

Table of Contents

CONTENTS	Page No.
ACKNOWLEDGEMENT	I
ABSTRACT	II
1. INTRODUCTION	1
1.1. Overview	1
1.2. Problem Statement	1
1.3. Mobile Application Development Need & Importance	1
1.4. Android Studio	4
2. SYSTEM REQUIREMENTS	7
2.1. Hardware and Software Requirements	7
3. SYSTEM DESIGN	8
3.1. XML Design	8
3.2. XML Code	10
4. IMPLEMENTATION	20
4.1. Description	20
4.2. Java Code	24
5. RESULTS	34
6. CONCLUSION AND FUTURE ENHANCEMENTS	37

List of Figures

Figure No.	Figure Name	Page No.
Figure 1.4.2	The Android Studio main window	5
Figure 3.1	XML design of MainActivity	8
Figure 3.2	XML design of PuzzleActivity	9
Figure 3.3	XML design of ScoreActivity	10
Figure 5.1	Homescreen	34
Figure 5.2	Game page	35
Figure 5.3	Result page	36

Chapter 1

INTRODUCTION

1.1 Overview

With the advent of new mobile technologies, the mobile application industry is advancing rapidly. Consisting of several operating systems like symbian OS, iOS, blackberry, etc., Android OS is recognized as the most widely used, popular and user-friendly mobile platform. This open-source linux kernel-based operating system offers high flexibility due to its customization properties making it a dominant mobile operating system. Android applications are programmed in java language. Google android SDK delivers a special software stack that provides developers an easy platform to develop android applications. Moreover, developers can make use of existing java IDEs which provides flexibility to the developers. Java libraries are predominant in the process of third-party application development. Cross-platform approaches make sure that developers do not have to develop platform-dependent applications. With the help of these approaches, an application can be deployed to several platforms without the need for changes in coding.

1.2 Problem Statement

The aim of this application is to show to build a word scrambler game where the duration of each trial is 5 minutes giving a maximum of 30 seconds to the user to guess the correct order of the word. The application will be implemented using the Java language. The objective of the application is to focus mainly on the concepts of Threads.

1.3 Mobile Application Development Need & Importance

In the past few years mobile app development has become a booming industry. Currently, it is estimated that there are 2.3 million mobile app developers who are devoted to keeping up with the industry demand.

In fact, according to Apple, in 2013 1.25 million apps were registered in the Apple app store and accounted for 50 billion downloads and \$5 billion paid to developers.

With these types of industry numbers, it soon becomes clear that mobile app development is a key factor for business success.

With the growing number of people accessing the Internet via smartphones and tablets, mobile app development has the unique ability to access a large number of potential consumers.

Not only have the sales of smartphone and tablets increased, but the amount of mobile apps installed has also grown exponentially. The PewResearch Internet Project indicates that approximately 50 percent of all smartphone users have mobile apps installed; of this percentage, two-thirds of the individuals are regular mobile app users. These statistics show that mobile apps have a unique opportunity to engage with an entirely new type of customer, one whom is constantly connected to the Internet and the global commerce space. In essence, a mobile app allows you to have millions of new customers at your fingertips. All that is left for you to do, is to develop an effective app and reap the benefits of your labors.

There are multiple benefits to creating and distributing a mobile app. Below are a few of the top benefits for businesses across a wide-variety of industries.

- **Build Loyalty**

Mobile apps work to consistently increase customer loyalty, especially in the retail sector.

- **Reinforce your Brand**

Mobile apps offer the unique opportunity for brand reinforcement through a new channel. Through mobile apps, customers are encouraged to download the free branded version, where they can customize preferences to fit their specific needs.

- **Increase your Accessibility**

Smartphone and tablet users are constantly on the go; this means that they don't always have time to sign into a mobile website. And these mobile websites are designed for readability and navigation, NOT for process management. Mobile apps allow users to have easy, functional access to information, products, services and processes that they need in real-time and are optimized for hands on interaction.

- **Increase Sell-through**

Recent analysis suggests that mobile app users spend more time on a company's mobile app, then they spend on the company's mobile website.

- **Reduce On-Premise Costs**

Most of the services that you provide at your business premises can be provided through android mobile applications. This would put you in a position where you do not need to pay workers to do that particular job.

- **Scope for Innovation**

With every year, Android brings up innovative ideas and trends that symbolize the future. The devices and technologies used by users to interact with business changes pertaining to users' behaviors and needs.

As we continue to evolve into a mobile-centric society, it comes as no surprise that mobile apps are at the center of the developmental push. Developing a mobile app can go a long way towards propelling your company into the hands of new customers and future business success.

1.4 Android Studio

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

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

1.4.1 Project structure

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

- Android app modules
- Library modules
- Google App Engine modules

Each app module contains the following folders:

- **manifests:** Contains the AndroidManifest.xml file.
- **java:** Contains the Java source code files, including JUnit test code.

