AUGMENTED REALITY ANDROID APP USING UNITY AND VUFORIA

ABSTRACT

In AR, a virtual environment is designed to coexist with the real environment, with the goal of being informative and providing additional data about the real world, which a user can access without having to do a search. For example, industrial AR apps could offer instant troubleshooting information when a handset is aimed at a piece of failing equipment. Virtual reality encompasses a complete environmental simulation that replaces the user's world with an entirely virtual world. Because these virtual environments are entirely fabricated, they are often designed to be larger than life. For example, VR could let a user box with a cartoon version of Mike Tyson in a virtual boxing ring.

Organisation Information:

HACKWIT is a professionally managed company with years of industry experience in developing and delivering Enterprise specific Software and Web development solutions using latest technologies. Quality is the buzz word in today's world without which no organization can survive. Along with quality we at HACKWIT. "Create Beyond" to take one step ahead and focus on Delivery of the solutions. We design processes that focus not just only on quality but also on delivery which increases the value to our global clients. Apart from training our employees on latest technologies, we also empower them to deliver exciting solutions to our clients. At the core HACKWIT operates in three specific domains namely Software Development, Website Design & Development and Blockchain Technology .We also see services in Database Administration services. Under each division we further provide specific industry solutions on focused domains. We emphasize on building relationships with our clients by delivering projectson time and within budget.

Programs and opportunities:

This ground up approach helps us deliver not only the solution to our clients but also add value to at the core HACKWIT operates in three specific domains namely Software Development, Website Design & Blockchain Technology. We also offer our services in building Database Administration services. Under each division we further provide specific industry solutions on focused domains with cutting edge technologies. We emphasize on building relationships withour clients by delivering projects on time and within budget.

Methodologies:

We follow a structured methodology for our projects which starts from designing the solution to the implementation phase. Well planned Project reduces the time to deliver the project and any additional ad-hoc costs to our clients, hence we dedicate majority of our time understanding our clients business and gather requirements. This ground up approach helps us deliver not only the solution to our clients but also add value to your investments.

Key parts of the report:

Under each division we further provide specific industry solutions on focused domains with cutting edge technologies.

Benefits of the Company/Institution through our report:

Under each division we further provide specific industry solution on focused domains with cutting edge technologies. We emphasize on building relationships with our clients by delivering projects on time and within budget.

CONDENTS

INTRODUCTION

Augmented Reality (AR) is a perfect blend of the digital world and the physical elements to create an artificial environment. Apps which are developed using AR technology for mobile or desktop to blend digital components into the real world. The full form of AR is Augment Reality. Example: AR technology helps to display score overlays on telecasted sports games and pop out 3D photos, text messages, and emails. **Virtual Reality (VR)** is a computer-generated simulation of an alternate world or reality. It is used in 3D movies and video games. It helps to create simulations similar to the real world and "immerse" the viewer using computers and sensory devices like headsets and gloves. Apart from games and entertainment, virtual reality is also used for training, education, and science. The full form of VR is Virtual reality.

AR uses computer vision, mapping as well as depth tracking in order to show appropriate content to the user. This functionality allows cameras to collect, send, and process data to show digital content appropriate to what any user is looking at. In Augmented reality, the user's physical environment is enhanced with contextually relevant digital content in real-time. You can experience (AR) augmented reality with a smartphone or with special hardware.

The focus of virtual reality is on simulating the vision. The user needs to put a VR headset screen in front of his/her eyes. Therefore, eliminating any interaction with the real world. In VR, two lenses are placed between the screen. The user needs to adjust eyes based on the individual movement of the eye and its positioning. Uses goggles, speakers, and sometimes handheld wearables to simulate a real-world experience. In virtual reality, you can also employ visual, auditory, and haptic (touch) stimulation, so the constructed reality is immersive.

Game Development can be undertaken by a large Game Development Studio or by a single individual. It can be as small or large as you like. As long as it lets the player interact with content and is able to manipulate the game's elements, you can call it a 'game'. To get involved in the Game Development process, you do not need to write code. Artists may create and design assets, while a Developer might focus on programming a health bar. A Tester may get involved to see that the game works as expected.

To resolve problems that game frameworks had, tools like libGDX and OpenGL were developed. They helped game development to be a lot faster and easier, providing lots of pre-made functions and features. However, it was still hard to enter the industry or understand a framework for someone coming from a nonprogrammer background, a common case in the game development scene.

Mixed Reality

Mixed Reality (MR) is the newest and most unique technology in the XR family. It can be thought of as a hybrid between VR and AR, although some people prefer to call it another form of AR. Like VR, MR requires the user to wear a specialized set of goggles and, like AR, MR glasses still allow the user to see the world around them. With Mixed Reality, the real and digital worlds blend seamlessly together as interfaces are projected through the glasses. Since the device is worn on the user's head, one can simply use hand gestures to interact with the content and navigate digital screens. MR is often referred to as hands-free computing since the device is more powerful then the average laptop which allows a user to perform almost any task a computer could.

How is VR being used today?

- Training Heavy-Duty Crane Operators
- Training Nuclear Facilities Operators
- VR for Manufacturing Training
- General Safety & Hazard Awareness Training
- Replicating Real-Life, High-Risk Scenarios With VR in Aviation
- XR for Remote Collaboration & Design

Summary & Recommendations

As demonstrated by the wide range of use cases throughout this report, XR technology has already exploded in popularity amongst many mid to large sized companies across the globe for things like education and training to productivity and performance. Although the technology is relatively new to the food and beverage sector with only a few known brands leveraging it today, XR has the potential to play a critical role in supporting companies of all sizes within the industry. In order to minimize risk and maximize the potential benefit for companies in the sector we recommend the following:

GET TO KNOW YOUR END USER AND THE PROBLEMS YOU ARE TRYING TO SOLVE: Before you dive into the deep end, it’s critical you understand who your audience is and the problems you’re trying to solve. There is a wide range of XR technologies and applications available for businesses to utilize. It’s important to understand that some populations will be much more receptive to using new technologies than others. XR can be incredibly beneficial for youth, people with disabilities or those who struggle to absorb knowledge through traditional teaching methods such as sitting in a classroom or reading training manuals. Understanding who the end user will be and including them in the conversation early on will help to identify considerations during the planning process.

