oVRthrow: VR Fitness Game

User Manual

Prepared by Ben Belay, Christopher Stuttard, Isaac Ray and Jonathan Yeap

Note: Red highlighted sections are features of the game that were still in development at the time of releasing this manual (28/4/22).

1. Overview

1.1 Manual Overview

This manual provides all the background required to play and understand the game oVRthrow.

This is a brief overview of the manual. The table above provides a link and the page number of each section.

Section 1 covers the Overview of everything included in the game and the purpose behind the creation of the game.

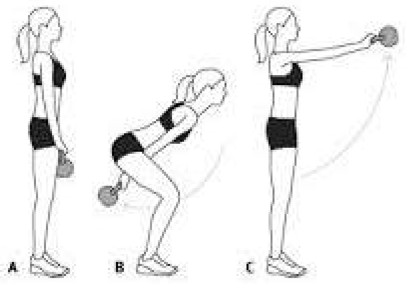
Section 2 outlines how to set up the game and any hardware and software requirements that will need to be met in order to play the game.

Section 3 covers all the mechanics of the game and talks you through the gameplay process. It begins with the menu and tutorial and then explores the 3 stages of gameplay and the scoring process for the game.

Section 4 outlines all maintenance and troubleshooting that is known to be necessary for the game. This includes problems with the VR Equipment and issues within the game itself.

1.2 Product Overview

oVRthrow is a Virtual Reality fitness game. It has been created to provide a fun and engaging alternative to traditional home fitness. The project was developed in response to the adverse effects of the COVID-19 Lockdown, which has impacted the health and fitness of many.
It currently incorporates 2 fitness activities: the Kettlebell Swing and the Squat, which users perform throughout the game. A sketch for each of these exercises is shown below[1].



Spotebi.com

The game stems from the Original Atari '80s' Ramparts' arcade Game. In this game players took part in an attack phase, launching cannonballs between castles to try and destroy them. The game would then have a repair phase where players try to rebuild as much of their castle as they can before combat recommences.
oVRthrow maintains this original basic concept reimagined in a creative and immersive new way.

The game combines the use of several IBM products including Watson SDK and Watson Text-To-Speech. The IBM Watson AI is used to process voice commands that are said by the user in order to invoke power ups within the game as an additional feature.
We currently have the voice command 'raise the shields' which we will discuss in **Section 3.4 Attack Phase**.

The game is built in the Unity Engine, and written in C#. Details on the implementation of SteamVR can be found in the next section. (**Section 2 Setup & Requirements**.)

[1] Kettle Swing: http://www.totalbodyworkout.club/bodylog/tag/WORKOUT
Squat: https://www.spotebi.com/exercise-guide/squat/

2. Setup & Requirements

It is advised not to use a VR headset when you are:

- Tired, anxious or suffering from flu-like symptoms including headaches and migraines as this can lead to an adverse reaction.
- If you are pregnant or have a serious medical condition consult a doctor before using a VR Headset.

oVRthrow is a fitness game where it is important to challenge yourself. However, if you feel lightheaded, nauseous or unwell at any point during your workout, take a break and decide whether it is appropriate to continue.

2.1 Play Space

We declare Play Space to be the allocated area in the real world for playing the game. For your own safety it is advised that you clear a 2x2 metre zone to play inside. This play area will keep your surroundings and others safe while you are playing; the game is very active so people should stand clear!

It is not advised to play in an area that is smaller than the 2x2 metre zone, but if you have to play in a smaller space than is recommended, please make sure your surroundings are clear from objects that may cause you harm; as well as making sure that the space in front of you is clear. Remember that you cannot see your surroundings whilst you are wearing the headset.

2.2 SteamVR

oVRthrow is built for the **SteamVR** Virtual Reality platform, and was developed using the **Quest 2 and Oculus Link**. SteamVR also makes the game compatible with a large range of Virtual Reality headsets, listed below.

- HTC Vive
- HTC Vive Pro
- HTC Vive Cosmos
- Oculus Quest
- PiMax 8K
- Valve Index

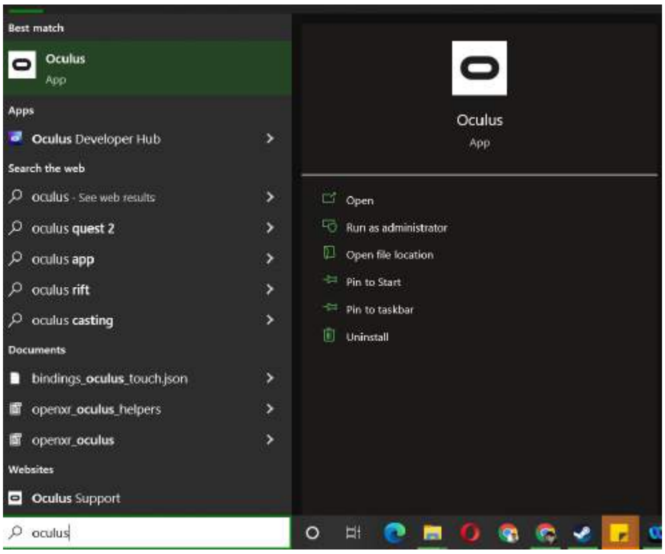
Note that all of these headsets (with the exception of Quest 2 via *AirLink*) must be connected to the PC via a cable.
In order to run this application on your PC, install the latest available version of SteamVR.
The best way to install SteamVR as a user is through the Steam launcher, here:
SteamVR on Steam

2.3 Setting Up The Quest 2 Headset

2.3.1 Oculus Link

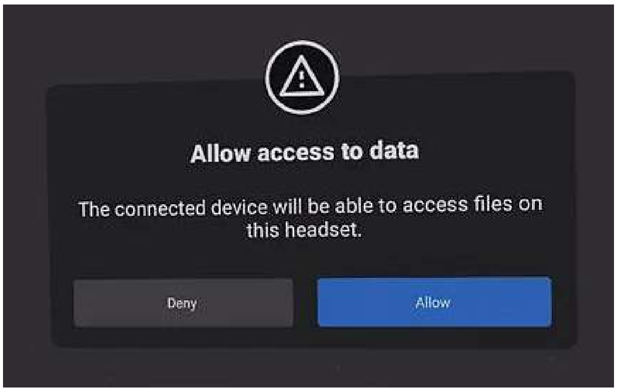
If you want to use a Quest headset to run this game, you will need to first activate Oculus Link.

1. Open the Oculus App



2. Connect the Quest 2 to your PC using a USB->USB-C Cable. Ensure you use a USB3 port.

3. Inside the headset, the Quest 2 will display a prompt. Select 'Deny'.



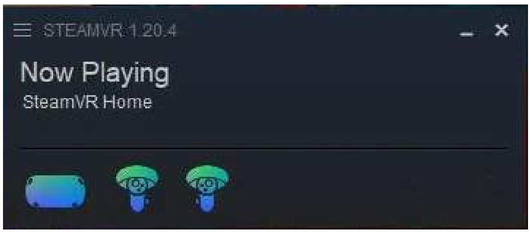
Allow access to data

The connected device will be able to access files on this headset.

4. The Quest should now display the option for Oculus Link to the PC. Select it to enable.

5. If SteamVR is already running on your PC, you will need to restart it. Otherwise, start SteamVR.

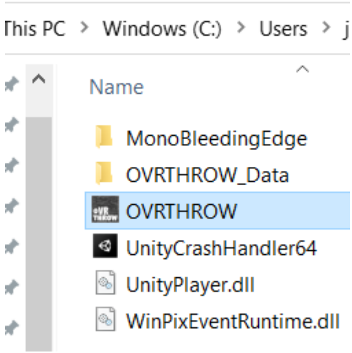
6. You should now have access to the SteamVR interface through your Quest 2 headset.



2.4 Running the Game

2.4.1 Quick Start Guide

STEP 1



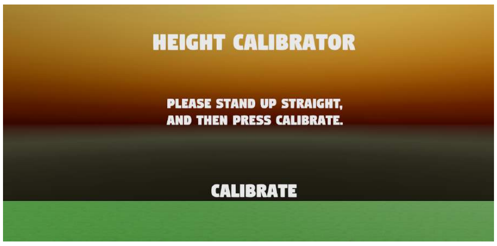
```
MonoBleedingEdge
OVRTHROW_Data
OVRTHROW
UnityCrashHandler64
UnityPlayer.dll
WinPixEventRuntime.dll
```

Make sure you have completed the Oculus link steps in **Section 2.3.1**. You must have SteamVR running in the background.

Locate “OVRTHROW.exe”. Double click on the executable to run the game.

You can now put on your headset. You will now need to use your pointer to interact with the game interface.

STEP 2

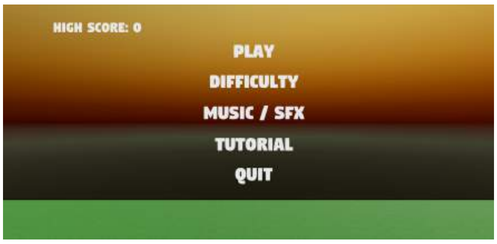


PLEASE STAND UP STRAIGHT, AND THEN PRESS CALIBRATE.

You will be greeted with the height calibration screen.

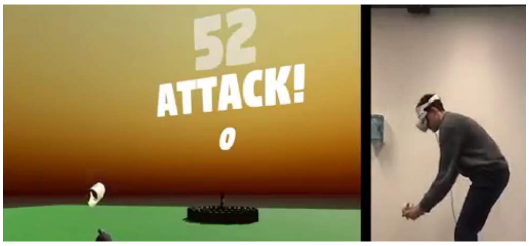
After you have completed this, your height will be saved with the executable. So, next time you open the game, you won’t have to calibrate.

STEP 3



The main menu screen loads. Click the “Tutorial” button.

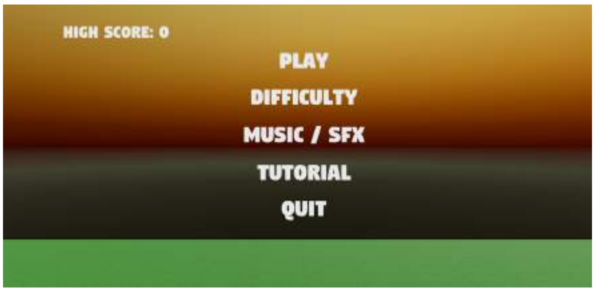
STEP 4



A tutorial video will play, showing the game played in third-person. You shall see how each of the three game phases work, as well as how to pause, quit and recalibrate your height. Text annotations will appear in the tutorial to help you with this walkthrough.

If you watch the full tutorial, you will automatically be taken back to the main menu. Or, you can end it early by clicking “Back”.

STEP 5



You have returned to the main menu. Click “Play” to run the game.

The game is run by default in “Easy” mode.

