VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI

A Mini Project Report

On

KIDZ QUIZ GAME

Submitted in Partial fulfillment of requirement for the

MOBILE APPLICATION DEVELOPMENT LABORATORY WITH MINI PROJECT (18CSMP68**)**

In

Computer Science & Engineering

By

S DEEPA	3VC20CS132
K PAVANI	3VC20CS079
HARSHITHA B	3VC20CS065

Faculty In-Charge

PRASANNA KUMAR S SHIVARADDI

Assistant Professor,

CSE Department, RYMEC

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ACCREDITED BY NATIONAL BOARD OF ACCREDITATION

RAO BAHADUR Y MAHABALESHWARAPPA ENGINEERING COLLEGE

CANTONMENT, BALLARI-583104, KARNATAKA

2022-23

VEERASHAIVA VIDYAVARDAHKA SANGHA'S

RAO BAHADUR Y MAHABALESHWARAPPA ENGINEERING COLLEGE

(AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM & APPROVED BY AICTE, NEWDELHI)
BALLARI – 583104, KARNATAKA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ACCREDITED BY NATIONAL BOARD OF ACCREDITATION

CERTIFICATE

This is to certify that project work entitled **"KIDZ QUIZ GAME"** is a bonafied Work carried out by **S DEEPA (3VC20CS132), K PAVANI (3VC20CS079), HARSHITHA B (3VC20CS065)** of 6th Semester in Partial fulfillment of requirement for **MOBILE APPLICATION DEVELOPMENT LABORATORY WITH MINI PROJECT (**18CSMP68**)** during the year **2022-23.**

Signature of Staff - Incharge
Mr. PRASANNA KUMAR S SHIVARADDI
Assistant Professor,
Dept. of CSE, RYMEC

Signature of HOD
Dr. GIRISHA H
HOD, Dept. of CSE
RYMEC.

Name of Examiners: **Signature with Date**

1.

2.

ACKNOWLEDGEMENT

The satisfaction and excitement that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success.

We wish to place on record our grateful thank to **Dr. T. HANUMANTHA REDDY**, Principal, RYMEC, Ballari for providing encouragement and guidance.

We wish a grateful thank to **Dr. H. GIRISHA**, Head of the Department, Computer Science and Engineering RYMEC, Ballari for providing encouragement and guidance.

We are highly indebted **Mr. PRASANNA KUMAR S SHIVARADDI** for guidance and constant supervision as well as for providing necessary information regarding the project & also for support incompleting the project.

We thank and appreciate the help of my friends in developing the project and people who have willingly helped me out with their abilities.

PROJECT ASSOCIATES

S DEEPA 3VC20CS132

K PAVANI 3VC20CS079

HARSHITHA B 3VC20CS065

ABSTRACT

This mini project on **KIDZ QUIZ GAME** displays a quiz game for kids, where they can learn the basics math from this project. This will help all the kids from the beginning to work under the math. The main aim of this is make the math much easier and friendly to the kids. This even may help the parents from knowing how the child can be trained up further in school. This project helps us to know all other basics knowledge of math.

This mini project is very user friendly for all the kids. Easily applicable and accessible by all the kids. It might help us to learn basics of math. This even helps the parents to free up for the kids to know what all the basics the kid knew in math.

These results have implications for educators, parents, and developers of educational apps, as they provide evidence of the effectiveness of math quiz game apps as supplementary learning tools. Future research could focus on investigating the long-term effects of using kidz quiz game apps, exploring customization options to cater to individual learning needs, and assessing the impact of social features for collaborative learning experiences.

Qualitative feedback from users further supported the positive impact of the math quiz game app on their learning experience. Participants expressed enjoyment and satisfaction with the app's user-friendly interface and challenging yet rewarding game play. They also highlighted the app's convenience and accessibility, enabling them to practice math skills anytime and anywhere.

CONTENTS

CHAPTERS

CHAPTER 1

INTRODUCTION

Android Development

The Android operating system is the largest installed base among various mobile platforms across the globe. Hundreds of mil ions of mobile devices are powered by Android in more than 190 countries of the world. It conquered around 75% of the global market share by the end of 2020, and this trend is growing bigger every other day. The company named Open Handset Alliance developed Android for the first time that is based on the modified version of the Linux kernel and other open-source software.

Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on Intel iJ IDEA . On top of Intel iJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- ➢ A flexible Double-barrelled build system
- ➢ A fast and feature-rich emulator
- ➢ A unified environment where you can develop for all Android devices
- ➢ Apply Changes to push code and resource changes to your running app without restarting your app
- ➢ Code templates and GitHub integration to help you build common app features and import sample code
- ➢ Extensive testing tools and frameworks
- ➢ Lint tools to catch performance, usability, version compatibility, and other problems
- ➢ C++ and NDK support

➢ Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

PROJECT STRUCTURE

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

1. Android app modules
2. Library modules
3. Google App Engine modules

By default, Android Studio displays your project files in the Android project view. This view is organized by modules to provide quick access to your project's key source files. Al the build files are

Visible at the top level under Gradle Scripts and each app module contains the following folders:

➢**manifests**: Contains the Android **Manifest.xml** file.

➢**java**: Contains the Java source code files, including JUnit test code.

➢**res**: Contains al non-code resources, such as XML layouts, UI strings, and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select Project from the **Project drop down**. You can also customize the view of the project files to focus on specific aspects of your app development. For example, selecting the Problems view of your project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element closing tag in a layout file.

Importance of Android Development

1. **Android Development will** give you good concepts of development and you can work on iOS too. However the language used here is different but I have read that an android developer can make an iOS application easily with a month of training.

2. With these android application learning you'll enhance your possibilities of going far away in **android development**.

3. An Android phone. Although this is optional, **emulators** tend to be very, very slow. Get an Android phone and enable developer options. After that enable USB debugging and you can use your **phone** to install and debug applications from the Studio.

4. **Software Development Kit**. An SDK is a bundle of tools that comprises an executable program. This includes documentation, debuggers, emulators, frameworks, libraries, profiles, and more. Android SDK is already included in Android Studio, but if you want to use another IDE, you can download it separately.

5. **Editors and IDE**. In theory, you can write Android apps in a regular text editor or command line, but the common approach is using an Integrated Development Environment. This tool integrates al SDK tools and helps to manage them more easily and in a more user-friendly manner. Android Studio is the official Android IDE, but other options are also popular. Eclipse is Studio's predecessor that can use plugins to expand the code to more languages. Intel iJ IDEA is a paid but highly customizable option

6. **Programming languages**. Java and Kotlin are listed as the official languages for Android programming but there are alternatives. You can also use C and C++ using the Android Native Development Kit – the tool for implementing parts of previously written apps in native code. There are also third-party tools allowing you to create native Android apps using your favorite languages such as Ruboto (Ruby) or Kivy(Python).

