

# Augmented Reality Integrated Comic Books

Vishwatmika Srivastava, 19332239  
M.Sc. Computer Science (AVR)  
School of Computer Science and Statistics  
Trinity College Dublin

## CONTENTS

<b>I</b>	<b>Introduction</b>	1
<b>II</b>	<b>Theoretical Background</b>	1
<b>III</b>	<b>Implementation</b>	1
III-A	Page.1: . . . . .	2
III-B	Page.2, Page.3, Page.4 and Page.5: . . . . .	2
III-C	Page.6: . . . . .	3
<b>IV</b>	<b>Experiments &amp; Results</b>	3
<b>V</b>	<b>Analysis</b>	3
<b>VI</b>	<b>Summary, Conclusion and Future Work</b>	3
<b>References</b>		3

## LIST OF FIGURES

1	3D objects on the marker . . . . .	1
2	Connection of Vuforia, Unity and devices. . . . .	1
3	Vuforia License for application. . . . .	1
4	Image Targets for application. . . . .	2
5	Placement of objects over image targets . . . . .	2
6	Image Targets . . . . .	2
7	Outputs . . . . .	3

# Augmented Reality Integrated Comic Books

**Abstract**—Youtube Link: <https://youtu.be/Xko0ydb7yjI>

This report demonstrates the creation of AR integrated comic books app using Unity Engine and Vuforia for Android smart phones.

This app enhances the user experience of reading the comic book, as it makes the characters lively and narrates the story too.

## I. INTRODUCTION

We all have been obsessed with comic books in our childhood. So, the inspiration is to bring everyone's favourite character back to life using the tools we have in our hands. This application turns the pages into live scenes to convey the story and narration happens in the background.

Here, the **Video Based** virtual content is integrated into real world views. The rendering of AR content happens in real-time. The objective is to create 3D scenes that exhibit the comic narration, set up the markers and integrate the 3D scenes into real world.

## II. THEORETICAL BACKGROUND

A 3D CG object can be easily integrated into a real world by extrinsic and intrinsic transformation of all vertices.

Pose Estimation is an essential problem in AR. It teaches the computer where to put the object, orientation and size of the object i.e. translation, rotation and scaling is specified to the machine so that it can reflect these in the real world and display the 3D object as shown in fig.1. The markers are detected using



Fig. 1: 3D objects on the marker

image processing or computer vision algorithms.

Projection is calculated using eq.1, where p is the projection, C is Calibration and P is Reconstruction.

$$p = C.P \quad (1)$$

## III. IMPLEMENTATION

The application is carried out in Vuforia and Unity using the pipeline shown in fig.2.



Fig. 2: Connection of Vuforia, Unity and devices.

The connection of Vuforia and Unity consists of a few steps [2]:

- 1) Creating a Vuforia account, defining the application and getting the license key as shown in fig.3.

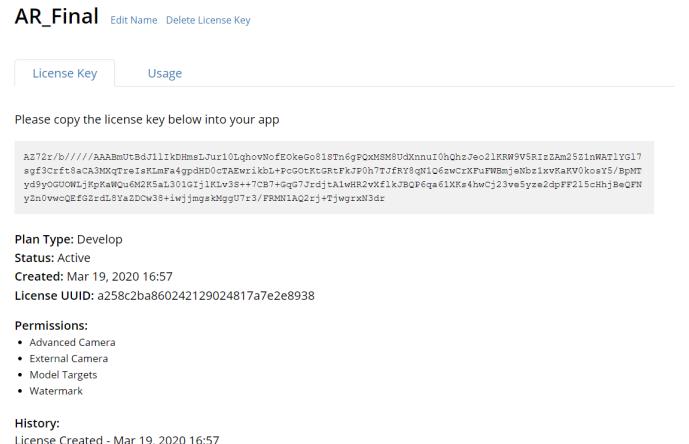


Fig. 3: Vuforia License for application.

- 2) Adding image targets to the database, as illustrated in fig.4. Here, I have used six image targets, each for one page of the book.

AR\_Finals Edit Name  
Type: Device

Targets (7)

Add Target Download Database (All)

Target Name	Type	Rating	Status	Date Modified
page_6	Single Image	★★★★★	Active	Apr 05, 2020 01:36
page_5	Single Image	★★★★★	Active	Apr 05, 2020 01:35
page_4	Single Image	★★★★★	Active	Apr 05, 2020 01:35
page_3	Single Image	★★★★★	Active	Apr 05, 2020 01:34
page_2	Single Image	★★★★★	Active	Apr 05, 2020 01:33
page_1	Single Image	★★★★★	Active	Apr 04, 2020 23:06
ar_marker	Single Image	★★★★★	Active	Mar 19, 2020 17:16

Last updated: Today 03:46 PM Refresh

Fig. 4: Image Targets for application.

- 3) Once the database is set up, AR camera is added to the Unity project and Image Target database is imported.
- 4) Now, Vuforia Augmented Reality Support is checked in Player Settings to complete the configuration.