WORK WITH XR EXPERTS TO HELP YOU NAVIGATE THE TECHNOLOGY: At the rapid evolution rate of technology, there will probably be a dozen new applications and XR devices on the market by the time you read this. Regardless of how tech savvy you and you r team might be, consider working with XR companies and industry experts to help quickly navigate the landscape and focus on the most impactful areas. Many companies will offer free consultation that could also help you quickly understand if XR is a viable solution. The last thing you want is to purchase new XR hardware or buy applications that end up sitting in a closet due to unexpected hurdles.

GOVERNMENT, INDUSTRY AND ACADEMIA SHOULD EXPLORE MORE COLLABORATIONS: It's no surprise that developing high end XR experiences can be costly and, therefore, a luxury. Only larger companies have been able to afford it. To have widespread impact with XR in the food and beverage industry, a collaborative approach will need to be taken. Government, industry, and academia will need to work together to identify critical problems faced by the industry, such as widespread labour shortages or specific skill gaps. Once these problems are identified, a consortium of partners can work together to build custom tools to be implemented by companies of all sizes. Take something like food safety for example, a fundamental program that hundreds of thousands of employees need to take prior to working in food and beverage. Although the upfront costs of developing a program like this might be high, the vast number of participants that would benefit from a virtual reality food safety course quickly outweighs the upfront cost. A single employer wouldn't be able to justify developing this alone but working together to share costs could make XR accessible to even the smallest of companies and create widespread benefit throughout the industry.

Project Overview:

The codeless template provides all the assets that are used in this example, including the image file for the playing card used. We can, use our own image target. All that remains to be set up in this project is the AR configuration and the scripts. Within the Assets window are several other folders, each containing the types of assets specified. Within the scene, are some buttons and text along with a chest. The chest is the object that will appear whenever the app locates a real-world queen of hearts playing card. It's incredibly small, and that's because the image target object that will be created later is also quite small. This is to reflect the real-world size of the card. If we were to leave the chest's scale at the default value of one, it would appear incredibly large in the app once the playing card was found. With this adjustment, it will appear as though it is roughly the same size as the card.

SYSTEM ANALYSIS

REQUIREMENT ANALYSIS :

Unity Engine:

Helping developers create great games that reach players where they are has always been and will stay at the heart of what we do. Our goal is to make the tools that make it easier for developers to realize their vision – and that includes supporting that vision with resources to help them turn their games into sustainable businesses, if and when they choose to do so. The combination of Unity and iron Source is transformational in that it will give mobile game developers the tools they need at each stage of their development journey: from building, publishing, and operating mobile games to monetizing them, if they choose to, and growing their player base across multiple channels. The combination of Unity and iron Source is transformational in that it will give mobile game developers the tools they need at each stage

of their development journey: from building, publishing, and operating mobile games to monetizing them, if they choose to, and growing their player base across multiple channels.

1. SOFTWARE REQUIREMENTS SPECIFICATIONS

System configurations

The software requirement specification can produce at the culmination of the analysis task. Thefunction and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, a representation of system behavior, and indication of performance and design constrain, appropriate validate criteria, and other information pertinent to requirements.

Software Requirements:

• Operating system	: Windows 10.
• Unity Version	: 2017 or above
• Game Interface	: Unity game engine.

Hardware Requirement:

- System : AMDA9
- Hard Disk : 1TB
- Ram : 8GB.

TECHNOLOGY

C#(C Sharp):

C# is one of the most used languages for VR development, and it's all thanks to Unity. Unity initially started as a game development framework but over the past few years it has begun to slowly transition to an all-purpose media creation toolBesides that, Unity offers many tutorials and examples on how to code a VR experience. Anything that Unity can't provide in terms of learning can be easily addressed by tapping into its large community.

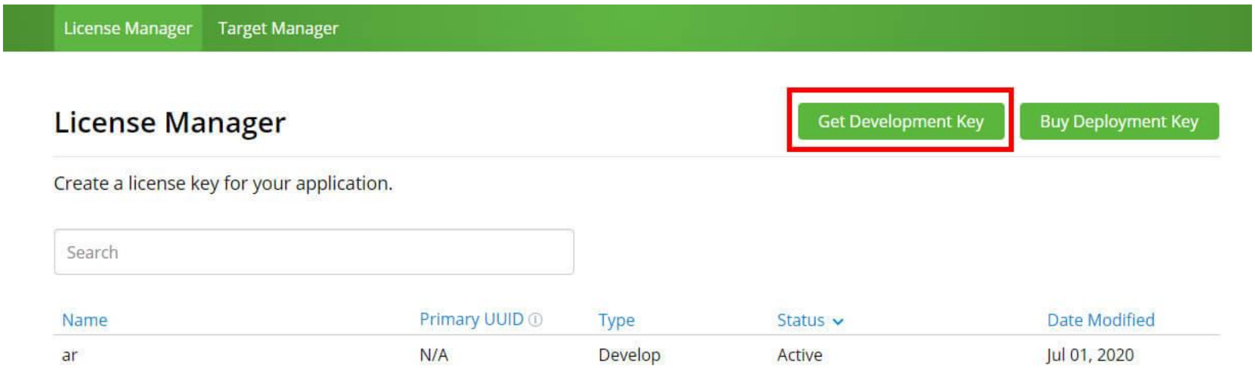
C# doesn't live in the VR space out of Unity's back alone. **StereoKit**, for example, is a C# library for creating VR applications that you can implement in your existing project. Coupled with .NET's powerful libraries, you can create VR experiences much more easily.If you're looking for a language with solid libraries and frameworks for VR development, C# is a good bet.

Mostly unity used **C#** programming language.

Vuforia Setup

Vuforia is an SDK for creating AR applications on mobile devices. It allows the application to detect real-world objects. The developer creates a database of "image targets" that will be fed into the Unity app. These image targets are, as you may have guessed, 2D images of what to look for in the real world.

To begin, first need a developer account on the Vuforia developer portal. After created the account and confirmed it, find the License Manager screen in the portal. The deployment key is what is needed if ever decide to publish an AR app with Vuforia, so it can be ignored.

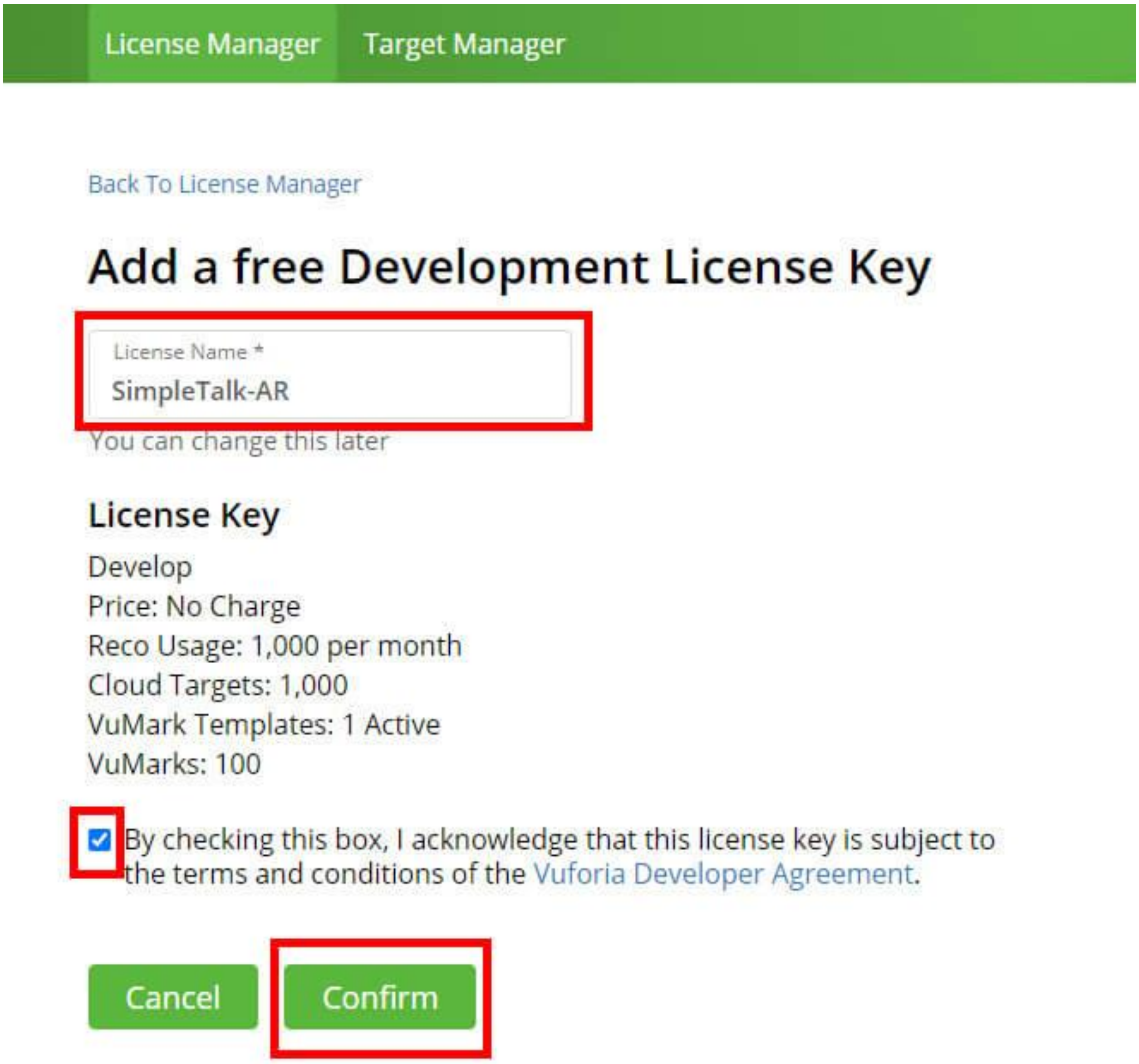


License Manager

Create a license key for your application.

Name	Primary UUID ⓘ	Type	Status ⌄	Date Modified
ar	N/A	Develop	Active	Jul 01, 2020

Figure 1: Getting a development key



Add a free Development License Key

License Name *
SimpleTalk-AR
You can change this later

License Key

Develop
Price: No Charge
Reco Usage: 1,000 per month
Cloud Targets: 1,000
VuMark Templates: 1 Active
VuMarks: 100

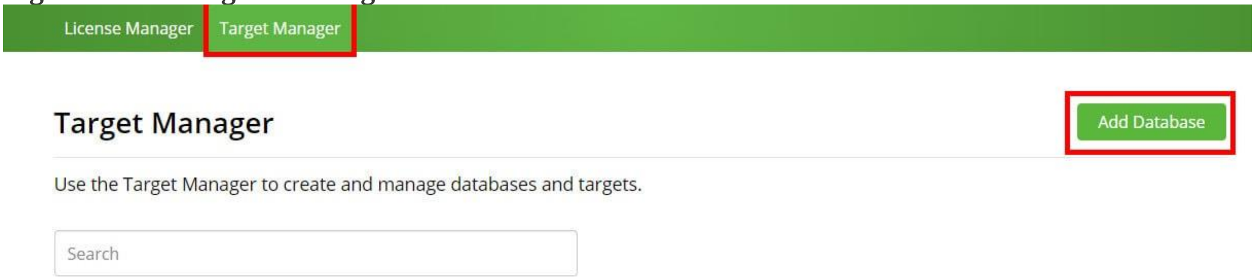
☑ By checking this box, I acknowledge that this license key is subject to the terms and conditions of the Vuforia Developer Agreement.

Figure 2: Creating a new license

SETUP DATABASE:

Now all the database needs is an image. The image can be anything . It’s only important that the image used is clear enough that Vuforia understands what object to look out for from the image. This example uses a playing card as the real-world item to look out for. The card image used can be found within the project files downloaded from the template. In Unity, there will be a folder named Card image in the Assets window that will contain the picture of the card. Right-click the card file and select Show in Explorer to locate the card in the file explorer. To upload this image, click on the name of the newly created database.

Figure 3: Creating a new target database

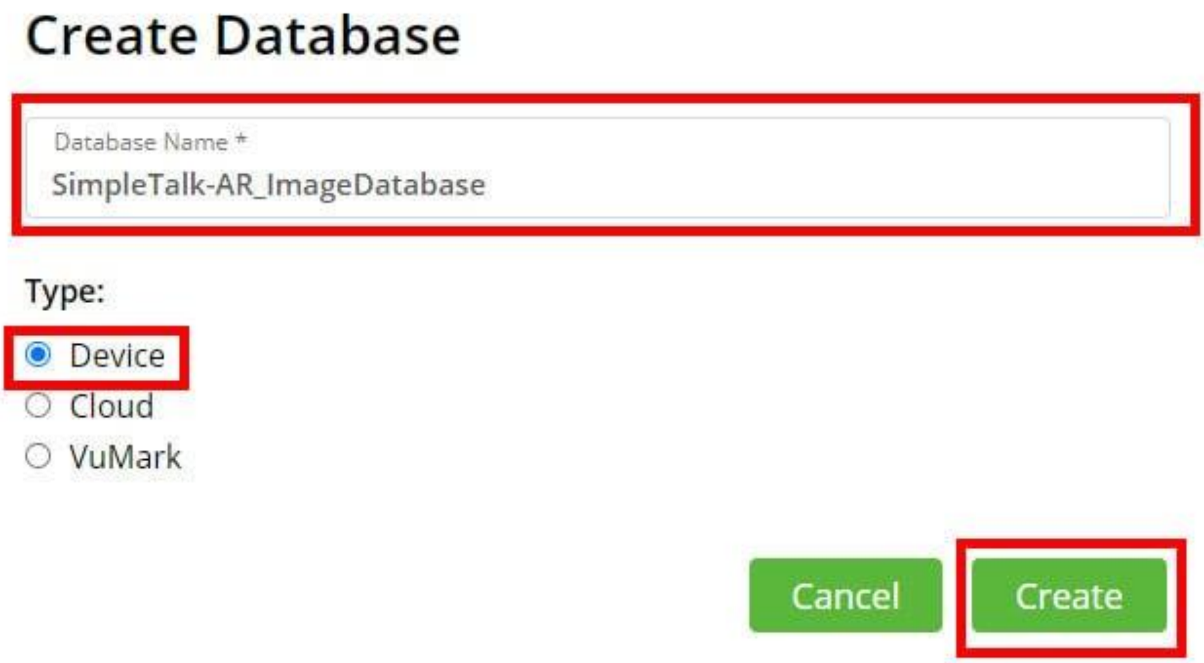


Target Manager

Use the Target Manager to create and manage databases and targets.

Search

Figure 4: Database creation



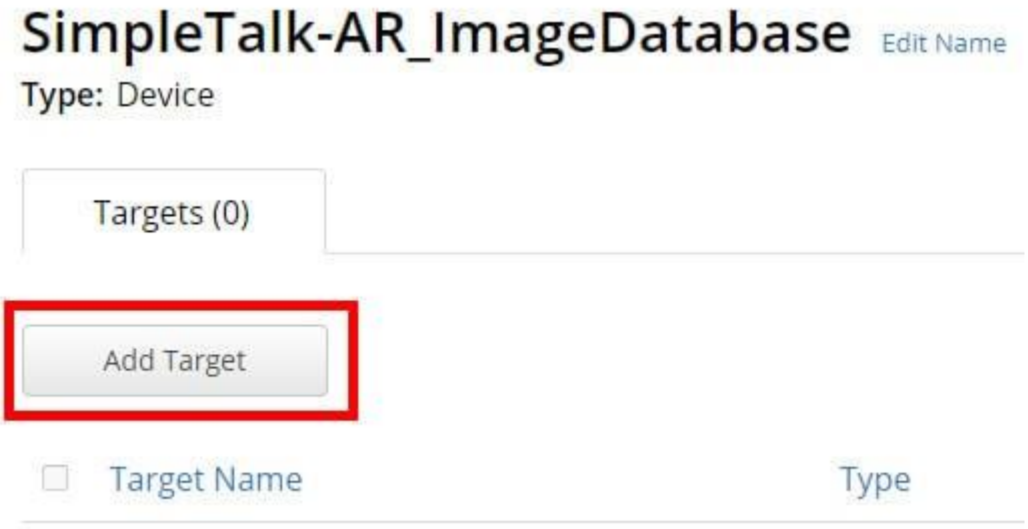
Create Database

Database Name *
SimpleTalk-AR_ImageDatabase

Type:

● Device
○ Cloud
○ VuMark

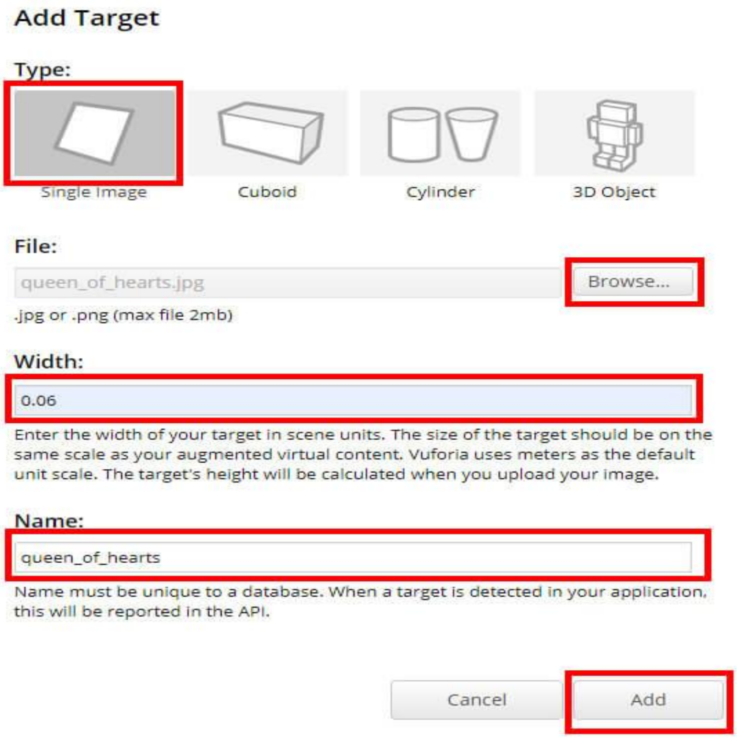
Figure 5: Adding a target to the database



SimpleTalk-AR_ImageDatabase

Type: Device

The target type specifies what sort of real-world object to look for. . By default, the type is a single image, which is what this app will use. Select the image file specify the width. The width is used to let the app know roughly how large to expect the target to be. It asks for the width in “scene units” which can change depending on the development software . Unity’s scene units are typically one meter per unit, so need to specify the width in meters of your target. In this , a typical playing card is about six centimeters wide so specify 0.06 for the width in Vuforia. The name can be whatever.



