DÉCOR : Augmented Reality Based Application To Enhance Interior Designing Using Marker-less Tracking

A PROJECT REPORT

Submitted by

REGISTER NO:211417104071

REGISTER NO:211417104066

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 **"AUTOMATED ALGORITHMIC ACCIDENT PREVENTION SYSTEM"** is the bonafide work of **" GULLAPALLI LAKSHMI SWARAJ (2017PECCS071), GAYATRI KP (2017PECCS066),** who carried out the project work under my supervision.

SIGNATURE	SIGNATURE
Dr.S.MURUGAVALLI,M.E.,Ph.D.,	**Dr.KAVITHA SUBRAMANI**
HEAD OF THE DEPARTMENT	**ASSISTANT PROFESSOR,**
DEPARTMENT OF CSE,	DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,	PANIMALAR ENGINEERING COLLEGE,
NAZARATHPETTAI,	NAZARATHPETTAI,
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 Dr. KAVITHA SUBRAMANI,(M.E., Ph.D.)** and all the faculty members of the Department of CSE for their advice and suggestionsfor the successful completion of the project.

G. LAKSHMI SWARAJ

GAYATRI KP

ABSTRACT

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.

In early days if users wanted to buy a furniture objects without visiting the shops it was possible but it was not possible to check how the object actually looks in home structure. Now in the proposed system, it is possible for user to buy the furniture objects sitting in the home without visiting the shops. The main purpose of the “DÉCOR : Augmented Reality Based Application To Enhance Interior Designing Using Marker-less Tracking” is to develop an android application for trying different furniture in virtual way using a mobile which supports AR camera. The application will eliminate the human efforts by physically visiting the furniture store which is very time-consuming activity. Besides, this it might be easier to use this technique in Online shopping as an option for user to try out the furniture items in their room they are thinking to buy and allows user to visualize the room how it will look after placing furniture in it. User can try out multiple combinations virtually, without physical movement of furniture items. The motivation here is to increase the time efficiency and improve the accessibility of furniture try on by creating furniture augmented reality application.

This system will help the customer to view the furniture object virtually in real environment before buying the object. Due to this system customer will come to know how his home structure would look after buying the furniture object. This system would let the user to try multiple combination of object virtually without physical movement of furniture objects. These will help the buyer to determine how to setup furniture in home structure.

Keywords: Augmented Reality, Marker Detection, Rendering

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

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.

1.2 PROBLEM DEFINITION

People usually buy furniture by physically going to the showroom. This sometimes might lead to a situation where after buying that piece of furniture the customer might not be satisfied with how it looks in his home. So, to overcome that we can use this application to check whether the furniture is adjustable or not which can be placed in the customer living area like home or office using augmented reality images.

Our application is a step in this direction, allowing users to view a 3D rendered model - a virtual resemblance of the physical furniture without any interruption of the markers - which can be viewed and configured in real time using our Augmented reality application.

This study proposes a new method for applying Augmented Reality technology to furniture, where a user can view virtual furniture and communicate with 3D virtual furniture data using a dynamic and flexible user interface.

2. LITERATURE SURVEY

The research for augmented reality technology has brought up development of various applications in the field of computer science. In this literature review, it shows how the implementation of augmented reality in various fields using unity 3D.

Kalyan P. Subbu, Chi Zhang has proposed Analysis and status quo of smartphone-based Indoor Localization Systems with critical analysis on their properties such as accuracies across various sensor designs, energy consumption and computational cost. . The need for location estimation, fueled mainly by inaccuracies of GPS indoors, has been addressed by specifically designed systems achieving a high localization accuracy but with a high deployment cost. Thus, the proposed system takes up the goal of surveying state-of-the-art smartphone based indoor localization system.

Johannes Klinglmay, Bernhard Bergmair, Evangelos Pournara has proposed a new concept of how individual consumers can follow their own understanding of sustainability, while at the same time benefiting from collective and participatory actions. They discuss how the means of ICT can be used to develop political consumerism further to transform individual policies into collective statements. Here they use ASSET platform to track the customers consumerism.

Liang Li , Xiuquan Qiao has proposed Rendering Optimization for Mobile Web 3D Based on Animation Data Separation and On-Demand Loading which does model-animation data separation and an on-demand loading mechanism to improve the data request and loading process of Web 3D models as the existing WebGL 3D JavaScript libraries for Web-based mobile 3D (represented by three.js and babylon.js) load the entire model file at once, large-size 3D models with complex interactions cannot be rendered smoothly due to limited data transmission, the weak computation capabilities of mobile Web browsers, and the latency of 3D model rendering.

Jingbin Liu , Ruizhi Chen has proposed a smartphone indoor positioning engine named HIPE that can be easily integrated with mobile LBS. HIPE is a hybrid solution that fuses measurements of smartphone sensors with wireless signals. The system will mitigate the risk of overfilled bins and unsanitary conditions that are caused by the lack of information that is present in the current collection process. This system uses Location based services, hidden Markov model.

Mami Mori, Jason Orlosky, Kiyoshi Kiyokawa, Haruo Takemura has proposed A Transitional AR Furniture Arrangement System with Automatic View Recommendation. Subjects were first asked to memorize a furniture layout presented on a desktop monitor. They were allowed to examine and manipulate the 3D model using a mouse for as long as they wished. Once satisfied, they wore the video see-through headset and used a game pad to place the furniture objects based on their memory of the scene. It uses Augmented Reality for visualization. It can be further made as a transitional AR furniture arrangement system that recommends a secondary view that can improve a user's understanding of a room layout.

Khushal Khairnar, Kamleshwar Khairnar, Sanket kumar Mane, Rahul Chaudhari has proposed a technique named "Furniture Layout Application Based on Marker Detection and Using Augmented Reality" to develop an application where user have to place the marker in a room where he want to try out furniture items. The user's webcam will be on and through the webcam he will capture the live feed of the room. Then application search the marker using fiducial marker detection algorithm. To identify the position of marker using direct linear transformation algorithm. Whichever furniture object the user want to try out he will select that object from the database. Then the application will superimpose 3D object. In three dimensional objects are overlaid on to the two dimensional image frame acquire from webcam.

Snehal Mangale, Nabil Phansopkar, Safwaan Mujawar, Neeraj Singh has proposed technique named “Virtual Furniture Using Augmented Reality”[2] which is a web based application where user, have to place the marker in a room where they want to try out furniture items. The user’s webcam will be on and through the webcam they will capture the live feed of the room. Application captures the image and passes through predefined marker detection algorithm. Algorithm is based on image processing techniques using color and other properties as the input to detect the marker. User initially selects the furniture to be placed from the given database. The application superimposes furniture on the original image with the center coinciding with the markers center in both directions. Furniture objects are overlaid on to the two dimensional image frame acquire from webcam. This will appear as if it is actually placed in the real world. And finally the user can view how the area looks with the furniture present.

SR NO	TITLE	YEAR	AUTHOR	CONCEPT	TECHNIQUE	FUTURE SCOPE	DRAWBACK
1.	ANALYSIS AND STATUS QUO OF SMARTPHONE-BASED INDOOR LOCALIZATION SYSTEMS	2017	KALYAN P. SUBBU, CHI ZHANG	The goal of surveying state-of-the-art smartphone based indoor localization systems with critical analysis on their properties such as accuracies	Analysis of existing indoor localization	The limitations are analyzed for the future way of execution	

				across various sensor designs, energy consumption and computational cost			
2.	Sustainable Consumerism via Self-regulation	2016	Johannes Klinglmay, Bernhard Bergmair, Evangelos Pournara	In this paper they introduce a new concept of how individual consumers can follow their own understanding of sustainability, while at the same time benefiting from collective and participatory actions.	ASSET platform is used to track the customers consumerism	build upon the intrinsic motivation of the individual consumer and at the same time facilitate self-regulation among consumers towards a sustainable consumerism.	Useful for analysing the consumerism. Cannot be used for real time implementation
3.	Rendering Optimization for Mobile	2020	LIANG LI , XIUQUAN QIAO	In this paper, they first propose model-	Augmented Reality in 3D modelling	Main focus is on preloading	Dealing with 3D models comes with high

	Web 3D Based on Animation Data Separation and On-Demand Loading			animation data separation and an on-demand loading mechanism to improve the data request and loading process of Web 3D models.		large-data volume animations using context-aware methods and a collaborative rendering method based on mobile edge servers and cloud servers	computation cost.
4.	A Hybrid Smartphone Indoor Positioning Solution for Mobile	2012	Jingbin Liu, Ruizhi Chen	This system proposes a smartphone indoor positioning engine named HIPE. HIPE is a hybrid solution that fuses measurements of smartphone sensors with wireless	Location based services, hidden Markov model	the system will mitigate the risk of overfilled bins and unsanitary conditions that are caused by the lack of information that is present in the current	In the future, other smartphone sensors, such as cameras and gyroscopes, will be integrated with HIPE to measure MDI.

