EX NO:3 **REGISTER NO:210701509**

DATE:

GRAPHICAL PRIMITIVES

AIM

Develop an android application to draw the circle, ellipse, rectangle and some text using Android Graphical primitives.

PROCEDURE:

- Open Android Studio and import the package
- In activity_main.xml drag and drop the buttons
- The button need to perform actions to change the colour, font size and background colour
- Click android virtual device that should control the toolbar
- Design the graphical layout with the textview and buttons
- Run the application
- The version of android and name is displayed
- The theme of the file is also mentioned in a file
- Run the file using the version which is displayed to the users.

PROGRAM CODE:

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools">

    <application
          android:allowBackup="true"
          android:dataExtractionRules="@xml/data_extraction_rules"
          android:fullBackupContent="@xml/backup_rules"
          android:icon="@mipmap/ic_launcher"
          android:label="@string/app_name"
          android:roundIcon="@mipmap/ic_launcher_round"
          android:supportsRtl="true"
          android:theme="@style/Theme.Shapes"
          tools:targetApi="31">
       <activity
             android:name=".MainActivity"
             android:exported="true">
          <intent-filter>
             <action android:name="android.intent.action.MAIN"/>

             <category android:name="android.intent.category.LAUNCHER"/>
          </intent-filter>
       </activity>
    </application>

</manifest>
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:ignore="MissingConstraints"

        tools:context=".MainActivity">

    <com.example.shapes.CanvasView
            android:id="@+id/canvasView"
            android:layout_width="match_parent"
            android:layout_height="match_parent" tools:layout_editor_absoluteY="0dp"
            tools:layout_editor_absoluteX="-16dp"
            android:background="@color/lavendar"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.kt

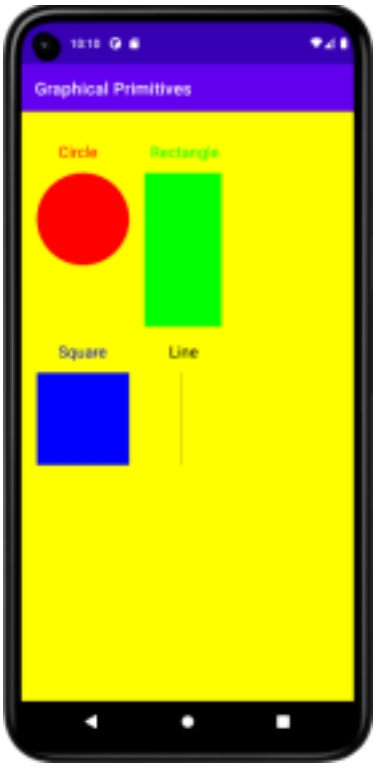
```
package com.example.shapes

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

CanvasView.kt

```
package com.example.shapes

import android.content.Context
import android.graphics.Canvas
import android.graphics.Color
import android.graphics.Paint
import android.util.AttributeSet
import android.view.View

class CanvasView @JvmOverloads constructor(
    context: Context, attrs: AttributeSet? = null, defStyleAttr: Int = 0
) : View(context, attrs, defStyleAttr) {

    override fun onDraw(canvas: Canvas) {
```

```
        super.onDraw(canvas)

        val paint:Paint=Paint()

        //line
        paint.setColor(Color.GREEN)
        paint.strokeWidth=8f
        canvas.drawLine(750f,800f,750f,1200f,paint)
        drawText(canvas, "Line", 690f, 750f)

        //circle
        paint.style=Paint.Style.FILL
        paint.setColor(Color.YELLOW)
        canvas.drawCircle(290f,350f,150f,paint)
        drawText(canvas, "Circle", 220f, 150f)

        //rectangle
        paint.style=Paint.Style.FILL
        paint.setColor(Color.RED)
        canvas.drawRect(850f,650f,600f,200f,paint)
        drawText(canvas, "Rectangle", 620f, 150f)


        //square
        paint.style=Paint.Style.FILL
        paint.setColor(Color.BLUE)
        canvas.drawRect(200f,1150f,500f,850f,paint)
        drawText(canvas, "Square", 250f, 750f)

    }

    private fun drawText(canvas: Canvas, s: String,x: Float, y: Float) {
        val textPaint = Paint().apply {
            color = Color.BLACK
            textSize = 50f
        }
        canvas.drawText(s, x, y, textPaint)
    }
}
```

OUTPUT:



RESULT: