COMP 4801 Final Year Project
Intermediate Report

MAXAR - Augmented Reality Presentation Mobile Application

Group member:
Li Chun Hang, 3035304259
Yung Tsz Hin, 3035277353

Date of submission:
20th January, 2019
Abstract

Visual aid like presentation slides are an indispensable part of oral presentations. However, it is difficult to integrate into presentations well due to the separation between the presenter and the slides. By deploying augmented reality into visual aid, presenters and the visual elements in the slides will both appear in the same scene, leading to a more natural use of presentation slides. To achieve this, an easy-to-use presentation slides editor and viewer should be developed. With this application, skills in 3D graphics and programming are no longer required to produce such visual aid. It is expected that the application to be developed will increase the popularity of augmented reality in presentations. Currently, the design of the application has been done and the development has entered the implementation stage.

Acknowledgement

We would like to thank Dr. Loretta Choi for the continuous support of this project.
1. Introduction 4

2. Previous Works 5
   2.1. Microsoft PowerPoint 5
   2.2. Prezi 5
   2.3. Nissan Teana Presentation in Guangzhou Auto Show 5

3. Design 6
   3.1. Basic Flow 6
   3.2. Presentation Editor 6
   3.3. Presentation Viewer 7
      3.3.1. Face Tracking 8
      3.3.2. Remote Control 8
      3.3.3. Partial Retake in Recorded Presentation 8
   3.4 Technologies Used 8

4. Current Status and Problems 8
   4.1. Visual Elements 8
      4.1.1. Text 8
      4.1.2. Stage 9
   4.2. Slides Editor 9
      4.2.1. Camera Movement 9
      4.2.2. Object Selection 10
      4.2.3. Object Movement 10
   4.3. Slides Player 10
      4.3.1. Object Displaying 10
      4.3.2. Object Loading 11
   4.4. Problems 11
      4.4.1. Use of “DontDestroyOnLoad” 11
      4.4.2. Ground Plane Orientation 11

5. Future Planning 12
   5.1 Features in the Second Demo 12
   5.2 Features in the Final Product 12

6. Conclusion 12

References 13
List of Figures

Figure 1  Basic flow of the program 6
Figure 2  Layout of presentation editor 7
Figure 3  Camera set up before presentation 7
Figure 4  3D configuration of the distance calculation 9
Figure 5  First test with a 15 cm x 15 cm stage 10
Figure 6  Same test but with stage shrunk to 10 cm x 10 cm 11

1. Introduction

Presentation slides are commonplace in oral presentations. Currently, they can be created using software such as Microsoft PowerPoint, Prezi and Apple Keynote. However, the slideshows made with these tools are isolated from the reality. This is especially problematic with the increased popularity of E-learning, where presentations are streamed online. In this setting, only one of the speaker or the slides is shown in the video, which may confuse the audience.

Augmented reality renders graphics on top of camera video, which allows the virtual objects to look and behave as they were real. This brings two advantages when used in presentations. First, it provides a more enjoyable and exciting experience to audiences. Second, it provides a way to display both the speaker and the slides in an E-learning session.

Augmented reality technologies has become mature in recent years. Compared to web-based product presentations, AR-based product presentations are said to be providing more effective communication benefit. Audience’s purchase intention and immersion is enhanced through enjoyment and excitement in AR experience [6]. Several software development kits have been released for making application with augmented reality support. However, this requires programming and thus not suitable for end-users. Therefore, it is beneficial to create a tool for the general public to take advantage of the technology in their presentations. Our solution to this problem is to develop an easy-to-use mobile application which allows users to create, edit and display augmented reality-enabled presentation slides.

This report first explores some popular presentation software and an example of augmented reality usage on presentation and highlight some of their features which should be considered in our project. Next, it introduces the design of our application on both the user interface and the technical details. Finally, it explains the difficulties encountered during the early stage of development and the mitigation strategies.
2. Previous Works

In this section, two presentation software and a presentation making use of augmented reality are examined. They all have their problem, but they also give ideas on what a successful presentation software should offer.

