FACULTY OF SCIENCE AND INFORMATION TECHNOLOGY

OBJECT ORIENTED PROGRAMMING

TEB1043

Application name: EduKids

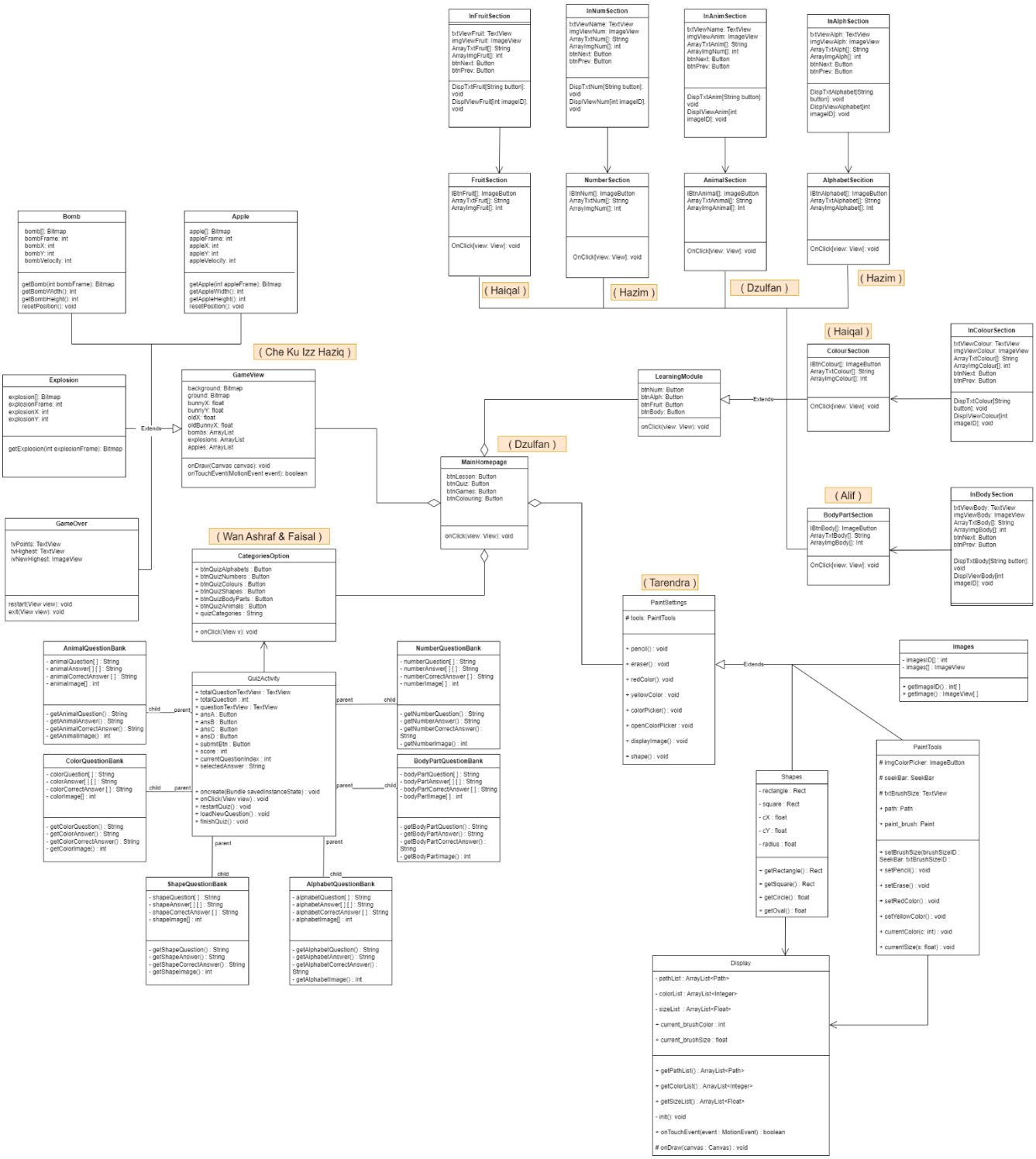
Prepared by:

NO.	NAME	STUDENT ID	PROGRAM
1	Dzulfan bin Mohd Noor	21001006	Computer Science
2	Tarendra a/l Muniandy	21000708	Computer Science
3	Muhammad Alif bin Mohd Ali Yusof	21000973	Computer Science
4	Ahmad Faisal Bin Mohd Termizi	21001125	Computer Science
5	Che Ku Izz Haziq Bin Che Ku Azam	21001325	Computer Science
6	Muhmmad Haiqal Bin Zulkifli	21001128	Computer Science
7	Muhammad Hazim Firdaus bin Jamil	21001174	Computer Science
8	Wan Muhammad Ashraf bin Wan Azrie	21001120	Computer Science