				signals.		collection process.	
5.	A Transitional AR Furniture Arrangement System with Automatic View Recommend ation	2016	Mami Mori, Jason Orlosky, Kiyoshi Kiyokawa, Haruo Takemura	Subjects were first asked to memorize a furniture layout presented on a desktop monitor. They were allowed to examine and manipulate the 3D model using a mouse for as long as they wished	Augmented Reality for visualization.	a transitional AR furniture arrangement system that recommends a secondary view that can improve a user's understanding of a room layout	It is desktop based application ands needs stereo camera and should be placed at minimal and marker based.
6.	Approach to The Interior Design using Augmented Reality Technology	2015	Jiang Hui	AR technology is applied to allow a person (the designer, the manufacturer, and the consumer) to combine real and virtual information and objects	To analyze and evaluate the impact of Augmented Reality Technology in interior design. The research context is associated at the consumers perception of the project in	interior design can reduce the cost and provide the multimedia augmentation of high vivid simulations for user	Requires huge amount of information and quality of object is low.

				in a physical, real world environment .	the management and execution of the interior design.	in real time	
7.	Use of Augmented Reality in furniture industry	2015	Elizabeth Carvalho, Gustaava Macaes, Isabel Varajao, Nuno Sousa, Paulo Brito	a configurable and modular platform that has been developed to solve this gap between the designer's and the client's visual models. It offers an immersive preview of the interior decoration, customizable to the spatial real world visualization area having Augmented Reality (AR) as its technologica l background	Use of Augmented Reality in furniture industry	Use Simultane ous Localizati on and Mapping (SLAM) that helps in fixing to rigid body	Marker based and can be placed only on the predefined images.

8.	Virtual Furniture Using Augmented Reality	2016	Snehal Mangale, Nabil Phansopkar , Safwaan Mujawar, Neeraj Singh	It is a web based application where user, have to place the marker in a room where they want to try out furniture items. The user's webcam will be on and through the webcam they will capture the live feed of the room.	Web based Application for trying out furniture.	It is faster in image capturing and provide high resolution for furniture 3D model	It is based on personal computer webcam and every time the captured image is to be loaded into web application for processing

Table 2.1 Literature Survey

3. SYSTEM ANALYSIS

EXISTING SYSTEM

Traditional methods of designing include advising and assisting customers who have relied upon a combination of verbal explanations and 2D drawings through online shopping application.

However, this medium of approach clearly restricted to the limit of explanations provided to customer for the particular placement of a furniture and makes him less efficient and confused to buy the furniture.

The main drawbacks in the mediums of existing system are:

• Static view of design which is unable to convey

• Cannot determine the furniture will fix to our needs.

• Information like height and breadth can’t be known.

PROPOSED SYSTEM

With the approach of augmented reality application, this can be easily achieved. Interior designing is a field where augmented reality has not been able to get its grip to its fullest. People today are well versed with the technology and are operating smartphones which support AR. Thus, the concept of creating a furniture layout-based application brings the designer 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 application, we use Vuforia to accurately detect the real- world environment, allowing users to place virtual objects into a real context.

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.

The basic premise of the proposed system is to overlay digital 3D models on top of real things using a camera.

• This Application will use AR supported mobile phone to scan the living area and display the augmented furniture object to check whether it adjusts or not and that helps in better choosing of the right furniture for our need.

• Augmented objects are the virtual objects (3D Model) which are similar to furniture tool

• The next step involves setting up light, shadow, and camera positioning of these models using various components of Unity 3D.

• Next, the furniture model is selected and the selected model is rendered and processed to be loaded on the scanned surface by Vuforia.

• Mapping of 3D model onto the smartphone screen takes place which decides the dimensions of the model which is then rendered and displayed onto the screen.

.

REQUIREMENT ANALYSIS AND SPECIFICAITON

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.

Software Requirements

Language : C#

Operating system : Android 8.0 or more

Tools

- Unity 3D
- Vuforia
- Android Studio

Hardware Requirements

- RAM Capacity : 4GB
- Memory : 120 MB
- Graphics Card : 1 GB
- Accessories : Smart phone with AR support

Software Specifications

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.

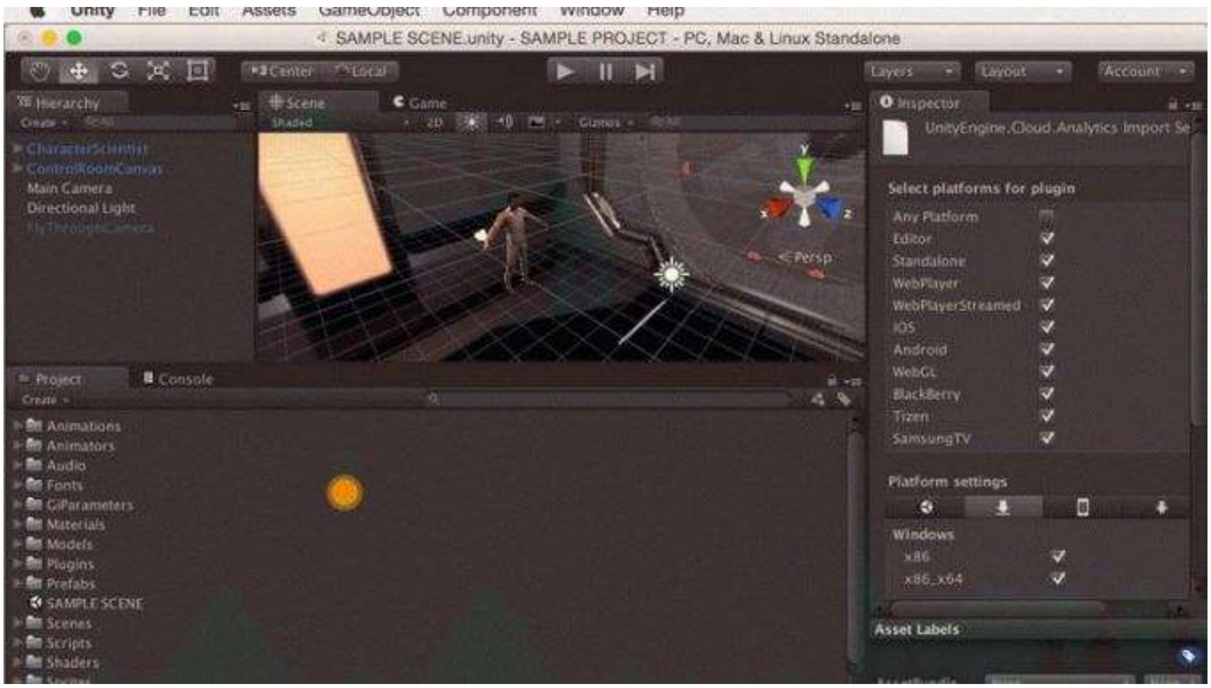


Fig 3.1: Unity software interface

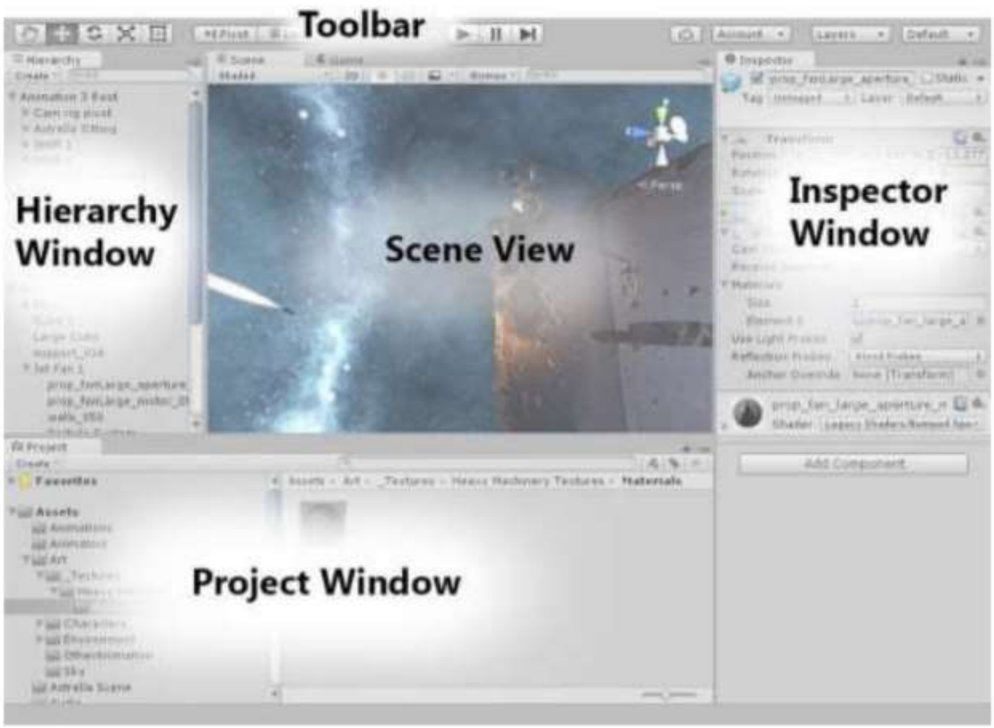


Fig 3.2: Interface options

As shown in above figure 3.2, 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.

