VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ■ CHENNAI
www.vit.ac.in

CSE4015 –Human Computer and Interaction

Slot: C2+TC2

Prof : Arun Kumar S

Project Title: Augmented Reality Gaming

Final Review Document

Project Team Members:

1) Rathnam Sasidhar Achari (17BCE0895)
2) Sirigiri Susanth Naidu (17BCE0942)
3) Asithureddy Jaganmohan Reddy (17BCE0478)

Declaration

We hereby declare that the project work entitled "Augmented Reality Gaming" submitted to the VIT Vellore, is a record of an original work done by us under the guidance of
Mr. Arun Kumar S , and this project work is submitted in the partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science& Engineering. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma

By

Sasidhar Rathnam

Susanth Sirigiri Naidu

Jaganmohan Reddy

Acknowledgment

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to VIT for their guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

We would like to express my gratitude towards our parents and member of VIT for their kind co-operation and encouragement which helped us in completion of this project.

Our thanks and appreciations also go to all faculty in developing the project and people who have willingly helped us out with their abilities.

Table of Contents:

Abstract

In Augmented Reality (AR), interfaces consist of a blend of both real and virtual content. In this paper we examine existing gaming styles played in the real world or on computers. We discuss the strengths and weaknesses of these mediums within an informal model of gaming experience split into four aspects; physical, mental, social and emotional. We find that their strengths are mostly complementary and argue that games built in AR can blend them to enhance existing game styles and open up new ones.

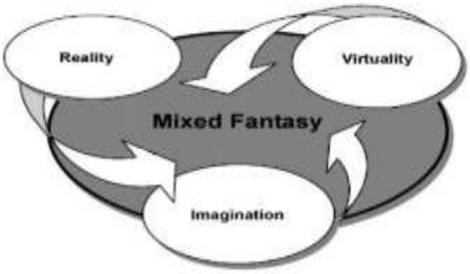


Figure 2 – The Mixed Fantasy Triad [5]

To illustrate these ideas, we present our work on developing game in which there exists a battle between two objects such as firing, smashing and moving using Augmented Reality. We discuss how AR has enabled us to start exploring interfaces for gaming, and present informal observations of players at several demonstrations. Finally, we present some ideas for AR games in the area of strategy and role playing games.

Back ground of the problem:

Augmented Reality (**AR**) is an interactive experience of a real-world environment whereby the objects that reside in the real-world are "augmented" by computer-generated perceptual information. The primary value of augmented reality is that it brings components of the digital world into a person's perception of the real world, and does so not as a simple display of data, but through the integration of immersive sensations that are perceived as natural parts of an environment.

Vuforia is an Augmented Reality Software Development Kit (SDK) for mobile devices that enables the creation of Augmented Reality applications. Vuforia uses the device's camera feed combined with accelerometer and gyroscope data to examine the world. Vuforia uses computer vision to understand what it 'sees' on the camera to create a model of the environment.

Vuforia can seek 'targets' on the camera feed to create anchor points or references to be used by **AR experiences**. Those targets can assume many forms, and any file that has been previously processed by Vuforia target manager can be recognized. Targets can be created by the developer or created during execution by the user.

Vuforia is used in **Unity3D** as SDK for creating apps and Games.**Unity3D** is the basic platform used in this project to create AR games.
Unity3D works very efficiently and also a user-friendly game development software available. Augmented reality is used to enhance natural environments or situations and offer perceptually enriched experiences. With the help of advanced AR technologies (e.g. adding computer vision and object recognisation) the information about the surrounding real world of the user becomes interactive and digitally manipulable.

Motivation of the proposed work:

Real world and computer games have their own distinct strengths.
By allowing us to combine these strengths, we can use AR to improve existing game styles and produce new ones.

For discussion, we consider a player's gaming experience as consisting of four parts; physical, social, mental and emotional. Research into AR gaming serves another purpose beyond the improvement of gaming styles and the development of new ones, as gaming environments are well suited for exploratory research.
In this section we examine the strengths of real world and computer games, the role of AR in combining and extending them, and the value of gaming as an exploratory research area.

We are going to learn how to generate a scene compatible with Vuforia and how to implement various scripts, including ray-tracing, in order to be able to interact with the object within the scene.
After the completion of this Project , We will be having enough Knowledge to work with Unity 3D Software and to design any minor games and to deal with Vuforia Tool
.
We will be knowing what exactly this Augmented Reality means.
We are always interested in learning new software's and in exploring the new world which is unknown for us. The Main Reason we took this Project is we three are so much excited after hearing the word "Augmented Reality". Because we are interested in learning the new and interested things. That motivated us a lot and we took this project as our challenge and working very hard to give you the best Output. You will be knowing what exactly we are doing in coming slides.

Focus on the Proposed Work:

Our Project mainly focuses on a game in which there exists a battle between objects which are augmented by computer into a wall or ground. Unlike the existed games , We operate the battle by touching the ground. But In Existed Work , They just project the objects only and operate the battle by clicking buttons which are in mobile. But in our project, we even project the buttons in the ground and operate the buttons by touching the ground where the buttons exist in Augmented world. For Example, In battle If I want to fire on the other object, I shouldn't click the fire button in Phone. Instead, I need to click(I.e., touch) the ground where the fire button is there (i.e.., augmented by computer).

If You see the above picture, He is playing the game by touching the button on screen. This is what we are going to ignore and we create our own augmented buttons on ground. This is how our project differs from the existed one.

Existing work in the context of proposed work:

As we already said that whatever game you take in augmented it will have only single user but there is no dual or multiplayer in any of the situation and more over regarding the instructions as you see here he is giving the instruction through his hands. They just project the objects only and operate the battle by clicking buttons which are in mobile. But In our project, we even project the buttons in the ground and operate the buttons by touching the ground where the buttons exist in Augmented.

Limitations of the existing work:

- Single user.
- Interaction mode (input signals) via device directly.
- Not more than one object.
- Unlike the existed games , We operate the battle by touching the ground.
- Virtual Buttons are so sensitive that even light shadow falls on button, button gets activated.