Add Target

Type: Single Image

File: queen_of_hearts.jpg

.jpg or .png (max file 2mb)

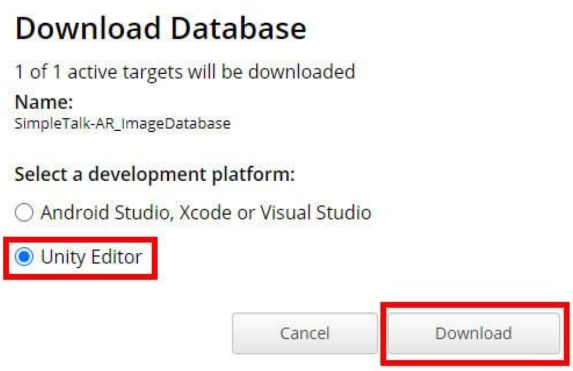
Width: 0.06

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name: queen_of_hearts

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Figure 6: Specifying the type, image, width, and name



Download Database

1 of 1 active targets will be downloaded

Name:
SimpleTalk-AR_ImageDatabase

Select a development platform:

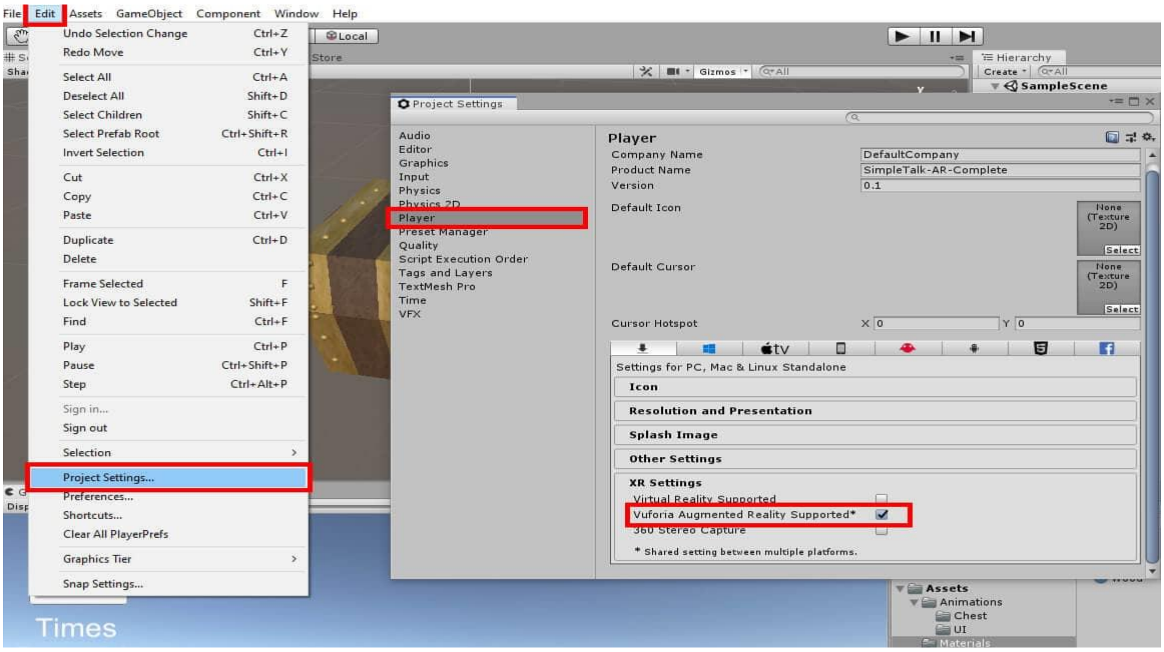
○ Android Studio, Xcode or Visual Studio

● Unity Editor

Figure 7: Download the database

Unity Project Setup

In order to develop an AR app, enable the Vuforia engine within Unity.



Project Settings → Player

Company Name: DefaultCompany

Product Name: SimpleTalk-AR-Complete

Version: 0.1

Settings for PC, Mac & Linux Standalone

XR Settings

☐ Virtual Reality Supported

☑ Vuforia Augmented Reality Supported*

☐ 360 Stereo Capture

* Shared setting between multiple platforms.

Figure 8: Enabling Vuforia

Now that Vuforia is being used, need apply the app key that were given on the developer portal and change the configuration. do this by creating an AR camera and clicking a button in the Inspector. To create the camera, first delete the Main Camera object. Then click the Create button in the Hierarchy window, then choose Vuforia Engine->AR Camera

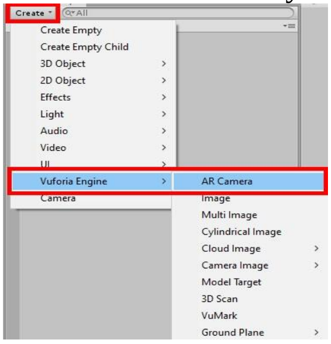


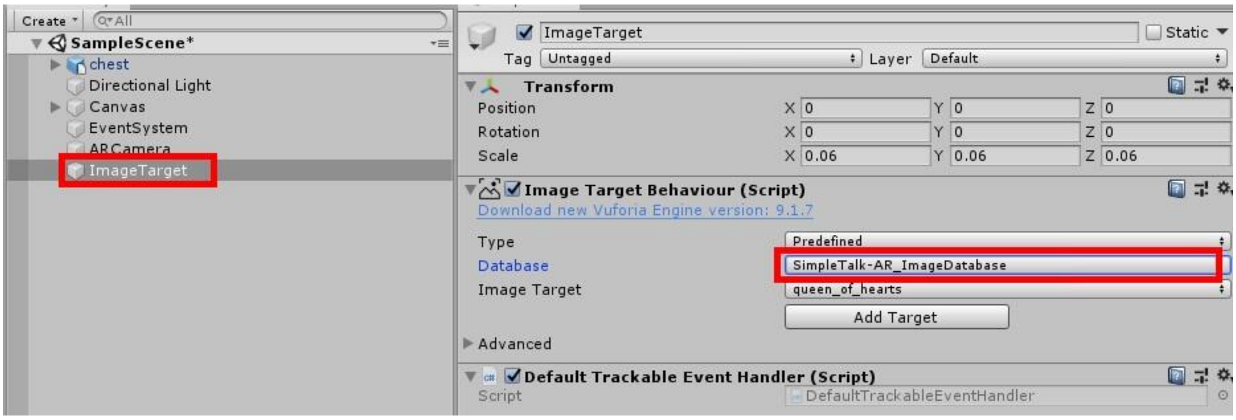
Figure 9: Creating an AR camera

After accepting a software license, the AR camera will be created. It’s possible that the camera might appear on top of the chest model or in another location normally considered undesirable. Now, what’s the difference between the two cameras? An AR camera is what’s used when the developer wants the device’s built-in camera to both view the real world as well as essentially turn the real world into the game world. Meanwhile, the default camera displays what’s currently in the “Unity world”.

