UNIVERSITI TUNKU ABDUL RAHMAN
Faculty of Information and Communication Technology

UCCD3223 Mobile Applications Development
(Jun 2025 Trimester)

Individual Practical Assignment

Name	**Soo Guo Sheng**
Student ID	**2202129**
Course	**Bachelor of Information Systems (Honours) (Information Systems Engineering)**
Practical Group	**P6**
Lecturer	**Mr.Tan Chiang Kang @ Thang Chiang Kang**

Marking scheme	Marks	Remarks
Correctness	× 2.5	
Design	× 3.5	
User Friendliness	× 2	
Neat Program Documentation		
Report Format		
TOTAL		

Introduction

This mobile app is designed to teach kindergarten children basic math through three interactive games: Counting, Number Recognition, and Missing Number. It provides two difficulty levels and features a kid-friendly interface with voice feedback and colorful visuals. The app was developed in Android Studio using Java.

App

1. Main Menu

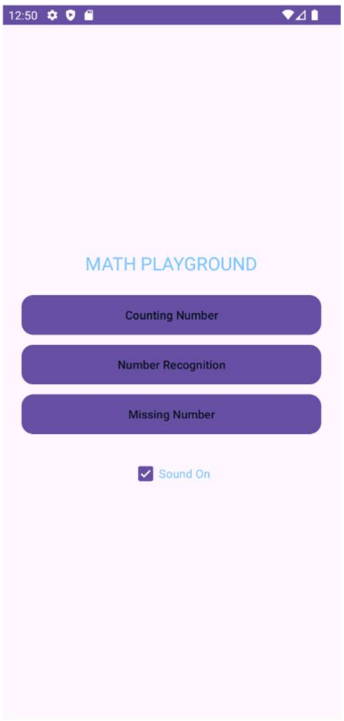

Figure 1

The Figure 1 is the starting screen of the app. It features large, colourful buttons for selecting the desired game mode: Counting, Number Recognition, or Missing Number. The design is child-friendly, a bright background, and easily readable text.

2. Difficulty Selection

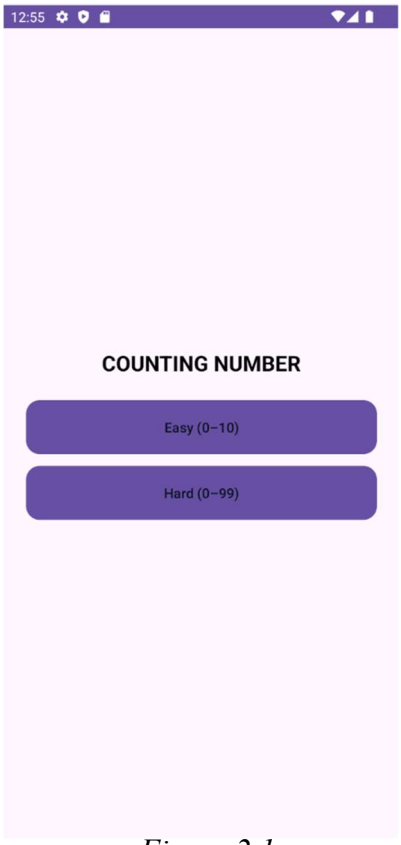

Figure 2.1

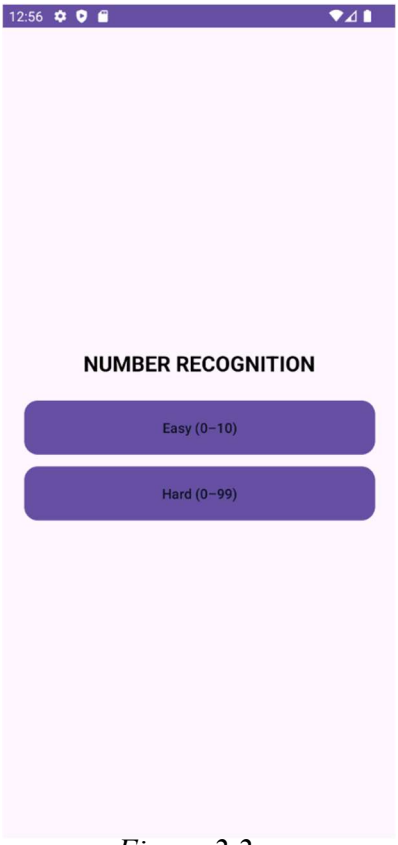

Figure 2.2

Figure2.3

The *Figure 2.1, Figure2.2 and Figure 2.3* allows the player to choose between **Easy** and **Hard** modes.

- **Easy Mode:** Uses numbers from 0 to 10, with simpler patterns and slower gameplay.
- **Hard Mode:** Uses numbers up to 99, with faster gameplay and more challenging patterns.

  The large button layout ensures children can easily make a selection.