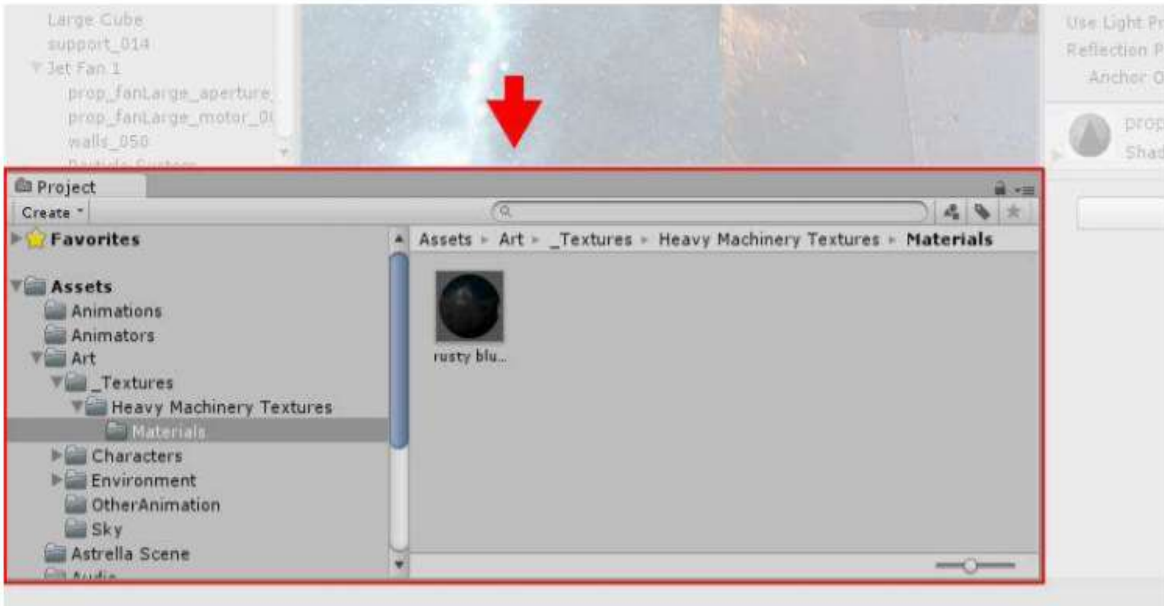


Fig 3.3: Project window

As shown in above figure 3.3, 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.

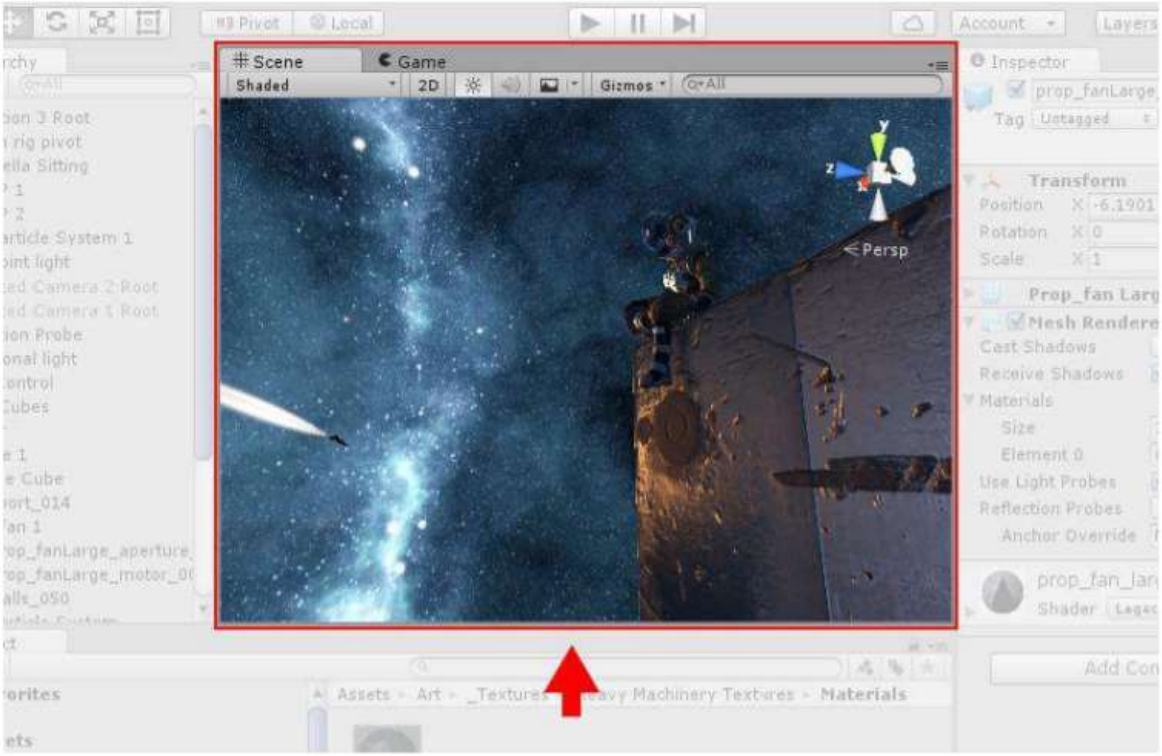


Fig 3.4: Scene view window

As shown in the above figure 3.4 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.

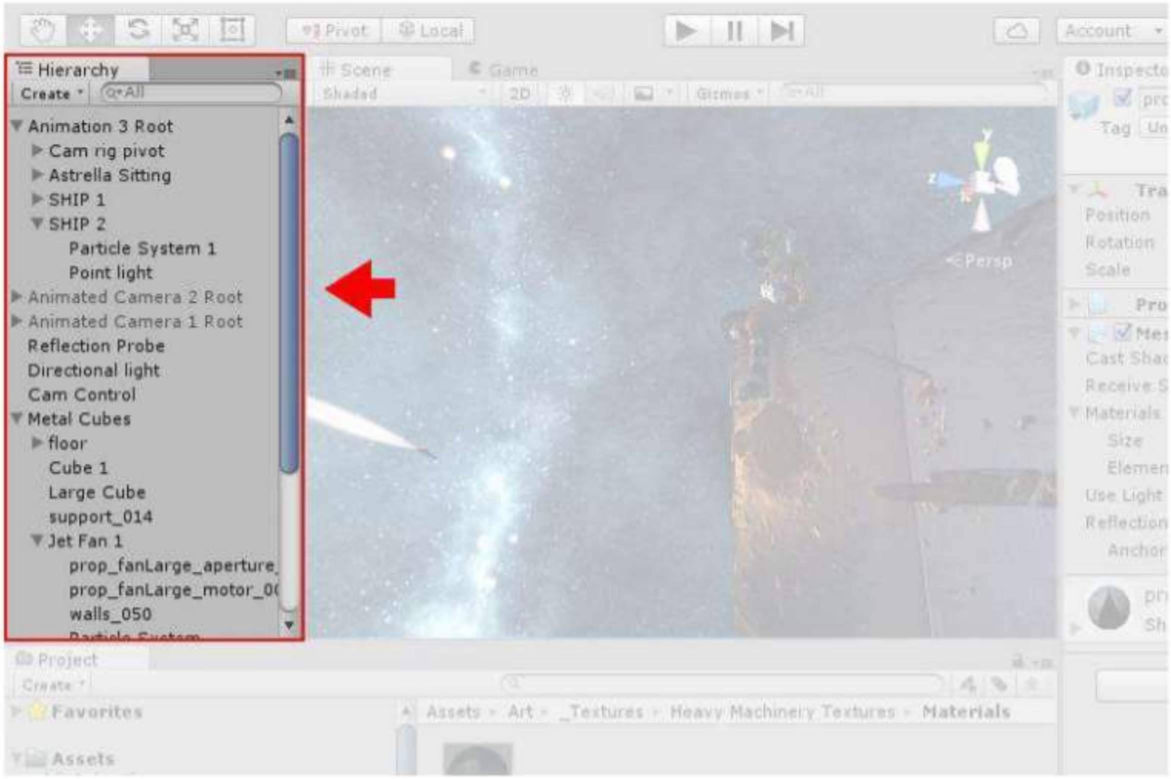


Fig 3.5: Hierarchy window

As shown in above figure 3.5, 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.