2.1. Microsoft PowerPoint

PowerPoint creates presentation slides which is displayed as a slideshow. Numerous templates and clip art are built into the application. Users can also apply animations on text, images and transition between slides. Because of these built in visual elements, users can quickly design aesthetically pleasing slides without having to possess good image processing and video editing skills. Starting from PowerPoint 2016, users can also add 3D objects to the presentation. Rotation can be made to the model during the presentation to show different features of the object [1]. The disadvantage with PowerPoint is that it does not produce impressive presentation slides. Templates can be boring and it sometimes clashes with text. Animation and sound can also be a distraction to the audience [2]. PowerPoint also works poorly when the presentation is recorded since speaker and slides cannot be displayed together at the same time. Video editing has to be done to show both simultaneously, which is time-consuming.

2.2. Prezi

Prezi create presentation slides based on zooming in and out on a giant canvas. This provides more room for users to design unique presentation slides. Like Microsoft PowerPoint, it does not work well in recorded presentation. However, Prezi is planning to add augmented reality functionality to their product, which overlays the slides on the video [3]. This solves the problem with recorded presentation. Although this functionality has not been released yet, there are some possible limitations shown in the previews. First, the images do not react to speakers’ gesture. Also, the images are always on top of the video, so it may look unnatural when part of the images are supposed to appear behind the speaker.

2.3. Nissan Teana Presentation in Guangzhou Auto Show

Nissan has used augmented reality in their presentation of Nissan Teana during the Guangzhou Auto Show in 2013 [4]. They used it to present the internals of the car. Compared to showing the objects in a presentation slides, this is more immersive and will give a more believable experience to the audience. Compared to Prezi, this is a better use of augmented reality technologies. In this presentation, visual elements are aware of the camera angle and the movement of the speaker. This gives a more realistic result. Similar presentations can also be found elsewhere, but it is often seen in big events done by professional. Our team hope to enable ordinary users to make use of augmented reality to enhance the visuals used in their presentations.
3. Design

Our project consists of a presentation editor and a presentation viewer. The project will be implemented as a mobile application so that users can create and deliver presentations using the same device. This leads to an easier setup for presentations compared to a desktop application since it does not require installing additional software on any computer or connecting camera to a computer.

3.1. Basic Flow

Figure 1 shows the flow of our program during normal usage. Users first have the choice to create or load a presentation. Then, the program brings users to the presentation editor. Apart from editing slides, users can also access the presentation viewer from it. Users may choose to share the screen or record the presentation. After the selection, the program guides users to set up the mobile device as a camera. Finally, the presentation slides are displayed.

3.2. Presentation Editor

Users create their slides in a 3D environment. A stage border is provided in the environment, users have to import 3d models or insert the text within the border given. Objects are rendered with the size and position relative to the ratio of border to object.
After the application gathers the information about the environment of the place in which the presentation is taken place, a representation of the stage will be displayed on the screen. Figure 2 shows the screen when editing a slide. At the top of the screen are the buttons for saving the slides, inserting objects and text and entering the presentation viewer. Users place objects and text on the stage and select them by tapping on them. When an object is selected, a menu will appear at the bottom of the screen for users to adjust object properties. Like Microsoft PowerPoint, objects can be associated with transitions and animations by moving and rotating.

3.3. Presentation Viewer

Presentation viewer displays slides created with the editor. It opens the camera on the device for capturing video on which objects are drawn on. Before the start of presentation, the device has to be placed in the same way as it is defined in the initial camera position. User has to define the size and the position of the stage border using pinch and spread gesture. Figure 3 shows the user interface design for user to locate the stage.
After setting up the camera, the presentation starts and so the program renders visual elements on the screen. The scenes produced are then projected onto a screen in a live presentation. In online presentations, the scenes are streamed through the internet instead. It also provides a way for users to record and export the presentation to a video file. Below are three special features which will be implemented in this part of the project:

3.3.1. **Face Tracking**

Sometimes, presenters may want to show large objects in the presentations. Because of their size, they will take up a substantial amount of screen space and thus the need for a zoomed out view of the stage. When the large objects are gone, presenters may want the video to focus on themselves. Machine learning based method is used to detect speaker in the video and so the application can zoom into where the speaker is. The application keeps track on the speaker until it is required to zoom out again.

3.3.2. **Remote Control**

A separate remote control application will be developed so that users can use a different device to control the presentation slides. This is particularly useful in a large stage, where the device hosting the slides is far away from the presenter.

3.3.3. **Partial Retake in Recorded Presentation**

After recording the presentation, users can review the resulting video clip. Users may retake parts of the presentation without having to edit the video themselves. Presentation parts can be selected by page numbers. After retaking the part, the new recording replaces the corresponding part of the video automatically.