3. Counting Game

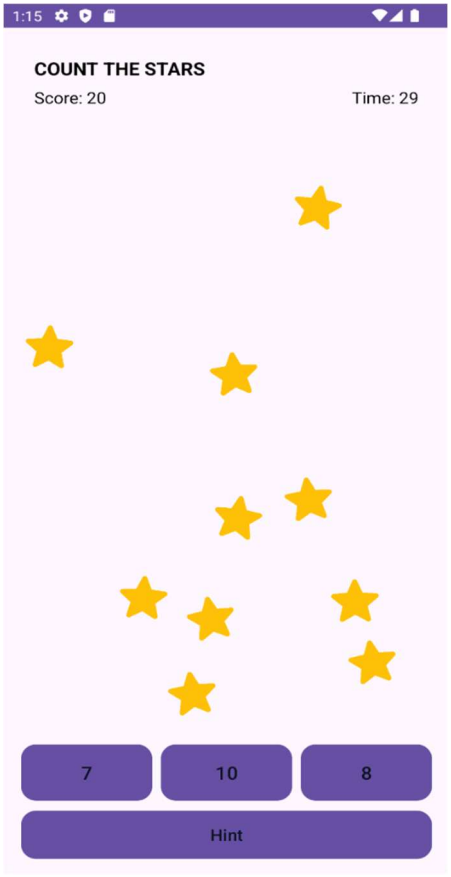

Figure 3.1

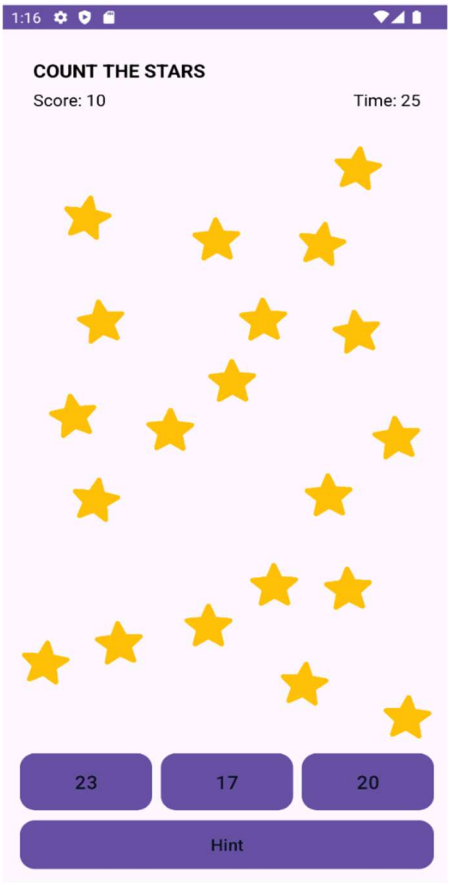

Figure 3.2

In the **EASY** mode Figure 3.1, the player is shown a random arrangement of objects (e.g., stars) on the screen. The task is to count the objects and choose the correct answer from multiple choices. Objects are randomly placed but do not overlap, making it easier for children to count. The Easy mode uses small quantities (0–10). The screen also includes a timer, score display, and voice feedback. In the **HARD** mode Figure 3.2, The objects appear in larger quantities (up to 99) and are placed more randomly across the screen to increase counting difficulty. The time limit is shorter, and the player needs to count faster. The background remains colourful, but the text is in dark colours for better visibility.

4. Number Recognition

Figure 4.1

Figure 4.2

In the **EASY** mode Figure 4.1, a number is displayed at the center of the screen, and the player must match it with the correct written word from multiple choices. For example, if "7" is shown, the correct choice would be "Seven." The Easy mode focuses on numbers from 0 to 10, with slow-paced gameplay and clear, dark-colored fonts for readability. The **HARD** mode Figure 4.2 increases the challenge by including numbers up to 99. The background remains the same, but the font color is dark to ensure good visibility even against light backgrounds. Players must answer quickly to avoid losing points.

5. Missing Number

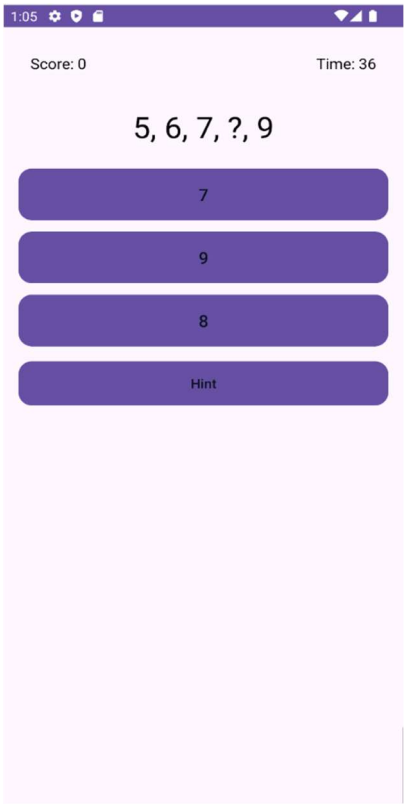

Figure 5.1

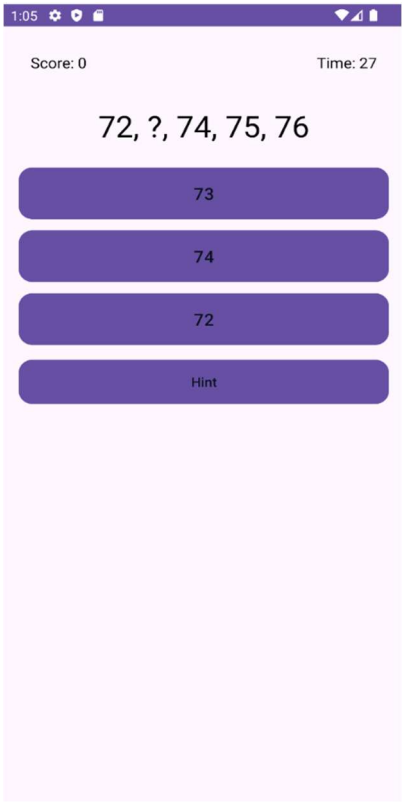

Figure 5.2

In the **EASY** mode, a simple number sequence is displayed with one number missing, such as "5, 6, 7, ?, 9" The player must select the correct missing number from the options given. Easy mode uses only numbers from 0 to 10 and straightforward sequences (counting by 1). This helps young children practice basic counting and sequencing skills. The Hard mode uses larger numbers (up to 99) such as (72, ?, 74, 75, 76). Only one number is missing in each sequence, making it challenging but still achievable. The gameplay encourages number pattern recognition.

6. Result Screen

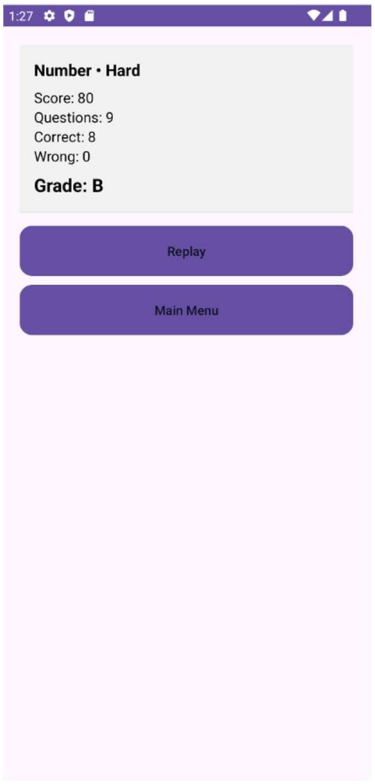

Figure 6

After completing a game, the Result Screen is shown. It displays:

- Score: Total points earned in the game
- Questions: Total number of questions attempted
- Correct Answers: Number of correct responses
- Wrong Answers: Number of incorrect responses
- Grade: A, B, C, or Fail based on performance

  The result screen uses large fonts and clear colors so children can easily see their achievements.