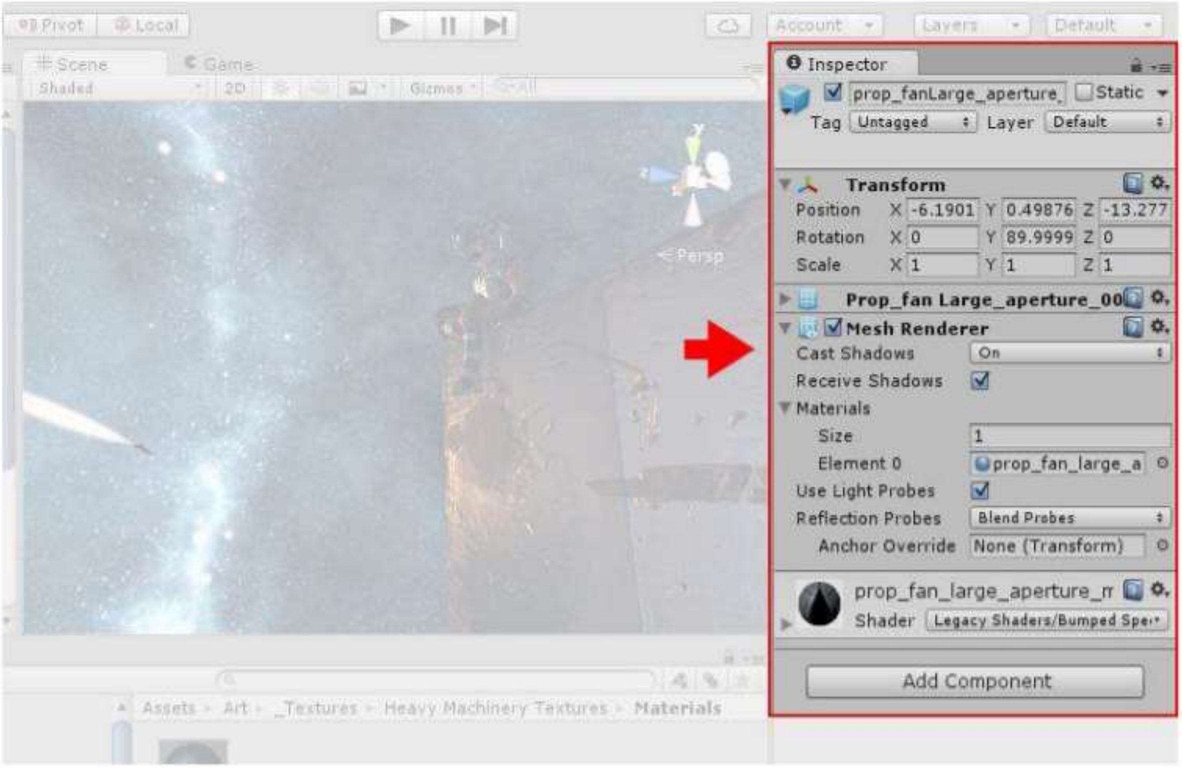


Fig 3.6: Inspector window

As shown in above figure 3.6, 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.

Fig 3.7: Toolbar window

As shown in above figure 3.7, 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 center 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.

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.

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.