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

1.4.2 The user interface

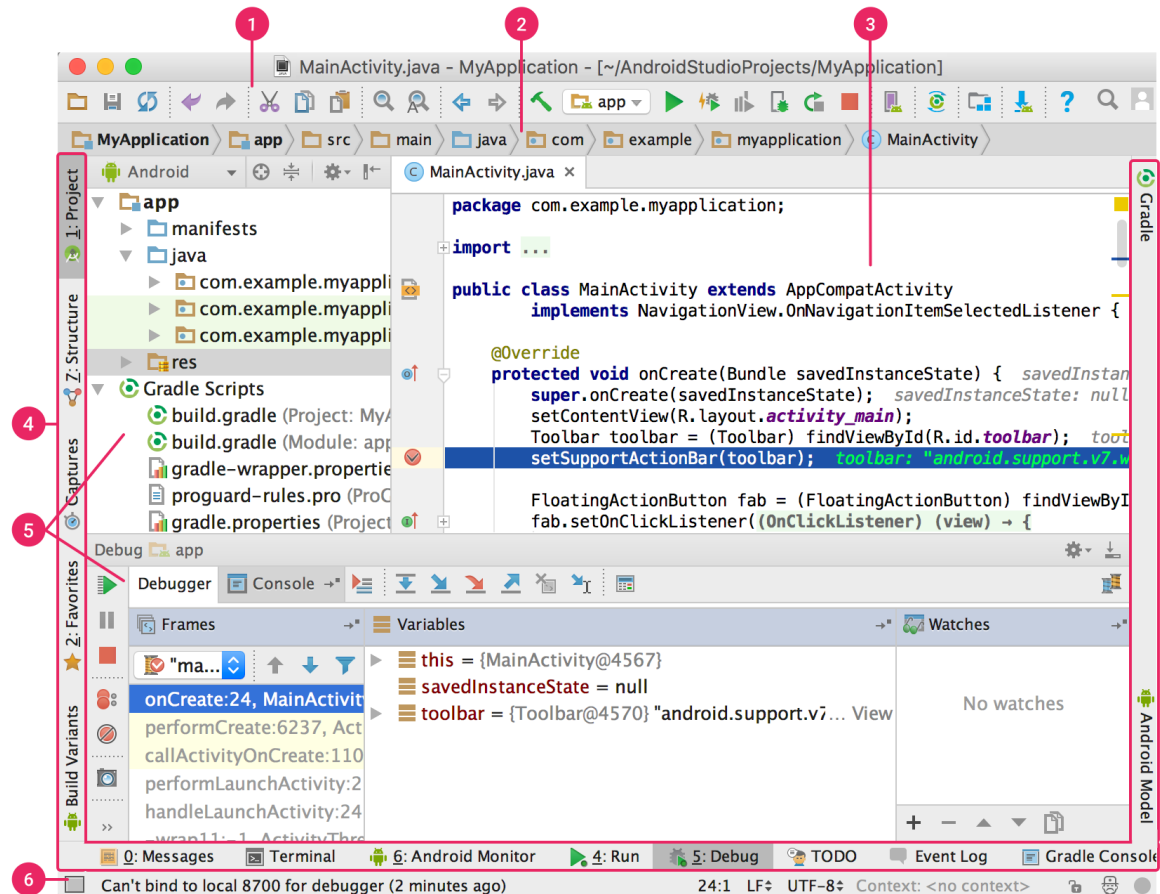


Figure 1.4.2. The Android Studio main window.

- **The toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
- **The navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the Project window.
- **The editor window** is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.

- **The tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.
- **The tool windows** give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.
- **The status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.

1.4.4 Gradle build system

Android Studio uses Gradle as the foundation of the build system, with more Android-specific capabilities provided by the Android plugin for Gradle. This build system runs as an integrated tool from the Android Studio menu, and independently from the command line. You can use the features of the build system to do the following:

- Customize, configure, and extend the build process.
- Create multiple APKs for your app, with different features using the same project and modules.
- Reuse code and resources across sourcesets.

By employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files. Android Studio build files are named `build.gradle`. They are plain text files that use Groovy syntax to configure the build with elements provided by the Android plugin for Gradle. Each project has one top-level build file for the entire project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

Chapter 2

SYSTEM REQUIREMENTS

2.1 Software Requirements

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application.

The following are the software requirements for the application:

- Operating System: Windows 10
- Java Development Kit
- Android Studio

2.2 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware.

- CPU: Intel or AMD processor
- Cores: Dual-Core (Quad-Core recommended)
- RAM: minimum 4GB (>4GB recommended)
- Graphics: Intel Integrated Graphics or AMD Equivalent
- Display Resolution: 1366x768 (1920x1080 recommended).

