SMART LEARNING SYSTEM USING AUGMENTED REALITY

A PROJECT REPORT

Submitted by

AJAY HERMAS T (953419104006)

NARAYANAN M (953419104038)

PRAVIN KUMAR R (953419104041)

RAHUL S (953419104042)

In the fulfilment for the award of the degree

BACHELOR OF ENGINEERING

In the stream

COMPUTER SCIENCE AND ENGINEERING

at

VV COLLEGE OF ENGINEERING

TISAIYANVILAI

MAY 2023

ANNA UNIVERSITY : CHENNAI

BONAFIDE CERTIFICATE

This is to certify that this project report entitled **SMART LEARNING SYSTEM USING AUGMENTED REALITY** submitted to **VV COLLEGE OF ENGINEERING, TISAIYANVILAI** is a Bonafide record of work done by **AJAY HERMAS T (953419104006), NARAYANAN M (953419104038), PRAVIN KUMAR R (953419104041), RAHUL S (953419104042)** under my supervision.

SIGNATURE	**SIGNATURE**
Dr. I. Muthulakshmi	**Dr. G.Sumilda Merlin**
Head of the Department	**Supervisor**
Computer Science and Engineering,	Computer Science and Engineering,
VV College of Engineering,	VV College of Engineering,
Tisaiyanvilai.	Tisaiyanvilai.

Submitted for the viva-voce held on **16 / 05 / 2023**

INTERNAL EXAMINAR **EXTERNAL EXAMINAR**

Acknowledgement

First and foremost, I would like to thank " God ", the lord of all creations who is by his abundant grace sustained me to work on this project successfully.

We are very grateful to our Secretary **Shri. S. Jagatheesan** for providing us with an environment to complete our project successfully.

We express my thanks to our principal **Dr. K. S. Saji** for her tremendous contribution and moral support towards the completion of this project.

We wish to express my sincere thanks to the Head of the Department **Dr. I. Muthulakshmi** for the useful suggestions given during the course of the project period.

We are very grateful to my guide **Dr. G. Sumilda Merlin** Assistant professor for being instrumental in the completion of my project with her complete guidance.

We also thank our project co-ordinator **Mrs. T. Karthija M.E., AP** for providing moral support and motivation, all the staff members, my family and friends for their help in making this project a successful one.

Our special thanks to families and friends. I am ensuring that this project was done by me and my team not copied from anywhere.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AR	–	Augmented Reality
QR	–	Quick Response
RAM	-	Random Access Memory
AMD	–	Advanced Micro Devices
SSD	–	Solid State Disk
GB	–	Giga Byte
JDK	–	Java Development Kit
NDK	-	Native Development Kit
SDK	–	Software Development Kit
APK	–	Android Package
UMP	–	Unit Manufacturing Process
JVM	–	Java Virtual Machine
API	–	Application Program Interface
JLS	–	Java Language Specification
SE	–	Standard Edition
UI	–	User Interface
JPG	–	Joint Picture Group
PNG	–	Portable Network Graphics
RGB	–	Red Green Blue

ABSTRACT

In this project created for learning 2D Images as 3D Models or Media using Augmented Reality technology. Learn with 2D images are complex to understand all structures. We employed to solve this issue using an Android Application with basic learning functionalities and features called as **AR Learn**.

The main purpose to made the new learning experience among students and creating positive impact between teacher and students. In this case we are providing a main menu with some basic details. User recognize the 2D Image target using Mobile Enabled Camera through created Android Application. They can also see the build in 3D models for respective 2D images through user interface in scene perception and also User can use his device camera to scan the QR and Play the Embedded videos on top of image targets and user selection option provided next to home user interface.

We are using UNITY 3D as Developing Environment. It more effective and flexible between Platforms. In this project Vuforia Engine used as Image Target Database for detecting 2D images or QR code and place the 3D models or Media in corresponding Position. C# is the Programming Language we are using to generate an actual Module.

Our intend to provide AR Learn Android Application that having build in 3D Models when displayed actual 2D image targets are recognized. The process of user captures the actual 2D Image targets or QR code and see the generated 3D models or Media on screen with some basic features. We are also employed Multi target features with some limitations.

Keywords: AR Learn, Complex Structure, Built in 3D Models, Actual 2D images, Scenes

CHAPTER 1

INTRODUCTION

In this chapter we are briefly discuss about Augmented Reality technology and project Aim, Overview, Objective and key features.

Augmented Reality is the well-known technology with more scope in future for talented consumers. It has the digital elements or three dimensional objects in real time or real world. We can able to visualize the augmented reality models or objects using any electronic devices that must be camera enabled. It will be classified into four different types Marker based, Marker less, Projection based and Superimposition based augmented reality. In this project we are using Marker based Augmented Reality technology. Compare than other types marker based technique is more effective and efficient.

1.1 Project Aim and Objective

One of the major problem in the classroom or when studying living and non-living entities structures, It’s hard to understand the structure and giving less interaction to students. To address this issue providing a Android application that enabled AR technology with following Objectives to satisfies the interaction.

Objectives,

- Providing new Learning Experience among students.
- Create an Environment is Simple to Learn complex Structure.
- Improving Learning Skills through visuals.
- To boosting the Participation and Interaction between students and teachers.
- Creating Positive impact about studies among students.

1.2 Overview

Augmented Reality is the most effective technology to communicate with real world through Digital elements such as 3D models or media. This project also providing the basic interaction between users and running environment.

1.3 Marker-based AR

The most effective and interactive Augmented technology compare than other types in the group of AR family. It scanning the image targets or markers to place the Augmented experience. When marker scanned then process will be triggered automatically or manually it based on system design.

1.4 AR in Education

It was the application of Augmented Reality. Vision is the mission followed by this tagline AR creating more interactive and participation in Education Industries. It transforms traditional way of learning style into new way of experience learning style.

1.5 Vision becomes Mission

Our day to day life teaching us to many different topics and experience. In this case our main intend to let’s do something different with future scope. At the end Vision of human through digital cameras become a Mission named Smart Learning System Using Augmented Reality.

CHAPTER 2

LITERATURE SURVEY

A Survey of Literature else Literature review in this project are analyzing and done research-oriented work for the field of interest with already Published Papers, Journal or Materials. The process which takes place in Initial stage of project development. It providing a perception to the project.

Yogita Bahuguna	Smart learning based on Augmented Reality with android platform	2018
Phum Natakuaithung	Development of AR Learning Assistance of 3D Clay-Sculping	2020
Jon Paddie	Augmented Reality Where we will all Live	2017

Smart learning based on Augmented Reality with android platform

Augmented reality is a fast and an emerging research field of computer science and technology. The word augmented reality came into existence in 1990 by Tom Caudel. Head mounted displays play a vital role in the field of AR. In Augmented Reality technology, the information of real world is incorporated with the virtual word in a manner that the integration of these two shows no indication of disparity and incongruity. If there is discrepancy, the end user may get puzzled. The AR algorithms work on the gathering of data from the real word and then converting it into electronic form for the user. There are numerous AR techniques; some of them make use of conventional input systems. The appropriate AR technique can increase the acuity of human being in various conditions and creating a delusion between virtual and real world.

