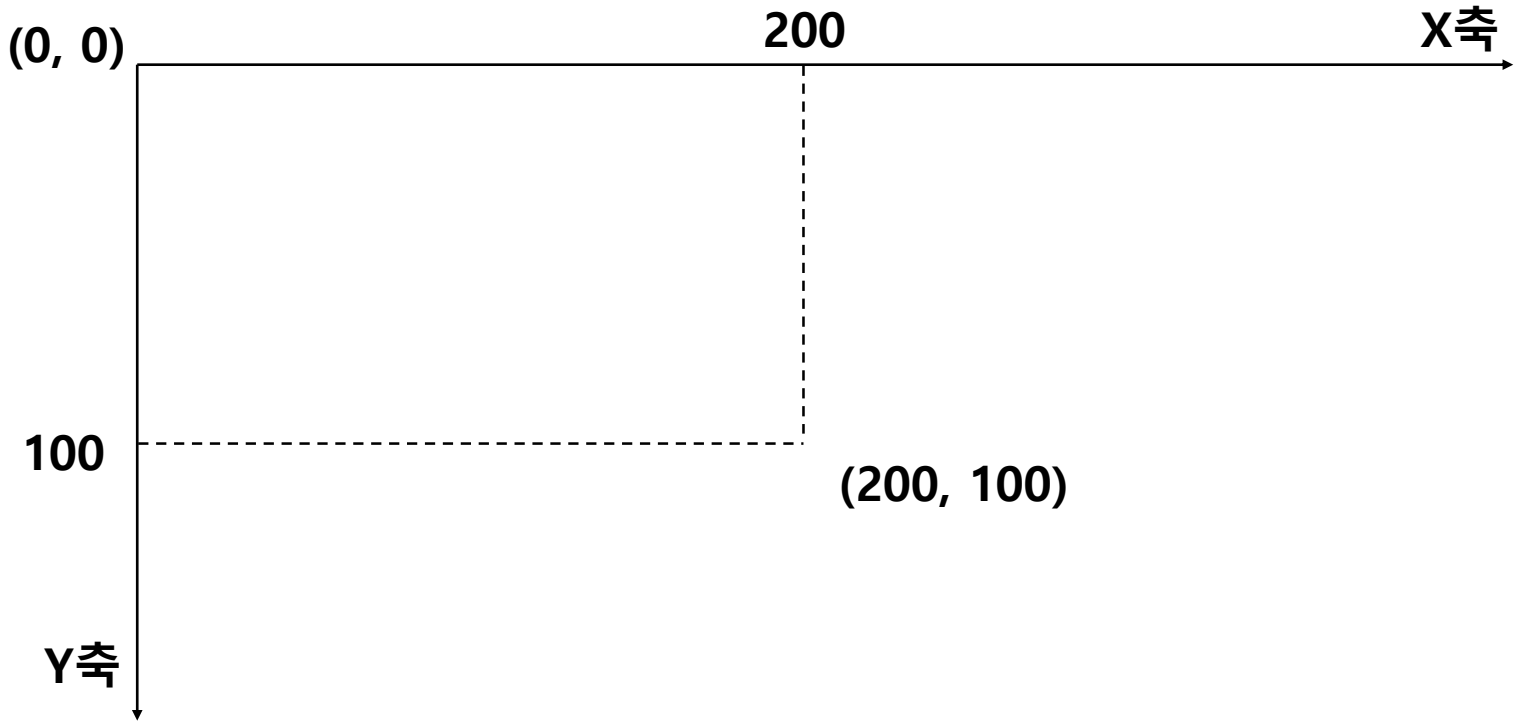
06

그래픽

Graphic

자바 좌표 시스템



자바클래스

- 자바에서 색을 나타내기 위해 Color 클래스의 객체를 사용한다.
- Color 클래스는 정의된 색들을 포함한다

색	객체	RGB 값
흑색	Color.black	0, 0, 0
녹색	Color.green	0, 255, 0
노란색	Color.yellow	255, 255, 0
백색	Color.white	255, 255, 255

색상 지정 : Color 클래스

- Color 클래스가 제공하는 대표적인 상수

상수	RGB 값
static Color black(또는 BLACK)	(0, 0, 0)
static Color blue(또는 BLUE)	(0, 0, 255)
static Color darkgray(또는 DARK_GRAY)	(64, 64, 64)
static Color lightGray(또는 LIGHT_GRAY)	(192, 192, 192)
static Color green(또는 GREEN)	(0, 255, 0)
static Color red(또는 RED)	(255, 0, 0)
static Color white(또는 WHITE)	(255, 255, 255)

```
Color(float r, float g, float b)
Color(float r, float g, float b, float a)
Color(int r, int g, int b)
Color(int r, int g, int b, int a)
Color(int rgb)
Color(int rgba, boolean hasalpha)
```