Chapter 3

SYSTEM DESIGN

3.1 XML Design



Figure 3.1 XML design of MainActivity

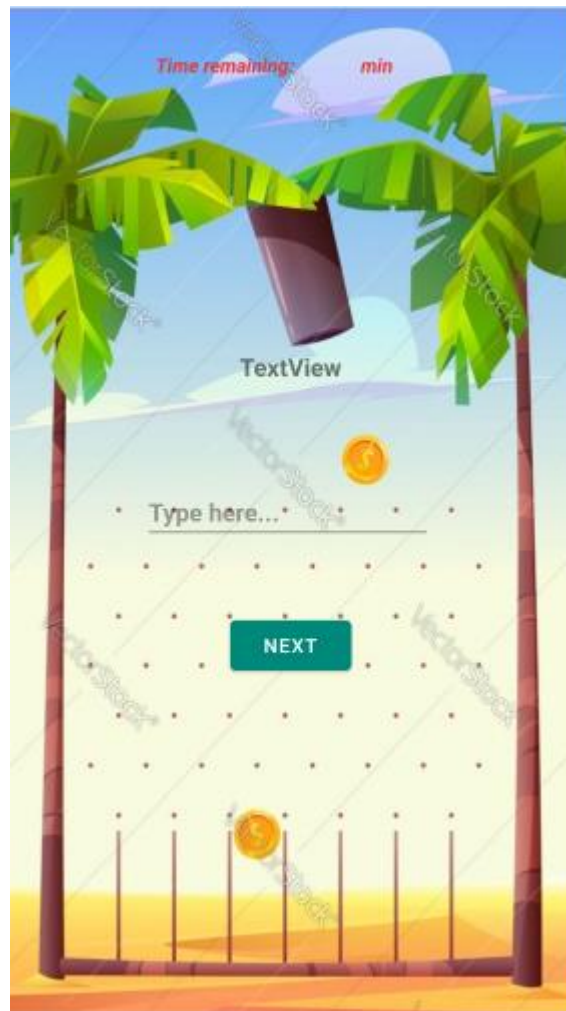


Figure 3.2 XML design of PuzzleActivity

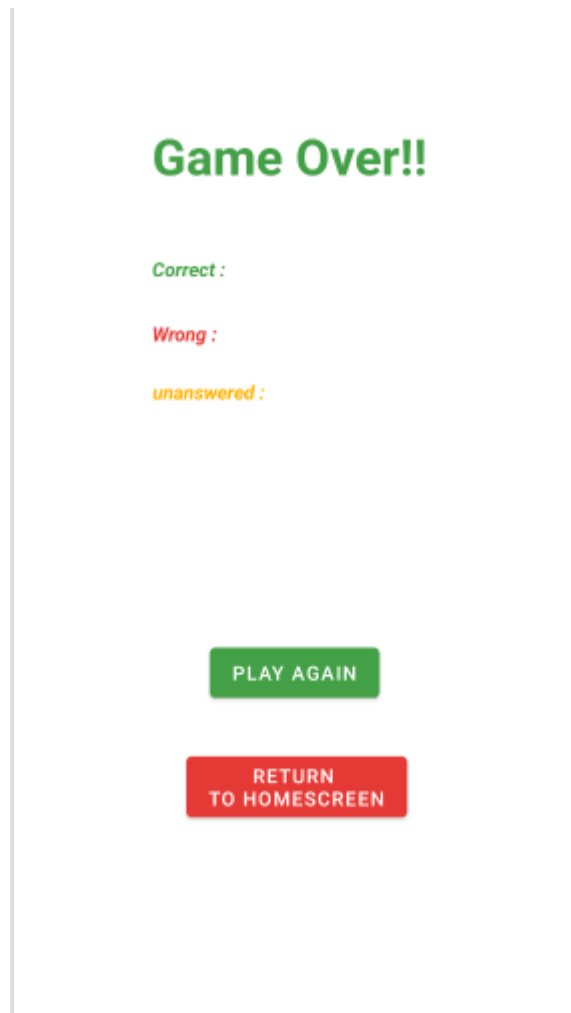


Figure 3.3 XML design of ScoreActivity

3.2 XML Code

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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"
android:background="@drawable/bg"
tools:context=".MainActivity">
    <Button
        android:id="@+id/play"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="play"
        app:backgroundTint="#43A047"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.576" />
    <Button
        android:id="@+id/exit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:shadowColor="#FDD835"
```