7. **Libraries**. Software developers use libraries for al kinds of tasks. They are snippets of pre-written code that automate a coder's job and eliminate

the need to reinvent the wheel. The Android community is generous about such free solutions. The most popular of them include GSON for serializing and deserializing Java objects to communicate with APIs,

Retrofit for API organization, and Event Bus for easy communication between different app element. To configure al these setting and organize the process of adding external libraries, Developers use the tool called Gradle.

8. **Plugins**. While libraries are used to automate project tasks, plugins are created for the augmentation of each software tool of IDE.

Project Goal

The following are the Goals/Objectives of kids Quiz Game application:-

- Create intuitive and user-friendly interface for seamless navigation.
- Generate random basic questions from various topics and difficulty levels.
- Implement a scoring system to track user performance and progress.
- Provide feedback and explanations for incorrect answers to aid learning.
- Enable social features to allow users to complete with friends and share achievements.
- Ensure compatibility and optimal performance across different mobile platforms.

Scope

A kids quiz game app can have a wide scope, depending on its target audience, features, and complexity. Here are some potential elements that can be included in a kids quiz game app:

1. Question Bank: The app can include a large database of kids questions covering various topics and difficulty levels. The questions can be categorized based on subjects like arithmetic, algebra, geometry, calculus, etc.

2. Multiple Game Modes: The app can offer different game modes to cater to different preferences and skill levels. For example, it can have a timed mode where users need to answer questions within a specific time limit, a practice mode without time constraints, a multiplayer mode to compete against friends, or a campaign mode with progressively challenging levels.

3. Difficulty Levels: The app can allow users to select different difficulty levels, such as easy, medium, or hard, to suit their skill level and progress gradually.

4. Score Tracking and Leaderboards: Users can track their scores and progress within the app. Additionally, the app can include leaderboards to encourage healthy competition among users.

5. Educational Content: Alongside the quiz component, the app can provide explanations and solutions for each question to help users learn from their mistakes. It can also include kids concepts, formulas, and examples as reference material.

6. Gamification Elements: To make the app engaging and enjoyable, it can incorporate gamification elements such as rewards, achievements, badges, or unlockable content as users progress through the quizzes.

7. Customization and Personalization: The app can allow users to customize their experience by selecting preferred themes, avatars, or backgrounds. It can also offer personalized recommendations based on the user's performance and areas of improvement.

8. Offline Mode: Users may want to access the app and play quizzes without an internet connection. Including an offline mode can enhance the app's accessibility and usability.

9. Social Features: The app can integrate social features, allowing users to connect with friends, share their scores or achievements on social media platforms, or even compete in real-time multiplayer quizzes.

10. Accessibility Options: Consider including accessibility features like adjustable font sizes, color schemes for visually impaired users, and support for screen readers to ensure inclusivity.

Remember that the scope of a kids quiz game app can vary depending on factors such as budget, development resources, and target audience. It's essential to prioritize key features and create a user-friendly experience that promotes learning and engagement.

CHAPTER 2

LITERATURE SURVEY

kids quiz game apps are popular educational tools that engage users in interactive and entertaining kids challenges. These apps aim to enhance kidsematical skills, improve problem-solving abilities, and foster a love for kidsematics in users of all ages. In this literature survey, we explore relevant studies and articles on kids quiz game apps, analyzing their effectiveness, impact, and design considerations.

kids quiz game apps have gained significant attention as valuable educational tools for improving kids sematical skills and fostering engagement. The reviewed studies highlight the positive impact of these apps on student motivation, learning outcomes, and attitudes towards kids sematics. Additionally, design principles and considerations provide insights for developers and educators to create effective and engaging kids quiz game apps.

The Android Architecture

Android architecture contains a number of components to support any android device needs. Android software contains an open-source Linux Kernel having a collection of C/C++ libraries which are exposed through an application framework services.

Among al the components Linux Kernel provides main functionality of operating system functions to smartphones and Dalvik Virtual Machine (DVM) provides a platform for running an android application.

The main components of android architecture are following:-

➢ **Applications** :-Applications is the top layer of android architecture. The pre-instaled applications like home, contacts, camera, galeryetc and third party applications downloaded from the playstore like chat applications, games etc.

➢**Application Frame work** :-Application Framework provides several important classes which are used to create an Android application. It provides a generic abstraction for hardware access and also helps in managing the user interface with application resources.

➢**Android Run time** :- Android Runtime environment is one of the most important parts of Android. It contains components like core libraries and the Dalvik virtual machine(DVM). Mainly, it provides the base for the application framework and powers our application with the help of the core libraries.

➢**Platform Libraries** :-The Platform Libraries includes various C/C++ core libraries and Java based libraries such as Media, Graphics, Surface Manager, OpenGL etc. to provide support for android development.

➢**Linux Kernel** :-Linux Kernel is the heart of the android architecture. It manages al the available drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc. which are required during the runtime.

Pictorial representation of android architecture with several main components and their sub components: As shown in the **Figure 2.1**