Drawbacks

In this case, it having only basic 3D models without any component names or labels. A visual experience must need good interaction to user.

Development of AR Learning Assistance of 3D Clay-Sculping

Learning is a method of translating and transferring knowledge and skills. Learning methods and materials are important to facilitate a learning process. There are many techniques that could be used, including documents, slides, videos, music and games. Each technique has its own characteristic and strength.

In this early year, technology has become a part of human life. The mobile device is a technology that is easy to access. Thus, the rationale behind this research is to address a difficulty to learn sculpting 3D model, using Augmented Reality (AR). AR is widely known and has been used in many fields. By this method, the learner will only require a smartphone or a device that has a camera, so that the full render 3D model can appear on the screen. AR can also provide the dimensional model, in which users can see the model in every direction. The users can easily learn and understand the shape of a 3D model.

Augmented Reality Where we will all Live

Augmented reality systems take us to the next phase in computer interfaces, and are unlike any interface we may be familiar with from the past. Prior to augmented reality our communications with the computer were via 2D, a flat, interface.

CHAPTER 3

SYSTEM ANALYSIS

In this chapter we are briefly discuss about the entire system of smart learning using augmented reality. It contains system specification and requirements of proposed system.

3.1 Software Requirements and Specification

Smart Learning system is the application-oriented project which helps users to Learn the 2D image targets into 3D objects else a relative clip.

3.1.1 Problem Statement

Every problem having a different solution, In this project following problems are occurred when we are made this projects, with a help of problem solution we are developed a Problem Statement as follows,

- 2D images are not allow complete learning satisfaction.
- Traditional way of Learning style.
- Less interaction among students.

To address these issues, Providing an Android Application that is Enabled with Augmented Reality technology for understanding intricate structure of both living and non-living entities.

3.1.2 System Requirement

A Smart Learning system using augmented reality contains functional and non-functional requirements for better experience. The Process of developing an android application including many built in packages and open source libraries. It has particular hardware and software requirements to deploy the complete user-friendly application based on android platform. Requirements as follow,

Functional Requirements,

- Every 3D models or video having respective 2D image targets.
- User must use the camera enabled and android supported device only.
- System should be able to display information about 3D models.

Non-functional requirements,

- When user access the system it will be more attractive and work fast between interfaces.
- The system must display accurate 3D objects for respective 2D image target.
- It must contain user friendly interface.

3.1.3 Hardware Requirements

RAM – Minimum 4GB RAM and Greater.

PROCESSOR – AMD A6 Pro or Better Processor.

DISK – 20 GB of disk space is required from SSD.

Graphical Interface – OpenGL 3.2 or Vulkan Capable or Nvidia and AMD.

3.1.4 Software Requirements

The following is the list of development kits and software components required to build fully functional android application that is embedded with Augmented Reality techniques,

Operating system – Windows (8, 10, 11), Linux Distros, Mac OS.

Development Environment – Unity 3D Engine 2021.3.22f1 or any latest version.

Image Target Database – Vuforia Engine 10.14 and base Vuforia SDK package.

Development kits – Java Development Kit (JDK), Native Development kit (NDK) and Android SDK tools.

3D Model Build Platform – Microsoft Paint 3D.

Packages – Unity Built-in AR libraries and Android Build Support

Code Editor – Visual Studio Code or any Platform.

The Unity is the game development engine more effective to create a 3D models and image targets-based media. Vuforia libraries are the key factor to development. It has image targets to detect and recognition of actual 2D images. Android SDK is used to deploy the Android package (APK). A Microsoft Paint 3D used for creating a 3D objects and it contains more flexible 3D object Libraries.

3.1.4.1 Unity Engine

The Unity engine launched in 2005, aiming to "democratize" the development by making it accessible to more developers. Unity Technologies used its game engine to transition into other industries using the real-time 3D platform, including film and automotive. Automakers use Unity's technology to create full-scale models of new vehicles in virtual reality, build virtual assembly lines, and train workers. Unity is a cross-platform engine. The Unity editor is supported on Windows, macOS, and the Linux platform, while the engine itself currently supports building games for more than 19 different platforms, including mobile, desktop, consoles, and Augmented Reality.

3.1.4.2 Vuforia Database

Vuforia Engine is a software development kit (SDK) for creating Augmented Reality apps. With the SDK, you add advanced computer vision functionality to your application, allowing it to recognize images, objects, and spaces with intuitive options to configure your app to interact with the real world. Vuforia Engine supports AR app development for Android, iOS, Lumin, and UWP devices.

3.1.4.3 Java Development Kit (JDK)

The Java Development Kit (**JDK**) is a distribution of Java Technology by Oracle Corporation. It implements the Java Language Specification (**JLS**) and the Java Virtual Machine Specification (**JVMS**) and provides the Standard Edition (**SE**) of the Java Application Programming Interface (**API**). It provides software for working with Java applications. Examples of included software are the virtual machine, a compiler, performance monitoring tools, a debugger, and other utilities.

3.1.4.4 Android SDK

The **Android SDK** is a software development kit that includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows 7 or later.

Android applications are packaged in android package (.apk) format and stored under / data / app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files.

3.1.4.5 Paint 3D

Paint 3D is a raster graphics and 3D computer graphics application. It is one of several 3D modelling and printing applications introduced or improved with the Windows 10 Creators Update, including View 3D, Windows Mixed Reality, Holograms, along with the CAD programs 3D Builder and 2D Builder.

3.2 Existing Model

In the early days' the traditional and conventional teaching method are not fully provides benefits among students. Because of understanding complex structures through 2D images is more difficult.

3.2.1 Drawbacks of exiting model

The major drawbacks of this exiting model are contain basic 3D objects without names or parts of particular element.

- Contains basic 3D objects.
- Less interaction.
- Not comfortable user interfaces.

3.3 Proposed Model

To solve the drawbacks mentioned in existing model using a Smart Learning system using augmented reality model is proposed. In this project, we are introducing Augmented Reality in familiar and one of the leading platform Android. The traditional way of teaching style on blackboard with deeper details about respective parts or elements are not most interactive between students. The teachers and students learn more interactive and much smarter than existing way of learning environment. It provides completely different environment to Learn. In the case of deployed system having smartphones or camera enabled devices that supports android platform applications to detect the actual 2D images of hard or soft copy. The Proposed model contains following features,

- Providing a 3D models for respective 2D image target with name parts or particular elements.
- The Model contains 2D planes embedded with image target for displaying a media clip.

The Image targets are generating 3D models or media based on quality of detecting medium or image target copies. User Interface having two options for QR recognition and normal 2D image target Recognition. Splash screen or main menu contains Initial start option to begin. It two categories are similar in process phase.

3.4 Tools Used

The entire project divided into two main parts such as Front end and back end. It having different components and scripting in between user and interface.

3.4.1 Front end

In this Project, Unity UI components are used to create an actual interface or font end. The widgets like Images, Button textmeshpro and text are used to develop an attractive interface in user friendly perception. Unity Engine's Scene window is the key feature for front development.

3.4.1.1 Unity User Interface Widgets

Unity UI is a UI toolkit for developing user interfaces for games and applications. It is a GameObject-based UI system that uses Components and the Game View to arrange, position, and style user interfaces. You cannot use Unity UI to create or change user interfaces in the Unity Editor. UI elements in Unity are not placed directly onto the scene. They are always placed as children of a special GameObject called the **Canvas**. The Canvas is like a "drawing sheet" for UI on the scene, where all UI elements will render. Creating a UI element from the **Create** context menu without an existing Canvas will automatically generate one. Following are the game objects used for creating user interface in this project,

- Canvas
- Panel
- Button – TextMeshPro
- TextMeshPro

- AR Camera
- Image Target

AR Camera – AR Camera access is required for all AR applications. Using and controlling the camera lets you adjust performance, focus modes, and for advanced use cases, the viewport of the camera.

3.4.2 Back end

In this case we are used C# scripting language to provide an interaction from user to application such as Event handling and Load scene. Vuforia Target Database are used for storing and detecting image target. Target Manager contains required image targets for displaying 3D objects.

3.4.2.1 Vuforia Image Target Database

Vuforia Engine is the most widely used platform for AR development, with support for leading phones, tablets, and eyewear. Developers can easily add advanced computer vision functionality to Android, iOS, and UWP apps, to create AR experiences that realistically interact with objects and the environment. Image Targets represent images that Vuforia Engine can detect and track. The Engine detects and tracks the image by comparing extracted natural features from the camera image against a known target resource database. Once the Image Target is detected, Vuforia Engine will track the image and augment your content seamlessly using best in market image tracking technology. Image Targets can be created with the Vuforia Target Manager using JPG or PNG images in RGB or grayscale. The size of the input images must be 2.25 MB or less and have a minimum width of 320 pixels.

Features extracted from these images are stored in a cloud or device database, of which the latter can be downloaded and packaged together with the application. The database can then be used by Vuforia Engine for runtime comparisons.

CHAPTER 4

SYSTEM DESIGN

It is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of the system. flow diagram is used in an interchangeable way in the meaning of a representation of a process.

4.1 Flow Diagram

Flow diagram is a collective term for a diagram representing a flow or set of dynamic relationships in a system. the following diagram representing this project with defined elements,

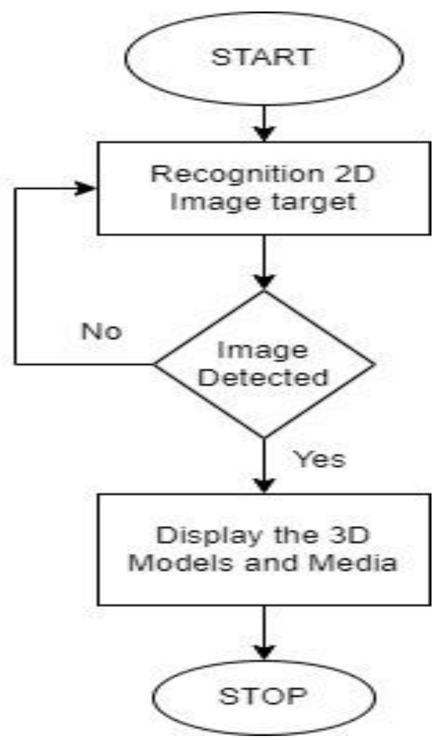

Fig 4.1: Flow chart

The system having a simple flow diagram with three different phases such as Recognition, Detection and Display. In this structure Image target has been detected through camera and Image recognition happens at back end using Vuforia. when target recognized corresponding 3D model displayed on top of image target.

4.2 Block Diagram

A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. In process control, block diagrams are a visual language for describing actions in a complex system in which blocks are black boxes that represent mathematical or logical operations that occur in sequence from left to right and top to bottom.

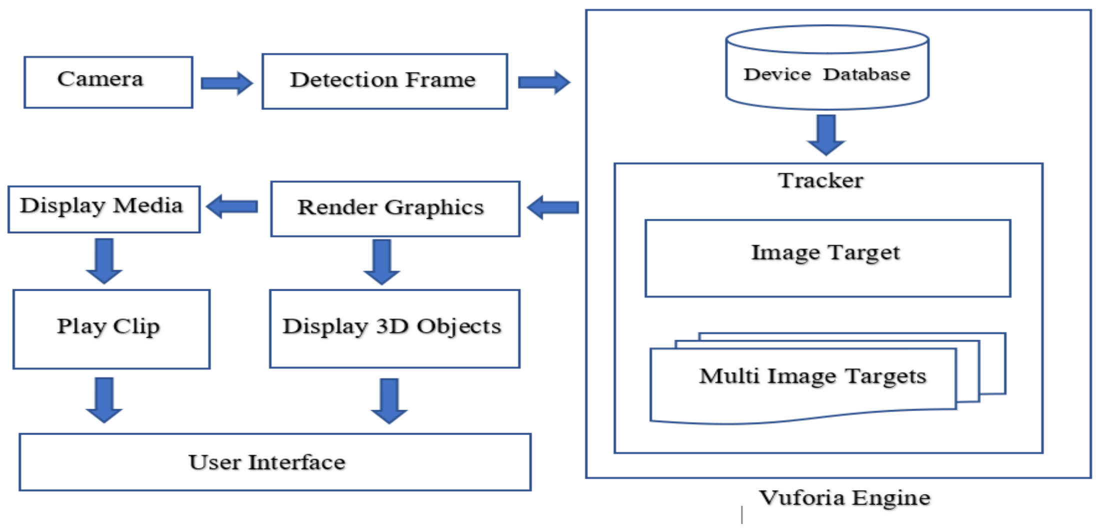

Fig 4.2: Block Diagram

4.3 Stage Chart Diagram

The Stage Chart diagram representing the stages are present in the entire process for getting actual result or output with corresponding inputs based on user requirements. The Stage chart having different stages and corresponding steps in each stage. Final or last stage in the chart has been used for getting a result of actual application or product.

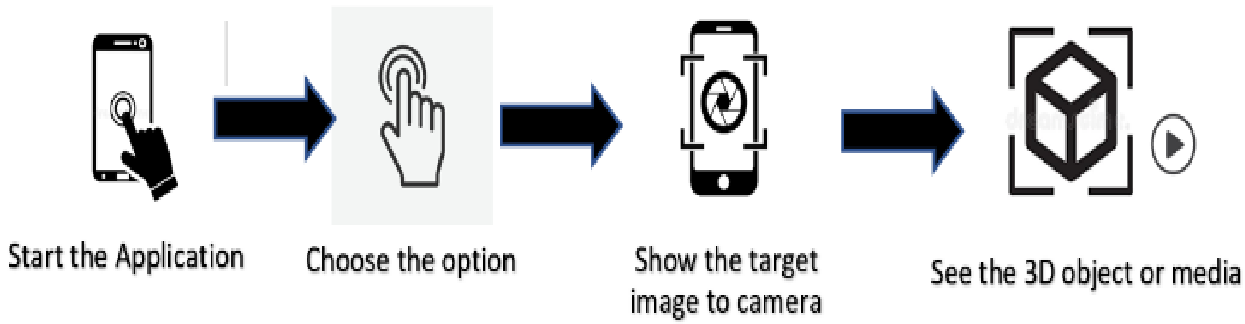

Fig 4.3: Stage Chart

In this stage chart, there are four different stages are mentioned for getting actual result of this project. The Process can be automated using unity user interface with user inputs.