```
        android:text="exit"
        app:backgroundTint="#E53935"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.711" />
<TextView
    android:id="@+id/textView3"
    android:layout_width="332dp"
    android:layout_height="60dp"
    android:text="You have 5 minutes, guess as many words as possible!"
    android:textColor="#8E24AA"
    android:textSize="16sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.632"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.427" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_puzzle.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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"
android:background="@drawable/bg"
tools:context=".PuzzleActivity">
<EditText
    android:id="@+id/answer"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Type here..."
    android:inputType="textPersonName"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.487"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<TextView
    android:id="@+id/textremtime"
    android:layout_width="102dp"
    android:layout_height="40dp"
    android:text="Time remaining: "
    android:textColor="#E53935"
    android:textStyle="bold|italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.346"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.046" />
```

```
<TextView
    android:id="@+id/question"
    android:layout_width="214dp"
    android:layout_height="59dp"
    android:gravity="center_horizontal"
    android:text="TextView"
    android:textAlignment="center"
    android:textSize="18sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.369" />
```

```
<TextView
    android:id="@+id/counter"
    android:layout_width="71dp"
    android:layout_height="55dp"
    android:gravity="center_horizontal"
    android:textAlignment="center"
    android:textColor="#D81B60"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.523"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.247" />
```

```
<Button
```



```
    android:id="@+id/next"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="next"
    app:backgroundTint="#00897B"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.641" />
<TextView
    android:id="@+id/remtime"
    android:layout_width="37dp"
    android:layout_height="33dp"
    android:gravity="center_horizontal"
    android:textAlignment="center"
    android:textColor="#E53935"
    android:textStyle="bold|italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.566"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.045" />
<TextView
    android:id="@+id/remtime2"
    android:layout_width="59dp"
    android:layout_height="27dp"
    android:text="min"
    android:textColor="#E53935"
```

```
        android:textStyle="bold|italic"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.727"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.045" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_score.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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=".ScoreActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif"
        android:text="Game Over!!"
        android:textColor="#43A047"
        android:textColorHighlight="#FFFFFF"
        android:textSize="36sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
```

```
    app:layout_constraintHorizontal_bias="0.473"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.121" />
<TextView
    android:id="@+id/correct"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Correct : "
    android:textColor="#43A047"
    android:textStyle="bold|italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.282"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.252" />
<TextView
    android:id="@+id/wrong"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Wrong : "
    android:textColor="#E53935"
    android:textStyle="bold|italic"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.277"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.318" />
<TextView
```

```
android:id="@+id/unanswered"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="unanswered : "
android:textColor="#FFB300"
android:textStyle="bold|italic"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.308"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.377" />
```

<TextView

```
android:id="@+id/result"
android:layout_width="365dp"
android:layout_height="79dp"
android:textAlignment="center"
android:textColor="#D81B60"
android:textSize="24sp"
android:textStyle="bold|italic"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.347"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.524" />
```

<Button

```
android:id="@+id/replay"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="play again"
```

```
    app:backgroundTint="#43A047"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.493"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.67" />
<Button
    android:id="@+id/exitscore"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Return \nto homescreen"
    app:backgroundTint="#E53935"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.794" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Chapter 4

IMPLEMENTATION

4.1 Description

- **Void onCreate (Bundle)**

The **entire lifetime** of an activity happens between the first call to onCreate(Bundle) and onDestroy(). An activity will do all setup of "global" state in onCreate(), and release all remaining resources in onDestroy().

- **public static interface View.OnClickListener**

Interface definition for a callback to be invoked when a view is clicked.

- **public class Intent**

An Intent provides a facility for performing late runtime binding between the code in different applications. Its most significant use is in the launching of activities, where it can be thought of as the glue between activities.

- **Toast.makeText (context, text, duration);**

A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. Toasts automatically disappear after a timeout. Use the makeText() method, which takes the following parameters:

The application Context, The text that should appear to the user, The duration that the toast should remain on the screen. The makeText() method returns a properly initialized Toast object.

- **Public class Thread extends Object implements Runnable**

A *thread* is a thread of execution in a program. Way to create a thread is to declare a class that implements the Runnable interface. That class then implements

the run method. An instance of the class can then be allocated, passed as an argument when creating Thread, and started.

- **OnBackPressed ()**

This allows Android to properly navigate to previous destinations when the Back button is pressed.

- **Public Bundle ()**

Constructs a new, empty Bundle. Android Bundle is used to pass data between activities. The values that are to be passed are mapped to String keys which are later used in the next activity to retrieve the values.

- **Public Handler ()**

A Handler allows you to send and process Message and Runnable objects associated with a thread's MessageQueue .

- **Public Random ()**

Creates a new random number generator.

- **Void run ()**

The run() method of thread class is called if the thread was constructed using a separate Runnable object otherwise this method does nothing and returns. When the run() method calls, the code specified in the run() method is executed.

- **public void finish ()**

When calling finish() on an activity, the method onDestroy() is executed. The ActivityResult is propagated back to whoever launched you via onActivityResult().