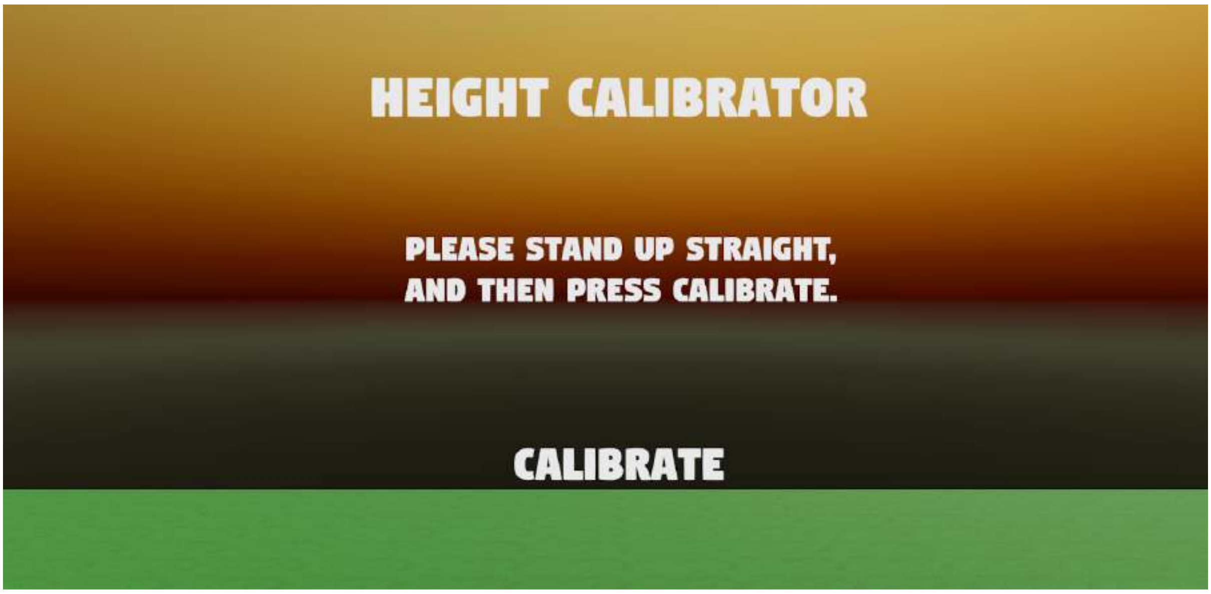
Enjoy the game! Please refer to the relevant sections in this manual for further support, and game customisation.

3. Mechanics & Gameplay

This section details each step of the gameplay process, looking at the relevant underlying functional mechanics involved in the game - for a thorough understanding.

The first step for new users is to calibrate their height by taking a measurement of how high their headset is above the ground.

3.1 Height Calibration



HEIGHT CALIBRATOR

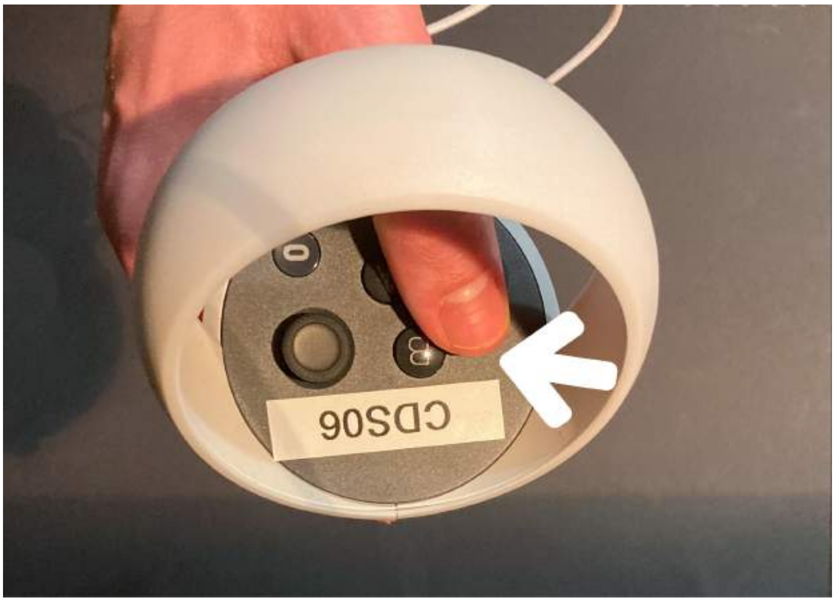
PLEASE STAND UP STRAIGHT,
AND THEN PRESS CALIBRATE.

This is the first screen that you will see when loading the game for the first time.

You must use the height calibrator before continuing to the main menu and playing the game. Simply stand up straight with your feet flat to the ground, and then click the “Calibrate” button on-screen with your controller.

You will then be taken to the main menu, shown in **Section 3.2**. Calibration is important to ensure that the Squat motion, shown in **Section 3.6**, can be achieved without injury to the user. Note that the application saves your height for use in later sessions, so when you next load the game you will be sent straight to the main menu.

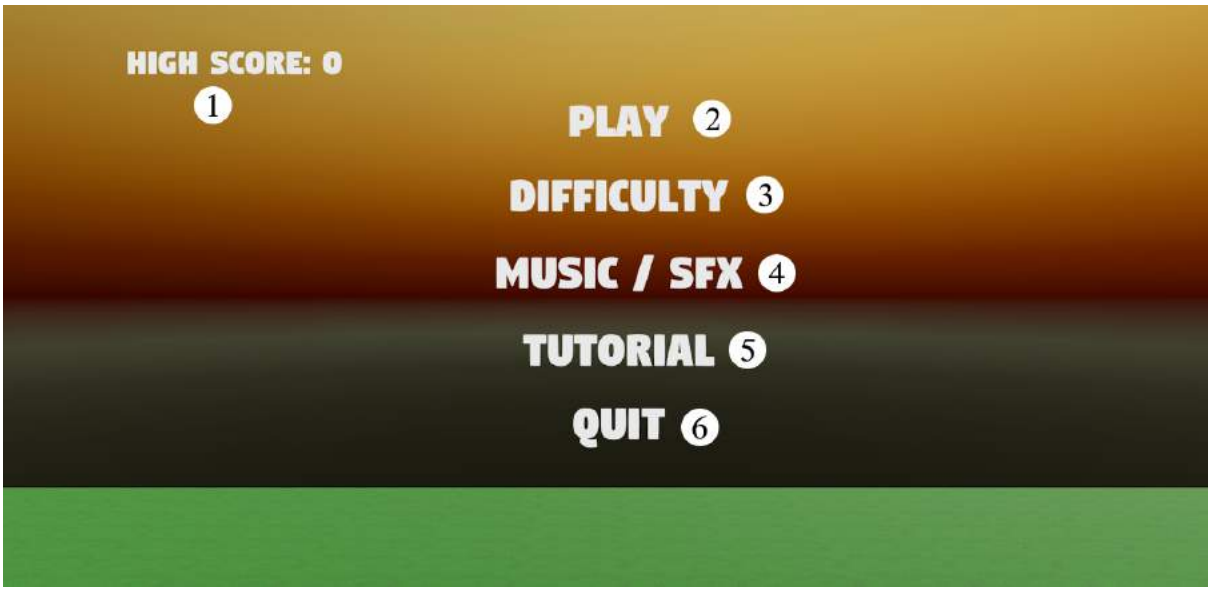
3.1.1 Re-calibrating



At any time while the game is running, you can recalibrate your height by hitting the “B” button on either controller (indicated above). This will load the height calibrator screen, as seen in **Section 3.1**. After hitting “Calibrate”, you will be returned to your current game scene.

3.2 Navigating the Menu

On the following page is the menu screen. In this manual we have given each feature on the menu screen a labelled number and we will discuss it further. Feature 1 is the only feature that is not a button.



1) This is the highest score from all games the user has played. See **Section 3.9.2** for more on the points system. This score is the target you would like to beat in order to achieve a new personal best.

2) *Run the game*. This button will load the main game scene. This will spawn you into the attack phase of the game at the difficulty level you have chosen using Feature 3.

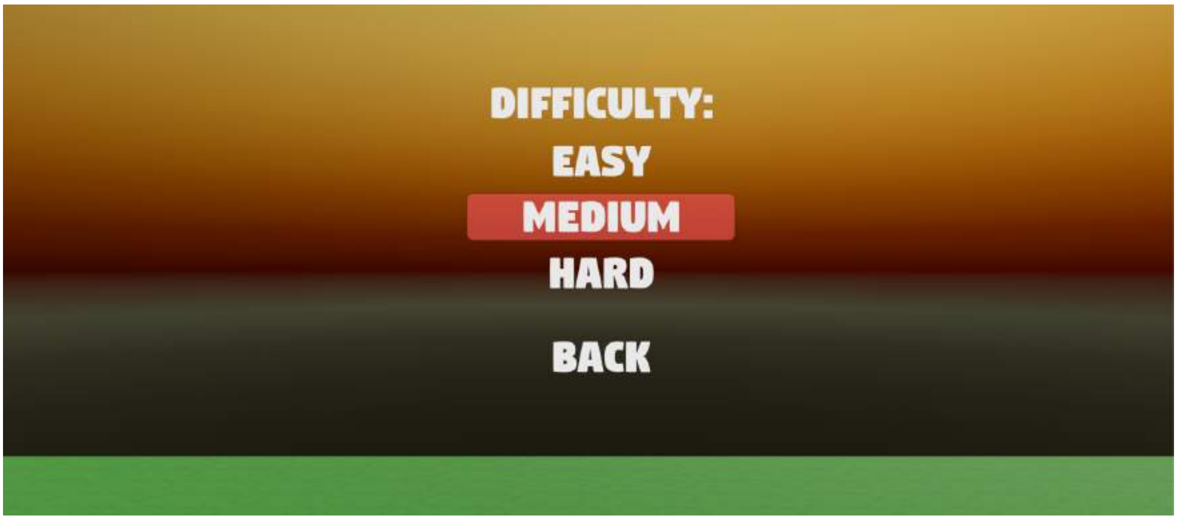
3) *Set a difficulty setting*. See “Difficulty Level” below. The difficulty level defaults to easy mode if you don’t choose a difficult level before starting the game.

4) *Sound volume controls*. See “Music and SFX” below.

5) *Tutorial for the exercises in the game*. See “Tutorial” below.

6) *Quits the Unity application and returns you to SteamVR Menu*. You can now remove the headset.

3.2.1 Difficulty Level

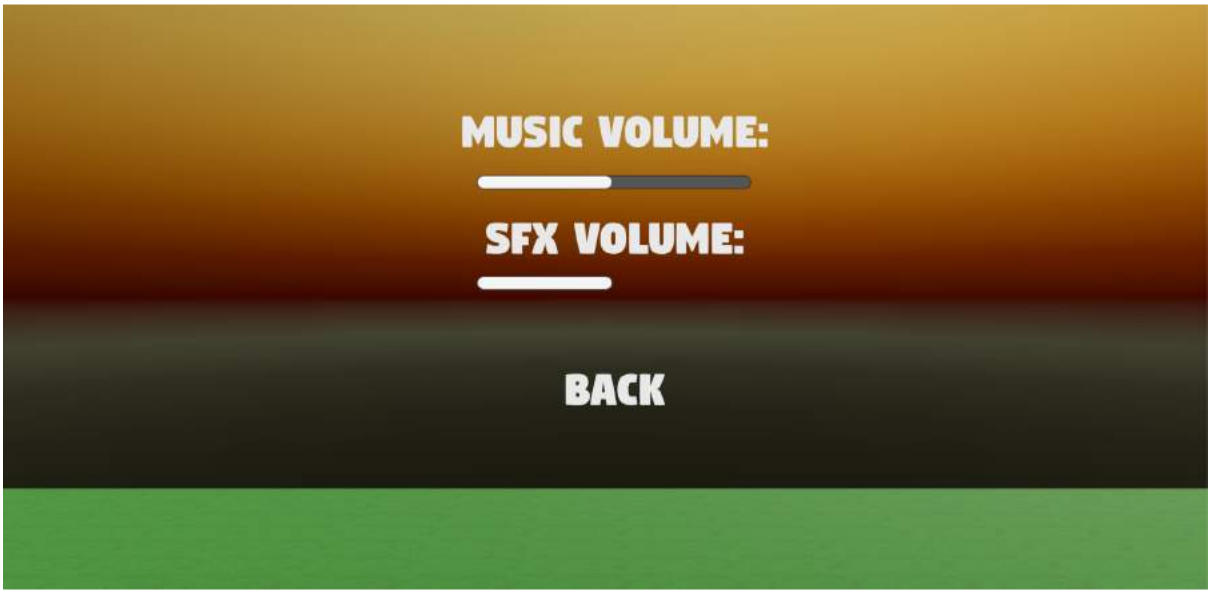


There are three difficulty modes to choose from: Easy, Medium and Hard. These modes set the following gameplay conditions:

Difficulty Level	Maximum Health[2] of enemy walls[3]	Cannon cool down[4]	Number of Rounds
Easy	3-5	6 seconds	5
Medium	4-7	5 seconds	7
Hard	5-9	4 seconds	10

The game will default to the “Easy” difficult level. We advise beginning with the easy difficulty level to get used to the game as it can be a lot more of a challenge than you would initially expect.

