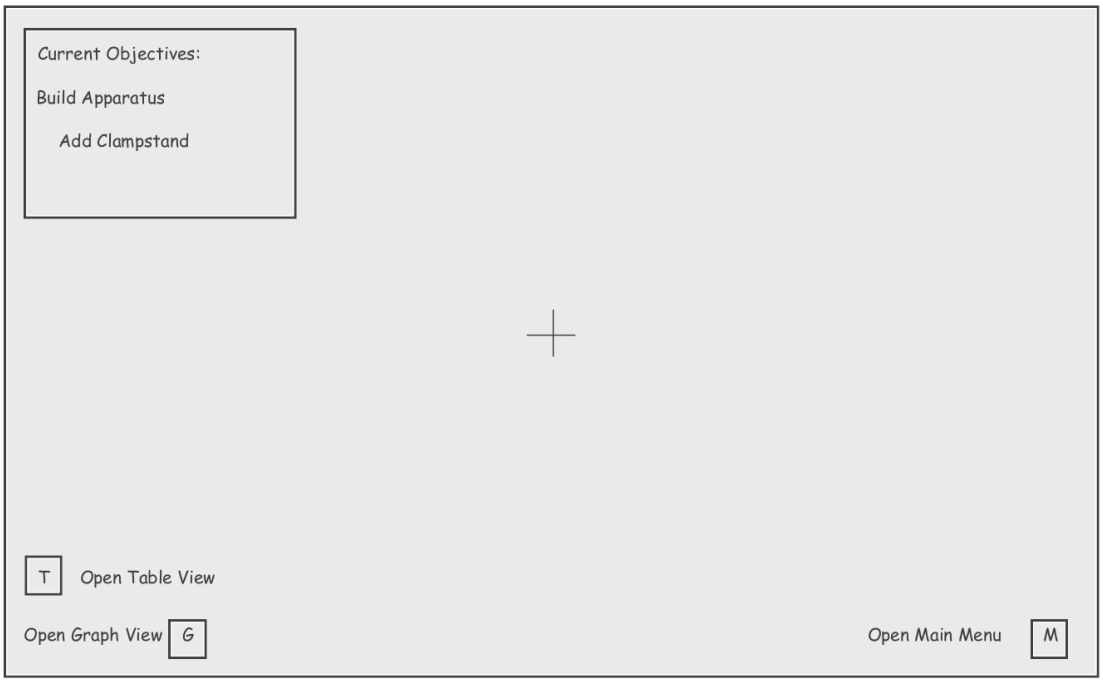
Question 10: Are there any other comments or suggestions you would like to make that could improve the game?

No

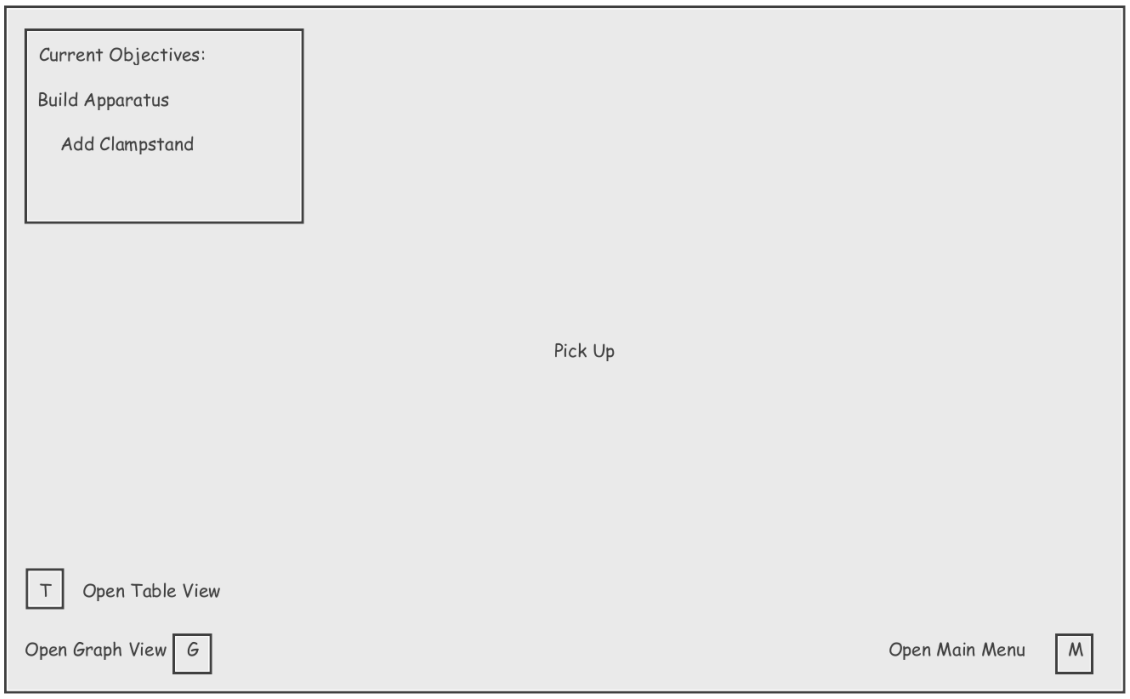
Appendix E: UI Designs

Stage 1 – Initial UI Designs:

HUD when looking around the room:

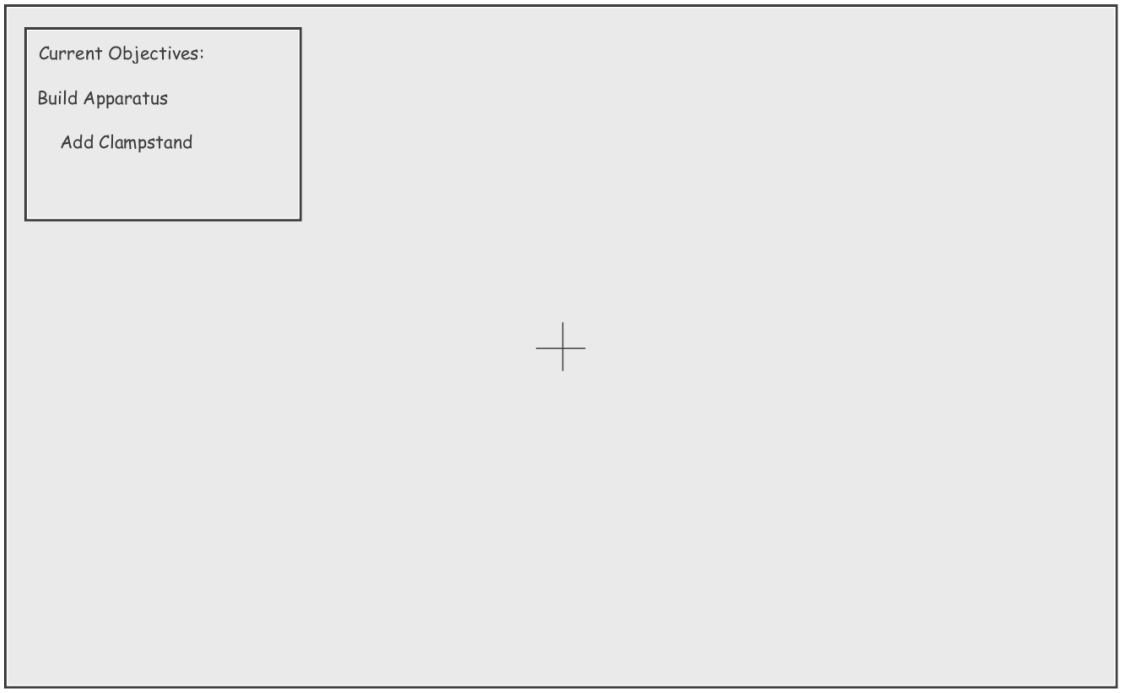


HUD when looking at an object:

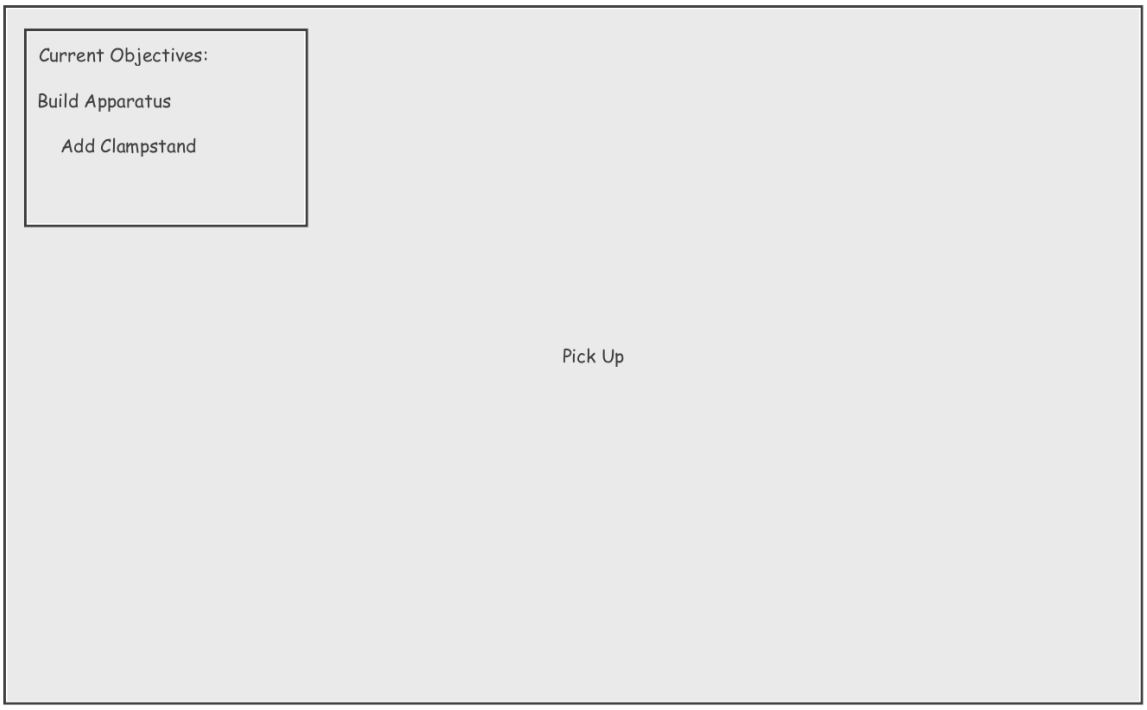


Stage 2 – UI Designs After Beginning Development:

HUD when looking around the room:

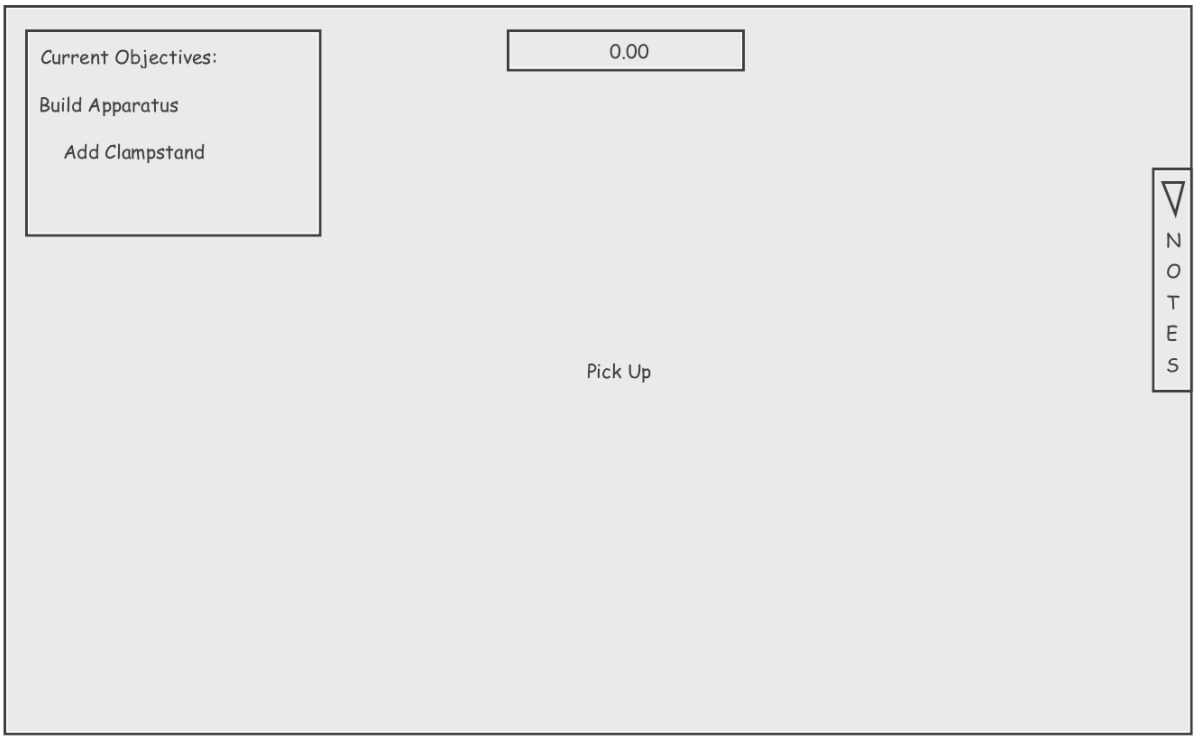


HUD when looking at an object:

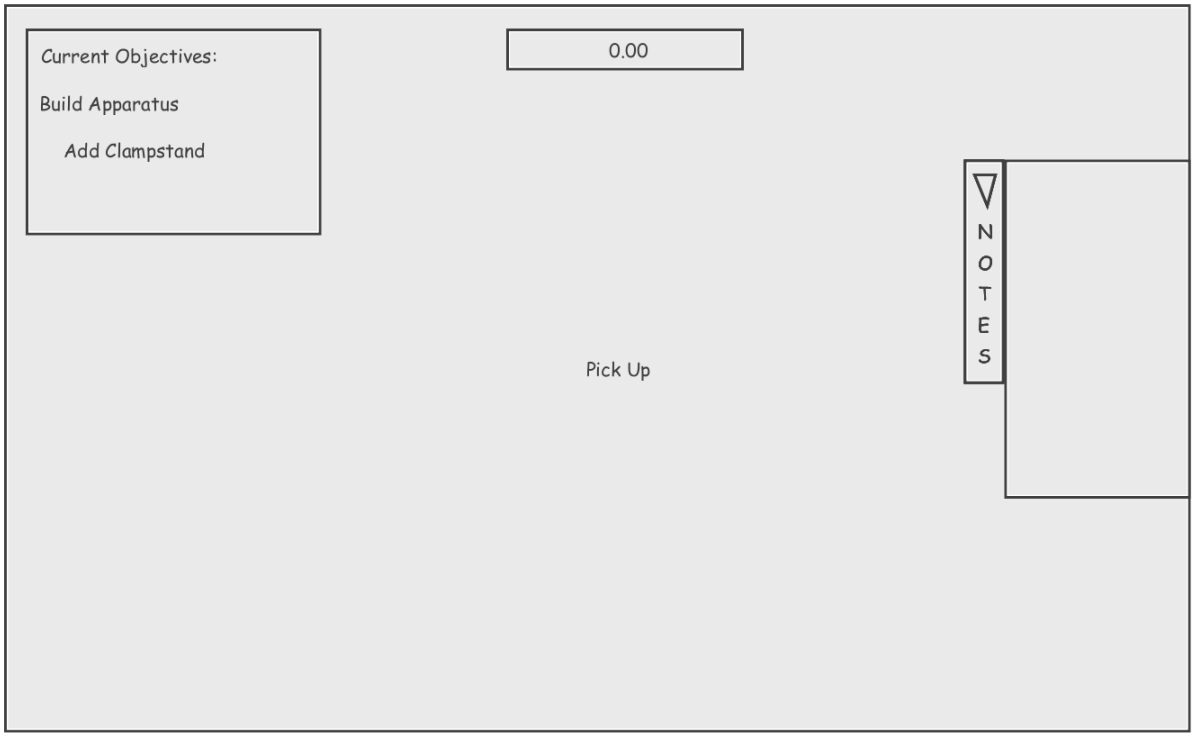


Stage 3 – UI Designs Following User Questionnaire:

HUD when notes screen is closed:



HUD when notes page is open



Appendix F: Evaluation Questionnaire, Participant Information Sheet, and Consent Form

Participant Information Sheet

What is the purpose of this project?

The purpose of this project is to create a virtual science experiment to provide the same at-home science learning, to individuals unable to access a laboratory environment. This questionnaire will give important feedback to improve this software to provide the best experience for the user.

Do I have to take part?

Participation in this questionnaire is entirely voluntary. You are permitted to withdraw at any point without incurring any penalties. Should you agree to take part you will be asked to sign a participant consent form

What will happen if I take part?

You will initially be asked some background information to attain a baseline of your science experience and ability. You will then playtest a completed version of the project and will be asked to record your score and feedback.

Is my questionnaire confidential?

Yes, no personal information will be collected, and all responses are entirely anonymous.

What will happen with the results of the questionnaire?

Your results will be used to evaluate the success of the game as a learning tool

Participant Consent Form

Please tick the following boxes to record consent

1	I have had the project explained to me and have read the participant information sheet	
2	I understand that participation in this questionnaire is voluntary and anonymous	
3	I consent to take part in this questionnaire	

Questionnaire

Question 1: What is your age?

Question 2: Please list your qualifications in biology, chemistry, and physics along with grades attained. E.g., GCSE, A Level, BSc?