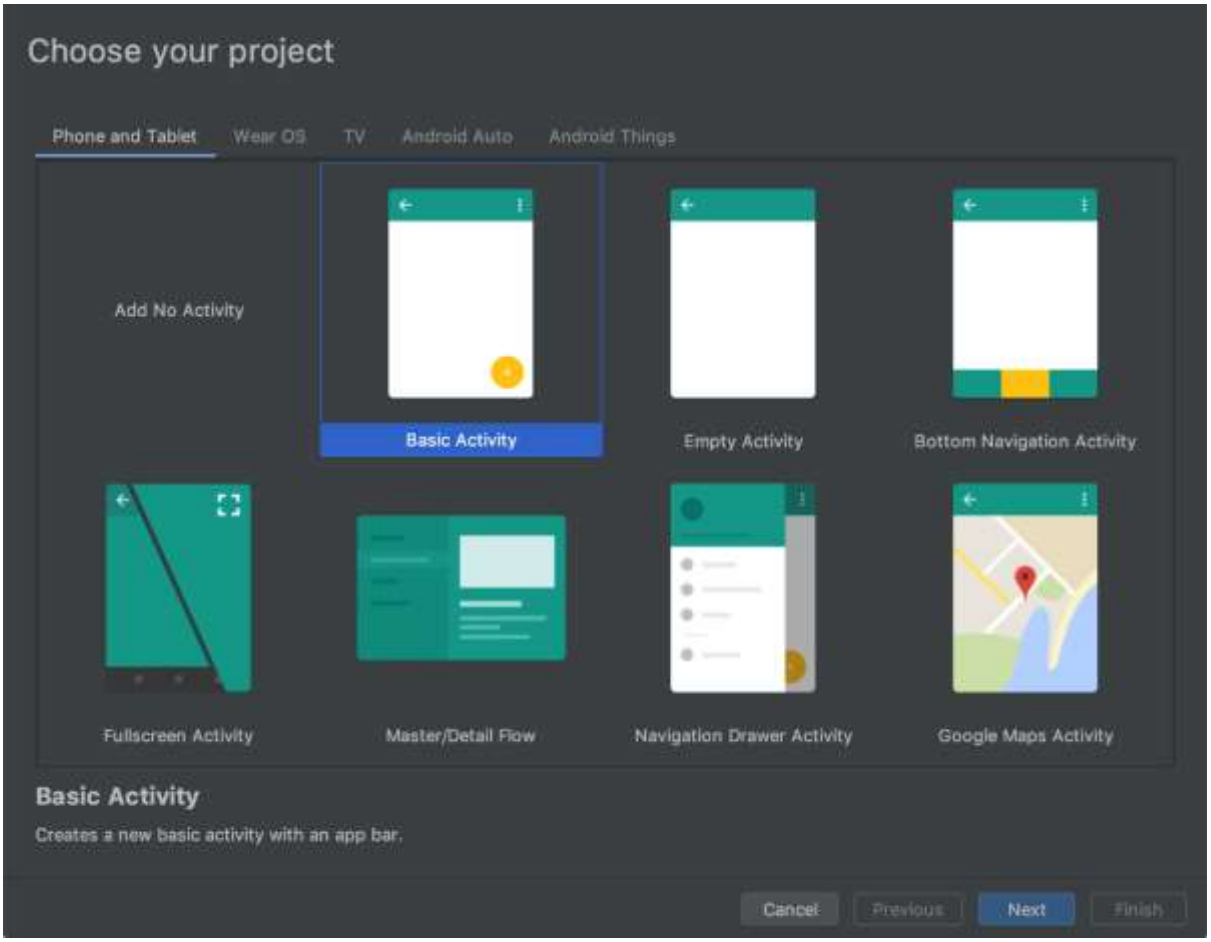


Fig 3.8 Android Studio

4. SYSTEM DESIGN

ER DIAGRAM FOR AUGMENTED REALITY SYSTEM

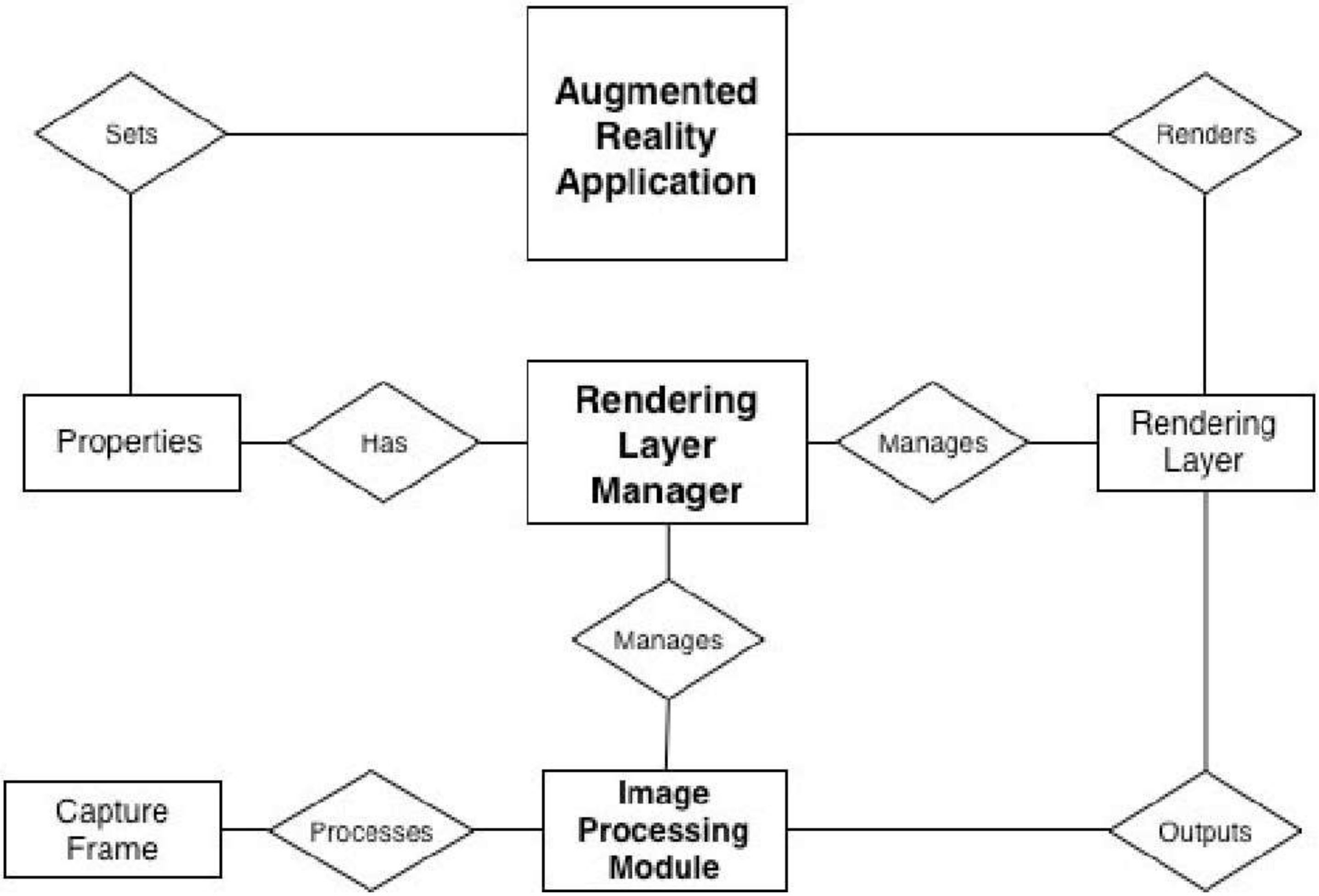


Fig 4.1 ER diagram

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 Language™ (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

USE CASE DIAGRAM

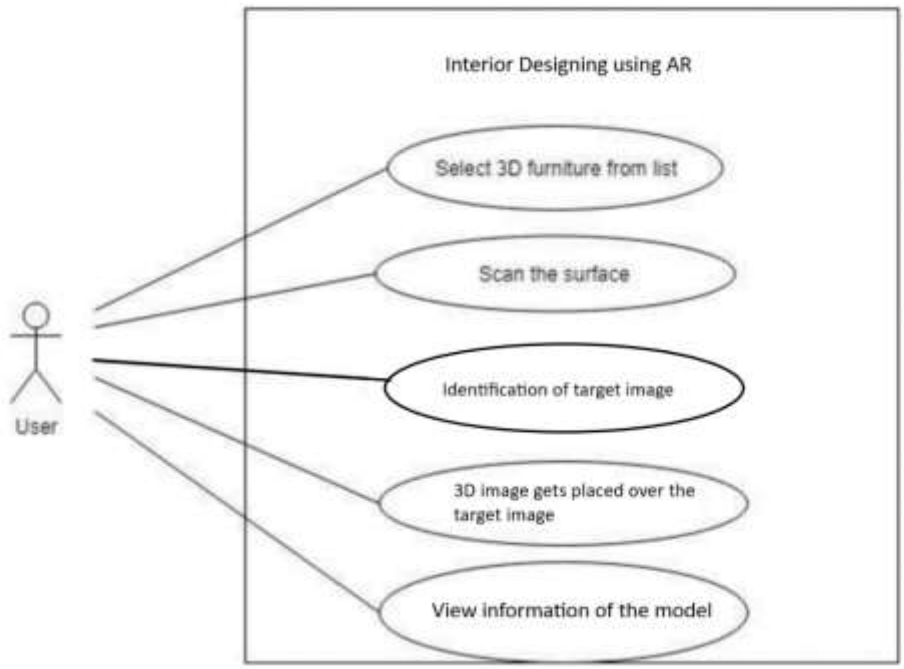


Fig 4.2: Use case diagram

As shown from the above figure 4.2, it describes the use case diagram of the application where the user interacts with application using Android device that supports AR camera. Initially, we select the model in which we are interested then scan the surroundings using camera of device which recognizes the target image and the 3D model gets placed on it based on which we can verify whether it fulfils our needs, if satisfied then we can view the information of that piece of furniture. The actor here is the user and uses cases select 3D furniture, scan the surface, identify target image, place the 3D furniture and view the information.

CLASS DIAGRAM

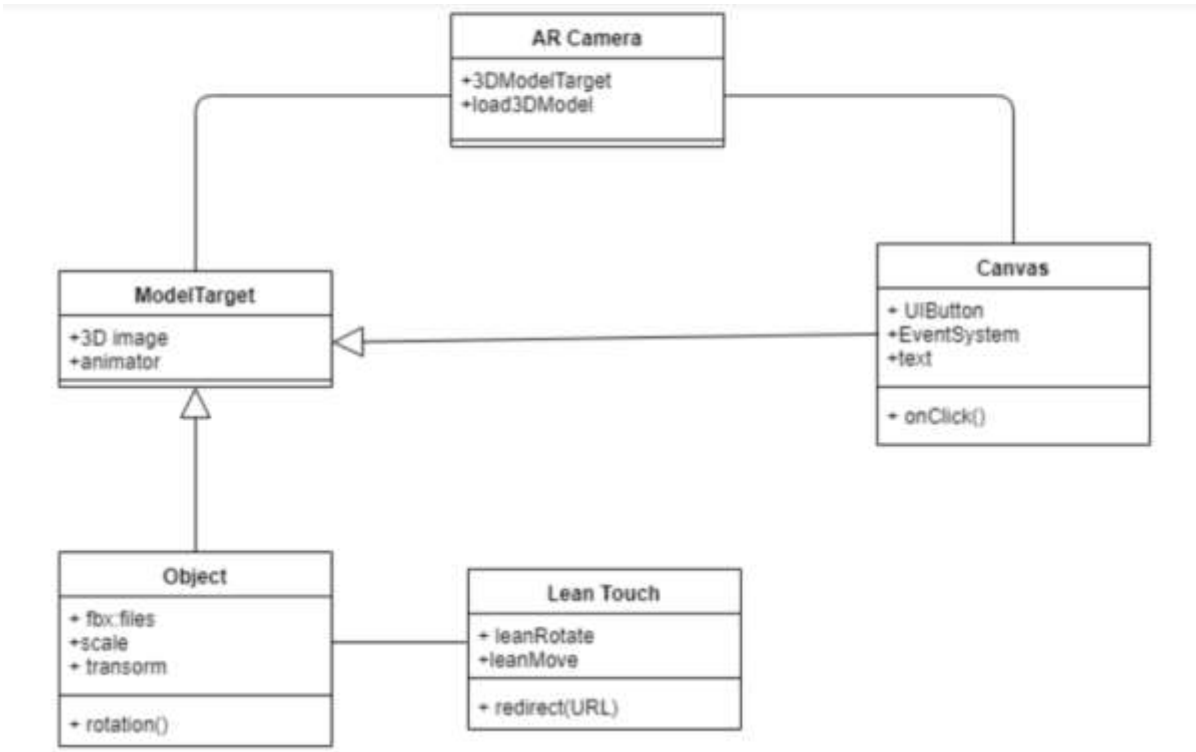


Fig 4.3: Class diagram

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.

SEQUENCE DIAGRAM

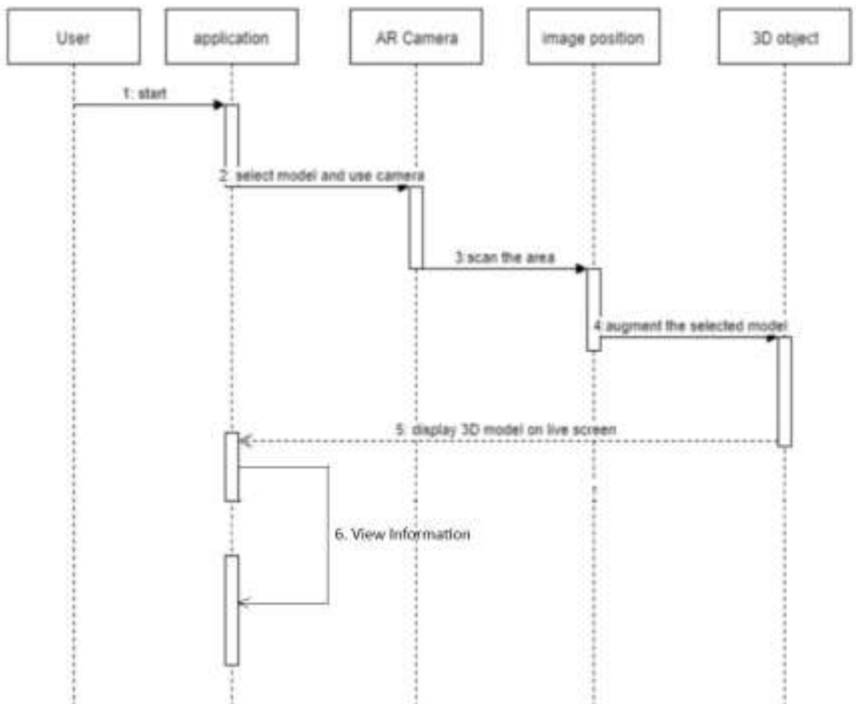


Fig 4.4: Sequence diagram

As shown in the above figure 4.4, 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. Then the user can view the information if he likes that item.

