

Gesture Based ui project

Year 4 – Software Development



Owen Kelly & Oskar ciebien

G00366614 & G00369579

**GitHub Repository:** [**[Click]**](https://github.com/Oskar-Ciebien/Gesture_Based_UI_Project)

Table of Contents

[Purpose of the application 2](#_Toc102609139)

[Main Menu 2](#_Toc102609140)

[Game Scene 2](#_Toc102609141)

[Levels 2](#_Toc102609142)

[Pause Menu 2](#_Toc102609143)

[Death Menu 3](#_Toc102609144)

[Gestures identified as appropriate for this application 4](#_Toc102609145)

[Gesture Ideas 4](#_Toc102609146)

[Gestures Used 4](#_Toc102609147)

[Hardware used in creating the application 5](#_Toc102609148)

[Android Device – Mobile Phone 5](#_Toc102609149)

[PC – Unity Editor / Development Device 5](#_Toc102609150)

[Architecture for the solution 6](#_Toc102609151)

[Libraries Used in the Project 6](#_Toc102609152)

[Conclusions & Recommendations 7](#_Toc102609153)

[What has been achieved? 7](#_Toc102609154)

[What has not been achieved? 7](#_Toc102609155)

[What have we liked and learned? 7](#_Toc102609156)

[Conclusion – Oskar 7](#_Toc102609157)

[Conclusion – Owen 7](#_Toc102609158)

[References 8](#_Toc102609159)

# Purpose of the application

This application utilises the user’s mobile phone to display an augmented reality experience of a custom variant of the classic 70’s arcade game Breakout. The application accesses hardware from the mobile device to project the game so that it may be experienced in any setting and completely independent from any external hardware or software. Each component of the game and subsequently the application was designed with these principles in mind. The following descriptions detail what was done to adhere to the application requirements to a satisfactory level.

## Main Menu

On the Main Menu screen, the player is greeted with stunning particle effects, colour matching elements and logo. They have two options to choose from.

* Play – Will move the player to the next screen which is the game scene.
* Quit – Will exit out of the game.

We have focused on the game to be very appealing to the player, and since this is an old game which most people have played. Our decision was to stick with a neon theme for the entire game.

While developing the application we were planning on adding voice control to our menus, in order to add more gestures to the project. But unfortunately, some features of the game took longer than expected and we were forced to abort the idea.

## Game Scene

The game scene is responsible for the whole gameplay. Firstly, the player is asked to scan the image of a maze which is available on the project’s GitHub repository. Once the image is focused, the text asking to scan the image disappears and the game board including the ball, paddle, blocks, and borders appears.

We have chosen Vuforia so that the player will be able to look up and down in order to see the blocks, as the board is made bigger, so that the player would not be able to see it in full size on the screen.

## Levels

We have decided to stick with an endless level type for this project. The score represents the seconds that have passed since the start of the game.

The levels keep changing [ ADD BIT HERE ABOUT LEVELS ]

## Pause Menu

The pause menu is accessible on the Game Scene at any point and time. A big button in the top right corner with two bars is used to pause the game and to display the pause menu.

The pause menu offers a few options. Those are:

* Controlling the volume.
* Resuming the game.
* Going back to the main menu.

As on the main menu, there are some engaging particles to once again make it more appealing to the player.

## Death Menu

This menu is only visitable after the player has used up all of their lives, where at the start of a new game they receive three lives.

Lives can also be acquired through collectibles, but the maximum amount is three, which means, additional lives collected through collectibles will not make a difference if the lives are already at three.

The death menu gives the player information on their score, as well as two options to pick from. The two options are the following:

* Restart – Which will transfer the player back to the game scene, restart all of the player’s lives and score.
* Main menu – This option will bring the player back to the main menu, where they can decide if they want to start a new game or quit.

## Testing Plan

We have tested each feature after implementing it and the other person tested the feature just before it was their turn to add a new feature to the game. This way we were sure that the feature was free of bugs or if it needed some improvements.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

# Gestures identified as appropriate for this application

## Gesture Ideas

Our game idea allows for many gestures which could have been used in the game in different ways. Unfortunately, we could not have added all of them to our project.

