Ćwiczenia 9 – Android studio – Paint, Canvas

Na koniec zajęć prześlij pliki źródłowe (.xml, .java)+ obrazek do zasobu w teams.

1. Utwórz nowy projekt o nazwie MyDraw na podstawie Empty Activity (dobrać odpowiednie API).
2. Otworzyć dokumentację:

https://developer.android.com/training/custom-views/create-view

https://developer.android.com/training/custom-views/custom-drawing

https://www.raywenderlich.com/142-android-custom-view-tutorial#toc-anchor-007

3. Utwórz klasę MyTestView i rozszerzyć o klasę View (extends) na kształt:

Kotlin Java

```
class PieChart extends View {
    public PieChart(Context context, AttributeSet attrs) {
        super(context, attrs);
    }
}
```

4. Dodaj konstruktor z dwoma parametrami (Context context, AttributeSet attrs)
5. Zaimplementuj metody:

@Override protected void onDraw(Canvas canvas)

@Override protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec)

private void init() {} - wywołaj w konstruktorze lub w onDraw()

6. Dodaj do activity_main.xml swój widok:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <com.example.mydraw.MyTestView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

```
private void init() {
    textPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
    textPaint.setColor(textColor);
    if (textHeight == 0) {
        textHeight = textPaint.getTextSize();
    } else {
        textPaint.setTextSize(textHeight);
    }

    piePaint = new Paint(Paint.ANTI_ALIAS_FLAG);
    piePaint.setStyle(Paint.Style.FILL);
    piePaint.setTextSize(textHeight);

    shadowPaint = new Paint(0);
    shadowPaint.setColor(0xff101010);
    shadowPaint.setMaskFilter(new BlurMaskFilter(8, BlurMaskFilter.Blur.NORMAL));

    ...
```

7. Narysuj koło z użyciem drawCircle(x,y,promien,redPaint) (redPaint.setColor(Color.rgb(255,0,0)))
8. Wykonaj testy na metodach:

- Draw text using `drawText()`. Specify the typeface by calling `setTypeface()`, and the text color by calling `setColor()`.
- Draw primitive shapes using `drawRect()`, `drawOval()`, and `drawArc()`. Change whether the shapes are filled, outlined, or both by calling `setStyle()`.
- Draw more complex shapes using the `Path` class. Define a shape by adding lines and curves to a `Path` object, then draw the shape using `drawPath()`. Just as with primitive shapes, paths can be outlined, filled, or both, depending on the `setStyle()`.
- Define gradient fills by creating `LinearGradient` objects. Call `setShader()` to use your `LinearGradient` on filled shapes.
- Draw bitmaps using `drawBitmap()`.

9. Przetestuj działanie.
10. Zdefiniuj własne atrybuty w **res/values/attrs.xml**

To define custom attributes, add `<declare-styleable>` resources to your project. It's customary to put these resources into a `res/values/attrs.xml` file. Here's an example of an `attrs.xml` file:

```
<resources>
    <declare-styleable name="PieChart">
        <attr name="showText" format="boolean" />
        <attr name="labelPosition" format="enum">
            <enum name="left" value="0"/>
            <enum name="right" value="1"/>
        </attr>
    </declare-styleable>
</resources>
```

11. Dodaj wybrane fonty dla czcionki, a następnie przetestuj napisy po ścieżce, po kole.
12. Narysuj pieciokąt wykorzystując rysowanie „po ścieżce”.

13. Stwórz klasę rysującą wykres funkcji sinus z układem, podziałką i skalą.

Wskazówki: użyj metod drawPoint i setStrokeWidth(5).

14. Stwórz klasę rysującą wykres funkcji tangens z układem, podziałką i skalą.
15. Dodaj obsługę gestów (skalowanie i przesunięcie),

https://developer.android.com/training/gestures/scale#scale

16. Zadania dodatkowe:
    a) zaprojektuj rysowanie gwiazdy na podstawie danych punktów ( )
    b) dodaj obsługę rysowania przez dotyk
17. KONIEC.