4.1.1 User defined functions and classes

- **Class ShuffleString**

```
class ShuffleString {  
    @RequiresApi(api = Build.VERSION_CODES.N)  
    public String shuffleJava8(String text){  
        List<Character> characters = text.chars().mapToObj( c ->
```

```
(char)c).collect(Collectors.toList());

StringBuilder result = new StringBuilder();

IntStream.range(0, text.length()).forEach((index) -> {

    int randomPosition = new Random().nextInt(characters.size());

    result.append(characters.get(randomPosition));

    characters.remove(randomPosition);

});

return result.toString();

}
```

- **Class MyCounter**

```
class MyCounter extends Thread

{

    @RequiresApi(api = Build.VERSION_CODES.N)

    public void run()

    {

        while(run)

        {

            if(counter==0) {

                unanswered++;

                ShuffleString s=new ShuffleString();

                randomPosition=random();

                word=s.shuffleJava8(words[randomPosition]);

                question.setText(word);

                counter=30;

                run=true;

            }

            counter--;

            handler.sendMessage(counter);

            try{
```



```
        Thread.sleep(1000);
    }
    catch(Exception e){}
}
}
}
```

- **Class GlobalCounter**

```
class Globalcounter extends Thread
{
    public void run()
    {
        String m,s,message;
        int flag=0;
        while(globalmincounter!=0)
        {
            if(globalseccounter==0){
                globalmincounter--;
                globalseccounter=59;
            }
            globalseccounter--;
            ghandler.sendEmptyMessage(globalmincounter);
            try
            {
                Thread.sleep(1000);
            }
            catch(Exception e){}
        }
        if(globalmincounter==0)
        {
```

```
        Bundle bundle = new Bundle();
        bundle.putString("correct", String.valueOf(correct));
        bundle.putString("wrong", String.valueOf(wrong));
        bundle.putString("unanswered", String.valueOf(unanswered));
        Intent newActivity;
        newActivity = new Intent(PuzzleActivity.this, ScoreActivity.class);
        newActivity.putExtra("data", bundle);
        startActivity(newActivity);
        finish();
    }
}
```

4.2 Java Code

MainActivity.java

```
package com.example.wordgame;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity implements
View.OnClickListener
{
    private Button play,exit;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

```
        play=(Button)findViewById(R.id.play);
        exit=(Button)findViewById(R.id.exit);
        play.setOnClickListener(this);
        exit.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {
        if(view.equals(play))
        {
            Intent intent = new Intent(MainActivity.this,PuzzleActivity.class);
            startActivity(intent);
        }
        else if(view.equals(exit))
        {
            finish();
        }
    }
}
```

PuzzleActivity.java

```
package com.example.wordgame;
import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
```

```
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;
```

```
import java.util.List;  
import java.util.Random;  
import java.util.stream.Collectors;  
import java.util.stream.IntStream;
```

```
public class PuzzleActivity extends AppCompatActivity implements
```

```
View.OnClickListener
```

```
{
```

```
    public static String[] words =new String[] { "squirrel", "chimpanzee",  
    "rabbit","goldfish","mouse","kitten","parrot","turtle","shrimp","flamingo","se  
    agull","goose","raven","woodpecker","penguin","panda","walrus","koala","don  
    key","kangaroo","elephant","giraffe","sheep","crocodile","buffalo","zebra","rhi  
    noceros","alligator","dinosaur","leopard","cheetah","tiger","lion","snake","tort  
    oise","chicken","monkey","antelope","gorilla","chipmunk","horse","camel","m  
    ongoose","turkey","lobster","pelican","jackal","jaguar","hyena","ostrich","vult  
    ure"};
```

```
    boolean run=true;
```

```
    int counter = 30,correct=0,wrong=0,unanswered=0,randomPosition;
```

```
    int globalmincounter=5;
```

```
    int globalseccounter=0;
```

```
    TextView c,question,remTime;
```

```
    Button next;
```

```
    EditText answer;
```

```
    String word;
```

```
@RequiresApi(api = Build.VERSION_CODES.N)
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_puzzle);
    c = (TextView) findViewById(R.id.counter);
    next=(Button)findViewById(R.id.next);
    next.setOnClickListener(this);
    question=(TextView)findViewById(R.id.question);
    answer=(EditText) findViewById(R.id.answer);
    remTime=(TextView)findViewById(R.id.remtime);
    ShuffleString s=new ShuffleString();
    randomPosition=random();
    word=s.shuffleJava8(words[randomPosition]);
    question.setText(word);
    new MyCounter().start();
    new Globalcounter().start();
}
@Override
public void onBackPressed() {
    super.onBackPressed();
    globalmincounter=0;
    this.finish();
}
public int random()
{
    int r= new Random().nextInt(words.length);
    return r;
}
@RequiresApi(api = Build.VERSION_CODES.N)
```