Research gaps from existing work:

As mentioned the bulk of AR applications that have been developed in the industrial space. However there have also been a number of examples of AR gaming interfaces. Here we introduce a few of the more interesting systems that cover a range of ways AR can enhance gaming. In general these can be divided into single-user experience and collaborative systems. Most of them will do the projects only by scanning the single item but here we are going to provide the device by two objects which will be very interesting for the user

Objectives:

The main objective of our project is now-a-days every augmented or virtual games are played by giving instructions on the device itself. But here we are going to overcome that defect and introducing the system of giving input not by touching the device but by the ground.
To analyse the advantages and disadvantages
To integrate it into daily lives to help the masses not only to children but also to adult who interested in gaming.

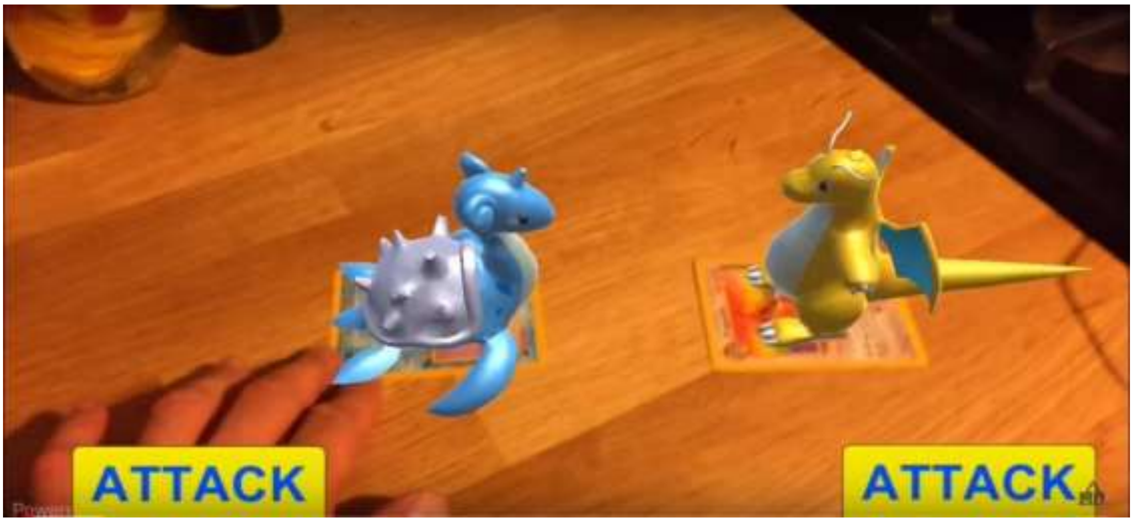
To achieve the feast which are limited in real world.
To introduce the back ground information of augmented reality.
We will have 2 objects which we will be going perform our project on them . We will scan the both the objects to our device . Now we will define the functions to be performed on the device on the objects we scanned through our web cams or tablets. Here the drawback of the every gamer is he will fix some detectives for his object.

But the instruction is given by the hand . they touch the instructions given to the objects . but here we are going to perform the operation only by touching the ground. Here we don't have any interaction between the device and the object . here as the game is augmented we shouldn't have any interaction directly . so we touch the ground where we scanned the objects and perform the instructions.
We will give some health to the both the objects and they we allow to fight. Then there will be two players who can play the game .

For example let us assume we have two objects like a dragon and a thor .We will give have some detectives on both the objects .
When by pressing the attack symbol both the objects will fight and they even loose the power i.e., health.

That's How we play the game

As you see here he is giving the instruction through his hands that the attack button is provided on the ground hence it is one of the drawback of the existing work . but here you see our overcome.

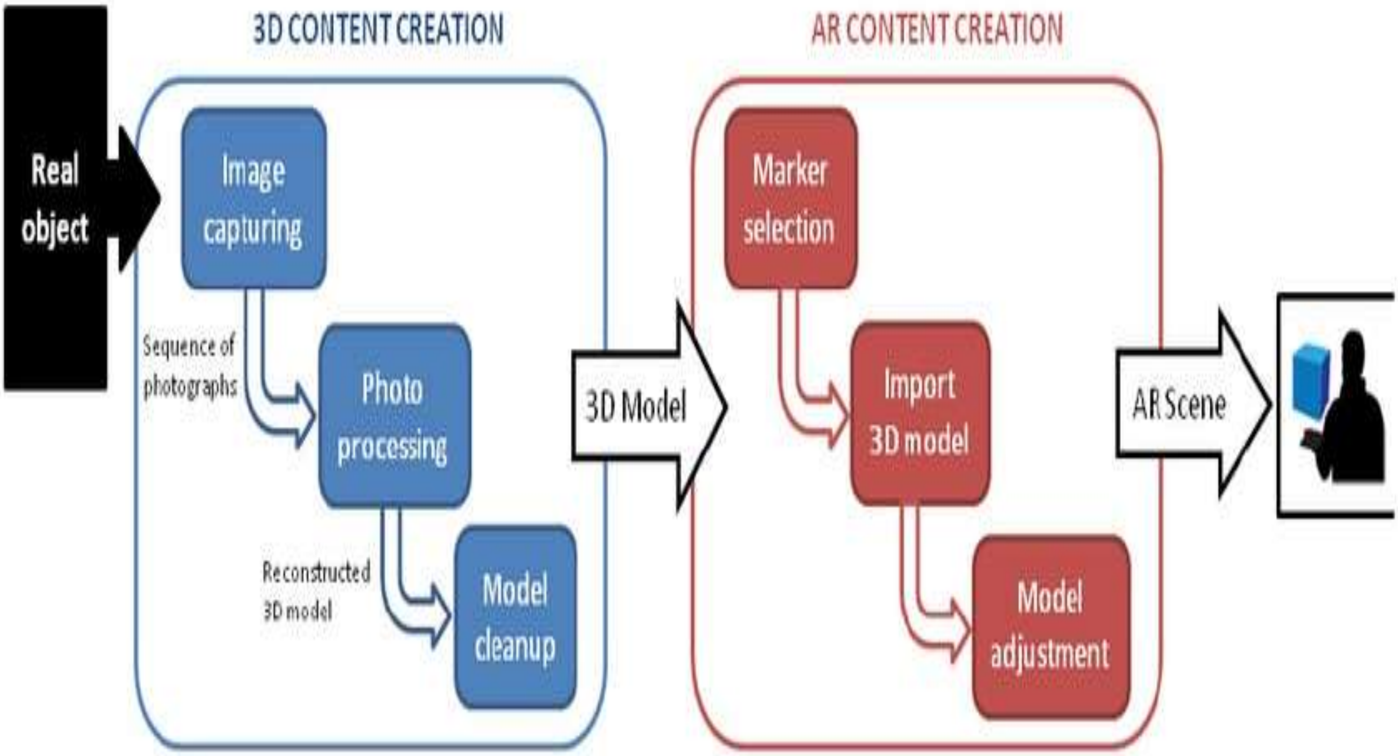


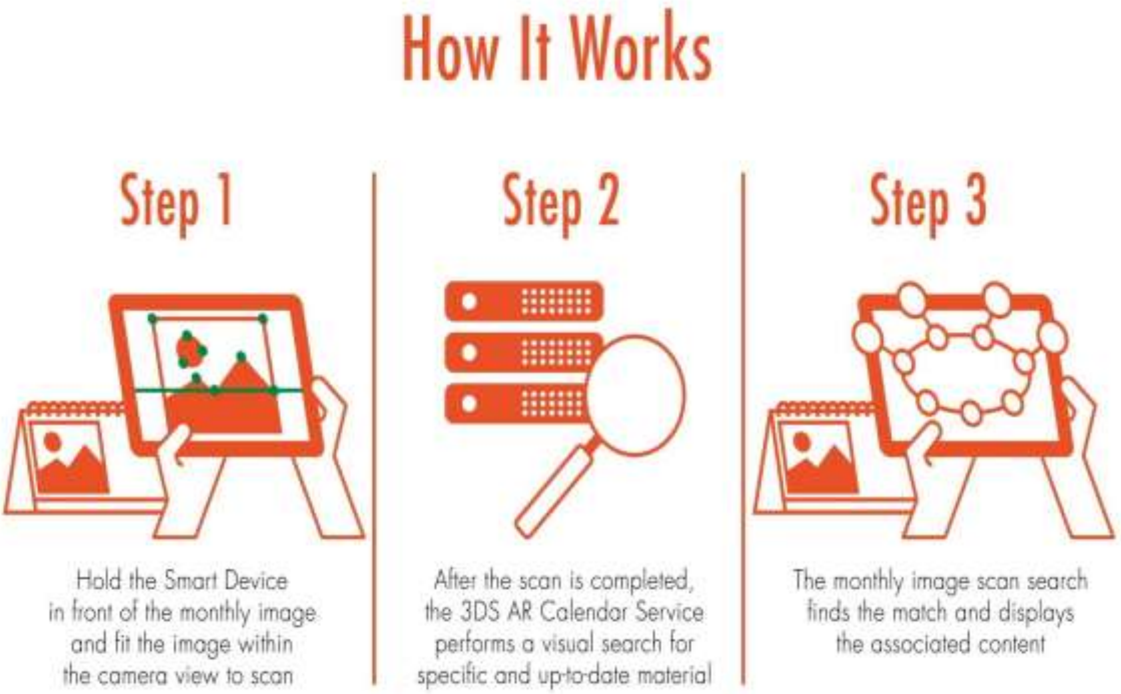
But as you see here we are going to provide the operation on the ground such that there is no link between the device and the object and this is our main objective.

Methods and Approaches:

- It utilizes the front facing camera to track an image target.
- We develop our Project in such a way that Front camera of Laptop or smart phone tracks the cards in the ground and automatically objects get projected from those cards and all the buttons and health factor of objects get projected.
- Then the battle between objects begins.
- It is a Dual Player game such that one person controls the one object and other person controls other object.
- They both simultaneously operate the buttons of their objects to kill the other object by firing at it.
- If the health factor of one object finishes before the other object then the other person is considered as winner.
- This is all about our Game.

Overview organization of the Proposed Work (Context Diagram)





