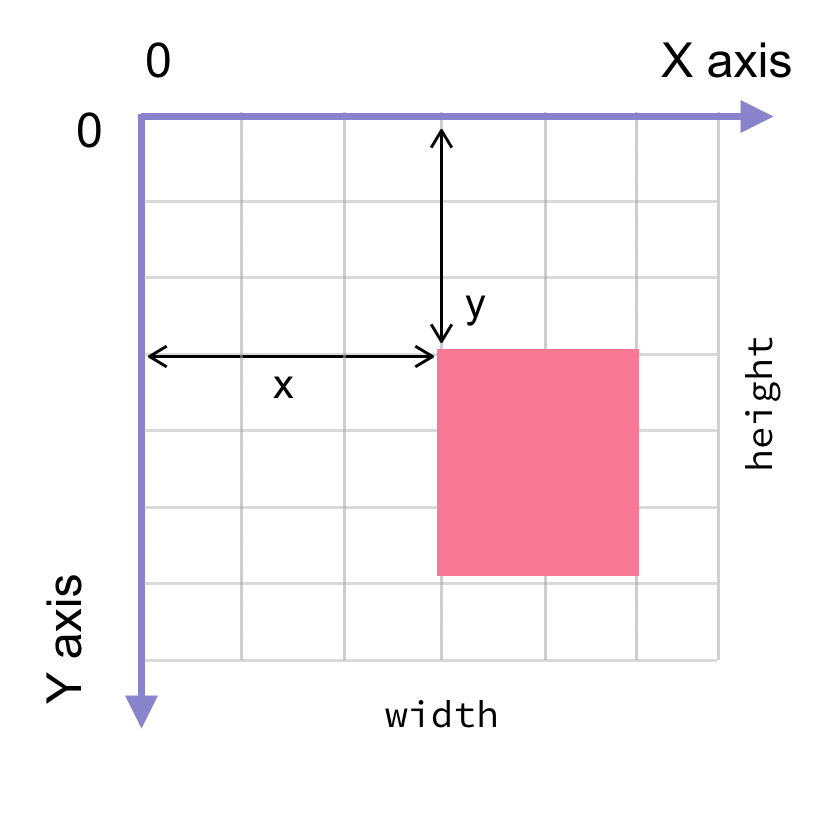
# Crtanje u Androidu

Za crtanje na CANVASU - platnu potrebno je:

* Bitmapa ili View— za držanje piksela na kojima će se platno nacrtati.
* Platno — za pokretanje naredbi za crtanje.
* Naredbe za crtanje — da naznačite platnu što treba nacrtati.
* Boja — da opišete kako nacrtati naredbe.

CANVAS KOORDINATNI SUSTAV



Primjer 1:

activity\_main.xlm

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:gravity="center"  
 android:orientation="vertical"**>  
 <**view  
 class="com.example.myapplication16.CustomView"  
  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"** />  
</**LinearLayout**>

MainActivity.java

**package** com.example.myapplication16;  
  
**import** android.app.Activity;  
**import** android.os.Bundle;  
  
**public class** MainActivity **extends** Activity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 }  
}

CustomView.java -krugovi

**package** com.example.myapplication16;  
**import** android.content.Context;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**import** android.util.AttributeSet;  
**import** android.view.View;  
  
**public class** CustomView **extends** View {  
 **public** CustomView(Context context, AttributeSet attr) {  
 **super**(context);  
 }  
  
 @Override  
 **public void** onDraw(Canvas canvas){  
  
 canvas.drawColor(Color.***DKGRAY***);  
  
 Paint redCirclePaint = **new** Paint();  
 redCirclePaint.setColor(Color.*rgb*(255, 0, 0));  
  
 Paint greenCirclePaint = **new** Paint();  
 greenCirclePaint.setColor(Color.*rgb*(0, 255, 0));  
  
 Paint blueCirclePaint = **new** Paint();  
 blueCirclePaint.setColor(Color.*rgb*(0, 0, 255));  
  
  
  
 canvas.drawCircle(800, 500, 200, redCirclePaint);  
 canvas.drawCircle(400, 200, 100, redCirclePaint);  
 canvas.drawCircle(325, 900, 300, greenCirclePaint);  
 canvas.drawCircle(800, 900, 100, greenCirclePaint);  
 canvas.drawCircle(900, 1600, 400, blueCirclePaint);  
 }  
}

Primjer 2: MainActivity i activity\_main ostaju isti

CustomView.java – pravokutnik, kvadrat, crta

**package** com.example.myapplication16;  
**import** android.content.Context;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**import** android.util.AttributeSet;  
**import** android.view.View;  
  
**public class** CustomView **extends** View {  
 **public** CustomView(Context context, AttributeSet attr) {  
 **super**(context);  
 }  
  
 @Override  
 **public void** onDraw(Canvas canvas){  
  
 canvas.drawColor(Color.***LTGRAY***);  
  