With the AR Camera object selected, find a button in the Inspector that Open Vuforia Engine Configuration. Copy app key from the Vuforia portal (accessed by clicking the name of app in License Manager), then paste it into the App Key field in Unity. Then, scroll down until find an option that says Track Device Pose. Check that box, and the Vuforia engine configuration will be finished.

Vuforia what images to look out for in the real world. Now need to pass that same information into Unity. When finishing the Vuforia setup, downloaded a database, which came as a Unity package file. Open that file now by clicking Assets -> Import Package -> Custom Package. Then navigate to the file you downloaded to import the database into Unity.

Once that’s been completed, create another object. This time the object is an image target, created by clicking Image under the Vuforia Engine sub-menu in the Create menu. Because imported the Vuforia database just before this step, the Image Target should use database and have the correct scale by default. If it doesn’t, then you can click on the object in the Hierarchy and then look at the Inspector window and search for a field named Database.



Create ▾ | All

▼ SampleScene*
- ▶ chest
- Directional Light
- ▶ Canvas
- EventSystem
- ARCamera
- ImageTarget

☑ ImageTarget ☐ Static

Tag Untagged Layer Default

▼ Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	0.06	0.06	0.06

▼ ☑ Image Target Behaviour (Script)

Download new Vuforia Engine version: 9.1.7

Type: Predefined

Database: SimpleTalk-AR_ImageDatabase

Image Target: queen_of_hearts

Add Target

▶ Advanced

▼ ☑ Default Trackable Event Handler (Script)

Script: DefaultTrackableEventHandler

Figure 10: Confirming your database is being used

Unity now knows what object to look for, but it’s not aware of what will be shown when the playing card is found. This is easily fixed by dragging the chest object onto the Image Target to make it a child of the target. The template is already set up so that the chest will fit on the image correctly, but if it doesn’t then just change the scale of the chest until it’s about the size of the card itself.

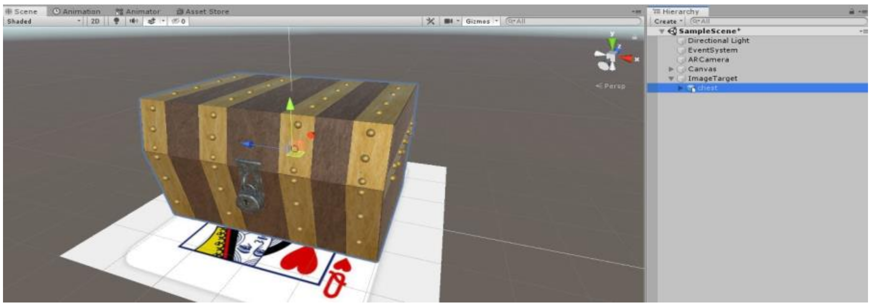


Figure 11: The chest should appear similar in size to the playing card.

All that remains is to set up collisions for the chest and camera. The chest collision should already be set up, so need only select the ARCamera object, click Add Component in the Inspector and add a Box Collider and Rigidbody. Box Collider, as the name implies, is the collision box for an object, while a Rigidbody us the physics component of an object. At least one of the colliding objects must have a Rigidbody for collision detection to work properly in Unity, and that's why one is being included in the camera object.

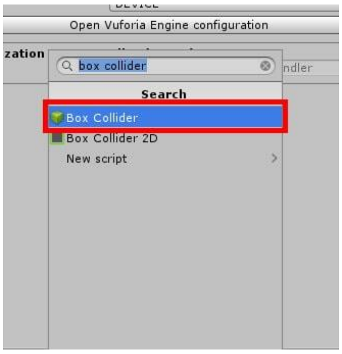
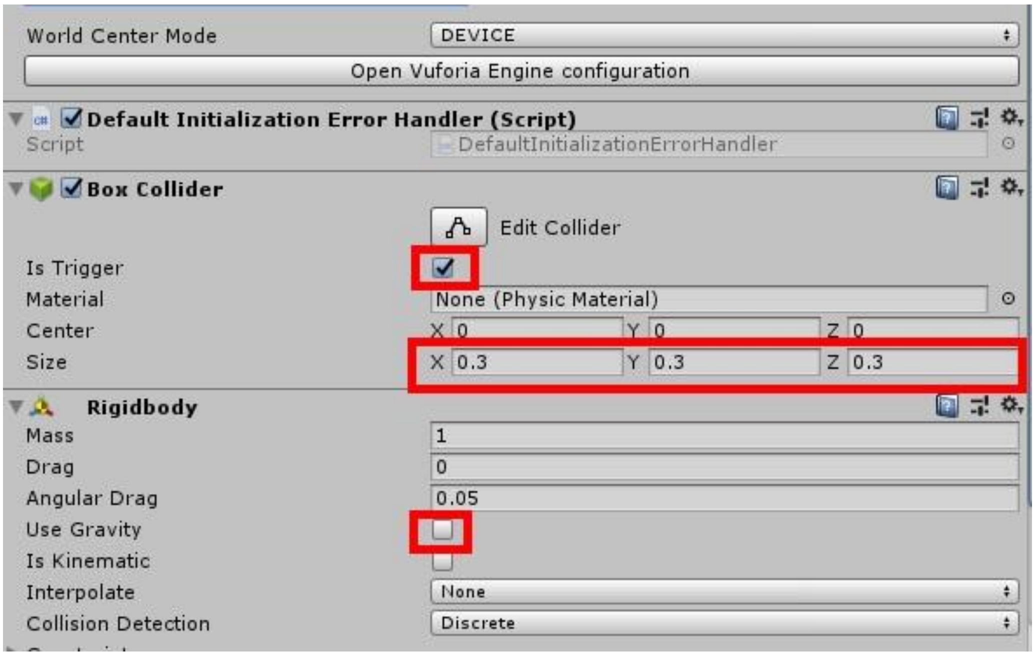


Figure 12: Adding a box collider



World Center Mode: DEVICE
Open Vuforia Engine configuration
☑ Default Initialization Error Handler (Script)
Script: DefaultInitializationErrorHandler
☑ Box Collider
Edit Collider
Is Trigger: ☑
Material: None (Physic Material)
Center: X 0 Y 0 Z 0
Size: X 0.3 Y 0.3 Z 0.3
Rigidbody
Mass: 1
Drag: 0
Angular Drag: 0.05
Use Gravity: □
Is Kinematic: □
Interpolate: None
Collision Detection: Discrete

Figure 13: Setting collider to trigger, changing size, and disabling *Use Gravity* on *ARCamera*

run this program now and find that it can already detect the card and spawn a chest on top of it. This is pretty cool, but it'd be even better if could interact with the object in some way. Plus, there are still those buttons that currently do nothing.