How It Works

Step 1

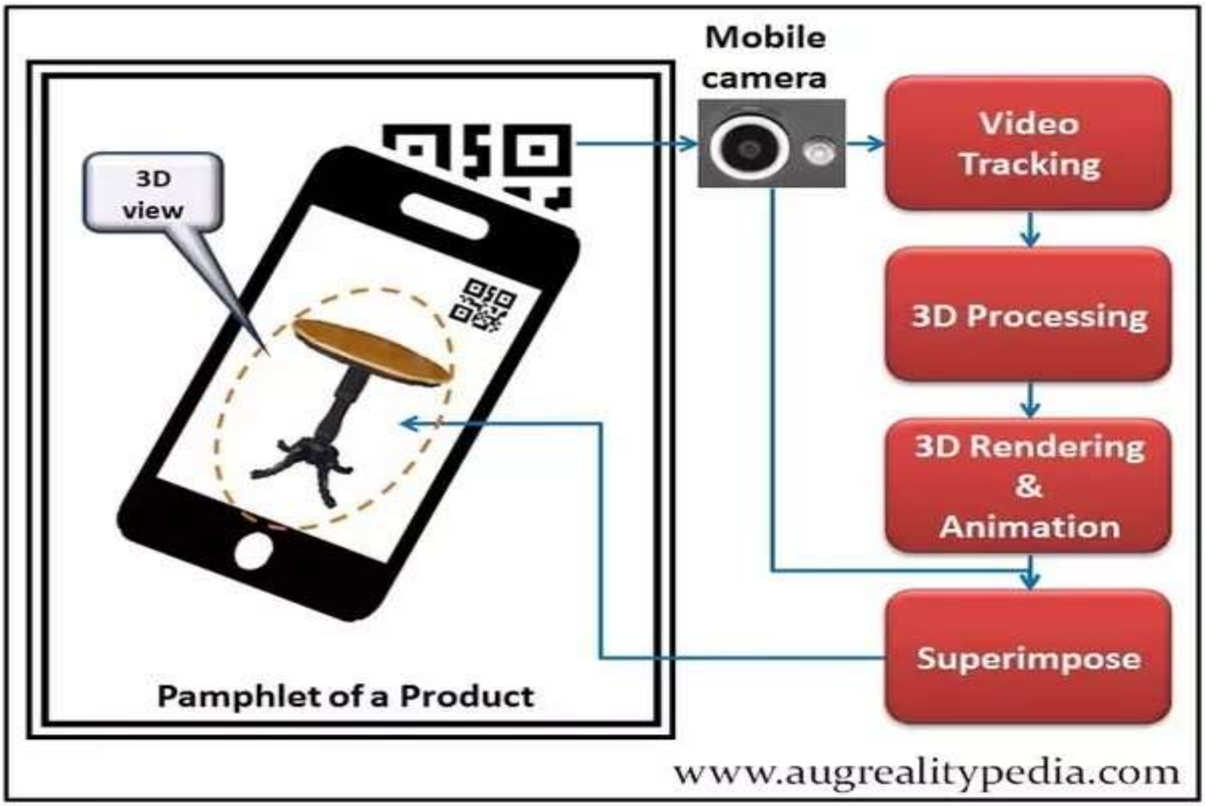
Hold the Smart Device in front of the monthly image and fit the image within the camera view to scan

Step 2

After the scan is completed, the 3DS AR Calendar Service performs a visual search for specific and up-to-date material

Step 3

The monthly image scan search finds the match and displays the associated content



Tools Used:

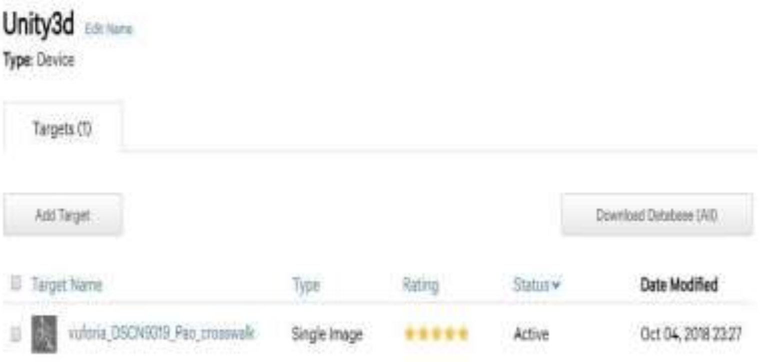
Main Platform for our Project is Unity 3D Software.

We created a Vuforia login account and added a database (image targets) in it.

We downloaded Vuforia Extension Package and imported in Unity.

We also downloaded 3D models and Image targets.

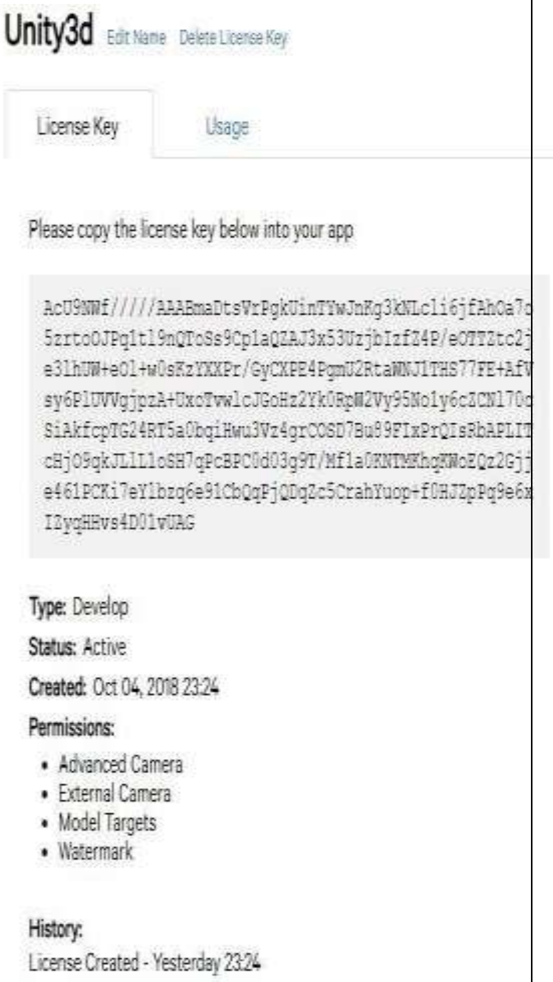
To achieve Augmented Reality, We needed a license key from Vuforia to add it in Vuforia Configuration.



Unity3d Edit Name

Type: Device

Target Name	Type	Rating	Status ▾	Date Modified
vuforia_DSCN9019_Pao_crosswalk	Single Image	★★★★★	Active	Oct 04, 2018 23:27



Unity3d Edit Name Delete License Key

Please copy the license key below into your app

```
AcU9NWf/////AAABmaDtsVrPgkUinTYwJnKg3kNLcli6jfAhOa7c
5zrtoOJPq1tl9nQToSs9Cp1aQZAJ3x53UzjbIzfZ4P/eOTTZtc2j
e3lhUW+eOl+w0sKzYXXPr/GyCXPE4PgmU2RtaWNJ1THS77FE+AfV
sy6PlUVVgjpzA+UxcTvwlcJGoHz2Yk0RpW2Vy95No1y6cZCN170c
SiAkfcpTG24RT5a0bqiHwu3Vz4grCOSD7Bu89FIxPrQIsRbAPLIT
cHj09gkJLlLloSH7qPcBPC0d03g9T/Mf1a0KNTMKhgKWoEQz2Gjj
e461PCKi7eYlbzq6e91CbQqPjQDqZc5CrahYuop+f0HJZpPq9e6x
IZyqHEvs4D01vUAG
```