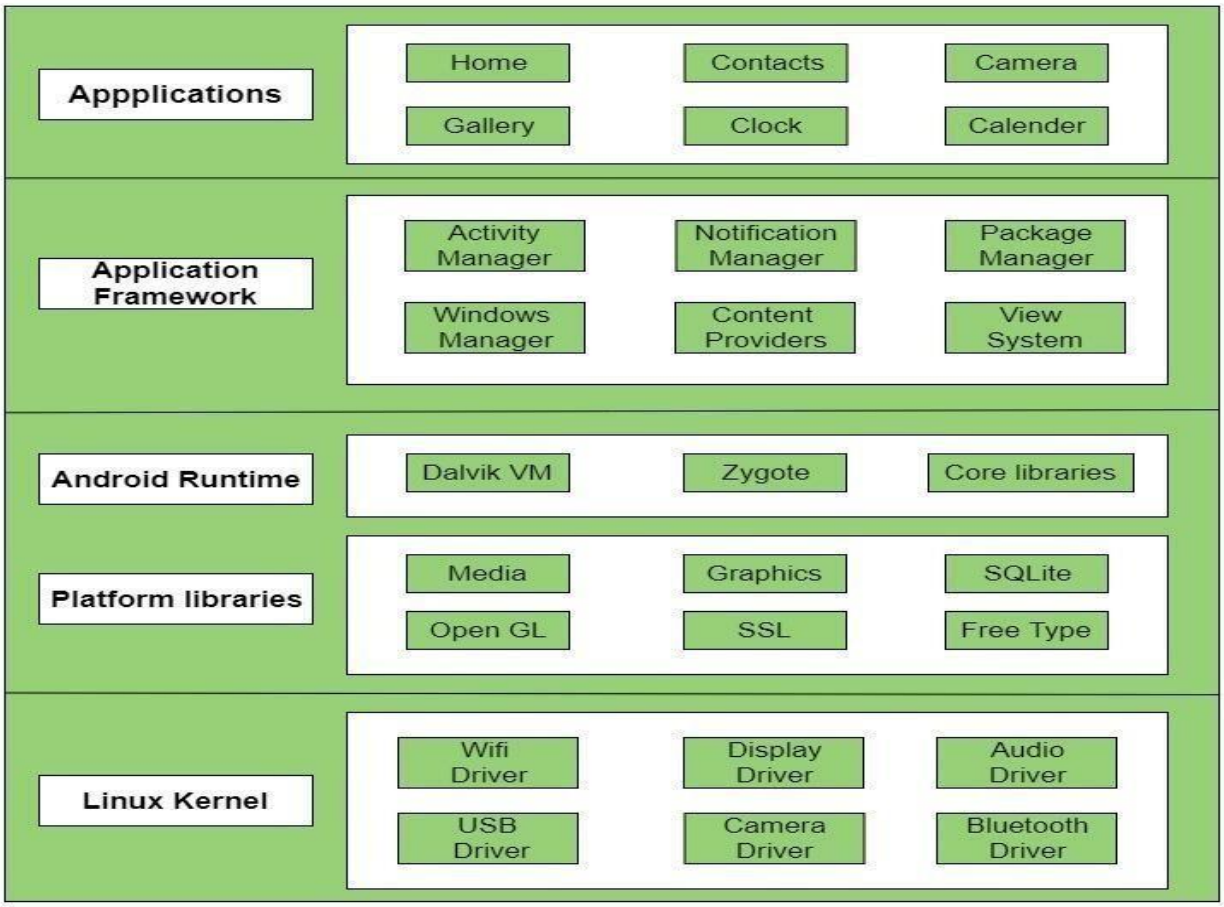


Figure 2.1 Android Architecture

About the Features of Android

Android is a powerful open-source operating system that open-source provides immense features and some of these are listed below.As shown **in Figure 2.2**

➢ Android Open Source Project so we can customize the OS based on our requirements.

➢ Android supports different types of connectivity for GSM, CDMA, Wi-Fi, Bluetooth, etc. for telephonic conversation or data transfer.

➢ Using wi-fi technology we can pair with other devices while playing games or using other applications.

➢ It contains multiple APIs to support location-tracking services such as GPS.

➢ We can manage al data storage related activities by using the file manager.

➢ It contains a wide range of media supports like AVI, MKV, FLV, MPEG4, etc. to play or record a variety of audio/video.

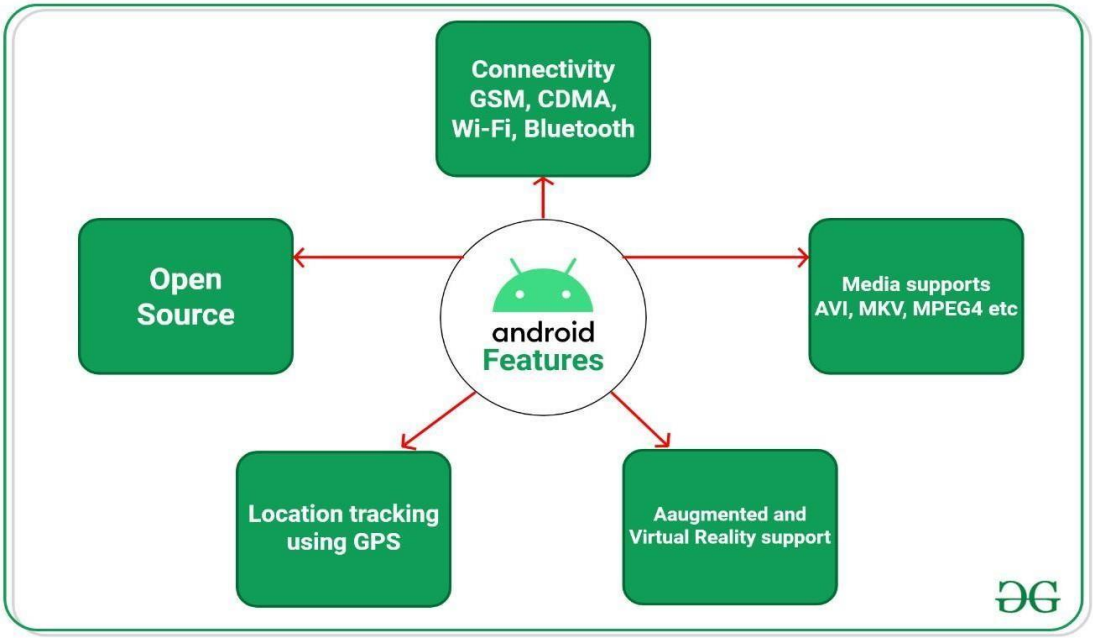
➢ It also supports different image formats like JPEG, PNG, GIF, BMP, MP3, etc.

➢ It supports multimedia hardware control to perform playback or recording using a camera and microphone.

➢ Android has an integrated open-source Web Kit layout-based web browser to support User Interface like HTML5, CSS3.

➢ Android supports multi-tasking means we can run multiple applications at a time and can switch in between them.

➢ It provides support for virtual reality or 2D/3D Graphics.



Basic Working of kids Quiz Game

A kids quiz game app typically follows a basic working principle that involves several components. Here's a simplified outline of how a kids quiz game app might work:

1. User Interface: The app presents a user interface where the player can interact with the game. This interface may include elements such as buttons, input fields, and a display area for questions and answers.

2. Game Modes and Difficulty Levels: The app may offer different game modes, such as timed challenges or multiple-choice quizzes. It may also provide various difficulty levels to cater to different skill levels of players.

3. Question Generation: The app generates kids questions dynamically based on the selected difficulty level or game mode. The questions can cover various kids semantical concepts, such as addition, subtraction, multiplication, division, fractions, or algebraic equations.

4. Displaying Questions: The app displays the generated question on the user interface, allowing the player to read and comprehend it.

5. User Input: The player provides their answer to the displayed question using the input fields or by selecting options in the case of multiple-choice questions.

6. Answer Evaluation: After the player submits their answer, the app evaluates it for correctness. It compares the player's answer with the correct answer, either by checking if they match directly or by applying the appropriate kidsematical operations.

7. Providing Feedback: The app provides immediate feedback to the player regarding the correctness of their answer. It may display a notification or change the color of the answer field to indicate correctness or incorrectness.

8. Scoring and Progress Tracking: The app keeps track of the player's score based on their correct and incorrect answers. It may display the current score and update it with each question.

9. Next Question: Once the player receives feedback on their answer, the app proceeds to the next question. It repeats the question generation, display, user input, and evaluation steps for each subsequent question.

10. Game Completion and Results: The app keeps track of the total number of questions answered, calculates the final score, and determines the player's performance. It may display the final score, a summary of the player's answers, and any achievements or rewards they have earned.