The gesture ideas that we had during the planning phase of this project are as follows:

* Tap – For buttons, navigation, game control.
* Swipe – Pausing game.
* Swipe with another image – Moving the paddle left or right.
* Tilting / Freehand – Allowing the player to move the camera in any direction, to see the game board better at different angles, for example allowing the player to see where the blocks and the ball is headed.
* Voice control – An alternative to some taps, for menu navigation and settings.

## Gestures Used

There have been many gestures to pick from, but we decided to go with the following gestures as they suited our game the most and were considered by us as the most comfortable for this type of game:

* Tap
* Swipe
* Tilt
* Freehand Movement

We have not gone with Voice control for the menus and settings, as we believed it would not been an adequate gesture or control system, as the user will have to use the swipe gestures on the phone anyway, and they will have to keep the phone in front of the maze picture to keep synced and to start the game.

# Hardware used in creating the application

## Android Device – Mobile Phone

We have used our Mobile Phones as testing devices with the use of Unity Remote 5 application, which is free to be downloaded on Google Play Store.

After every major change in the game, we have built the game onto our android devices to feel the game better than while testing the game. More as a black box testing, which helped in familiarising with the game and coming up with more ideas and fixes for certain parts of the project.

## PC – Unity Editor / Development Device

We have used the Unity Editor sometimes during longer developing periods, without the use of Vuforia components and instead using Main Camera.

Unity Editor allowed us to save a lot of time on testing and we used it only to test out simple features of the game such as, checking if the game changes scenes correctly and if the score and lives count are displayed as expected.

# Architecture for the solution

## Libraries Used in the Project

Most of the libraries that we have used in our project are the libraries that we were familiar with as they have been covered in the Gesture Based UI Development and Mobile Applications Development modules. The libraries that we have used in this project are the following:

* TMPro – Which is a very important library to use, when trying to add high quality text design into a project. It provides many features and formatting options as well as it is very simple to use. We have used this library specifically to enhance the appearance of our text fields and buttons. We were able to set custom fonts, and style them a lot better than with the standard Text/UI library in Unity.
* SceneManagement – This is a library which allows us to most importantly change from scene to scene, for example when a player decides to start the game and presses on the Play button on the Main Menu scene, the game transfers the player to the Game Scene. We also use this library to check if we are on the right scene, depending on the type of script that we use it in. There are many more features, which this library provides, such as getting the number of currently loaded scenes, creating scenes at runtime and many more.
* Vuforia – Is a Software Development Kit (SDK), used for creating Augmented Reality Applications or Games. It recognises images and objects, and therefore can have many different applications or uses, such as interacting with the real world. In this project we have used Vuforia to connect with our Breakout game, through an image, and therefore allowing us to have a view of the game, while seeing our surroundings through the camera. We use Vuforia to have a better view of the game board and on what is exactly happening in the game.

## Communications

We stayed up to date, by having multiple meetings every day. We have worked on Pair Programming to help each other out with some features as well as.

# Conclusions & Recommendations

## What has been achieved?

As a team we were able to finish this project to a stage where we have been happy. There is always room for improvement, but unfortunately other deadlines and the time was not on our side and some features took us longer than expected, as well as Unity and git merging causes problems, therefore we were forced to allow the other team member to work on their feature first. But in that time the other person worked on the wiki, research or the documentation.

## What has not been achieved?

Through the whole development process of this project. We have run into issues where we were able to fix some of them, but there’s unfortunately some that we did not get to fix. These include:

* The collisions with the paddle and the left and right border do not work as we have expected them to work. They differ from gameplay to gameplay as the connections depend on the Vuforia image focus. If Vuforia is not connected properly it does not give us the right distances. But the distances and coordinates are correct, as they have been tested without Vuforia and sometimes it does connect as expected and the game works fine.

## What have we liked and learned?

With every grouped project, comes some advantages and disadvantages.

## Extras

For our extras part of this project, we have decided to make wiki pages on certain topics on the project’s GitHub repository. The player can have a read through instructions on how to play the game, how to contact us, general information about this game etc. The wiki is available [**[here]**](https://github.com/Oskar-Ciebien/Gesture_Based_UI_Project/wiki).

## Conclusion – Oskar

## Conclusion – Owen

# References

TMPro   
<https://learn.unity.com/tutorial/working-with-textmesh-pro#5f86410eedbc2a00249a4925>

SceneManagement  
<https://docs.unity3d.com/ScriptReference/SceneManagement.SceneManager.html>

Vuforia  
<https://developer.vuforia.com/support>