Type: Develop

Status: Active

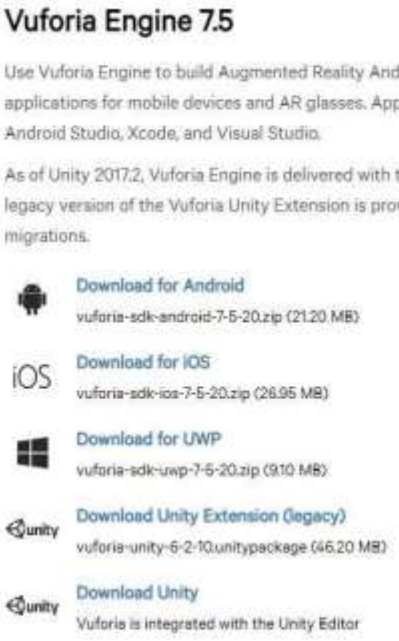
Created: Oct 04, 2018 23:24

Permissions:

- Advanced Camera
- External Camera
- Model Targets
- Watermark

History:

License Created - Yesterday 23:24



Vuforia Engine 7.5

Use Vuforia Engine to build Augmented Reality Android, iOS, and UWP applications for mobile devices and AR glasses. Apps can be built with Unity, Android Studio, Xcode, and Visual Studio.

As of Unity 2017.2, Vuforia Engine is delivered with the latest version of Unity. A legacy version of the Vuforia Unity Extension is provided to assist with project migrations.

Download for Android
vuforia-sdk-android-7-5-20.zip (21.20 MB)

iOS Download for iOS
vuforia-sdk-ios-7-5-20.zip (26.95 MB)

Download for UWP
vuforia-sdk-uwp-7-5-20.zip (9.10 MB)

unity Download Unity Extension (legacy)
vuforia-unity-6-2-10.unitypackage (46.20 MB)

unity Download Unity
Vuforia is integrated with the Unity Editor

C# Script for left button of Blue Dragon:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;

public class script : MonoBehaviour,IVirtualButtonEventHandler {
    public GameObject Blue_Dragon;
    public GameObject button;
    public float xforce=20;
    public Rigidbody brb;


void Start ()
{
    button= GameObject.Find("lbb");
    button.GetComponent<VirtualButtonBehaviour>().RegisterEventHandl
er(this);
    brb = Blue_Dragon.GetComponent<Rigidbody> ();
}

public void OnButtonPressed(VirtualButtonAbstractBehaviour vb)
{
    Debug.Log("Button Pressed lbb");
    brb.AddForce(xforce * Time.deltaTime, 0, 0,ForceMode.VelocityCha
nge);
}
public void OnButtonReleased(VirtualButtonAbstractBehaviour vb)
{
    Debug.Log("Button released");
}
void Update()
{

}

}
```

C# Script for Right button of Blue Dragon:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;

public class rbscript : MonoBehaviour,IVirtualButtonEventHandler
{
    public GameObject Blue_Dragon;
    public GameObject button2;
    public float xforce=20;
    public Rigidbody brb2;


    void Start ()
    {
        button2= GameObject.Find("rbb");
        button2.GetComponent<VirtualButtonBehaviour>().RegisterEvent
Handler(this);
        brb2 = Blue_Dragon.GetComponent<Rigidbody> ();
    }
    public void OnButtonPressed(VirtualButtonAbstractBehaviour vb)
    {
        Debug.Log("Button Pressed rbb");
        brb2.AddForce(-
xforce * Time.deltaTime, 0, 0,ForceMode.VelocityChange);
    }
    public void OnButtonReleased(VirtualButtonAbstractBehaviour vb)
    {
        Debug.Log("Button released");
    }
    void Update()
    {

    }
}
```

C# Script for left button of Red Dragon:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;

public class lrscript : MonoBehaviour,IVirtualButtonEventHandler {
    public GameObject Red_Dragon;
    public GameObject button3;
    public float xforce=20;
    public Rigidbody brb3;


    void Start ()
    {
        button3= GameObject.Find("lrb");
        button3.GetComponent<VirtualButtonBehaviour>().RegisterEvent
Handler(this);
        brb3 = Red_Dragon.GetComponent<Rigidbody> ();
    }

    public void OnButtonPressed(VirtualButtonAbstractBehaviour vb)
    {
        Debug.Log("Button Pressed lrb");
        brb3.AddForce(-
xforce * Time.deltaTime, 0, 0,ForceMode.VelocityChange);
    }
    public void OnButtonReleased(VirtualButtonAbstractBehaviour vb)
    {
        Debug.Log("Button released");
    }
    void Update()
    {

    }
}
```

C# Script for Right button of Red Dragon:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;

public class rrscript : MonoBehaviour,IVirtualButtonEventHandler {
    public GameObject Red_Dragon;
    public GameObject button4;
    public float xforce=20;
    public Rigidbody brb4;


    void Start ()
    {
        button4= GameObject.Find("rrb");
        button4.GetComponent<VirtualButtonBehaviour>().RegisterEvent
Handler(this);
        brb4 = Red_Dragon.GetComponent<Rigidbody> ();
    }

    public void OnButtonPressed(VirtualButtonAbstractBehaviour vb)
    {
        Debug.Log("Button Pressed rrb");
        brb4.AddForce(xforce * Time.deltaTime, 0, 0,ForceMode.Veloci
tyChange);
    }
    public void OnButtonReleased(VirtualButtonAbstractBehaviour vb)
    {
        Debug.Log("Button released");
    }
    void Update()
    {

    }
}
```

C# Script for Shoot button of Red Dragon:

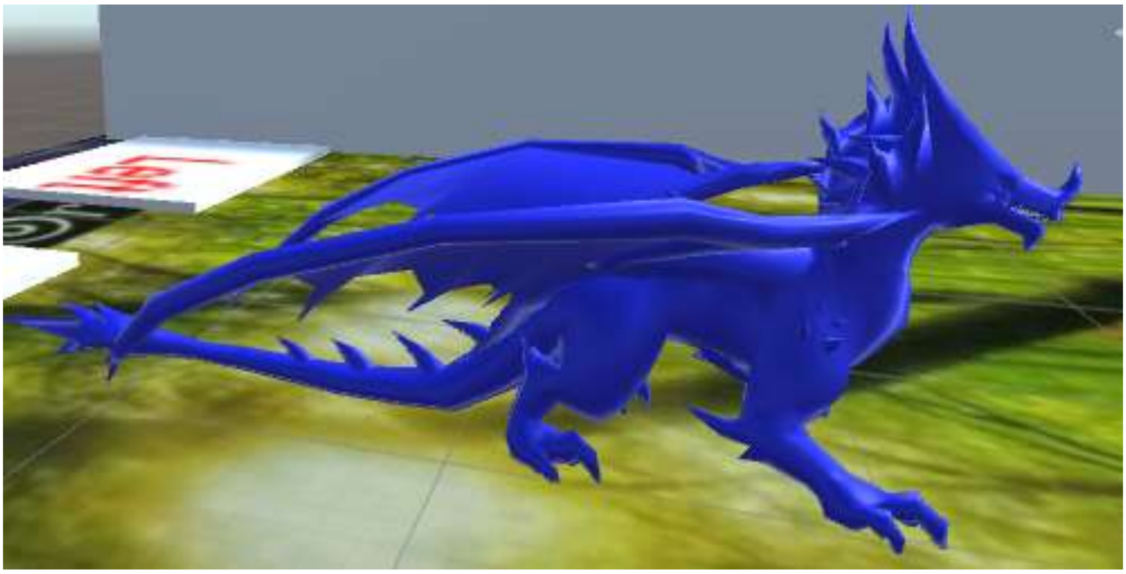
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Networking;
using Vuforia;

public class srscript : MonoBehaviour,IVirtualButtonEventHandler
{
    public GameObject bulletPrefab;
    public Transform Red_Dragon;
    public GameObject button6;
    public Rigidbody brb6;


    void Start ()
    {
        button6= GameObject.Find("srb");
        button6.GetComponent<VirtualButtonBehaviour>().RegisterEvent
Handler(this);
        brb6 = Red_Dragon.GetComponent<Rigidbody> ();

    }

    public void OnButtonPressed(VirtualButtonAbstractBehaviour vb)
    {
        Fire ();
    }
    void Fire()
    {
        // Create the Bullet from the Bullet Prefab
        var bullet = (GameObject)Instantiate(
            bulletPrefab,
            Red_Dragon.position,
            Red_Dragon.rotation);

        // Add velocity to the bullet
        bullet.GetComponent<Rigidbody>().velocity = bullet.transform
.forward * 3;

        // Destroy the bullet after 2 seconds
        Destroy(bullet, 2.0f);
    }
    public void OnButtonReleased(VirtualButtonAbstractBehaviour vb)
    {
        Debug.Log("Button released");
    }}
```

C# Script for Shoot button of Blue Dragon:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Networking;
using Vuforia;

public class sbscript : MonoBehaviour,IVirtualButtonEventHandler
{
    public GameObject bulletPrefab;
    public Transform Blue_Dragon;
    public GameObject button5;
    public Rigidbody brb5;


    void Start ()
    {
        button5= GameObject.Find("sbb");
        button5.GetComponent<VirtualButtonBehaviour>().RegisterEvent
Handler(this);
        brb5 = Blue_Dragon.GetComponent<Rigidbody> ();

    }

    public void OnButtonPressed(VirtualButtonAbstractBehaviour vb)
    {
        Fire ();
    }
    void Fire()
    {
        // Create the Bullet from the Bullet Prefab
        var bullet = (GameObject)Instantiate(
            bulletPrefab,
            Blue_Dragon.position,
            Blue_Dragon.rotation);

        // Add velocity to the bullet
        bullet.GetComponent<Rigidbody>().velocity = bullet.transform
.forward * 3;

        // Destroy the bullet after 2 seconds
        Destroy(bullet, 2.0f);
    }
    public void OnButtonReleased(VirtualButtonAbstractBehaviour vb)
    {
        Debug.Log("Button released");
    }}
```