11. Additional Features: The app may include additional features such as leader boards to compare scores with other players, the ability to customize settings like the number of questions or time limits, or options to review answers or explanations after completing the quiz.

Overall, a kids quiz game app aims to engage users in an interactive and challenging kids quiz experience while providing feedback and tracking their progress. The specific implementation details may vary depending on the design choices and features of the app.

CHAPTER 3

SYSTEM REQUIREMENTS

Hardware Requirements

The hardware requirements are very minimal and the software can be made to run on most of the machines

➢ 4 GB RAM minimum (8 GB RAM recommended)

➢ 2 GB of available disk space minimum, 4 GB recommended

➢ 500 MB for IDE plus 1.5 GB for Android SDK and emulator system image

Software Requirements

➢ Operating System: Windows- 10/8/7(64 Bit)

➢ IDE: JDK and Android Studio -4.1

➢Emulator:AVD Emulator 2.

Development Platform

➢ WINDOWS 10

Development tool

➢ Android Studio

Language Used In Coding

➢ JAVA

CHAPTER 4

DESIGN

Existing System

In this fast moving world kids, I mean future citizens of our country are addicted to the technology and even we don't have any kind of time to spend with them and make them learn some basics. We ourselves fill kids as a trickiest subject. If the basics are good we can do wonders in kids. This app helps us to get rid of all the easiest basics in kids sematics for kids. This app consists of simple kids problems within 30 seconds we ourselves know how speed we are in solving the problems.

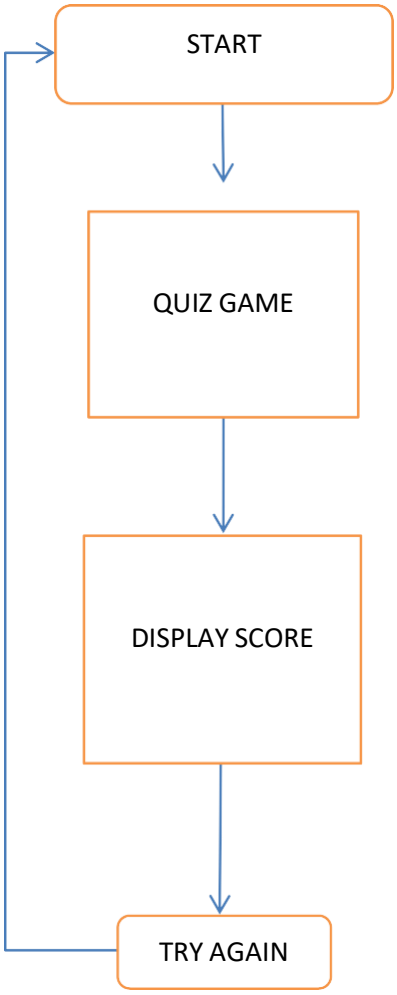
Proposed System

Then the user needs to install the application in his android device and give the necessary permissions to the application. Once the application is instead it is ready to be used. The user needs to focus on the game panel which opens with the title of the game application on the mobile screen. This block of text upon loading for a moment it shows or appears the game started panel in the mobile screen which contains some buttons for some instructions which are present in top and bottom of the start game screen panel. This owes the user to start the game by making their own high scores in the game.

Low Level Design

Low Level Design describes detailed description of each and every module means it includes actual logic for every system component and it goes deep into each modules specification. It is also known as micro level/detailed design.

The kids quiz game is classified into 4 fragments



1. START: This fragment helps to begin with the game which takes to another fragment to quiz game.

2. QUIZ GAME: This fragment generate random basic addition maths questions with four options where we have to click on one of them.

3. DISPLAY SCORE: This fragment displays the score of the player ,played for a instance. It calculates the score within a threshold time .

4. TRY AGAIN: This fragment redirects again to the start menu which results to play again as a new instance.

User Defined Functions

JAVA Main Function:

```
package com.astro.kids_quiz;

import androidx.appcompat.app.AppCompatActivity;

import androidx.constraintlayout.widget.ConstraintLayout;

import android.annotation.SuppressLint;

import android.os.Bundle;

import android.os.CountDownTimer;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import java.util.ArrayList;

import java.util.Random;


public class MainActivity extends AppCompatActivity {

    Button startBtn;

    TextView TimeTextView;

    TextView ScoreTextView;

    TextView AlertTextView;

    TextView QuestionTextView;

    TextView FinalScoreTextView;

    Button button0;

    Button button1;

    Button button2;
```

```
Button button3;
CountDownTimer countDownTimer;
ConstraintLayout constraintLayout;
ConstraintLayout  lastLayout;
Button buttonPlayAgain;

Random random =new Random();
int a;
int b;
int indexOfCorrectAnswer;
ArrayList<Integer> answers = new ArrayList<Integer>();
int points = 0;
int totalQuestions = 0;
@SuppressLint("SetTextI18n")
@Override
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);
   startBtn=(Button)findViewById(R.id.btnStart);

   TimeTextView = findViewById(R.id.TimeTextView);
   ScoreTextView = findViewById(R.id.ScoreTextView);
   FinalScoreTextView=findViewById(R.id.FinalscoreTextView);
   AlertTextView = findViewById(R.id.AlertTextView);
   QuestionTextView =findViewById(R.id.QuestionTextView);
```

```
    button0 = findViewById(R.id.button0);
    button1 = findViewById(R.id.button1);
    button2 = findViewById(R.id.button2);
    button3 = findViewById(R.id.button3);
    buttonPlayAgain=findViewById(R.id.buttonPlayAgain);

    constraintLayout=findViewById(R.id.quizUi);
    lastLayout=findViewById(R.id.lastUi);

    lastLayout.setVisibility(View.INVISIBLE);
    constraintLayout.setVisibility(View.INVISIBLE);


}

@SuppressLint("SetTextI18n")
public void NextQuestion(){
    a = random.nextInt(10);
    b = random.nextInt(10);
    QuestionTextView.setText(a+"+"+b);

    indexOfCorrectAnswer = random.nextInt(4);
    answers.clear();
    for(int i = 0; i<4; i++){

        if(indexOfCorrectAnswer == i){
```

```
            answers.add(a+b);
        }else {
            int wrongAnswer = random.nextInt(20);
            while(wrongAnswer==a+b){

                wrongAnswer = random.nextInt(20);
            }
            answers.add(wrongAnswer);
        }
    }

    button0.setText(Integer.toString(answers.get(0)));
    button1.setText(Integer.toString(answers.get(1)));
    button2.setText(Integer.toString(answers.get(2)));
    button3.setText(Integer.toString(answers.get(3)));

}

public void optionSelect(View view){
    totalQuestions++;
    if(Integer.toString(indexOfCorrectAnswer).equals(view.getTag().toString())){
        points++;
        AlertTextView.setText("Correct");

    }else {
        AlertTextView.setText("Wrong");
```

```
    }

    ScoreTextView.setText(Integer.toString(points)+"/"+Integer.toString(totalQuestions));

    NextQuestion();
}

public void playAgain(View view){
    points=0;
    totalQuestions=0;
    ScoreTextView.setText(Integer.toString(points)+"/"+Integer.toString(totalQuestions));
    countDownTimer.start();
    lastLayout.setVisibility(View.INVISIBLE);
    constraintLayout.setVisibility(View.VISIBLE);

}

public void start(View view) {
    startBtn.setVisibility(View.INVISIBLE);
    constraintLayout.setVisibility(View.VISIBLE);
    NextQuestion();
    countDownTimer = new CountDownTimer(30000,1000) {
        @Override
        public void onTick(long millisUntilFinished) {
            TimeTextView.setText(String.valueOf(millisUntilFinished/1000)+"s");
        }
```

```
        @Override
        public void onFinish() {
           TimeTextView.setText("Time Up!");

FinalScoreTextView.setText(Integer.toString(points)+"/"+Integer.toString(totalQuestions));
           constraintLayout.setVisibility(View.INVISIBLE);
           lastLayout.setVisibility(View.VISIBLE);


        }
      }.start();



   }
}
```