Once the connection is done, the 3D scenes can be created in Unity.

I have used pre-existing 3D models of the characters to create the scenes. All the animations and effects were created by me in Unity.

All the objects were made children of corresponding image targets, so that once they can be revealed once the targets are detected, as shown in fig.5. The image targets are highlighted with orange. I have used six image targets here and the



Fig. 5: Placement of objects over image targets

implementation is explained briefly below:

#### A. Page.1:

The first page is the title page of the Avengers comic book (fig.6a) that shows all the characters in the frame. Its AR representation is shown in fig.1. All the models are placed into a plane and the background track is timed with the tracking of the marker. As soon as the marker is detected, the track will start playing.

#### B. Page.2, Page.3, Page.4 and Page.5:

The second (fig.6b), third (fig.6c), fourth (fig.6d) and fifth page (fig.6e) narrates the story and illustrates corresponding



(a) Image Target for Page 1



(b) Image Target for Page 2



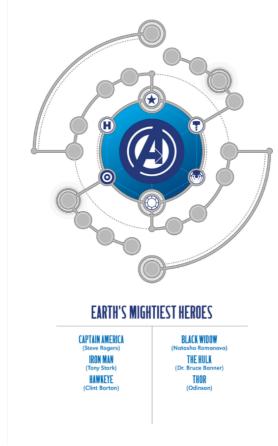
(c) Image Target for Page 3



(d) Image Target for Page 4



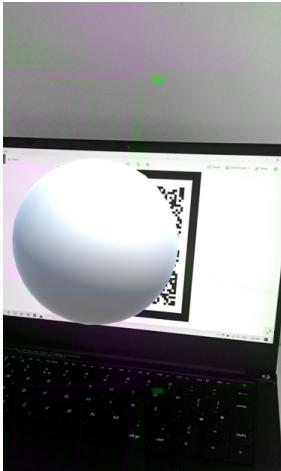
(e) Image Target for Page 5



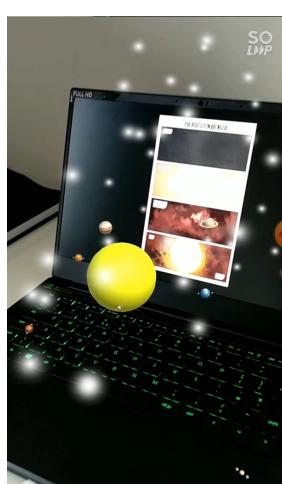
(f) Image Target for Page 6

Fig. 6: Image Targets

animations. These narrations were recorded and then timed



(a) Verification of Vuforia and Unity connection



(b) Output for Page 2



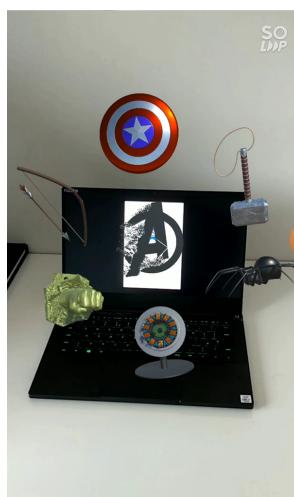
(c) Output for Page 3



(d) Output for Page 4



(e) Output for Page 5



(f) Output for Page 6

with the Unity animations to serve a better user experience. For the scenes, I have used particle systems and gameobjects to describe the story. The objects were taken from **Sketchfab** [1] and all the animations were constructed in Unity using **Animator** and **Animation** components.

### C. Page.6:

There's no narration in this page, and it shows the logos of all six original Avengers. Hence, I have tried to create the same thing with animation as shown in fig.7f.

## IV. EXPERIMENTS & RESULTS

To validate the connection between Vuforia and Unity, I used an additional image target and tried to put a sphere on it. Once this is done as demonstrated in fig.7a, I started creating the 3D scenes.

After that, the results are shown in fig.1 and fig.7.

## V. ANALYSIS

The two main observations in this project are:

- 1) The detection of image targets is very swift and it doesn't take a lot of time. But there are few targets which are less augmentable than others due to their textures, as shown in fig.4 some images have less augmentable rating than others.
- 2) In some scenes, the objects are quivering too much. Maybe it's because I haven't used physical markers and the light coming from laptop's screen might end up in flickering of objects.

## VI. SUMMARY, CONCLUSION AND FUTURE WORK

The overall results seem to be decent. The outputs look engaging and users might like the experience.

The usage of Vuforia with Unity is not that tough, but there are more or less no tutorials for it, which makes the task more interesting as we get to tinker with the software ourselves. I have only done this for a small part of the comic, therefore in future, with more time and efforts, whole comic books can be integrated with AR.

## REFERENCES

- [1] Sketchfab for 3D objects <https://sketchfab.com/feed>
- [2] Build an AR application with vuforia sdk for Unity3D. <https://medium.com/@laurent.heneman/build-an-ar-application-with-vuforia-sdk-for-unity3d-6a91e101aa4>

Fig. 7: Outputs