

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](#_Toc102411869)

[Main Menu 2](#_Toc102411870)

[Game Scenes 2](#_Toc102411871)

[Levels 2](#_Toc102411872)

[Pause Menu 2](#_Toc102411873)

[Death Menu 2](#_Toc102411874)

[Gestures identified as appropriate for this application 3](#_Toc102411875)

[Gesture Ideas 3](#_Toc102411876)

[Gestures Used 3](#_Toc102411877)

[Hardware used in creating the application 4](#_Toc102411878)

[Android Device – Mobile Phone 4](#_Toc102411879)

[PC – Unity Editor / Development Device 4](#_Toc102411880)

[Architecture for the solution 5](#_Toc102411881)

[Libraries Used in the Project 5](#_Toc102411882)

[Conclusions & Recommendations 6](#_Toc102411883)

[What has been achieved? 6](#_Toc102411884)

[What has not been achieved? 6](#_Toc102411885)

[What have we liked and learned? 6](#_Toc102411886)

[Conclusion – Oskar 6](#_Toc102411887)

[Conclusion – Owen 6](#_Toc102411888)

[References 7](#_Toc102411889)

# Purpose of the application

## Main Menu

## Game Scenes

## Levels

## Pause Menu

## Death Menu

# 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.

## 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

# 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.

# Conclusions & Recommendations

## What has been achieved?

## What has not been achieved?

## What have we liked and learned?

## 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>