Source Code

1.Main Menu

1.1 MainMenuActivity.java

```
package com.sooguosheng.mathplayground;

import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.CheckBox;

import androidx.appcompat.app.AppCompatActivity;

public class MainMenuActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_main_menu);

        CheckBox chk = findViewById(R.id.chkSound);
        if (chk != null) {
            chk.setChecked(Prefs.isSoundOn(this));
            chk.setOnCheckedChangeListener((v, on) ->
Prefs.setSoundOn(this, on));
        }

        Button btnCounting = findViewById(R.id.btnCounting);
        Button btnNumber   = findViewById(R.id.btnNumber);
        Button btnMissing  = findViewById(R.id.btnMissing);

        btnCounting.setOnClickListener(v -> openDifficulty("Counting
NUMBER"));
        btnNumber.setOnClickListener(v   -> openDifficulty("Number
RECOGNITION"));
        btnMissing.setOnClickListener(v  -> openDifficulty("Missing
NUMBER"));
    }

    private void openDifficulty(String mode){
        Intent i = new Intent(this, DifficultyActivity.class);
        i.putExtra("mode", mode);
        startActivity(i);
    }
}
```

1.2 activity_main_menu.xml

```
2  <LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="24dp"
    android:layout_width="match_parent"
```

```
        android:layout_height="match_parent"
        android:background="@drawable/bg_app">

        <TextView
            android:text="MATH PLAYGROUND"
            android:textSize="22sp"
            android:textColor="@color/primary"
            android:layout_marginBottom="24dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <Button
            android:id="@+id/btnCounting"
            android:text="Counting Number"
            android:textColor="@color/onPrimary"
            android:background="@drawable/btn_primary"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>

        <Button
            android:id="@+id/btnNumber"
            android:text="Number Recognition"
            android:textColor="@color/onPrimary"
            android:background="@drawable/btn_primary"
            android:layout_marginTop="12dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>

        <Button
            android:id="@+id/btnMissing"
            android:text="Missing Number"
            android:textColor="@color/onPrimary"
            android:background="@drawable/btn_primary"
            android:layout_marginTop="12dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>

        <CheckBox
            android:id="@+id/chkSound"
            android:text="Sound On"
            android:textColor="@color/primary"
            android:layout_marginTop="24dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    </LinearLayout>
```

2. Difficulty Selection

2.1 DifficultyActivity.java

package com.sooguosheng.mathplayground;

import android.content.Intent;

```
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class DifficultyActivity extends AppCompatActivity {
    private String mode;

    @Override
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_difficulty);

        mode = getIntent().getStringExtra("mode");
        ((TextView)findViewById(R.id.txtMode)).setText(mode.toUpperCase());

        findViewById(R.id.btnEasy).setOnClickListener(v -> go("Easy"));
        findViewById(R.id.btnHard).setOnClickListener(v -> go("Hard"));
    }

    private void go(String diff){
        Intent i;
        switch (mode){
            case "Counting NUMBER": i = new Intent(this, CountingActivity.class); break;
            case "Number RECOGNITION":   i = new Intent(this,
NumberRecognitionActivity.class); break;
            default:        i = new Intent(this, MissingNumberActivity.class); break;
        }
        i.putExtra("difficulty", diff);
        startActivity(i);
```

```
    }
}
```

2.2 activity_difficulty.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="24dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg_app">

    <!-- Title: changed to black for visibility -->
    <TextView
        android:id="@+id/txtMode"
        android:text="COUNTING NUMBER"
        android:textSize="22sp"
        android:textStyle="bold"
        android:textColor="@android:color/black"
        android:layout_marginBottom="24dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <!-- Easy Button -->
    <Button
        android:id="@+id/btnEasy"
        android:text="Easy (0–10)"
        android:textColor="@color/onPrimary"
        android:background="@drawable/btn_primary"
```

```
        android:layout_width="match_parent"
        android:layout_height="56dp" />

    <!-- Hard Button -->
    <Button
        android:id="@+id/btnHard"
        android:text="Hard (0–99)"
        android:textColor="@color/onPrimary"
        android:background="@drawable/btn_primary"
        android:layout_marginTop="12dp"
        android:layout_width="match_parent"
        android:layout_height="56dp" />
</LinearLayout>
```

3. Counting game

3.1 CountingActivity.java

```
package com.sooguosheng.mathplayground;