4.4 Use Case Diagram

It is the primary form of system or software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation

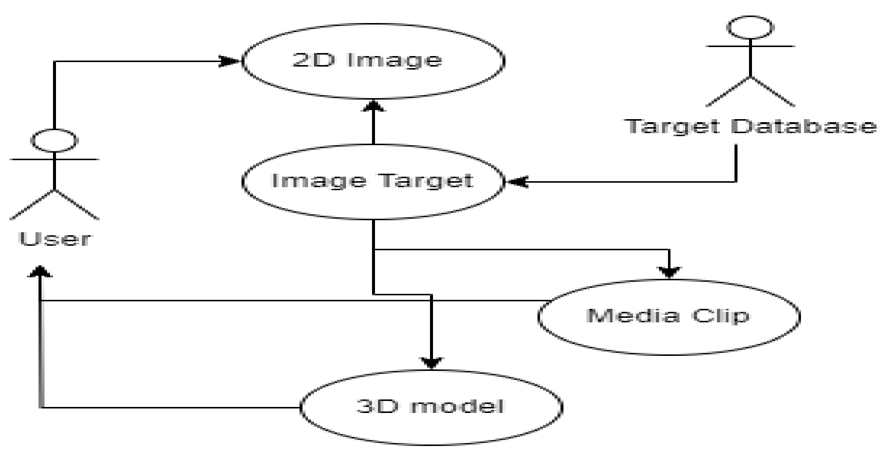

Fig 4.4: Use case Diagram

4.5 Development Architecture

It is a detailed step-by-step process for developing or changing an enterprise architecture and it classified into different phases of this project.

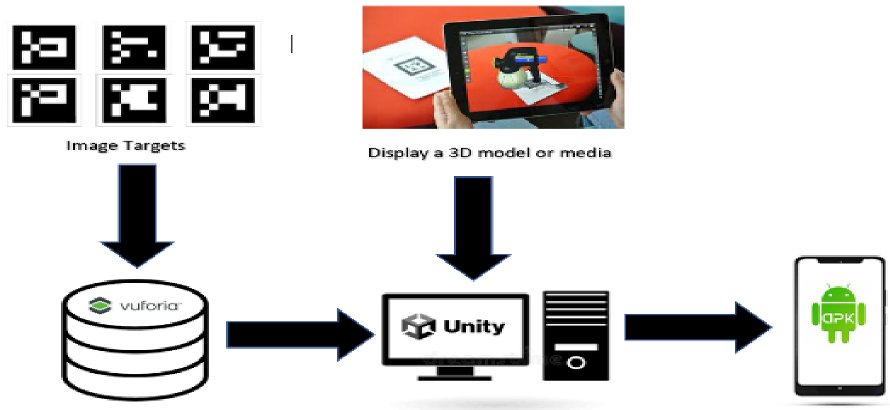

Fig 4.5: Development Structure

The Development phase consists following three major stages to deploy the fully functional android application. the stages are,

- Creating a User interface (main menu) to interact with users.
- Generating Image targets using Vuforia SDK and Image target database.

Collecting 3D model dataset and also media dataset, with the help of Unity engine concatenating respective image target and 3D models or media. Following are the components of developing structure,

- Unity Engine
- Vuforia Database
- Image Targets
- 3D Models

In the case of developing system must consider the camera resolution and Image length to detect in meters. These factors are key to develop error free user interface to generate 3D model or media.

CHAPTER 5

IMPLEMENTATION

In the project an implementation is a realization of a technical specification algorithm such as a program, software component, other computer system through computer programming and deployment. Many implementations may exist for a given specification or standard.

5.1 Installation

The process of installing developing environment with necessary package for development for implementing an actual design.

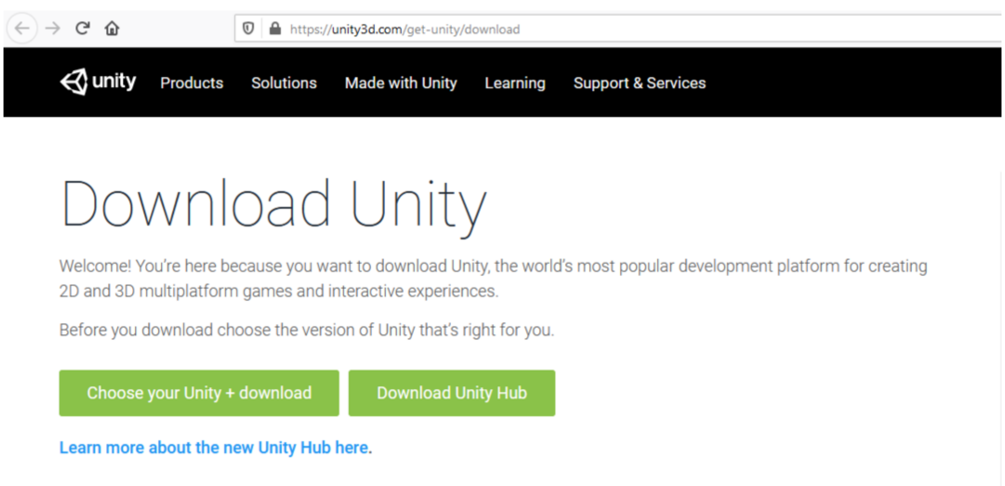

Fig 5.1: Download Unity Hub

The first stage in implementation is download the unity hub from official unity website with creation of specific login account for move further in work place. It is the standalone application for accessing the unity ecosystem. This is used for things like managing your unity projects, installing unity editor based on version, licensing and install add-on components.

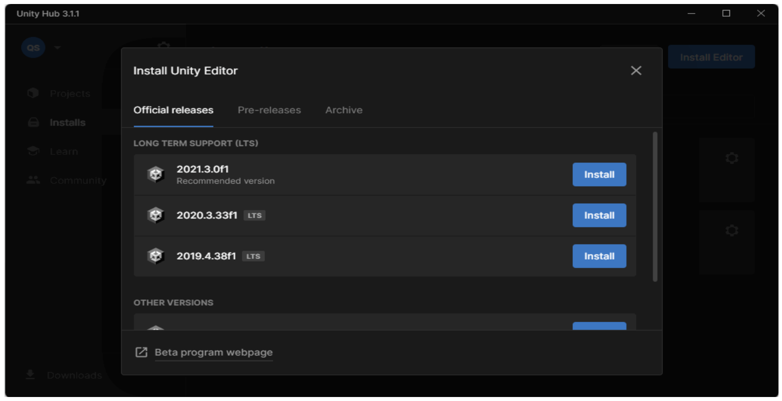

Fig 5.2: Download Unity Editor

Next stage in installation download and install the Unity Editor or Work Place for development and also download required SDK through Unity Hub.