3.2.2 Music and SFX



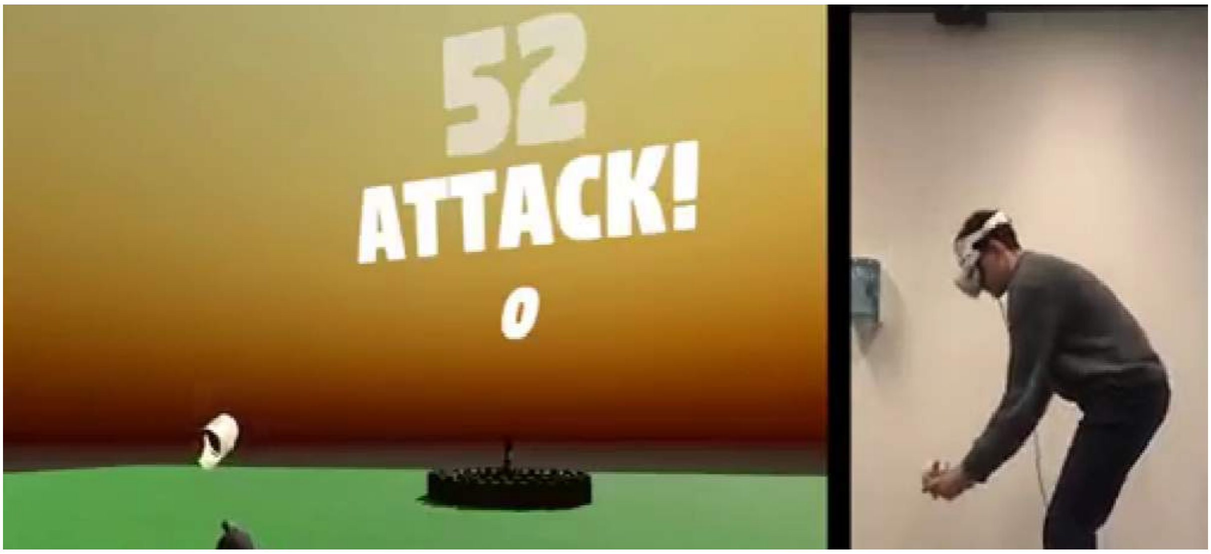
“Music volume” controls the volume of background music in the game, and “SFX volume” controls the volume of sound effects in the game. Moving these sliders up or down will make the volumes louder or quieter. “Back” will load the main menu screen once again, while saving the volume states. Adjust the volume to whatever feels most pleasant for you. We advise watching the tutorial before playing the game for the first time to check the volume is at the correct level for you. The volume selections you make will be saved between exit-ing and launching the game each time.

[2] Maximum Health is described in Section 3.4.7.

[3] Increases with every other round. e.g. Easy mode values will be 3, 3, 4, 4, 5 for rounds 1 to 5.

[4] Cooldown is the wait time between each cannonball being fired.

3.2.3 Tutorial



Clicking “Tutorial” will load an example video of how each phase of the game is played (Attack Phase, Repair Phase, Rest Phase).

The video is a third-person perspective of the game on the left, and a profile video of what actions the player is making with their headset and controllers. The voiceover describes how each action is being performed and what is happening within the game.

At any time, you can click “Back” to leave the tutorial and reload the main menu if you are happy and ready to begin playing.

3.3 Game Stages

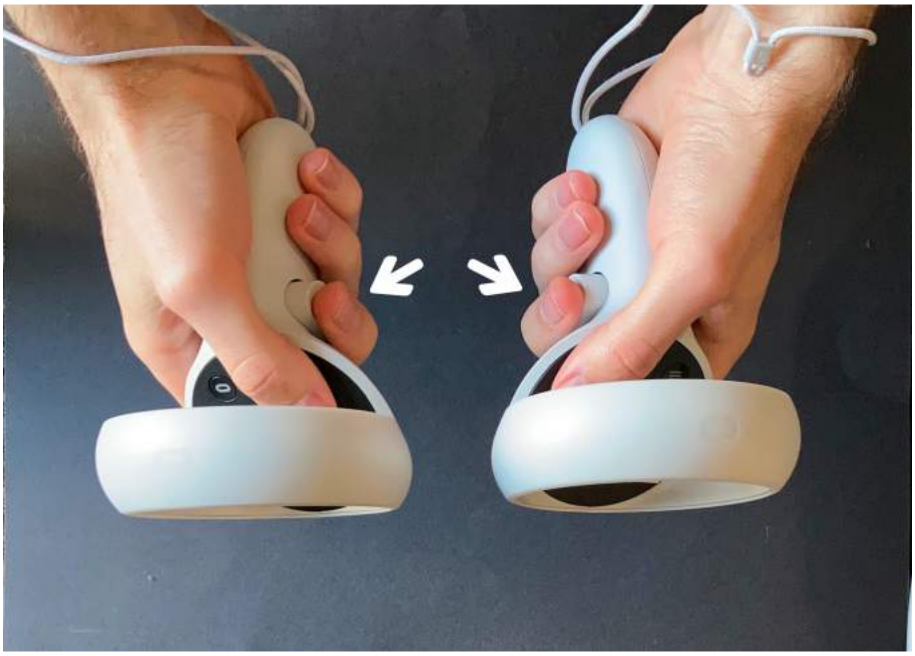
The Game Consists of 3 Stages or Phases. Each Stage has a different duration and different things happen within each phase.
The table below outlines the key facts about each Phase before we discuss the mechanics and gameplay that takes place in each one in further detail.

Phase	Duration (seconds)	Exercise
Attack	60	Kettlebell Swing
Repair	30	Squat
Rest	20	-

3.4 Attack Phase

In the **Attack Phase** players have 60 seconds to throw cannonballs at the enemy in order to score points. They will score points for every successful hit they manage to land on the enemies castle (see **Section 3.9.2** for more on the points system). The user picks up the cannonball and throws it using a kettlebell swing in the direction of the enemies castle. We will walk you through how each game mechanic works and the gameplay process in this stage.
The user can also invoke a power-up during this phase by shouting the command ‘raise the shields’. We will discuss this and the Watson AI after talking through the integral mechanics.

3.4.1 Picking Up Cannonballs



Cannonballs are picked up from the feeder by holding both controller grips at once (indicated above).

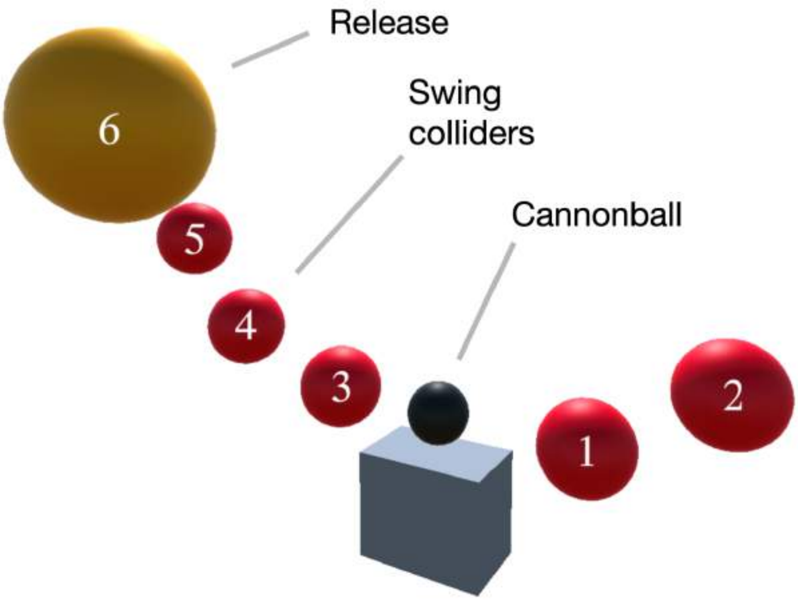
Once holding the grips, you must throw the cannonballs using the correct kettlebell swing. To understand exactly how this must be done, see the rest of this section. Cannonballs are released by letting go of both grips at once.

After throwing cannonballs, new cannonballs will be supplied in your castle’s feeder tray. There is an unlimited respawn of cannon balls for the 60 seconds of the attack phase and one will respawn as soon as the previous cannonball has been thrown.

3.4.2 Kettlebell Swing

Making a successful kettlebell swing throw of a cannonball allows you to damage the enemy walls. The game uses an analyser that will detect if a user makes a successful or incorrect kettlebell swing throw. If the throw is unsuccessful the cannonball will simply fade and vanish, and not cause damage to the opponent.

3.4.3 Swing Analyser



The above diagram shows the invisible mechanism that monitors the quality of the user's kettlebell swing. Numbers 1-6 indicate collider objects[5] that must be triggered by the user's cannonball in a specific order. Triggering these in the correct order will cause a "success" output from the analyser. Any other input from the user will result in a "failure" output from the analyser.

The order must be: 1, 2, 2, 1, 3, 4, 5, 6, and the user must release their grip buttons while passing through collider 6. If the user doesn't trigger the colliders in the correct order, or doesn't trigger them all, there will be an error prompt. See **Section 3.4.5** for a table of the prompts.

Importantly, only a successful kettlebell swing will allow your cannonball to damage the opponent. So, to cause more damage to the enemy, and gain more points, you must make as many accurate swings as possible.

In the following section we will take a further look at guidance for performing an optimum kettlebell swing.

[5] A Unity Collider object. See: https://docs.unity3d.com/ScriptReference/Collider.html

Note: If the behaviour of the kettlebell swing analyser is unexpected (i.e. you feel your kettlebell swing is adequate), then you may need to recalibrate your height.
To do so, press the “B” button on either controller. For more detail see **Section 3.1.1.**

3.4.4 Kettlebell Swing Guidance

We recommend reading professional advice on how to perform an optimum kettlebell swing. Here is one source:
https://www.coachmag.co.uk/kettlebell-exercises/1730/how-to-do-a-kettlebell-swing

In the Attack Phase, it is important that you perform a safe kettle swing. Not doing so runs a high risk of injury. The following steps assume that the player is already in the attack phase, wearing the headset.

1. Stand with your fleet flat on the ground, shoulder-width apart. Your eyesight should remain forwards for as much as the kettlebell swing as possible.