import android.content.Intent;
import android.graphics.RectF;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.view.ViewTreeObserver;
import android.widget.Button;
import android.widget.FrameLayout;
import android.widget.ImageView;
import android.widget.TextView;
```

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.ContextCompat;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Random;

public class CountingActivity extends AppCompatActivity {

    private String difficulty;
    private UX ux;
    private final Random rng = new Random();



    private TextView txtScore, txtTimer;
    private FrameLayout canvas;
    private Button btnA, btnB, btnC, btnHint;



    private int score = 0, timeLeft = 30;
    private int correctAnswer = 0;
    private int streak = 0;
    private CountDownTimer timer;
    private int canvasW = 0, canvasH = 0;

    // Performance report
    private long roundStartMs = 0L;
    private boolean started = false;
    private int questions = 0;
    private int correct = 0;
    private int wrong = 0;
```

```
    private boolean answered = false;

    @Override
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_counting);

        difficulty = getIntent().getStringExtra("difficulty");
        ux = new UX(this);

        txtScore = findViewById(R.id.txtScore);
        txtTimer = findViewById(R.id.txtTimer);
        canvas   = findViewById(R.id.canvas);
        btnA     = findViewById(R.id.btnA);
        btnB     = findViewById(R.id.btnB);
        btnC     = findViewById(R.id.btnC);
        btnHint  = findViewById(R.id.btnHint);

        timeLeft = "Easy".equals(difficulty) ? 40 : 30;
        txtTimer.setText("Time: " + timeLeft);

        btnA.setOnClickListener(v -> checkAnswer(((Button)v).getText().toString()));
        btnB.setOnClickListener(v -> checkAnswer(((Button)v).getText().toString()));
        btnC.setOnClickListener(v -> checkAnswer(((Button)v).getText().toString()));
        btnHint.setOnClickListener(v -> ux.speak("Count the stars and choose " +
correctAnswer));


        canvas.getViewTreeObserver().addOnGlobalLayoutListener(new
```

```
ViewTreeObserver.OnGlobalLayoutListener() {
        @Override public void onGlobalLayout() {
            canvas.getViewTreeObserver().removeOnGlobalLayoutListener(this);
            canvasW = canvas.getWidth();
            canvasH = canvas.getHeight();

            if (!started) {
                started = true;
                roundStartMs = System.currentTimeMillis();
                startTimer();
                nextQuestion();
            }
        }
    });
  }

  private void startTimer(){
    timer = new CountDownTimer(timeLeft * 1000L, 1000) {
        public void onTick(long ms){ txtTimer.setText("Time: " + (ms/1000)); }
        public void onFinish(){ endRound(); }
    }.start();
  }

  private int rangeMax(){
    int base = "Easy".equals(difficulty) ? 10 : 20 + rng.nextInt(80);
    if ("Easy".equals(difficulty)) base = 10 + Math.min(5, streak);
    return Math.min(99, base);
  }

  private void nextQuestion(){
    if (canvasW == 0 || canvasH == 0) return;
```

```
    answered = false;
    setButtonsEnabled(true);

    int max = rangeMax();
    correctAnswer = "Easy".equals(difficulty) ? rng.nextInt(11) : rng.nextInt(max + 1);

    placeRandomStars(correctAnswer);

    ArrayList<Integer> opts = new ArrayList<>();
    opts.add(correctAnswer);
    while (opts.size() < 3){
       int w = Math.max(0, correctAnswer + rng.nextInt(7) - 3);
       if (!opts.contains(w)) opts.add(w);
    }
    Collections.shuffle(opts);
    btnA.setText(String.valueOf(opts.get(0)));
    btnB.setText(String.valueOf(opts.get(1)));
    btnC.setText(String.valueOf(opts.get(2)));


    questions++;

    ux.speak("How many stars?");
}

private void checkAnswer(String pickedText){

    if (answered) return;
    answered = true;
```

```
    setButtonsEnabled(false);

    int val = Integer.parseInt(pickedText);

    if (val == correctAnswer){

        correct++;
        score += 10;
        streak++;
        txtScore.setText("Score: " + score);
        ux.speak("Correct. " + correctAnswer);
    } else {

        wrong++;
        streak = Math.max(0, streak - 1);
        score = Math.max(0, score - 5);
        txtScore.setText("Score: " + score);
        ux.buzzShort();
        ux.speak("Wrong");
    }


    canvas.postDelayed(this::nextQuestion, 600);
}

private void setButtonsEnabled(boolean enabled){
    btnA.setEnabled(enabled);
    btnB.setEnabled(enabled);
    btnC.setEnabled(enabled);
}
```

```
    private void endRound(){
        int stars = score >= 90 ? 3 : score >= 60 ? 2 : 1;
        Prefs.setBestStars(this, "Counting", difficulty, stars);

        long elapsedSec = Math.max(1, (System.currentTimeMillis() - roundStartMs) /
1000);

        Intent i = new Intent(this, ResultActivity.class);
        i.putExtra("mode", "Counting");
        i.putExtra("difficulty", difficulty);
        i.putExtra("score", score);
        i.putExtra("stars", stars);
        i.putExtra("questions", questions);
        i.putExtra("correct", correct);
        i.putExtra("wrong", wrong);
        i.putExtra("elapsedSec", elapsedSec);

        startActivity(i);
        finish();
    }

    private int dp(int d){
        return (int) (d * getResources().getDisplayMetrics().density);
    }


    private void placeRandomStars(int count){
        canvas.removeAllViews();

        int sizePx = dp(44);
        int gapPx  = dp(8);
```

```
        ArrayList<RectF> placed = new ArrayList<>();
        Drawable icon = ContextCompat.getDrawable(this, R.drawable.star);
        if (icon == null) icon = ContextCompat.getDrawable(this,
android.R.drawable.btn_star_big_on);

        int maxX = Math.max(0, canvasW - sizePx);
        int maxY = Math.max(0, canvasH - sizePx);

        for (int i = 0; i < count; i++){
            boolean ok = false;
            for (int tries = 0; tries < 300 && !ok; tries++){
                int x = rng.nextInt(Math.max(1, maxX + 1));
                int y = rng.nextInt(Math.max(1, maxY + 1));
                RectF rect = new RectF(x, y, x + sizePx, y + sizePx);

                boolean overlaps = false;
                for (RectF r : placed){
                    RectF inf = new RectF(r.left - gapPx, r.top - gapPx, r.right + gapPx, r.bottom
+ gapPx);
                    if (RectF.intersects(inf, rect)) { overlaps = true; break; }
                }
                if (overlaps) continue;

                placed.add(rect);
                ImageView star = new ImageView(this);
                star.setImageDrawable(icon);
                FrameLayout.LayoutParams lp = new FrameLayout.LayoutParams(sizePx,
sizePx);
                lp.leftMargin = x; lp.topMargin = y;
                star.setLayoutParams(lp);
```

```
            star.setScaleType(ImageView.ScaleType.CENTER_INSIDE);
            star.setRotation(rng.nextInt(21) - 10);
            canvas.addView(star);
            ok = true;
          }
          if (!ok) break;
        }
      }

      @Override
      protected void onDestroy() {
        super.onDestroy();
        if (timer != null) timer.cancel();
        if (ux != null) ux.shutdown();
      }
}
```

3.2 activity_counting.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="16dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg_app">


    <LinearLayout
        android:background="@drawable/bg_panel"
        android:orientation="vertical"
```

```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:padding="12dp">

<TextView
    android:id="@+id/txtTop"
    android:text="COUNT THE STARS"
    android:textSize="18sp"
    android:textStyle="bold"
    android:textColor="@android:color/black"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="6dp">

    <TextView
        android:id="@+id/txtScore"
        android:text="Score: 0"
        android:textColor="@android:color/black"
        android:textSize="16sp"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1" />

    <TextView
        android:id="@+id/txtTimer"
        android:text="Time: 30"
```

```
            android:textColor="@android:color/black"
            android:textSize="16sp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>
</LinearLayout>


<FrameLayout
    android:id="@+id/canvas"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:foregroundGravity="center" />


<LinearLayout
    android:orientation="horizontal"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp">

    <Button
        android:id="@+id/btnA"
        android:background="@drawable/btn_primary"
        android:textColor="@color/onPrimary"
        android:textSize="18sp"
        android:layout_width="0dp"
        android:layout_height="56dp"
        android:layout_weight="1" />
```