 Paint paint = **new** Paint();  
 paint.setColor(Color.***BLUE***);  
 paint.setTextSize(50);  
 Paint paintC =**new** Paint(); *//color circle* paintC.setColor(Color.***RED***);  
 paintC.setTextSize(50);  
 Paint paintS= **new** Paint();  
 paintS.setColor(Color.***MAGENTA***);  
 paintS.setTextSize(50);  
 *//a Rectangle* canvas.drawText(**"Rectangle"**, 420, 150, paint);  
 canvas.drawRect(400, 200, 650, 700, paint);  
  
 *//a Circle* canvas.drawText(**"Circle"**, 120, 150, paintC);  
 canvas.drawCircle(200, 350, 150, paintC);  
  
 *//a Square* canvas.drawText(**"Square"**, 120, 800, paintS);  
 canvas.drawRect(50, 850, 350, 1150, paintS);  
  
 *//a vertical Line* paint.setStrokeWidth(2);  
 canvas.drawText(**"VLine"**, 480, 800, paint);  
  
 paint.setStyle(Paint.Style.***STROKE***);  
 paint.setStrokeWidth(10);  
 canvas.drawLine(520, 850, 520, 1150, paint);  
  
 *//a horizontal line* paint.setStrokeWidth(2);  
 canvas.drawText(**"HLine"**, 100, 1400, paint);  
  
 paint.setStrokeWidth(10);  
 canvas.drawLine(50, 1500, 1000, 1500, paint);  
  
 }  
}

## Paint i Canvas

* **mPaint.setAntiAlias(true);** //Anti-aliasing kistovi -metoda uklanjanja neravnina
* **mPaint.setColor(Color.WHITE);** //Podesite boju kista
* **mPaint.setStyle(Paint.Style.FILL);** //Podesite vrstu grafičke ispune koja se iscrtava,
* FILL je unutarnje popunjenje, STROKE je samo obrub, sadržaj nije ispunjen.
* **mPaint.setStrokeWidth(mDensity \* 2);** //Postavite širinu obruba. Na primjer, širina pravokutnika, širina teksta. Argumenti su jedinice piksela.
* **mPaint.setTextSize(mDensity \* 20);** //Postavite veličinu fonta prilikom crtanja teksta

## Paint - način ispune

* **Paint.Style.FILL** ispunjava interijer
* **Paint.Style.FILL\_AND\_STROKE** Ispuna i potez
* **Paint.Style.STROKE** samo skice bez ispune

## Podešavanje boje pozadine

* canvas.drawColor(Color.BLUE);
* canvas.drawRGB(255, 255, 0);

## Crtanje linije

* drawLine (float startX, float startY, float stopX, float stopY, Paint paint)

Opis parametara:

* startX: početna X koordianata
* startY: početna Y koordinata
* stopX: početna X koordinata
* stopY: završna Y koordinata

## Crtanje točke

Jedna točka

* drawPoint (float x, float y, Paint paint)

Više točaka

* drawPoints (float[] pts, Paint paint)
* drawPoints (float[] pts, int offset, int count, Paint paint)

canvas.drawPoint(100, 100, paint);

float []pts={10,10,100,100,200,200,400,400};

canvas.drawPoints(pts, 2, 4, paint);

## Crtanje pravokutnika

* drawRect (float left, float top, float right, float bottom, Paint paint)
* drawRect (RectF rect, Paint paint)
* drawRect (Rect rect, Paint paint)

## Crtanje pravokutnika sa zaobljenim vrhovima

* drawRoundRect (RectF rect, float rx, float ry, Paint paint)

### Crtanje kruga

* drawCircle (float cx, float cy, float radius, Paint paint)

Opis parametara

* cx: x koordinata središta kružnice
* cy: y koordinata središta kružnice
* Radius: radijus kruga

### Crtanje elipse

* drawOval (RectF oval, Paint paint)

Elipsa se generira iz pravokutnika, a lik elipse se uspostavlja uzimanjem osi X pravokutnika kao velika os elipse, a os Y pravokutnika kao mala os elipse.

## Crtanje lukova

Luk je dio elipse, a elipsa dio pravokutnika onda je i luk dio pravokutnika.

* drawArc (RectF oval, float startAngle, float sweepAngle, boolean useCenter, Paint paint)

## Crtanje ravne linije s putanjom -path

Opis funkcija

* Move To (float x, float y): premjestiti određenu točku na sljedeću poziciju
* lineTo(float x, float y): povezati početnu točku i sljedeću poziciju u stazu
* Close: Ako se krajnja točka zadnje staze puta ne podudara s početnom točkom, tada pozivanje close() povezuje početnu i završnu točku kako bi se formiralo zatvaranje

Path path = new Path();

//postavlja početnu točku

path.moveTo(10\*mDensity, 10\*mDensity);

//završetak prve crte je ujedno početak druge

path.lineTo(10\*mDensity, 100\*mDensity);

//Crta drugu liniju

path.lineTo(200\*mDensity,100\*mDensity);

//povezuje crte u zatvorenu cjelinu

path.close();

canvas.drawPath(path, mPaint);