ACTIVITY DIAGRAM

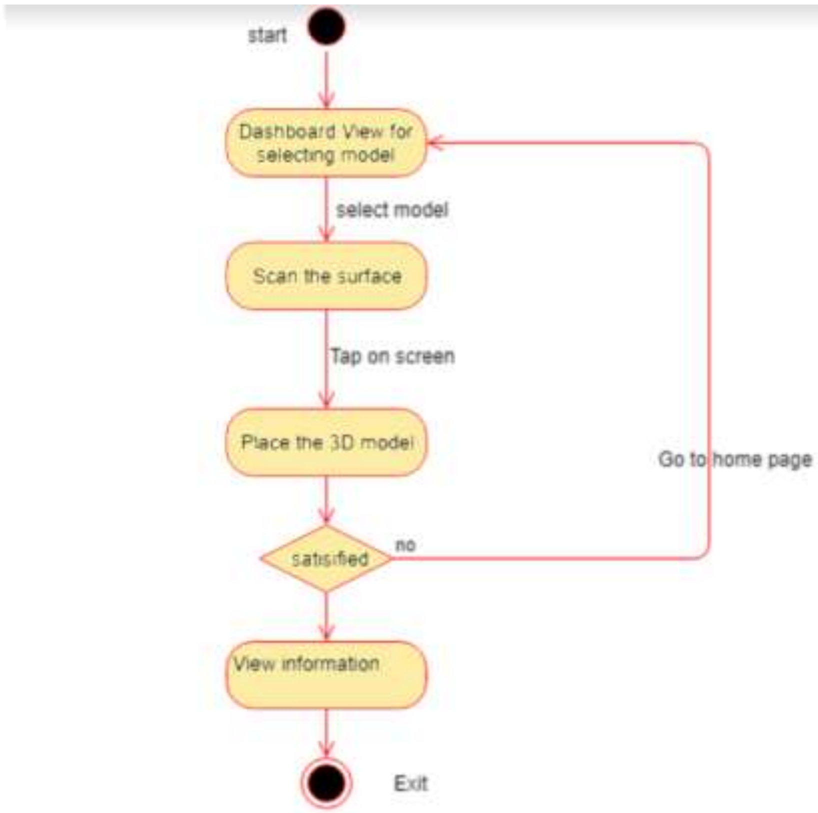


Figure 4.5: Activity diagram

As shown from the above figure 4.5, 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, place 3D model and finally if satisfied view information.

5. ARCHITECTURE

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 description and 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.

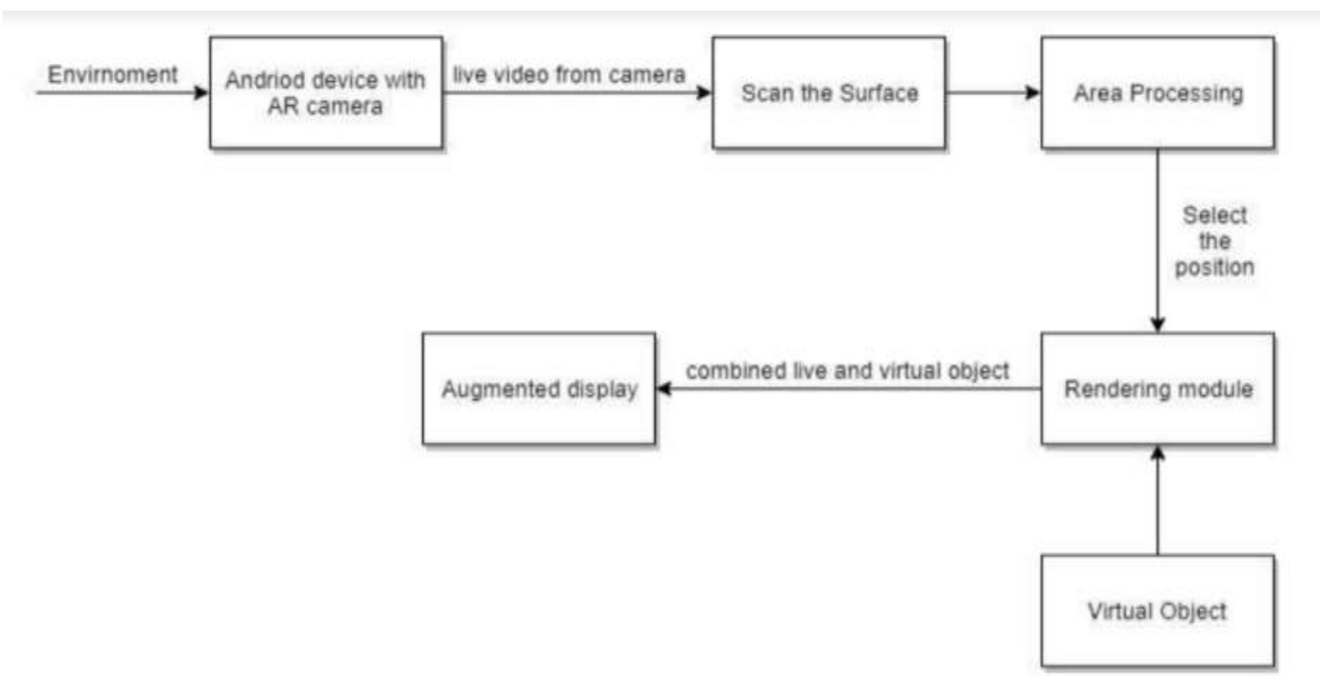


Fig 5.1: System Architecture of Application

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.

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:

3.2.1 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.

3.2.2 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.

he 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.

3.2.3 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.

3.2.4 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#.

6. SYSTEM IMPLEMENTATION

/* Add Vuforia engine package */

```
using System;
using System.IO;
using System.Linq;
using System.Text;
using UnityEditor;
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 = "9.8.5";
   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 AddVuforiaEnginePackage()
   {
      if (Application.isBatchMode)
         return;

      var manifest = Manifest.JsonDeserialize(sManifestJsonPath);

      if (!IsUsingRightGitUrl(manifest))
         DisplayAddPackageDialogue(manifest);
   }
```

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

    static bool VersionNumberIsTheLatest(string package)
    {
        var version = package.Split('#');
        if (version.Length >= 2)
        {
            version[1] = version[1].TrimEnd(new []{ '"' });
            return IsCurrentVersionHigher(version[1]);
        }

        return false;
    }

    static bool IsCurrentVersionHigher(string currentVersionString)
    {
        if (string.IsNullOrEmpty(currentVersionString) ||
string.IsNullOrEmpty(VUFORIA_VERSION))
            return false;

        var currentVersion = TryConvertStringToVersion(currentVersionString);
        var updatingVersion = TryConvertStringToVersion(VUFORIA_VERSION);
        if (currentVersion >= updatingVersion)
            return true;

        return false;
    }

    static Version TryConvertStringToVersion(string versionString)
    {
        Version res;
        try
        {
            res = new Version(versionString);
        }
```

```
        catch (Exception e)
        {
            return new Version();
        }

        return new Version(res.Major, res.Minor, res.Build);
    }


    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}\n\n" +
            $"Please make sure that Git is installed and on your PATH environment variable.", "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(':');
      dependencies[i] = kvp[0] + ": " + versionEntry;
      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 { }
}
```

/* Change scene */

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

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

    public void bedScene()
    {
        SceneManager.LoadScene("Beds");
    }

    public void flowerScene()
    {
        SceneManager.LoadScene("FlowerPots");
    }

    public void couchScene()
    {
        SceneManager.LoadScene("Couches");
    }

    public void tableScene()
```

```
    {
        SceneManager.LoadScene("Tables");
    }

    public void infoScene()
    {
        SceneManager.LoadScene("Information");
    }
    public void page2Scene()
    {
        SceneManager.LoadScene("Page2");
    }
    public void page3Scene()
    {
        SceneManager.LoadScene("Page3");
    }
    public void exitApp()
    {
        Application.Quit();
    }
}
```

/*Rotating the model*/

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

public class RotateModels : MonoBehaviour
{
    public GameObject activeModel, model2, model3;
    public int x, y, z;

    void Start()
    {
        activeModel.SetActive(false);
        model2.SetActive(false);
        model3.SetActive(false);

    }
```

```
    public void onclick()
    {
        activeModel.SetActive(true);
        model2.SetActive(false);
        model3.SetActive(false);
    }

    public void onclickmodel2()
    {
        activeModel.SetActive(false);
        model2.SetActive(true);
        model3.SetActive(false);
    }
    public void onclickmodel3()
    {
        activeModel.SetActive(false);
        model2.SetActive(false);
        model3.SetActive(true);
    }
    public void rotateLeft()
    {
        activeModel.transform.Rotate(x, y, z);
    }

    public void rotateRight()
    {
        activeModel.transform.Rotate(-x, -y, -z);
    }


}
```

/*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"
  }
}
```

7. TESTING

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 haveintegrated all modules into one system, so we have to check if this runs OK andthat no errors are produced because all the modules are in one system.