```
    <Button
        android:id="@+id/btnB"
        android:background="@drawable/btn_primary"
        android:textColor="@color/onPrimary"
        android:textSize="18sp"
        android:layout_width="0dp"
        android:layout_height="56dp"
        android:layout_marginStart="8dp"
        android:layout_weight="1" />

    <Button
        android:id="@+id/btnC"
        android:background="@drawable/btn_primary"
        android:textColor="@color/onPrimary"
        android:textSize="18sp"
        android:layout_width="0dp"
        android:layout_height="56dp"
        android:layout_marginStart="8dp"
        android:layout_weight="1" />
</LinearLayout>

<Button
    android:id="@+id/btnHint"
    android:text="Hint"
    android:textSize="16sp"
    android:background="@drawable/btn_primary"
    android:textColor="@color/onPrimary"
    android:layout_marginTop="10dp"
    android:layout_width="match_parent"
```

```
        android:layout_height="48dp" />
</LinearLayout>
```

4. Number Recognition

4.1 NumberRecognitionActivity.java

```
package com.sooguosheng.mathplayground;

import android.content.Intent;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Random;

public class NumberRecognitionActivity extends AppCompatActivity {
    private String difficulty;
    private UX ux;
    private final Random rng = new Random();

    private TextView txtScore, txtTimer, txtNumber;
    private Button btnA, btnB, btnC, btnHint;

    private int score = 0, timeLeft = 30;
    private String correctWord;
```

```
private int correctNumber;
private int streak = 0;
private CountDownTimer timer;

// Performance
private long roundStartMs;
private int questions = 0, correct = 0, wrong = 0;

// Count-once guard
private boolean answered = false;

@Override
protected void onCreate(Bundle b) {
    super.onCreate(b);
    setContentView(R.layout.activity_number_recognition);

    difficulty = getIntent().getStringExtra("difficulty");
    ux = new UX(this);
    setVolumeControlStream(android.media.AudioManager.STREAM_MUSIC);

    txtScore  = findViewById(R.id.txtScore);
    txtTimer  = findViewById(R.id.txtTimer);
    txtNumber = findViewById(R.id.txtNumber);
    btnA      = findViewById(R.id.btnA);
    btnB      = findViewById(R.id.btnB);
    btnC      = findViewById(R.id.btnC);
    btnHint   = findViewById(R.id.btnHint);

    timeLeft = "Easy".equals(difficulty) ? 40 : 30;
    txtTimer.setText("Time: " + timeLeft);
```

```
    btnA.setOnClickListener(v -> checkAnswer(btnA.getText().toString()));
    btnB.setOnClickListener(v -> checkAnswer(btnB.getText().toString()));
    btnC.setOnClickListener(v -> checkAnswer(btnC.getText().toString()));
    btnHint.setOnClickListener(v -> speakHint());

    roundStartMs = System.currentTimeMillis();
    startTimer();
    nextQuestion();
}

private void startTimer(){
    timer = new CountDownTimer(timeLeft * 1000L, 1000) {
        public void onTick(long ms){ txtTimer.setText("Time: " + (ms/1000)); }
        public void onFinish(){ endRound(); }
    }.start();
}

private int rangeMax(){
    int base = "Easy".equals(difficulty) ? 10 : 20 + rng.nextInt(80);
    if ("Easy".equals(difficulty)) base = 10 + Math.min(5, streak);
    return Math.min(99, base);
}

private void nextQuestion(){
    answered = false;
    setButtonsEnabled(true);

    int max = rangeMax();
    correctNumber = "Easy".equals(difficulty) ? rng.nextInt(11) : rng.nextInt(max + 1);
    correctWord = Words.numToWord(correctNumber);
```

```
        txtNumber.setText(String.valueOf(correctNumber)); // big black digits (layout sets
color)

        ArrayList<String> opts = new ArrayList<>();
        opts.add(correctWord);
        while (opts.size() < 3){
            String w = Words.numToWord(rng.nextInt(100));
            if (!opts.contains(w)) opts.add(w);
        }
        Collections.shuffle(opts);
        btnA.setText(opts.get(0));
        btnB.setText(opts.get(1));
        btnC.setText(opts.get(2));

        questions++;                    // count this question once
        ux.speak("Find the word for number");
    }

    private void checkAnswer(String picked){
        if (answered) return;           // only first tap counts
        answered = true;
        setButtonsEnabled(false);

        if (picked.equals(correctWord)){
            correct++;
            score += 10;
            streak++;
            txtScore.setText("Score: " + score);
            ux.speak("Correct");
        } else {
            wrong++;
```

```
            streak = Math.max(0, streak - 1);
            score = Math.max(0, score - 5);
            txtScore.setText("Score: " + score);
            ux.buzzShort();
            ux.speak("Wrong");
        }

        txtScore.postDelayed(this::nextQuestion, 600);
    }

    private void setButtonsEnabled(boolean en){
        btnA.setEnabled(en); btnB.setEnabled(en); btnC.setEnabled(en);
    }

    private void speakHint() {
        // Partial hint only: initial letter + letter count (never reveal answer)
        String word = correctWord;
        String firstPart = word.contains("-") ? word.substring(0, word.indexOf('-')) : word;
        char initial = Character.toUpperCase(firstPart.charAt(0));
        int letters = word.replace("-", "").length();
        ux.speak("Hint: starts with " + initial + " and has " + letters + " letters.");
    }

    private void endRound(){
        int stars = score >= 90 ? 3 : score >= 60 ? 2 : 1;
        Prefs.setBestStars(this, "Number", difficulty, stars);

        long elapsedSec = Math.max(1, (System.currentTimeMillis() - roundStartMs) /
1000);