Question 3: Do you have experience with videogames such that playing one is intuitive?

Question 4: What grade/score did you achieve after playing the game?

Question 5: What was your personalised feedback given to you after completing the game?

Appendix G: Evaluation Questionnaire Results

Questionnaire 1

Question 1: What is your age?

21

Question 2: Please list your qualifications in biology, chemistry, and physics along with grades attained. E.g., GCSE, A Level, BSc?

GCSE Biology – A*

GCSE Physics - A

GCSE Chemistry - A

A Level Physics - B

A Level Maths - A

Question 3: Do you have experience with videogames such that playing one is intuitive?

Yes, extensively

Question 4: What grade/score did you achieve after playing the game?

41 - A

Question 5: What was your personalised feedback given to you after completing the game?

To improve your value of g, repeat the experiment as unpredictable variables can impact your results

Questionnaire 2

Question 1: What is your age?

20

Question 2: Please list your qualifications in biology, chemistry, and physics along with grades attained. E.g., GCSE, A Level, BSc?

GCSE Biology – A

GCSE Physics - A

GCSE Chemistry - A

A Level Physics - A

Question 3: Do you have experience with videogames such that playing one is intuitive?

Yes

Question 4: What grade/score did you achieve after playing the game?

40 - A

Question 5: What was your personalised feedback given to you after completing the game?

To improve your value of g, repeat the experiment as unpredictable variables can impact your results

Read up on independant and dependant variables to understand what which are which

Questionnaire 3

Question 1: What is your age?

21

Question 2: Please list your qualifications in biology, chemistry, and physics along with grades attained. E.g., GCSE, A Level, BSc?

GCSE Core Science – A

GCSE Additional Science – C

GCSE Triple Science - A

A Level Biology - B

Question 3: Do you have experience with videogames such that playing one is intuitive?

Yes

Question 4: What grade/score did you achieve after playing the game?

37 - A

Question 5: What was your personalised feedback given to you after completing the game?

Remember to label your grid lines with a consistent scale on both axis

To improve your value of g, repeat the experiment as unpredictable variables can impact your results

Remember to label your grid lines with a consistent scale on both axis'

Questionnaire 4

Question 1: What is your age?

22

Question 2: Please list your qualifications in biology, chemistry, and physics along with grades attained. E.g., GCSE, A Level, BSc?

GCSE Biology – B

GCSE Physics - B

GCSE Chemistry - A

Question 3: Do you have experience with videogames such that playing one is intuitive?

Somewhat

Question 4: What grade/score did you achieve after playing the game?

32 - B

Question 5: What was your personalised feedback given to you after completing the game?

Remember to label your grid lines with a consistent scale on both axis'

When creating the graph, remember to label each axis with the variable you are plotting as well as giving the graph a relevant title

In the table remember to plot the units in the headers

Questionnaire 5

Question 1: What is your age?

22

Question 2: Please list your qualifications in biology, chemistry, and physics along with grades attained. E.g., GCSE, A Level, BSc?

GCSE Core Science – C

GCSE Additional Science - B

Question 3: Do you have experience with videogames such that playing one is intuitive?

Yes

Question 4: What grade/score did you achieve after playing the game?

35 - B

Question 5: What was your personalised feedback given to you after completing the game?

Remember to label your grid lines with a consistent scale on both axis'

Draw the line of best fit through all the graph points

When creating the graph, remember to label each axis with the variable you are plotting as well as giving the graph a relevant title

In the table remember to plot the units in the headers

Questionnaire 6

Question 1: What is your age?

28

Question 2: Please list your qualifications in biology, chemistry, and physics along with grades attained. E.g., GCSE, A Level, BSc?

GCSE Biology – B

GCSE Physics - B

GCSE Chemistry - B

Question 3: Do you have experience with videogames such that playing one is intuitive?

No

Question 4: What grade/score did you achieve after playing the game?

42 - A

Question 5: What was your personalised feedback given to you after completing the game?

Draw the line of best fit through all the graph points

Questionnaire 7

Question 1: What is your age?

21

Question 2: Please list your qualifications in biology, chemistry, and physics along with grades attained. E.g., GCSE, A Level, BSc?

GCSE Core Science – B

GCSE Additional Science - B

Question 3: Do you have experience with videogames such that playing one is intuitive?

Yes

Question 4: What grade/score did you achieve after playing the game?

39 - A

Question 5: What was your personalised feedback given to you after completing the game?

Fill in all fields in the table

Draw the line of best fit through all the graph points

In the table remember to plot the units in the headers

Questionnaire 8

Question 1: What is your age?

26

Question 2: Please list your qualifications in biology, chemistry, and physics along with grades attained. E.g., GCSE, A Level, BSc?

GCSE Biology – B

GCSE Physics - B

GCSE Chemistry – A

A Level Chemistry - B

Question 3: Do you have experience with videogames such that playing one is intuitive?

Not really

Question 4: What grade/score did you achieve after playing the game?

33 - B

Question 5: What was your personalised feedback given to you after completing the game?

Fill in all fields in the table

When creating the graph, remember to label each axis with the variable you are plotting as well as giving the graph a relevant title

In the table remember to plot the units in the headers

Questionnaire 9

Question 1: What is your age?

20

Question 2: Please list your qualifications in biology, chemistry, and physics along with grades attained. E.g., GCSE, A Level, BSc?

GCSE Biology – B

GCSE Physics - A

GCSE Chemistry – C

A Level Maths - B

Question 3: Do you have experience with videogames such that playing one is intuitive?

Some

Question 4: What grade/score did you achieve after playing the game?

34 - B

Question 5: What was your personalised feedback given to you after completing the game?

Fill in all fields in the table

When creating the graph, remember to label each axis with the variable you are plotting as well as giving the graph a relevant title

In the table remember to plot the units in the headers

Draw the line of best fit through all the graph points

Questionnaire 10

Question 1: What is your age?

21

Question 2: Please list your qualifications in biology, chemistry, and physics along with grades attained. E.g., GCSE, A Level, BSc?

