AR BASED SMART EDUCATION IN ANDROID PLATFORM WITH UNITY

A PROJECT REPORT

Submitted by

Aishwarya C [211417104006]

Devika Vinod [211417104049]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2020

BONAFIDE CERTIFICATE

Certified that this project report **"AR BASED SMART EDUCATION IN ANDROID PLATFORM USING UNITY"** is the bonafide work of **"AISHWARYA C(211417104006), DEVIKA VINOD(211417104049)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.S.MURUGAVALLI,M.E.,Ph.D.,	**VINMATHI M S**
HEAD OF THE DEPARTMENT	**ASSOCIATE PROFESSOR**
DEPARTMENT OF CSE,	DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,	PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,	NASARATHPETTAI,
POONAMALLEE,	POONAMALLEE,
CHENNAI-600 123.	CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on...........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Thiru.C.SAKTHIKUMAR,M.E.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide VINMATHI M S** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

AISHWARYA C,

DEVIKA VINOD

ABSTRACT

Augmented reality is a fast and an emerging research field of computer science and technology AR brings digital to life by interacting with the environment in real time. Now-a-days information and communication technology support the development of human interaction with physical, computer and virtual environment such as science, commercial, banking, education, etc. Augmented reality is a field of computer research which deals combination of reality with computer related data.

It is a well-known fact that our current education system lags behind in certain aspects. We still follow the traditional blackboard method of teaching which fails to provide every student with proper guidance. By introducing Augmented reality into the education system, teachers can convey their knowledge more precisely and in a very attractive manner. Some concepts are difficult to understand and even harder to properly depict and present in a textbook. However, augmented reality serves as a new medium where it is much easier to achieve. In this application the topics can be observed in 3D graphical image and video format replacing the traditional 2D view. This system is designed to provide virtual training to the hyperactive person as well to conduct intelligent assessment to the same person. Therefore this deployment can also be done for creating a classroom environment for all the students by using augmented reality which initiates more interaction to students via MQTT protocol as well as they

can see all practical implementation of their subject through a virtual mode which reduces the stress of students in understanding a subject.

This system will help the students to view the 2D images in 3D which gives students better visualization and helps it easier to understand various layers, structures which otherwise is difficult to understand based on just a 2D structure.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Augmented reality has been a hot topic in software development circles for a number of years, but it's getting renewed focus and attention with the release of products like Google Glass. Augmented reality is a technology that works on computer vision-based recognition algorithms to augment sound, video, graphics and other sensor-based inputs on real world objects using the camera of your device. It is a good way to render real world information and present it in an interactive way so that virtual elements become part of the real world. Augmented reality displays superimpose information in your field of view and can take you into a new world where the real and virtual worlds are tightly coupled. It is not just limited to desktop or mobile devices.

A simple augmented reality use case is: a user captures the image of a real-world object, and the underlying platform detects a marker, which triggers it to add a virtual object on top of the real-world image and displays on your camera screen.

CHAPTER 2

LITERATURE SURVEY

Muhammed S.Khan has proposed a Using Convolutional Neural Networks for Smart Classroom Observation which uses Convolution Neural Network (CNN) for smartclass. It used class observation audio data to classify classroom activities into those based on the Stallings Classroom Snapshot where a classroom observer takes a 360 degree view of a classroom every 5 minutes or so for a 50-minute class for 15 seconds and then records his observations using a structured coding sheet.

Nikitha Kommera, Faisal Kaleem, Syed Mubashir Shah Harooni have proposed a Smart Augmented Reality Glasses in Cybersecurity and Forensic Education which is an augmented reality technology based teaching and learning approach to enhance cybersecurity education by using the leading-edge technology that gives rise to new ways to motivate and educate students about cybersecurity and forensics field.

Gambo Yusufu, Nachandiya Nathan have proposed a Novel Model Of Smart Education For The Development Of Smart University System which is a student-centric intelligent learning environment enriched with digital learning resources to provide smart pedagogies that support smart learners' personalized learning experiences anywhere at any time using smart portable device and linked across educational institution or training workforce through the advancement and superiority of smart and wireless technologies.

Nallapaneni Manoj Kumar, P. Ranjith Krishna have proposed Use of Smart Glasses in Education which are Applications of wearable smart glass in education include the augmented reality, documentation of lecture, on-site report preparation, recording lectures as videos, capturing essential points as images, tele-mentoring, trainee's evaluation, understanding the listener's experience and nature, student concentration evaluation.

SR NO	TITLE	YEAR	AUTHOR	CONCEPT	DRAWBACKS
1.	A Smart Phone Integrated Smart Classroom	2016	Mahesh.G Jayahari K.R	Wi Fi router would be connected to the mobile phone and it will be connected to a server to control the attendance of the students and takes face recognition.Internet would not be needed.The Wi Fi would be confined only inside the classroom.	Each student requires smart phone
2.	Using Mobile Devices & Social Media in Supporting Engineering Education	2014	Khan M M Jeffrey C L Chiang	Investigates how students at university use smart phones with respect to engagement and interaction in various learning activities. Studies how students engage with learning tasks and what social interactions occur when they are trying to achieve their academic goals.	It includes few tools and everyone would require a smart mobile phones.

3.	Smart learning Environment	2019	Xinxin Deng Rong Zhang	IOT is used here as it completely studies the matching degree, satisfaction degree, advantages and disadvantages between the general multimedia classrooms and the smart learning classrooms.	No interaction with the smart class environment
4.	Use of Smart Glasses in Education	2018	Nallapaneni Manoj Kumar P. Ranjith Krishna	Applications of wearable smart glass in education include the augmented reality, documentation of lecture, on-site report preparation, recording lectures as videos, capturing essential points as images, telementoring, trainee’s evaluation, understanding the listener's experience and nature, student concentration evaluation.	Implementation of smart glass services would be difficult as they involve the complex network.

5.	A Novel Model Of Smart Education For The Development Of Smart University System	2020	Gambo Yusufu Nachandiya Nathan	Smart education IS “a student-centric intelligent learning environment enriched with digital learning resources to provide smart pedagogies that support smart learners’ personalised learning experiences anywhere at any time using smart portable device and linked across educational institution or training workforce through the advancement and superiority of smart and wireless technologies.	Investment in smart technologies and digital devices comes with cost on the educational institution. New security and safeguards policies needed to be developed.
6.	Intelligent classroom information system	2020	Zijuan Zhou	Mobile application based. It uses a set of wireless monitoring and terminal mobile client of the system which can transmit the monitoring data for clients for	Suitable for distance learning not for classroom

				enquiry and thereby resolve the query.	
7.	Virtual Smart Classroom	2017	Noawanti Songkram	The application, 'VSLI' standing for Virtual Smart Learning and Innovation System, is used. The functions of this application consist of chat logs, webboards, hall of fame, learners' tasks, open resources, and learners' data collection during the lesson allowing the teacher to access the data and use it for assessment	Implementation is costlier
8.	Smart Augmented Reality Glasses in Cybersecurity and Forensic Education	2016	Nikitha Kommera Faisal Kaleem Syed Mubashir Shah Harooni	This paper presents an augmented reality technology based teaching and learning approach to enhance cybersecurity education by using the leading-edge technology that gives rise to new ways to motivate and educate students about cybersecurity and	IOC,An important tool for forensic education is still not implemented.

				forensics field; the innovative use of cutting-edge technology to develop new interactive courses and modules in cybersecurity and forensics will attract and retain students with STEM focus.	
9.	Using Convolutional Neural Networks for Smart Classroom Observation	2020	Muhammed S.Khan	Convolution Neural Network (CNN) for smartclass.It used class observation audio data to classify classroom activities into those based on the Stallings Classroom Snapshot where a classroom observer takes a 360 view of a classroom every 5 minutes or so for a 50-minute class for 15 seconds and then records his observations using a structured coding sheet.	Real time implementation is difficult

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Traditional methods of teaching include a verbal explanation with the help of the curriculum textbooks and written text or drawings written or drawn by the teacher on the blackboard in the classroom.

However, this type of approach is not suitable for every student as every student and his/her power of grasping things varies. Modern generation of school students is finding it harder to pay attention to textbooks, blackboards, and paper handouts.

The drawbacks of the existing system are:

- Leads to boredom very easily.
- Not every student grasps the concepts.
- Concepts stay in students brain for a less amount of time due to its monotonicity.
- Inflexible learning.
- Lack of motivation.

3.2 PROPOSED SYSTEM

With the approach of augmented reality the above mentioned drawbacks can be over-comed. In the field of education AR has not been implemented to the fullest. People today are well versed with the technology and are operating smartphones which support AR. Thus, the concept of creating a “AR based smart education application” brings the current education system a step closer to being technologically advanced. With the recent emergence of better cameras and more accurate sensors in soon-to-be mainstream devices. In our current implementations of the application, we use Vuforia to accurately detect the respective material and the virtual 3D image of the particular concept that we require to be visualized.

The proposed system uses Marker-less Augmented Reality as a basis for enhancing user experience and for a better perception of things. Marker less tracking is a method of positional tracking – the determination of position and orientation of an object within its environment. This is a very important feature in augmented reality (AR), making it possible to know the field-of view and perspective of the user - allowing for the environment to react accordingly or the placement of augmented reality content in accordance with real world. While marker-based methods of motion tracking use specific optical markers, marker-less positional tracking does not require them, making it a more flexible method. It also avoids the need for a prepared environment in which fiducial markers are placed.

3.3 Requirement Analysis and Specification

The requirement engineering process of feasibility study, requirements elicitation and analysis, requirement specification, requirements validation and requirement management. Requirement elicitation and analysis is an iterative process that can be represented as a spiral of activities, namely requirements discovery, requirements classification and organization, requirement negotiation and requirements documentation.

3.4 SOFTWARE REQUIREMENT

Language : C#

Operating system : Android 8.0 or more

Tools

- Unity 3D
- Vuforia
- Android Studio

3.5 HARDWARE REQUIREMENT

• **RAM Capacity** : 4GB

• **Memory** : 120 MB

• **Graphics Card** : 1 GB

• **Accessories** : Smart phone with AR support

3.6 SOFTWARE SPECIFICATIONS

3.6.1 Unity

Unity3D is a “game development ecosystem”, it includes an environment for the development of interactive 2D and 3D content including a rendering and physics engine, a scripting interface to program interactive content, a content exporter for many platforms (desktop, web, mobile) and a growing knowledge sharing community. As shown in below figure 1.1 Unity 2019.4.16f1 introduces new tools that help artists & designers tell better visual stories, new ways for teams to collaborate more productively, and more features than ever to help you succeed in the gaming industry.