2. Pick up the cannonball by slightly bending your knees, keeping a straight back, and leaning towards the cannonball with outstretched arms. Upon touching the cannonball, your back should be at roughly 45 degrees. If your arms aren’t outstretched fully when touching the cannonball, you may need to stand further back.

3. Pick up the cannonball, using both controllers.

4. Keeping outstretched arms, swing the cannonball through your legs and behind you, as far back as is natural.

5. Then, swing the cannonball forward, driving your hips forward and straightening your back, to send the controllers up to shoulder height. Release the cannonball. At this point you should be standing up fully straight. It is crucial to straighten your back and drive your hips forward to launch the cannonball up.

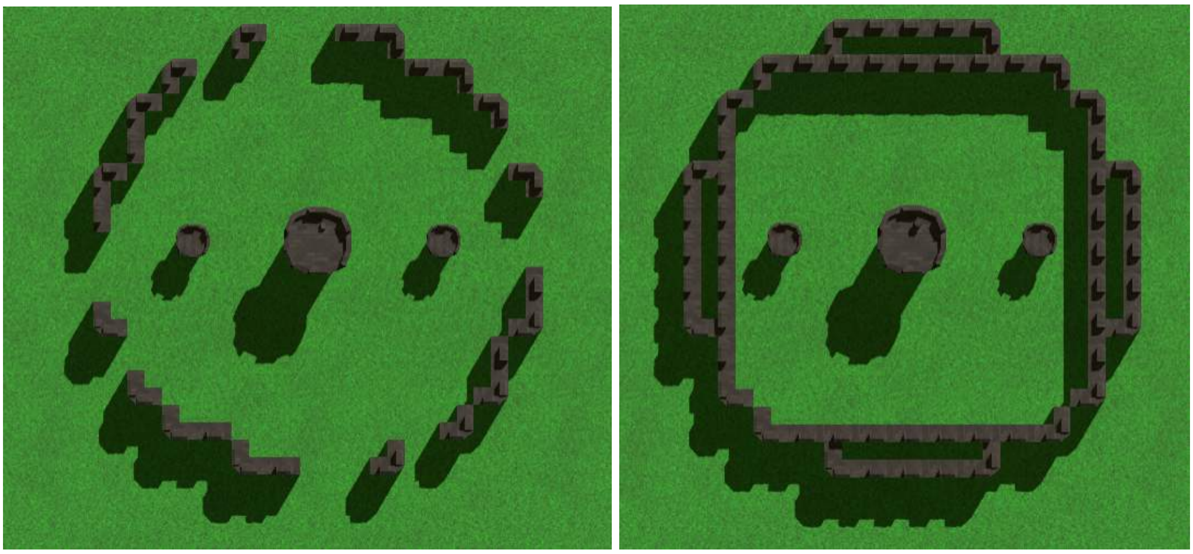
6. Let the controllers return back between your legs and repeat the motion. Naturally return to the position to pick up the next cannonball in one smooth motion.

3.4.5 Kettlebell Swing Prompts

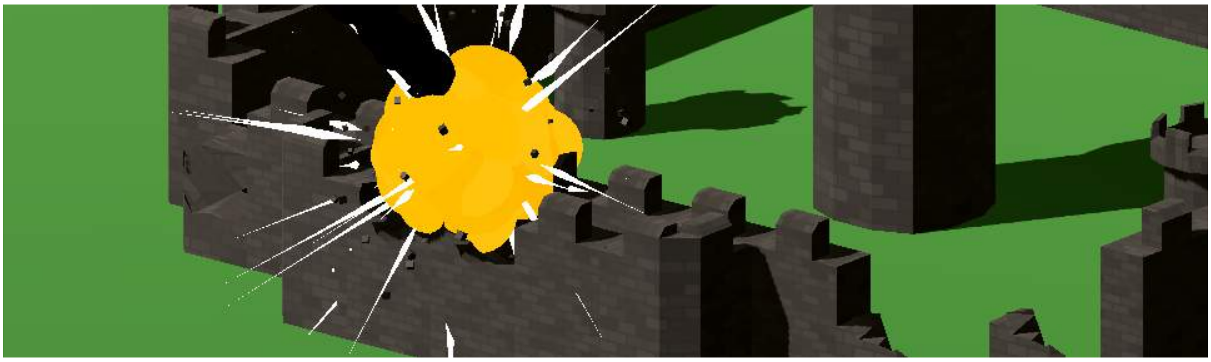
Prompt	**Reason(s)**
“You need to swing behind you first!”	The user has hit colliders 3,4,5 before 1.
“You need to swing back further!”	1. User hit collider 1 but not collider 2. OR: 2. User hit collider 1 and 2 but did not exit and hit collider 2 again.
“You need to swing further forwards!”	The user did hit colliders 1, 2, 2 but then did not complete hitting colliders 3,4,5 or 6.
“Bad throw! Try again...”	1. The user hit a collider other than collider 1 to start with. OR: 2. The user didn’t release the cannonball during collider 6.

3.4.6 Wall Initialisation

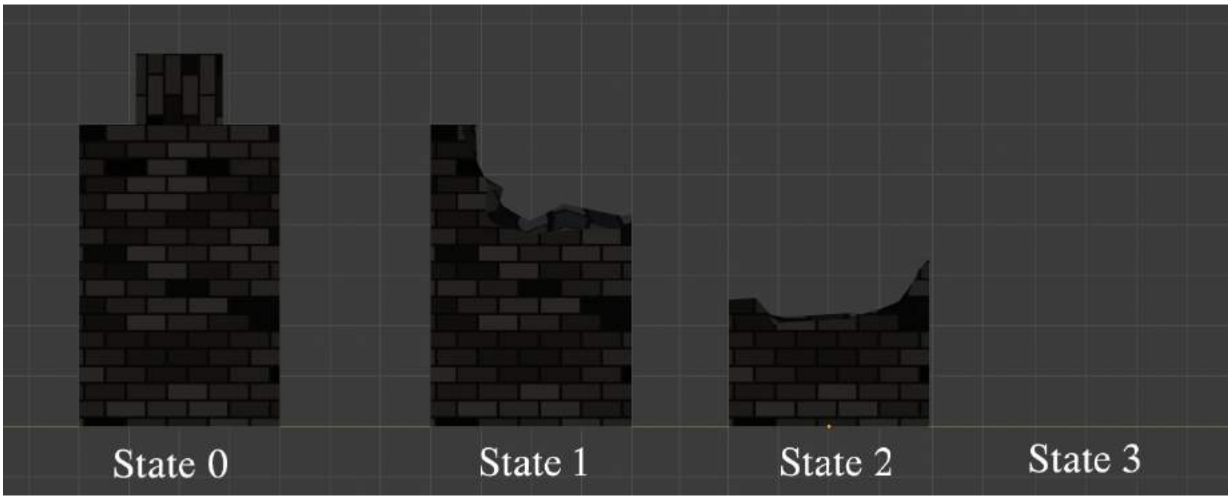
The castle walls are composed of **sections**. These are either a corner piece or a straight piece. Each wall is procedurally generated from a random pool of sections. Below is a birds-eye view, showing before and after generation. The right castle is what the user will be spawned into, when starting the game.



3.4.7 Wall Destruction



Each section can be damaged individually. For each section, there are three visual states which change to reflect its level of damage. These are shown on the following page. The state is calculated using a simple equation involving the amount of hits it has taken and the amount of hits it can take before it is fully destroyed.



Section Health	Visual State
0%	State 0
1-49%	State 1
50-99%	State 2
100%	State 3

The following calculation is used to determine the **Section Health**. This value corresponds to a visual state as seen in the table above.

$$Section\, Health \;=\; 100 \;-\; (\frac{Hits\, Taken}{Maximum\, Health} \;\times\; 100)$$

Maximum Health refers to the total health the wall has for that given round. This is a discrete number. It can be thought of as the number of hits a wall will take before destruction.

Hits Taken refers to the number of times a cannonball has hit that wall.

As an example,

- **Maximum Health** will be 3 when you begin the game in easy mode.

- Say the player then hits a section twice with a cannonball, **hits taken** will be 2. Thus, the **Section Health** will be $100 \;-\; (\frac{2}{3} \times\; 100) \;=\; 34\%$[6], and the section will go to visual state 1.

Once a wall takes its maximum amount of damage it is destroyed and disappears (visual state 3).

Note: New game rounds sometimes increase the **Maximum Health** value of enemy walls. This means that the **Section Health** of an enemy wall will *decrease*. Because of this, the enemy wall appears to "rebuild" itself when a new round begins. For clarity, a segment of the table from **Section 3.2.1** is shown below.

Difficulty Level	Maximum Health of enemy walls[7]	Number of Rounds
Easy	3-5	5
Medium	4-7	7
Hard	5-9	10

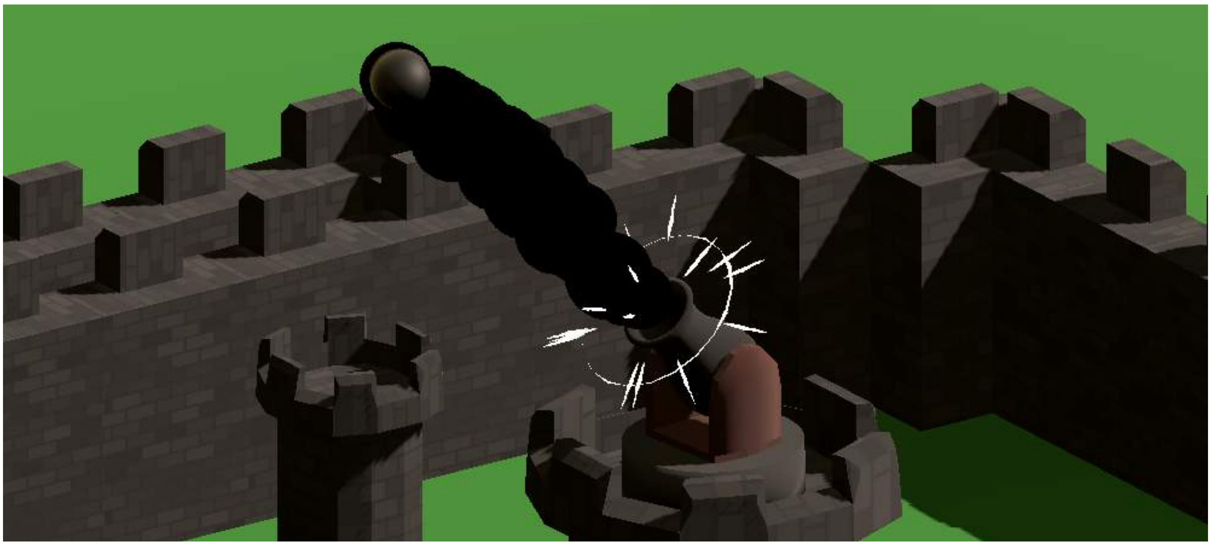
The idea behind this is that while the player is repairing their own walls during the **Repair Phase**, the enemy is doing the same. However, once the player fully destroys an enemy f, it will not rebuild.

It is recommended that you read **Section 3.9.1** to understand what rounds are. To understand how walls are rebuilt by the player during the **Repair Phase**, see **Section 3.6**.

[6] *Section Health* value is floored.

[7] Increases with repetition. E.g., for Easy mode, the values will be 3, 3, 4, 4, 5.