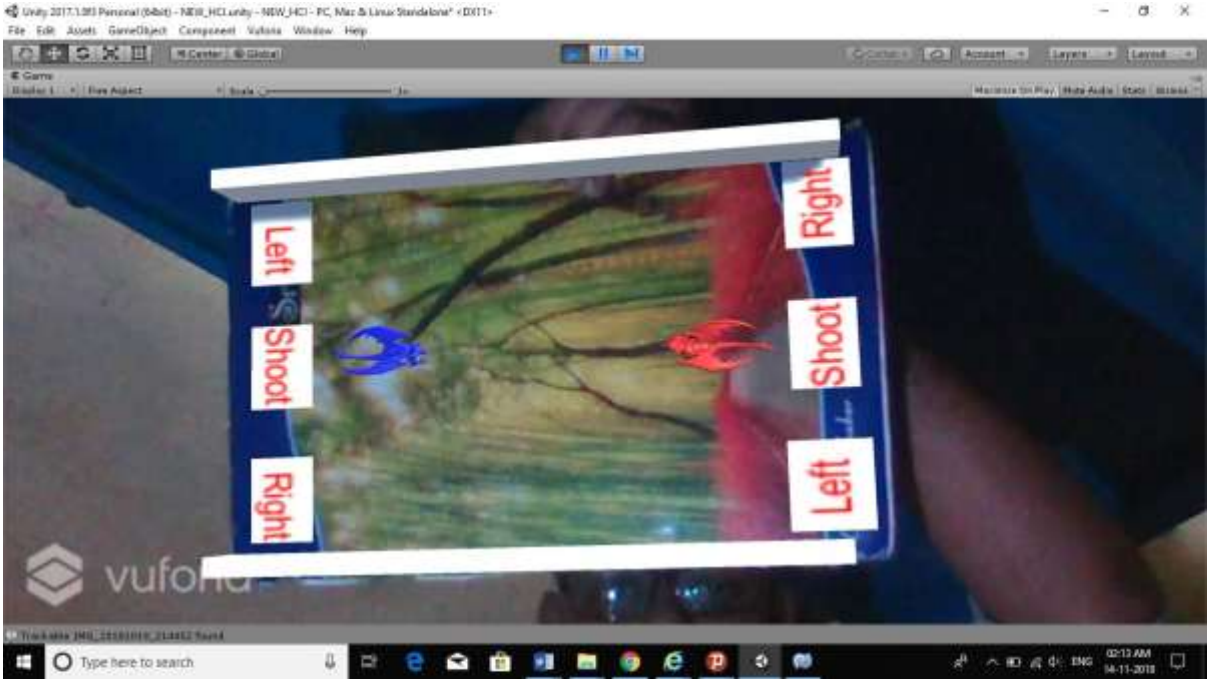
3D Models

Blue Dragon:



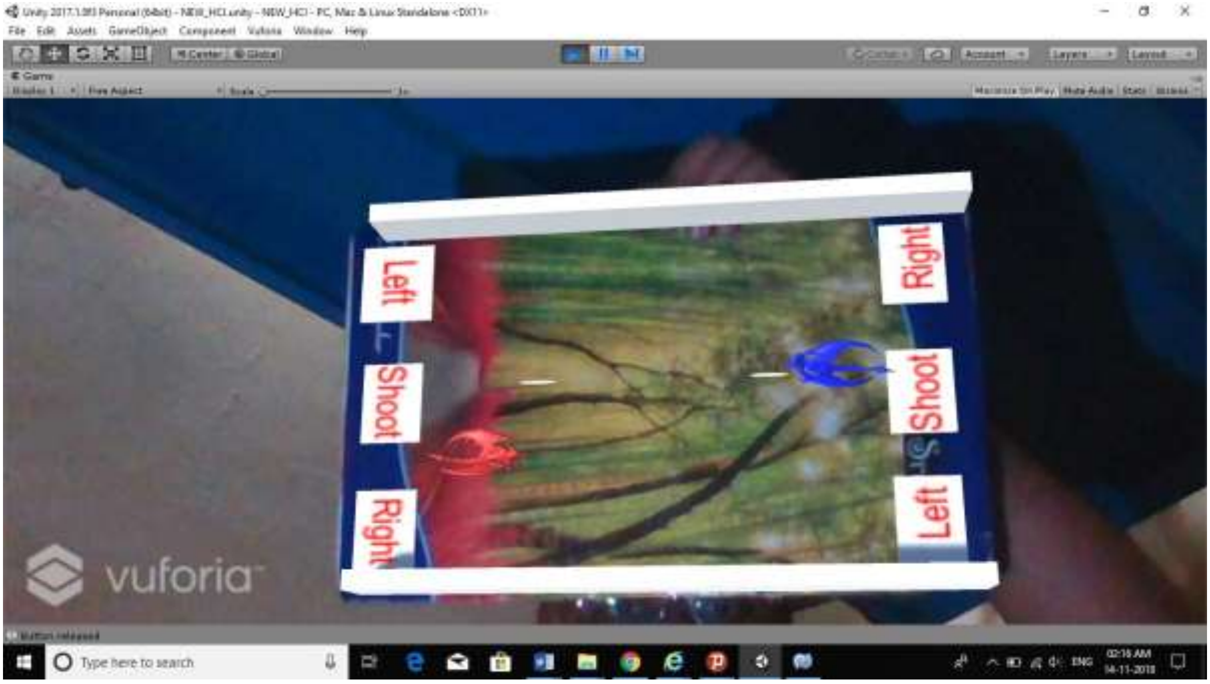
Red Dragon:

Results and Discussion



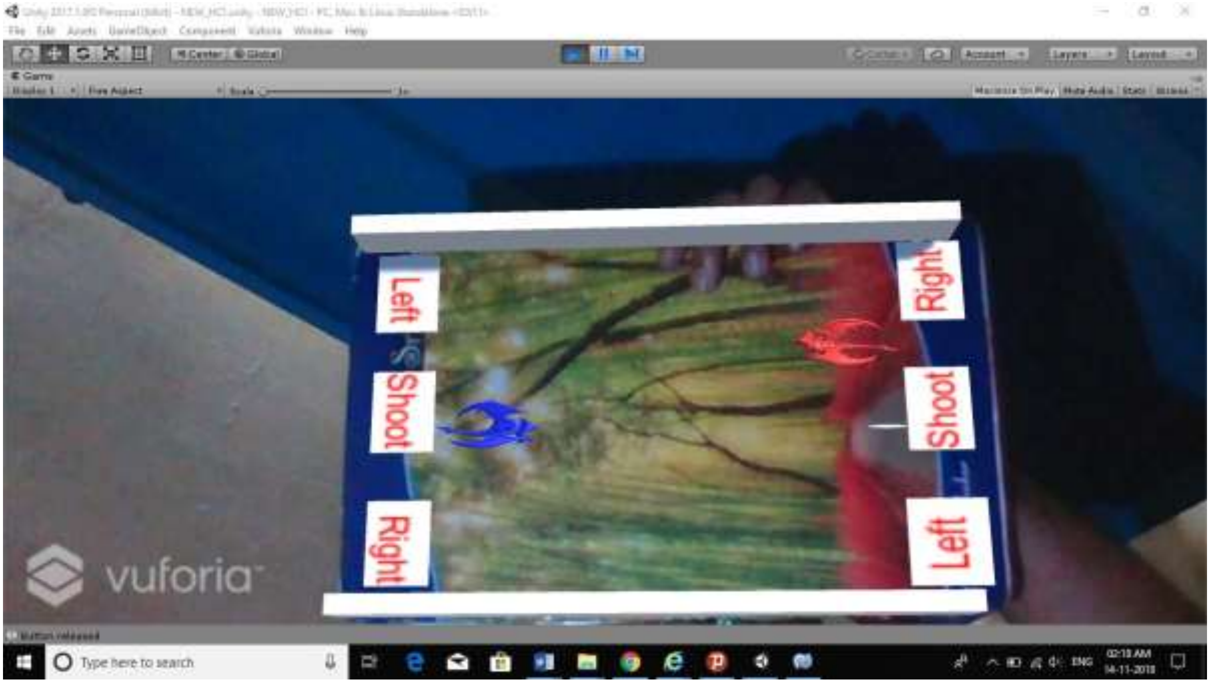
Unity 2017.1.0f3 Personal (64bit) - NEW_HCI.unity - NEW_HCI - PC, Mac & Linux Standalone* <DX11>