Three scripts resolve this issue.
PlayerInput performs actions when the user taps on the chest.
ChestCollision handles collision between the user and the chest.
GUIFunctions contains the functionality for the buttons.

PlayerInput

```
public class PlayerInput : MonoBehaviour, ITrackableEventHandler

private TrackableBehaviour mTrackableBehaviour;
public GameObject objectFoundText;
public Animator chestAnim;

void Start()
{
            mTrackableBehaviour = GetComponent<TrackableBehaviour>();
            if (mTrackableBehaviour)
                        mTrackableBehaviour.RegisterTrackableEventHandler(this);
}
Void Update()
{
if (Input.GetMouseButtonDown(0))
{
            Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
            if (Physics.Raycast(ray, out RaycastHit hit))
                        chestAnim.SetTrigger("Tapped");
}
}
public void OnTrackableStateChanged(
        TrackableBehaviour.Status previousStatus,
        TrackableBehaviour.Status newStatus)
{
            if (newStatus == TrackableBehaviour.Status.DETECTED ||
                        newStatus == TrackableBehaviour.Status.TRACKED ||
                        newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
                        objectFoundText.SetActive(true);
            else
                        objectFoundText.SetActive(false);
}
```

ChestCollision

```
public Text gameText;

void OnTriggerEnter(Collider col)
{
            if (col.CompareTag("MainCamera"))
                        gameText.text = "Near object";
}
void OnTriggerExit(Collider col)
{
            if (col.CompareTag("MainCamera"))
                        gameText.text = "Object Found!";
}
```

GUIFunctions

```
public Animator guiAnim;
public Text pressedText;
private int numOfPresses = 0;


public void ShowOrHideButton()
{
        guiAnim.SetTrigger("Visible");
}

public void ShowOrHideButton()
{
        guiAnim.SetTrigger("Visible");
}
public void ExitGame()
{
        Application.Quit();
}
```

Finishing Touches

The scripts should already be attached to their respective objects, except for ImageTarget as have created that object while following along need to attach PlayerInput to ImageTarget by clicking the object, then clicking Add Component and searching for PlayerInput. Once that's done, all that's needed is to fill the empty fields. Start with ImageTarget, which has the PlayerInput script. Find the ObjectLocatedText in the Hierarchy (located under Canvas) and drag that into Game Text. Then grab the chest (found under ImageTarget) object, which should be found under ImageTarget, and drag it into ChestAnim.

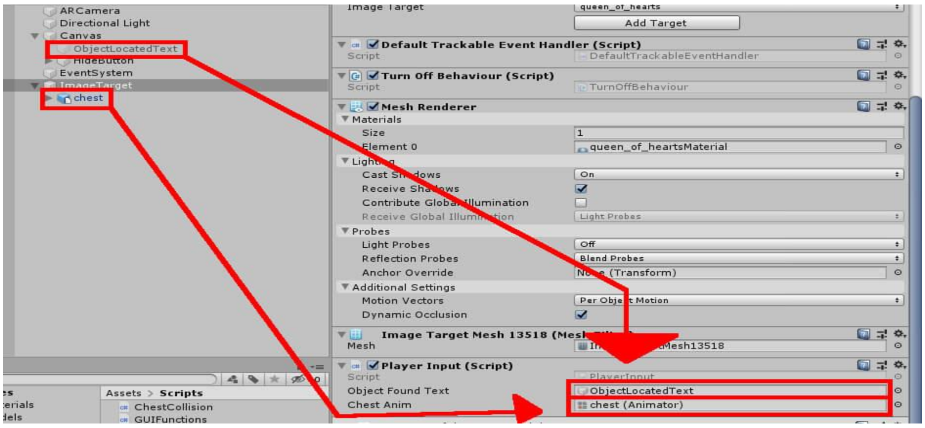


Image Target: queen_of_hearts

Default Trackable Event Handler (Script)
Script: DefaultTrackableEventHandler

Turn Off Behaviour (Script)
Script: TurnOffBehaviour

Mesh Renderer
Materials
Size: 1
Element 0: queen_of_heartsMaterial
Lighting
Cast Shadows: On
Receive Shadows: ☑
Contribute Global Illumination: □
Receive Global Illumination: Light Probes
Probes
Light Probes: Off
Reflection Probes: Blend Probes
Anchor Override: None (Transform)
Additional Settings
Motion Vectors: Per Object Motion
Dynamic Occlusion: ☑

Image Target Mesh 13518 (Mesh Filter)
Mesh: ImageTargetMesh13518

Player Input (Script)
Script: PlayerInput
Object Found Text: ObjectLocatedText
Chest Anim: chest (Animator)

Figure 14: Setting the *Player Input* fields.

Next, click Canvas and fill out the fields in GUIFunctions. Drag HideButton into GuiAnim and TimesPressedText (found under HideButton) into Pressed Text.

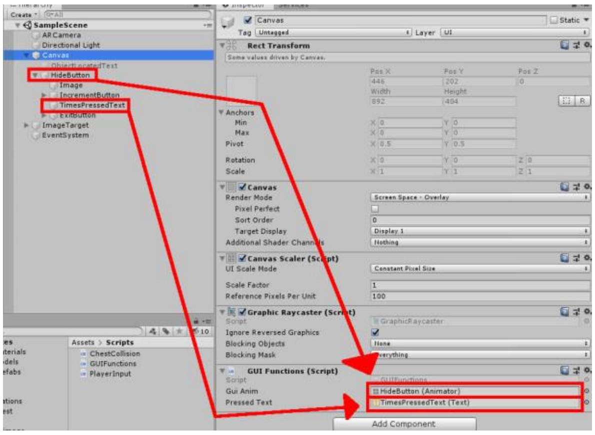


Figure 15: Setting the *GUIFunctions* fields.