Prepared for : Dr Nordin Zakaria

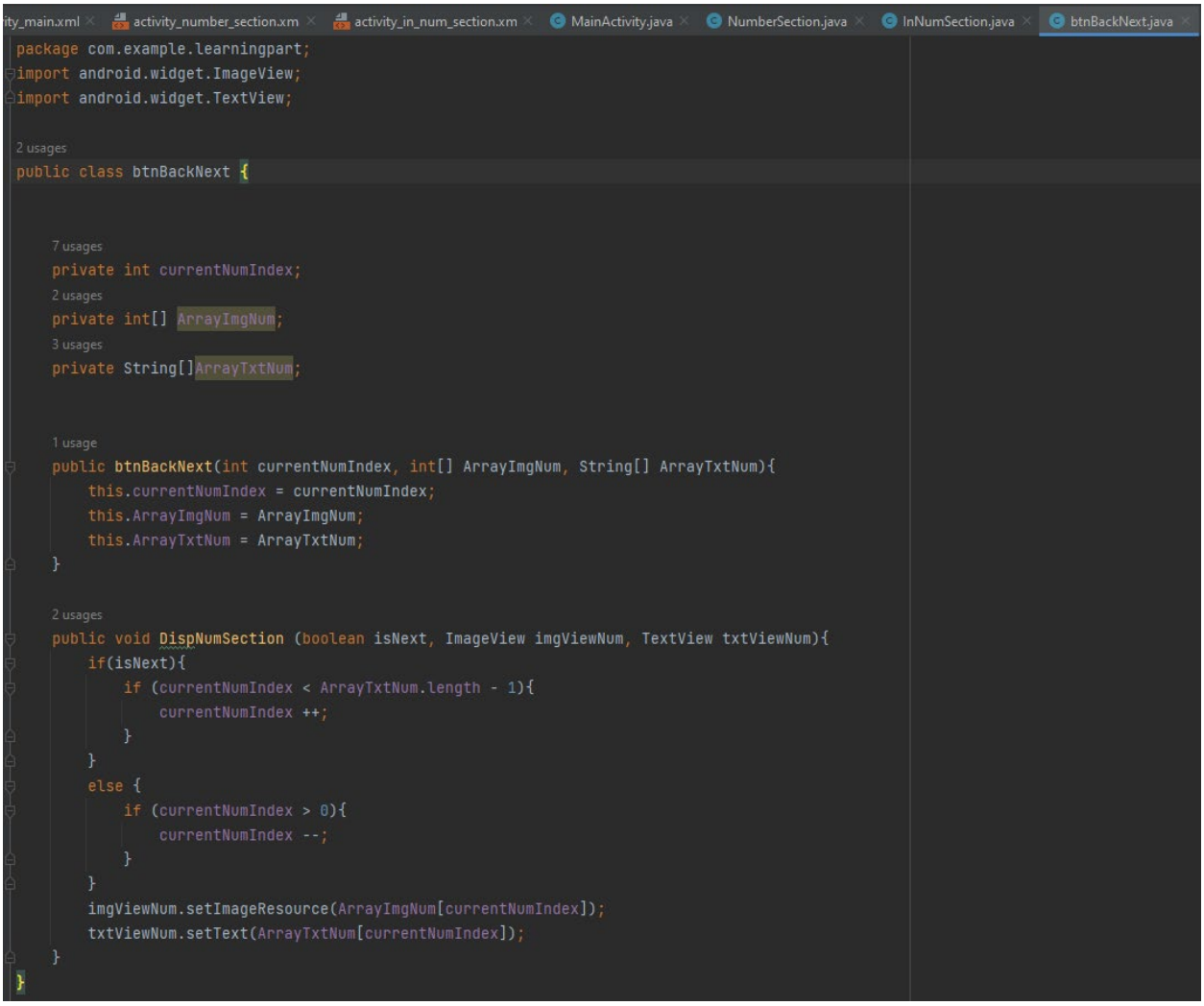
TABLE OF CONTENTS

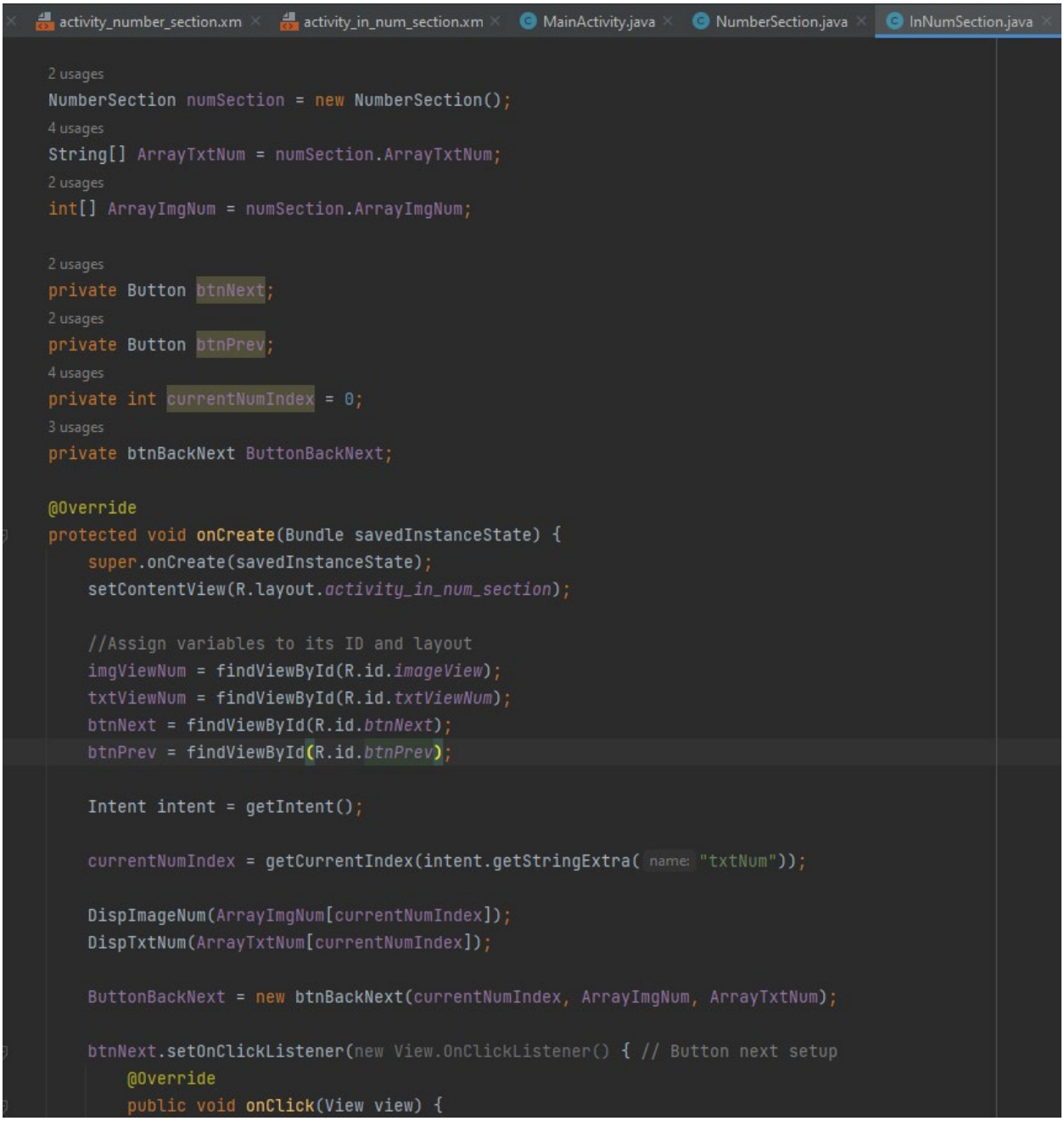
UML Design

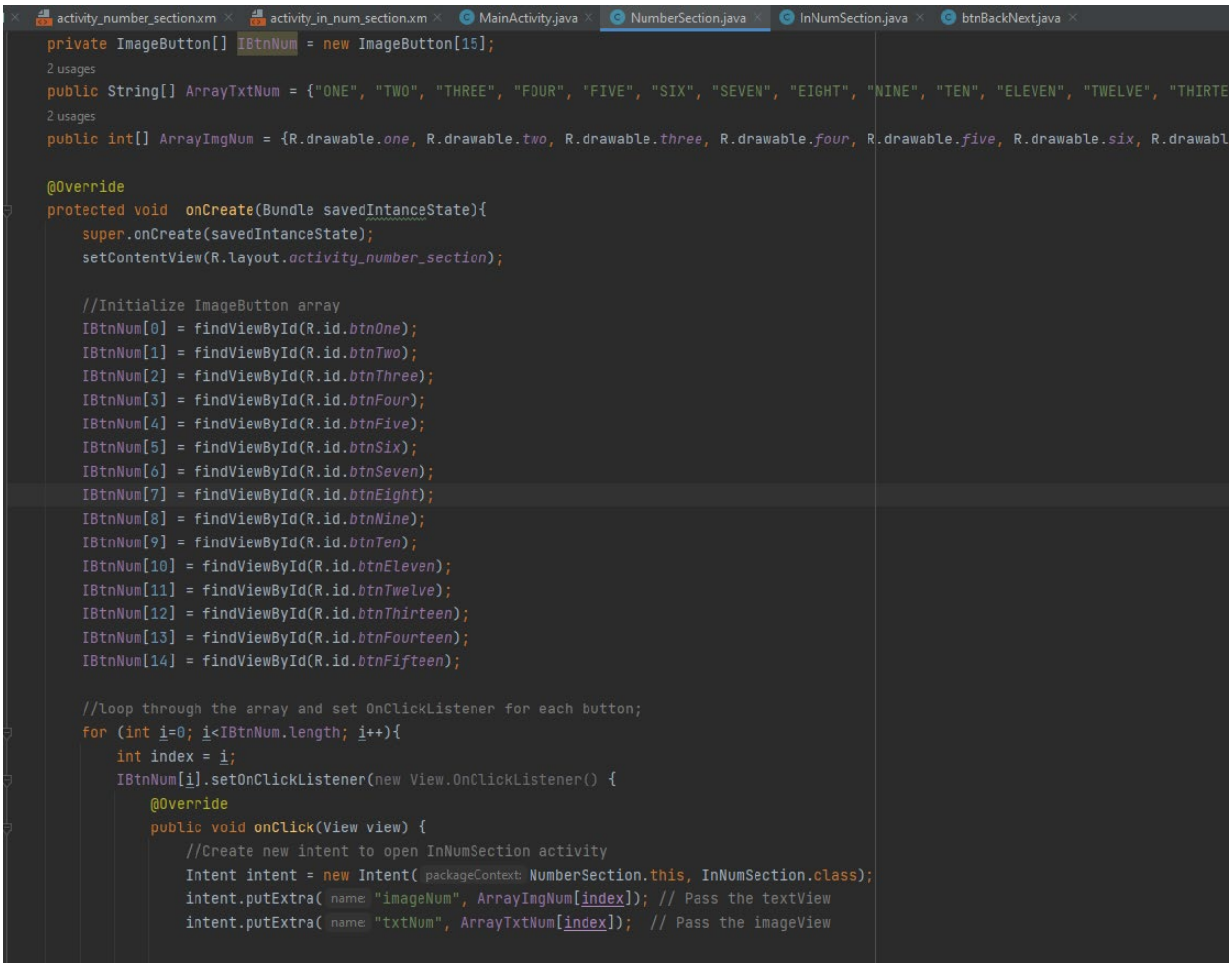

InFruitSection
txtViewFruit: TextView
imgViewFruit: ImageView
ArrayTxtFruit[]: String
ArrayImgFruit[]: int
btnNext: Button
btnPrev: Button
DispTxtFruit(String button): void
DisplViewFruit(int imageID): void
InNumSection
txtViewName: TextView
imgViewNum: ImageView
ArrayTxtNum[]: String
ArrayImgNum[]: int
btnNext: Button
btnPrev: Button
DispTxtNum(String button): void
DisplViewNum(int imageID): void
InAnimSection
txtViewName: TextView
imgViewAnim: ImageView
ArrayTxtAnim[]: String
ArrayImgNum[]: int
btnNext: Button
btnPrev: Button
DispTxtAnim(String button): void
DisplViewAnim(int imageID): void
InAlphSection
txtViewAlph: TextView
imgViewAlph: ImageView
ArrayTxtAlph[]: String
ArrayImgAlph[]: int
btnNext: Button
btnPrev: Button
DispTxtAlphabet(String button): void
DisplViewAlphabet(int imageID): void
FruitSection
IBtnFruit[]: ImageButton
ArrayTxtFruit[]: String
ArrayImgFruit[]: Int
OnClick(view: View): void
NumberSection
IBtnNum[]: ImageButton
ArrayTxtNum[]: String
ArrayImgNum[]: Int
OnClick(view: View): void
AnimalSection
IBtnAnimal[]: ImageButton
ArrayTxtAnimal[]: String
ArrayImgAnimal[]: Int
OnClick(view: View): void
AlphabetSecition
IBtnAlphabet[]: ImageButton
ArrayTxtAlphabet[]: String
ArrayImgAlphabet[]: Int
OnClick(view: View): void
(Haiqal)
(Hazim)
(Dzulfan)
(Hazim)
Bomb
bomb[]: Bitmap
bombFrame: int
bombX: int
bombY: int
bombVelocity: int
getBomb(int bombFrame): Bitmap
getBombWidth(): int
getBombHeight(): int
resetPosition(): void
Apple
apple[]: Bitmap
appleFrame: int
appleX: int
appleY: int
appleVelocity: int
getApple(int appleFrame): Bitmap
getAppleWidth(): int
getAppleHeight(): int
resetPosition(): void
(Che Ku Izz Haziq)
Explosion
explosion[]: Bitmap
explosionFrame: int
explosionX: int
explosionY: int
getExplosion(int explosionFrame): Bitmap
Extends
GameView
background: Bitmap
ground: Bitmap
bunnyX: float
bunnyY: float
oldX: float
oldBunnyX: float
bombs: ArrayList
explosions: ArrayList
apples: ArrayList
onDraw(Canvas canvas): void
onTouchEvent(MotionEvent event): boolean
