Integrating 'Leap Motion' with 'ARCore' in Android Operating System to Enhance and Enrich Academic Learning Process and Research Progress

by

Fardin Fahad
14101194
Md. Asifullah
14101108
Farah Ibnat
19241030
Asadullah Monsur
17101342

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2020

Declaration

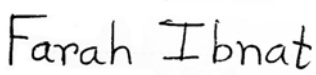
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

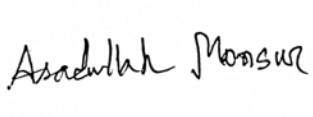
Student's Full Name & Signature:

Fardin Fahad
14101194

Md. Asifullah
14101108



Farah Ibnat
19241030



Asadullah Monsur
17101342

Approval

The thesis/project titled "Integrating 'Leap Motion' with 'ARCore' in Android Operating System to Enhance and Enrich Academic Learning Process and Research Progress" submitted by

1. Fardin Fahad (14101194)
2. Md. Asifullah (14101108)
3. Farah Ibnat (19241030)
4. Asadullah Monsur (17101342)

Of Summer, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 05, 2020.

Examining Committee:
Supervisor:
(Member)

Moin Mostakim
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Name of Program Coordinator
Designation
Department
Brac University

Head of Department:
(Chair)

Name of Head of Department
Designation
Department of Computer Science and Engineering
Brac University

Abstract

With the advancement of science and technology, human lives have become more convenient than the past. As technology is rapidly changing and enhancing day by day, we also need more efficient and enriched approaches especially in academic teaching, research activities and other technological fields as well. Moreover, a better visual representation and advanced technology are must needed pre-requisite for educational progress. Keeping these issues in mind, we have thought about an integrated augmented and mixed real environment where Leap Motion and ARCore of Google will play a significant role for upcoming generations' learning, teaching, different research fields, problem solving, planning etc. that are not only of computer science sectors but also other sectors which can be a paradigm shift for multi-media sectors. However, in our paper, we will focus on android operating system where integration will be based on mobile platform or android operating system which is used enormously now-a-days and is more flexible and economically convenient.

Keywords: Leap Motion; ARCore; Android; Operating System

Acknowledgement

Firstly, all praise to the Almighty Allah, the All-Merciful Who has bestowed all the blessings to complete this thesis succesfully. Secondly, we would like to express our gratitude to our adviser Mr. Moin Mostakim and co-adviser Mr. Apurbo Ghosh for all the support and guidance.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$3D$ 3 Dimensional

API Application Programming Interface

App Android or Desktop or Web Application

AR Augmented Reality

Docs Documentation

IDE Integrated Development Environment

MR Mixed Reality

NDK Native Development Kit

OOP Object Oriented Programming

PC Personal Computer

SDK Software Development Kit

UI User Interface

VR Virtual Reality

$Web - app$ Web Application

Chapter 1

Introduction

Development and patronization of science and technology has given us the taste of VR or AR or MR including the Leap Motion Devices. Augmented reality seems to be one of the biggest role models in the future especially in the multi-media sector. Leap Motion is another approach that has not gotten official recognition for more than a decade. There is a little confusion among people about what differentiates among AR, VR and MR. To eradicate this confusion, we can simply tell about these agendas that virtual reality (VR) connects users where the environment is totally simulated and virtual content spins in the real-world area in augmented reality (AR). Other than these two, mixed reality (MR) mirrors the user interaction jointly in the virtual world alongside the physical world[7]. Additionally, Leap Motion Devices serves as a sensor. Depending on such situations, we have tried to focus on integration of ARCore with Leap Motion.

1.1 Major Motives

At present, the traditional academic teaching method is incomplete without advanced technology. Research and industrial development are also complex, inflexible as well as costly. We lack modern multimedia facilities that suit within our economic capabilities. Even commercial areas are also needed to be more attractive and realistic features for visual representation. For this purpose, 3D contents are gifts. Sensors that free hand motion interacting with the 3D contents improves digestion software simulations.

Our major goals are:

- Using 3d, Enrich academic education system,
- Make interactive 3d contents for Research convenience
- Free Hand Motion Detecting Data improvement
- Reduce hardware costs, replacing PC by Android Smartphone or Tab or similar.

Our achievement will also contribute in:

- Expansion of other high technology-based fields such as gaming.

1.2 Thesis Outline

Upto this (chapter 1) we have summarized our motive of achievement. In chapter 2, we introduced briefly about Leap Motion and ARCore with examples and former accomplishments. Subsequently, our working strategies and workflows have been demonstrated with flowcharts and pictures (in chapter 3). The chapter 4, describes the project requirements to implement the proposed model. The step by step implementation process has been organised including project source code snippets in the chapter 5. Finally, after result analysis in chapter 6, we concluded our whole work.

Chapter 2

Literature Review

2.1 Peek in Leap Motion

Leap Motion is a fascinating and revolutionary invention in modern technology which distinguishes gestures as well as hand motion. The 'Motion Control' industry that manufactures the leap motion controller. The founders of this industry are Michael Buckwald and David Holz and the company started their journey from 2010. Leap motion has been used in various projects and has achieved remarkable support from the developer zone. Now, we have a game like fruit ninja where leap motion is integrated. Through the upbringings of Perdana, I. (2014)[2] & Niechwiej-Szwedo, E., Gonzalez, D., Nouredanesh, M. and Tung, J.(2018)[4], we can understand how leap motion helps in learning and co-curricular activities of children. To learn deeply and get better understanding of functionality of Leap Motion we have looked up a few additional webites and [3] [6] [16] [20] .

2.2 Peek in ARCore

On the other hand, ARCore of Google is a very new technology where it is released in march of 2018. It is based on the android operating system where java is the trade-mark of it. We can have a taste of augmented reality through ARcore. It can track motion, sense the environment and also estimate light using a camera of a mobile phone relating virtual content. Beer Pong AR, Porsche Mission E etc are some of the most well known ARCore applications. Again, Kesi, M. & Ozarslan, Y. (2012)[1] upholded potential play-role of AR in the fields of education. The official documentation and overview of ARCore is available at Google's Developer site[8].

2.3 Proposed Model

Although mixed reality or augmented reality or virtual reality[5] is not a new topic and a good amount of researches have been performed and excellent outcomes have already been achieved to some extent, the success has been took place in individual sectors for example:- either using AR or using Leap Motion or in combined one in different operating systems (such as windows operating system) where the process and environmental set up are both complex and costly to implement. As we have

mentioned earlier that we are about to focus on integrating these separate technologies (ARCore [Android NDK] and Leap Motion) using the android operating system which has not yet been published and is still under research issues. One thing we have to keep in mind is that we are not producing different hardware currently as it is beyond our capacity or maybe in the future, it will be integrated. To be specific we need the help windows or mac or linux operating systems to handle web-app (specified later) that gives us detected data from Leap Controller. Output will be shown in ARCore Android App for now. Hence, it is our goal to achieve.

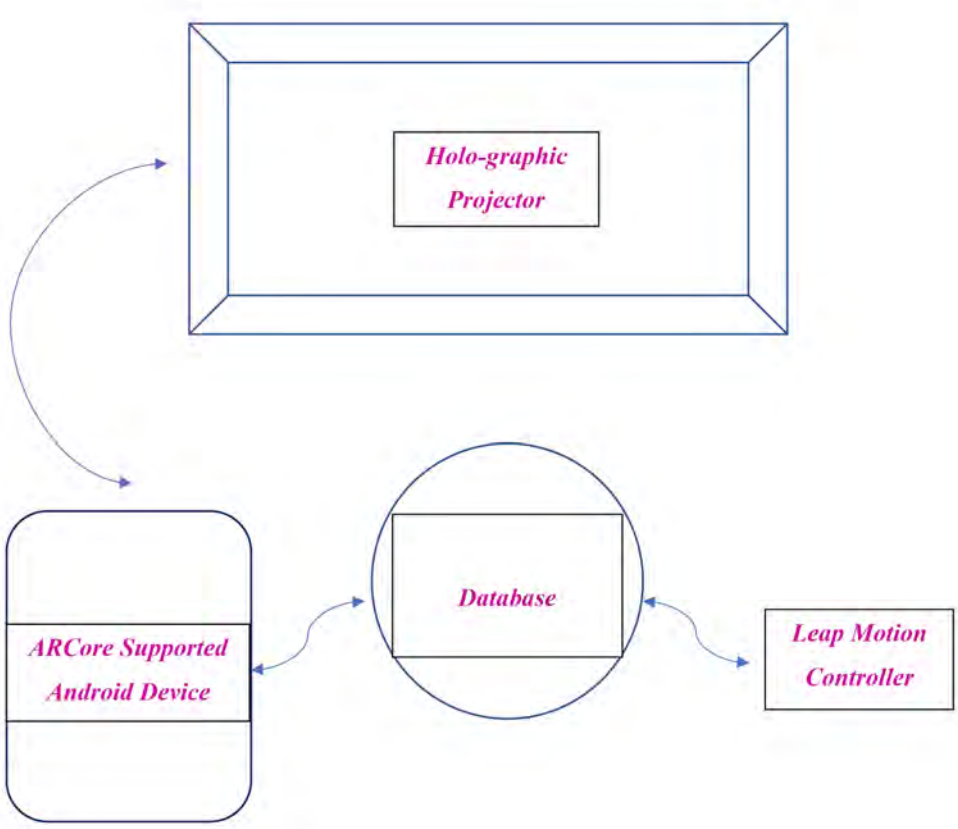


Figure 2.1: Overall Environment

Chapter 3

Procedure and Work-plan

3.1 Methodology

As per requirement, for the implementation of the proposal, we ought to collect data in various ways and modify them as required. We have collected observed, experimental, simulated and derived data that are all interrelated. Hence, we fulfilled both the category of quantitative and qualitative data.

3.2 Workflow

At the very beginning, we will focus on setting up our mixed environment. As we have already mentioned the overall environment model for our proposal, we will move based on that. Now, we have four different steps to integrate:

1. Sense data through Leap Motion Controller
2. Get sensed data via firebase to ARCore (Android) App
3. Configure our ARCore App for 3D Representation
4. Output inside ARCore App and on Holographic Projector or Cup

For the first step, we need a web-app. Because we need an application that takes sensed data to Android App. A web is used Firebase Admin SDK to take data from Leap Motion controller via Firebase database to Android App. Obviously, the most popular database for Android Apps is Firebase. Firebase has functionality of auto update when new data is inserted. So, when Leap Motion Controller sends input Android App (ARCore App) receives the update dynamically.
Definitely, to get free hand motion tracking we need a Leap Motion Controller and integrate it's SDK to the sample web app project and connect the controller to the database too.