Primjer 3: - trokuti

CustomView.java

**package** com.example.myapplication16;  
**import** android.content.Context;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**import** android.graphics.Path;  
**import** android.graphics.RectF;  
**import** android.util.AttributeSet;  
**import** android.view.View;  
  
**public class** CustomView **extends** View {  
 **public** CustomView(Context context, AttributeSet attr) {  
 **super**(context);  
 }  
  
 @Override  
 **public void** onDraw(Canvas canvas){  
  
 *//canvas.drawColor(Color.YELLOW);* Paint paint = **new** Paint();  
 paint.setColor(Color.***BLUE***);  
 paint.setTextSize(50);  
  
 Path path=**new** Path();  
 path.moveTo(800,800);  
 path.lineTo(800,1200);  
 path.lineTo(100,1200);  
 path.close();  
 canvas.drawPath(path,paint);  
 path.moveTo(50,1150);  
 path.lineTo(50,750);  
 path.lineTo(750,750);  
 path.close();  
 canvas.drawPath(path,paint);  
 }  
}

Primjer 4 – pravokutnici

CustomView.java

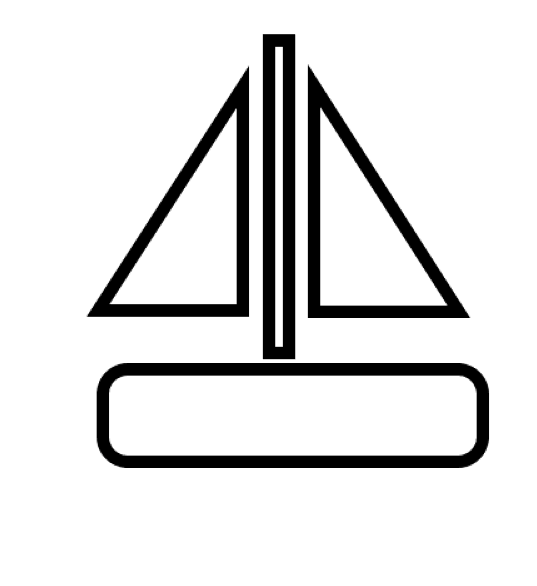
**package** com.example.myapplication16;  
**import** android.content.Context;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**import** android.graphics.Path;  
**import** android.graphics.RectF;  
**import** android.util.AttributeSet;  
**import** android.view.View;  
  
**public class** CustomView **extends** View {  
 **public** CustomView(Context context, AttributeSet attr) {  
 **super**(context);  
 }  
  
 @Override  
 **public void** onDraw(Canvas canvas){  
  
 *//canvas.drawColor(Color.YELLOW);* Paint paint = **new** Paint();  
 paint.setColor(Color.***BLUE***);  
 paint.setTextSize(50);  
 canvas.drawRect(0,0,200,200,paint);  
 paint.setStyle(Paint.Style.***STROKE***);  
 paint.setStrokeWidth(12);  
 canvas.drawRect(**new** RectF(0,300,200,500),paint);  
  
 canvas.drawRoundRect(**new** RectF(0,700,800,900),50,50,paint);  
  
 }  
}

Primjer 5– lukovi

CustomView.java

**package** com.example.myapplication16;  
**import** android.content.Context;  
**import** android.graphics.Canvas;  
**import** android.graphics.Color;  
**import** android.graphics.Paint;  
**import** android.graphics.Path;  
**import** android.graphics.RectF;  
**import** android.util.AttributeSet;  
**import** android.view.View;  
  
**public class** CustomView **extends** View {  
 **public** CustomView(Context context, AttributeSet attr) {  
 **super**(context);  
 }  
  
 @Override  
 **public void** onDraw(Canvas canvas){  
  
 *//canvas.drawColor(Color.YELLOW);* Paint paint = **new** Paint();  
 paint.setColor(Color.***BLUE***);  
  
 RectF rectF\_Arc1= **new** RectF (0,10,600,400);  
 canvas.drawArc(rectF\_Arc1,0,-120,**true**,paint);  
 RectF rectF\_Arc2= **new** RectF(0,310, 600,700);  
 canvas.drawArc(rectF\_Arc2,0,-120,**false**,paint);  
 paint.setStyle(Paint.Style.***STROKE***);  
 RectF rectF\_Arc3=**new** RectF(-200,600,600,1100);  
 paint.setStrokeWidth(12);  
 canvas.drawArc(rectF\_Arc3,0,-120,**false**,paint);  
  
 }  
}

Zadatak za DZ:



FIUMANKA

Pomoć:

Paint paint = **new** Paint();  
paint.setColor(Color.***BLACK***);  
paint.setTextSize(50);  
Path path =**new** Path();  
path.moveTo(700,500);  
path.lineTo(700,1000);  
path.lineTo(400,1000);  
path.close();  
paint.setStyle(Paint.Style.***STROKE***);  
paint.setStrokeWidth(5);  
canvas.drawPath(path,paint);