LearningModule
btnNum: Button
btnAlph: Button
btnFruit: Button
btnBody: Button
onClick(view: View): void
Extends
(Haiqal)
ColourSection
IBtnColour[]: ImageButton
ArrayTxtColour[]: String
ArrayImgColour[]: int
OnClick(view: View): void
InColourSection
txtViewColour: TextView
imgViewColour: ImageView
ArrayTxtColour[]: String
ArrayImgColour[]: int
btnNext: Button
btnPrev: Button
DispTxtColour(String button): void
DisplViewColour(int imageID): void
(Dzulfan)
MainHomepage
btnLesson: Button
btnQuiz: Button
btnGames: Button
btnColouring: Button
onClick(view: View): void
(Alif)
BodyPartSection
IBtnBody[]: ImageButton
ArrayTxtBody[]: String
ArrayImgBody[]: int
OnClick(view: View): void
InBodySection
txtViewBody: TextView
imgViewBody: ImageView
ArrayTxtBody[]: String
ArrayImgBody[]: int
btnNext: Button
btnPrev: Button
DispTxtBody(String button): void
DisplViewBody(int imageID): void
GameOver
tvPoints: TextView
tvHighest: TextView
ivNewHighest: ImageView
restart(View view): void
exit(View view): void
(Wan Ashraf & Faisal)
CategoriesOption
+ btnQuizAlphabets : Button
+ btnQuizNumbers : Button
+ btnQuizColours : Button
+ btnQuizShapes : Button
+ btnQuizBodyParts : Button
+ btnQuizAnimals : Button
+ quizCategories : String
+ onClick(View v): void
(Tarendra)
PaintSettings
tools: PaintTools
+ pencil() : void
+ eraser() : void
+ redColor(): void
+ yellowColor : void
+ colorPicker() : void
+ openColorPicker : void
+ displayImage() : void
+ shape() : void
Extends
Images
- imagesID[] : int
- images[] : ImageView
+ getImageID() : int[ ]
+ getImage() : ImageView[ ]
AnimalQuestionBank
- animalQuestion[ ] : String
- animalAnswer[ ] [ ] : String
- animalCorrectAnswer [ ] : String
- animalImage[] : int
- getAnimalQuestion() : String
- getAnimalAnswer() : String
- getAnimalCorrectAnswer() : String
- getAnimalImage() : int
child
parent
QuizActivity
+ totalQuestionTextView : TextView
+ totalQuestion : int
+ questionTextView : TextView
+ ansA : Button
+ ansB : Button
+ ansC : Button
+ ansD : Button
+ submitBtn : Button
+ score : int
+ currentQuestionIndex : int
+ selectedAnswer : String
+ oncreate(Bundle savedInstanceState) : void
+ onClick(View view) : void
+ restartQuiz() : void
+ loadNewQuestion() : void
+ finishQuiz() : void
parent
child
NumberQuestionBank
- numberQuestion[ ] : String
- numberAnswer[ ] [ ] : String
- numberCorrectAnswer [ ] : String
- numberImage[ ] : int
- getNumberQuestion() : String
- getNumberAnswer() : String
- getNumberCorrectAnswer() : String
- getNumberImage() : int
ColorQuestionBank
- colorQuestion[ ] : String
- colorAnswer[ ] [ ] : String
- colorCorrectAnswer [ ] : String
- colorImage[] : int
- getColorQuestion() : String
- getColorAnswer() : String
- getColorCorrectAnswer() : String
- getColorImage() : int
BodyPartQuestionBank
- bodyPartQuestion[ ] : String
- bodyPartAnswer[ ] [ ] : String
- bodyPartCorrectAnswer [ ] : String
- bodyPartImage[ ] : int
- getBodyPartQuestion() : String
- getBodyPartAnswer() : String
- getBodyPartCorrectAnswer() : String
- getBodyPartImage() : int
ShapeQuestionBank
- shapeQuestion[ ] : String
- shapeAnswer[ ] [ ] : String
- shapeCorrectAnswer [ ] : String
- shapeImage[] : int
- getShapeQuestion() : String
- getShapeAnswer() : String
- getShapeCorrectAnswer() : String
- getShapeImage() : int
AlphabetQuestionBank
- alphabetQuestion[ ] : String
- alphabetAnswer[ ] [ ] : String
- alphabetCorrectAnswer [ ] : String
- alphabetImage[] : int
- getAlphabetQuestion() : String
- getAlphabetAnswer() : String
- getAlphabetCorrectAnswer() : String
- getAlphabetImage() : int
Shapes
- rectangle : Rect
- square : Rect
- cX : float
- cY : float
- radius : float
+ getRectangle() : Rect
+ getSquare() : Rect
+ getCircle() : float
+ getOval() : float
PaintTools
imgColorPicker: ImageButton
seekBar: SeekBar
txtBrushSize: TextView
+ path: Path
+ paint_brush: Paint
+ setBrushSize(brushSizeID : SeekBar, txtBrushSizeID :
+ setPencil() : void
+ setErase() : void
+ setRedColor() : void
+ setYellowColor() : void
+ currentColor(c: int) : void
+ currentSize(s: float) : void
Display
- pathList : ArrayList<Path>
- colorList : ArrayList<Integer>
- sizeList : ArrayList<Float>
+ current_brushColor : int
+ current_brushSize : float
+ getPathList() : ArrayList<Path>
+ getColorList() : ArrayList<Integer>
+ getSizeList() : ArrayList<Float>
- init(): void
+ onTouchEvent(event : MotionEvent) : boolean
# onDraw(canvas : Canvas) : void