3.4 **Technologies Used**

Vuforia is chosen for handling computer vision-related tasks, particularly on plane detection and visual odometry functionalities. Other software development kits such as ARCore and ARKit may provide similar features, but Vuforia has the advantage of supporting both iOS and Android and also older devices.

Unity is used as a 3D rendering engine in our project because of its good integration with Vuforia. Although it is a game engine, it allows scripting such that developers can customize the behaviour of objects in their projects. Because of this, Unity can be used for applications other than games.

4. **Current Status and Problems**

The team is currently implementing the core components in the design. For example, the mechanisms for visual element manipulation and sharing data between scenes have been implemented in the demo application.

4.1. **Visual Elements**

4.1.1. **Text**

Text is a common feature in presentation slides. Because of depth in 3D slides, characters should be rendered with meshes to make them stand in a scene. Unity uses TextMesh Pro in accomplishing such effect by default. This package draws characters by putting textures into rectangles. It is capable of generating 3D text with different fonts.
and typographical emphasis. However, the characters TextMesh Pro creates are not volumetric, which looks unnatural when viewed at an angle.

The team has found another package on Unity Asset Store called “Simple Helvetica”. This package creates volumetric characters but it only supports one font and a limited set of characters. This is because all the character models are precomputed and it uses a script to bring the models to the scene. If there is no matching model to the character, errors may occur.

Currently, Simple Helvetica is used in the project since it gives a more natural look than TextMesh Pro does. It also has the benefit of being simple in design and so it is easier to extend its functionality. The team will look into generating character meshes dynamically during runtime. Once the models are generated, the mechanism used by Simple Helvetica to display characters can be used.

4.1.2. Stage

Stage models the physical stage on which a presentation is delivered. It is a special type of visual element found in the project such that there is always a single instance of it in a presentation. In the slides editor, Stage is represented by a horizontal plane. The same rectangle also appears in the slides player as a marker of the detected ground plane.

Stage object stores the dimensions of the stage in metres. Its size is reflected in its representation in both the editor and the player so as to visualize the scale of the visual elements in the presentation.

4.2. Slides Editor

4.2.1. Camera Movement

In Unity, users’ point of view is controlled by the Camera object. Since users may want to look at different parts of the stage, the program allows users to rotate and pan the camera by swiping on the screen.

The rotation is implemented follow the Euler’s rotation theorem. Since users control in a 2D space, only 2 axes are used in the rotation. To prevent the scene from tilting left and right, the camera will rotate about X and Y axes. The magnitude of the translation is determined by the difference between the touch positions in each frame. This requires the conversion between screen coordinates and world coordinates and thus the need for a distance between the touch and the camera.

To calculate the distance, first we get the vector from the origin to the camera. Then, we get the projection of the vector to the horizontal plane at \( y = 0 \), which is denoted by the black solid line in Figure 4. Finally, we take the magnitude of the projection as the distance. This distance calculation works regardless of whether users drag on an object. However, it is not effective when the magnitude of the projection is close to 0. To overcome this limitation, the team will also implement a lower limit for the distance.

![Figure 4: 3D configuration of the distance calculation](image)
4.2.2. **Object Selection**

Users may select any visual element by tapping on it. The program uses ray casting to check if an object is touched. The ray shoots from the camera to the touch position on the screen and continues forward. If the ray hits an object, the object will be selected. If the ray does not hit anything, any selected object will be deselected.

4.2.3. **Object Movement**

Once an object is selected, users may change its attributes, including its position. Objects may either move horizontally or vertically. To move an object, users drag on the screen and the object will follow the touch position. On the start of object movement, the program creates a logical plane which intersects with the touch position. A horizontal plane will be created for horizontal movement and a vertical plane will be created otherwise. For each update, ray casting is used to find the touch position on the plane. If such point exist, the object will move to the location.

4.3. **Slides Player**

4.3.1. **Object Displaying**

The slides player utilizes Vuforia’s ground plane feature to place objects in the environment. This feature first scans for a horizontal plane from the camera video. Then, it anchors the scene on the detected plane at where the device is pointing to.

According to Vuforia, the visual guide displayed in the Unity Editor serves as a guide to scaling objects [5]. The visual guide is a 1 x 1 square in Unity and it represents 1 m x 1 m in real life. Therefore, 1 unit in Unity equals to 1 m in real life.

The team has tested the scaling in the project. First, a 15 cm ruler is placed on a flat surface. Then, a 15 cm x 15 cm stage is created in the application together with some text. After that, the scale is checked in the slides player.

![Figure 5](image_url)

**Figure 5** First test with a 15 cm x 15 cm stage

From Figure 5, it is shown that the side of the stage matches with the measurement on the ruler. Also, the margin between the text and the stage is the same in both the editor and the player.
For comparison, the stage was changed to 10 cm x 10 cm while the scale of the text remained unchanged.

Figure 6  
Same test but with stage shrunk to 10 cm x 10 cm

From Figure 6, we can see that the stage is shrunk to 10 cm x 10 cm in the player. Also, the text was having roughly the same width as the same just like in the editor.

These two tests showed that the scale is carried over from the editor to the player properly. If Vuforia manages to detect horizontal planes correctly, the visual elements will appear as expected.

4.3.2. Object Loading

Currently, objects are loaded into the player by keeping the relevant objects from the editor. This is achieved by using Unity’s “DontDestroyOnLoad” method, which prevents objects from being destroyed during scene switching from the editor to the player.

At the start of the player scene, the program clones the objects in the editor and make them children of the ground plane object. The positions of the object relative to the origin is preserved and hence the scene is reconstructed in the slides player.

4.4. Problems

4.4.1. Use of “DontDestroyOnLoad”

Currently, our project uses Unity’s “DontDestroyOnLoad” function for passing data between scenes. There problem with this approach is that it makes the code unmaintainable.

Using this function requires adding code in every class which is to be kept. First, we have to invoke the function once during initialization. Second, the object needs to be able to detect duplication and delete itself if it is not the original instance. Finally, the objects needs to be disabled so that they can be hidden from the user completely. This makes the code related to visual elements complicated.

To avoid the use of “DontDestroyOnLoad”, we will create a dedicated static class for storing presentation data. Ultimately, this static class will store everything which will be saved to a file such as stage dimensions, visual elements and animations. Whenever a scene needs to gather such information, it will load it from the static class instead of relying on any object in another scene.

4.4.2. Ground Plane Orientation

Vuforia generally detects ground planes correctly. However, it also tries to detect the orientation of the plane, which is not always accurate. As a result, the rendered scene may sometimes be rotated at an undesirable angle.

To mitigate this problem, the slides player should allow users to manually rotate the stage before the start of a presentation.
5. Future Planning

Our team will start working on the implementation of the real application, which includes both the editor and viewer. After 17th March, 2019, no major change will be made to the application to leave time for testing and optimization. Documentation is updated throughout the development of the project to enhance maintainability of code.

Due to the small size of the team, only a subset of the features can be implemented at a time. As a result, the program will only be feature complete in the final version. Because of the tight schedule, the actual features to be included in the final product depend on the progress of our implementation.

5.1 Features in the Second Demo

The first demo is demonstrated in the first presentation. The second demo will be done by 31st January, 2019. This demo includes the presentation editor, which means users can create their own slides and display them using the viewer. In the editor, users can create 3D text and simple shapes such as pyramids and spheres. The program should also be able to save and load presentation slides from the local storage of the mobile device. Moreover, recording and saving function should also be included in the second demo.

5.2 Features in the Final Product

After the discussion on project scope with our supervisor, we have decided to include the following features in the final product.

- 3D graphs
- Dynamic text mesh generation
- Animations
- Built-in 3D models
- Face Tracking/ Full viewing mode

Since the main focus of this project is about the application of augmented reality, these features are adequate in delivering a workable proof of concept of a augmented reality presentation software. After finishing the implementation of the features above, the team will also try to implement model import and remote control features for completeness. Only when these features are in place will the team attempt to implement the remaining features mentioned in this report.

6. Conclusion

We have examined the current status of presentation application and the usage of augmented reality in presentation. The current presentation applications do not cope with the popularization of E-learning because of their way of displaying slides. Despite the maturity of augmented reality technologies, it is still too complicated to use for most people.
This report has introduced the design of an augmented reality presentation mobile application targeted to the general public. Some of the core components in the project have been implemented, but more work has to be done before the product can be considered useable.

Once the project is finished, augmented reality will become more accessible to normal people. Presenters will have a better way to display their slides while audiences can enjoy the new visual effects. Both of them will have greater exposure to augmented reality. With so many presentations delivered every day, the application could be an effective way to promote augmented reality.

References