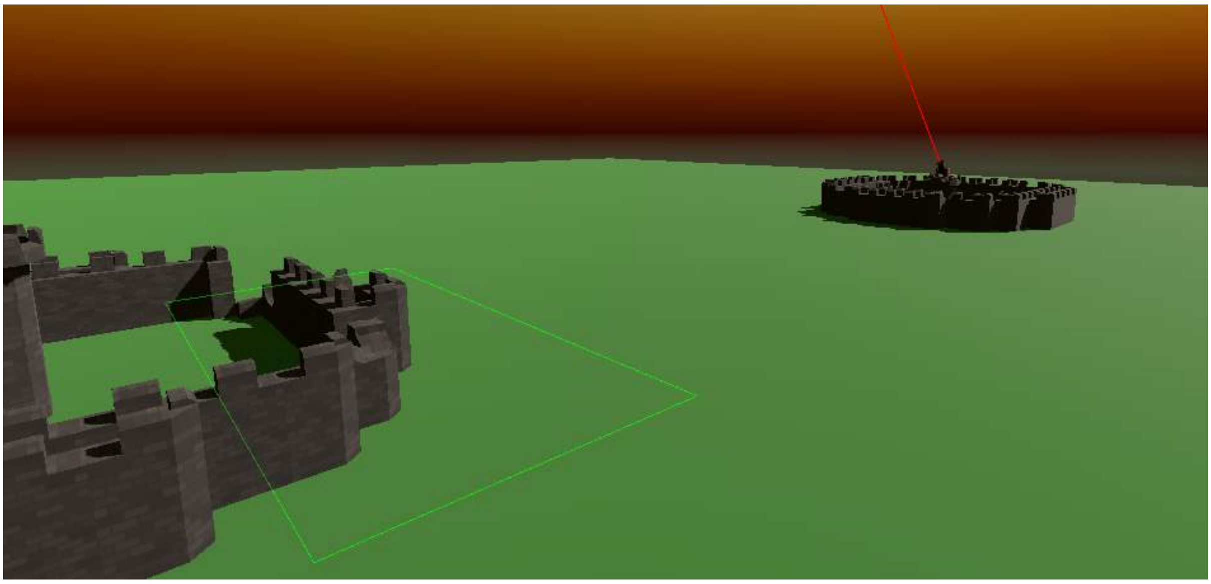
3.4.8 Enemy Cannons



The player is under attack by enemy **cannons**, which are placed within their own castle. These cannons fire cannonballs which can damage the player's walls during the Attack Phase. They fire after waiting for a cooldown period. This cooldown will change depending on the difficulty level chosen by the player, so there is more damage taken at a harder difficulty level. This is outlined in the below table.

Difficulty Level	Cannon cooldown
Easy	6 seconds
Medium	5 seconds
Hard	4 seconds

The cannons are automated to fire at the player's castle. The trajectory of a fired cannonball will randomly fall within a rectangular area around the player's castle (shown below). This means that some shots will hit, and some will miss - emulating the behaviour of a human opponent.



3.5 Speech Commands

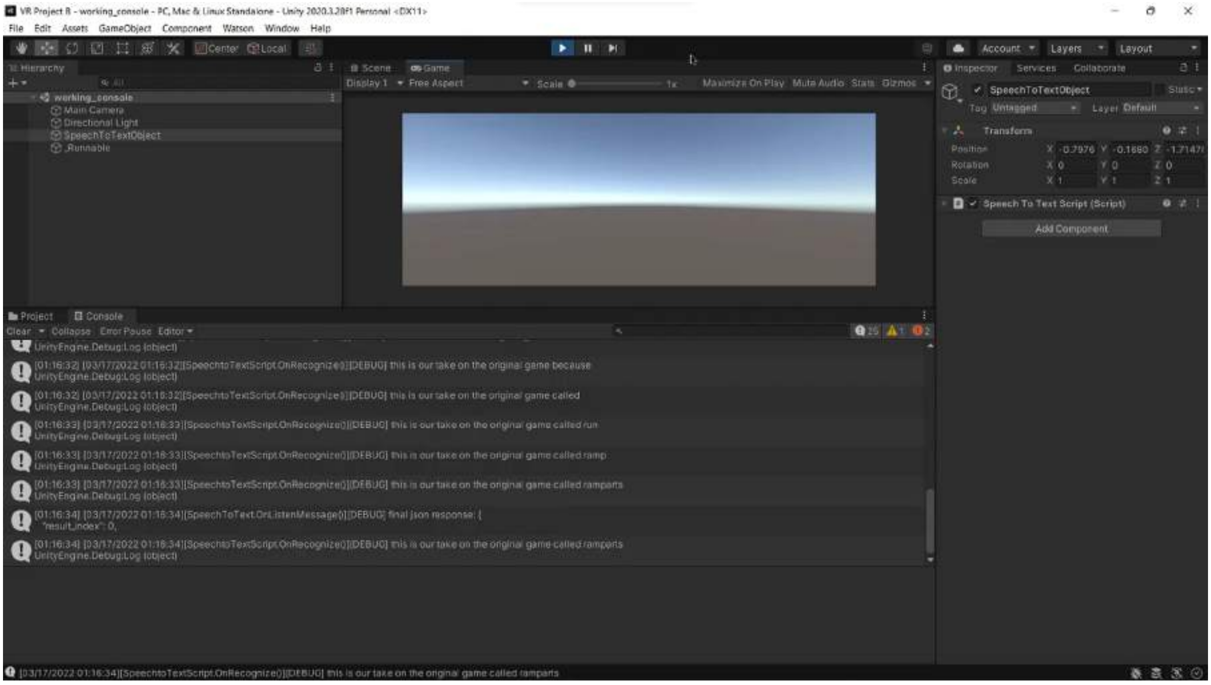
The game incorporates spoken commands from the user, and when specific command words are said, the game will be impacted in some way.
The speech commands are created using two specific IBM tools: IBM Watson Speech To Text and IBM Watson Unity SDK. They work in conjunction with each other to allow the user to interact with the game using their voice; as well as to showcase the capabilities of IBM tools within a VR dimension.

3.5.1 Watson Unity SDK

Watson Unity SDK is the framework which allows the IBM Watson's Speech To Text tool to reside and function within Unity. Not only does it allow for the natural processing of transcription to take place in Unity, but it also allows for the incorporation of "keywords", words that can be preassigned, which when recognised by Unity can be designed to perform some sort of action.

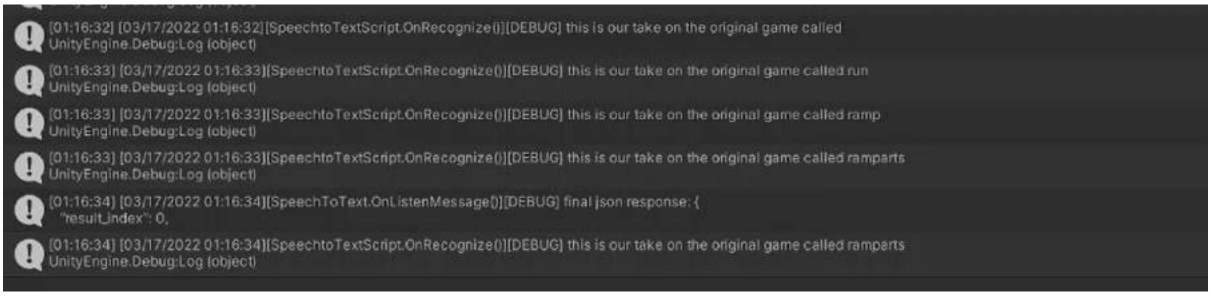
3.5.2 Watson Speech To Text

Watson Speech To Text is an active and accurate transcription tool, converting spoken words into text. The spoken input is received from the microphone on the Oculus Quest 2 headset, which is then registered within the Unity SDK framework, inside Unity. This means that words spoken, when in the game, will be recognised:


VR Project 8 - working_console - PC, Mac & Linux Standalone - Unity 2020.3.28f1 Personal <DX11>
File Edit Assets GameObject Component Watson Window Help
Center Local
Account Layers Layout
Hierarchy
working_console
Main Camera
Directional Light
SpeechToTextObject
_Runnable
Scene Game
Display 1 Free Aspect Scale 1x Maximize On Play Mute Audio Stats Gizmos
Inspector Services Collaborate
SpeechToTextObject Static
Tag Untagged Layer Default
Transform
Position X -0.7976 Y -0.1680 Z -1.7147
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Speech To Text Script (Script)
Add Component
Project Console
Clear Collapse Error Pause Editor
25 1 2

```
UnityEngine.Debug:Log (object)
[01:16:32] [03/17/2022 01:16:32][SpeechtoTextScript.OnRecognize()][DEBUG] this is our take on the original game because
UnityEngine.Debug:Log (object)
[01:16:32] [03/17/2022 01:16:32][SpeechtoTextScript.OnRecognize()][DEBUG] this is our take on the original game called
UnityEngine.Debug:Log (object)
[01:16:33] [03/17/2022 01:16:33][SpeechtoTextScript.OnRecognize()][DEBUG] this is our take on the original game called run
UnityEngine.Debug:Log (object)
[01:16:33] [03/17/2022 01:16:33][SpeechtoTextScript.OnRecognize()][DEBUG] this is our take on the original game called ramp
UnityEngine.Debug:Log (object)
[01:16:33] [03/17/2022 01:16:33][SpeechtoTextScript.OnRecognize()][DEBUG] this is our take on the original game called ramparts
UnityEngine.Debug:Log (object)
[01:16:34] [03/17/2022 01:16:34][SpeechToText.OnListenMessage()][DEBUG] final json response: {
  "result_index": 0,
[01:16:34] [03/17/2022 01:16:34][SpeechtoTextScript.OnRecognize()][DEBUG] this is our take on the original game called ramparts
UnityEngine.Debug:Log (object)

[03/17/2022 01:16:34][SpeechtoTextScript.OnRecognize()][DEBUG] this is our take on the original game called ramparts
```



```
[01:16:32] [03/17/2022 01:16:32][SpeechtoTextScript.OnRecognize()][DEBUG] this is our take on the original game called
UnityEngine.Debug:Log (object)
[01:16:33] [03/17/2022 01:16:33][SpeechtoTextScript.OnRecognize()][DEBUG] this is our take on the original game called run
UnityEngine.Debug:Log (object)
[01:16:33] [03/17/2022 01:16:33][SpeechtoTextScript.OnRecognize()][DEBUG] this is our take on the original game called ramp
UnityEngine.Debug:Log (object)
[01:16:33] [03/17/2022 01:16:33][SpeechtoTextScript.OnRecognize()][DEBUG] this is our take on the original game called ramparts
UnityEngine.Debug:Log (object)
[01:16:34] [03/17/2022 01:16:34][SpeechToText.OnListenMessage()][DEBUG] final json response: {
  "result_index": 0,
[01:16:34] [03/17/2022 01:16:34][SpeechtoTextScript.OnRecognize()][DEBUG] this is our take on the original game called ramparts
UnityEngine.Debug:Log (object)
```

The example above demonstrates the ability of words being able to be registered by the Watson Speech To Text tool. As you can see, multiple responses are recorded within Unity; the purpose of this will be explained next.

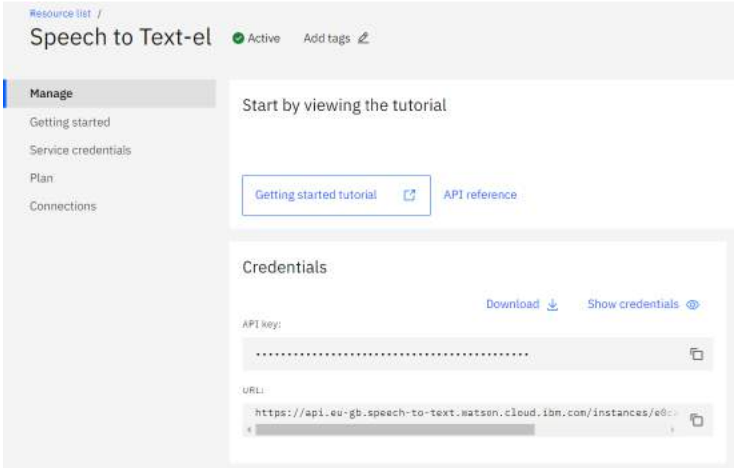
The next stage following this is assigning keywords. These keywords, when registered by the Watson Unity SDK, allow for the actions to take place if actions are tied to objects. For example, if an action is tied to the keyword "Hello", when the word "Hello" is said, only then is the action able to be performed.

The multiple responses that were just mentioned allows for more accurate performance. It is a failsafe feature included so it is less likely that saying a keyword is not misheard (by Watson) as another word. Watson Speech To Text converts the spoken input into three possibilities for each word or sentence, each phrase with their own likelihood/accuracy rating for the words spoken. Instead of Watson simply recognising the words with the greatest likelihood, all three possibilities are considered. As a result there is a greatly reduced chance that saying the word "Hello" is not misheard as the word "Yellow" for example.

3.5.3 Setup

The process of setting up Watson Unity SDK is difficult, but is aided by the use of a specific Github Repository: https://github.com/watson-developer-cloud/unity-sdk . The SDK has already been deployed into this project, so you do not need to set it up yourself.

To use Watson Speech To Text AI, you need to access material from the IBM Cloud website. You must use the generated API key to use the functionality of Watson; the key is found on the website as shown:



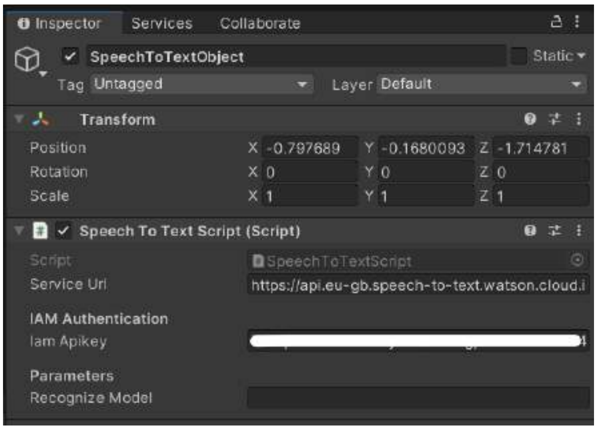
Start by viewing the tutorial

Credentials

API key:
··

URL:
https://api.eu-gb.speech-to-text.watson.cloud.ibm.com/instances/e0

This needs to be added directly into Unity when prompted:



☑ SpeechToTextObject
Tag Untagged
Layer Default

	X	Y	Z
Position	-0.797689	-0.1680093	-1.714781
Rotation	0	0	0
Scale	1	1	1

Script SpeechToTextScript
Service Url https://api.eu-gb.speech-to-text.watson.cloud.i

IAM Authentication
Iam Apikey [redacted]

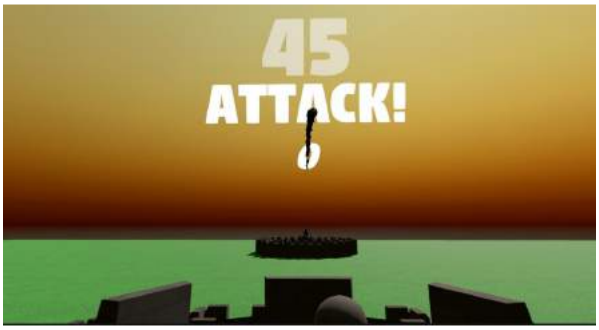
Parameters
Recognize Model

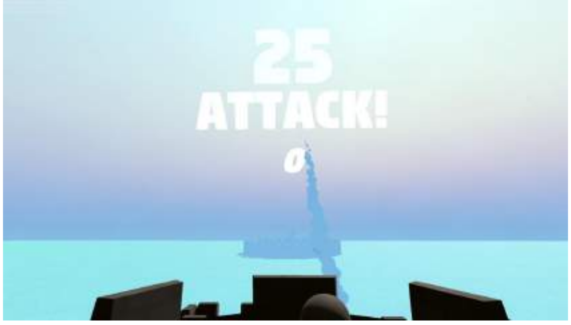
Once setup is complete, you will need to begin creating scripts that allow for the desired functionality of creating the power-ups.

3.5.4 Features and Actions

Watson Speech To Text has been used to allow the user to perform power-ups; more specifically for the purpose of giving the user an advantage within the game.
These power-ups can only be used at certain times, and strictly during the Attack phase of the game; there will be a pop-up message as to when power-ups are available.

An example of an in-game power-up can be performed by saying "raise the shields" out loud during the Attack phase. This invokes a shield that surrounds the user's castle and prevents any oncoming damage from enemy cannonballs. The shield will only last 10 seconds.





Here is a side-by-side of what the power-up does.

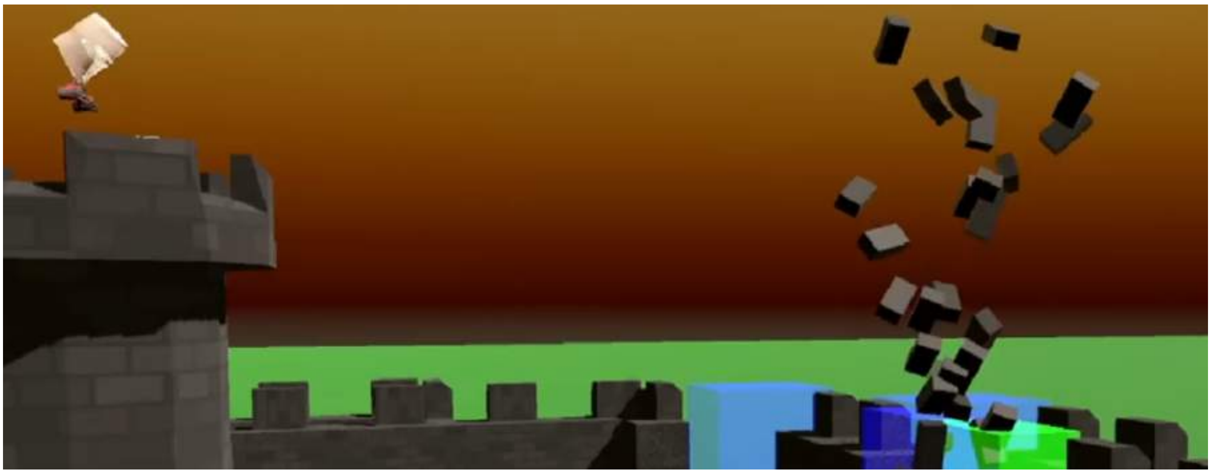
The tutorial (mentioned in **Section 3.2.3**) explains to the user what power-ups are available and how power-ups can be enabled, and demonstrates where and when they can be used.
The power-ups available are currently just "Raise the shields", but further power-ups such as "Double Damage" (double points) and "Bullseye" (direct aim) will be included in the near future.

Power-ups are invoked in many ways; ranging from reaching a certain score, to depending on the difficulty level of the game.
The number of times a power-up can be used depends on the difficulty level chosen by the user; a higher difficulty level would mean less power-up opportunities would be available as the power-ups effectively make the game easier to play (e.g. having shields up means user's walls are damaged less therefore the user will need to perform less squats).

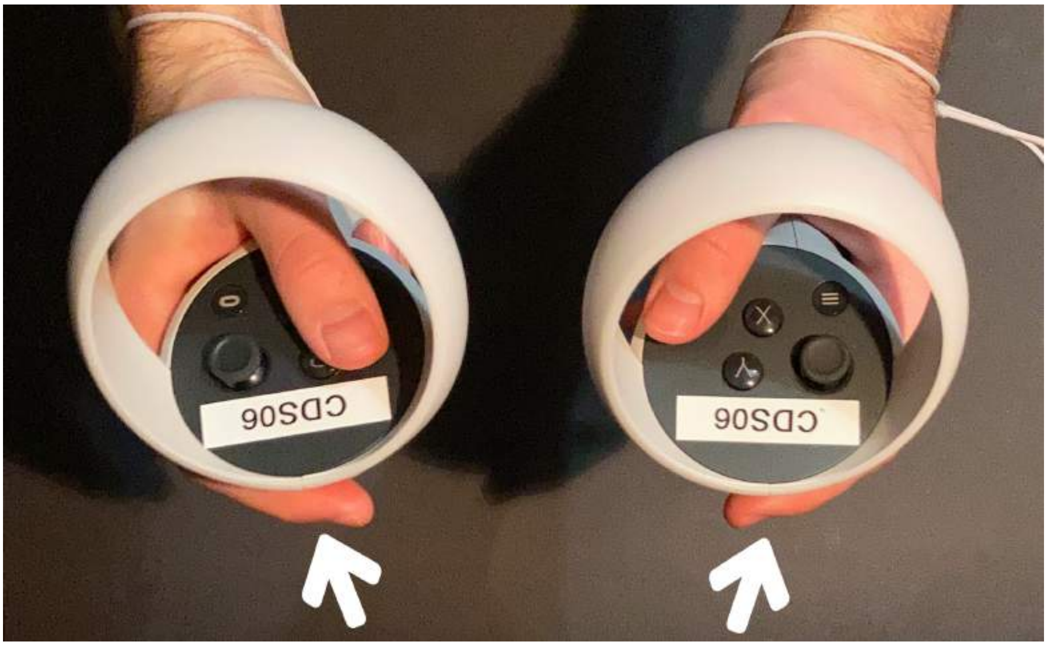
For example, looking at a Medium difficulty setting, the "Raise the shields" power-up is only able to be used three times in total; and is only available if you reach a minimum score of 300. An Easy difficulty setting would allow the "Raise the shields" command to be used 5 times no matter the score; Hard difficulty setting would only allow the user to use it once after a score of 500 is reached.

3.6 Repair Phase

In the repair phase, players have 30 seconds to try and repair any damaged walls in their castle from the attack phase. Players perform the repair by choosing a broken blue wall and performing a squat in order to rebuild it.