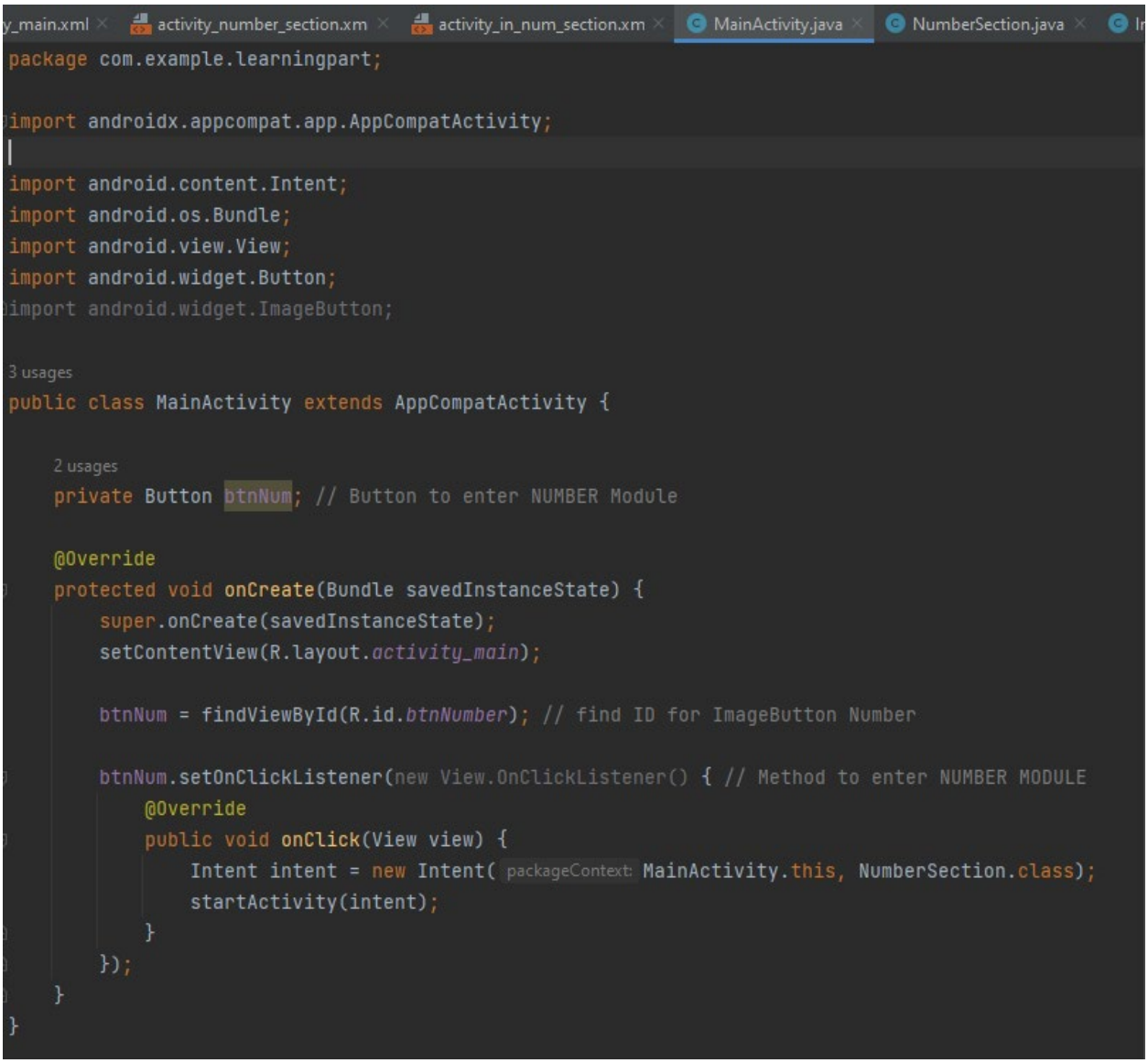
Program Codes

1) Learning Feature




activity_number_section.xm
activity_in_num_section.xm
MainActivity.java
NumberSection.java
InNumSection.java
2 usages
NumberSection numSection = new NumberSection();
4 usages
String[] ArrayTxtNum = numSection.ArrayTxtNum;
2 usages
int[] ArrayImgNum = numSection.ArrayImgNum;
2 usages
private Button btnNext;
2 usages
private Button btnPrev;
4 usages
private int currentNumIndex = 0;
3 usages
private btnBackNext ButtonBackNext;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_in_num_section);
//Assign variables to its ID and layout
imgViewNum = findViewById(R.id.imageView);
txtViewNum = findViewById(R.id.txtViewNum);
btnNext = findViewById(R.id.btnNext);
btnPrev = findViewById(R.id.btnPrev);
Intent intent = getIntent();
currentNumIndex = getCurrentIndex(intent.getStringExtra(name: "txtNum"));
DispImageNum(ArrayImgNum[currentNumIndex]);
DispTxtNum(ArrayTxtNum[currentNumIndex]);
ButtonBackNext = new btnBackNext(currentNumIndex, ArrayImgNum, ArrayTxtNum);
btnNext.setOnClickListener(new View.OnClickListener() { // Button next setup
@Override
public void onClick(View view) {

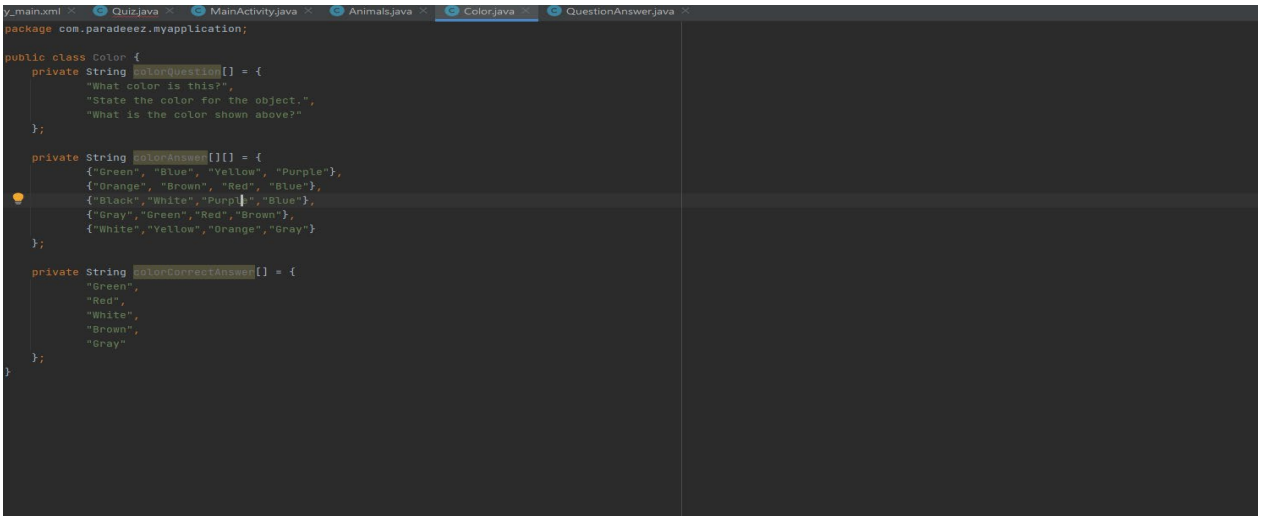

activity_number_section.xm
activity_in_num_section.xm
MainActivity.java
NumberSection.java
InNumSection.java
btnBackNext.java
private ImageButton[] IBtnNum = new ImageButton[15];
2 usages
public String[] ArrayTxtNum = {"ONE", "TWO", "THREE", "FOUR", "FIVE", "SIX", "SEVEN", "EIGHT", "NINE", "TEN", "ELEVEN", "TWELVE", "THIRTE
2 usages
public int[] ArrayImgNum = {R.drawable.one, R.drawable.two, R.drawable.three, R.drawable.four, R.drawable.five, R.drawable.six, R.drawabl
@Override
protected void onCreate(Bundle savedIntanceState){
super.onCreate(savedIntanceState);
setContentView(R.layout.activity_number_section);
//Initialize ImageButton array
IBtnNum[0] = findViewById(R.id.btnOne);
IBtnNum[1] = findViewById(R.id.btnTwo);
IBtnNum[2] = findViewById(R.id.btnThree);
IBtnNum[3] = findViewById(R.id.btnFour);
IBtnNum[4] = findViewById(R.id.btnFive);
IBtnNum[5] = findViewById(R.id.btnSix);
IBtnNum[6] = findViewById(R.id.btnSeven);
IBtnNum[7] = findViewById(R.id.btnEight);
IBtnNum[8] = findViewById(R.id.btnNine);
IBtnNum[9] = findViewById(R.id.btnTen);
IBtnNum[10] = findViewById(R.id.btnEleven);
IBtnNum[11] = findViewById(R.id.btnTwelve);
IBtnNum[12] = findViewById(R.id.btnThirteen);
IBtnNum[13] = findViewById(R.id.btnFourteen);
IBtnNum[14] = findViewById(R.id.btnFifteen);
//loop through the array and set OnClickListener for each button;
for (int i=0; i<IBtnNum.length; i++){
int index = i;
IBtnNum[i].setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
//Create new intent to open InNumSection activity
Intent intent = new Intent(packageContext: NumberSection.this, InNumSection.class);
intent.putExtra(name: "imageNum", ArrayImgNum[index]); // Pass the textView
intent.putExtra(name: "txtNum", ArrayTxtNum[index]); // Pass the imageView


y_main.xml
activity_number_section.xm
activity_in_num_section.xm
MainActivity.java
NumberSection.java
package com.example.learningpart;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
3 usages
public class MainActivity extends AppCompatActivity {
2 usages
private Button btnNum; // Button to enter NUMBER Module
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
btnNum = findViewById(R.id.btnNumber); // find ID for ImageButton Number
btnNum.setOnClickListener(new View.OnClickListener() { // Method to enter NUMBER MODULE
@Override
public void onClick(View view) {
Intent intent = new Intent(packageContext: MainActivity.this, NumberSection.class);
startActivity(intent);
}
});
}
}

2) Quiz feature

```java
public void onClick(View view) {

    ansA.setBackgroundColor(android.graphics.Color.WHITE);
    ansB.setBackgroundColor(android.graphics.Color.WHITE);
    ansC.setBackgroundColor(android.graphics.Color.WHITE);
    ansD.setBackgroundColor(android.graphics.Color.WHITE);

    Button clickedButton = (Button) view;
    if(clickedButton.getId()==R.id.submit_btn){
        if(selectedAnswer.equals(QuestionAnswer.correctAnswers[currentQuestionIndex])){
            score++;
        }
        currentQuestionIndex++;
        loadNewQuestion();


    }else{
        //choices button clicked
        selectedAnswer  = clickedButton.getText().toString();
        clickedButton.setBackgroundColor(Color.MAGENTA);

    }

}

3 usages
void loadNewQuestion(){

    if(currentQuestionIndex == totalQuestion ){
        finishQuiz();
        return;
    }

    questionTextView.setText(QuestionAnswer.question[currentQuestionIndex]);
    ansA.setText(QuestionAnswer.choices[currentQuestionIndex][0]);
    ansB.setText(QuestionAnswer.choices[currentQuestionIndex][1]);
    ansC.setText(QuestionAnswer.choices[currentQuestionIndex][2]);
    ansD.setText(QuestionAnswer.choices[currentQuestionIndex][3]);

}

1 usage
```

```
ty_main.xml  Quiz.java  MainActivity.java  Animals.java  Color.java  QuestionAnswer.java
package com.paradeeez.myapplication;

public class Animals {

    public static String question[] ={
            "What animal is this?",
            "Do you know the name of the animal?",
            "Could you identify the animal by name?",
            "What do you call this animal?",
            "What is the name of the above animal?"


    };

    public static String choices[][] = {
            {"Cat", "Bird", "Goat", "Cow"},
            {"Armadillo", "Cat", "Giraffe", "Elephant"},
            {"Duck","Horse","Dog","Chicken"},
            {"Rabbit","Grasshopper","Dolphin","Bee"},
            {"Fish","Lion","Snake","Deer"},
            {"Monkey","Gorilla","Bat","Bear"},
            {"Bear","Cat","Giraffe","Rabbit"},
            {"Rabbit","Fox","Fish","Bee"}

    };

    public static String correctAnswers[] = {
            "Goat",
            "Giraffe",
            "Chicken",
            "Rabbit",
            "Lion",
            "Monkey",
            "Bear",
            "Fox"
    };

}
```


y_main.xml
Quiz.java
MainActivity.java
Animals.java
Color.java
QuestionAnswer.java
package com.paradeeez.myapplication;
public class Color {
private String colorQuestion[] = {
"What color is this?",
"State the color for the object.",
"What is the color shown above?"
};
private String colorAnswer[][] = {
{"Green", "Blue", "Yellow", "Purple"},
{"Orange", "Brown", "Red", "Blue"},
{"Black","White","Purple","Blue"},
{"Gray","Green","Red","Brown"},
{"White","Yellow","Orange","Gray"}
};
private String colorCorrectAnswer[] = {
"Green",
"Red",
"White",
"Brown",
"Gray"
};
}

3) Game feature

```
no usages
public Bomb(Context context) {
    bomb[0] = BitmapFactory.decodeResource(context.getResources(), R.drawable.bomb0);
    bomb[1] = BitmapFactory.decodeResource(context.getResources(), R.drawable.bomb1);
    bomb[2] = BitmapFactory.decodeResource(context.getResources(), R.drawable.bomb2);
    bomb[3] = BitmapFactory.decodeResource(context.getResources(), R.drawable.bomb3);
    random = new Random();
    resetPosition();
}

no usages
public Bitmap getBomb(int bombFrame) { return bomb[bombFrame]; }

1 usage
public int getBombWidth() { return bomb[0].getWidth(); }

no usages
public int getBombHeight() { return bomb[0].getHeight(); }

1 usage
public void resetPosition() {
    bombX = random.nextInt(GameView.dWidth - getBombWidth());
    bombY = -200 + random.nextInt(600) * -1;
    bombVelocity = 25 + random.nextInt(10);
}
```

```
no usages
public class MainActivity extends AppCompatActivity {

    no usages
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
    }

    no usages
    public void startGame(View view) {
        GameView gameView = new GameView(this);
        setContentView(gameView);
    }
}
```

```
public GameView(Context context) {
    super(context);
    this.context = context;
    background = BitmapFactory.decodeResource(getResources(), R.drawable.game_bg);
    ground = BitmapFactory.decodeResource(getResources(), R.drawable.ground_game);
    bunny = BitmapFactory.decodeResource(getResources(), R.drawable.bunny);
    Display display = ((Activity) getContext()).getWindowManager().getDefaultDisplay();
    Point size = new Point();
    display.getSize(size);
    dWidth = size.x;
    dHeight = size.y;
    rectBackground = new Rect(0,0,dWidth,dHeight);
    rectGround = new Rect(0, dHeight - ground.getHeight(), dWidth, dHeight);
    handler = new Handler();
    runnable = new Runnable() {
        no usages
        @Override
        public void run() { invalidate(); }
    };
    textPaint.setColor(Color.rgb(255,165,0));
    textPaint.setTextSize(TEXT_SIZE);
    textPaint.setTextAlign(Paint.Align.LEFT);
    textPaint.setTypeface(ResourcesCompat.getFont(context, R.font.rackis));
    healthPaint.setColor(Color.GREEN);
    random = new Random();
    bunnyX = dWidth / 2 - bunny.getWidth() / 2;
    bunnyY = dHeight - ground.getHeight() - bunny.getHeight();
    bombs = new ArrayList<>();
    explosions = new ArrayList<>();
```

```
@Override
public void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.game_over);
    tvPoints = findViewById(R.id.tvPoints);
    tvHighest = findViewById(R.id.tvHighest);
    ivNewHighest = findViewById(R.id.ivNewHighest);
    int points = getIntent().getExtras().getInt("points");
    tvPoints.setText("" + points);
    sharedPreferences = getSharedPreferences("my_pref", 0);
    int highest = sharedPreferences.getInt("highest", 0);
    if (points > highest) {
        ivNewHighest.setVisibility(View.VISIBLE);
        highest = points;
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putInt("highest", highest);
        editor.commit();
    }
    tvHighest.setText("" + highest);
}

no usages
public void restart(View view) {
    Intent intent = new Intent(GameOver.this, MainActivity.class);
    startActivity(intent);
    finish();
}

no usages
public void exit(View view) { finish(); }
```

4) Painting feature

a) PaintSettings.java

```java

import static com.example.oopprojectcolouringapp.Display.current_brushColor;
import static com.example.oopprojectcolouringapp.PaintTools.*;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.SeekBar;
import android.widget.TextView;

import yuku.ambilwarna.AmbilWarnaDialog;

public class PaintSettings extends AppCompatActivity{

    //Section A: Data attributes & objects
    PaintTools tools = new PaintTools();

    //Section B: Operations
    //-----------------------------------------------------------------------------------------------
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        setBrushSize(findViewById(R.id.brushSize), findViewById(R.id.txtBrushSize));
    }
```

```java
        setBrushSize(findViewById(R.id.brushSize), findViewById(R.id.txtBrushSize));
    }

    private void openColorPicker() {

        AmbilWarnaDialog ambilWarnaDialog = new AmbilWarnaDialog( context: this, current_brushColor, new AmbilWarnaDialog.OnAmbilWarnaListener() {
            @Override
            public void onCancel(AmbilWarnaDialog dialog) {

            }

            @Override
            public void onOk(AmbilWarnaDialog dialog, int color) {
                current_brushColor = color;
                paint_brush.setColor(color);
                currentColor(paint_brush.getColor());  ////----<<<<<<
            }
        });
        ambilWarnaDialog.show();
    }

    public void pencil(View view) {
        //tools.setPencil();
        setPencil();
    }

    public void eraser(View view) {
        //tools.setEraser();
        setEraser();
    }
```

```
61
62      public void redColor(View view) {
63          //tools.setRedColor();
64          setRedColor();
65      }
66
67      public void yellowColor(View view) {
68          //tools.setYellowColor();
69          setYellowColor();
70      }
71
72      public void colorPicker(View view) {
73          openColorPicker();
74
75      }
76
77  }
78
```