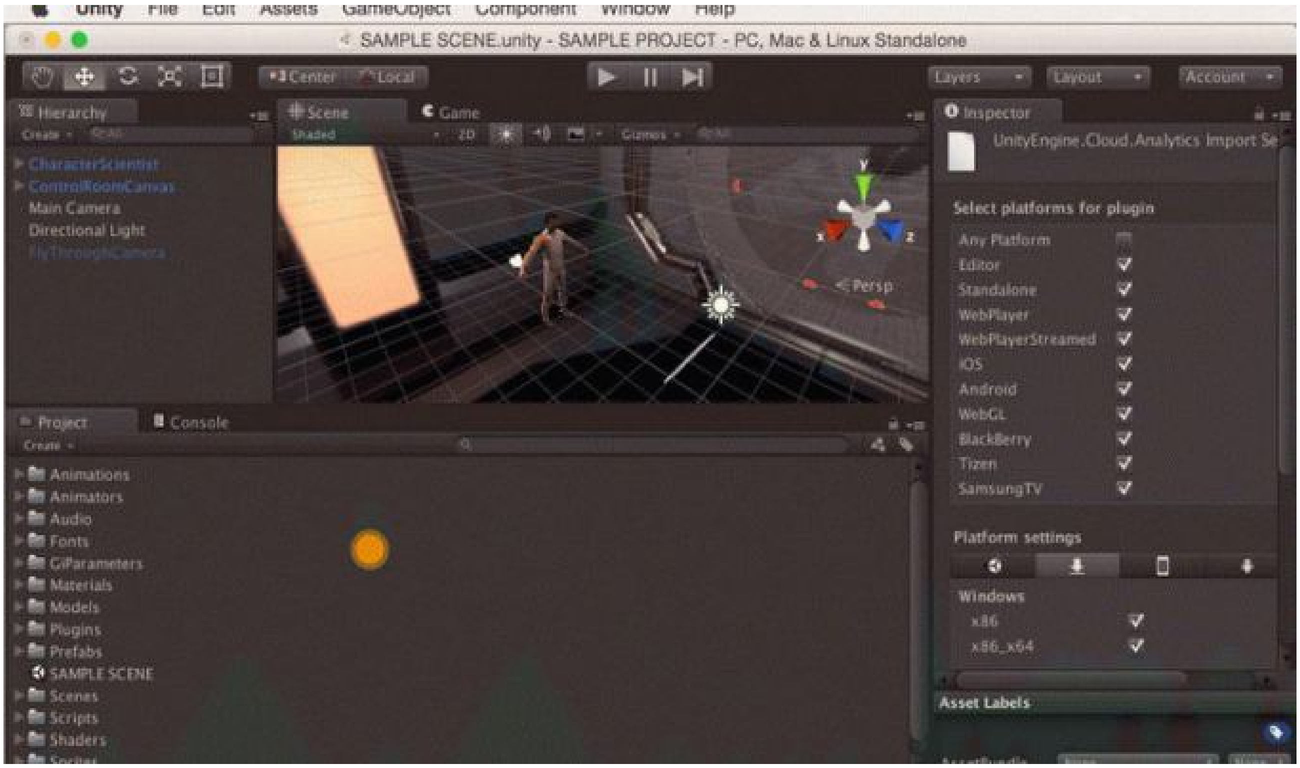


Figure 1.1: Unity software interface

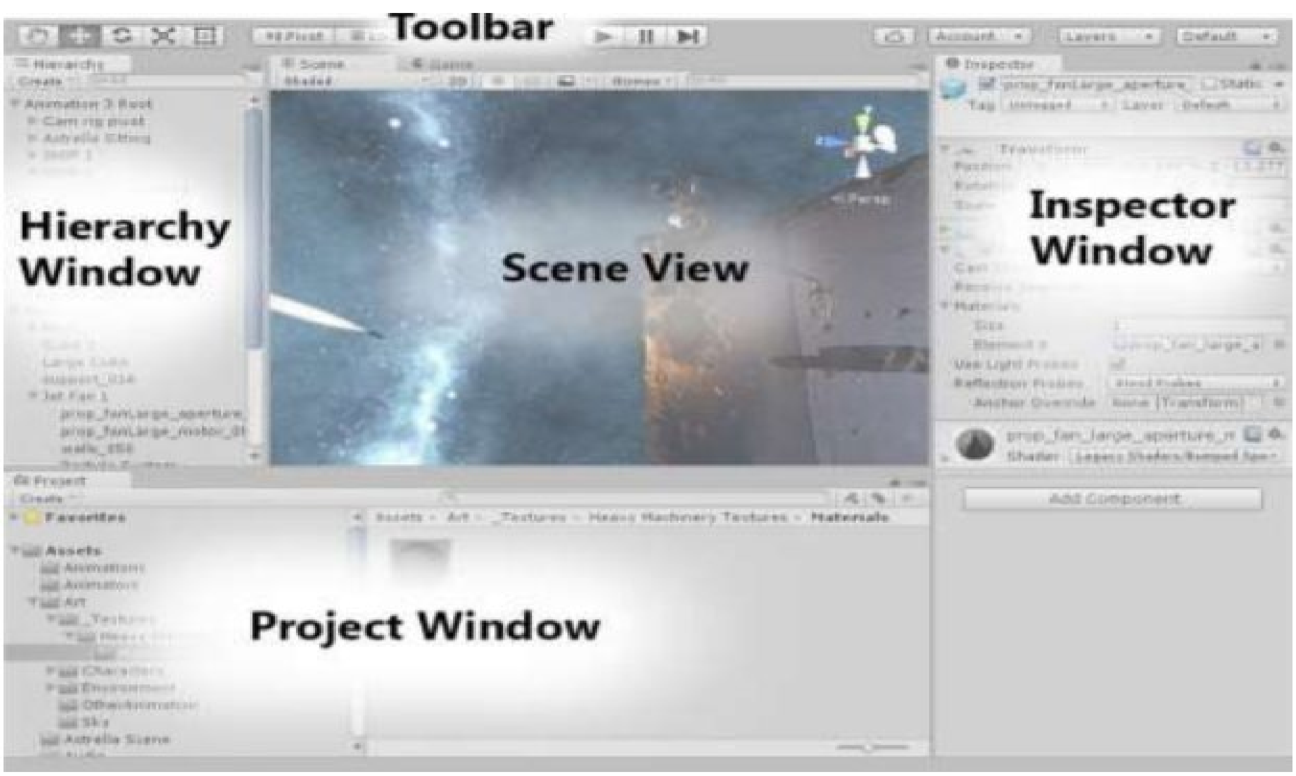


Figure 1.2: Interface options

As shown in above figure 1.2, the main editor window is made up of tabbed windows which can be rearranged, grouped, detached and docked. This means the look of the editor can be different from one project to the next, and one developer to the next, depending on personal preference and what type of work you are doing. The default arrangement of windows gives you practical access to the most common windows. If you are not yet familiar with the different windows in Unity, you can identify them by the name in the tab.

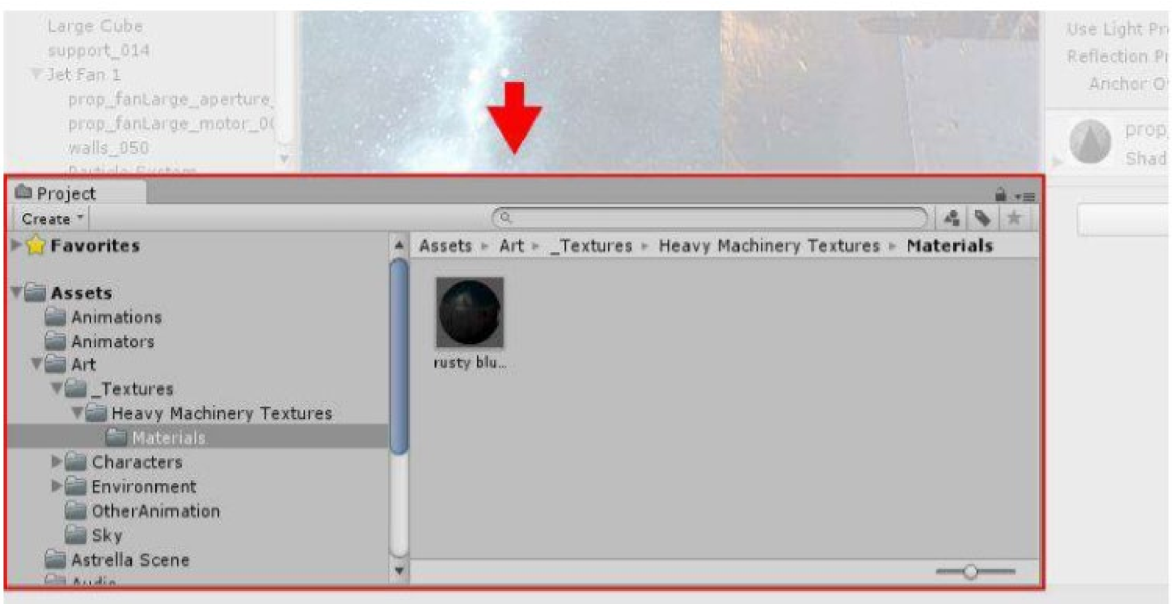


Figure 1.3: Project window

As shown in above figure 1.3, the Project window displays your library of assets that are available to use in your project. When you import assets into your project, they appear here.

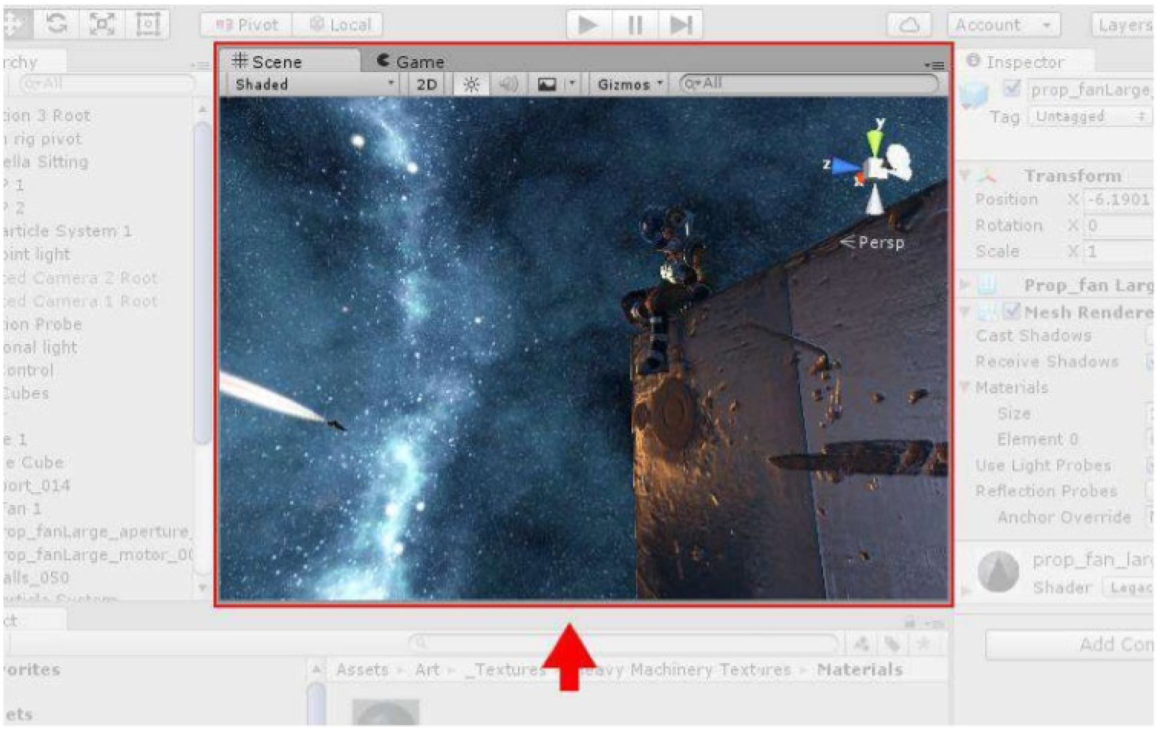


Figure 1.4: Scene view window

As shown in the above figure 1.4 allows you to visually navigate and edit your scene. The scene view can show a 3D or 2D perspective, depending on the type of project you are working on.

