|  |
| --- |
| Dublin Institute of Technology |
| Augmented Reality Development Project |
| Produced by Jamie Mulvaney |
| Document Revision Version: **4** | Document Revision Date: 14.03.2019 |

|  |
| --- |
|  |

Contents

[1. Purpose 2](#_Toc3495559)

[1.1 Objectives 2](#_Toc3495560)

[2. Process Description 2](#_Toc3495561)

[2.1 In Scope 2](#_Toc3495562)

[2.2 Out of Scope 2](#_Toc3495563)

[2.3 Targeted & Purposed Specifications 3](#_Toc3495564)

[2.4 The [Purposed] Process Flow 3](#_Toc3495565)

[2.4.1 Blind Overlay Flow 3](#_Toc3495566)

[2.4.2 Curtain Overlay Flow 4](#_Toc3495567)

[2.5 [Purposed] Process Flow Mock-ups 5](#_Toc3495568)

[2.5.1 Blinds Process Flow Mock-ups 5](#_Toc3495569)

[2.5.2 Curtains Process Flow Mock-ups 10](#_Toc3495570)

[3. Phase One – Features 14](#_Toc3495571)

[3.1 Features Explained 14](#_Toc3495572)

[4. Phase One - Build 16](#_Toc3495573)

[4.1 Unity Project Explained (Internal Use) 16](#_Toc3495574)

[5. Appendix 19](#_Toc3495575)

[5.1 Ground Plane 19](#_Toc3495576)

[5.2 Logo 19](#_Toc3495577)

# 1. Purpose

The development of an augmented reality application on behalf of DIT (Dublin Institute of Technology) for [INSERT COMPANY NAME].

The purpose of this augmented reality application is to allow potential customers of [INSERT COMPANY NAME] the convenience / ability to overlay and then view an [X] number of window blind and or curtain designs using a mobile device.

The aforementioned augmented reality application is intended to be MVP (Minimal Viable Product) and will be developed over an eight-week period which for the purpose of this document will be referred to as phase one going forward. All development files and the final version of the MVP will be handed over to DIT at the end of phase one.

[Note: Exact technical details for the MVP will be noted in the [below section](#_2.3_Targeted_&), such as targeted mobile platform and device]

## 1.1 Objectives

* Allow a user to setup a designated window blind and or curtain area using a mobile device that is capable of running an augmented reality application once installed.
* Once a designated area has been established and confirmed by the user, the application should overlay a designated first product example, which the user will then be able to view in an augmented reality setting.
* Allow the user to change between [X] number of product examples, the application will adjust and overlay the new example once / if it is selected.
* There should be little to no time visible to the user between loading any new examples. All product examples will be designed and constrained to the same designated area.

[Note: These objectives may be adjusted based on feedback from the interested parties and from any testing carried out]

# 2. Process Description

## 2.1 In Scope

Features that are within scope of the phase one development cycle allowed for this project:

* One augmented reality application for the specified targeted platform. This augmented reality application will require the user to designate the exact area they wish to overlay product examples over.
  + The user must then remain within that area in order to provide the best augmented reality experience., otherwise a reset of the designated area may be needed.
* The application will contain an [X] number of product examples for the user to select from. The application will then update and overlay the chosen example within the bounds of designated area for the user to view.

## 2.2 Out of Scope

Features that are outside the scope of the phase one development cycle allowed for this project:

* Automatic detection of the window area by the application through use of:
  + An image database
  + Sensor targeting
  + Reflection or light detection algorithm
* Automatic readjustment of the product example on the overlaid area, if the user adjusts their position or the device position after designating the area.
* Detailed descriptions of products and ability to order / review from within the application itself.

[Note: Depending on time and resources certain out of scope aspects may be explored.]

## 2.3 Targeted & Purposed Specifications

Targeted hardware and software specifications:

* **Devices**:
  + Mobile Phone (Built In-Camera)
* **Operating System**:
  + Latest Version of the Android OS
* **Augmented Reality Software:**
  + Vuforia Suite
* **Development Engine**:
  + Unity *(Engine Version: 2018.3.2f1)*

Purposed application specifications:

[Discussion needed on number of product examples, screens etc that the application should contain]

## 2.4 The [Purposed] Process Flow

Walkthrough of the process flow based on information taken from the [objectives](#_1.1_Objectives) and [in scope](#_2.1_In_Scope) sections noted above.