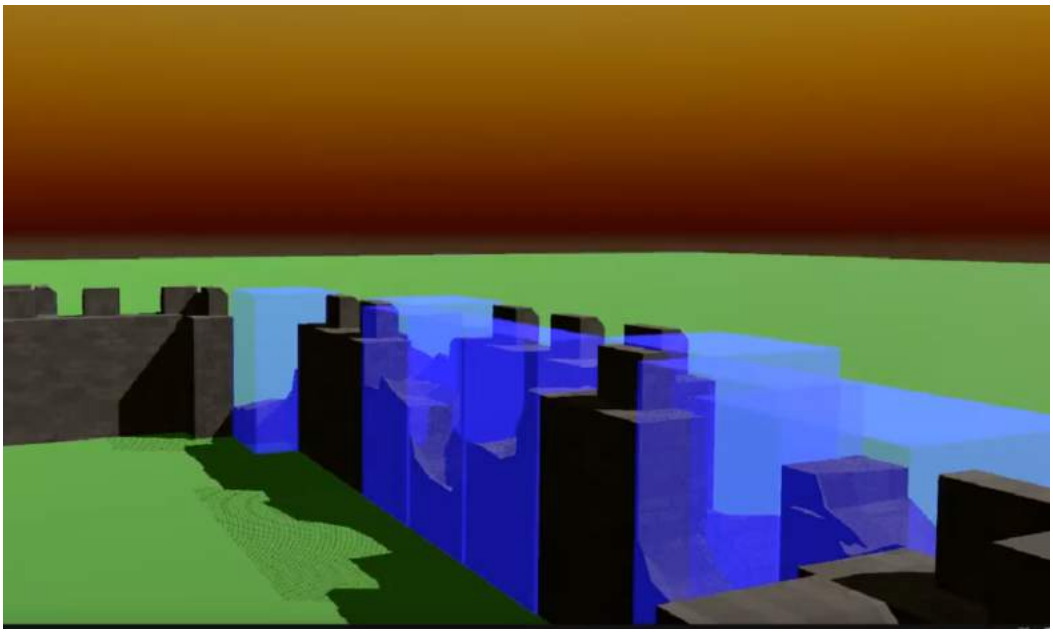


3.6.1 Repairing Walls

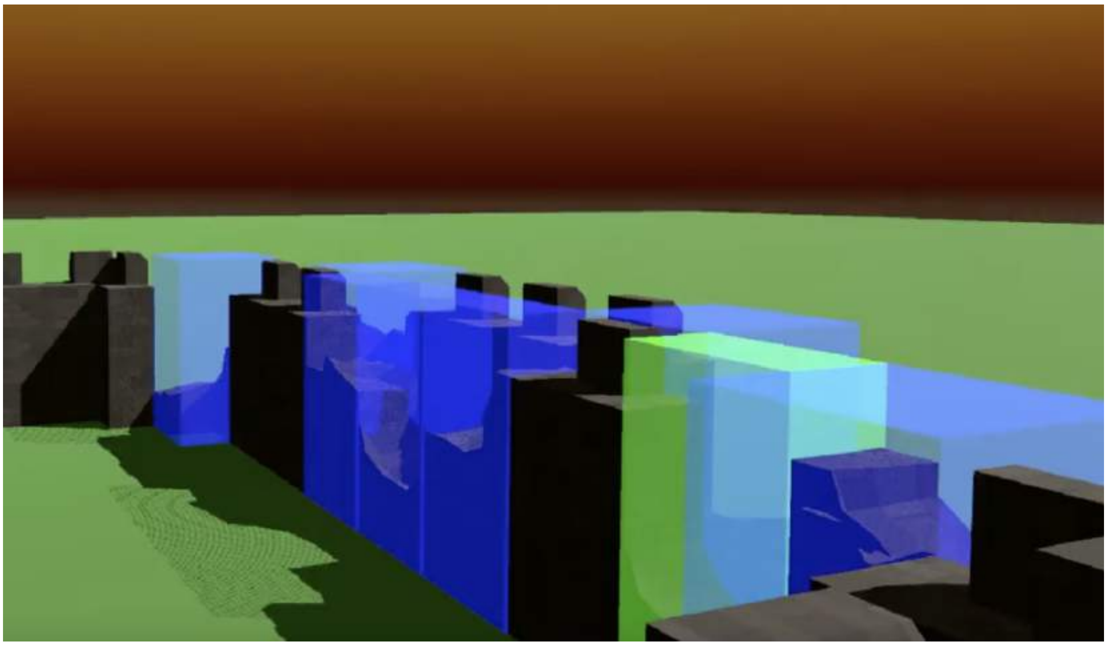


You are able to select a wall piece by facing it and holding both controller triggers at once (indicated above).

The wall pieces that need to be repaired are highlighted in a translucent blue colour when the Repair Phase begins:



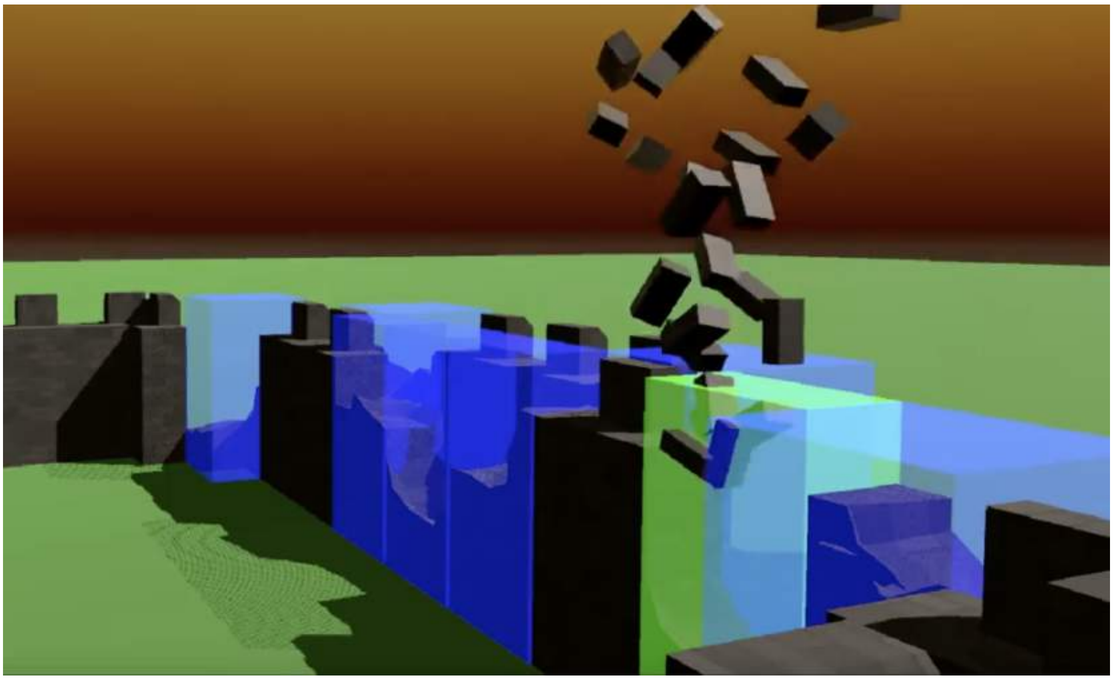
To repair a wall piece, you must first face a blue highlighted wall piece, which will turn a translucent green colour to show that you have locked onto it for repairing:



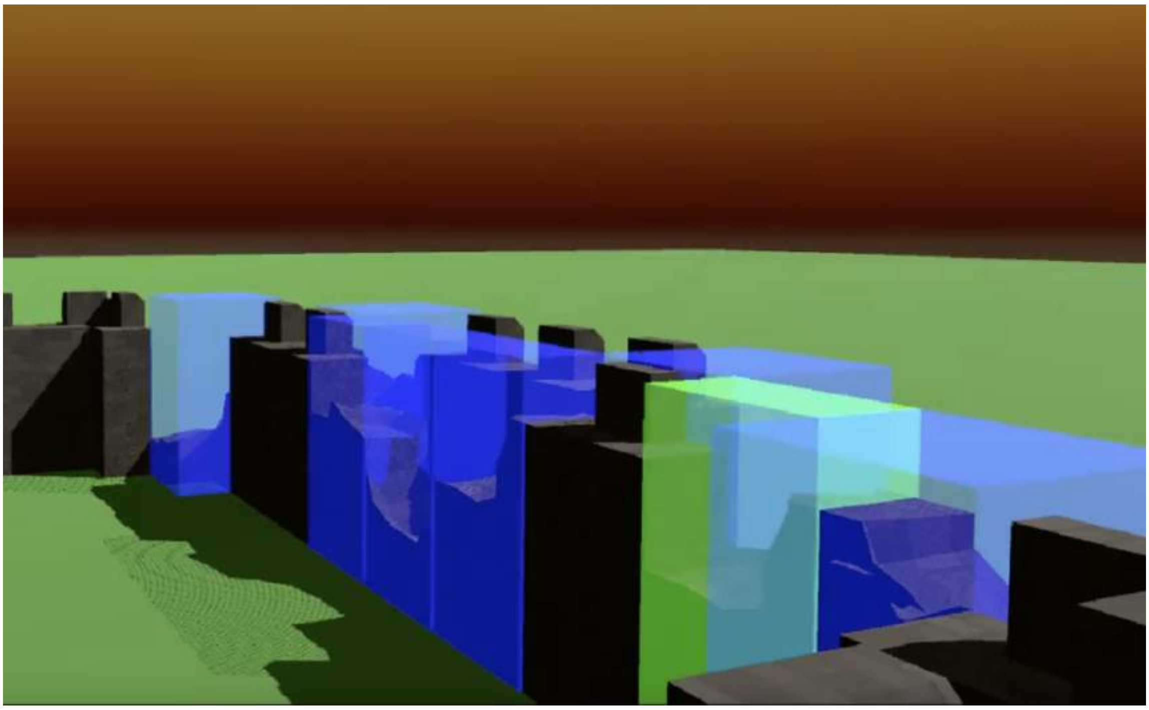
Then, hold down the controller triggers. You are ready to perform your squat.

You must hold the triggers from the moment you start your squat, to the lowest point of the squat. (See **Section 3.4.7** for detail on wall destruction states.)

As you squat, an animation of bricks should appear, showing that you are successfully repairing the wall:



At the lowest point of your squat, you must release the triggers. The wall has now been repaired once:



Then, you should stand up straight, ready to perform the next squat repair.

You will often have to perform multiple squats on one wall piece to fully repair it. The number of squats you have to make depends on the Maximum Health of the wall piece for that round, and how many hits it has taken. To understand Maximum Health and how it is affected by rounds, see **Section 3.4.7**, and to understand what rounds are, see **Section 3.9.1**.

Note: If your squats are not registering, or you feel you have to squat too deeply to repair the walls, you may need to recalibrate your height. To do so, press the “B” button on either controller. For more detail see **Section 3.1.1.**

The program simply checks that you squat to around half your height. The value of your height is taken from the height calibration.

3.6.2 Squat Guidance

Here is professional advice on how to perform the optimum squat taken from: https://subiaco-physiotherapy.com.au/four-key-components-to-a-perfect-squat/

1. Start with your hips
   The first movement of a squat should always be pushing your hips back behind you.

2. Knees over ankles
   In a perfect squat, your shins should stay near vertical and ankles should move very little.
3. Keep your low back neutral
   This is really important! Your low back should remain neutral the whole way through your squat (i.e. with a very light curve in it).
   If you notice that your low back starts to slump towards the bottom of the movement, then come back up.

4. Maintain width between your knees
   Ideally, your knees should remain in line with your second and third toes as you lower down into the squat position.

3.7 Rest Phase

The **Rest Phase** is a 20 second period of recovery within the game. The reason for this phase is to allow people to rest after each ‘set’ of exercises like they would in a typical workout. After the Rest Phase the user goes straight back into the Attack Phase until the final round when your final score is given to you and you can see if you have made it onto the leaderboard.

If you feel light headed or Queasy during a rest phase we recommend pausing or quitting the game and having a drink of water or a sit down until you feel more comfortable. Then make a decision whether or not to continue with the game or not.

3.8 Pausing and Quitting

You can pause the game at any time by clicking the “A” button on either controller. This will reload the main menu as seen in **Section 3.2**, with the only differences being the “Play” button is replaced with “Resume”, the “Difficulty” button is omitted, and there is a “Quit” button. Clicking “Quit” will end your gameplay, and take you back to the main menu.