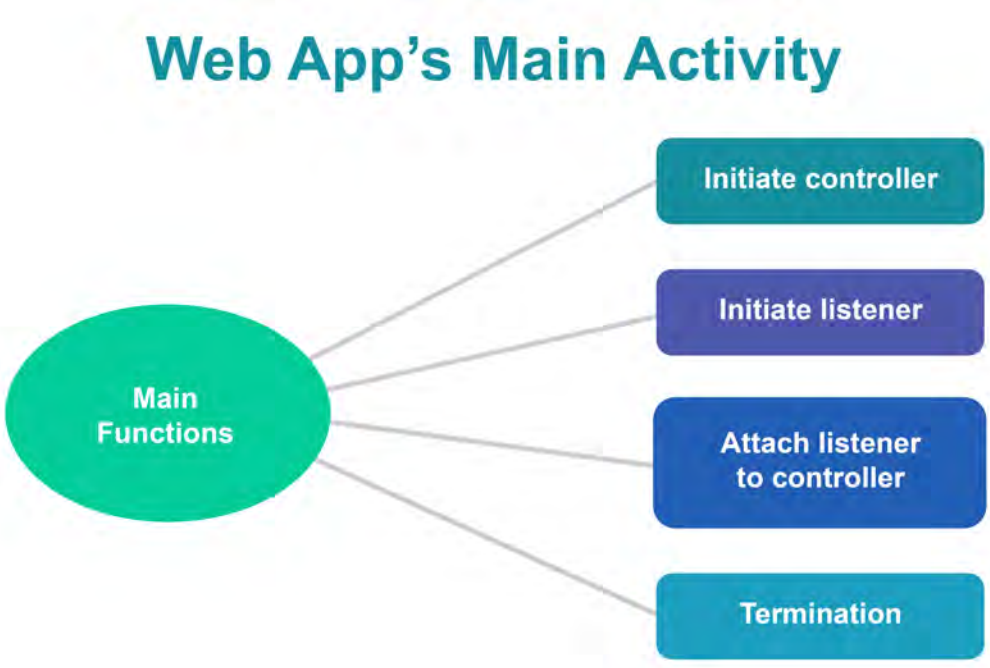


Figure 3.1: Web-App Functionality

In the second phase, we need to view the Leap Motion Controller Functionality. In general, this web-App project can be considered Leap Motion App. So, second phase is titled as Leap Motion:

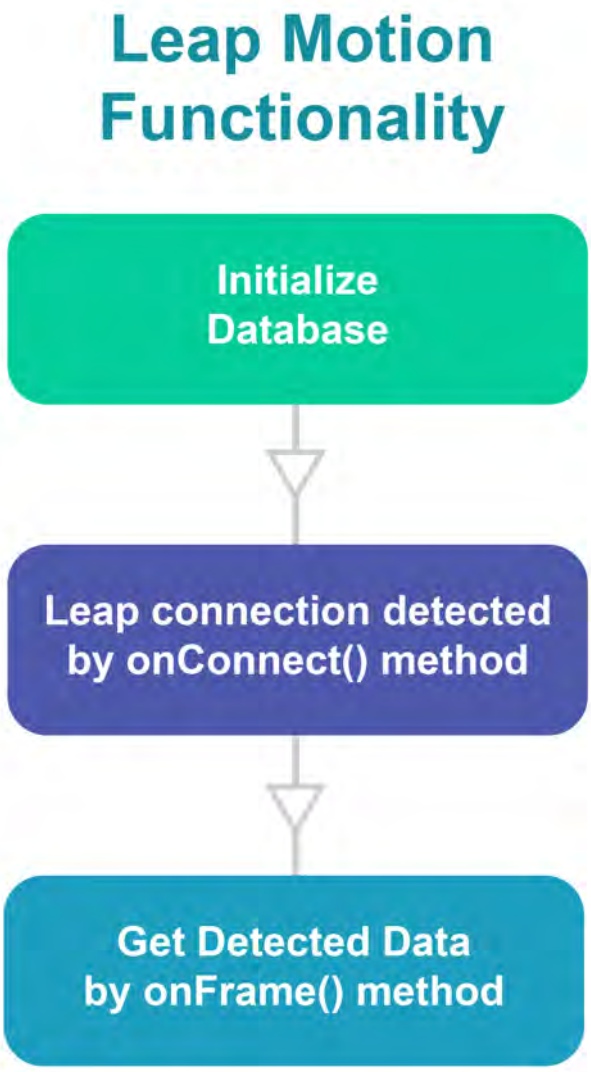


Figure 3.2: Leap Motion Functionality (Brief)

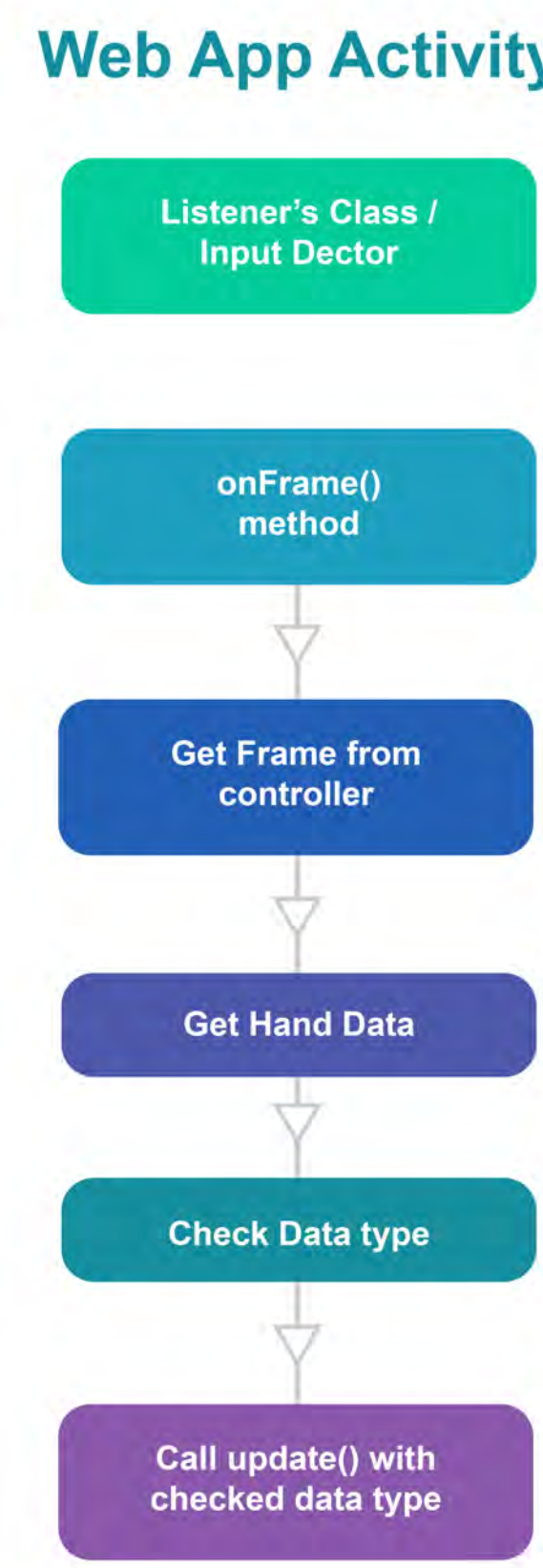


Figure 3.3: Leap Motion Functionality (Detail)

ARCore App's operation (phase 3) is broad. We have described it in the implementation part. However, Android App takes firebase update and pushes the data to 3D Scene:

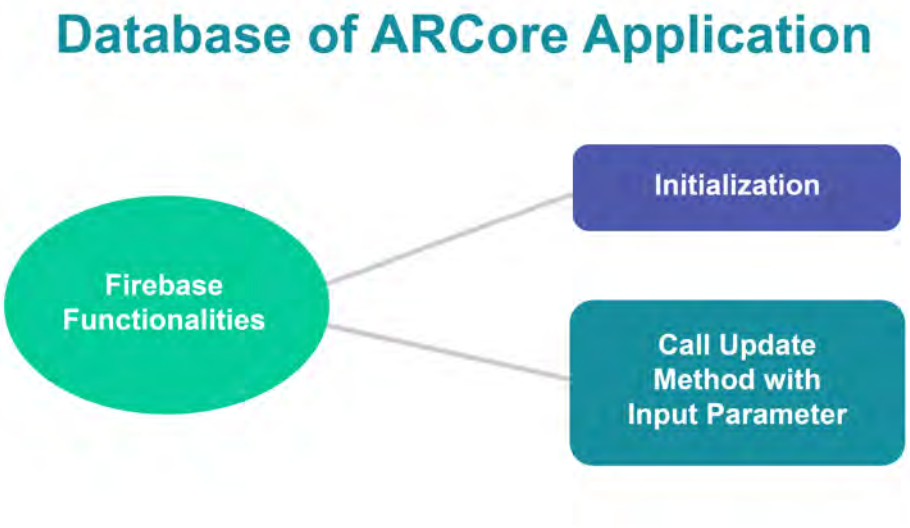


Figure 3.4: Firebase Functionality (in Android)

Figure 3.5: ARCore Functionality

Fourth part will cover 3D view inside Android Application and outside (in Holographic Projector). Therefore, we need to set up an Android Studio IDE connecting the 3D data model with it via Google's ARCore. After that we need to set a database connection with it. Moreover, we need a holographic projector for real-life view through a specialized system which will be placed over a mobile phone. Thus, our accomplishment is on the road to fulfill.
Now, we will see the full project's major functionalities in upcoming diagrams:

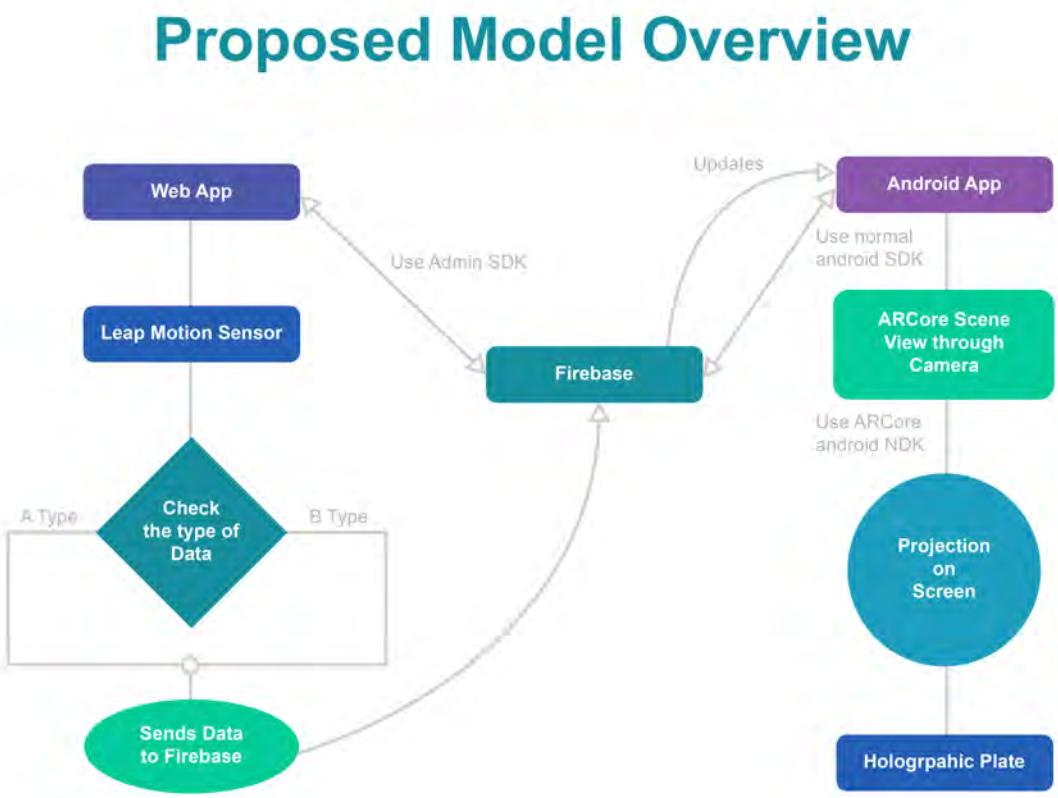


Figure 3.6: Project's Overview

Chapter 4

Tools and Component Collection

4.1 Overview of Major Tools

- 3D data models or dataset
- 3D Holographic Projector or Plate or such specialized instrument
- Google Scene-form SDK and Plugin
- Google's Cloud API
- Google's ARCore Library for AR facility
- Leap Motion Controller for free hand motion tracking
- Database (Firebase)
- Firebase Admin SDK
- Android Operating System ARCore Supported Devices (a mobile or tab)
- Android Studio (IDE)
- Java (OOP language)

4.2 Data Collection

Data collection is used to encourage credible and accurate research and utilize analysis to make further future improvements if necessary. Referring to the collection we collected the software data and tools mostly from official websites and which also includes reputed websites and documentation from Leap Motion, Google and Android. For Leap Motion we have followed documentation of [15]. Again, for ARCore and for Firebase documentation of [17] and[13] respectively are taken to be pursued. Moreover, for our requirement , we took guidelines from Android Docs [12] and Google Cloud APIs[14] and[18].

Apart from documentation, 3D models or object files required for visualization. 3D models are collected from Google's Poly collection[10] and[11].

4.3 Instrumental Collection

Instrument collection processes are both soft-ware and hard-ware based approaches. For hardware, we required an AR supported phone (Redmi Note 7 pro in our case), a Leap Motion Controller (hand motion gesture detector), a Windows PC, Holographic Glass Projector.

In the case of software, we needed Android IDE, Java supported IDE, 3D software. So, We have used Android Studio (IDE), IntelliJ IDEA Community Edition (IDE) Blender (3D soft-ware). All of these are free to use (open plat-form) according to license agreement.

One more thing to make clear that, we need a strong internet connection for our project to implement, test and finally to run after execution.

Chapter 5

Implementation of Proposal

5.1 Web App Initialization

In the beginning of our implementation, we need to download the artifact or demo project structure of web-app. To do it we have Spring Initializer[19] After downloading and opening it in IntelliJ IDE we have to build the project. We also need to include the Firebase library (Admin SDK) to the project's build.gradle file.

The outlook will be like snippet below:

```
dependencies {

  implementation  'com.google.firebase:firebase-admin:7.0.0'

  implementation 'org.springframework.boot:spring-boot-starter-
web'

  testImplementation 'org.springframework.boot:spring-boot-
starter-test'

}
```

Figure 5.1: Web-app's build.gradle

Next diagram shows the structure of main class:

```
import org.springframework.boot.SpringApplication;

import
org.springframework.boot.autoconfigure.SpringBootApplication;


@SpringBootApplication

public class Main {

  public static void main(String[] args) {

    SpringApplication.run(LeapArCoreApplication.class, args);

  }

}
```

Figure 5.2: Main Class of the Web-app

Now, we will proceed to integrate Leap Motion Controller with the database part inside the main method and will be discussed in the next phase.

5.2 Leap Motion Integration

Leap Motion integration simply divided into two part:

1. initialize controller listener:
   InputDetector detector = new InputDetector();
   Controller controller = new Controller();
   Here, InputDetector is a custom listener extending the Listener class of Leap Motion.

2. attach listener to controller:
   controller.addListener(.....);
   // addListener() method takes a single input parameter of InputDetector

Listener class will use the onFrame() method to get every frame detected by sensor. Database will be an inside part of the listener. So, structure of the listener class:

```
public class InputDetector extends Listener {
  private Database db;
  public InputDetector(){
      ........
  }
  public void onConnect(Controller controller) {
      .......
  }

  public void onFrame(Controller controller) {
      .........
   }
 }
}
```

We will get hand selection (left or right hand) from the frame object and call update() method of the database that takes the detected info(1 for left hand and 2 for right hand). Later the update() method of in Database class updates data in the database (ARCore will the update notification and act according to it).

Database part is according to Firebase documentation (Admin SDK).

The whole web-app's class specification used in this project noted by next 3 images:

```
public class DataBase{
  private FileInputStream serviceAccount;
  private  FirebaseOptions;

  public Database(){
     try{

        serviceAccount =  new FileInputStream("./serviceAccount.json");

        options = new FirebaseOptions.Builder()
           .setCredentials(GoogleCredentials.fromStream(serviceAccount))
           .setDatabaseUrl("https://arcore-with-leap-motion.firebaseio.com")
           .build();

        FirebaseApp.initializeApp(options);

     }
     catch(Exception e){
        e.printStackTrace();
        System.out.println("error");
     }
  }

private void update(int input){
   FirebaseDatabase db = FirebaseDatabase.getInstance();
   DatabaseReference dBR = db.getReference();
   DatabaseReference dBCR = dBR.child("input");

  dBCR.child("choice").setValue(input,new DatabaseReference.CompletionListener() {
        @Override
        public void onComplete(DatabaseError error, DatabaseReference ref) {
        System.out.println("Successfully updated");
        }
        });

   }
}
```

```
public class InputDetector extends Listener {

    private Database db;

    public InputDetector(){
          db  =  new  Database();
    }

    public void onConnect(Controller controller) {
        System.out.println("Connected");
    }

    public void onFrame(Controller controller) {
        System.out.println("Frame available");

        Frame frame = controller.frame();
        HandList hands = frame.hands();
        Hand hand = hands.get(0);

        if(hand.isLeft()){
            db.update(1);   //  for x_rotation,   1 is put
        }

        else if(hand.isRight()){
            db.update(2);    //  for y_rotaion,     2  is put
        }

    }
}
```

```
public class Main {

    public static void main(String   []args) {

        InputDetector detector = new InputDetector();
        Controller controller = new Controller();

        controller.addListener(detector);

        System.out.println("To Stop, Press Enter");

        try {
            System.in.read();
        }
        catch (Exception e) {
            e.printStackTrace();
        }

        controller.removeListener(listener);

    }
}
```

5.3 ARCore App Integration

The last part is to configure ARCore application to visualize the output using 3D Content movement according to user input detected by Leap Motion Controller and sent via Firebase.

Firstly, we need to import libraries as well as plugins. To support ARCore scene-view, after going to the android marketplace; we need the Google Sceneform plugin and install it and then restart the IDE and open the project again.

Now, we need to configure both build.gradle (apps gradle and projects gradle). The next 2 figures uphold the apps build.gradle file structure:

Figure 5.3: App's Gradle (Part 01)

```
Review Dependency:

dependencies {
    ..........
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'com.google.android.material:material:1.0.0'
    implementation 'androidx.cardview:cardview:1.0.0'
    implementation
        'androidx.lifecycle:lifecycle-extensions:2.2.0'
    implementation 'androidx.recyclerview:recyclerview:1.0.0'
    implementation
       'androidx.recyclerview:recyclerview-selection:1.0.0'
    implementation 'com.google.ar:core:1.15.0'
    implementation 'com.google.ar.sceneform:core:1.15.0'
    implementation 'com.google.ar.sceneform:assets:1.15.0'
    implementation
           'com.google.ar.sceneform.ux:sceneform-ux:1.15.0'
    implementation
           'com.google.ar.sceneform:animation:1.15.0'

    implementation 'com.google.gms:google-services:4.3.3'
    implementation
         'com.google.android.gms:play-services-auth:18.0.0'
    implementation
         'com.google.firebase:firebase-analytics:17.4.2'
    implementation
         'com.google.firebase:firebase-appindexing:19.1.0'
    implementation
         'com.google.firebase:firebase-auth:19.3.1'
    implementation
         'com.google.firebase:firebase-database:19.3.0'
    implementation
         'com.google.firebase:firebase-messaging:20.2.0'
    implementation
         'com.google.firebase:firebase-storage:19.1.1'
    ..............
}
```

Figure 5.4: App's Gradle (Part 02)

The next figure shows the project's build.gradle structure:

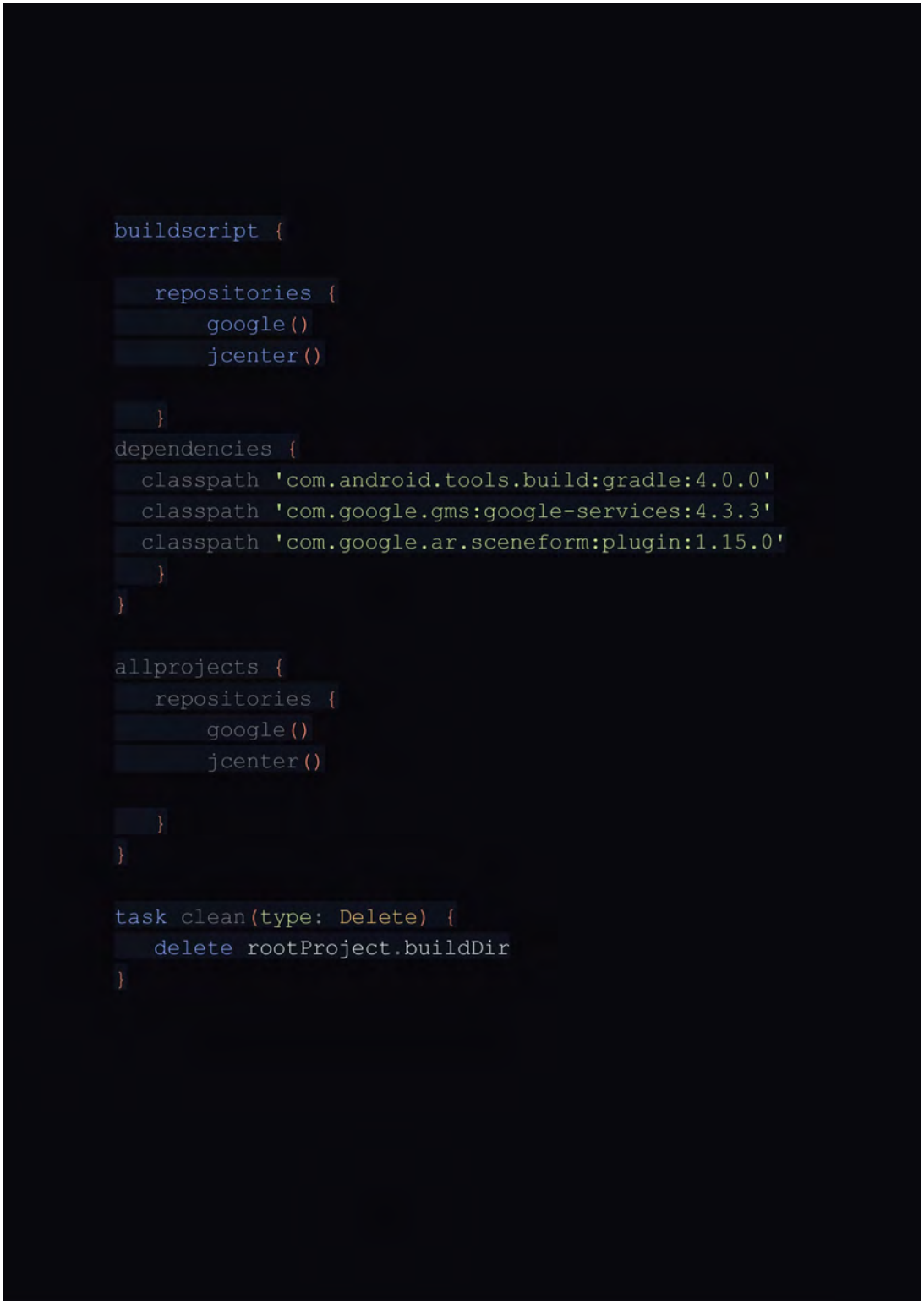


Figure 5.5: Projects's Gradle

Secondly, we have to work with AndroidManifest.xml file and it's structure will be according to the next 2 figures for our thesis project.

Figure 5.6: Manifest XML (Part 01)

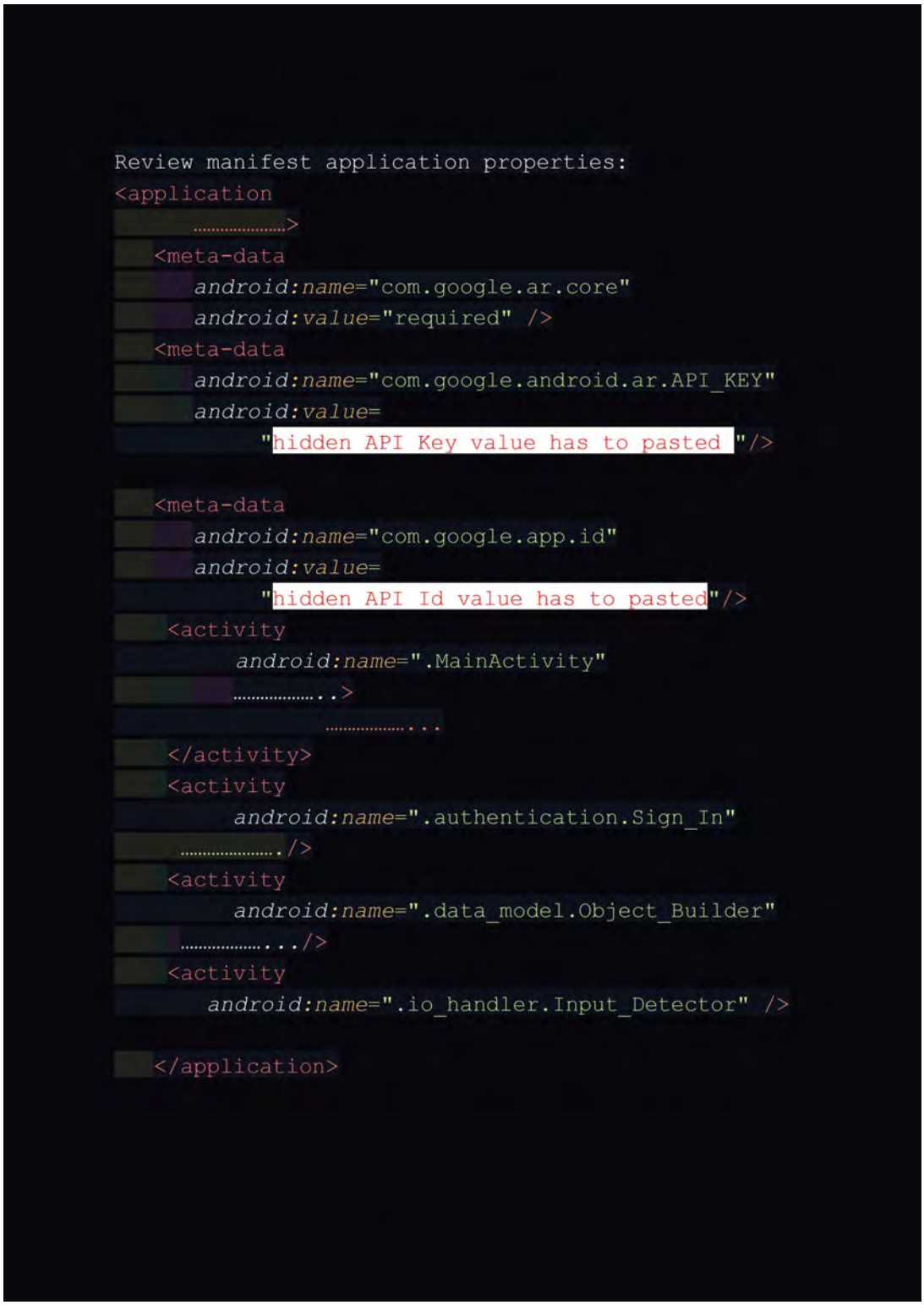


Figure 5.7: Manifest XML (Part 02)

Finally, we move to our app's main activity class and operational activities. Significant libraries to import for Main Activity class is shown in next figure:

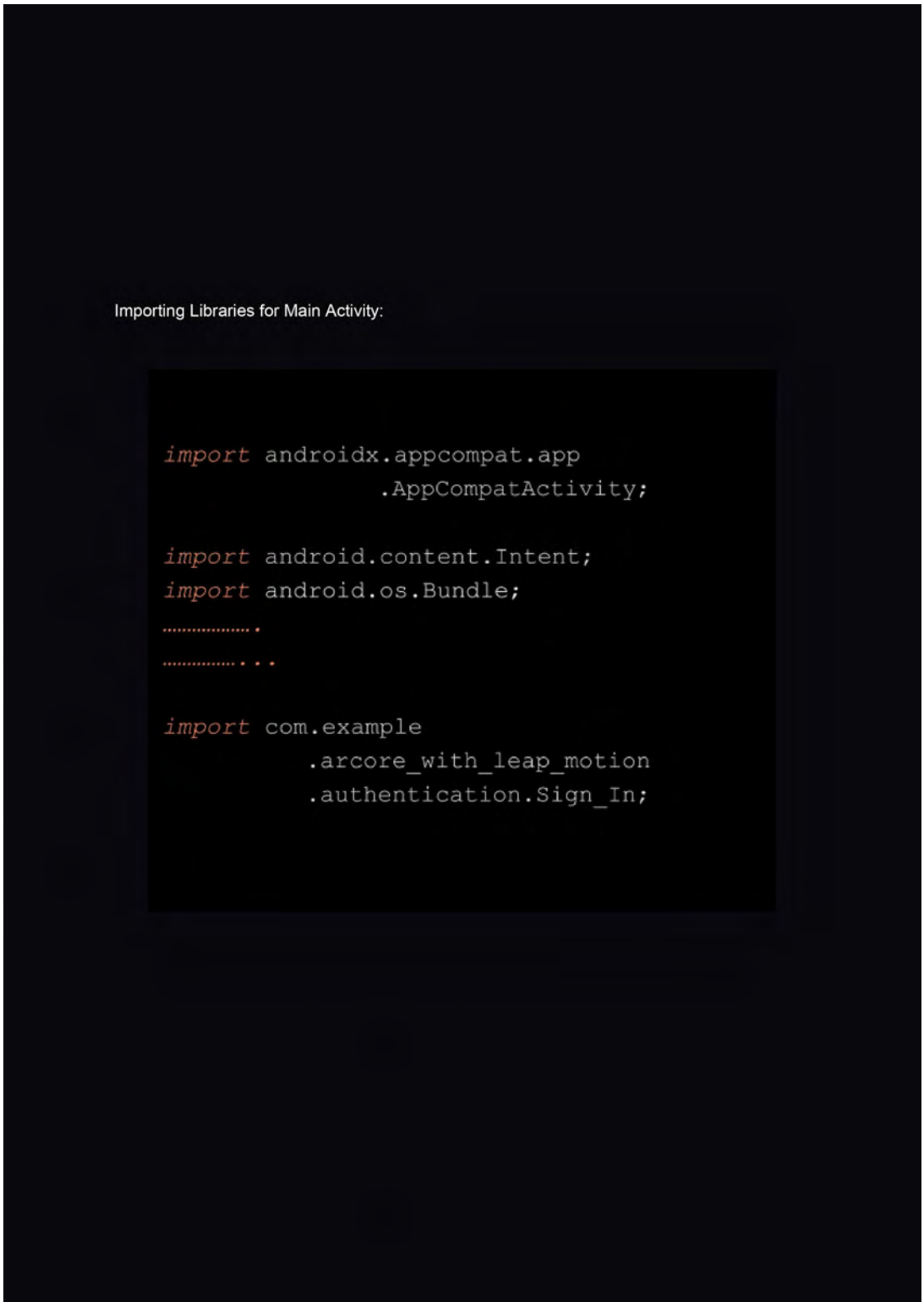


Figure 5.8: Importing Libraries for Main Activity

Then the next figure is about the basic structure of the Main activity:

Main Activity Overview:

```
public class MainActivity
             extends AppCompatActivity {

   @Override
   protected void onCreate(Bundle
                                savedInstanceState) {
        ..........
        ..........

        move_to_sign_in();
   }

   private void move_to_sign_in(){
     try{
        Intent intent = new Intent(this,
                                   Sign_In.class);
        startActivity(intent);
       }
     catch (Exception ex){
        ..............
       }
   }
}
```

Figure 5.9: Main Activity Class

The major task of Main activity class is done by method 'move_to_sign_in()'. It will just start a new activity named Sign_In activity to verify user login to the app and this is required for Firebase and Google API aspects.

So, libraries requirements and the Sign_In activity overview are shown in next 5 figures:

Importing Libraries for Sign_In Activity:

```
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
.....
import com.example.arcore_with_leap_motion.R;
import com.example.arcore_with_leap_motion
        .data_model.Object_Builder;
import com.example.arcore_with_leap_motion
        .io_handler.Input_Detector;

import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api
        .signin.GoogleSignInOptions;
import com.google.android.gms.auth.api
        .signin.GoogleSignInResult;
import com.google.android.gms.common
        .ConnectionResult;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api
        .GoogleApiClient;
```

Figure 5.10: Importing Libraries for Sign_In Activity

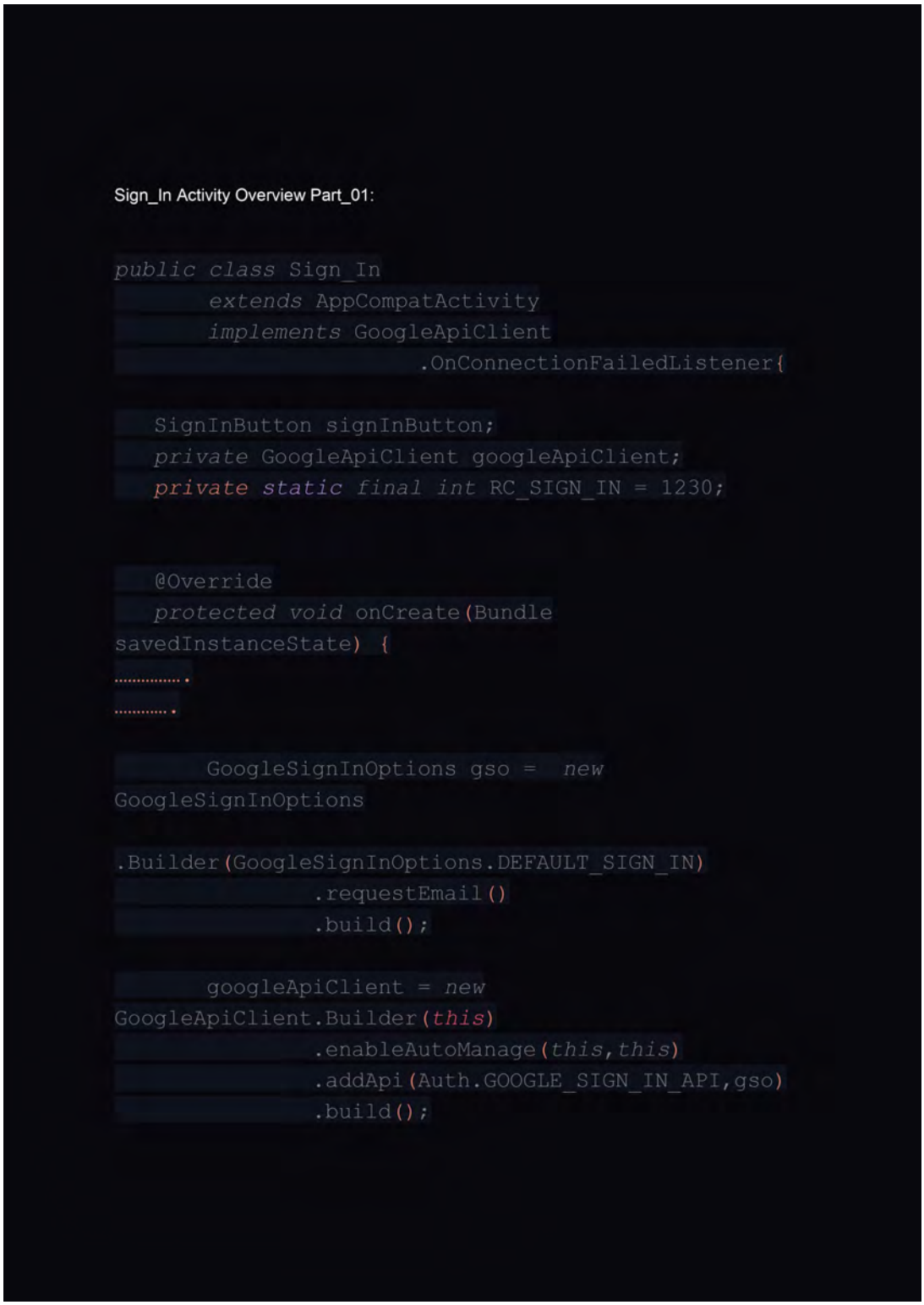


Figure 5.11: Sign_In Activity Class (Part 01)

Sign_In Activity Overview Part_02:

```
  signInButton = findViewById(
                            R.id.sign_in_button);

  signInButton.setOnClickListener(
                        new View.OnClickListener() {
     @Override
     public void onClick(View view) {
         Intent intent = Auth.GoogleSignInApi
                  .getSignInIntent(googleApiClient);

         startActivityForResult(intent,RC_SIGN_IN);
           }
       });
   }       // end of onCreate(..) method
.....

@Override
protected void onActivityResult(int requestCode,
                    int resultCode, Intent data) {

  super.onActivityResult(requestCode,
                                  resultCode, data);

  if(requestCode==RC_SIGN_IN){
   GoogleSignInResult result = Auth.GoogleSignInApi
                  .getSignInResultFromIntent(data);

   handleSignInResult(result);
       }

}
```

Figure 5.12: Sign_In Activity Class (Part 02)

Sign_In Activity Overview Part_03:

```
private void handleSignInResult(GoogleSignInResult
                                              result){
   if(result.isSuccess()){
       move_to_data_model();
           ..........................
   }
   else{
       ..............
   }
}

private void move_to_data_model(){
  final Dialog dialog = new Dialog(this);
  ..............
  Button left_option = dialog.findViewById
                            (R.id.go_to_input_btn);
  Button right_option = dialog.findViewById
                            (R.id.go_to_arcore_btn);

  left_option.setOnClickListener(new
                            View.OnClickListener() {
     @Override
     public void onClick(View v) {
        Intent intent = new Intent(
                          getApplicationContext(),
                          Input_Detector.class);
        startActivity(intent);
        ..............
     }
  });
```

Figure 5.13: Sign_In Activity Class (Part 03)

Sign_In Activity Overview Part_04:

```
  right_option.setOnClickListener(new
                                View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(
                               getApplicationContext(),
                               Object_Builder.class);

        startActivity(intent);
        ..................
    }
  });
      ..................

 }  // end of move_to_data_model() method

.........
.....
}   // end of Sign In Activity
```

Figure 5.14: Sign_In Activity Class (Part 04)

In the 'onCreate(...)' method of Sign_In activity, we initialized Google authentication service and by sign_in_button action listener and by 'onActivityResult(..)' method we have handled user login tasks. If the login process is accepted then 'handleSignInResult(...)' is called inside 'onActivityResult(..)'. 'handleSignInResult(...)' method calls the method 'move_to_data_model()' to select the Object_Builder activity that is responsible for ARCore main view and is responsible for controlling 3D view in the ARCore android App.

Therefore, the Object_Builder activity is shown in next 2 figures:

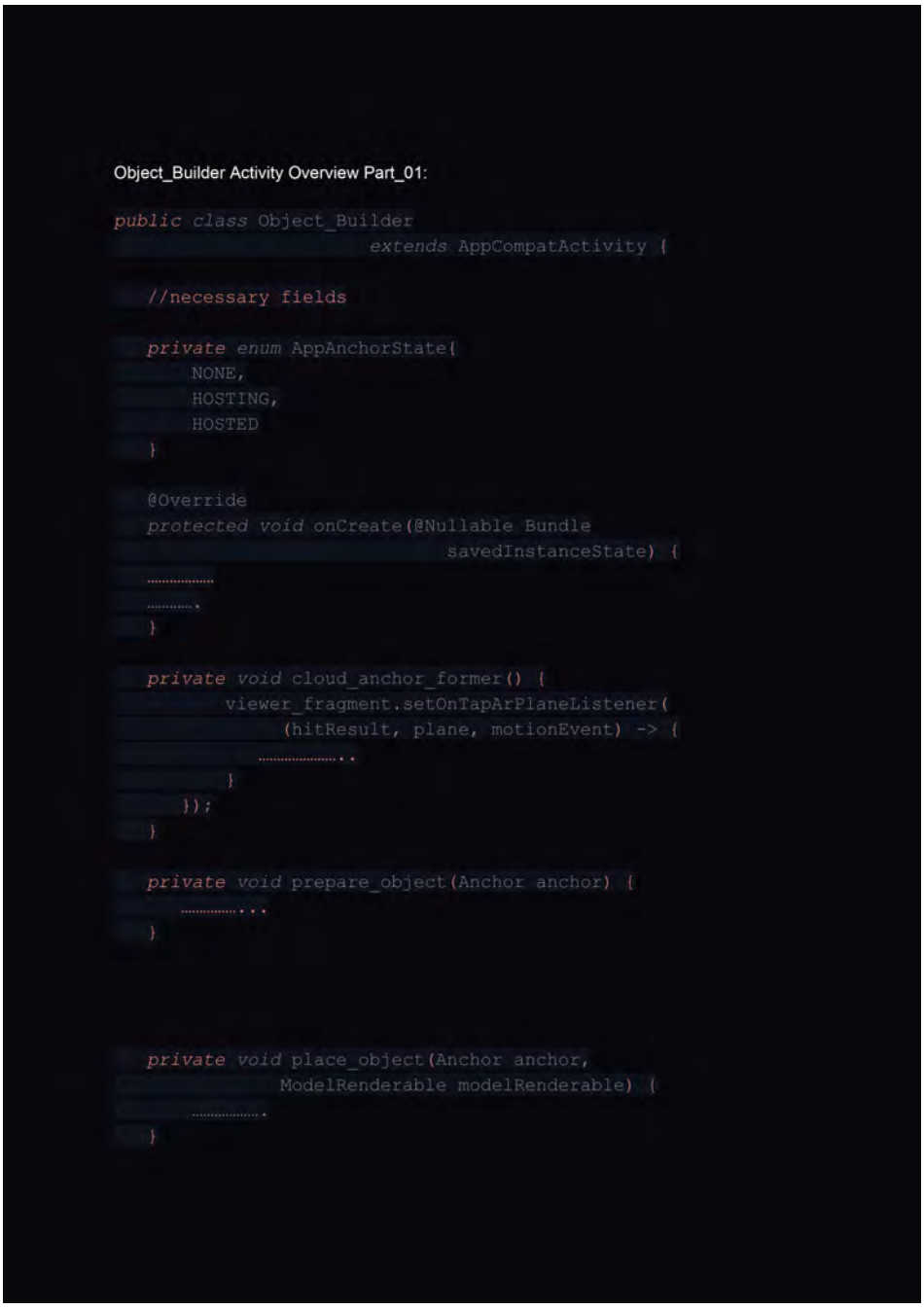


Figure 5.15: Object_Builder Activity Class (Part 01)

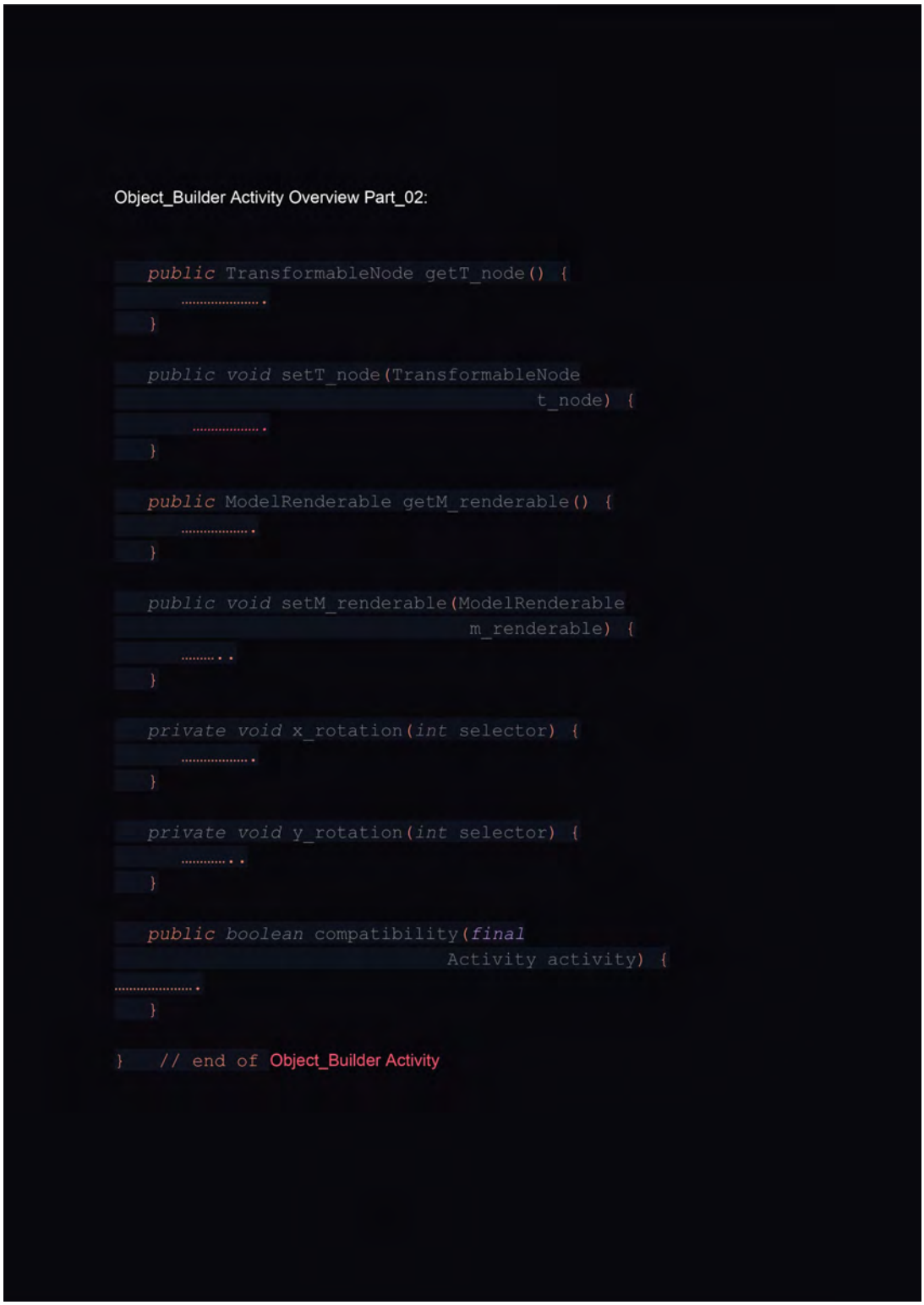


Figure 5.16: Object_Builder Activity Class (Part 02)

To describe the Object_Builder activity, we should talk about the 'compatibility(..)' method. This method checks whether the device is compatible with the ARCore functionality. If supported, then it will work. Otherwise, it will show notification of incompatibility.

Whenever ARCore compatibility is confirmed it will call 'cloud_anchor_former()' and after detecting the plane and listening to the tap in scene view screen will generate 3D content(in our case Cubes). This process works with the combination of 'prepare_object(...)', 'place_object(..)', 'setM_renderable(..)', getM_renderable() etc methods.

In the end 'x_rotation(...)' and 'y_rotation(...)' are called when clicked or Firebase database value is updated according to input detection and when the object's anchor is resolved

There is one more important thing and it is about ARCore Cloud session hosting and maintaining ARFragment scene-view. Custom_AR_Fragment will do the job and it's overview is given in the next figure:

Figure 5.17: Custom_AR_Fragment Activity Class

To keep the description (report) short we ignored insignificant methods and full functionality description code.

We will close the implementation by showing: how a 3D content is look like after successfully finishing the project's build run.

This figure is the representation of the 3D object[10]. Later, we used a different 3D model for our final ARcore App[11]. We also took help from the youtube channel[9].

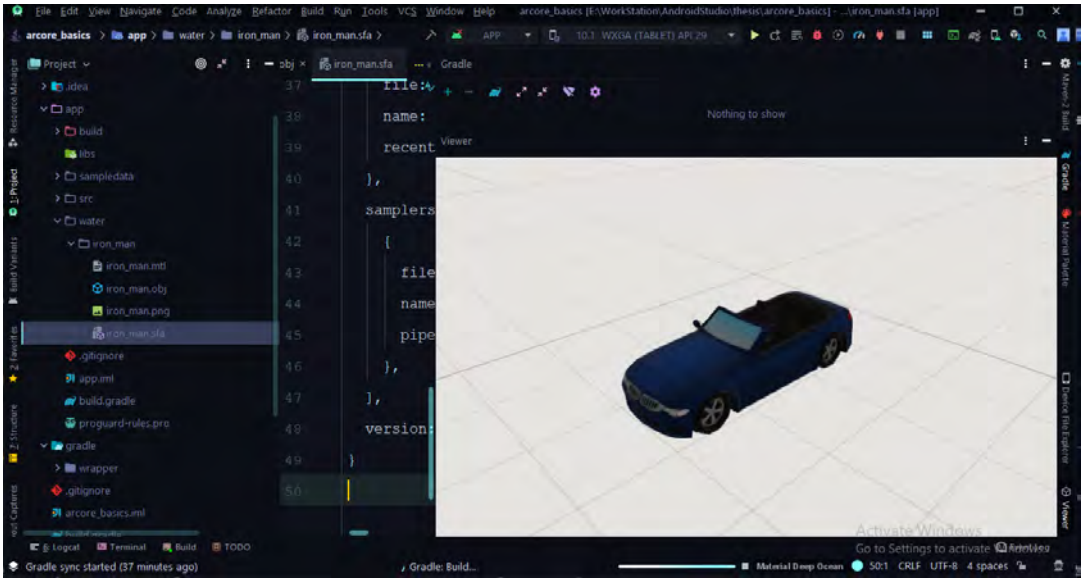


Figure 5.18: Successful Build of an 3D assets in Android Studio IDE

Chapter 6

Result Analysis

6.1 Effective Output

After completing implementing code works, we build an apk file and generate an Android App. Then we get active UI components and consequently our desired output. The next 5 figures shows the UI operation and output of our App:

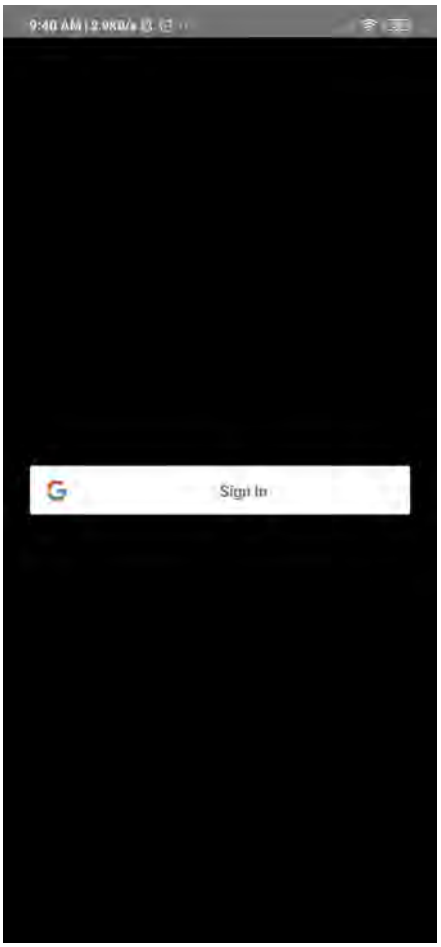


Figure 6.1: UI (Part 01)

Figure 6.2: UI (Part 02)

Figure 6.3: UI (Part 03)

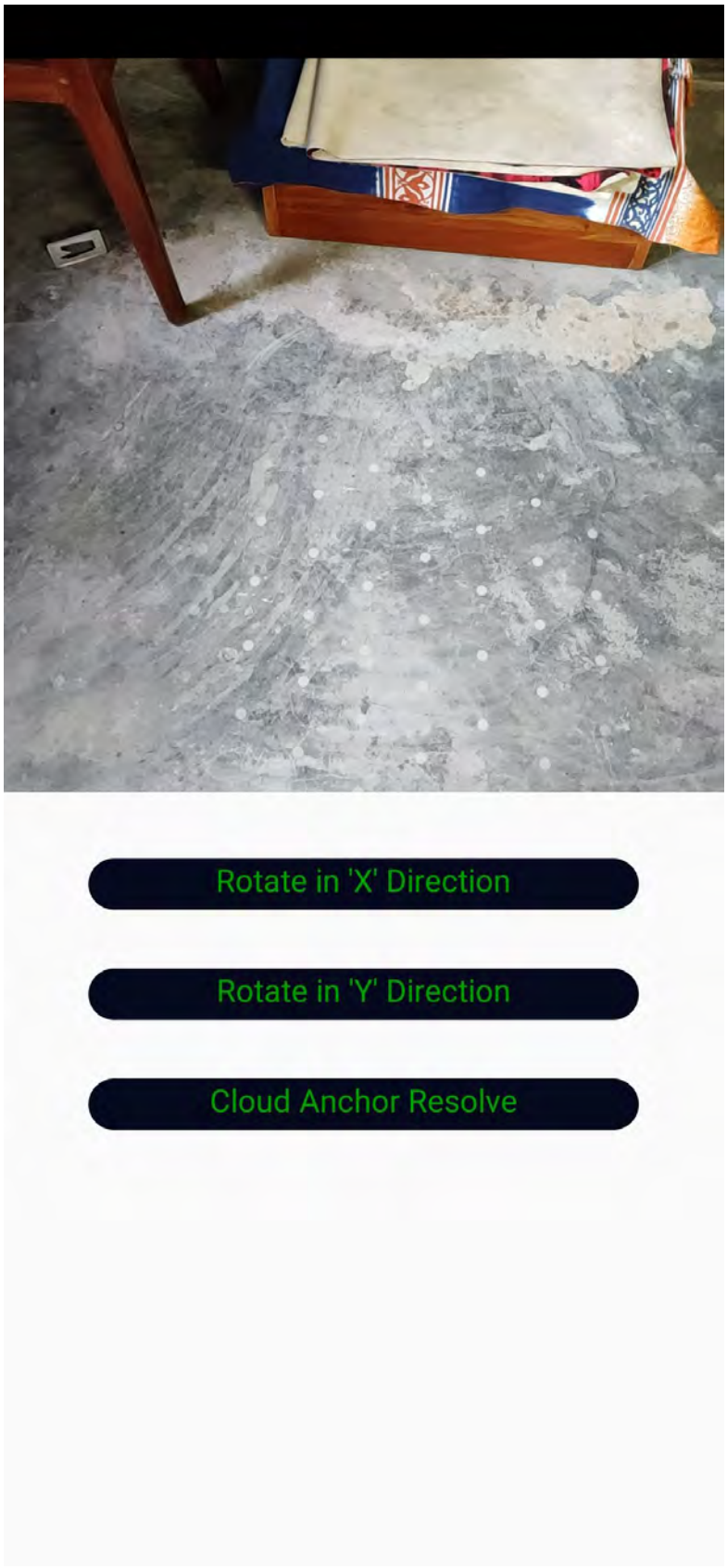


Figure 6.4: UI (Part 04)

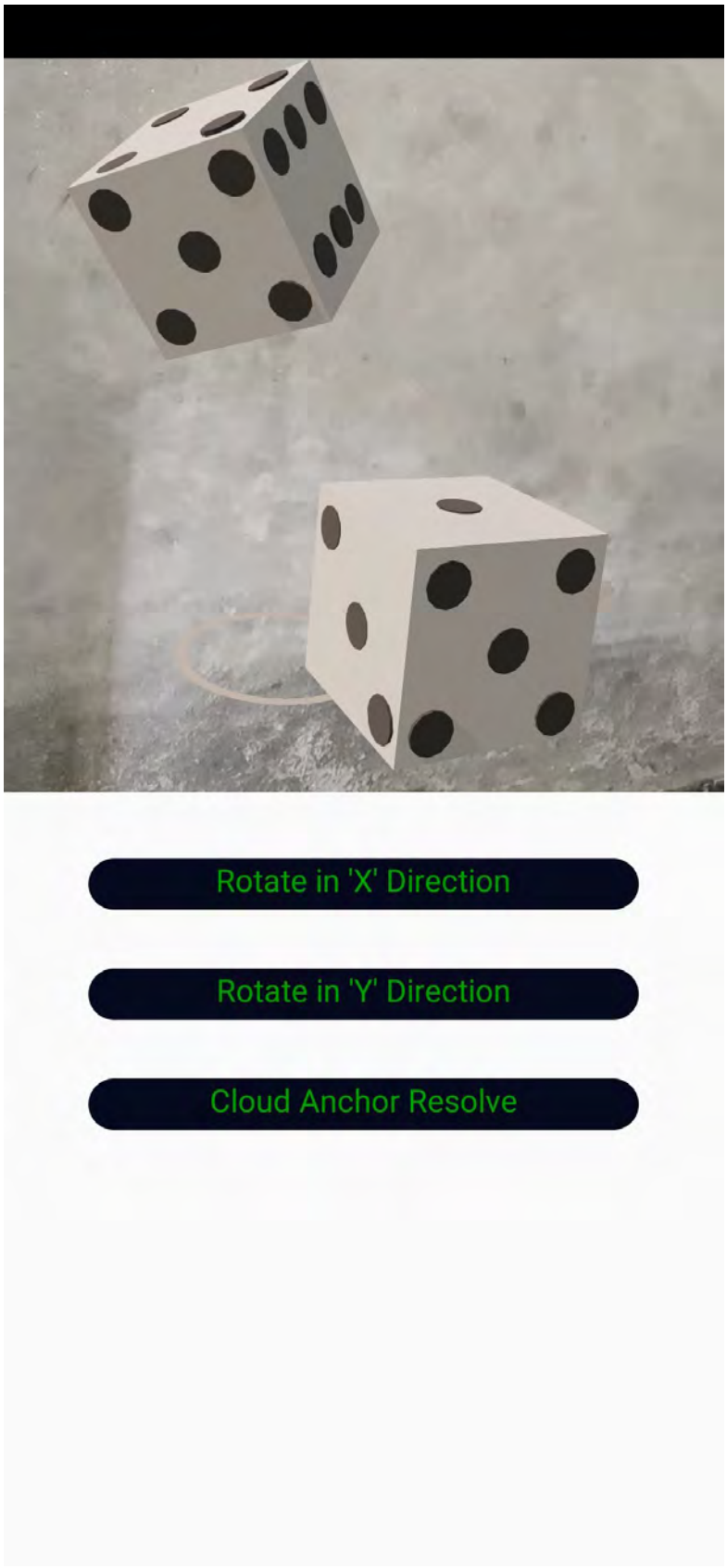


Figure 6.5: UI (Part 05)

Here, in figure UI(Part 01) we open our app and see the Sign In button and Sign In progress is continued to figure UI(Part 03). And figure UI(Part 04) is the main view where we will see 3D content. After detecting the plane and tapping on the main view, 3D content appeared in figure UI(Part 05). It is the cube that we were looking for (Augmented Object). Now we can manipulate the cube using buttons [Rotate in 'X' Direction] and [Rotate in 'Y' Direction] inside ARCore App and by Leap Motion Controller (both facilities).

The next 2 figures shows the X-axis rotation (either from Leap Motion sensor input or from ARCore App Button):

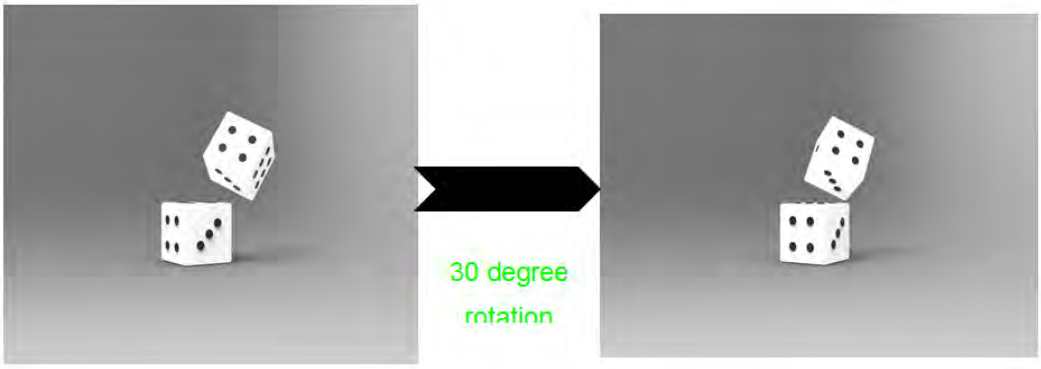


Figure 6.6: Object Manipulation (Part 01)

The next 2 figures shows the Y-axis rotation (either from Leap Motion sensor input or from ARCore App Button):