GCSE Core Science – B

GCSE Additional Science - C

Question 3: Do you have experience with videogames such that playing one is intuitive?

No

Question 4: What grade/score did you achieve after playing the game?

41 - A

Question 5: What was your personalised feedback given to you after completing the game?

Read up on independent and dependent variables to understand what which are which

Remember to label your grid lines with a consistent scale on both axis'

Appendix H: Functional and Non-Functional Requirements and Test Cases

Functional Requirements

ID	Requirement
1	Player can move around using the 'WASD' keys and look around using the mouse
2	Game objects (including player) cannot pass through one another
3	Player can pick up and carry relevant game objects
4	Game objects are correctly added and removed when building apparatus in correct order
5	Light gate records the time taken for the card to fall through
6	HUD outputs correct message when looking at a given game object
7	Notes page should open and record input
8	Table page should create a table with given row and column values and accept user input
9	HUD displays correct stage of the experiment
10	Graph allows users to plot points and draw a line of best fit
11	System should grade player on their performance

Non-Functional Requirements

ID	Requirement
1	User interface is clear and easy to read
2	Game educates users on the scientific method and principles
3	Gameplay is smooth and without glitches
4	Game plays successfully on multiple screen sizes and operating systems
5	Game menu should allow for addition of new levels

Functional Requirement Tests:

Test ID:	**1**
Requirement to Test:	Functional Requirement 1
Test Case Description:	WASD moves player in game world and the mouse rotates the camera.
Test Steps:	1. Hold W 2. Hold A 3. Hold S 4. Hold D 5. Move mouse around screen
Expected Outcome:	The player moves forward when holding 'W', left when holding 'A', backward when holding 'S' and left when holding 'D'. When moving the mouse around the screen, the camera follows
Pass/Fail:	Pass

Test ID:	2
Requirement to Test:	Functional Requirement 2
Test Case Description:	Player attempts to move through objects to ensure they cannot pass through one another
Test Steps:	1. Walk player into tables 2. Walk player into equipment 3. Walk player into walls
Expected Outcome:	The player stops moving when colliding with tables and walls and pushes the game objects when colliding with the equipment
Pass/Fail:	Pass

Test ID:	3
Requirement to Test:	Functional Requirement 3
Test Case Description:	Picking up and dropping each piece of equipment
Test Steps:	For each piece of equipment: 1. Look at object and hold the left mouse click 2. Release mouse click
Expected Outcome:	Whilst the mouse button is down, the object is held in front of the player. When the mouse is released, the object falls
Pass/Fail:	Pass

Test ID:	4
Requirement to Test:	Functional Requirement 4
Test Case Description:	The apparatus is built in the correct order
Test Steps:	1. Player carries each object to the build area in the correct order and releases the object 2. Player carries each object to the build area in the incorrect order and releases the object
Expected Outcome:	The object being held, and current apparatus is deleted, with the next apparatus stage appearing. When the object is released in the wrong order, the apparatus doesn't change, and the object falls
Pass/Fail:	Pass

Test ID:	5
Requirement to Test:	Functional Requirement 5
Test Case Description:	Timer outputs the time for the card to fall through the light gate
Test Steps:	1. Drop the card at a height of 10cm 2. Drop the card at a height of 20cm 3. Drop the card at a height of 30cm 4. Drop the card at a height of 40cm 5. Drop the card at a height of 50cm
Expected Outcome:	The time output on the timer decreases as the height of each drop increases
Pass/Fail:	Pass

Test ID:	6
Requirement to Test:	Functional Requirement 6
Test Case Description:	The pickup text displays the correct value
Test Steps:	1. Look at info page 2. Look at equation sheet 3. Look at final page 4. Look at table page 5. Look at graph page 6. Look at equipment 7. Look at build area whilst holding correct stage of equipment 8. Look at build area whilst holding the incorrect stage of equipment 9. Look at table 10. Look at completed apparatus whilst holding card
Expected Outcome:	1. Pickup text displays 'Open Info Page' 2. Pickup text displays 'Open Equation Page' 3. Pickup text displays 'Open Final Page' 4. Pickup text displays 'Open Table' 5. Pickup text displays 'Open Graph' 6. Pickup text displays 'Pick Up' 7. Pickup text displays 'Build' 8. Pickup text does not appear 9. Pickup text does not appear 10. Pickup text displays 'Drop'
Pass/Fail:	Pass

Test ID:	7
Requirement to Test:	Functional Requirement 7
Test Case Description:	Tests functionality of notes page
Test Steps:	1. Press tab button 2. Type letters and numbers 3. Press tab button
Expected Outcome:	1. Notes page appears 2. Text appears on notes page 3. Notes page closes
Pass/Fail:	Pass

Test ID:	8
Requirement to Test:	Functional Requirement 8
Test Case Description:	Tests table can create rows and columns with input fields
Test Steps:	1. Vary column slider 2. Vary row slider 3. Enter text in each input field
Expected Outcome:	The row and column amounts change to match the slider values, with input fields generating in each field. Each input field accepts user text.
Pass/Fail:	Pass

Test ID:	9
Requirement to Test:	Functional Requirement 9
Test Case Description:	Ensures each stage of the instructions matches relevant experiment stage
Test Steps:	1. Start Game 2. Close info page 3. Close equation page 4. Add clamp stand 5. Add light gate 6. Add ruler 7. Add timer 8. Add wires 9. Close table page 10. Close graph page 11. Close final page
Expected Outcome:	1. Shows info instructions 2. Shows equation instructions 3. Shows clamp stand instruction 4. Shows light gate instruction 5. Shows ruler instruction 6. Shows timer instruction 7. Shows wire instructions 8. Shows table instructions 9. Shows graph instructions 10. Shows final instructions 11. Shows card dropping instructions
Pass/Fail:	Pass