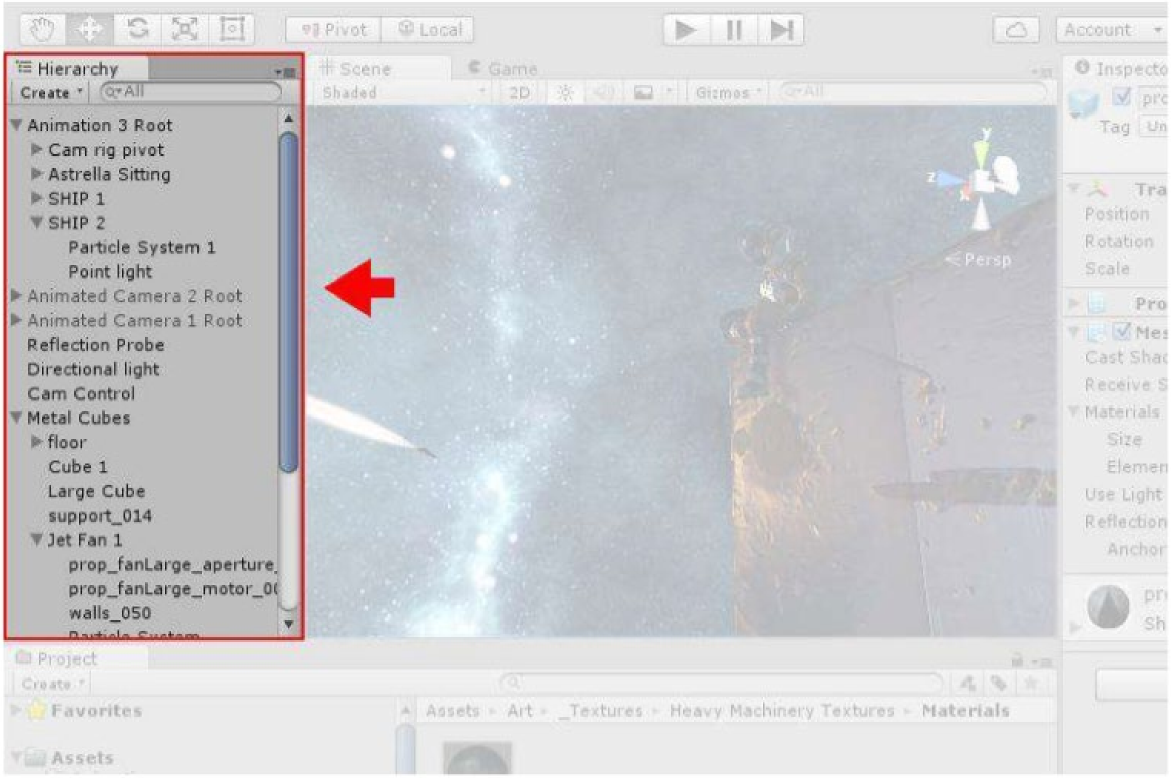


Figure 1.5: Hierarchy window

As shown in above figure 1.5, the Hierarchy window is a hierarchical text representation of every object in the scene. Each item in the scene has an entry in the hierarchy, so the two windows are inherently linked. The hierarchy reveals the structure of how objects are attached to one another.

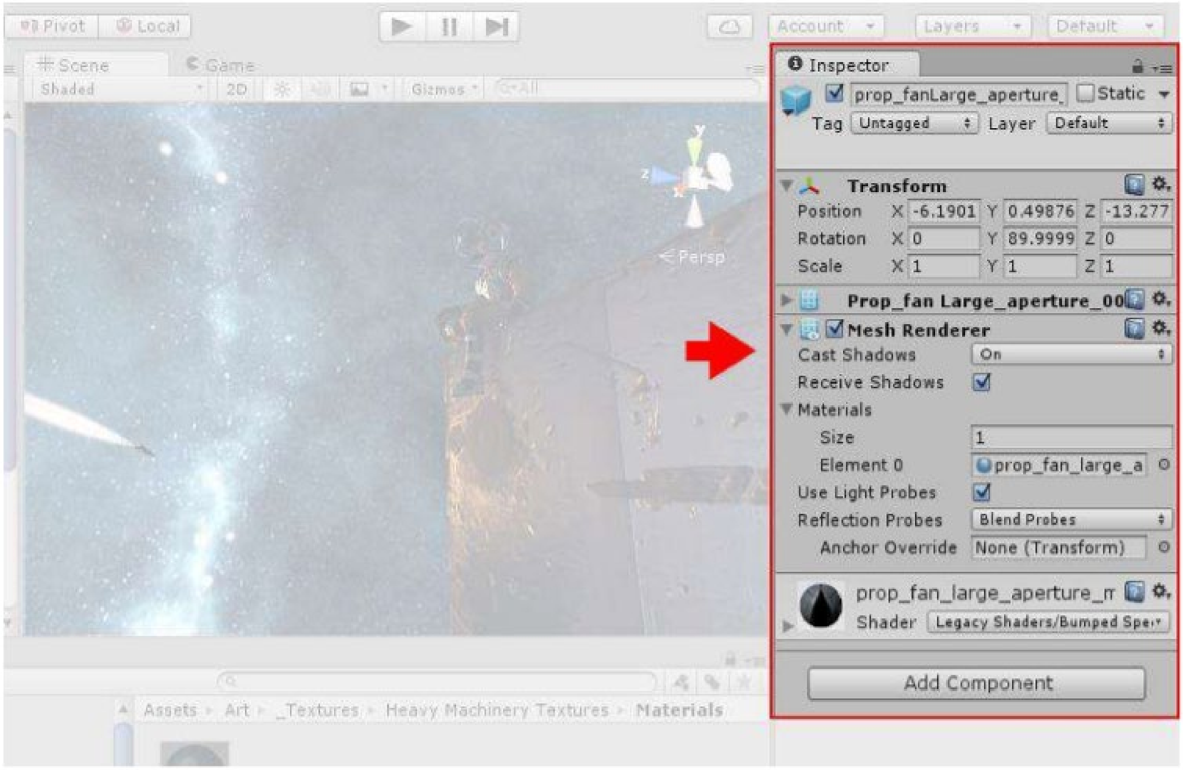


Figure 1.6: Inspector window

As shown in above figure 1.6, the Inspector window allows you to view and edit all the properties of the currently selected object. Because different types of objects have different sets of properties, the layout and contents of the inspector window will vary.

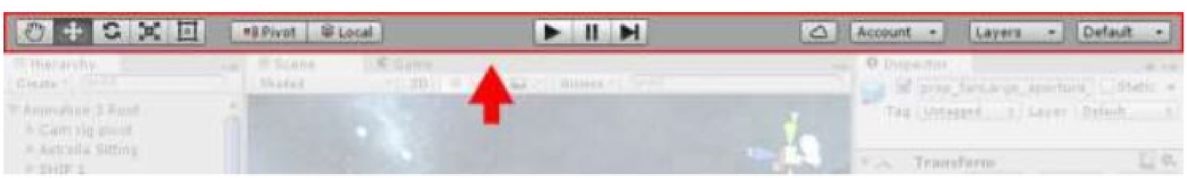


Figure 1.7: Toolbar window

As shown in above figure 1.7, the Toolbar window provides access to the most essential working features. On the left it contains the basic tools for manipulating the scene view and the objects within it. In the centre are the play, pause and step controls. The buttons to the right give you access to your Unity, Cloud Services and your Unity Account, followed by a layer visibility menu, and finally the editor layout menu.

3.6.2 Vuforia

Vuforia is an augmented reality software development kit for mobile devices that enables the creation of augmented reality applications. It uses computer vision technology to recognize and track planar images and simple 3D objects, such as boxes, in real time. This image registration capability enables developers to position and orient virtual objects, such as 3D models and other media, in relation to real world images when they are viewed through the camera of a mobile device. The virtual object then tracks the position and orientation of the image in real time so that the viewer's perspective on the object corresponds with the perspective on the Image Target. It thus appears that the virtual object is a part of the real-world scene.

The Vuforia SDK supports a variety of 2D and 3D target types including 'marker-less' Image Targets, 3D Multi-Target configurations, and a form of addressable Fiducial Marker, known as a Vu-Mark. Additional features of the SDK include localized Occlusion Detection using 'Virtual Buttons', runtime image target selection, and the ability to create and reconfigure target sets programmatically at runtime.

3.6.3 Android Studio

Android studio is an IDE(Integrated Development Environment) developed and managed by google, which is being used to devlope android application. Android studio provides auto generated code editor which means that to write your XML java or kotlin code you need not to write it completely it generates it for you. It's gradle feature helps you in compiling and building the application, in short android studio is best IDE for Android app development. You can connect your smart phone as well as can download smart phone in android studio for simulation process.

Everything you need to build on Android

Android Studio is Android's official IDE. It is purpose built for Android to accelerate your development and help you build the highest-quality apps for every Android device.

It offer tools custom-tailored for Android developers, including rich code editing, debugging, testing, and profiling tools.

Code and Iterate Faster Than Ever

Based on Intellij IDEA, Android Studio provides the fastest possible turnaround on your coding and running workflow.

Instant Run

Android Studio's Instant Run feature pushes code and resource changes to your running app. It intelligently understands the changes and often delivers them without restarting your app or rebuilding your APK, so you can see the effects immediately.