```
@Override
public void onClick(View view) {
    if(view.equals(next))
    {
        String userInput=answer.getText().toString();
        answer.setText("");
        if(userinput.equals(""))
        {
            Toast.makeText(getBaseContext(), "You Missed!",
Toast.LENGTH_LONG).show();
            unanswered++;
        }
        else if(userinput.toLowerCase().equals(words[randomPosition]))
        {
            Toast.makeText(getBaseContext(), "Correct",
Toast.LENGTH_LONG).show();
            correct++;
        }
        else
        {
            Toast.makeText(getBaseContext(), "Wrong",
Toast.LENGTH_LONG).show();
            wrong++;
        }
        randomPosition=random();
        ShuffleString s=new ShuffleString();
        word=s.shuffleJava8(words[randomPosition]);
        question.setText(word);
        counter=30;
        run=true;
    }
}
```

```
}  
  
class h extends Handler  
{  
    @Override  
    public void handleMessage(@NonNull Message msg) {  
        super.handleMessage(msg);  
        c.setText(String.valueOf(msg.what));  
    }  
}  
  
class globalhandler extends Handler  
{  
    @Override  
    public void handleMessage(@NonNull Message msg) {  
        super.handleMessage(msg);  
        remTime.setText(String.valueOf(msg.what));  
    }  
}  
  
Handler handler= new h();  
Handler ghandler=new globalhandler();  
  
class MyCounter extends Thread  
{  
    @RequiresApi(api = Build.VERSION_CODES.N)  
    public void run()  
    {  
        while(run)  
        {  
            if(counter==0) {  
                unanswered++;  
                ShuffleString s=new ShuffleString();  
                randomPosition=random();  
                word=s.shuffleJava8(words[randomPosition]);  
            }  
        }  
    }  
}
```

```
        catch(Exception e){ }
    }
    if(globalmincounter==0)
    {
        Bundle bundle = new Bundle();
        bundle.putString("correct", String.valueOf(correct));
        bundle.putString("wrong", String.valueOf(wrong));
        bundle.putString("unanswered", String.valueOf(unanswered));
        Intent newActivity;
        newActivity = new Intent(PuzzleActivity.this,ScoreActivity.class);
        newActivity.putExtra("data", bundle);
        startActivity(newActivity);
        finish();
    }
}

class ShuffleString {
    @RequiresApi(api = Build.VERSION_CODES.N)
    public String shuffleJava8(String text){
        List<Character> characters = text.chars().mapToObj( c ->
(char)c).collect(Collectors.toList());
        StringBuilder result = new StringBuilder();

        IntStream.range(0,text.length()).forEach((index) -> {
            int randomPosition = new Random().nextInt(characters.size());
            result.append(characters.get(randomPosition));
            characters.remove(randomPosition);
        });
        return result.toString();
    }
}
```

ScoreActivity.java

```
package com.example.wordgame;
import androidx.appcompat.app.AppCompatActivity;
import java.lang.*;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import static java.lang.Integer.*;
public class ScoreActivity extends AppCompatActivity implements
View.OnClickListener{
    TextView c, w, a,res;
    Button replay,exit2;
    String correct, wrong, unansnswered;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_score);
        c=(TextView)findViewById(R.id.correct);
        w=(TextView)findViewById(R.id.wrong);
        a=(TextView)findViewById(R.id.unanswered);
        res=(TextView)findViewById(R.id.result);
        replay=(Button)findViewById(R.id.replay);
        exit2=(Button)findViewById(R.id.exitscore);
        replay.setOnClickListener(this);
        exit2.setOnClickListener(this);
        Bundle bundle=getIntent().getBundleExtra("data");
        correct=bundle.getString("correct");
        wrong=bundle.getString("wrong");
```