Test ID:	10
Requirement to Test:	Functional Requirement 10
Test Case Description:	Check graph points and line of best fit functionality
Test Steps:	1. Drag points to the graph page and release 2. Click on point to move to a new position 3. Hold and drag a line of best fit onto screen
Expected Outcome:	1. Points follow the mouse and stop where mouse is released 2. Points reposition when picked up again 3. Line of best fit draws following the mouse clicks
Pass/Fail:	Pass

Test ID:	11
Requirement to Test:	Functional Requirement 11
Test Case Description:	Grading system successfully grades player
Test Steps:	1. Enter 'height' as independent variable 2. Enter 'Velocity' as dependent variable 3. Enter a change in y of 14 and a change in x of 2 4. Add a graph label above the graph with the text 'Velocity and Distance', add a graph label on the left of the graph with the text '(V)' and add a graph label underneath the graph with no text 5. Add two grid line labels on the left of the graph with the text '5' and '10' in ascending order. Add three grid line

	labels underneath the graph with the text '10', '20', '30' in ascending order 6. Create a table with three columns. In the first field in column one add the text '(s)', in the first field of column two add the text '(s)' and in the first field of column three add the text '(m/s)' 7. Create a table with 4 rows and 4 columns. Enter a random string of text in every box
Expected Outcome:	1. Gets 3/5 for independent variable 2. Gets 5/5 for dependent variable 3. Gets 3/5 for gravity value 4. Gets 4/6 for graph 5. Gets 3/6 for grid line labels 6. Gets 3/6 for table units 7. Gets 5/5 for no blanks in table Player receives mark of E
Pass/Fail:	Pass

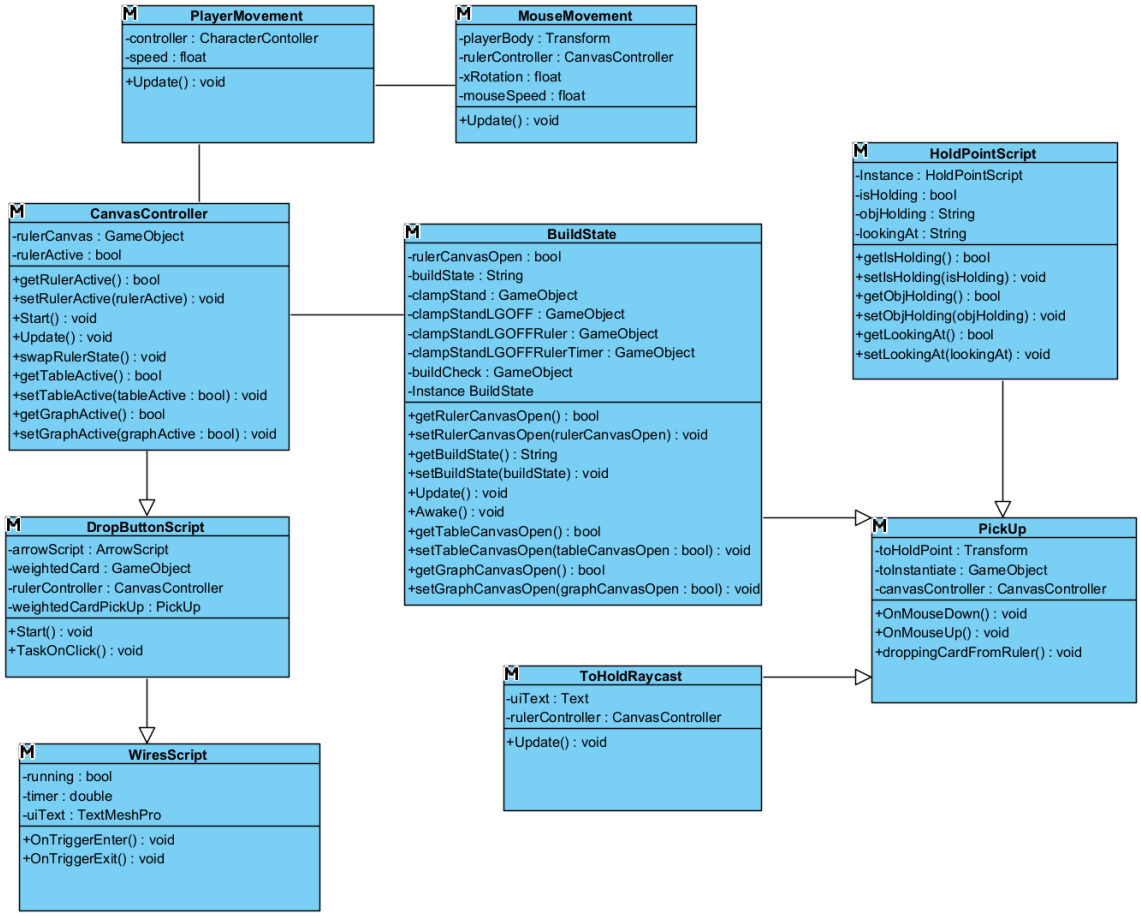
Non-Functional Requirement Tests:

Test ID:	**12**
Requirement to Test:	Non-Functional Requirement 3
Test Case Description:	Tests smoothness of game
Test Steps:	Player plays through game recording any issues in quality
Expected Outcome:	There are no issues in game quality
Pass/Fail:	Pass