Move on to the Chest object, which contains the ChestCollision script. Simply drag the ObjectLocatedText object in the Hierarchy into the Game Text field.
Now all that's left is to assign the functions created in GUIFunctions to their corresponding buttons. Start with HideButton and move to the Inspector. Find the OnClick event list and add a new event. Drag Canvas into the object field.

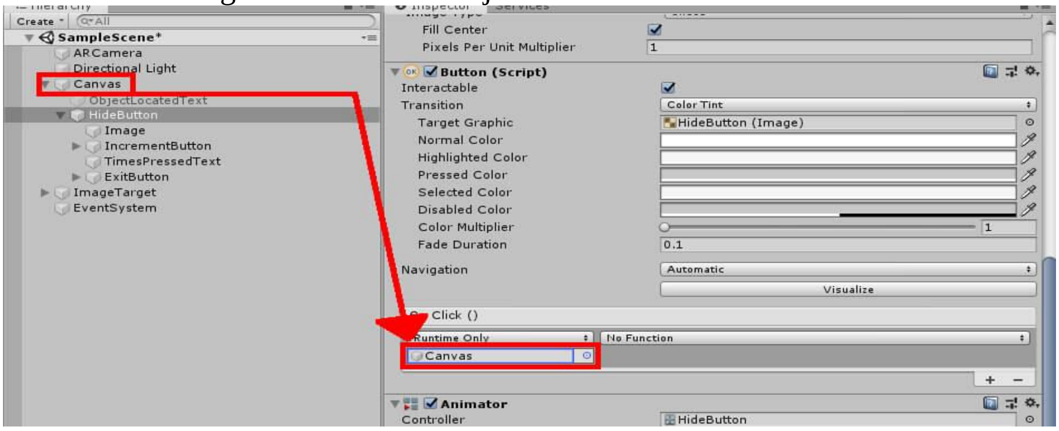


Figure 16: Creating a new *OnClick* event for *HideButton*.

Select the button that says NoFunction and navigate to GUIFunctions->ShowOrHideButton

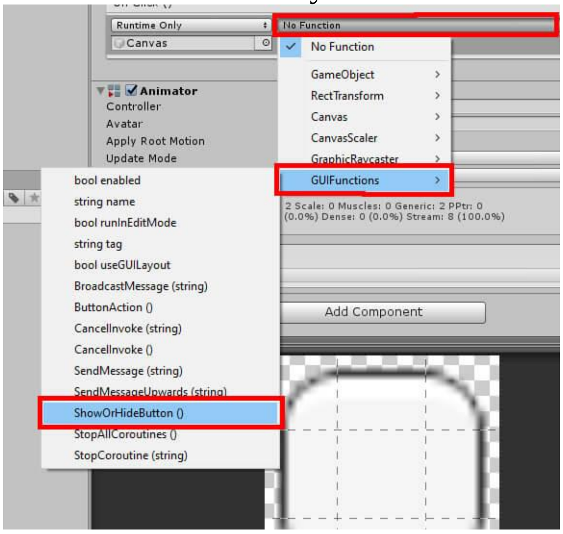


Figure 17: Selecting the function.

Repeat this process for *IncrementButton* and *ExitButton.*
Choose GUIFunctions->ButtonAction
for *IncrementButton and GUIFunctions->ExitGame* for *ExitButton.*

This will complete the setup for the project. If a webcam attached to computer test out the program with that.
Otherwise, test the app on mobile device by selecting *File->Build and Run* from the top menu. The chest will appear if the app sees the card.

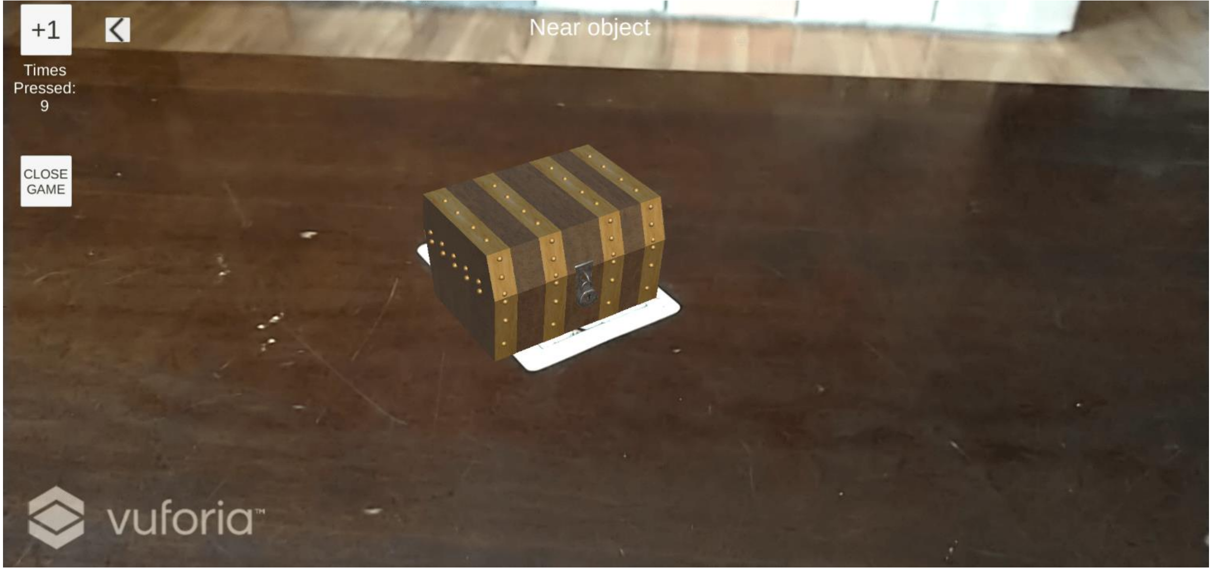


Figure 18: The chest appears when the queen of hearts is found.

Conclusion

As mentioned from the start, this is merely a sampling of AR development. There are other features available such as allowing the user to place objects in the world in real-time. With Vuforia, we can also detect a variety of targets beyond simple images.

BIBLIOGRAPHY

WEBLINKS:

1. https://youtu.be/b7roQ0ftqJc covering all the most important things to navigate. This tutorial is primarily for new users. Some basics.

2. Download Assets: https://assetstore.unity.com
3. https://www.youtube.com/watch?v=G1FDs2sqxeI covering the create basic AR mobile application with help of unity , vuforiya ,Android SDK's.