Unit 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.

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.	Rotating the 3D object in clock wise direction	Rotation successful.	Rotation successful.	Pass
8.	Changing the scene from one mode to another mode	Changing of scene from one mode to another mode was successful.	Changing of scene from one mode to another mode was	Pass

			successful.	
9.	Building app.	App has been built.	App has been built.	Pass
10.	Building app.	App has been built.	App has been built.	Pass

Table 7.1 Test Cases

7. CONCLUSION AND FUTURE SCOPE

CONCLUSION

The main objective of this "AUGMENTED REALITY BASED APPLICATION TO ENHANCE INTERIOR DESIGNING USING MARKERLESS TRACKING" is to analyse the use of augmented reality to render the furniture model in real world. Augmented reality technology that allows the customers to decide and interact the furniture with the real world, offering new possibilities for furniture online shopping. It helps the customer to view and understand the furniture for his requirements. Due to this customer will come to know how their home structure would look after purchasing and placing the furniture object with multi-colour option. These helps the buyer in determining how to setup the furniture in their home structure. Augmented reality support for furniture help in creating many new opportunities for future research to anticipate new ideas in the field of online shopping as customer will get benefit with these types of applications and gives a better understanding and decision making for purchasing a furniture in an efficient way. Augmented reality is new evolving technology in the field of computer science and will make us much more helpful than the traditional technologies .

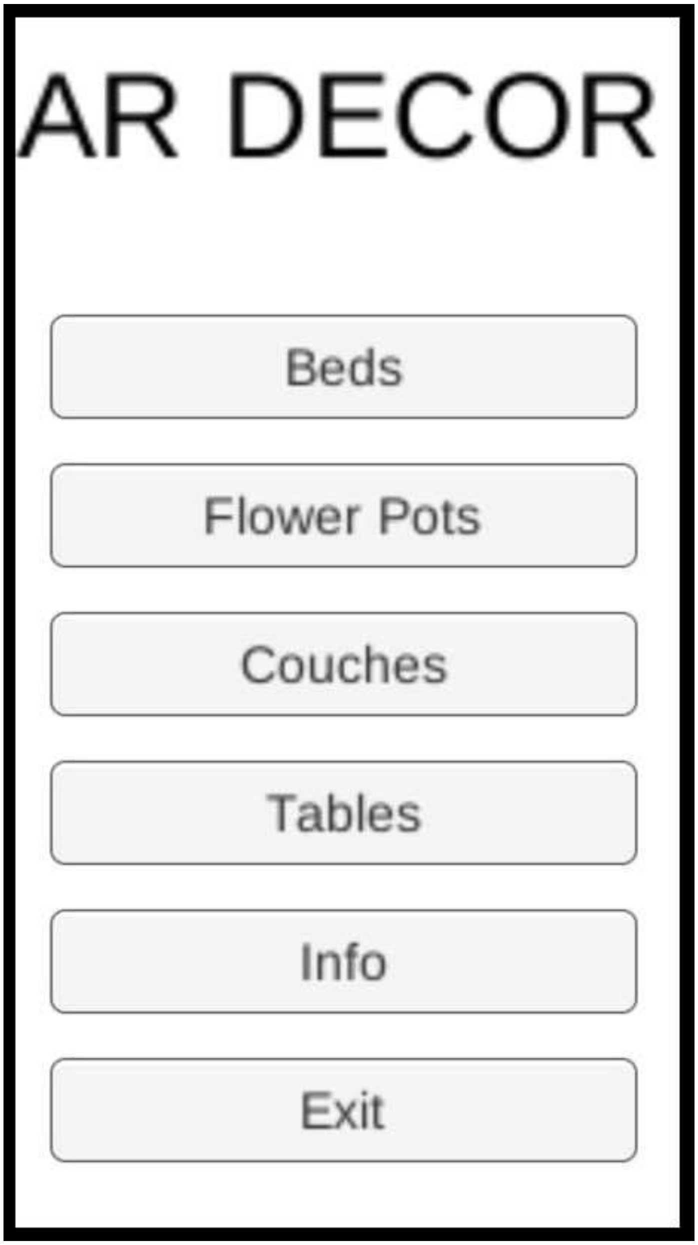
FUTURE ENHANCEMENT

In future our "AUGMENTED REALITY BASED APPLICATION TO ENHANCE INTERIOR DESIGNING USING MARKERLESS TRACKING" dataset and scope will be scalable. The user might not only be able to try out different furniture objects but they can also try out this application by trying on garments, goggles, watches, hair styles etc. It can also be used for various applications in shopping malls, interior designing, Medical Science etc. New technology may come into existence in future that will help in developing 3D models automatically.