Actors involved:

1. User – potential customer
2. Application – the augmented reality system

### 2.4.1 Blind Overlay Flow

Phase one process flow:

1. The User loads the application on a compatible device.
2. The Application triggers / requests access to use the user’s compatible device camera.
3. Once the camera is active, the user aligns the window to be within the centre of their device screen.
4. The User triggers the blind overlay setup.
5. The Application spawns a solid white image tile at the centre of the screen. This image tile will be a base size of one unity metric unit.
6. Once the application spawns the image tile, it also makes visible four touch-screen buttons. These buttons will allow the user to adjust the scale of the spawned image tile.
7. The User using the width and length buttons adjusts the image tile till it covers the window that the user centred around during step 3.
8. Once the scale has been adjusted, the user triggers the completion of the overlay setup.
9. The Application replaces the solid white image tile, with an image of the designated first product example.
10. The Application spawns a scroll menu containing [X] number of different product examples, that the user may select from to change what is being overlaid.

### 2.4.2 Curtain Overlay Flow

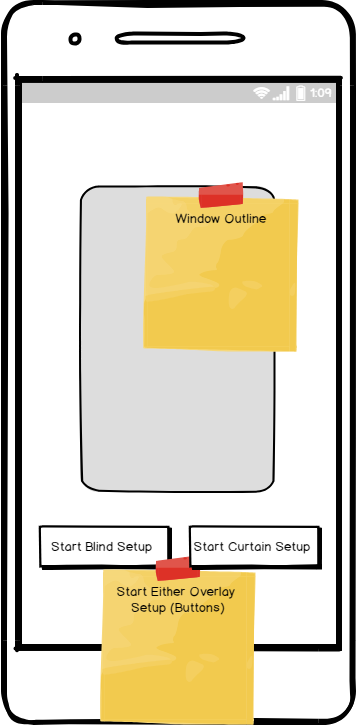
Phase one process flow:

1. The User loads the application on a compatible device.
2. The Application triggers / requests access to use the user’s compatible device camera.
3. Once the camera is active, the user aligns the window to be within the centre of their device screen.
4. The User triggers the curtain overlay setup.
5. The Application spawns two solid white image tile on the left and right side of the user’s device. These image tiles will be a base size of one unity metric unit.
6. **[Step Six Added After User Testing]** Once the application spawns the image tiles, it also makes visible two touch-screen buttons. These buttons will allow the user to adjust the space between the blinds and will allow them to position correctly on the sides of the window frame.
7. Once the user is done placing the blinds, the application makes visible four touch-screen buttons. These buttons will allow the user to adjust the scale of the spawned image tiles.
8. The User using the width and length buttons adjusts the image tiles till it covers the window that the user centred around during step 3.
9. Once the scale has been adjusted, the user triggers the completion of the overlay setup.
10. The Application replaces the solid white image tiles, with an image of the designated first product example.
11. The Application spawns a scroll menu containing [X] number of different product examples, that the user may select from to change what is being overlaid.