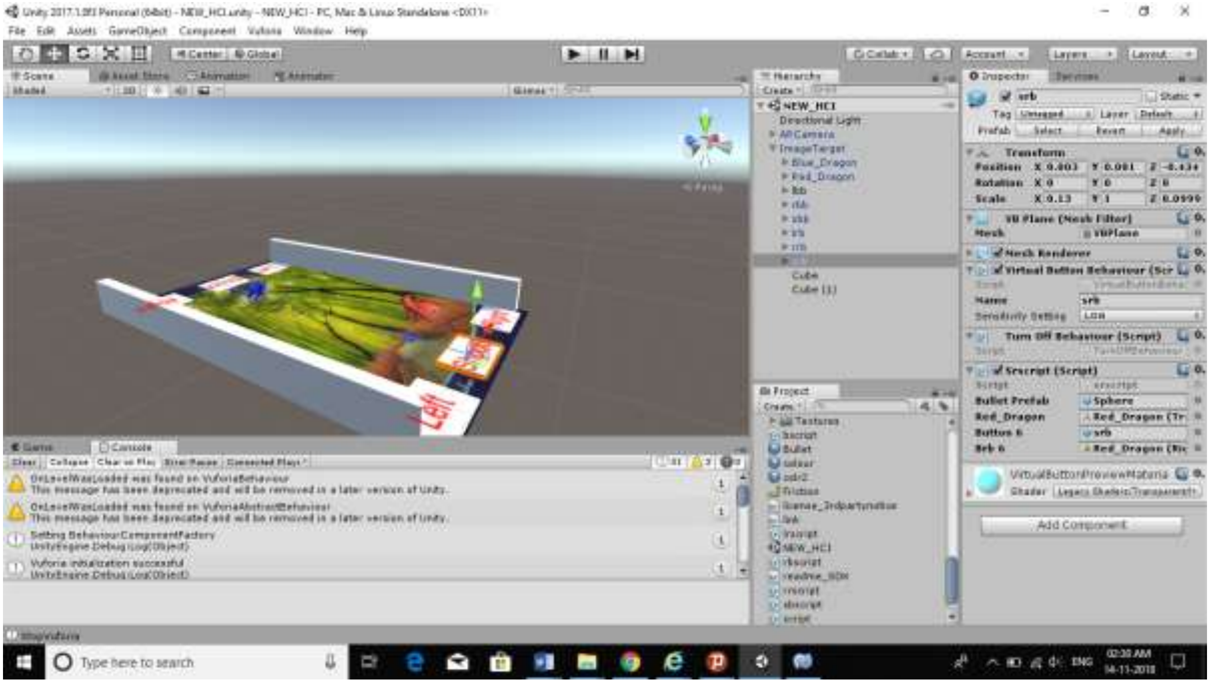
Trackable IMG_[illegible] found



Unity 2017.1.0f3 Personal (64bit) - NEW_HCI.unity - NEW_HCI - PC, Mac & Linux Standalone <DX11>

Button released




Unity 2017.1.0f3 Personal (64bit) - NEW_HCI.unity - NEW_HCI - PC, Mac & Linux Standalone <DX11>

srb Static
Tag Untagged Layer Default
Transform
Position X 0.003 Y 0.081 Z -0.434
Rotation X 0 Y 0 Z 0
Scale X 0.13 Y 1 Z 0.0999
VB Plane (Mesh Filter)
Mesh VBPlane
Virtual Button Behaviour (Script)
Name srb
Sensitivity Setting LOW
Turn Off Behaviour (Script)
Srscript (Script)
Bullet Prefab Sphere
Red_Dragon Red_Dragon (Tr
Button 6 srb
Srb 6 Red_Dragon (Ri
VirtualButtonPreviewMaterial
Shader Legacy Shaders/Transparent/

```
OnLevelWasLoaded was found on VuforiaBehaviour
This message has been deprecated and will be removed in a later version of Unity.
OnLevelWasLoaded was found on VuforiaAbstractBehaviour
This message has been deprecated and will be removed in a later version of Unity.
Setting BehaviourComponentFactory
UnityEngine.Debug:Log(Object)
Vuforia initialization successful
UnityEngine.Debug:Log(Object)
```

Conclusion

Finally we came to know that Virtual Buttons are so sensitive that even light or shadow falls on it, button gets activated. So We need to be careful while running the project. And we used a book as an image target. It would be good if we have used some other pic as image target which will have good features to target. When that image is set to run, it tracks well and all augmented models will be easily projected from image target. That's all about our review.

We would like to thank our faculty Mr. Arun Kumar S for giving this chance and providing full freedom to develop this project. We hope that we utilized it well and gave you the best output.

Thank You

References

https://unity3d.com/learn/tutorials/projects/space-shooter-tutorial/creating-shots?playlist=17147

https://unity3d.com/learn/tutorials/projects/space-shooter/shooting-shots?playlist=17147

https://unity3d.com/learn/tutorials/topics/multiplayer-networking/creating-player-movement-single-player

https://www.youtube.com/watch?v=j48LtUkZRjU&list=PLPV2KyIb3jR5QFsefuO2RlAgWEz6EvVi6

https://answers.unity.com/questions/37195/compiler-errors.html