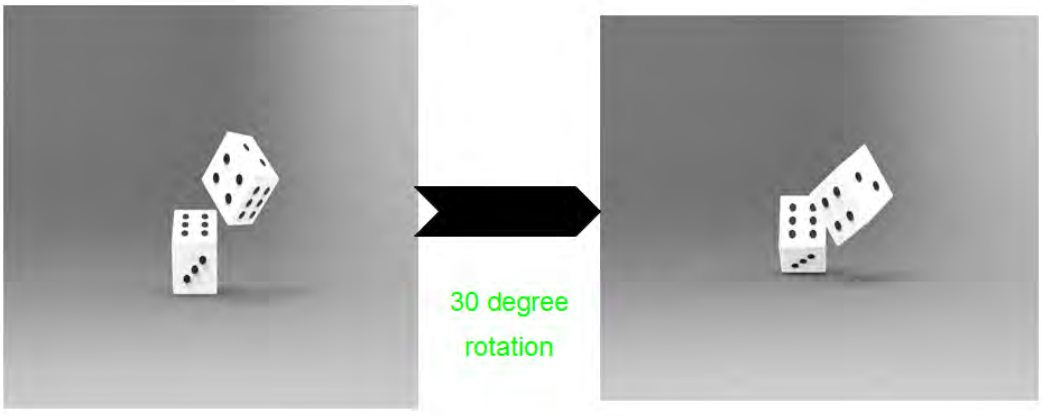


Figure 6.7: Object Manipulation (Part 02)

However, our final 3D representation looks like the next pictures using projection in holographic display[it will be attached to the top of phone screen] and we can see it by our eyes outside the ARCore supported phone:

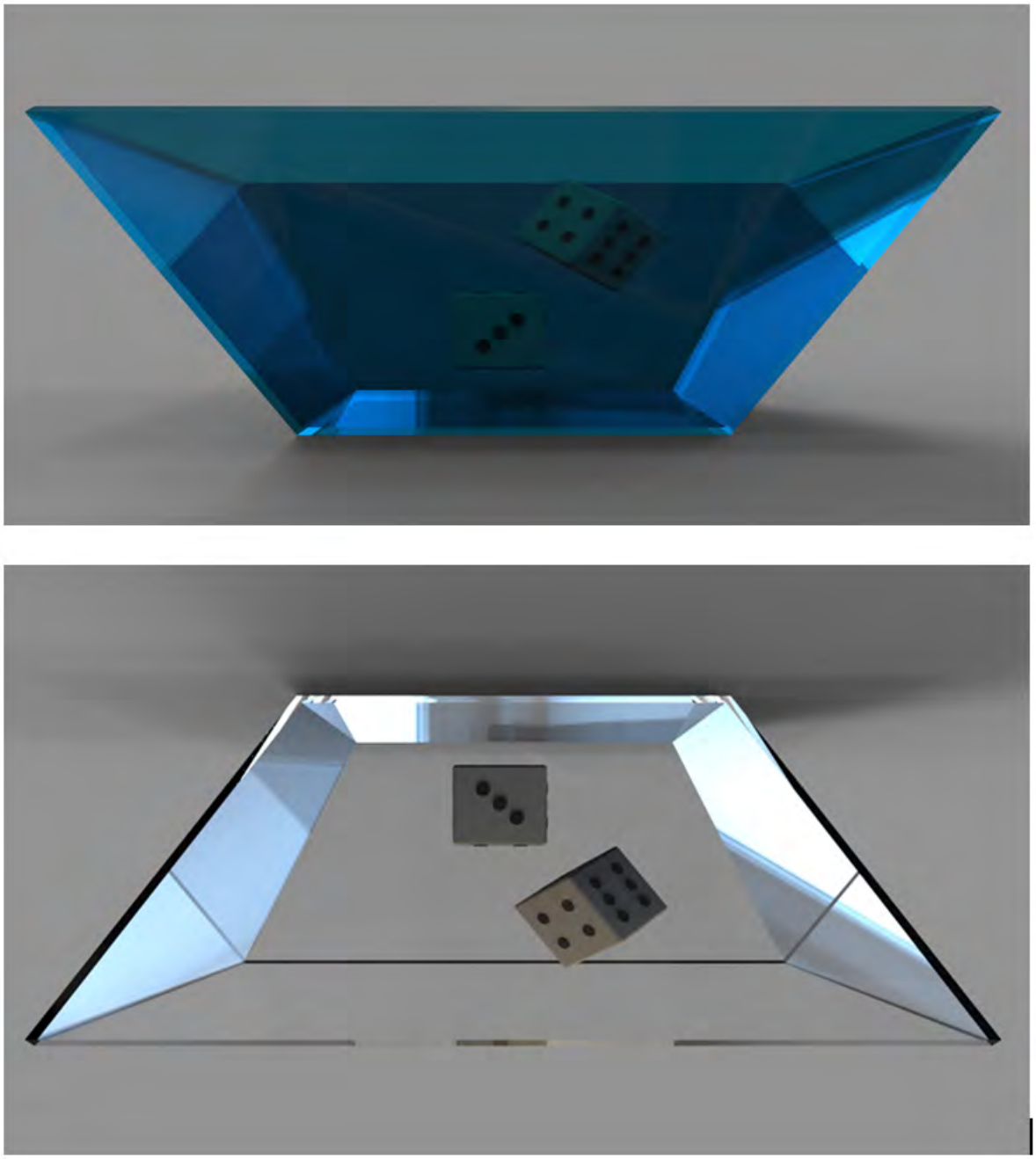


Figure 6.8: Holographic Display Projection (Simplified Image)

6.2 Drawbacks

It is quite common that during the work progress we faced a lot of issues. Output of ARCore Android Application is sound but its configuration and implementing it is very tough and tricky. Secondly, the version miss-match is also another problem. Thirdly, having a compatible device is under huge cost as the ARCore is currently not available in the older device. Besides arranging, Leap Motion Controller is also very hard because it is not available easily in the local area. Again, to implement and test it we must use internet facilities. It mainly operates through the internet (Firebase).

Chapter 7

Conclusion

Although the Leap motion and ARCore jointly exist together before, we know that: it was via other operating systems or platforms (such as Unity). So, we worked on Android Platform where ARCore has been previously integrated or Leap motion only with Android Apps. We specifically have worked on ARCore and Leap Motion via Firebase using Android Platform (with help of Windows : as we have no android hardware available that sends data from Leap Motion).

However, we have shown how to integrate them and successfully completed our goal.

7.1 Aspects & Prospects

There is no denying the fact that some works at first sight seem nothing special in the technological fields. Still, there is hope and scope beyond if we treat it more deeply. In our case, it is just an integration. Nevertheless, it will help software companies and developers zone in the Augmented Reality Research sectors and 3D manipulation of data with free hand movement in the aspects of Leap Motion Controller. Again, children learning , art, architectural development etc also will be benefited (interactive 3D classroom representation) if a hardware integrating leap sensor with an android operating system (to exchange information via Android) or talking help from windows or linux or mac operating system which are already existing. 3D concepts and mathematical sectors will also be benefited if our idea is extended and implemented on merging them in a single hardware component. However, technology is developing and it is a small effort to keep pace with it.

Bibliography

[1] M. Kesim and Y. Ozarslan, "Augmented reality in education: Current technologies and the potential for education", *Procedia - Social and Behavioral Sciences*, vol. 47, pp. 297–302, 2012. DOI: 10.1016/j.sbspro.2012.06.654.

[2] I. Perdana, "Teaching elementary school students new method of music performance with leap motion", *2014 International Conference on Virtual Systems Multimedia (VSMM)*, 2014. DOI: 10.1109/vsmm.2014.7136655.

[3] Ccarr@farotech.com, *Review of leap motion technology: Leap motion review*, Apr. 2017. [Online]. Available: https://farotech.com/blog/review-leap-motion-new-touchless-technology/.

[4] E. Niechwiej-Szwedo, D. Gonzalez, M. Nouredanesh, and J. Tung, "Evaluation of the leap motion controller during the performance of visually-guided upper limb movements", *Plos One*, vol. 13, no. 3, 2018. DOI: 10.1371/journal.pone.0193639.

[5] ARchy, *The history of ar and vr*, Jul. 2019. [Online]. Available: https://arvrjourney.com/the-history-of-ar-and-vr-3faea3f1e94b.

[6] *Leap motion*, Aug. 2020. [Online]. Available: https://en.wikipedia.org/wiki/Leap_Motion.

[7] *What's the difference between ar, vr, and mr?*, Jan. 2020. [Online]. Available: https://www.fi.edu/difference-between-ar-vr-and-mr.

[8] *Arcore overview — google developers*. [Online]. Available: https://developers.google.com/ar/discover.

[9] *Build ar apps in android studio using sceneform sdk (arcore)*. [Online]. Available: https://www.youtube.com/playlist?list=PLsOU6EOcj51cEDYpCLK_bzo4qtjOwDWfW.

[10] *Convertible*. [Online]. Available: https://poly.google.com/view/dggOiBLYyuR.

[11] *Dice roll*. [Online]. Available: https://poly.google.com/view/4eW4uL5poDZ.

[12] *Documentation : Android developers*. [Online]. Available: https://developer.android.com/docs.

[13] *Documentation — firebase*. [Online]. Available: https://firebase.google.com/docs.

[14] *Java programming language — developer tools — google cloud*. [Online]. Available: https://cloud.google.com/java.

[15] *Java sdk documentation¶*. [Online]. Available: https://developer-archive.leapmotion.com/documentation/java/index.html.

[16] *Leapmotion*. [Online]. Available: https://medium.com/@LeapMotion.

[17] *Package index — arcore — google developers*. [Online]. Available: https://developers.google.com/ar/reference/java/packages.

[18] *Share ar experiences with cloud anchors — arcore — google developers*. [Online]. Available: https://developers.google.com/ar/develop/java/cloud-anchors/overview-android.

[19] *Spring boot*. [Online]. Available: https://spring.io/projects/spring-boot.

[20] Ultraleap, *World-leading hand tracking: Small. fast. accurate.: Ultraleap*. [Online]. Available: https://www.leapmotion.com/technology.

Comments

We were questioned by only two of the honourable faculty members from the thesis panel. We have included their comments hereby:

Matin Saad Abdullah, PhD:

He asked us a couple of questions. Among of them: he wanted to get convinced about if the ARCore part can be handled through Cloud computing. He has reasoned us that, whether the mobile device is strong enough to render 3D contents.

Md. Golam Rabiul Alam, PhD:

He wanted us to include more effective elaboration of last part of the topic where we included in the topic the term 'Enrich Educational Process and Research Progress'.