## horizontal line



AR Museum

Last update 25/10/2017

**─**

Roi Glink, Shay Michali, Elad Alon

GIP

234329

[**Overview**](#_au51mny0sx6) **2**

[**System**](#_5sz71nbhu5n3) **3**

[**Application overview**](#_8l19rhv35wn8) **4**

[**Development process**](#_ihd36oeuyssx) **6**

[**Image Targets**](#_dn62ui20ix4v) **8**

[**Screenshots**](#_srqhfl1hnu8n) **10**

[**appendix**](#_xh2ebl3xd1zv) **12**

[Image 1](#_apk26ewcha6n) 12

[Image 2](#_pakwo1aw0gz5) 13

[Image 3](#_pgrczso3r1l2) 14

[Image 4](#_nagwwdah6o7x) 15

[Image 5](#_29l4zr293vg) 16

[Image 6](#_71xn7op82p6o) 17

[Useful links](#_h2eh12jzlcxc) 18

# Overview

We developed an Oculus Android gearVR application with both entertainment and educational purposes. Our application mainly offer an AR gameplay while still integrating some VR scenes and thus creating a Mixed reality experience.

The user will be able to stroll at any random location with the headset and with minimal preparation of specific set of images and key points will visit an “AR Museum”. That experience is gained by our application which replaces these images with an art images and 3d models, therefore, offering various options for the user to experience in one room.

Additionally, the application offers both VR and AR interactions with the user. The user can control the navigation of scenes and the content of the museum to his points of interest.

# 

# 

# 

# 

# 

# System

We created an Application for Android, targeted specifically for the Gear VR headset.

The application was developed using Unity 2017.1.1f1 (64-bit) game engine, Vuforia platform and oculus plugin.

Our application is available at the Oculus gear VR Store for free, with the application image set for the user to print and use.

Equipment required

1. Gear VR headset and compatible android device.

* The gear VR headset provides an in game gaze ray mechanism for user interaction. touch pad and a back button are located at the right side of the device as well.



Some relevant links:

https://unity.com/

https://vuforia.com/

https://developer.oculus.com/downloads/package/oculus-utilities-for-unity-5/

# Application overview

The application consist of 3 scenes for the user to Navigate.

1. Main menu - VR scene
2. Exploration scene - AR Scene
3. Zoom scene - VR scene

Main menu scene is intractable with the in game gaze ray pointer and the headset touchpad to sense a user click. This scene allows the user to navigate to the exploration scene or to exit the application.

Exploration Scene, allows the user to see the world throughout the android device camera (AR) and apply the replacement of the images set with a chosen theme from the user.

Using the touchpad and its swipe options the user will be able to switch AR museum themes. By clicking the touchpad button the user will receive an indication of the current theme that is used by the application.

By swiping right or left the user will be able to browse between the themes available in the application.

In addition, One of the images set will act as a gallery and the user will be able to switch its image with swiping up and down.

One image target role is an AR button, meaning the user will click it in the real world, this feature will navigate the user to the Zoom scene .

The headset back button will navigate the user to the Main menu scene.

Zoom scene - Offers a 360 sphere scene that correspond to the theme that was selected in the exploration scene by the user, meaning it content is dynamic. The user will be able to use the back button to return to the Exploration mode.

|  |  |  |
| --- | --- | --- |
| Scene 1 - main menu, VR | Scene 2 - AR exploration | Scene 3 - VR scene |
| com.RSE.ARMuseum-20171019-100945.jpg | com.RSE.ARMuseum-20171019-101445.jpg | com.RSE.ARMuseum-20171019-101502.jpg |

# Development process

To accomplish our final product we used unity 2017 in order to build the application for android with minimal effort from our side. That is because Unity offers a native support for compiling its code for android. We then targeted our application for the gear VR headset, Because the ability to move freely and explore without being linked to a pc was one of our main goals. We then started experiment on the ability to “link” a unity camera object to the headset movement. After some research we discovered that unity offers native support for oculus device as well as some other headsets. Achieving support for the headset in VR scenes was accomplished with somehow medium effort, and was completed when we found the right tutorial in the unity documentation. However simple viewing the scene was not enough for us as we wanted to add some user interactions. We then continued searching and found 2 additional tutorials who use the Oculus utilities plugin. These tutorials helped us adding a gaze ray from the headset to the VR scenes and create VR menu.

We used the vuforia platform because it meet our requirement perfectly. Vuforia algorithm recognizes a predefined image set and creates a new virtual object located on top of it and also offers the AR buttons. therefore, we achieved this part of our application with their documentation only.

Our main problem during production was getting the integration of both VR and AR scenes in the application to work, that is changing the camera mode from projection of the scene to getting the stream out of the android camera into the vuforia object and achieving a good experience for the user when moving.

First attempts with just the Oculus SDK support for virtual reality devices build failed and resulted with black screens or game crashes when navigating to AR vuforia scene. We then discovered that there is a fallout mechanism in that list and that adding other devices to it will give the application the chance to “choose” the appropriate device. However the result was that AR scenes worked, but VR scenes refused to work. After some reading we found that unity offers runtime modification of that list we tried it but these attempts failed as well because the gear VR platform will refuse to change game modes when game is already running and we left that path of solution as it seemed as a dead end. Some additional tutorials claiming to get the needed effect failed and after a lot of research we did find a very old tutorial that worked for us in the link:

https://library.vuforia.com/articles/Solution/Integrating-Gear-VR-and-the-AR-VR-Sample-in-Unity-5-3-and-above

The tutorial created some integration between the vuforia and oculus prefabs game objects and we then could build the application for oculus SDK only as in the beginning and make it work. Some strict options were forced on us in order for that to work and the process was mainly based on trial and error and research on forums and was time consuming.

Some basic explanation is added here, the full tutorials links and information about these steps is found at the appendix chapter under useful links.

|  |  |
| --- | --- |
|  | The native support from unity to oculus and virtual reality devices in the Virtual Reality SDKs list. The fallout mechanism is seen in the list. |
|  | These images shows the integration between the two game camera objects. Vuforia and oculus utilities merge together in order to allow AR and VR scenes transition in the game. |
|  |

# Image Targets

Our application is set to recognize these specific 5 images. Every image role is explained in the table. High resolution copy is also available in the end of the report for printing purposes. Screenshots are added in next chapter to allow better understanding.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Image | Name | Role |
| 1 | 1.jpg | 1.jpg | Serves as a basic image #1. |
| 2 | 2.jpg | 2.png | Serves as a basic image #2. |
| 3 | 3.jpg | 3.jpeg | Serves as a basic image #3. |
| 4 | 4.jpg | 4.jpg | This image target servers as a gallery. Swiping up or down in the touchpad will browse the current theme images. |
| 5 | 5.jpg | 5.jpg | This image serves as an AR button the user will be able to navigate to the VR scene by clicking it. |

# Screenshots

|  |  |
| --- | --- |
| Screenshot | Explanation |
| com.RSE.ARMuseum-20171019-100945.jpg | Opening scene - entering the museum from virtual scene into the ar camera.  The user will be able to press these buttons using the button on the gear VR touchpad button and while gazing on it with the blue gaze ring. See headset buttons at System page. |
| com.RSE.ARMuseum-20171019-101122.jpg | The stickers in the “real world”. Looking on the stickers with the gear vr headset will transform these stickers to pictures. |
| com.RSE.ARMuseum-20171019-101429.jpg | After the transform. The images replace the stickers so whoever wears the headset will see this view. |
| com.RSE.ARMuseum-20171019-101406.jpg | The virtual menu. Switching the theme of the displaying pictures is possible by hand hover. |
| com.RSE.ARMuseum-20171019-101445.jpg | Switching from 2D picture, to 360 virtual scene. |
| com.RSE.ARMuseum-20171019-101502.jpg | Inside the 360 degree virtual scene, after entering from the museum. |

# appendix

## Image 1



## Image 2

## 2.jpg

## 

## 

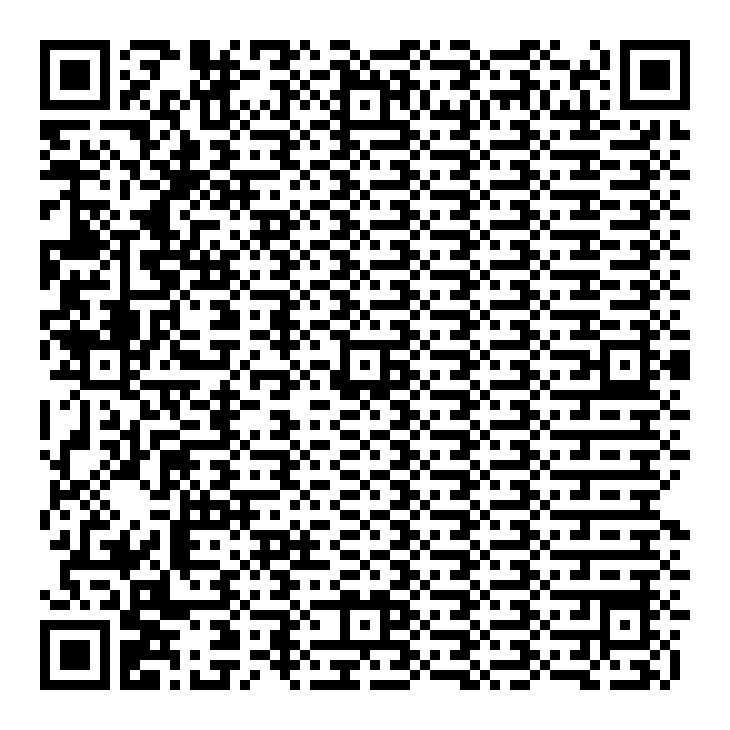
## Image 3

## 3.jpg

## Image 4

## 4.jpg

## Image 5



## Useful links

1. <https://library.vuforia.com/articles/Solution/Integrating-Gear-VR-and-the-AR-VR-Sample-in-Unity-5-3-and-above>
2. <https://tutorialsforvr.com/setting-up-unity-gear-vr-development/>
3. <https://tutorialsforvr.com/creating-a-gaze-based-ui-for-gear-vr-in-unity/>
4. <https://developer.oculus.com/distribute/latest/concepts/publish-mobile-manifest/>
5. <http://answers.unity3d.com/questions/62256/create-keystore.html>
6. <http://www.technoreply.com/creating-an-android-keystore-on-unity/>