Intelligent code editor

The code editor helps you write better code, work faster, and be more productive by offering advanced code completion, refactoring, and code analysis. As you type, Android Studio provides suggestions in a dropdown list. Simply press Tab to insert the code.

Fast and feature-rich emulator

The Android Emulator installs and starts your apps faster than a real device and allows you to prototype and test your app on various Android device configurations:

phones, tablets, Android Wear, and Android TV devices. You can also simulate a variety of hardware features such as GPS location, network latency, motion sensors, and multi-touch input.

Configure Builds Without Limits

Android Studio's project structure and Gradle-based builds provide the flexibility you need to generate APKs for all device types.

Robust and flexible build system

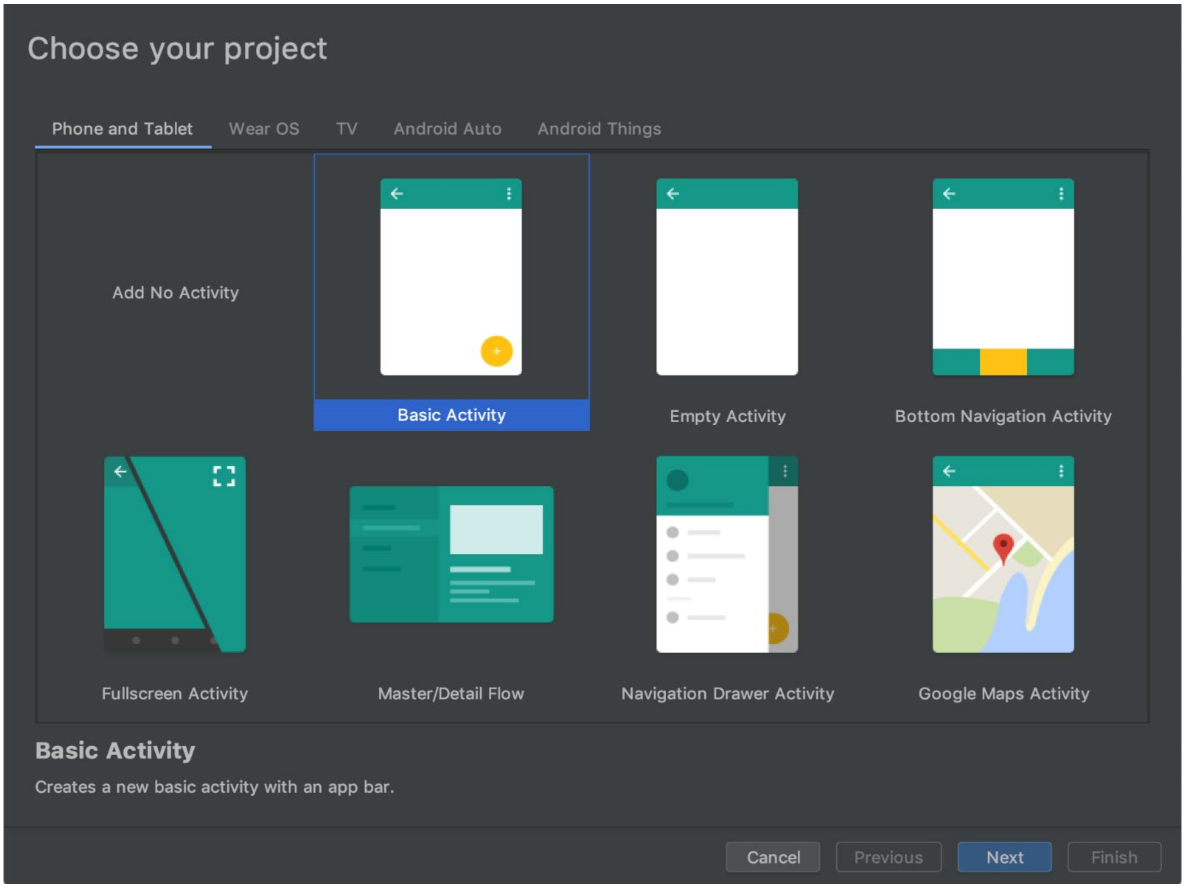
Android Studio offers build automation, dependency management, and customizable build configurations. You can configure your project to include local and hosted libraries, and define build variants that include different code and resources, and apply different code shrinking and app signing configurations.

Designed for teams

Android Studio integrates with version control tools, such as GitHub and Subversion, so you can keep your team in sync with project and build changes. The open source Gradle build system allows you to tailor the build to your environment and run on a continuous integration server such as Jenkins.

Optimized for all Android devices

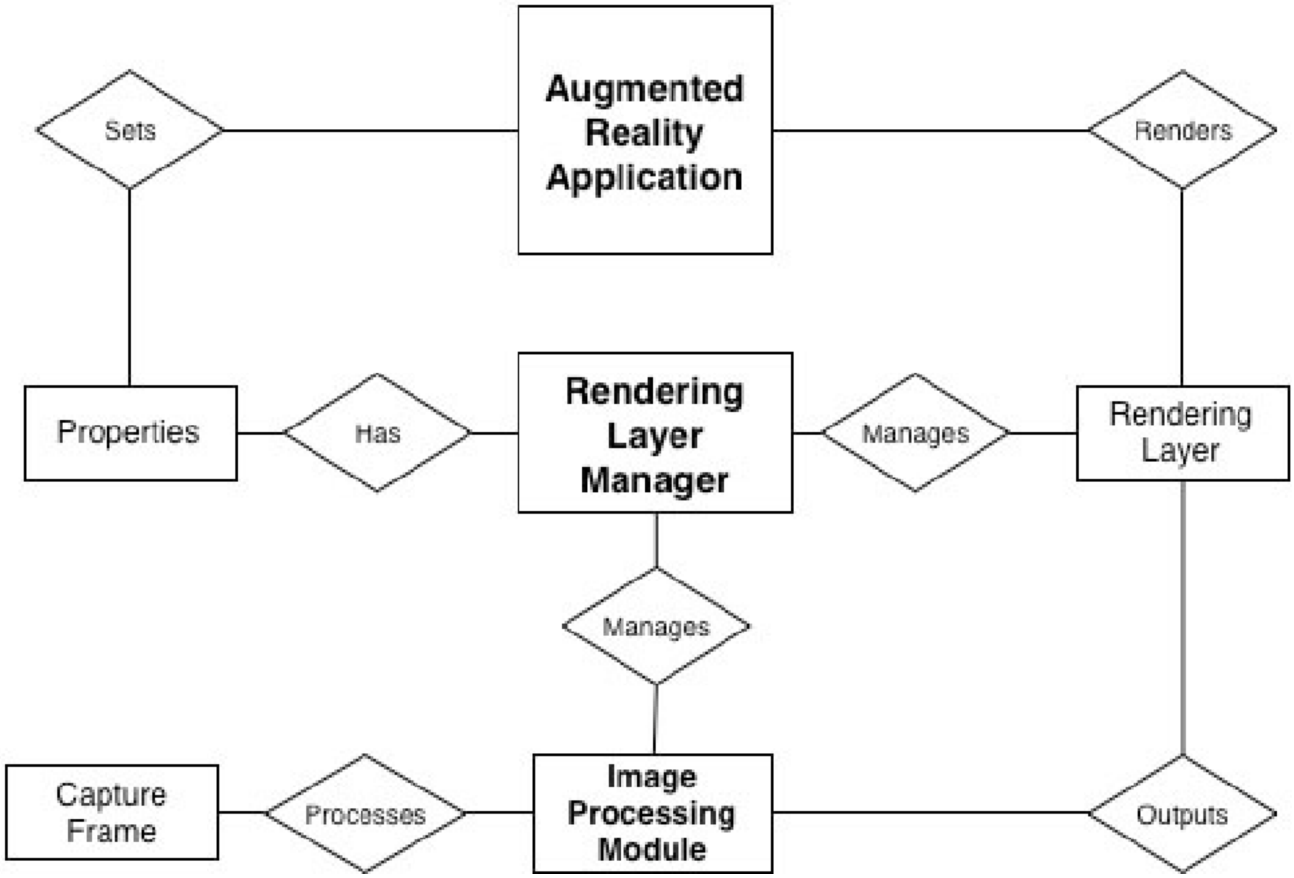
Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.


Choose your project
Phone and Tablet
Wear OS
TV
Android Auto
Android Things
Add No Activity
Basic Activity
Empty Activity
Bottom Navigation Activity
Fullscreen Activity
Master/Detail Flow
Navigation Drawer Activity
Google Maps Activity
Basic Activity
Creates a new basic activity with an app bar.
Cancel
Previous
Next
Finish

CHAPTER 4

SYSTEM DESIGN

4.1 ER DIAGRAM FOR AUGMENTED REALITY SYSTEM



4.2 UML DIAGRAMS

UML stands for Unified Modeling Language. It’s a rich language to model software solutions, application structures, system behavior and business processes. There are 14 UML diagram types to help you model these behaviors. Unified Modeling LanguageTM (UML®) is a standard visual modeling language intended to be used for

- modeling business and similar processes,
- analysis, design, and implementation of software-based systems

4.2.1 USE CASE DIAGRAM

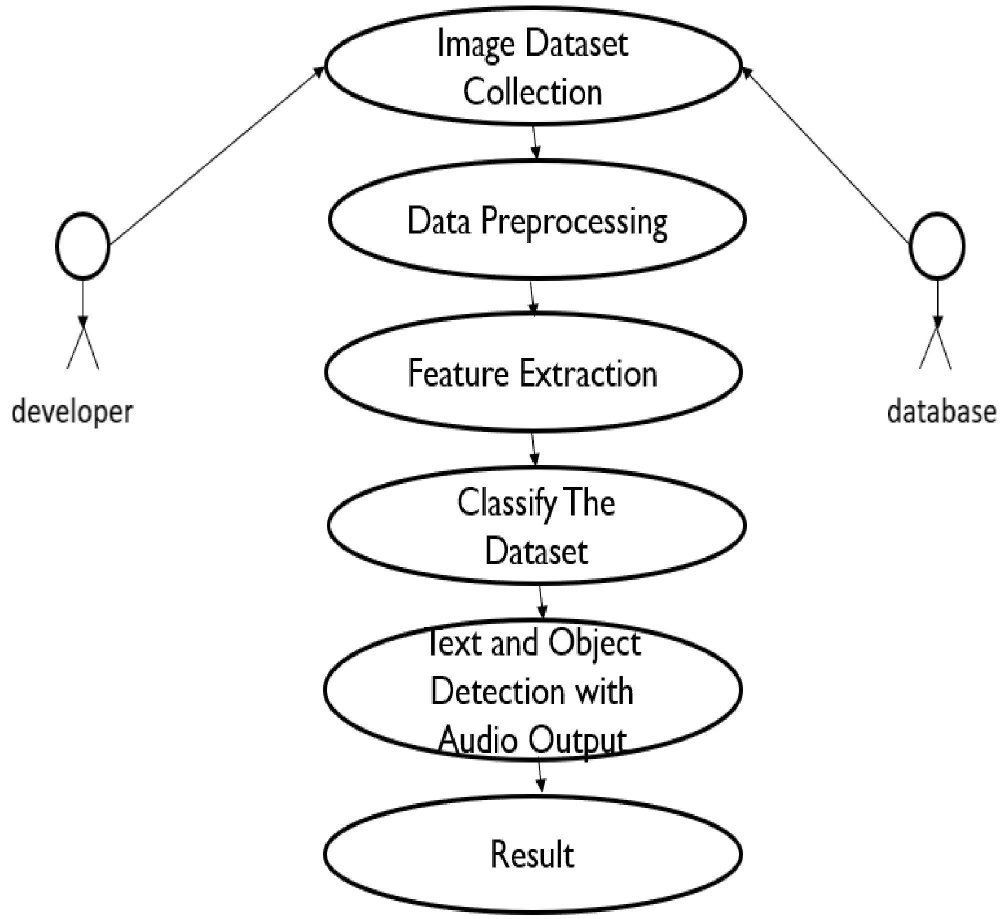


Figure 3.4: Use case diagram

As shown from the figure 3.4 above, we first collect all the target images required from which the Vuforia extracts the features such that it identifies the target image. After that we will classify the various target images based on the 3D images we want to display (if it's a normal image, image with rotation or a video with audio). Then accordingly, when we place our camera over the target image, the respective 3D model will get displayed.

4.2.2 CLASS DIAGRAM

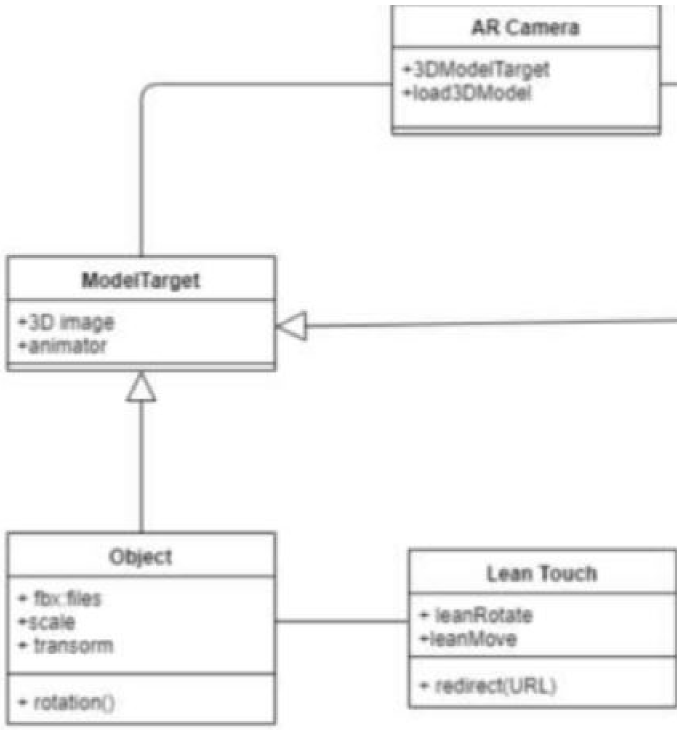


Fig 4.3.Class Diagram

As shown from the above figure 4.3, it describes the class diagram of the application where the main classes include AR Camera, Model Target, Object, Lean touch, Canvas. AR Camera consists of the main configuration to be set to get the 3D model when model target is given by user. Canvas class consists of the buttons and texts that are used in the application interface.

Object is the 3D model and consists of the information of scaling, transform. Lean touch class consists of how to rotate application i.e. portrait or landscape and all user interactions with the device using lean touch scripts.

4.2.3 Sequence Diagram

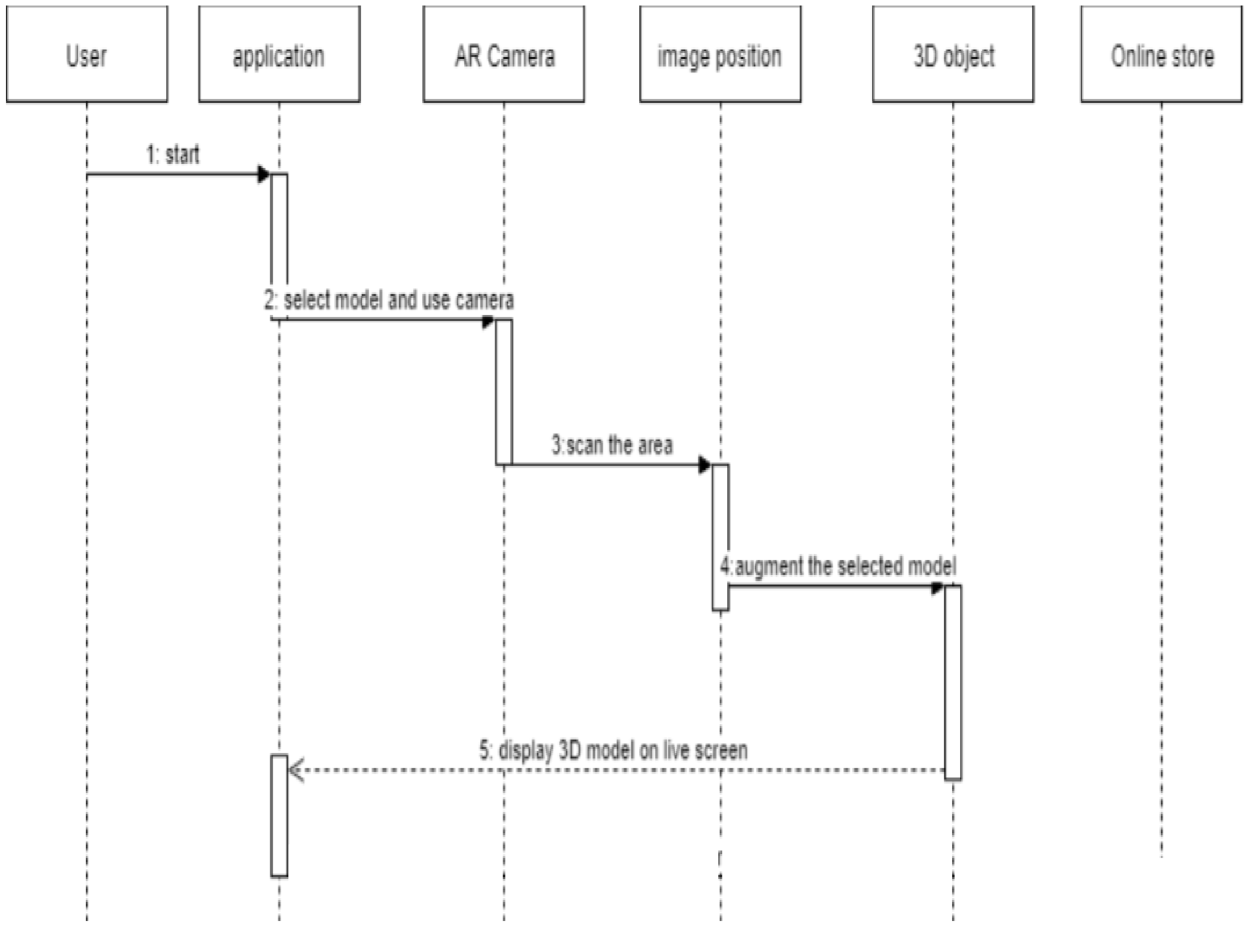


Figure 3.6: Sequence diagram

As shown in the above figure 3.6, it describes the sequence diagram of the application, i.e. how the application is started and to the end of the resultant augmented display in a sequence of interactions. User, application, AR Camera, image position, 3D object, Online store are the main objects of this diagram. First user starts the application and holds the living area by scanning surface. As soon as the area is scanned the 3D model selected is placed over the area.

4.2.4 ACTIVITY DIAGRAM

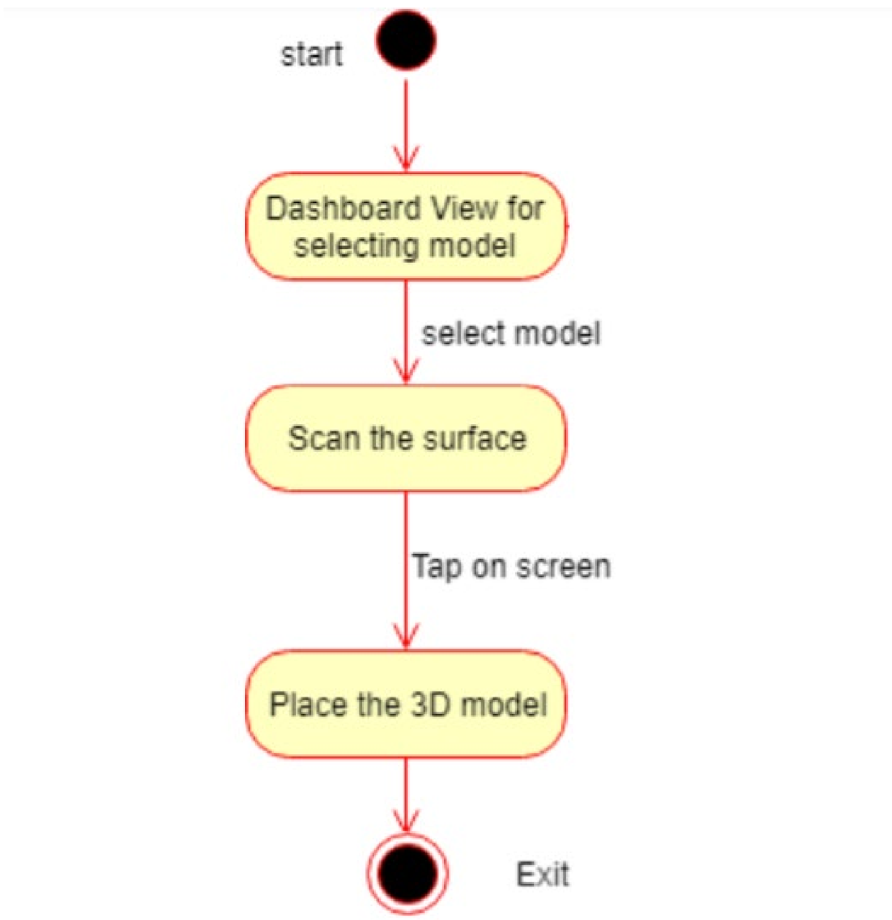


Figure 3.7: Activity diagram

As shown from the above figure 3.7, it describes the activity diagram of the application that consists of flow of the application which has the actions Dashboard view for selecting the model i.e. home page. Next scan the surface and place the 3D model.

CHAPTER 5
ARCHITECTURE

5.1 SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior, and more
views of a system. An architecture description is a formal descriptionand representation
of a system, organized in a way that supports reasoning about the structures and behaviors
of the system.

The system basically uses mobile phone built-in camera which supports Augmented reality to
collect view as the real scene view observed by human eye and stacks the 3D furniture models
on the screen displayed. First of all, we need to setup the scenes in Unity 3D for User Interface
of application like buttons, text areas, background image and virtual object selection.

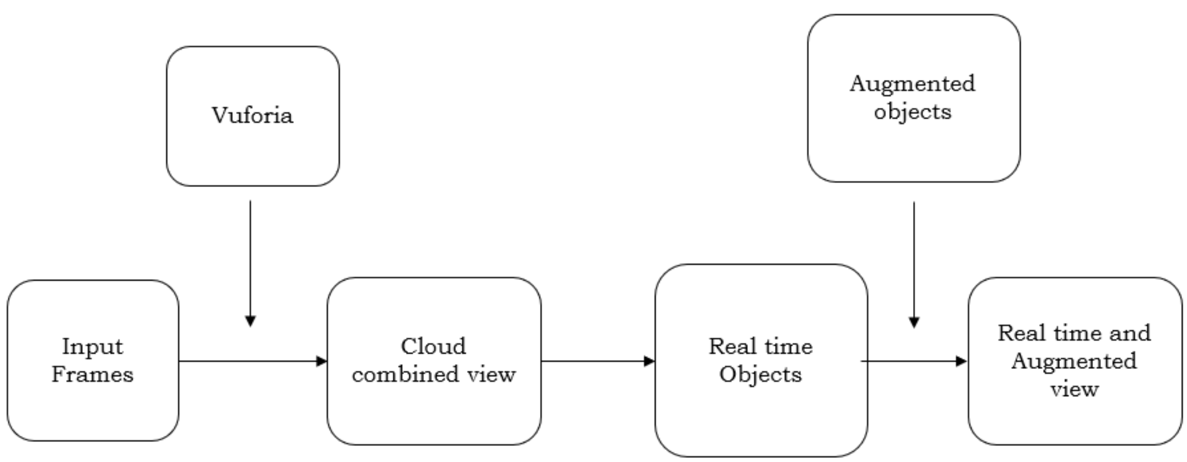


Fig 5.1: System Architecture of Application

As shown from the above figure 5.1, it describes the architecture of the application that take the real view as input with the help of AR camera then process it with virtual object to get the resultant output as augmented display.

5.2 SYSTEM MODULE

The application implementation consists of four modules.

- Creating Augmented Reality Objects.
- Developing Scenes for User Interface.
- Place the Object on the Surface Area.
- Verification of placed objects.

MODULES EXPLAINATION:

5.2.1 Creating Augmented Reality Objects

First, we have to open the unity3D project. Then we need to select the GameObject menu in the menu bar. The GameObject Menu has more objects to create a game. From there we can select a 3D object and pick the shape options given according to our requirement. We can select the rotation tool which helps to rotate our plane object. It also rotates all the objects in our scene. We can create the assets by using the hierarchy view which helps us to combine various shapes within the unity. There is a pro builder in Unity which allows us to edit our shapes helping us to obtain our 3D furniture objects.

5.2.2 Developing Scenes for User Interface

In this module we create scenes for every slide of application using Unity 3D. The main interface contains furniture model, buttons that helps to move to next model, to scan the surface area of living room, description of model such as length, width, height.

In order to implement these functions, we make the scene display to ratio of Android display and add the buttons to the scene that helps in moving to next scene. Later we use the 3D model developed in Unity 3D and functionalities like rotating the chair will be displayed using C# code for that object and add functionalities to move to next scene.The Unity 3D platform that is used to create scene for the chair model that is used as furniture which is developed in unity. For every model of the furniture we will create an individual scene and in the end we combine all scenes together.

5.2.3 Place the Virtual Object on the Surface Area

For this project Android studio is used with the Vuforia package. Vuforia packages is used because it offers 3D model demonstrations, to create applications for customers to personalize their products and gives a robust AR experience with the vision technology. We upload our target image in the Vuforia cloud, it renders the features of the target image and stores it such that when we scan the environment it identifies the target image.

After the identification, our 3D object gets placed over the target image. Target image acts as a position indicator such that the 3D furniture image gets fixed wherever we place our target image.

5.2.4 Verification of placed objects

Once the user thinks that the object is well suited to his need, he can check the description by selecting the information button that helps in describing the width, height and length of object.

In order to view this description, we create another scene that helps in displaying all the required information. We add one more button that helps in rotating object by an angle of 30°.

For this rotation and link redirection we use two classes which are programmed with C#.

CHAPTER 6

SYSTEM IMPLEMENTATION

```
/* Add Vuforia engine package */
using System.IO;
using System.Linq;
using System.Text;
using UnityEditor;
using UnityEditor.PackageManager;
using UnityEditor.PackageManager.Requests;
using UnityEngine;

[InitializeOnLoad]
public class AddVuforiaEnginePackage
{
   static readonly string sPackagesPath = Path.Combine(Application.dataPath, "..",
"Packages");
   static readonly string sManifestJsonPath = Path.Combine(sPackagesPath,
"manifest.json");
   const string VUFORIA_VERSION = "9.7.5";
   const string PACKAGE_KEY = "com.ptc.vuforia.engine";
   const string GIT_URL = "git+https://git-packages.developer.vuforia.com";

   static readonly ScopedRegistry sVuforiaRegistry = new ScopedRegistry()
   {
      name = "Vuforia",
      url = "https://registry.packages.developer.vuforia.com/",
      scopes = new[] {"com.ptc.vuforia"}
   };
```

```
    static ListRequest sListRequest;

    static AddVuforiaEnginePackage()
    {
        if (Application.isBatchMode)
            return;

        var manifest = Manifest.JsonDeserialize(sManifestJsonPath);

        if (!IsUsingGitUrl(manifest))
            DisplayAddPackageDialogue(manifest);
    }

    static bool IsUsingGitUrl(Manifest manifest)
    {
        var dependencies = manifest.Dependencies.Split(',').ToList();
        return dependencies.Any(d => d.Contains(PACKAGE_KEY) &&
d.Contains(GIT_URL));
    }



    static void DisplayAddPackageDialogue(Manifest manifest)
    {
        if (EditorUtility.DisplayDialog("Add Vuforia Engine Package",
            $"Would you like to update your project to include the Vuforia Engine
{VUFORIA_VERSION} package from Git?\n" +
```

```
        $"If an older Vuforia Engine package is already present in your project it will
be upgraded to version {VUFORIA_VERSION}", "Update", "Cancel"))
      {
        UpdateManifest(manifest);
      }
    }


    static void UpdateManifest(Manifest manifest)
    {
      //remove existing, outdated NPM scoped registry if present
      var registries = manifest.ScopedRegistries.ToList();
      if (registries.Contains(sVuforiaRegistry))
      {
        registries.Remove(sVuforiaRegistry);
        manifest.ScopedRegistries = registries.ToArray();
      }

      //add specified vuforia version via Git URL
      SetVuforiaVersion(manifest);

      manifest.JsonSerialize(sManifestJsonPath);

      AssetDatabase.Refresh();
    }

    static void SetVuforiaVersion(Manifest manifest)
```

```
{
    var dependencies = manifest.Dependencies.Split(',').ToList();

    var versionEntry = $"\"{GIT_URL}#{VUFORIA_VERSION}\"";

    var versionSet = false;
    for (var i = 0; i < dependencies.Count; i++)
    {
        if (!dependencies[i].Contains(PACKAGE_KEY))
            continue;

        var kvp = dependencies[i].Split(':');

        kvp[1] = versionEntry;

        dependencies[i] = string.Join(":", kvp);

        versionSet = true;
    }

    if (!versionSet)
        dependencies.Insert(0, $"\n    \"{PACKAGE_KEY}\": {versionEntry}");

    manifest.Dependencies = string.Join(",", dependencies);
}

class Manifest
```

```
{
    const int INDEX_NOT_FOUND = -1;
    const string DEPENDENCIES_KEY = "\"dependencies\"";

    public ScopedRegistry[] ScopedRegistries;
    public string Dependencies;

    public void JsonSerialize(string path)
    {
        var jsonString = GetJsonString();

        var startIndex = GetDependenciesStart(jsonString);
        var endIndex = GetDependenciesEnd(jsonString, startIndex);

        var stringBuilder = new StringBuilder();

        stringBuilder.Append(jsonString.Substring(0, startIndex));
        stringBuilder.Append(Dependencies);
        stringBuilder.Append(jsonString.Substring(endIndex, jsonString.Length -
endIndex));

        File.WriteAllText(path, stringBuilder.ToString());
    }

    string GetJsonString()
    {
        if (ScopedRegistries.Length > 0)
```

```
            return JsonUtility.ToJson(
                new UnitySerializableManifest {scopedRegistries = ScopedRegistries,
dependencies = new DependencyPlaceholder()},
                true);

        return JsonUtility.ToJson(
            new UnitySerializableManifestDependenciesOnly() {dependencies = new
DependencyPlaceholder()},
            true);
    }


public static Manifest JsonDeserialize(string path)
{
    var jsonString = File.ReadAllText(path);

    var registries =
JsonUtility.FromJson<UnitySerializableManifest>(jsonString).scopedRegistries ??
new ScopedRegistry[0];
    var dependencies = DeserializeDependencies(jsonString);

    return new Manifest {ScopedRegistries = registries, Dependencies =
dependencies};
}

static string DeserializeDependencies(string json)
{
```

```
        var startIndex = GetDependenciesStart(json);
        var endIndex = GetDependenciesEnd(json, startIndex);

        if (startIndex == INDEX_NOT_FOUND || endIndex == INDEX_NOT_FOUND)
            return null;

        var dependencies = json.Substring(startIndex, endIndex - startIndex);
        return dependencies;
    }

    static int GetDependenciesStart(string json)
    {
        var dependenciesIndex = json.IndexOf(DEPENDENCIES_KEY,
StringComparison.InvariantCulture);
        if (dependenciesIndex == INDEX_NOT_FOUND)
            return INDEX_NOT_FOUND;

        var dependenciesStartIndex = json.IndexOf('{', dependenciesIndex +
DEPENDENCIES_KEY.Length);

        if (dependenciesStartIndex == INDEX_NOT_FOUND)
            return INDEX_NOT_FOUND;

        dependenciesStartIndex++; //add length of '{' to starting point

        return dependenciesStartIndex;
    }
```

```
    static int GetDependenciesEnd(string jsonString, int dependenciesStartIndex)
    {
        return jsonString.IndexOf('}', dependenciesStartIndex);
    }
}

class UnitySerializableManifestDependenciesOnly
{
    public DependencyPlaceholder dependencies;
}

class UnitySerializableManifest
{
    public ScopedRegistry[] scopedRegistries;
    public DependencyPlaceholder dependencies;
}

[Serializable]
struct ScopedRegistry
{
    public string name;
    public string url;
    public string[] scopes;

    public override bool Equals(object obj)
    {
```

```
    if (!(obj is ScopedRegistry))
        return false;

    var other = (ScopedRegistry) obj;

    return name == other.name &&
           url == other.url &&
           scopes.SequenceEqual(other.scopes);
}

public static bool operator ==(ScopedRegistry a, ScopedRegistry b)
{
    return a.Equals(b);
}

public static bool operator !=(ScopedRegistry a, ScopedRegistry b)
{
    return !a.Equals(b);
}

public override int GetHashCode()
{
    var hash = 17;

    foreach (var scope in scopes)
        hash = hash * 23 + (scope == null ? 0 : scope.GetHashCode());
```

```
        hash = hash * 23 + (name == null ? 0 : name.GetHashCode());
        hash = hash * 23 + (url == null ? 0 : url.GetHashCode());

        return hash;
    }
}

[Serializable]
    struct DependencyPlaceholder { }
}
```