CHAPTER 5

IMPLEMENTATION

Functio ns

Functions are "self contained" modules of code that accomplish a specific task. Functions usualy "take in" data, process it, and "return" a result. Once a function is written, it can be used over and over and over again. Functions can be "called" from the inside of other functions.

In kids quiz game the functions are:

In XML:

1. activity_main.xml :- it corresponds to the View classes and subclasses, such as those for widgets and layouts for the main design of game look which consist of toolbar , tab item , view pager , Relative layout, text view etc

2. ic_launcher_background.xml :- It corresponds to the background of the buttons and icons which we interact inside the application.

Functions Used In JAVA

With the Create New Class dialog and file templates, Android Studio helps you to quickly create the following new classes and types:

- Java classes
- Enumeration and singleton classes
- Interface and annotation types

After you fill in the Create New Class dialog fields and click OK, Android Studio creates a .java file containing skeleton code, including a package statement, any necessary imports, a header, and a class or type declaration. Next, you can add your code to this file.

In kid quiz game the functions are:

MainActivity.java:- this class consists of the background working of acitvity_main.xml.

Import Libraries

The import process prompts you to migrate anylibrary and project dependencies to Android Studio, and add the dependency declarations to the build. gradle file.

Some of the important import libraries are listed below:-

1. import androidx.appcompat.app.AppCompatActivity;

2. import androidx.constraintlayout.widget.ConstraintLayout;

3. import android.annotation.SuppressLint;

4. import android.os.Bundle;

5. import android.os.CountDownTimer;

6. import android.view.View;

7. import android.widget.Button;

8. import android.widget.TextView;

9. import java.util.ArrayList;

10. import java.util.Random;

Main Function

There are two main function in kids quiz game, there are :-

- **activity_main.xml**
- **MainActivity.java**

1. activity_main.xml

The activity_main.xml is a layout file available in the res/layout directory, that is referenced by your application when building its interface. You will modify this file very frequently to change the layout of your application. This XML layout file resides in activity_my. xml (activity_main) , and contains some settings and Textview(other) elements. So the activity_main. xml determines how the Activity (Main Activity in this case) should look.

activity_main. xml: This file is located in the res/layout folder. It defines what components to display and how to display them in the referenced activity view. Double click this file, you can see the content in the right panel, it has a Design view and a Text view, you can click the bottom Design and Text tab to switch.

```
<RelativeLayout xmlns:android=http://schemas.android.com/apk/res/android”
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width = "match_parent"
    android:layout_height = "match_parent"
    tools:context=  ".MainActivity" >
```

2. MainActivity.java

An activity provides the window in which the app draws its UI. Typical y, one activity in an app is specified as the main activity, which is the first screen to appear when the user launches the app. Each activity can then start another activity in order to perform different actions.

app > java > package name > MainActivity.java

Knowing how to find and open up files like this is crucial to understanding Android Studio.

```
public class MainActivity extends AppCompatActivity {

    Button startBtn;

    TextView TimeTextView;

    TextView ScoreTextView;

    TextView AlertTextView;

    TextView QuestionTextView;

    TextView FinalScoreTextView;

    Button button0;

    Button button1;

    Button button2;

    Button button3;

    CountDownTimer countDownTimer;

    ConstraintLayout constraintLayout;

    ConstraintLayout lastLayout;

    Button buttonPlayAgain;

    Random random =new Random();

    int a;

    int b;

    int indexOfCorrectAnswer;
```

```
ArrayList<Integer> answers = new ArrayList<Integer>();
int points = 0;
int totalQuestions = 0;


@SuppressLint("SetTextI18n")
@Override
```

Steps for installation or to run the Application

Step 1: Head over to the link to get the Android Studio executable or zip file.

Figure 5.1 Step 1 of Installation process

Step 2: Click on the Download Android Studio Button.Click on the "I have read and agree with above terms and conditions" checkbox followed by the download button.

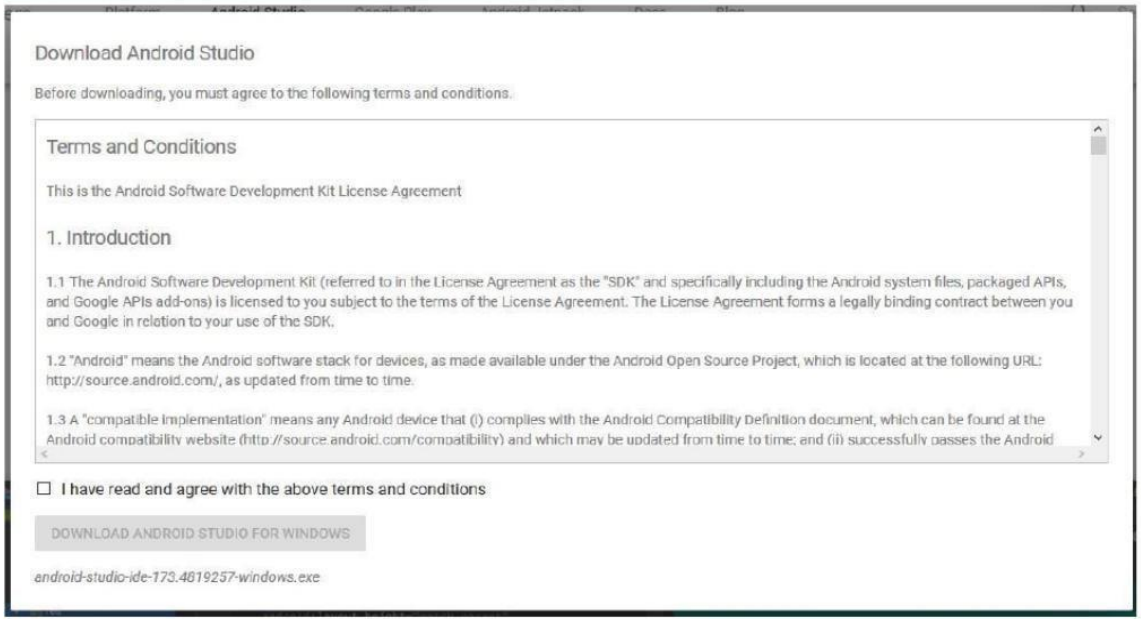


Figure 5.2 Step 2 of Installation process

Step 3: After the downloading has finished, open the file from downloads and run it. It wil prompt the following dialog box.