        Intent i = new Intent(this, ResultActivity.class);
```

```
        i.putExtra("mode", "Number");
        i.putExtra("difficulty", difficulty);
        i.putExtra("score", score);
        i.putExtra("stars", stars);
        i.putExtra("questions", questions);
        i.putExtra("correct", correct);
        i.putExtra("wrong", wrong);
        i.putExtra("elapsedSec", elapsedSec);
        startActivity(i);
        finish();
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        if (timer != null) timer.cancel();
        if (ux != null) ux.shutdown();
    }
}
```

4.2 activity_number_recognition.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="16dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg_app">
```

```
<!-- Top stats -->
<LinearLayout
    android:orientation="horizontal"
    android:padding="12dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/txtScore"
        android:text="Score: 0"
        android:textColor="@android:color/black"
        android:textSize="16sp"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1" />

    <TextView
        android:id="@+id/txtTimer"
        android:text="Time: 30"
        android:textColor="@android:color/black"
        android:textSize="16sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>

<!-- BIG number prompt -->
<TextView
    android:id="@+id/txtNumber"
    android:text="0"
    android:layout_marginTop="24dp"
    android:textSize="40sp"
```

```
    android:textColor="@android:color/black"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<!-- Options -->
<Button
    android:id="@+id/btnA"
    android:layout_marginTop="24dp"
    android:textColor="@color/onPrimary"
    android:background="@drawable/btn_primary"
    android:textAllCaps="false"
    android:textSize="18sp"
    android:layout_width="match_parent"
    android:layout_height="56dp"/>

<Button
    android:id="@+id/btnB"
    android:layout_marginTop="12dp"
    android:textColor="@color/onPrimary"
    android:background="@drawable/btn_primary"
    android:textAllCaps="false"
    android:textSize="18sp"
    android:layout_width="match_parent"
    android:layout_height="56dp"/>

<Button
    android:id="@+id/btnC"
    android:layout_marginTop="12dp"
    android:textColor="@color/onPrimary"
    android:background="@drawable/btn_primary"
```

```
        android:textAllCaps="false"
        android:textSize="18sp"
        android:layout_width="match_parent"
        android:layout_height="56dp"/>

    <Button
        android:id="@+id/btnHint"
        android:text="Hint"
        android:layout_marginTop="16dp"
        android:textColor="@color/onPrimary"
        android:background="@drawable/btn_primary"
        android:textAllCaps="false"
        android:layout_width="match_parent"
        android:layout_height="48dp"/>
</LinearLayout>
```

5. Missing Number

5.1 MissingNumberActivity.java

```
package com.sooguosheng.mathplayground;

import android.content.Intent;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;
```

```
import java.util.Collections;
import java.util.Random;

public class MissingNumberActivity extends AppCompatActivity {

    private String difficulty;
    private UX ux;
    private final Random rng = new Random();

    private TextView txtScore, txtTimer, txtSequence;
    private Button btnA, btnB, btnC, btnHint;

    private int score = 0, timeLeft = 30;
    private int correctAnswer;
    private int streak = 0;
    private CountDownTimer timer;

    // performance
    private long roundStartMs;
    private int questions = 0, correct = 0, wrong = 0;

    // count-once guard
    private boolean answered = false;

    @Override
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_missing_number);

        difficulty = getIntent().getStringExtra("difficulty");
        ux = new UX(this);
```

```
        txtScore    = findViewById(R.id.txtScore);
        txtTimer    = findViewById(R.id.txtTimer);
        txtSequence = findViewById(R.id.txtSequence);
        btnA        = findViewById(R.id.btnA);
        btnB        = findViewById(R.id.btnB);
        btnC        = findViewById(R.id.btnC);
        btnHint     = findViewById(R.id.btnHint);

        // ensure dark text
        int black = getResources().getColor(android.R.color.black);
        txtScore.setTextColor(black);
        txtTimer.setTextColor(black);
        txtSequence.setTextColor(black);

        timeLeft = "Easy".equals(difficulty) ? 40 : 30;
        txtTimer.setText("Time: " + timeLeft);

        btnA.setOnClickListener(v -> checkAnswer(btnA.getText().toString()));
        btnB.setOnClickListener(v -> checkAnswer(btnB.getText().toString()));
        btnC.setOnClickListener(v -> checkAnswer(btnC.getText().toString()));
        btnHint.setOnClickListener(v -> ux.speak("Find the missing number. Count
carefully."));

        roundStartMs = System.currentTimeMillis();
        startTimer();
        nextQuestion();
    }

    private void startTimer(){
        timer = new CountDownTimer(timeLeft * 1000L, 1000) {
```

```
            public void onTick(long ms){ txtTimer.setText("Time: " + (ms/1000)); }
            public void onFinish(){ endRound(); }
        }.start();
    }

    private void nextQuestion(){
        answered = false;
        setButtonsEnabled(true);

        final int length = 5;
        final int LO = "Easy".equals(difficulty) ? 0 : 0;
        final int HI = "Easy".equals(difficulty) ? 10 : 99;

        // STEP is always 1 for kindergarten level (no +2, +3...)
        int step;
        if ("Easy".equals(difficulty)) {
            step = 1;               // only forward counting 0..10
        } else {
            step = rng.nextBoolean() ? 1 : -1; // Hard can be +1 or -1 (optional backward
counting)
        }

        // choose a start that keeps all terms in [LO, HI]
        int minStart, maxStart;
        if (step >= 0) {
            minStart = LO;
            maxStart = HI - (length - 1) * step;
        } else {
            minStart = LO - (length - 1) * step; // step is -1 → LO + (length-1)
            maxStart = HI;
        }
```

```
// safe pick
int start = minStart + rng.nextInt(Math.max(1, (maxStart - minStart + 1)));

// build sequence
ArrayList<Integer> seq = new ArrayList<>(length);
for (int i = 0; i < length; i++) seq.add(start + i * step);

// pick a middle slot for "?"
int missingIndex = 1 + rng.nextInt(length - 2);
correctAnswer = seq.get(missingIndex);

// show sequence
StringBuilder sb = new StringBuilder();
for (int i = 0; i < length; i++) {
   sb.append(i == missingIndex ? "?" : seq.get(i));
   if (i < length - 1) sb.append(", ");
}
txtSequence.setText(sb.toString());

// options: correct + neighbors (±1) within bounds
ArrayList<Integer> opts = new ArrayList<>();
opts.add(correctAnswer);

while (opts.size() < 3) {
   int neighbor = correctAnswer + (rng.nextBoolean() ? 1 : -1);
   if (neighbor >= LO && neighbor <= HI && !opts.contains(neighbor)) {
      opts.add(neighbor);
   }
}
Collections.shuffle(opts);
btnA.setText(String.valueOf(opts.get(0)));
```

```
    btnB.setText(String.valueOf(opts.get(1)));
    btnC.setText(String.valueOf(opts.get(2)));

    questions++;                    // record once per question
    ux.speak("What number is missing?");
}

private void checkAnswer(String picked){
    if (answered) return;           // only first tap counts
    answered = true;
    setButtonsEnabled(false);

    int val = Integer.parseInt(picked);
    if (val == correctAnswer){
        correct++;
        score += 10;
        streak++;
        txtScore.setText("Score: " + score);
        ux.speak("Correct");
    } else {
        wrong++;
        streak = Math.max(0, streak - 1);
        score = Math.max(0, score - 5);
        txtScore.setText("Score: " + score);
        ux.buzzShort();
        ux.speak("Wrong");
    }

    txtScore.postDelayed(this::nextQuestion, 600);
}
```

```
    private void setButtonsEnabled(boolean en){
        btnA.setEnabled(en); btnB.setEnabled(en); btnC.setEnabled(en);
    }

    private void endRound(){
        int stars = score >= 90 ? 3 : score >= 60 ? 2 : 1;
        Prefs.setBestStars(this, "Missing", difficulty, stars);

        long elapsedSec = Math.max(1, (System.currentTimeMillis() - roundStartMs) /
1000);

        Intent i = new Intent(this, ResultActivity.class);
        i.putExtra("mode", "Missing");
        i.putExtra("difficulty", difficulty);
        i.putExtra("score", score);
        i.putExtra("stars", stars);
        i.putExtra("questions", questions);
        i.putExtra("correct", correct);
        i.putExtra("wrong", wrong);
        i.putExtra("elapsedSec", elapsedSec);
        startActivity(i);
        finish();
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        if (timer != null) timer.cancel();
        if (ux != null) ux.shutdown();
    }
}
```