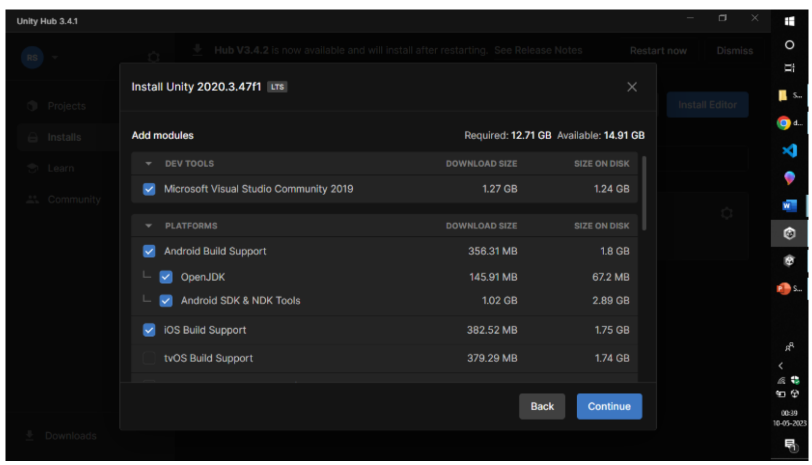

Fig 5.3: Install require tools

In the installing window check required tools such as java development kit, Android development kit and native development kit. It add path to unity editor without any path issue automatically using unity hub, If you want to build and application for ios, let check the IOS Build and install the package. For setting a unity editor to android development, change the platform from windows to android using “Switch Platform” in Build Setting menu.

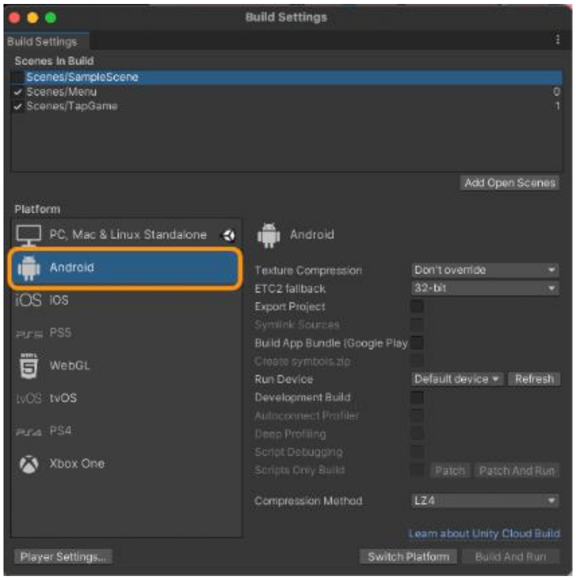

Fig 5.4: Switch Platform to Android

5.2 User Interface

A User Interface (UI) is the space where interactions between humans and machines occur. The goal of this interaction is to allow effective operation and control of the machine from the human end, while the machine simultaneously feeds back information that aids the operators decision-making process. Generally, the goal of user interface design is to produce a user interface that makes it easy.

5.2.1 Splash Screen

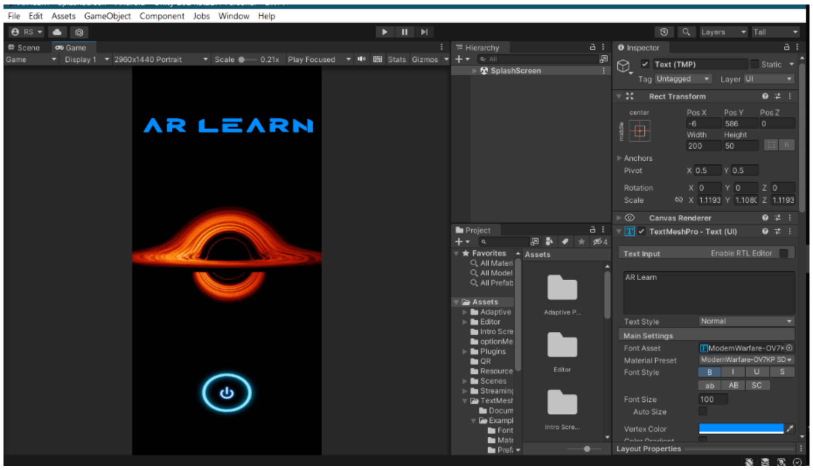

Fig 5.5: Splash screen

5.2.2 Option Menu

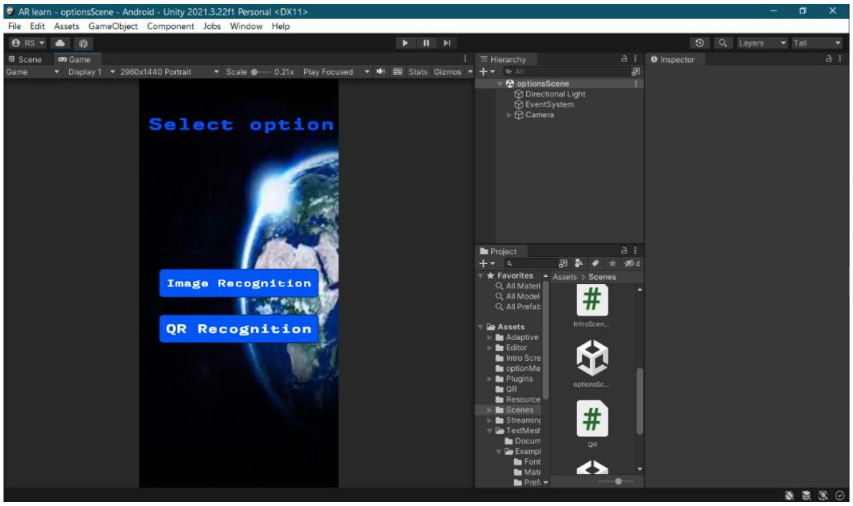

Fig 5.6: option menu

5.3 Image Target Database

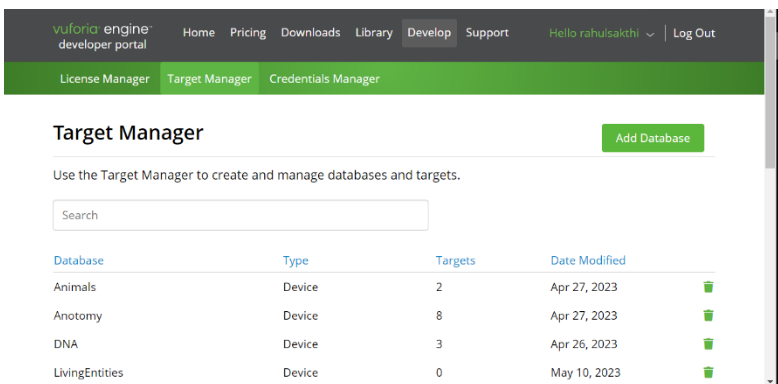

Fig 5.7: Target Database

5.4 3D model generation

Three-dimensional (3D) models represent a physical body using a collection of points in 3D space, connected by various geometric entities such as triangles, lines, curved surfaces, etc. The Microsoft Paint 3D is used for creating 3D objects with name of parts on 3D objects in three-dimension perception for understanding purpose.