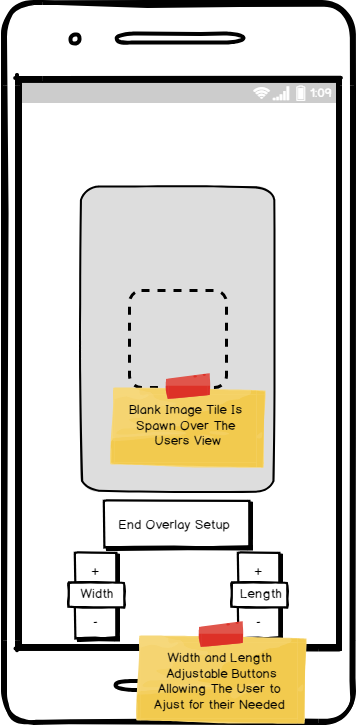
## 2.5 [Purposed] Process Flow Mock-ups

### 2.5.1 Blinds Process Flow Mock-ups

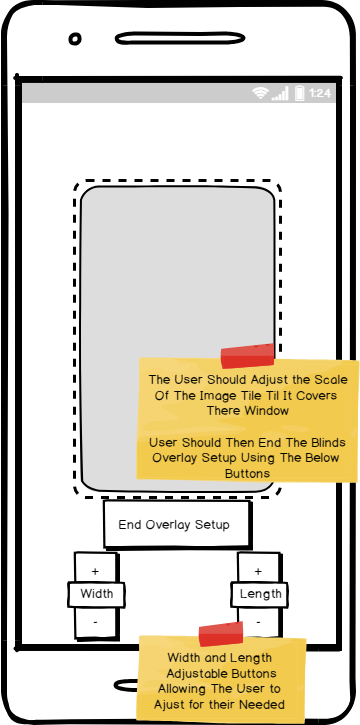
**Blind Process One**



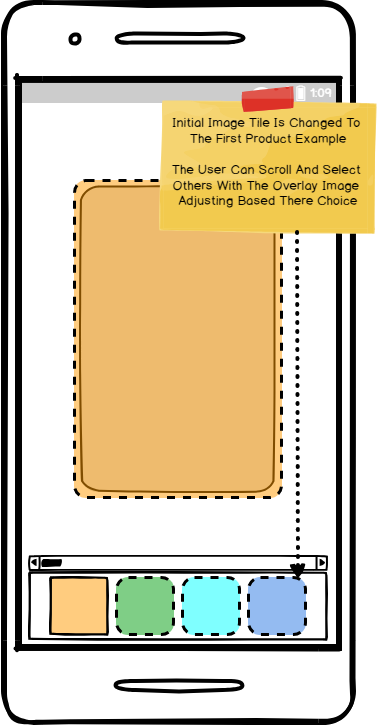
**Blind Process Two**

****

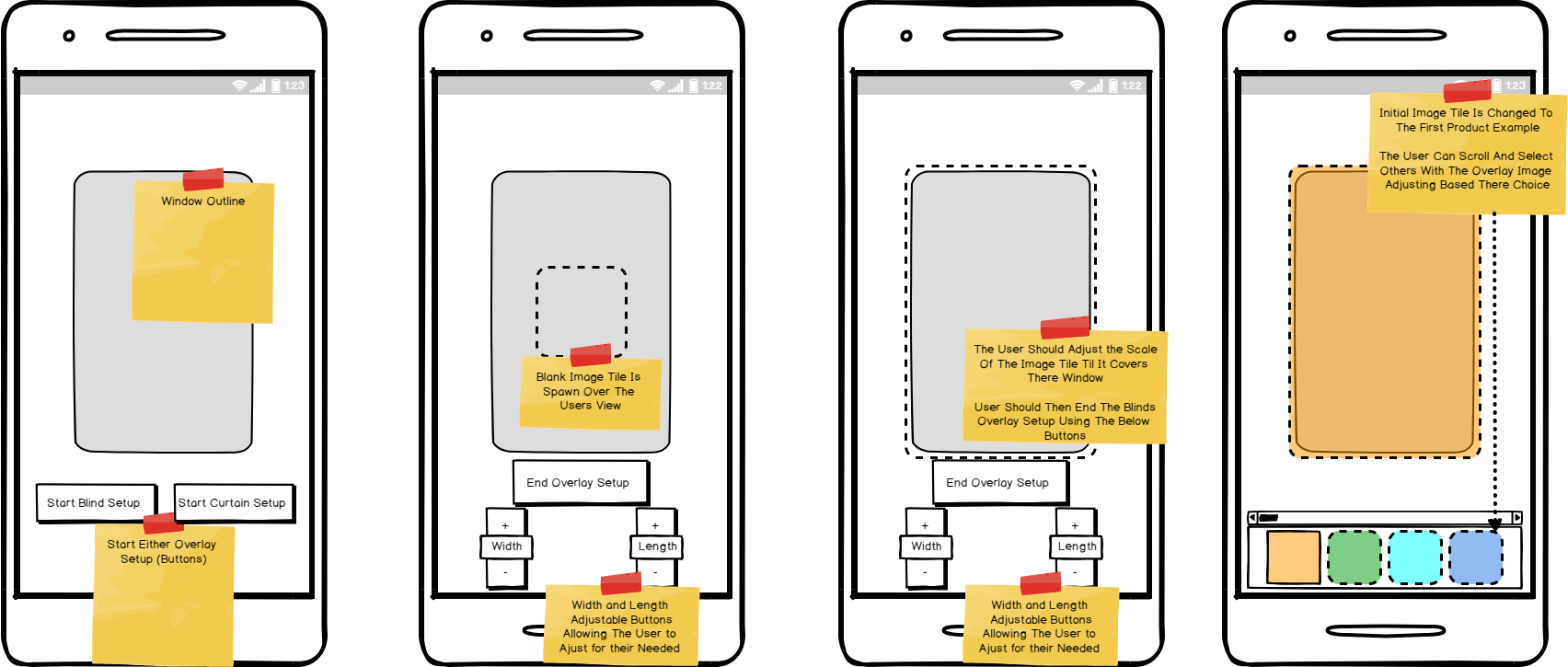
**Blind Process Three**



**Blind Process Four**

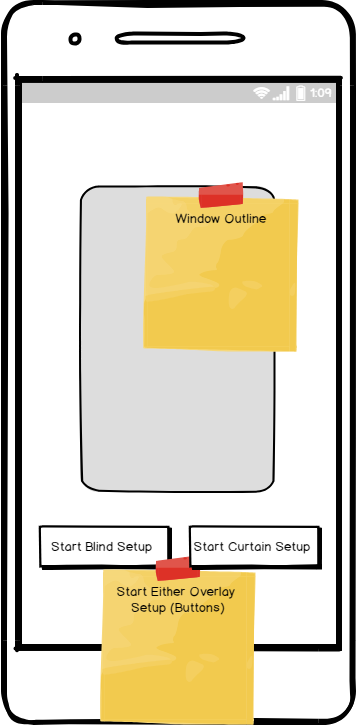
****

**Blind All Processes**

****

### 2.5.2 Curtains Process Flow Mock-ups

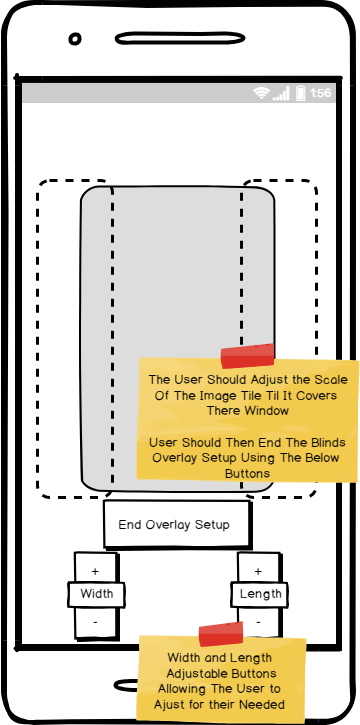
**Curtain Process One**



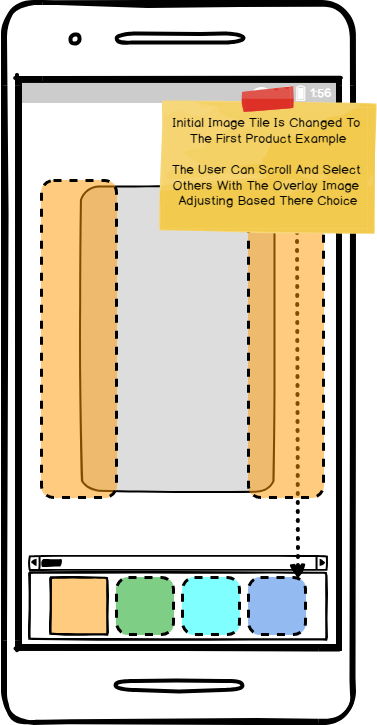
**Curtain Process Two**



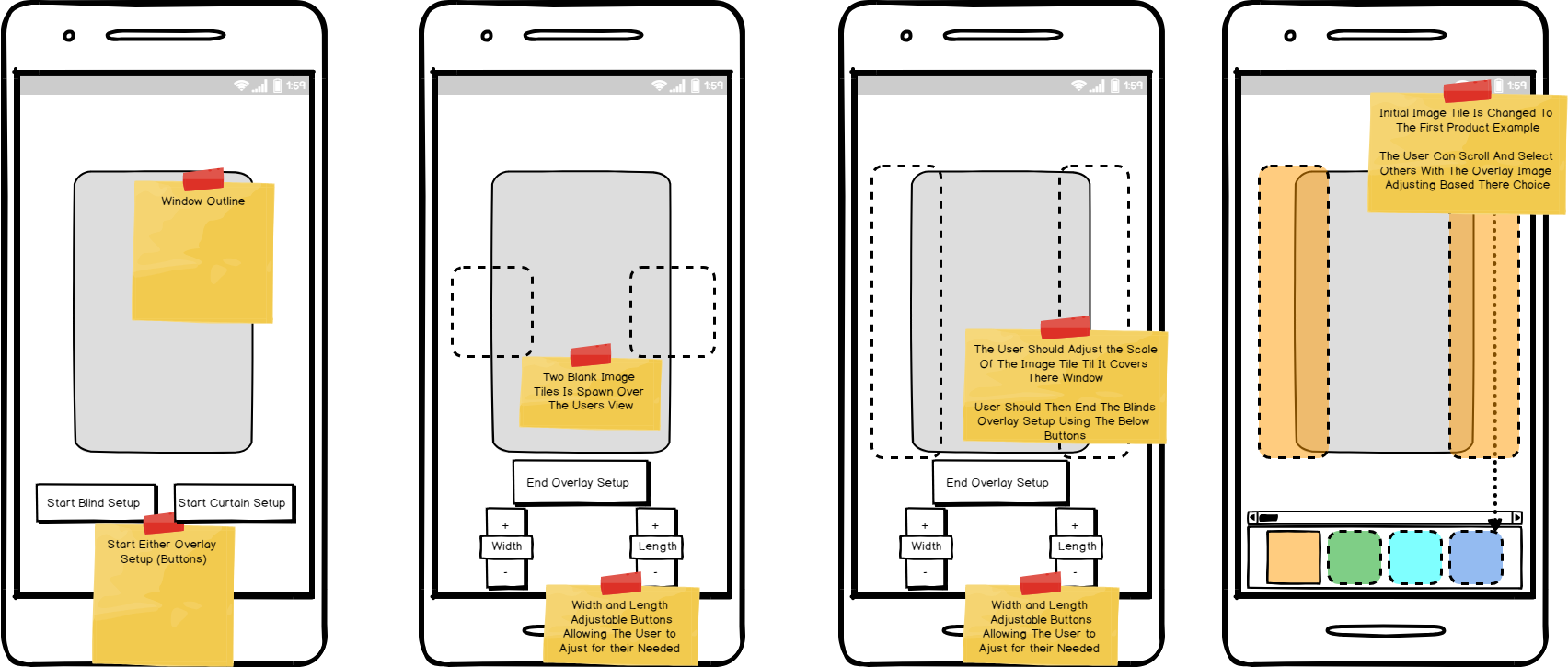
**Curtain Process Three**



**Curtain Process Four**

****

**Curtain All Processes**

****

# 3. Phase One – Features

This section features all of the systems completed, prototyped or investigated for phase one of this eight-week project along with notes on their overall status.