Figure 5.3 Step 3 of Installation process

Step 4: It will start the instalation, and once it is completed, it will be like the image shown below

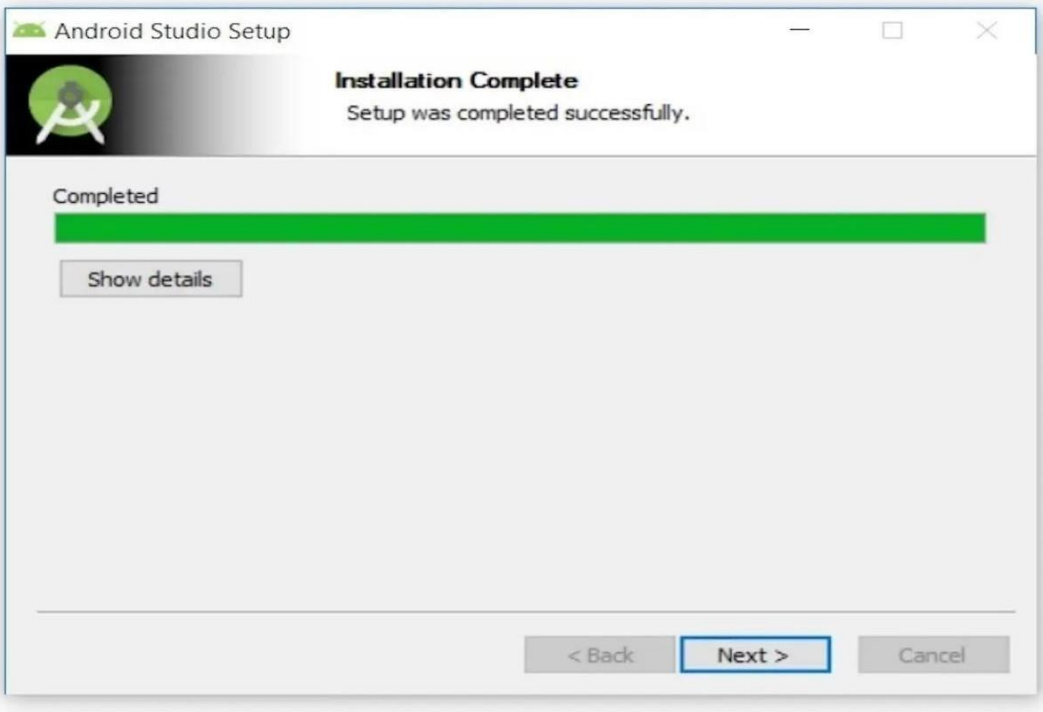


Figure 5.4 Step 4 of Installation process

Step 5: Once "Finish" is clicked, it wil ask whether the previous settings need to be imported [if the android studio had been instal ed earlier], or not. It is better to choose the 'Don't import Settings option'.

Figure 5.5 Step 5 of Installation process

Step 6: This will start the Android Studio.Meanwhile, it wil be finding the available SDK components

Figure 5.6 Step 6 of Installation process

Step 7: After it has found the SDK components, it wil redirect to the Welcome dialog box.

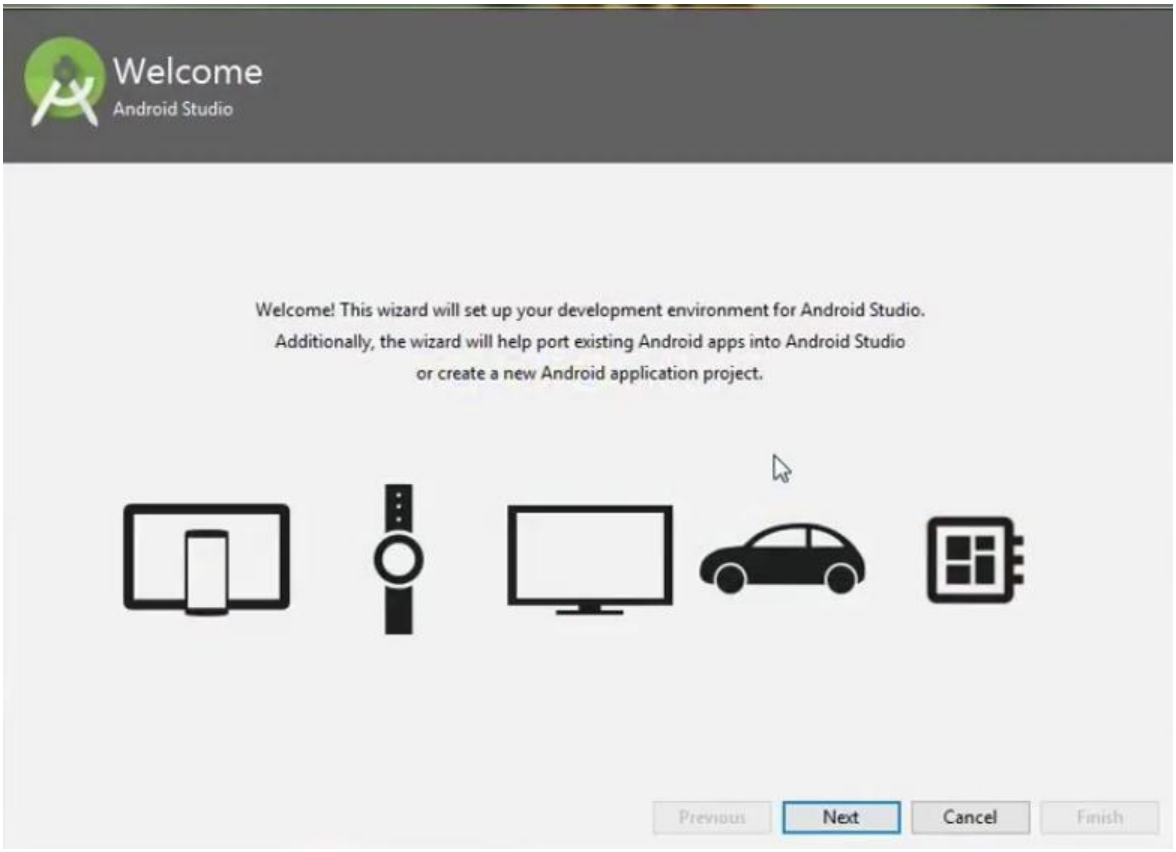


Figure 5.7 Step 7 of Installation process

Step 8: Now it is time to download the SDK components. Click on Finish. Components begin to download and let it complete