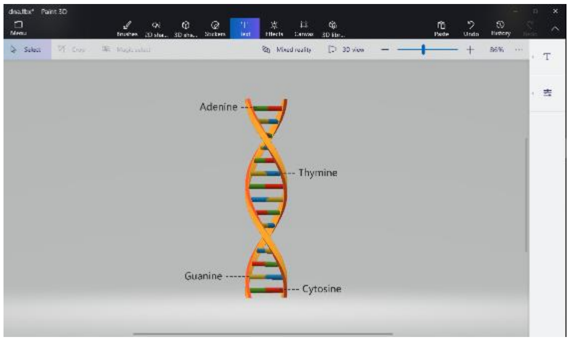

Fig 5.8: DNA 3D model

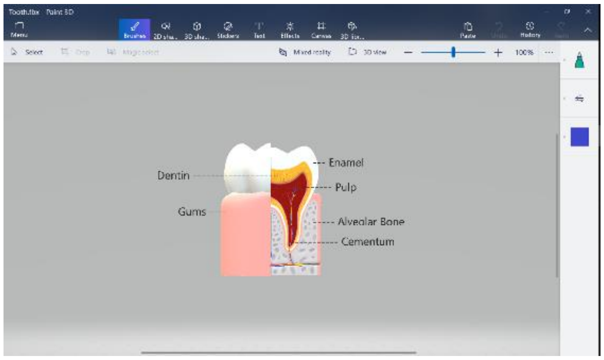

Fig 5.9: Human Tooth 3D model

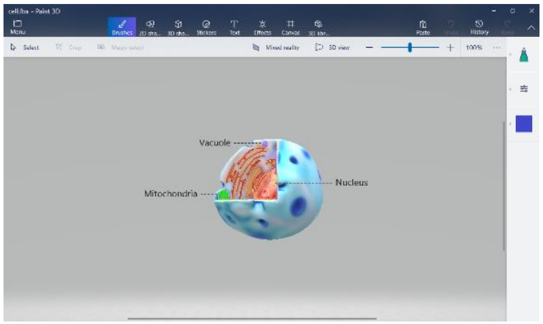

Fig 5.10: Human cell 3D model

Followed by above figures, 3D models are implemented with parts name for both living and non-living entities. Each 3D models having different image targets. The targets are 2D images or any plane with uploaded in image target database. 3D models are surfaced to respective image targets in Unity work place in following stages.

5.5 Implementation on work place

In this stages image target database added to unity work place and respective 3D models placed on top of 2D images with Enable Licenced AR Camera on device work place. Multiple Image targets are stored in database individually and Single corresponding 3D model displayed when multiple similar features image recognized through camera.

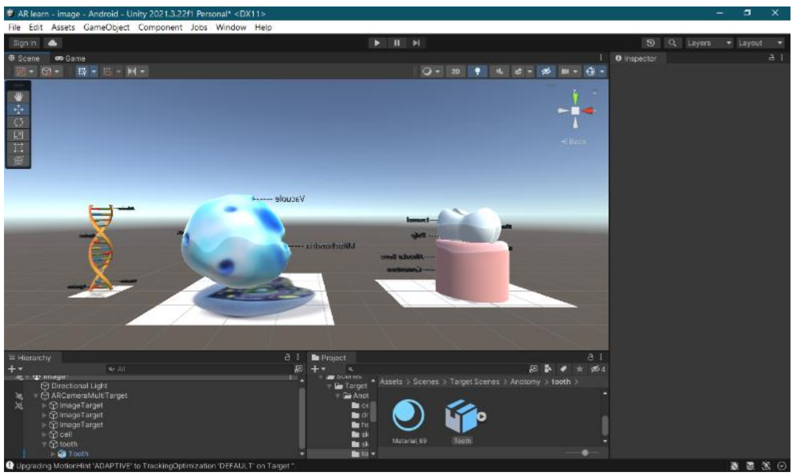

Fig 5.11: Workplace Implementation 1

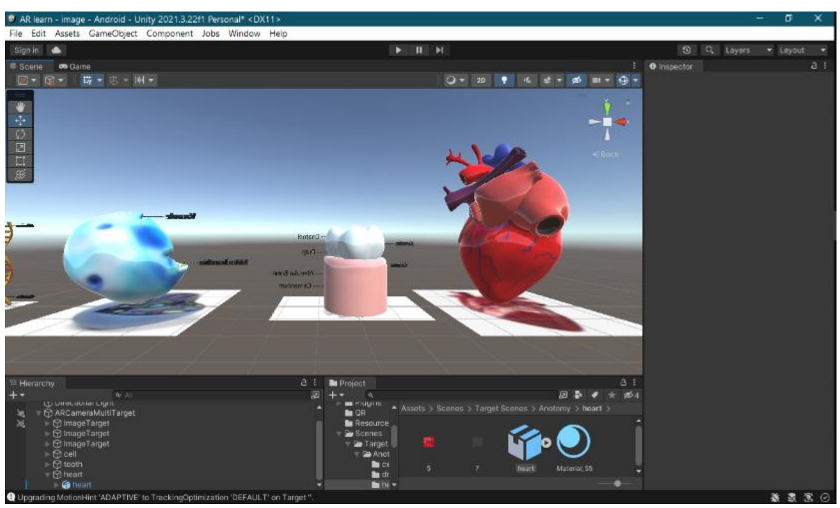

Fig 5.12: Workplace Implementation 2

CHAPTER 6

SYSTEM TESTING

Testing is a set of activity that can be planned and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system.

Testing Objectives:

There are several rules that can serve as testing objectives that are,

- Testing is a process of executing a program with the intent of finding an error
- A good test case is one that has high probability of finding an undiscovered error.
- A successful test is one that uncovers an undiscover error.

If testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software or application. Also testing demonstrate that software functions appear to the working specifications.

There are three ways to test a program

- Correctness testing
- Implementing efficiency
- Computational complexity

Tests for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear for large programs.

Testing for implementation efficiency attempt to find ways to make a correct program faster or use less storage.

Testing for computational complexity amount to an experimental analysis of the complexity of an algorithm.

- Testing Correctness
    1. Prevention Measures
    2. Spot Checks
    3. Testing all part of program
    4. Test Data
    5. Looking Error
    6. Testing time
    7. Re-testing

The entire testing process can be divided into three type of phases,

- Unit Testing
- Integration Testing
- System Testing

6.1 Unit Testing

As this system was GUI based application, the following were tested in this phase

1. Button Click order
2. Requirement analysis
3. Front end validation

In this system Unit testing has been successfully handled. The test data was given to each and every module in all respects and got the desired output. Each module has been tested found working properly.

6.2 Integration Testing

Test data should be prepared carefully it only determines the efficiency and accuracy of the system. Every data validated with input data.

6.2.1 Top-down integration testing

It starts with the highest-level modules and integrates them with lower-level modules. It helps to identify and resolve issues that may arise when different units of the software are combined.

6.2.2 Bottom-up integration testing

It starts with the lowest-level modules and integrates them with higher-level modules. It helps to improve the overall reliability and stability of the software.