5.2 activity_missing_number.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="16dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg_app">

    <!-- Top stats -->
    <LinearLayout
        android:orientation="horizontal"
        android:padding="12dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/txtScore"
            android:text="Score: 0"
            android:textColor="@android:color/black"
            android:textSize="16sp"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1" />

        <TextView
            android:id="@+id/txtTimer"
            android:text="Time: 30"
            android:textColor="@android:color/black"
            android:textSize="16sp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>

    <!-- Sequence with a ? -->
    <TextView
        android:id="@+id/txtSequence"
        android:text="1, 2, 3, ?, 5"
        android:layout_marginTop="24dp"
        android:textSize="32sp"
        android:textColor="@android:color/black"
        android:gravity="center"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <!-- Options -->
    <Button
        android:id="@+id/btnA"
        android:layout_marginTop="24dp"
        android:textColor="@color/onPrimary"
        android:background="@drawable/btn_primary"
        android:textSize="18sp"
        android:layout_width="match_parent"
        android:layout_height="56dp"/>
```

```
    <Button
        android:id="@+id/btnB"
        android:layout_marginTop="12dp"
        android:textColor="@color/onPrimary"
        android:background="@drawable/btn_primary"
        android:textSize="18sp"
        android:layout_width="match_parent"
        android:layout_height="56dp"/>

    <Button
        android:id="@+id/btnC"
        android:layout_marginTop="12dp"
        android:textColor="@color/onPrimary"
        android:background="@drawable/btn_primary"
        android:textSize="18sp"
        android:layout_width="match_parent"
        android:layout_height="56dp"/>

    <Button
        android:id="@+id/btnHint"
        android:text="Hint"
        android:layout_marginTop="16dp"
        android:textColor="@color/onPrimary"
        android:background="@drawable/btn_primary"
        android:layout_width="match_parent"
        android:layout_height="48dp"/>
</LinearLayout>
```

6. Result

6.1 ResultActivity.java

```
package com.sooguosheng.mathplayground;

import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class ResultActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_result);

        String mode       = getIntent().getStringExtra("mode");
        String difficulty =
getIntent().getStringExtra("difficulty");
        int score        = getIntent().getIntExtra("score", 0);
        int questions    = getIntent().getIntExtra("questions", 0);
        int correct      = getIntent().getIntExtra("correct", 0);
```

```
        int wrong        = getIntent().getIntExtra("wrong", 0);

        // Grade by accuracy only (simple rubric)
        double accuracyPct = questions > 0 ? (correct * 100.0 /
questions) : 0.0;
        String grade = gradeFromAccuracy(accuracyPct);

        ((TextView)findViewById(R.id.txtSummary))
                .setText(mode + " • " + difficulty);
        ((TextView)findViewById(R.id.txtScore)).setText("Score: " +
score);

((TextView)findViewById(R.id.txtQuestions)).setText("Questions: " +
questions);
        ((TextView)findViewById(R.id.txtCorrect)).setText("Correct:
" + correct);
        ((TextView)findViewById(R.id.txtWrong)).setText("Wrong: " +
wrong);
        ((TextView)findViewById(R.id.txtGrade)).setText("Grade: " +
grade);

        Button replay = findViewById(R.id.btnReplay);
        Button menu   = findViewById(R.id.btnMenu);

        replay.setOnClickListener(v -> {
            Intent i;
            switch (mode){
                case "Counting": i = new Intent(this,
CountingActivity.class); break;
                case "Number":   i = new Intent(this,
NumberRecognitionActivity.class); break;
                default:         i = new Intent(this,
MissingNumberActivity.class); break;
            }
            i.putExtra("difficulty", difficulty);
            startActivity(i);
            finish();
        });

        menu.setOnClickListener(v -> {
            startActivity(new Intent(this, MainMenuActivity.class));
            finish();
        });
    }

    private String gradeFromAccuracy(double acc){
        if (acc >= 90) return "A";
        if (acc >= 75) return "B";
        if (acc >= 60) return "C";
        return "Fail";
    }
}
```

6.2 activity_result.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="20dp"
    android:gravity="center_horizontal"
    android:background="@drawable/bg_app"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:background="#F2F2F2"
        android:padding="16dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:elevation="2dp">

        <!-- Title line -->
        <TextView
            android:id="@+id/txtSummary"
            android:text="Counting • Easy"
            android:textSize="18sp"
            android:textStyle="bold"
            android:textColor="@android:color/black"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <!-- Only these five lines -->
        <TextView
            android:id="@+id/txtScore"
            android:text="Score: 0"
            android:textSize="16sp"
            android:textColor="@android:color/black"
            android:layout_marginTop="8dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <TextView
            android:id="@+id/txtQuestions"
            android:text="Questions: 0"
            android:textSize="16sp"
            android:textColor="@android:color/black"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <TextView
            android:id="@+id/txtCorrect"
            android:text="Correct: 0"
            android:textSize="16sp"
            android:textColor="@android:color/black"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <TextView
```

```
            android:id="@+id/txtWrong"
            android:text="Wrong: 0"
            android:textSize="16sp"
            android:textColor="@android:color/black"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <TextView
            android:id="@+id/txtGrade"
            android:text="Grade: A"
            android:textSize="20sp"
            android:textStyle="bold"
            android:textColor="@android:color/black"
            android:layout_marginTop="8dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>

    <Button
        android:id="@+id/btnReplay"
        android:text="Replay"
        android:background="@drawable/btn_primary"
        android:textColor="@color/onPrimary"
        android:layout_marginTop="16dp"
        android:layout_width="match_parent"
        android:layout_height="56dp"/>

    <Button
        android:id="@+id/btnMenu"
        android:text="Main Menu"
        android:background="@drawable/btn_primary"
        android:textColor="@color/onPrimary"
        android:layout_marginTop="10dp"
        android:layout_width="match_parent"
        android:layout_height="56dp"/>
</LinearLayout>
```