b) PaintTools.java

```
2
3   import static com.example.oopprojectcolouringapp.Display.getColorList;
4   import static com.example.oopprojectcolouringapp.Display.current_brushColor;
5   import static com.example.oopprojectcolouringapp.Display.current_brushSize;
6   import static com.example.oopprojectcolouringapp.Display.getPathList;
7   import static com.example.oopprojectcolouringapp.Display.getSizeList;
8
9   import androidx.appcompat.app.AppCompatActivity;
10
11  import android.content.Context;
12  import android.graphics.Color;
13  import android.graphics.Paint;
14  import android.graphics.Path;
15  import android.os.Bundle;
16  import android.util.AttributeSet;
17  import android.view.View;
18  import android.widget.Button;
19  import android.widget.ImageButton;
20  import android.widget.SeekBar;
21  import android.widget.TextView;
22
23  import yuku.ambilwarna.AmbilWarnaDialog;
24
25  public class PaintTools extends PaintSettings {
26
27      //Section A: Declare the data attributes
28      protected static ImageButton imgColorPicker;
29      protected static SeekBar seekBar;
30      protected static TextView txtBrushSize;
31      //int defaultColor;
```

```
32
33      public static Path path = new Path();
34      public static Paint paint_brush = new Paint();
35
36      //Section B: Operations
37      //-----------------------------------------------------------------------------------------------
38      public static void setBrushSize(SeekBar brushSizeID, TextView txtBrushSizeID) {
39          seekBar = brushSizeID;
40          txtBrushSize = txtBrushSizeID;
41
42          seekBar.setProgress(20);
43          txtBrushSize.setText((int) current_brushSize + "dp");
44
45          seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
46              @Override
47              public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
48                  txtBrushSize.setText(i + "dp");
49                  current_brushSize = i;
50                  paint_brush.setStrokeWidth(i);
51                  currentSize(paint_brush.getStrokeWidth());
52                  seekBar.setMax(50);
53
54              }
55
56              @Override
57              public void onStartTrackingTouch(SeekBar seekBar) {
58
59              }
60
```

```
60
61              @Override
62              public void onStopTrackingTouch(SeekBar seekBar) {
63
64              }
65          });
66      }
67
68      public static void setPencil(){
69          paint_brush.setColor(Color.BLACK);
70          currentColor(paint_brush.getColor());
71      }
72
73      public static void setEraser(){
74          getPathList().clear();
75          getColorList().clear();
76          getSizeList().clear();
77          path.reset();
78      }
79
80      public static void setRedColor(){
81          paint_brush.setColor(Color.RED);
82          currentColor(paint_brush.getColor());
83      }
84
85      public static void setYellowColor(){
86          paint_brush.setColor(Color.YELLOW);
87          currentColor(paint_brush.getColor());
88      }
89
```

```
    public static void currentColor(int c){
        current_brushColor = c;
        path = new Path();
    }

    //Brush size
    public static void currentSize(float s){
        current_brushSize = s;
        path = new Path();
    }

}

```

c) Display.java

```

import static com.example.oopprojectcolouringapp.PaintTools.paint_brush;
import static com.example.oopprojectcolouringapp.PaintTools.path;

import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Path;
import android.util.AttributeSet;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.Nullable;

import java.util.ArrayList;

public class Display extends View {

    //Declare data attributes
    private static ArrayList<Path> pathList = new ArrayList<>();
    private static ArrayList<Integer> colorList = new ArrayList<>();
    private static ArrayList<Float> sizeList = new ArrayList<>();    ////////
    public ViewGroup.LayoutParams params;
    public static int current_brushColor = Color.BLACK;
    public static  float current_brushSize = 10;  ///////

```

```
29
30          // Operations
31          public Display(Context context) {
32              super(context);
33              init(context);
34          }
35
36          public Display(Context context, @Nullable AttributeSet attrs) {
37              super(context, attrs);
38              init(context);
39          }
40
41          public Display(Context context, @Nullable AttributeSet attrs, int defStyleAttr) {
42              super(context, attrs, defStyleAttr);
43              init(context);
44          }
45
46          public static ArrayList<Path> getPathList() { return pathList; }
49
50          public static ArrayList<Integer> getColorList() { return colorList; }
53
54          public static ArrayList<Float> getSizeList() { return sizeList; }
57
58          private void init(Context context){
59              paint_brush.setAntiAlias(true);
60              paint_brush.setColor(Color.BLACK);
61              paint_brush.setStyle(Paint.Style.STROKE);
62              paint_brush.setStrokeCap(Paint.Cap.ROUND);
63              paint_brush.setStrokeJoin(Paint.Join.ROUND);
```

```
64              paint_brush.setStrokeWidth(10f);
65
66              params = new ViewGroup.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.WRAP_CONTENT);
67          }
68
69          @Override
70          public boolean onTouchEvent(MotionEvent event) {
71              float x = event.getX();
72              float y = event.getY();
73              switch (event.getAction()){
74                  case MotionEvent.ACTION_DOWN:
75                      path.moveTo(x,y);
76                      invalidate();
77                      return true;
78                  case MotionEvent.ACTION_MOVE:
79                      path.lineTo(x,y);
80                      pathList.add(path);
81                      colorList.add(current_brushColor);
82                      sizeList.add(current_brushSize);  //////
83                      invalidate();
84                      return true;
85                  default:
86                      return false;
87              }
88          }
89
```

```
    @Override
    protected void onDraw(Canvas canvas) {
        for (int i = 0; i< pathList.size(); i++){
            paint_brush.setColor(colorList.get(i));
            paint_brush.setStrokeWidth(sizeList.get(i));   ///////
            canvas.drawPath(pathList.get(i), paint_brush);
            invalidate();
        }
    }
}
```

d) Images.java

```
import android.widget.ImageView;

public class Images extends PaintSettings{
    //  In progress // Have not compiled //

    //Data attributes
    private static int imagesID[];
    private static ImageView images[];

    //Operations
    public static int[] getImageID(){

    }

    public static ImageView[] getImage(){

    }
}
```