6.3 System Testing

In this testing case tester evaluate the whole system with specified requirement. In this project we are used following types of testing,

6.3.1 End to End Testing

In this testing method testing involves complete application environment.

6.3.2 Happy path testing

The testing is to test an application successfully on a positive flow.

6.3.3 Performance Testing

It is designed to test the run-time performance of software within the context of an integrated system. It is used to test the speed and effectiveness of the program. It is also called load testing. In it we check, what is the performance of the system in the given load.

CHAPTER 7

RESULTS

7.1 Single Image Recognition

Fig 7.1: Display Lion and heart 3D model

7.2 Multi Image Targets

Fig 7.2: Different image target for single 3D model

7.3 QR Recognition

Fig 7.3: Play Clip on QR image target

CHAPTER 8

CONCLUSION & FUTURE IMPLEMENTATION

8.1 Conclusion

At the end Augmented Reality through smartphones and camera enabled devices are more helpful for students and teacher to learn every elements and parts in depth. Furthermore, It most effective among students because of less requirements. It needs basic smartphone with better camera quality to visualizing the AR experience.

Generated 3D models are creating good impact and new way of learning experience to make future ready talent. Every lesson is bored when studying with two dimensional images, this project gives you to different immersive experience.

8.2 Future Implementation

In Future, we are combining the machine learning models with Augmented Reality applications for more effective way of learning and creating fully functional Virtual classrooms for complete interaction between students and teacher.

It having more interaction and real-time environment. It helpful to students and teacher to learn more exciting things and also gain more knowledge about other fields compare than older way of learning perception. It also resolves current disadvantages in this project.

CHAPTER 9

REFERENCE

[1] Aashish Verma, Smart learning based on augmented reality with android platform and its applicability, *Dept ECE, Tula's Institute, Dehradun. Published in 2018 on IEEE.*

[2] Phum Natakuaithung, Development of AR Learning Assistance Tool for Clay-Sculping D Model, *Dept of IT, Mahanakorn University of technology, Bankkok, Published in 2020 on IEEE*

[3] ‘’ Augmented Reality Where we will all Live ” *by Jon paddie.*

[4] Naval Kshirsagar, Educative Reality - Augmented Reality Application for Education, Electronics and Telecommunications Symbiosis Institute of Technology, Symbiosis International, (Deemed University), Pune, India. Published in 2023 on IEEE.

[5] Preetham N, Enhancing Education Experience using Augmented Reality Information Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, India. Published 2020 on IEEE.

[6] Riya Aggarwal, Augmented Reality and its effect on our life, ASET, Amity University Uttar Pradesh, Noida, India. Published in 2019 on IEEE.

[7] Mohammad Fahim Hossain, Augmented Reality for Education; AR Children's Book Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh. Published in 2019 on IEEE.

Appendices

Introscreen.cs

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

public class IntroSceneChange : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {


    }


    // Update is called once per frame
    void Update()
    {


    }
    public void calloption()
    {
        SceneManager.LoadScene("optionsScene");
    }
}
```

imagerecgonition.cs

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

public class IntroSceneChange : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
    public void calloption()
    {
        SceneManager.LoadScene("optionsScene");
    }
}
```

QRrecgonition.cs

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

public class QR : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
    public void QRScene()
    {
        SceneManager.LoadScene("QR");
    }
}
```

TrackableEventHandler.cs

```
using System;
using UnityEngine;
using UnityEngine.Events;
using Vuforia;
using UnityEngine.Events;

/// <summary>
/// A custom handler that implements the ITrackableEventHandler interface.
///
/// Changes made to this file could be overwritten when upgrading the Vuforia version.
/// When implementing custom event handler behavior, consider inheriting from this class instead.
/// </summary>
public class TrackableEventHandler : MonoBehaviour
{
   public enum TrackingStatusFilter
   {
      Tracked,
      Tracked_ExtendedTracked,
      Tracked_ExtendedTracked_Limited
   }

   /// <summary>
   /// A filter that can be set to either:
   /// - Only consider a target if it's in view (TRACKED)
   /// - Also consider the target if's outside of the view, but the environment is tracked (EXTENDED_TRACKED)
   /// - Even consider the target if tracking is in LIMITED mode, e.g. the environment is just 3dof tracked.
```

```
/// </summary>
public TrackingStatusFilter StatusFilter =
TrackingStatusFilter.Tracked_ExtendedTracked_Limited;

public bool UsePoseSmoothing = false;
public AnimationCurve AnimationCurve = AnimationCurve.Linear(0, 0,
LERP_DURATION, 1);

public UnityEvent OnTargetFound;
public UnityEvent OnTargetLost;

public UnityEvent onTrack;
public UnityEvent onLost;
protected ObserverBehaviour mObserverBehaviour;
protected TargetStatus mPreviousTargetStatus = TargetStatus.NotObserved;
protected bool mCallbackReceivedOnce;

const float LERP_DURATION = 0.3f;

PoseSmoother mPoseSmoother;

protected virtual void Start()
{
    mObserverBehaviour = GetComponent<ObserverBehaviour>();

    if (mObserverBehaviour)
    {
        mObserverBehaviour.OnTargetStatusChanged += OnObserverStatusChanged;
        mObserverBehaviour.OnBehaviourDestroyed += OnObserverDestroyed;
```

```
                    OnObserverStatusChanged(mObserverBehaviour,
mObserverBehaviour.TargetStatus);
                    SetupPoseSmoothing();
                }
            }

            protected virtual void OnDestroy()
            {
                if(VuforiaBehaviour.Instance != null)
                    VuforiaBehaviour.Instance.World.OnStateUpdated -= OnStateUpdated;

                if (mObserverBehaviour)
                    OnObserverDestroyed(mObserverBehaviour);

                mPoseSmoother?.Dispose();
            }

            void OnObserverDestroyed(ObserverBehaviour observer)
            {
                mObserverBehaviour.OnTargetStatusChanged -= OnObserverStatusChanged;
                mObserverBehaviour.OnBehaviourDestroyed -= OnObserverDestroyed;
                mObserverBehaviour = null;
            }

            void OnObserverStatusChanged(ObserverBehaviour behaviour, TargetStatus
targetStatus)
            {
                var name = mObserverBehaviour.TargetName;
                if (mObserverBehaviour is VuMarkBehaviour vuMarkBehaviour &&
vuMarkBehaviour.InstanceId != null)
```

```
            {
                name += " (" + vuMarkBehaviour.InstanceId + ")";
            }

            Debug.Log($"Target status: {name} {targetStatus.Status} --
{targetStatus.StatusInfo}");

            HandleTargetStatusChanged(mPreviousTargetStatus.Status, targetStatus.Status);
            HandleTargetStatusInfoChanged(targetStatus.StatusInfo);

            mPreviousTargetStatus = targetStatus;
        }

        protected virtual void HandleTargetStatusChanged(Status previousStatus, Status
newStatus)
        {
            var shouldBeRendererBefore = ShouldBeRendered(previousStatus);
            var shouldBeRendererNow = ShouldBeRendered(newStatus);
            if (shouldBeRendererBefore != shouldBeRendererNow)
            {
                if (shouldBeRendererNow)
                {
                    onTrack.Invoke();
                    OnTrackingFound();
                }
                else
                {
                    onLost.Invoke();
                    OnTrackingLost();
                }
```