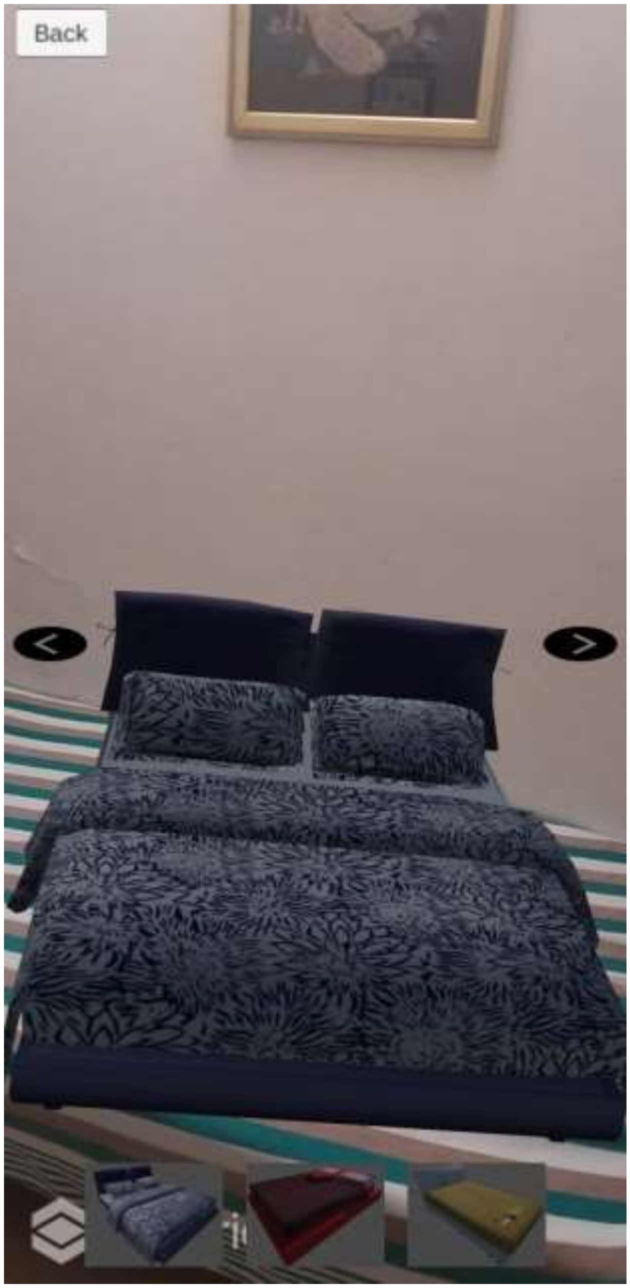
APPENDICES

A 1. SAMPLE SCREENS

(i) HOME PAGE

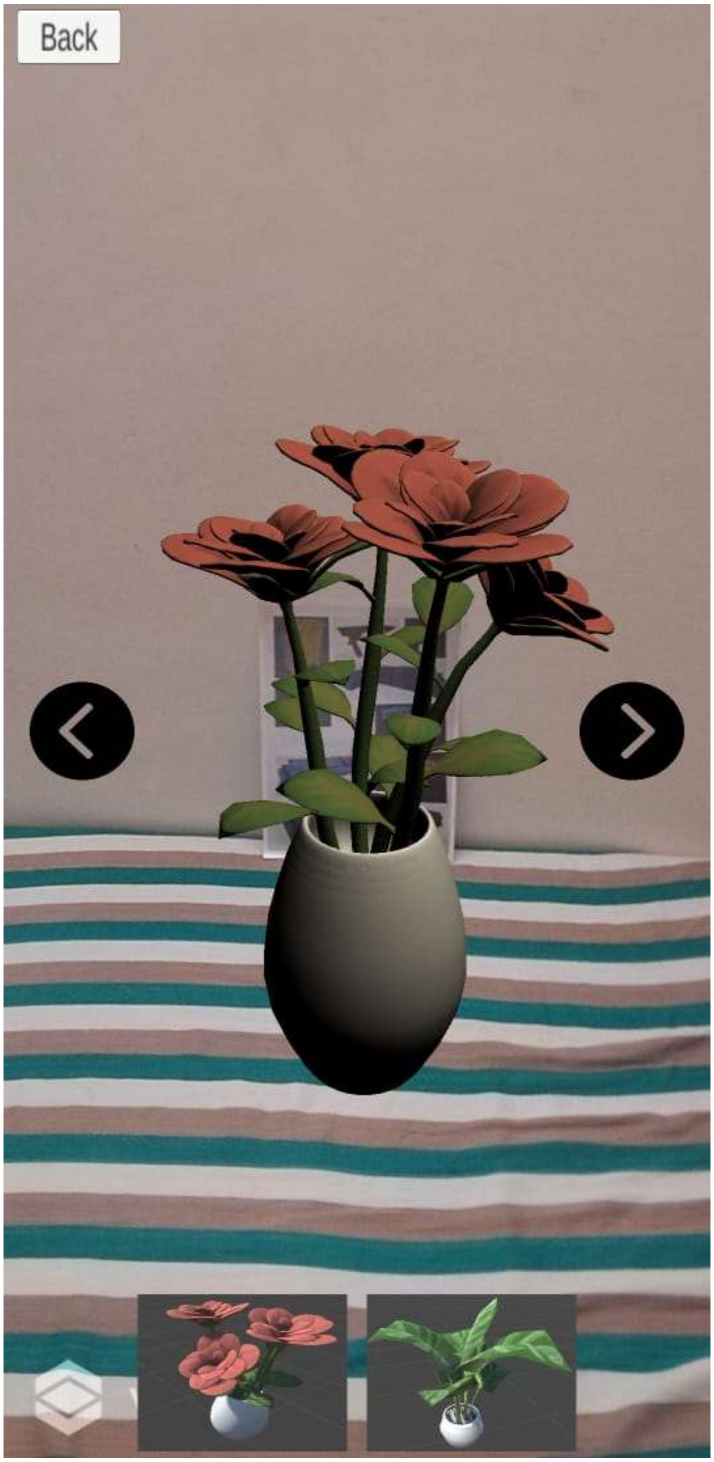


(ii) 3D BED DISPLAY

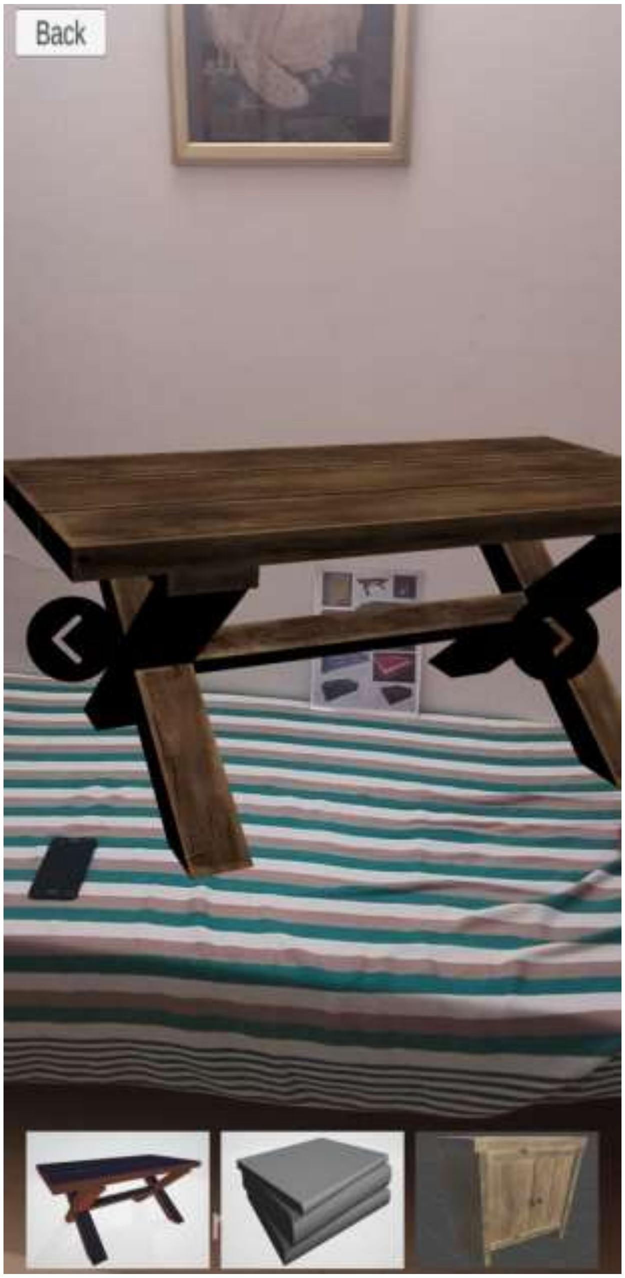


(iii) 3D SOFA DISPLAY

(iv) 3D FLOWER POT DISPLAY



(v) 3D TABLE DISPLAY



(vi) INFORMATION PAGE

REFERENCES

[1] Mami Mori, Jason Orlosky, Kiyoshi Kiyokawa, Haruo Takemura. (2016, Sep.). A Transitional AR Furniture Arrangement System with Automatic View Recommendation. IEEE Adjunct. [Online]. 21(3).pp.21-24. ISBN: 978-1-5090-3740-7. Available: https://ieeexplore.ieee.org/document/7836488

[2] Snehal Mangale, Nabil Phansopkar, Safwaan Mujawar, Neeraj Singh. (2016, May). Virtual Furniture Using Augmented Reality. IOSR Journal of Computer Engineering. [Online]. eISSN: 2278-0661, p-ISSN: 2278-8727, pp.42-46. Available: http://www.iosrjournals.org/iosrjce/papers/Conf.16051/Volume-1/9.%2042-46.pdf?id=7557

[3] Elizabeth Carvalho, Gustavo Maçães, Isabel Varajão, Nuno Sousa and Paulo Brito. (2011, Nov.). Use of Augmented Reality in the furniture industry. Presented at Center for Computer Graphics. [Online]. Available: https://www.researchgate.net/publication/236863499_Use_of_Augmented_Reality_in_the_furniture_industry

[4] Khushal Khairnar, Kamleshwar Khairnar, Sanket kumar Mane, Rahul Chaudhari. (2015, Oct.). Furniture Layout Application Based on Marker Detection. International Research Journal of Engineering and Technology. [Online]. 02(07). p-ISSN: 2395-0072, e-ISSN: 2395- 0056. Available: https://www.irjet.net/archives/V2/i7/IRJET-V2I780.pdf

[5] Billinghurst, M. (2002) "Augmented reality in education" in New Horizons for Learning, 2nd ed., vol.3, New York: McGraw-Hill, 2010, pp. 123-135.

[6] J. M. Prouty, "Robotic construction site marking apparatus,"U.S. Patents 471 883, Nov. 21, 2013.

[7] Trimble. Trimble RTS Series Robotic Total Stations. Accessed: Feb. 24,2019. [Online]. Available: https://buildings.trimble.com/layout/rts

[8] S. Lee and Ö. Akin, "Augmented reality-based computational fieldworksupport for equipment operations and maintenance," Autom. Construct.,vol. 20, no. 4, pp. 338–352, 2011.

[9] R. Sacks, C. M. Eastman, G. Lee, and P. Teicholz, BIM Handbook:A Guide to Building Information Modeling for Owners, Designers,Engineers, Contractors, and Facility Managers, 3rd ed. Hoboken, NJ,USA: Wiley, 2018. [10] O. Bimber and R. Raskar, Spatial Augmented Reality: Merging Real andVirtual Worlds. Boca Raton, FL, USA: CRC Press, 2005.