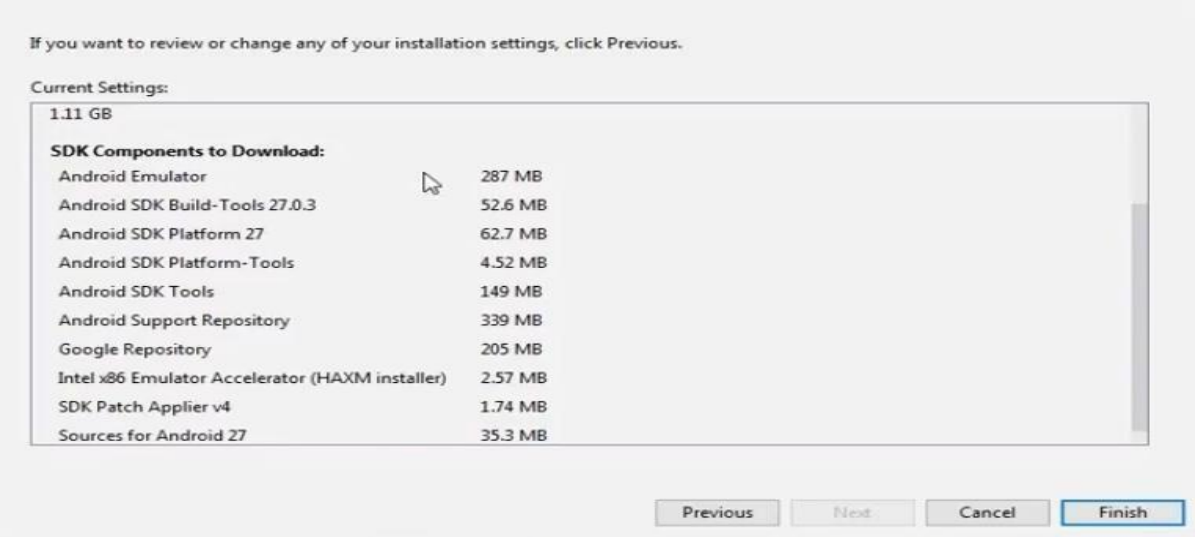


Figure 5.8 Step 8 of Installation process

Step 9: Click on Start a new Android Studio project to build a new app.

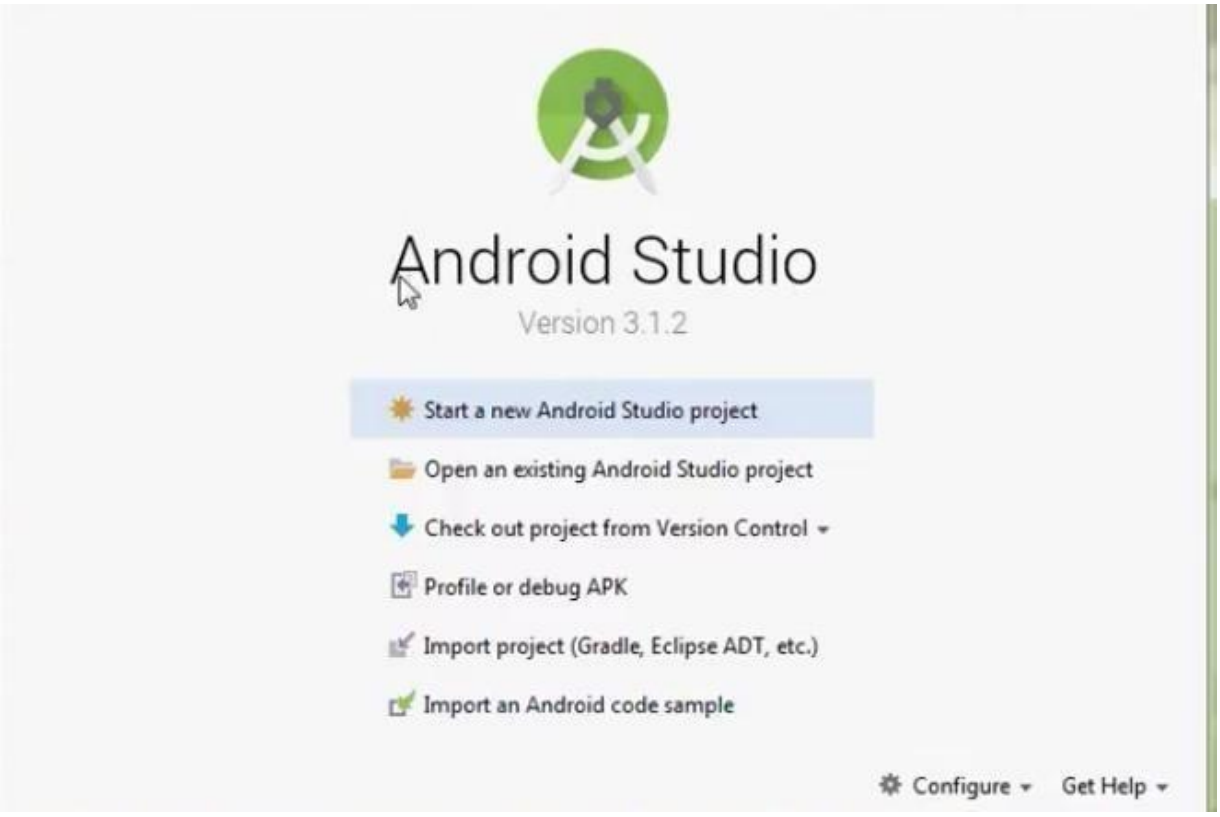
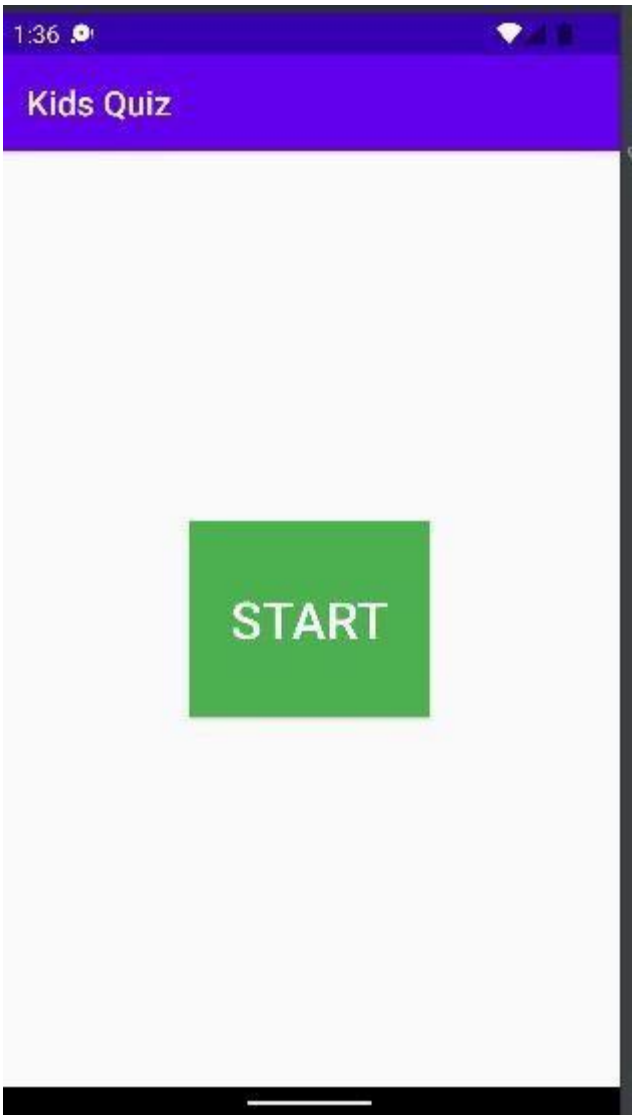