## 3.1 Features Explained

**Loading / Introduction Scene**

As was mentioned at the development meeting, most applications in this realm have a simple but affective loading / introduction scene.

This application also contains a loading scene between the user is brought to the main application home page. At present this scene just contains a button that will allow the user to move to the main application home page. But it can be easily upgraded to include the [INSERT COMPANY NAME] logo / name. In terms of the Unity Build for this project, this feature is contained within the **00\_LoadScreen scene** and at present are functioning as intended.

**Status**: **Complete**

As was requested a number of another AR options where investigated, these features have been prototyped in a separate scene in order to not effect the current in-scope build listed in the previous section of this report. These AR options where taken from the [out of scope section](#OutOfScope).

**Image Database Detection (Used in Final APK Build)**

Similar to the Mid-Air / Ground Plane detection feature an image database containing a single Image [COMPANY LOGO] sample was created. When loaded into the project build and setup correctly all a user would have to do is print off and place the [COMPANY LOGO] centre placed on the window.

The feature would then replace the [COMPANY LOGO] with either the blind / curtain image tile and like the application mock-ups above it fully allows the user to adjust the size and position of the overlaid area.

The use of this detection feature allows the application to automatically readjust as the user moves giving them a more realistic AR experience, this was stated as an out of scope feature back in the out of scope section.

In terms of the Unity Build for this project, this feature is contained within the **03\_ImageDatabasePrototype** and at present are functioning as intended.

**Status**: **Complete**

**In-Scope Application (Not Used in Final APK Build)**

As stated in the [in scope section](#InScope) all features have been implemented and have undergone unofficial testing with a limited user base.