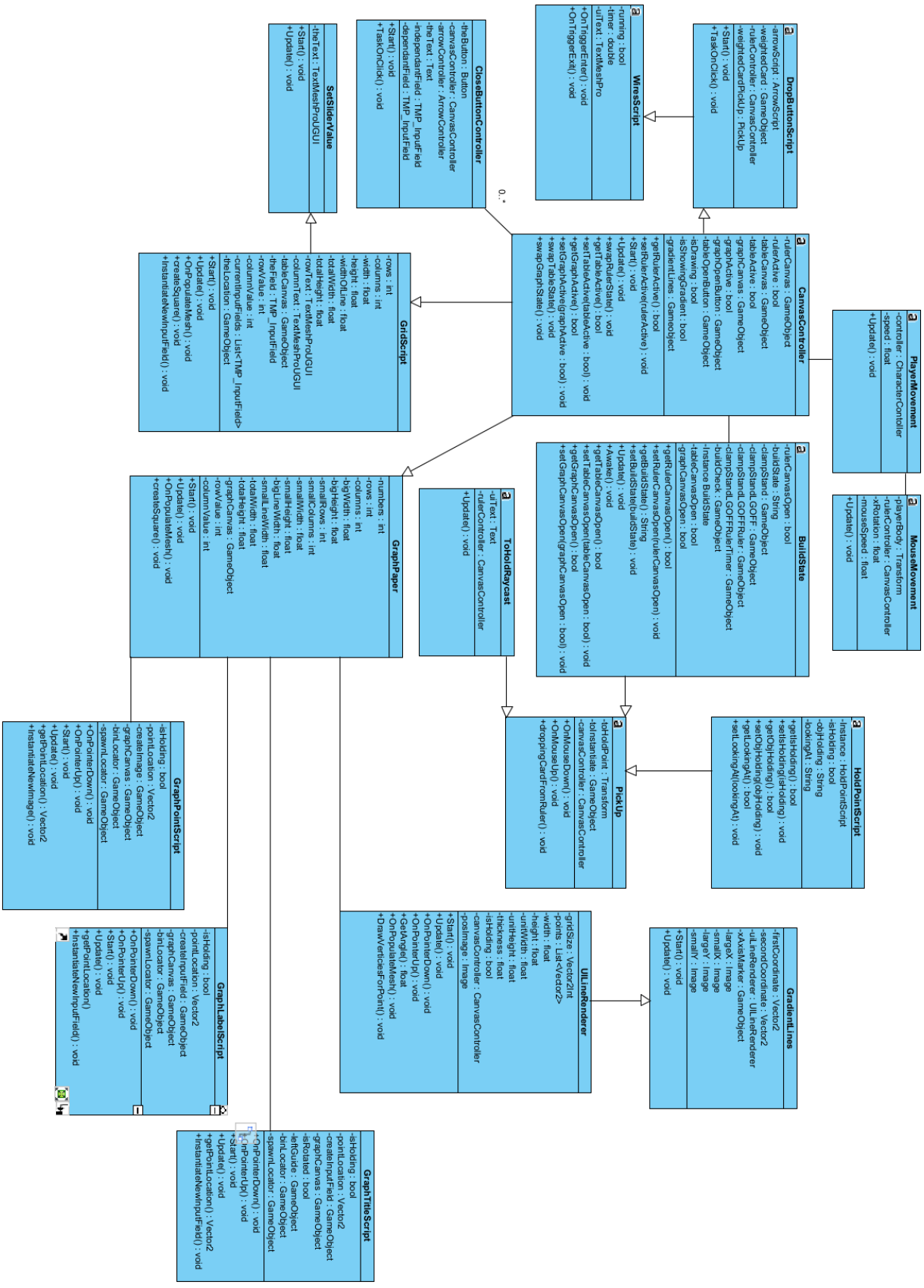
Test ID:	**13**
Requirement to Test:	Non-Functional Requirement 4
Test Case Description:	Tests game functionality on multiple devices
Test Steps:	1. Play through game on high processor device speed 2. Play through game on low processor device speed 3. Play through game on large screen 4. Play through game on medium screen 5. Play through game on small screen
Expected Outcome:	The game plays correctly and identically on all processor speeds and screen sizes
Pass/Fail:	Pass

Appendix I: Class Diagrams

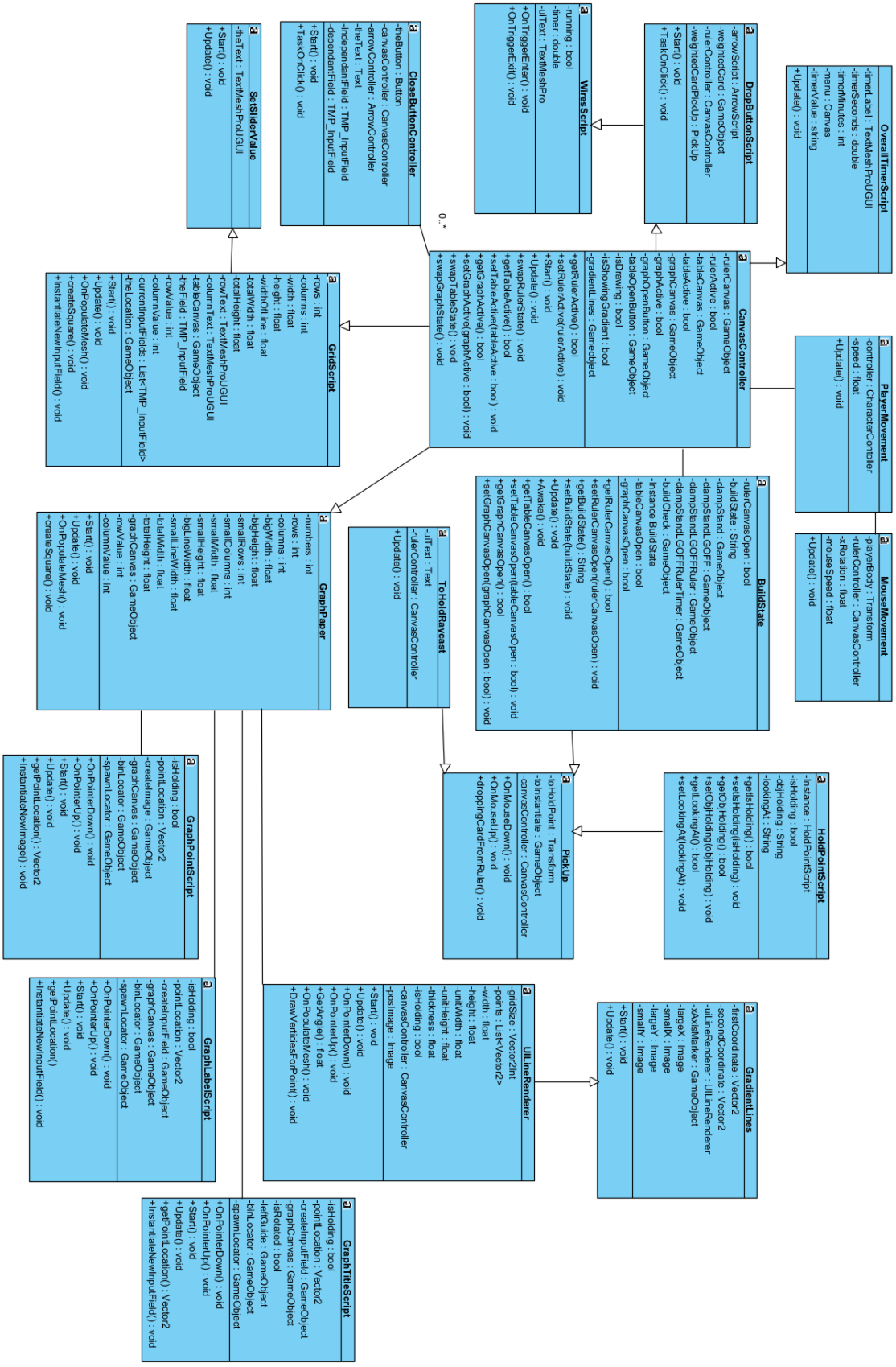
Build 1 Class Diagram:



Build 2 Class Diagram

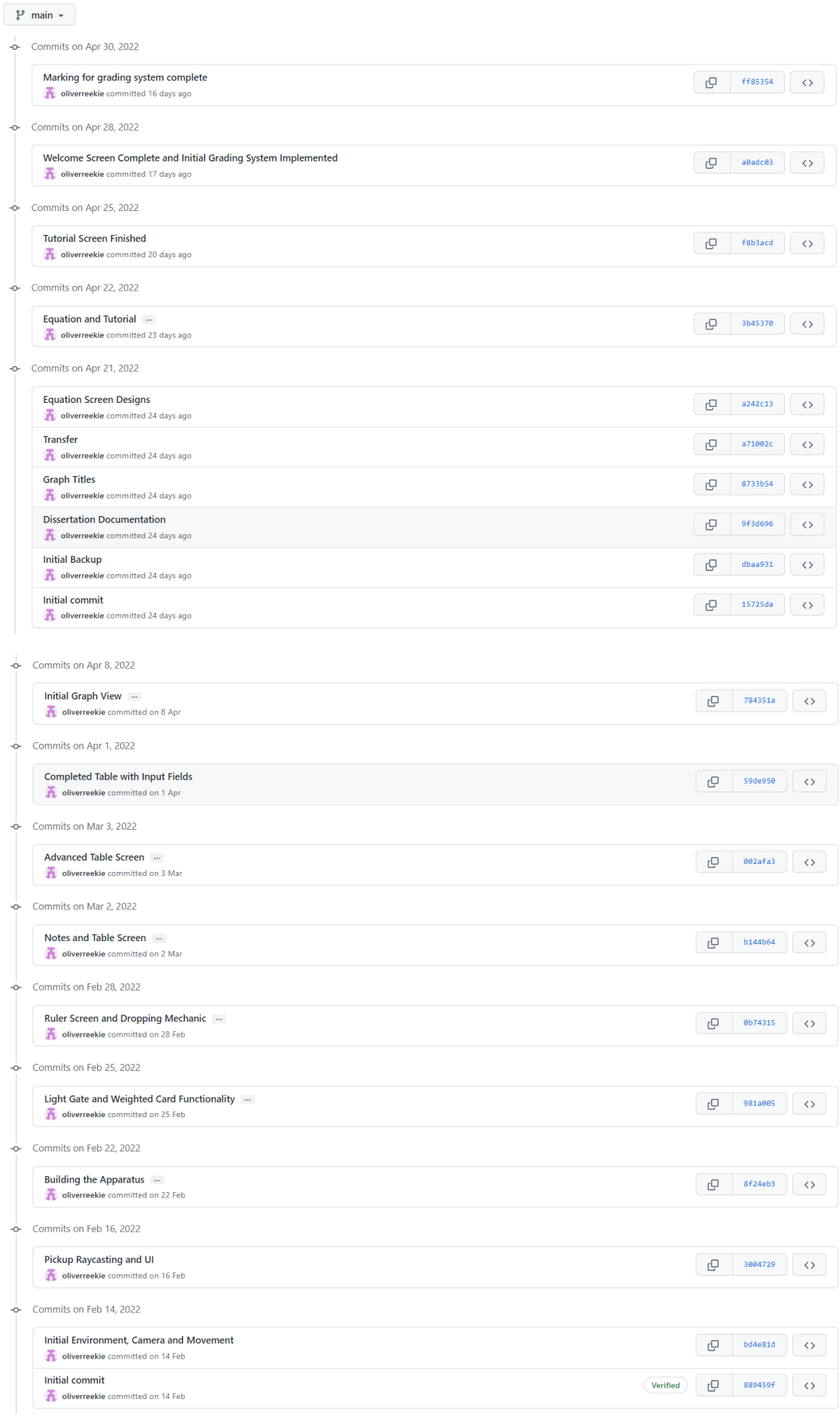


Build 2 Class Diagram – Following User Feedback Questionnaire



Build 3 Class Diagram

Appendix J: GitHub Revision History



Appendix K: Build and Execute Instructions

The executable file and source code were submitted on Moodle.

This project is intended to be run on PC

The executable file can be found in the 'Builds' folder. The file is called 'Individual Project 2'.

To run the application through the Unity Editor you must:

1. Download UnityHub and login to your Unity account
2. Download the latest version of the editor
3. Within UnityHub, click on 'Add' and open the project folder
4. When the project is open click File- Build and Run. This will run the Unity Project