```
        unanswered=bundle.getString("unanswered");
        c.setText(c.getText()+correct);
        w.setText(w.getText()+wrong);
        a.setText(a.getText()+unanswered);
        if(Integer.parseInt(correct)>Integer.parseInt(wrong) &&
Integer.parseInt(correct)>Integer.parseInt(unanswered)){
            res.setText("Congratulations!! :)");
        }
        else{
            res.setText("Better luck next time :(");
        }
    }
    @Override
    public void onClick(View view) {
        if(view.equals(replay)){
            Intent intent = new Intent(ScoreActivity.this,PuzzleActivity.class);
            startActivity(intent);
            finish();
        }
        else if(view.equals(exit2)){
            Intent intent = new Intent(ScoreActivity.this, MainActivity.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            intent.putExtra("EXIT", true);
            startActivity(intent);
            if (getIntent().getBooleanExtra("EXIT", false)) {
                finish();
            }
        }
    }
}
```

Chapter 5

RESULTS

- **The Home page**

Figure 5.1 shows the first page of the app. The starting page of the app displays an instruction stating that the player will get 5 minutes to solve the word puzzle. Then there are two buttons clearly stating play and exit which is press play to continue else press exit to leave the app.



Figure 5.1 Homescreen

- **The Game page**

Figure 5.2 shows the second page which is displayed after when the user presses play button. The jumbled word is displayed the player is given 30 seconds to solve and enter answer depending on the correctness of the result, a toast message stating the same will be displayed and the next question will be displayed and so on.