Note that the difficulty of a game can only be changed after quitting the gameplay, and returning to the main menu.

- If you feel unwell, pause the game.
- If you are too tired to play the round, pause the game.

If you pause the game it will not count toward your high score as you had additional rest time but you can still enjoy and benefit from the game.

If you want to quit the game at any point you can just simply press the home button on the controller. If you quit the next time you restart the game you will just be taken to the main menu screen. Any current game progress will be lost.

3.9 Global Features

3.9.1 Rounds

When you press play, the game will begin and will play for **5 rounds**. A round consists of one cycle through the Attack, Repair and Rest phases, and takes 110 seconds in total. After the final round concludes, you are returned to the menu screen.

3.9.2 Points

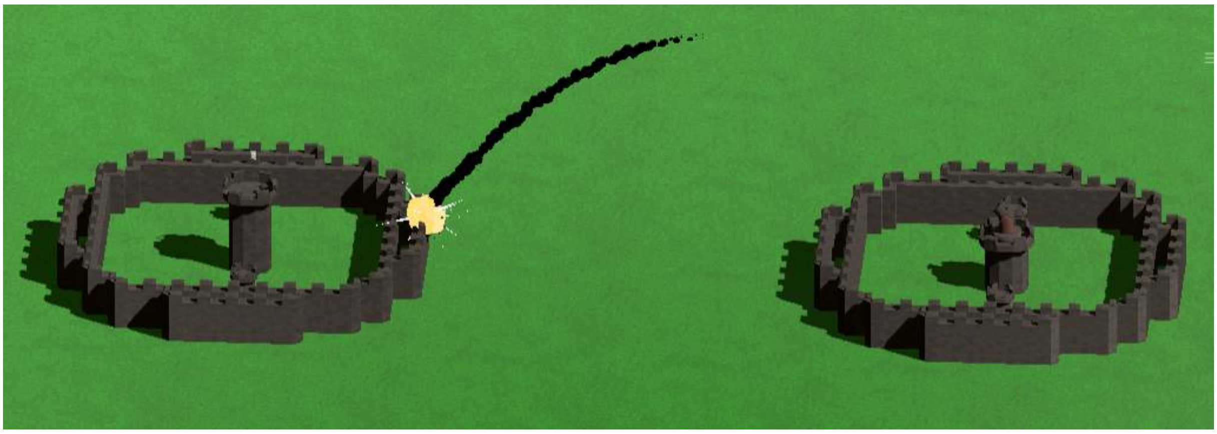
Your performance in the game is scored using a points system, which rewards you for exercising. Try and beat your high score each time you play!

During the Attack Phase, successfully **hitting a wall** with a thrown cannonball earns you 20 points, and **destroying a section** earns 15 points.

While in the Repair Phase, each **successful repair** you perform by squatting earns 10 points. Leaving a damaged section of wall **unrepaired** when the time runs out incurs a penalty of 5 points.

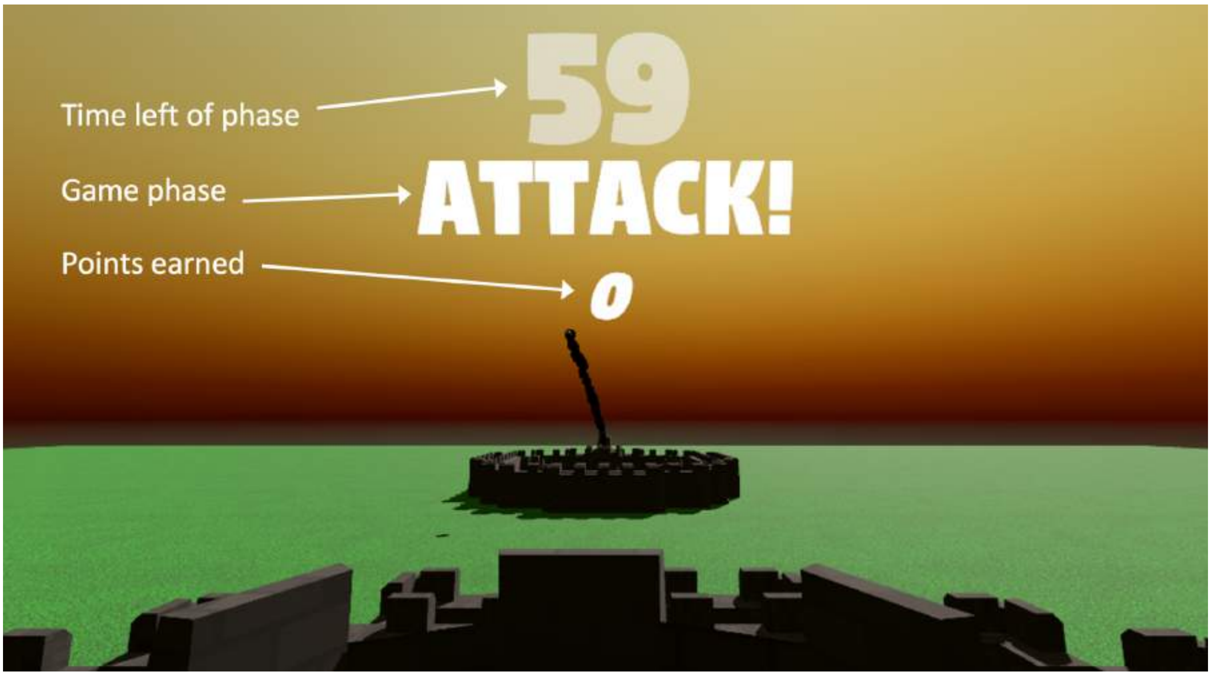
Action	**Points**
Damage the enemy wall with a cannonball	+20
Entirely destroy an enemy wall	+15
Repair a section of wall	+10
Wall still damaged after Repair Phase	-5

3.9.3 Environment



When you spawn into the beginning of the game, you will be facing the enemy castle and should have a clear view over your castle. If your view isn't clear, you may need to recalibrate your height. To do so, press the "B" button on either controller. For more detail see **Section 3.1.1.**

Below is an annotated Point-of-View of what you will see, when you spawn at the start of the attack phase.



As you progress into new phases, the timer at the top will reset, while the game phase title will update. A total of all the points you have earned so far is kept at the bottom.

4. Maintenance & Troubleshooting

4.1 Common Issues

4.1.1 Headset Not Responsive

Sometimes the VR headset does not correctly connect to the application when you run it. Unfortunately owing to the variety of headsets and means of linking to the PC, it can be difficult to determine where the problem lies.

Here are a few common issues and fixes:

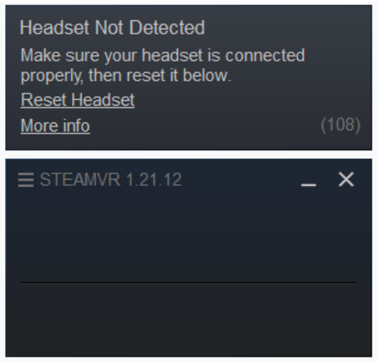
Cable Connection

The cable connecting the headset to the PC can sometimes come loose at one end. Alternatively, the connection through the cable encounters an error and stops behaving correctly. Unplugging and reconnecting the cable is often enough to restore the connection.

4.1.2 VR Platform Issues

SteamVR

SteamVR may need rebooting if it does not detect your headset. This often occurs when you connect the headset when SteamVR is already running. (SteamVR itself has certain troubleshooting systems in place.)



Headset Not Detected

Make sure your headset is connected properly, then reset it below.

Oculus Link

The link to your Quest 2 headset may fail, which loses the connection with your PC and SteamVR. If this happens you will most likely need to restart Oculus Link (following the steps in **Section 2 Setup & Requirements**), then restart SteamVR.

Graphics Drivers

If no other solutions fix the headset’s issues, you may want to try updating your computer’s video drivers if you have not done so in a while. Please be aware that very new drivers are not always supported by SteamVR until a day or so after they release - sometimes it is better to remain on less recent but stabler driver versions rather than updating.

4.1.3 Play Space Placement

Most VR headsets require that you define a play area when you first use them. While there are often minimum dimensions you have to meet, ensure you give yourself enough room to play in (2x2m as specified in **Section 2 Setup & and Requirements**).

If the game begins and you are not positioned correctly (in the centre of the castle tower, with the cannonball platform in front of you), you may need to reset your game position. On the Valve Index this is achieved by holding down the menu button on the right controller. On Quest 2, you must hold down the menu button and the trigger.

Please consult your headset’s own instructions on how to recalibrate the play space if necessary.

4.1.4 VR Hardware Maintenance

It is important to make sure your headset and controllers are charged and do not have low batteries. Low power can cause the equipment to not behave as expected - it may lose tracking or not respond reliably to inputs.

4.1.5 Player Action Troubleshooting

The key actions the player performs may not be working as expected. If squatting does not repair the walls, or your swings are unexpectedly treated as incorrect, there is most likely an issue with your height calibration. Go back to the menu and **repeat the height calibration**, making sure you are standing straight to provide an accurate height.

4.2 Unexpected Errors

The software may run incorrectly if its data is corrupt from a bad download, or if the PC is underpowered or overwhelmed. In this case, it is recommended to either **restart or redownload the application**, or to **restart your computer** and give it time to recover.

Virtual Reality applications are moderately taxing for most computers, so overheating after long periods of play is a known issue. This can cause the software to underperform or lose tracking of the headset. Leave time for your PC to cool off, or ensure you have suitable cooling or ventilation systems in place.

5. Appendix - Class Functionality

This section details all of the new C# programs created for the project, and how they relate to one another and the game as described in this manual.

Program File	Description	Related Section(s)	Dependencies
CannonBot.cs	Controls enemy cannon behaviour.	3.4.8 (Enemy Cannons)	Projectile.cs
GameManager.cs	Manages round timing, switching between Phases.	3.3 (Game Stages) 3.4 (Attack Phase) 3.5 (Repair Phase) 3.6 (Rest Phase) 3.9 (Global Features)	CannonBot.cs RepairSystem.cs ScoreSystem.cs
KillParts.cs	Cleans up particle effects.	3.4.8 (Enemy Cannons)	
MenuScript.cs	Manages main menu functionality.	3.2 (Navigating the Menu)	
MusicSliders.cs	Controls music sliders, updates game settings correspondingly.	3.2.2 (Music and SFX)	
Projectile.cs	Attached to cannonballs used by player and enemy. Deals damage, increases score on hit.	3.4.7 (Wall Destruction) 3.4.8 (Enemy Cannons) 3.9.2 (Points)	Wall.cs ScoreSystem.cs
RepairSystem.cs	Handles Repair Phase mechanics, squat detection/analysis. Player input while repairing.	3.6 (Repair Phase)	Wall.cs ScoreSystem.cs
SpeechToTextScript.cs	Converts words spoken by the user into text, and recognises the keywords for the power-ups	3.5 (Speech Commands)	
ScoreSystem.cs	Handles points during rounds, updates total and calculates high scores.	3.9.2 (Points)	
Spawner.cs	Instantiates Cannonball projectiles for the player.	3.4.1 (Picking Up Cannonballs)	
Wall.cs	Stores wall section parameters, damage states. Updates visual state on damage.	3.4.6 (Wall Initialisation) 3.4.7 (Wall Destruction) 3.6.1 (Repairing Walls) 3.9.2 (Points)	ScoreSystem.cs
WallGen.cs	Procedurally places wall sections at game start.	3.4.6 (Wall Initialisation)	Wall.cs