INFO PART DISPLAY FOR BRAIN MODULE

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

public class arui : MonoBehaviour
{
    public GameObject Panel;

    public void OpenPanel()
    {
        if(Panel !=null)
        {
            bool isActive = Panel.activeSelf;
            Panel.SetActive(!isActive);
        }
    }
```

```
}
```

TO CHANGE SCENES

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

public class MenuScene : MonoBehaviour
{
    public void menuScene()
    {
        SceneManager.LoadScene("Menu");
    }
    public void book()
    {
        SceneManager.LoadScene("ARBook");
    }

    public void sun_earth()
    {
        SceneManager.LoadScene("SunEarth");
    }

    public void brain()
    {
        SceneManager.LoadScene("Brain");
```

```
    }
    public void ExitApp()
    {
        Application.Quit();
    }

}
```

Rotating the Spheres

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

public class RotateSphere : MonoBehaviour
{
    Vector3 movement;
    public int x1, y1, z1;
    // Start is called before the first frame update
    void Start()
    {
        movement = new Vector3(x1, y1, z1);
    }

    // Update is called once per frame
    void Update()
    {
        transform.Rotate(movement * Time.deltaTime);
    }
```

}

VIRTUAL BUTTONS

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

public class VirtualButton : MonoBehaviour
{
    public GameObject buttonGo;
    VirtualButtonBehaviour vrb;
    // Start is called before the first frame update
    void Start()
    {
        buttonGo.SetActive(false);

        vrb = GetComponentInChildren<VirtualButtonBehaviour>();

        vrb.RegisterOnButtonPressed(onButtonPressed);
        vrb.RegisterOnButtonReleased(onButtonReleased);
    }

    // Update is called once per frame
    public void onButtonPressed(VirtualButtonBehaviour vb)
    {
        buttonGo.SetActive(true);
```

```
        Debug.Log("Button Pressed");
    }

    public void onButtonReleased(VirtualButtonBehaviour vb)
    {
        buttonGo.SetActive(false);
        Debug.Log("Button Released");

    }
}
```

ELEPHANT ANIMATION

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

public class Elephant : MonoBehaviour
{
    Animator animator;
    // Start is called before the first frame update
    void Start()
    {
        animator = GetComponent<Animator>();
    }
// Update is called once per frame
    void Update()
    {
        gameObject.GetComponent<Animator>().enabled = false;
```

```
    }
}
```

/*Working*/

```
 using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Working : MonoBehaviour
{
        // Start is called before the first frame update
        void Start()
        {
                Screen.SetResolution(1920, 1080, true);
        } // Update is called once per frame
        void Update()
        {
        }
        }
```

/*Manifest.json*/