```
            }
            else
            {
                if (!mCallbackReceivedOnce && !shouldBeRendererNow)
                {
                    // This is the first time we are receiving this callback, and the target is not
visible yet.
                    // --> Hide the augmentation.
                    OnTrackingLost();
                }
            }

            mCallbackReceivedOnce = true;
        }

        protected virtual void HandleTargetStatusInfoChanged(StatusInfo newStatusInfo)
        {
            if (newStatusInfo == StatusInfo.WRONG_SCALE)
            {
                Debug.LogErrorFormat("The target {0} appears to be scaled incorrectly. " +
                                     "This might result in tracking issues. " +
                                     "Please make sure that the target size corresponds to the size of the
" +
                                     "physical object in meters and regenerate the target or set the
correct " +
                                     "size in the target's inspector.", mObserverBehaviour.TargetName);
            }
        }

        protected bool ShouldBeRendered(Status status)
```

```
        {
            if (status == Status.TRACKED)
            {
                // always render the augmentation when status is TRACKED, regardless of
filter
                return true;
            }

            if (StatusFilter == TrackingStatusFilter.Tracked_ExtendedTracked && status ==
Status.EXTENDED_TRACKED)
            {
                // also return true if the target is extended tracked
                return true;
            }

            if (StatusFilter == TrackingStatusFilter.Tracked_ExtendedTracked_Limited &&
                (status == Status.EXTENDED_TRACKED || status == Status.LIMITED))
            {
                // in this mode, render the augmentation even if the target's tracking status is
LIMITED.
                // this is mainly recommended for Anchors.
                return true;
            }

            return false;
        }

        protected virtual void OnTrackingFound()
        {
            if (mObserverBehaviour)
```

```
        {
            var rendererComponents =
mObserverBehaviour.GetComponentsInChildren<Renderer>(true);
            var colliderComponents =
mObserverBehaviour.GetComponentsInChildren<Collider>(true);
            var canvasComponents =
mObserverBehaviour.GetComponentsInChildren<Canvas>(true);

            // Enable rendering:
            foreach (var component in rendererComponents)
                component.enabled = true;

            // Enable colliders:
            foreach (var component in colliderComponents)
                component.enabled = true;

            // Enable canvas':
            foreach (var component in canvasComponents)
                component.enabled = true;
        }

        OnTargetFound?.Invoke();
    }

    protected virtual void OnTrackingLost()
    {
        if (mObserverBehaviour)
        {
            var rendererComponents =
mObserverBehaviour.GetComponentsInChildren<Renderer>(true);
```

```
            var colliderComponents =
mObserverBehaviour.GetComponentsInChildren<Collider>(true);
            var canvasComponents =
mObserverBehaviour.GetComponentsInChildren<Canvas>(true);

            // Disable rendering:
            foreach (var component in rendererComponents)
                component.enabled = false;

            // Disable colliders:
            foreach (var component in colliderComponents)
                component.enabled = false;

            // Disable canvas':
            foreach (var component in canvasComponents)
                component.enabled = false;
        }

        OnTargetLost?.Invoke();
    }

    protected void SetupPoseSmoothing()
    {
        UsePoseSmoothing &= VuforiaBehaviour.Instance.WorldCenterMode ==
WorldCenterMode.DEVICE; // pose smoothing only works with the DEVICE world center
mode
        mPoseSmoother = new PoseSmoother(mObserverBehaviour, AnimationCurve);

        VuforiaBehaviour.Instance.World.OnStateUpdated += OnStateUpdated;
    }
```

```
void OnStateUpdated()
{
    if (enabled && UsePoseSmoothing)
        mPoseSmoother.Update();
}

class PoseSmoother
{
    const float e = 0.001f;
    const float MIN_ANGLE = 2f;

    PoseLerp mActivePoseLerp;
    Pose mPreviousPose;

    readonly ObserverBehaviour mTarget;
    readonly AnimationCurve mAnimationCurve;

    TargetStatus mPreviousStatus;

    public PoseSmoother(ObserverBehaviour target, AnimationCurve animationCurve)
    {
        mTarget = target;
        mAnimationCurve = animationCurve;
    }

    public void Update()
    {
```

```
        var currentPose = new Pose(mTarget.transform.position,
mTarget.transform.rotation);
        var currentStatus = mTarget.TargetStatus;

        UpdatePoseSmoothing(currentPose, currentStatus);

        mPreviousPose = currentPose;
        mPreviousStatus = currentStatus;
    }

    void UpdatePoseSmoothing(Pose currentPose, TargetStatus currentTargetStatus)
    {
        if (mActivePoseLerp == null && ShouldSmooth(currentPose,
currentTargetStatus))
        {
            mActivePoseLerp = new PoseLerp(mPreviousPose, currentPose,
mAnimationCurve);
        }

        if (mActivePoseLerp != null)
        {
            var pose = mActivePoseLerp.GetSmoothedPosition(Time.deltaTime);
            mTarget.transform.SetPositionAndRotation(pose.position, pose.rotation);

            if (mActivePoseLerp.Complete)
            {
                mActivePoseLerp = null;
            }
        }
    }
```

```
        /// Smooth pose transition if the pose changed and the target is still being reported as "extended tracked" or it has just returned to
        /// "tracked" from previously being "extended tracked"
        bool ShouldSmooth(Pose currentPose, TargetStatus currentTargetStatus)
        {
            return (currentTargetStatus.Status == Status.EXTENDED_TRACKED || (currentTargetStatus.Status == Status.TRACKED && mPreviousStatus.Status == Status.EXTENDED_TRACKED)) &&
                   (Vector3.SqrMagnitude(currentPose.position - mPreviousPose.position) > e || Quaternion.Angle(currentPose.rotation, mPreviousPose.rotation) > MIN_ANGLE);
        }

        public void Dispose()
        {
            mActivePoseLerp = null;
        }
    }

    class PoseLerp
    {
        readonly AnimationCurve mCurve;
        readonly Pose mStartPose;
        readonly Pose mEndPose;
        readonly float mEndTime;

        float mElapsedTime;

        public bool Complete { get; private set; }
```

```
        public PoseLerp(Pose startPose, Pose endPose, AnimationCurve curve)
        {
            mStartPose = startPose;
            mEndPose = endPose;
            mCurve = curve;
            mEndTime = mCurve.keys[mCurve.length - 1].time;
        }

        public Pose GetSmoothedPosition(float deltaTime)
        {
            mElapsedTime += deltaTime;

            if (mElapsedTime >= mEndTime)
            {
                mElapsedTime = 0;
                Complete = true;
                return mEndPose;
            }

            var ratio = mCurve.Evaluate(mElapsedTime);
            var smoothPosition = Vector3.Lerp(mStartPose.position, mEndPose.position,
ratio);
            var smoothRotation = Quaternion.Slerp(mStartPose.rotation,
mEndPose.rotation, ratio);

            return new Pose(smoothPosition, smoothRotation);
        }
    }
}
```