7. Prefs.java

```
package com.sooguosheng.mathplayground;

import android.content.Context;
import android.content.SharedPreferences;

public class Prefs {
    private static final String FILE = "math_prefs";
    private static SharedPreferences p(Context c){ return
c.getSharedPreferences(FILE, Context.MODE_PRIVATE); }

    public static void setBestStars(Context c, String mode, String
difficulty, int stars){
        String k = "best_" + mode + "_" + difficulty;
        int cur = p(c).getInt(k, 0);
        if (stars > cur) p(c).edit().putInt(k, stars).apply();
```

```
    }
    public static int getBestStars(Context c, String mode, String
difficulty){
        return p(c).getInt("best_" + mode + "_" + difficulty, 0);
    }
    public static void setSoundOn(Context c, boolean
on){ p(c).edit().putBoolean("sound_on", on).apply(); }
    public static boolean isSoundOn(Context c){ return
p(c).getBoolean("sound_on", true); }
}
```

8. UX.java

```
package com.sooguosheng.mathplayground;

import android.app.Activity;
import android.content.Context;
import android.media.AudioAttributes;
import android.media.AudioManager;
import android.os.Build;
import android.os.Bundle;
import android.os.VibrationEffect;
import android.os.Vibrator;
import android.speech.tts.TextToSpeech;

import java.util.Locale;

public class UX {
    private final Context ctx;
    private TextToSpeech tts;

    // Defaults — change here or via the public setters below
    private float ttsVolume = 10.0f; // 0.0f..1.0f (multiplies
device media volume)
    private float ttsRate   = 0.75f; // 1.0f = normal speed
    private float ttsPitch  = 1.0f;  // 1.0f = normal pitch

    public UX(Context c){
        ctx = c.getApplicationContext();

        // Make hardware volume keys adjust MEDIA volume inside this
screen
        if (c instanceof Activity) {
            ((Activity)
c).setVolumeControlStream(AudioManager.STREAM_MUSIC);
        }

        tts = new TextToSpeech(ctx, status -> {
            if (status == TextToSpeech.SUCCESS) {
                tts.setLanguage(Locale.ENGLISH);
                tts.setSpeechRate(ttsRate);
                tts.setPitch(ttsPitch);
                if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.LOLLIPOP) {
                    tts.setAudioAttributes(new
```

```
AudioAttributes.Builder()
                                .setUsage(AudioAttributes.USAGE_MEDIA)
                                .setContentType(AudioAttributes.CONTENT_
TYPE_SPEECH)
                                .build());
                    }
                }
            });
    }

    // ---- Public controls you can call from Activities ----
    public void setTtsRate(float rate){           // 0.5f .. 2.0f
typical
        ttsRate = clamp(rate, 0.3f, 3.0f);
        if (tts != null) tts.setSpeechRate(ttsRate);
    }

    public void setTtsPitch(float pitch){         // 0.5f .. 2.0f
typical
        ttsPitch = clamp(pitch, 0.3f, 3.0f);
        if (tts != null) tts.setPitch(ttsPitch);
    }

    public void setTtsVolume(float volume){       // 0.0f .. 1.0f
        ttsVolume = clamp(volume, 0f, 1f);
    }

    /** Speak text at current volume/rate/pitch (respects Prefs
sound toggle). */
    public void speak(String text){
        if (text == null || text.isEmpty() || !Prefs.isSoundOn(ctx)
|| tts == null) return;

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            Bundle params = new Bundle();
            params.putFloat(TextToSpeech.Engine.KEY_PARAM_VOLUME,
ttsVolume);
            tts.speak(text, TextToSpeech.QUEUE_FLUSH, params,
"tts1");
        } else {
            // Old API fallback (no per-utterance volume control)
            tts.speak(text, TextToSpeech.QUEUE_FLUSH, null);
        }
    }

    /** Queue speech (doesn't interrupt current utterance). */
    public void speakQueue(String text){
        if (text == null || text.isEmpty() || !Prefs.isSoundOn(ctx)
|| tts == null) return;

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
            Bundle params = new Bundle();
            params.putFloat(TextToSpeech.Engine.KEY_PARAM_VOLUME,
ttsVolume);
            tts.speak(text, TextToSpeech.QUEUE_ADD, params, "ttsQ");
        } else {
            tts.speak(text, TextToSpeech.QUEUE_ADD, null);
```

```
        }
    }

    public void buzzShort(){
        Vibrator v = (Vibrator)
ctx.getSystemService(Context.VIBRATOR_SERVICE);
        if (v == null) return;
        if (Build.VERSION.SDK_INT >= 26)
            v.vibrate(VibrationEffect.createOneShot(60,
VibrationEffect.DEFAULT_AMPLITUDE));
        else v.vibrate(60);
    }

    public void shutdown(){
        if (tts != null) {
            tts.stop();
            tts.shutdown();
            tts = null;
        }
    }

    private static float clamp(float x, float lo, float hi){
        return Math.max(lo, Math.min(hi, x));
    }
}
```

9. Words.java

```
package com.sooguosheng.mathplayground;

public class Words {
    private static final String[] ones =
{"zero","one","two","three","four","five","six","seven","eight","nin
e",

"ten","eleven","twelve","thirteen","fourteen","fifteen","sixteen","s
eventeen","eighteen","nineteen"};
    private static final String[] tens =
{"","","twenty","thirty","forty","fifty","sixty","seventy","eighty",
"ninety"};
    public static String numToWord(int n){
        if (n < 20) return ones[n];
        int t = n/10, u = n%10;
        return u==0 ? tens[t] : tens[t] + "-" + ones[u];
    }
}
```