```
{
 "dependencies": {
 "com.ptc.vuforia.engine": "git+https://git-packages.developer.vuforia.com#9.8.5",
"com.unity.collab-proxy": "1.5.7",
"com.unity.ide.rider": "2.0.7",
"com.unity.ide.visualstudio": "2.0.8",
"com.unity.ide.vscode": "1.2.3",
"com.unity.test-framework": "1.1.24",
"com.unity.textmeshpro": "3.0.6",
```

"com.unity.timeline": "1.4.8",
"com.unity.ugui": "1.0.0",
"com.unity.modules.ai": "1.0.0",
"com.unity.modules.androidjni": "1.0.0",
"com.unity.modules.animation": "1.0.0",
"com.unity.modules.assetbundle": "1.0.0",
"com.unity.modules.audio": "1.0.0",
"com.unity.modules.cloth": "1.0.0",
"com.unity.modules.director": "1.0.0",
"com.unity.modules.imageconversion": "1.0.0",
"com.unity.modules.imgui": "1.0.0",
"com.unity.modules.jsonserialize": "1.0.0",
"com.unity.modules.particlesystem": "1.0.0",
"com.unity.modules.physics": "1.0.0",
"com.unity.modules.physics2d": "1.0.0",
"com.unity.modules.screencapture": "1.0.0",
"com.unity.modules.terrain": "1.0.0",
"com.unity.modules.terrainphysics": "1.0.0",
"com.unity.modules.tilemap": "1.0.0",
"com.unity.modules.ui": "1.0.0",
"com.unity.modules.uielements": "1.0.0",
"com.unity.modules.umbra": "1.0.0",
"com.unity.modules.unityanalytics": "1.0.0",
"com.unity.modules.unitywebrequest": "1.0.0",
"com.unity.modules.unitywebrequestassetbundle": "1.0.0",
"com.unity.modules.unitywebrequestaudio": "1.0.0",

"com.unity.modules.unitywebrequesttexture": "1.0.0",
"com.unity.modules.unitywebrequestwww": "1.0.0",
"com.unity.modules.vehicles": "1.0.0",
"com.unity.modules.video": "1.0.0",
"com.unity.modules.vr": "1.0.0",
"com.unity.modules.wind": "1.0.0",
"com.unity.modules.xr": "1.0.0"
}
}

CHAPTER 7

TESTING

7.1 SYSTEM TESTING

The testing approach document is designed for Information and Technology Services' upgrades to PeopleSoft. The document contains an overview of the testing activities to be performed when an upgrade or enhancement is made, or a module is added to an existing application. The emphasis is on testing critical business processes, while minimizing the time necessary for testing while also mitigating risks. It's important to note that reducing the amount of testing done in an upgrade increases the potential for problems after go-live. Management will need to determine how much risk is acceptable on an upgrade by upgrade basis. System testing is simply testing the system as a whole; it gets all the integrated modules of the various components from the integration testing phase and combines all the different parts into a system which is then tested. Testing is then done on the system as all the parts are now integrated into one system the testing phase will now have to be done on the system to check and remove any errors or bugs. In the system testing process the system will be checked not only for errors but also to see if the system does what was intended, the system functionality and if it is what the end user expected.

There are various tests that need to be conducted again in the system testing which include:

- Test Plan
- Test Case
- Test Data

If the integration stage was done accurately then most of the test plan and test cases would already have been done and simple testing would only have to be done in order to ensure there are no bugs because this will be the final product. As in the integration stage, the above steps would need to be re-done as now we have integrated all modules into one system, so we have to check if this runs OK and that no errors are produced because all the modules are in one system.

7.2Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

7.3 TEST CASES

SR NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	FINAL RESULT
1	Add image target in the Hierarchy	Image target added to the hierarchy.	Image target added to the hierarchy.	Pass
2	Press play button to see the animation	Animation can be seen.	Animation can be seen.	Pass
3	Opening software to make a new AR scene	Software was able to be opened.	Software was able to be opened.	Pass
4	Changing scene from one level to another level	Scenes could be changed.	Scenes could be changed.	Pass
5	Press the button to play the object	Object was played.	Object was played.	Pass
6	Tap to place the 3D object	3D object is placed	3D object is placed	Pass
7	3D object shows up only when virtual button is clicked	3D object when clicked successful	3D object when clicked successful	Pass
8	Information button is visible when clicked on image	Info part Visible	Info part Visible	Pass
9	Building app.	App has been built.	App has been built.	Pass
10	Open the application and target the 3D model	Application runs successfully.	Application runs successfully.	Pass

CHAPTER 7

CONCLUSION

7.1 Conclusion

The main objective of this “AUGMENTED REALITY BASED APPLICATION FOR EDUCATION” is to analyse the use of augmented reality in the field of education. This system will help us to enhance our current traditional education system which mostly makes use of blackboards and textbooks. This application will allow the students to get better understanding of certain topics as it will display 3D models live and thus provides a detailed view and explanation of the topics.

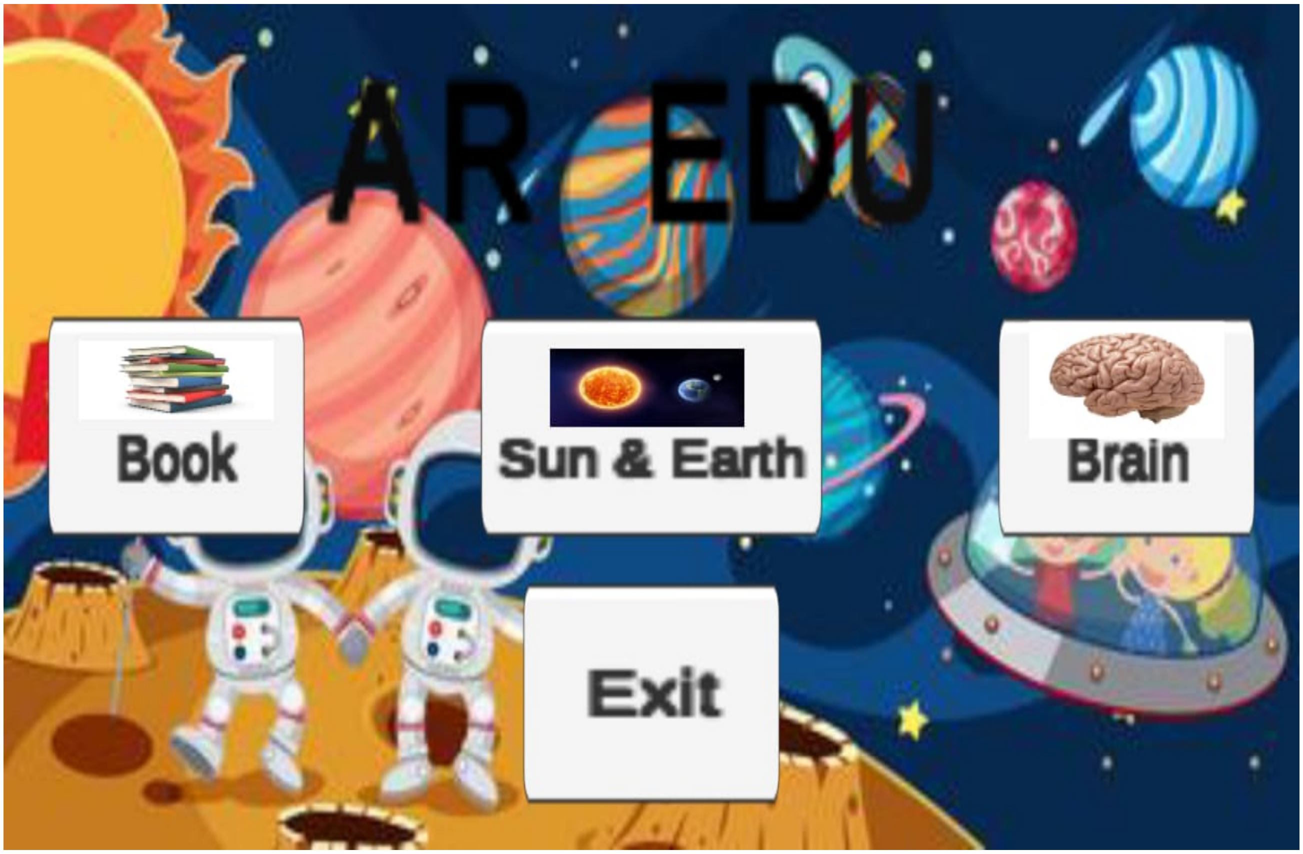
7.2 FUTURE ENHANCEMENTS

In future our “AUGMENTED REALITY BASED APPLICATION FOR EDUCATION” dataset and scope will be scalable.The system will be capable of tracking all kinds of images on its own by implementing ML with AR rather than uploading each and every tracking image.Once clicked on the 3D image it would take us to the website page .

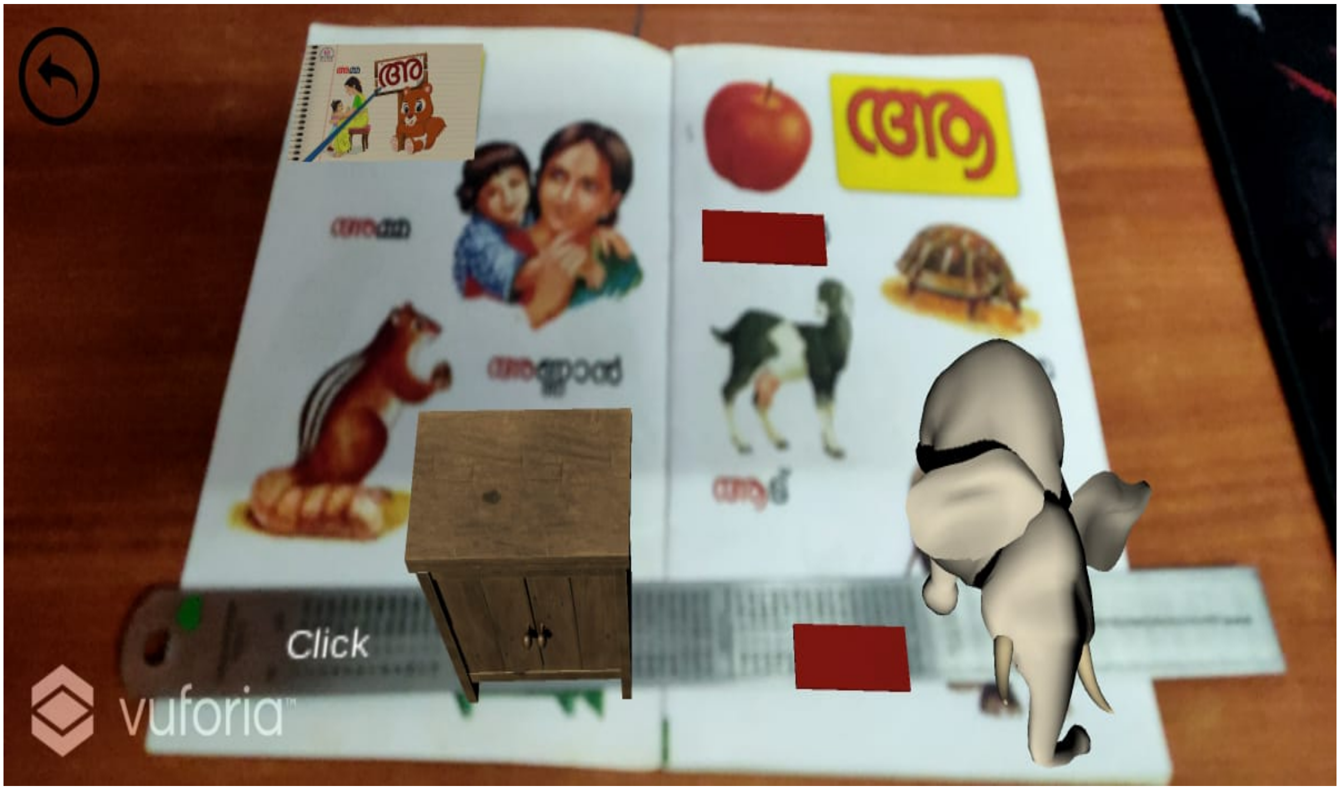
APPENDICES

A1 SAMPLE SCREENS

(i)HOME PAGE

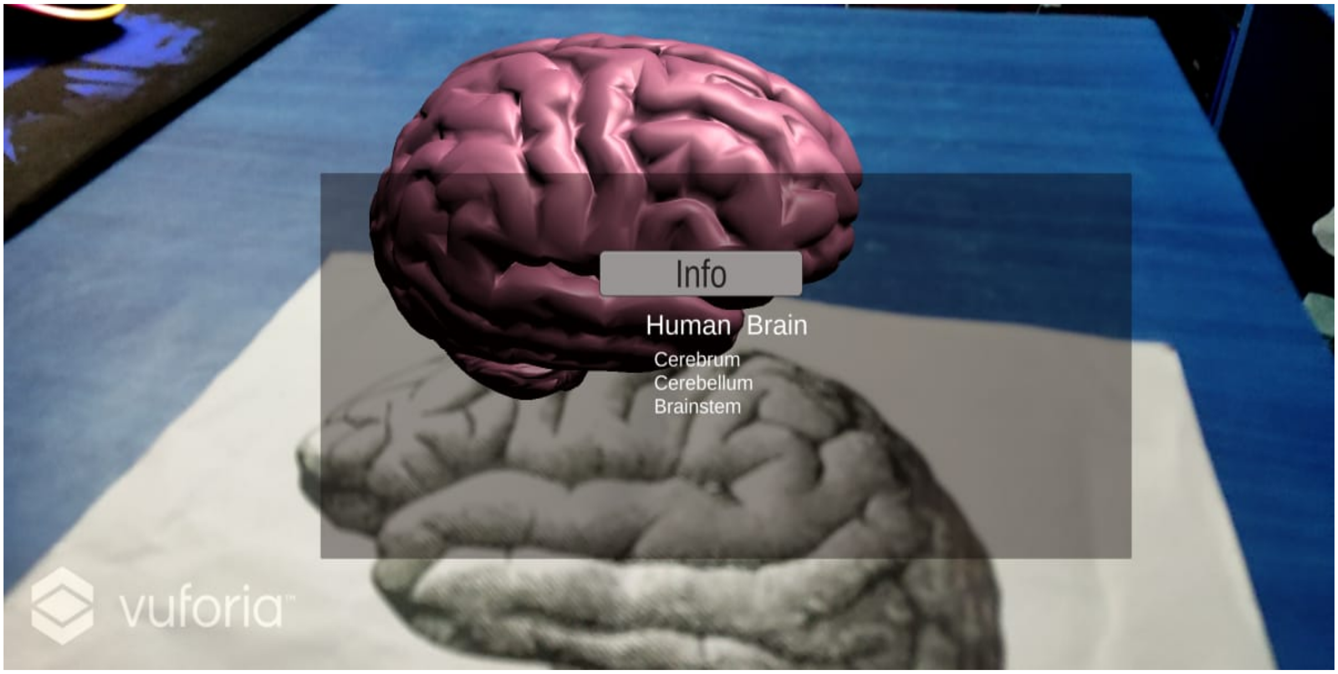


(ii)AR BOOK



(iii) SUN AND EARTH

(iv)AR BRAIN



REFERENCES

1. W. Lu, J. Han and L. Geng, "Intelligent Classroom Information System," 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA), Xiangtan, China, 2019, pp. 259-262, doi: 10.1109/ICSGEA.2019.00067.
2. M. S. Khan and I. Zualkernan, "Using Convolutional Neural Networks for Smart Classroom Observation," 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Fukuoka, Japan, 2020, pp. 608-612, doi: 10.1109/ICAIIC48513.2020.9065260.
3. X. Deng and R. Zhang, "Smart Learning Environment: A Case on the Construction of Smart Classrooms in Colleges and Universities in Guangzhou," 2019 International Symposium on Educational Technology (ISET), Hradec Kralove, Czech Republic, 2019, pp. 262-264, doi: 10.1109/ISET.2019.00062.
4. M. G., J. K.R. and K. Bijlani, "A Smart Phone Integrated Smart Classroom," 2016 10th International Conference on Next Generation Mobile Applications, Security and Technologies (NGMAST), Cardiff, UK, 2016, pp. 88-93, doi: 10.1109/NGMAST.2016.31.
5. N. Songkram, "Virtual smart classroom to enhance 21st century skills in learning and innovation for higher education learners," 2017 Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU), Toyama, Japan, 2017, pp. 1-4, doi: 10.23919/ICMU.2017.8330109.

6. Nallapaneni Manoj Kumar , P. Ranjith Krishna, Pavan Kumar Pagadala, N. M. Saravana Kumar “Use of Smart Glasses in Education” 2018 Proceedings of the Second International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2018,doi: 10.1109/I-SMAC.2018.8653666

7. Khan M M H, Jeffrey C L Chiang “Using Mobile Devices & Social Media in Supporting Engineering Education, 2014 IEEE Global Engineering Education Conference (EDUCON), Istanbul, Turkey,doi10.1109/EDUCON.2014.6826241
8. Gambo Yusufu, Nachandiya Nathan ,” A Novel Model Of Smart Education For The Development Of Smart University System” 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), Ayobo,Nigeria,doi: 10.1109/ICMCECS47690.2020.240912
9. Nikitha Kommera , Faisal Kaleem , Syed Mubashir Shah Harooni ,” Smart Augmented Reality Glasses in Cybersecurity and Forensic Education”,2016 IEEE Conference on Intelligence and Security Informatics (ISI), Tucson, AZ, USA,doi:10.1109/ISI.2016.7745489