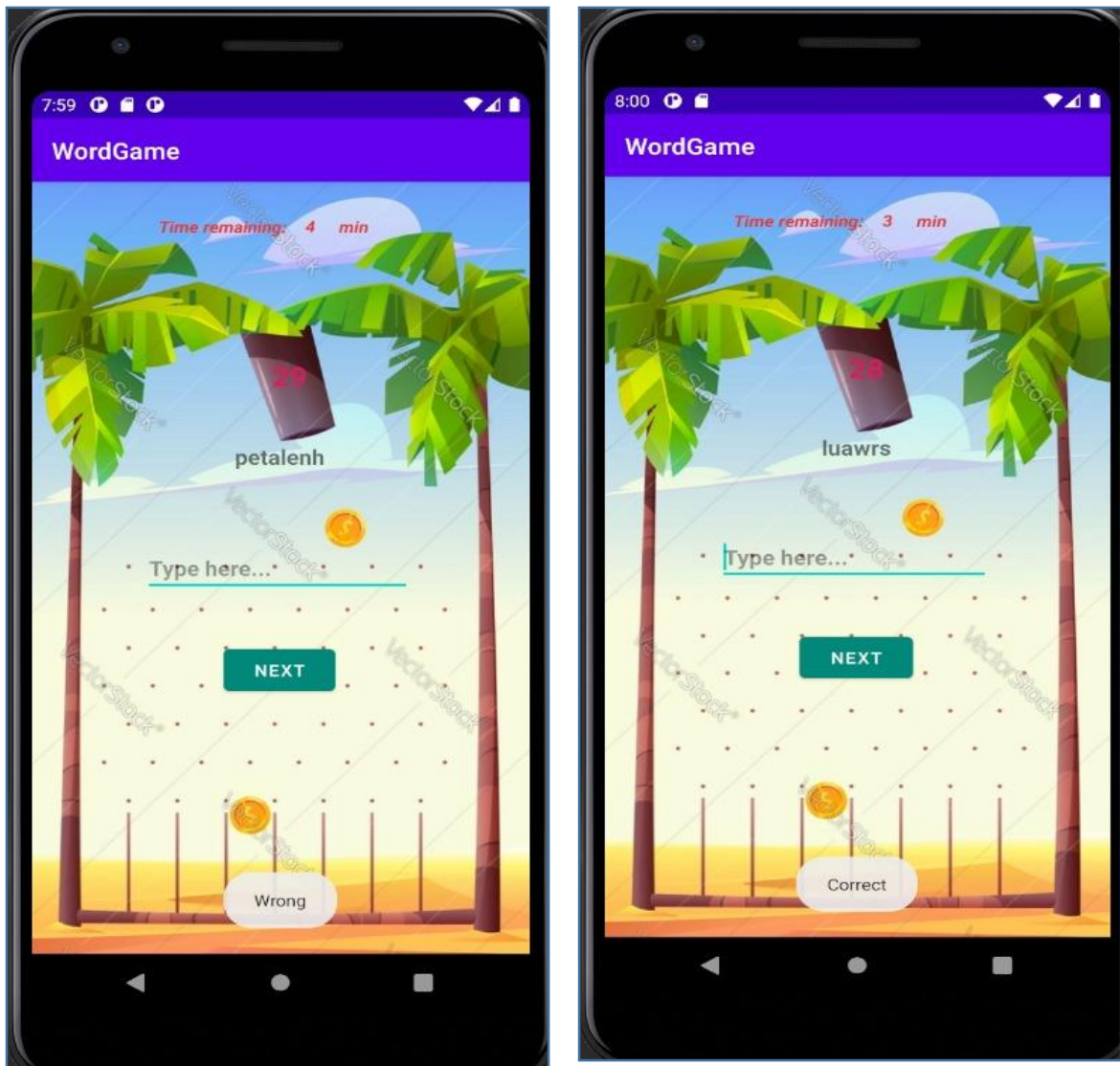


Figure 5.2 Game page

- **The Score page**

Figure 5.3 shows the final page where number of correct and incorrect answers are displayed as well as unanswered questions are displayed too. If the user wants to play again or wants to return to homescreen, he can press the respective button.

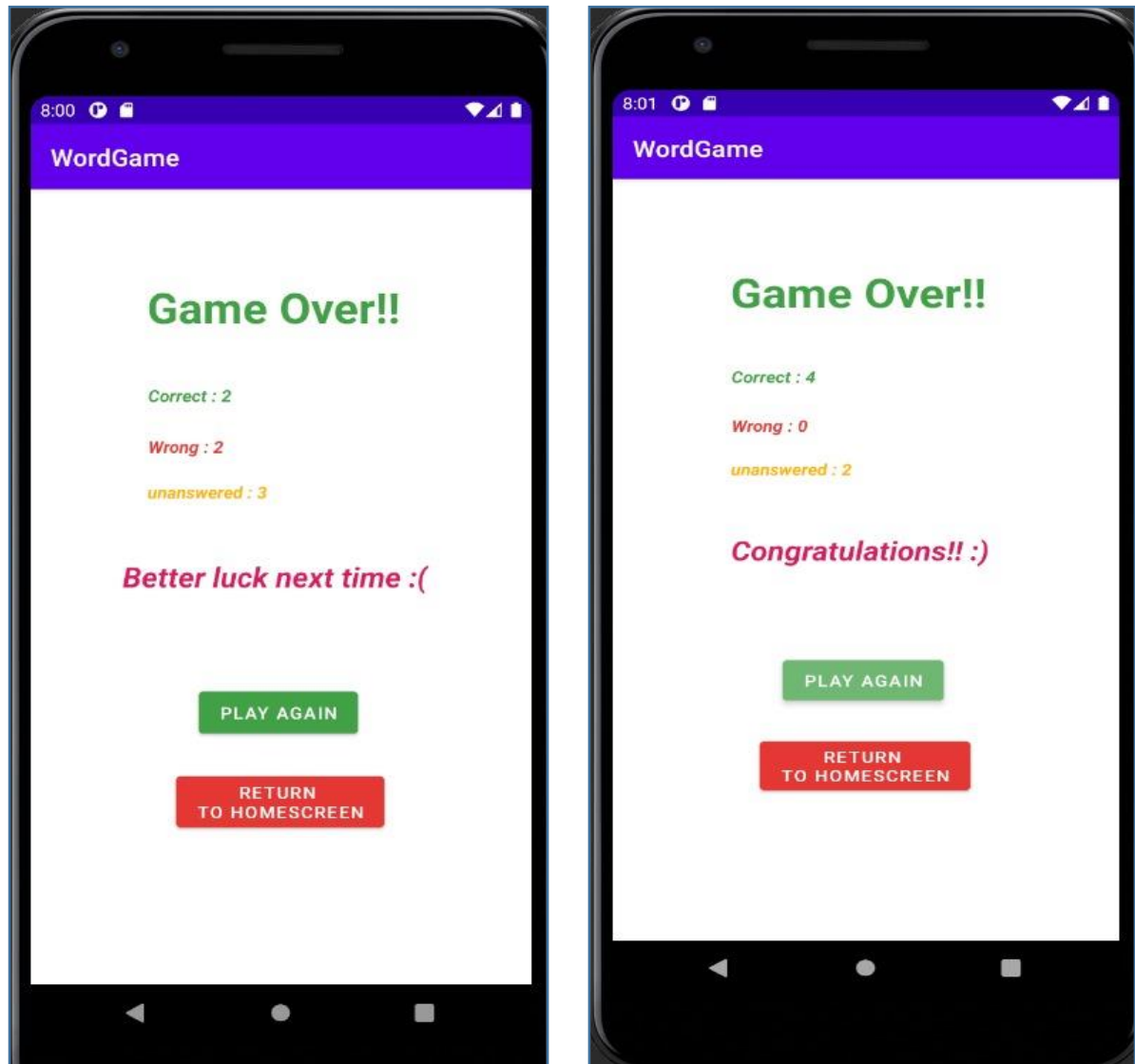


Figure 5.3 Result page

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENTS

It is no secret that as we age, our brain function declines. Gaming apps just aren't meant to freshen up your mood but also makes your brain active. This word game is a simple yet fun game. One's age does not matter to play this game. This is a game which makes you mentally active. Just need to solve the unjumbled letters into meaningful one.

FUTURE ENHANCEMENT:

Definitely in the era of technology there are many scopes of improvements that can be done in this small app (project). The project can be further developed by adding few advance function and user can be given an option to add a player as well.

1. Rewards can be added to encourage player that he/she is doing great and will dwell more interest from user side.
2. As mentioned above we can give option for player to add more players to make it competitive.
3. Game levels can be added and as the player clears up level he/she will get badge.
4. Sound effects would make this game more interesting.
5. In order to make this game more realistic, a word data base can be integrated.