Figure 5.9 Step 9 of Installation process

CHAPTER 6



SNAPSHOTS

Figure 6.1 HOME PAGE: START

Figure 6.2 QUIZ

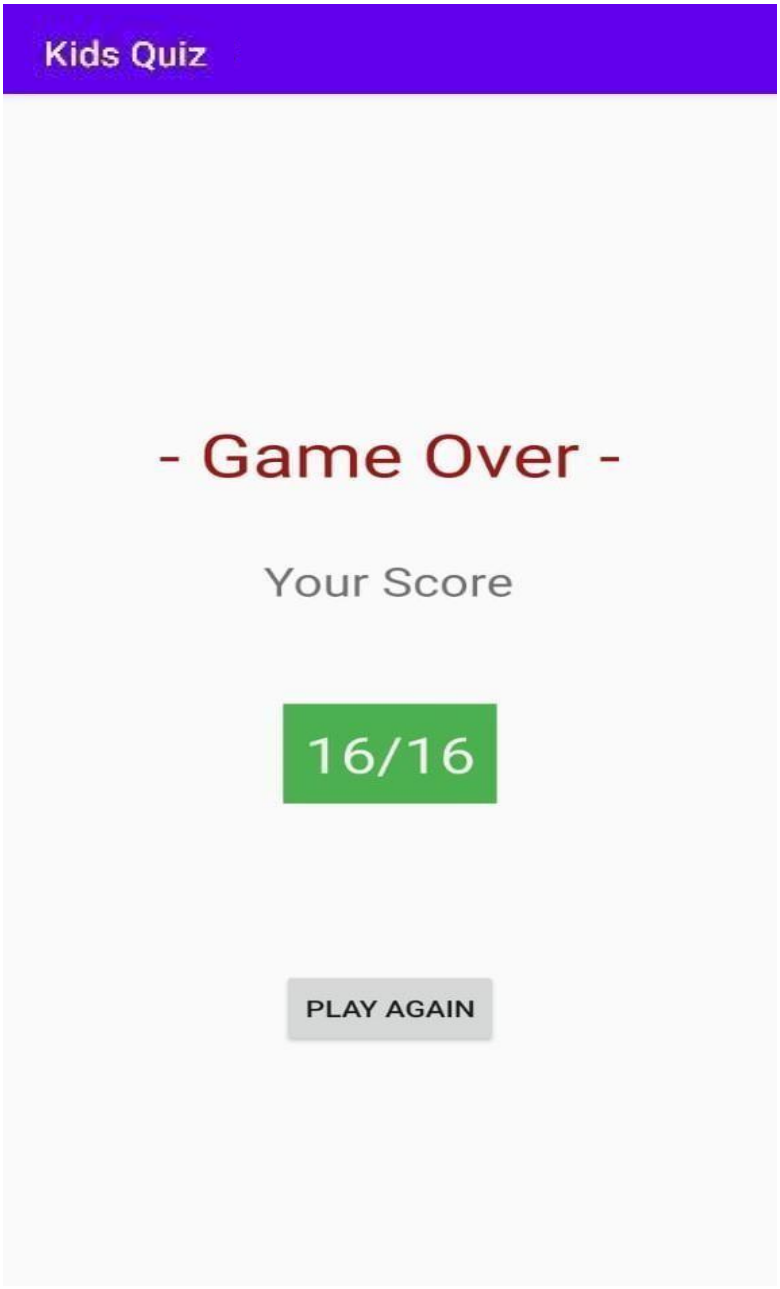


Figure 6.2 DISPLAY SCORE

CONCLUSION

In conclusion, the basic addition kids quiz game is a fun and educational tool for children to practice and improve their addition skills. By engaging in a quiz format, children are encouraged to actively participate and think critically to solve addition problems. This game helps them develop essential math abilities such as mental calculation, number recognition, and problem-solving. Additionally, the game fosters a positive learning environment by incorporating elements of play and competition. Overall, the basic addition kids quiz game is an effective and enjoyable resource that can enhance children's mathematical proficiency and confidence in a fun and interactive way.

FUTURE ENHANCEMENTS

1. Difficulty levels: Introduce multiple difficulty levels to cater to different age groups and skill levels. This would allow children to progress gradually from simple addition problems to more complex ones, ensuring a continuous challenge and promoting growth.

2. Interactive visuals: Enhance the game's visual elements with appealing graphics, animations, and interactive features. Engaging visuals can capture children's attention, make learning more enjoyable, and provide visual cues to aid in understanding addition concepts.

3. Multiplayer mode: Add a multiplayer mode that allows children to compete against their friends or classmates in real-time addition challenges. This fosters a sense of healthy competition and promotes social interaction while reinforcing their addition skills.

4. Multiplayer mode: Add a multiplayer mode that allows children to compete against their friends or classmates in real-time addition challenges. This fosters a sense of healthy competition and promotes social interaction while reinforcing their addition skills.

5. Customizable options: Allow customization of the game settings, such as selecting specific number ranges or focusing on specific addition concepts (e.g., carrying over). This flexibility enables parents and teachers to tailor the game to suit individual learning needs and target specific areas for improvement.

REFERENCES

[1] For Theory Concepts

Google Developer Training, "Android Developer Fundamentals Course - Concept Reference", Google Developer Training Team, 2017

[2] For Concept & details of working

https://developer.android.com/studio

[3] For Programming

https://www.tutorialspoint.com/android/android_studio.htm

[4] For Installation of Android Studio

https://www.geeksforgeeks.org/guide-to-install-and-set-up-android-studio

[5] For Emulator Details

https://docs.expo.dev/workflow/android-studio-emulator