The above process mock-ups outline this current functionality step by step. In terms of the Unity Build for this project, these features are all contained within the **01\_NonARVersion** and at present are functioning as intended.

**Status**: **Complete**

**Mid-Air / Ground Plane Detection**

Mid-Air / Ground Plane detection is a new feature of Vuforia that allows objects to be spawned without the need for a real-world trackable item, such as an image. For this application both options where investigated fully. Due to the hardware requirement for this Vuforia feature it was only developed to the prototype stage and is untested on mobile at present.

The current build requires the Vuforia ground plane sample image in order to function, but it does function as intended allow the user to spawn either the blinds or curtain objects on tapping the screen and like the in-scope application it fully allows the user to adjust the size and position of the overlaid area.

The use of this detection feature allows the application to automatically readjust as the user moves giving them a more realistic AR experience, this was stated as an out of scope feature back in the out of scope section.

In terms of the Unity Build for this project, these features are all contained within the **02\_GroundMidPrototype** and at present are functioning as prototyped.

A demo video is available at this [location](https://drive.google.com/file/d/1IBi5GWPkJBp9GSnh043o3WjfVO75ZkWF/view?usp=sharing) and features a walkthrough of the blinds flow structure.

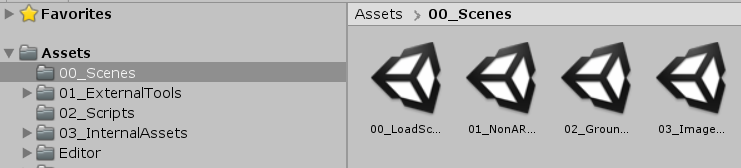
**Status**: **Prototyped**

[Note: All status: **Prototyped** features mentioned in this section are in-complete and non-tested on mobile devices]

[Note: See Appendix for Logo reference and Ground Plane detection reference images]

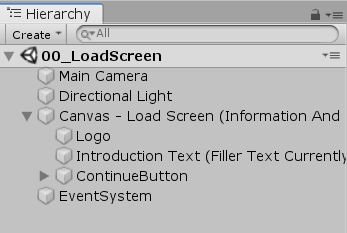
# 4. Phase One - Build

## 4.1 Unity Project Explained (Internal Use)



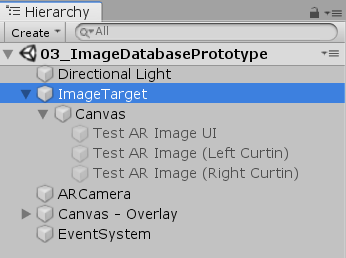
All Unity build scenes are located within the 00\_Scenes folder and numbered according to there development at the time. Scenes explained:

* **00\_LoadScreen:** 
  + First scene the user will see when they open the application. Basic interface containing a Logo, Introduction text and a continue button that will allow the user to move to the main application (as displayed below)

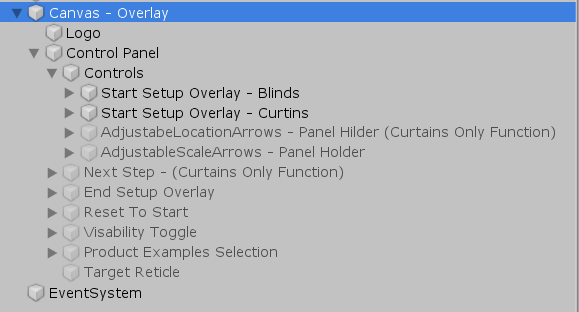
****

[Note: There is currently three different versions of the main application for the purpose of this part of the document I will only be explaining the one used in the final application. But all of the application versions / scenes follow the same structure overall]

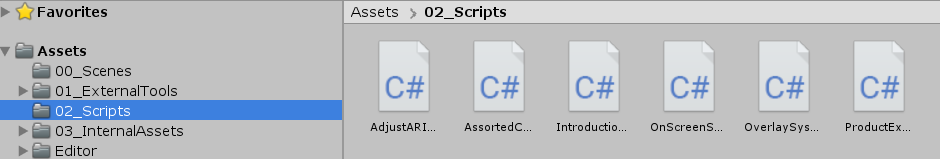
* **03\_ImageDatabasePrototype:**
  + Main application scene and main point of interaction for the user. For the purpose of this section I have broken its hierarchy into two sections.
    - **Image Detection**
    - The application uses Vuforia’s standard setup for image detection, but has three unique elements, that the user will trigger on and off depending on their choice of either blinds or curtains.



* + - These are all wrapped inside a Canvas object that is set to the world space this allows the full AR experience.
    - **UI System**
    - All controls inside the Canvas – Overlay are to do with the UI Controls that the user will use to use the application features.
    - Each UI Element has been named to correspond to the function it carries out.

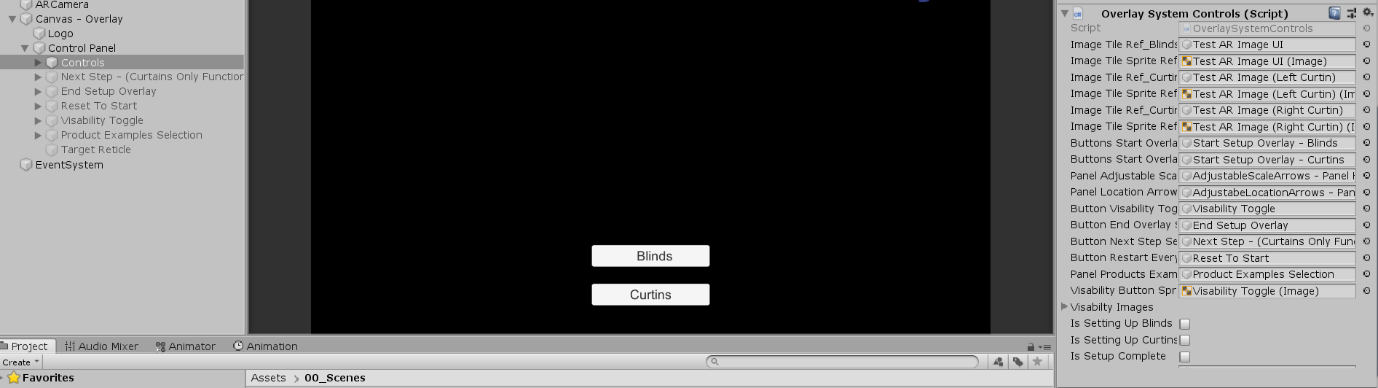
****

[Note: All Scripts supplied are heavily commented in the Unity build itself, so the next section will explain just the basic functions of all self-created scripts for easy reference.]

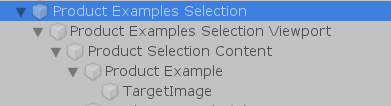
****

All scripts created for this project are located within the 02\_Scripts folder and named to match the function that the play in the overall application. Scripts explained:

* AdjustARImageScale Script:
  + Handles all the functions related to the manipulation of the Image tile which is overlaid into the real world. Such as being able to adjust its width, length and in relation to the curtains flow the space between each tile.
* AssortedControls Script:
  + Contains a single function for the continue button which is used on the initial load scene. Uses Unity’s build in scene manager system.
* OverlaySystemControls Script:
  + Main application function script. Contains most of the core commands for the application flows listed in the previous section.
  + All functions within the script have been commented to explain what they do.
  + Attached to the **Controls** object it contains all references to external objects within the scene. See the below image for references.



* ProductExamplesControls Script:
  + Contains the functionality for allowing the user to switch between different product examples that are listed within a scroll menu at the bottom of the application.
  + Uses an ever-adjusting array of target images that is set when the user selects one of the images from the UI scroll menu
  + **Changing a Product Image:** In order to change a product image, you do not need to modify the code you simply drag and drop a new image onto the **TargetImage** GameObject



* + **Adding a New Product Image to the Scroll:** In order to add a new product image to the scroll menu, all you need to do is duplicate the **Product Example** button and then simply drag and drop a new image onto the **TargetImage** GameObject. The scroll menu and UI will automatically adjust the Product Image there is no need to manually adjust any of the UI objects.

[Note the following two scripts where not used but may contain useful code for future development purposes.]

* **IntroductionSystem Script:**
  + Contains functions related to the setup of the introduction system. Note this system was not developed onwards and is not included in the main application scene any longer.
* **OnScreenSteps Script:**
  + Note this script was not developed onwards and is not included in the main application scene any longer.

# 5. Appendix

## 5.1 Ground Plane



## 5.2 Logo