도형그리기

- **그래픽스 클래스의 메소드의 매개변수들은 좌표들과 크기들을 나타낸다.**
- **모양의 내부를 채울 수 있거나 채우지 않을 수도 있다.**
- **채워진 모양은 내부가 현재 색으로 채워진다.**

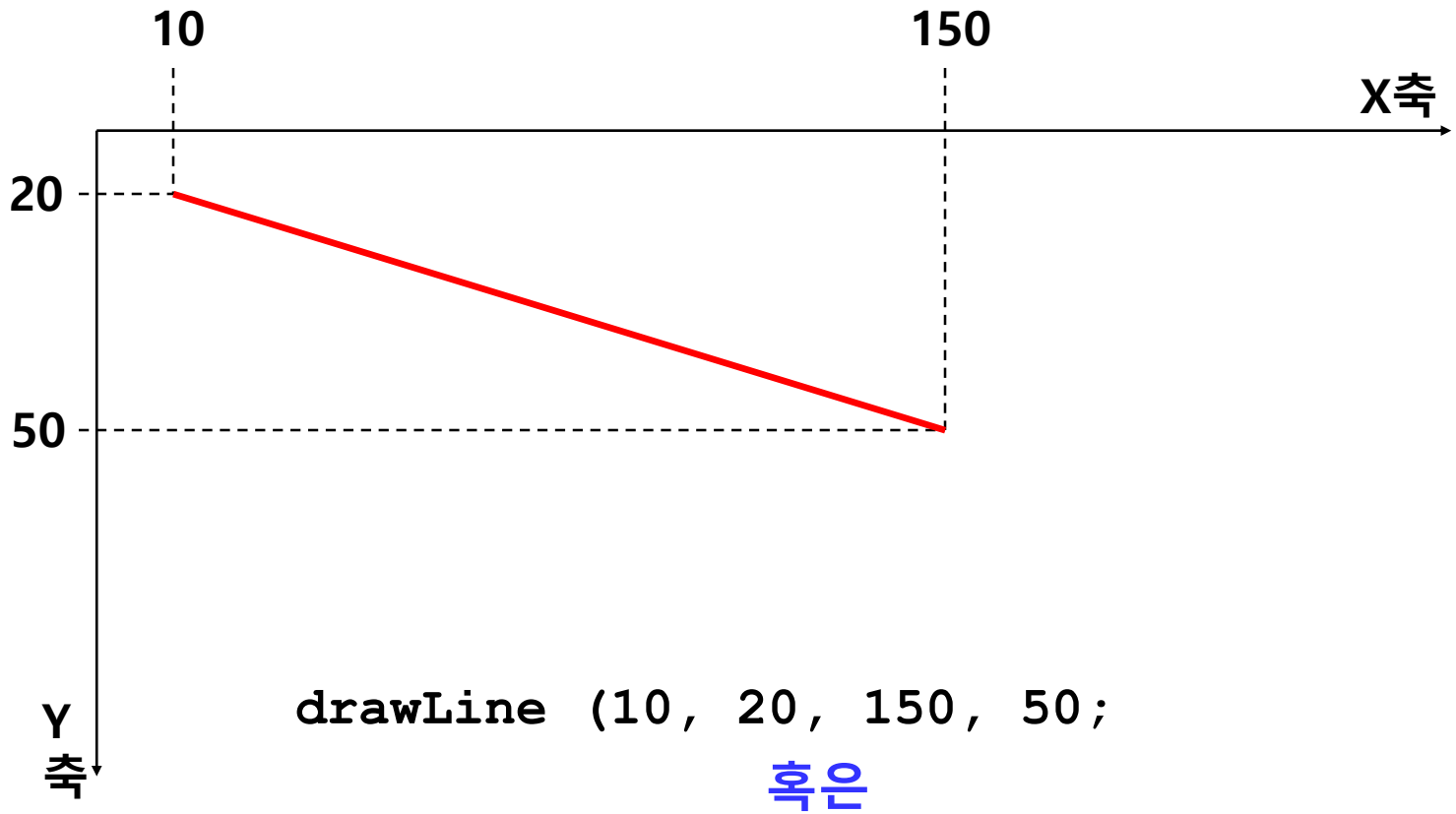
- **setBackground**
  - 모든 그리는 화면은 바탕색을 가진다. 바탕색을 정하는 메소드는 setBackground이다.

```
void setBackground(Color c)
void setForeground(Color c)
```

- **setColor**
  - 모든 도형의 내부는 전경 색(foreground color)을 가진다. 전경 색을 정하는 메소드는 setColor이다.

```
void setColor(Color c)
```

선그리기

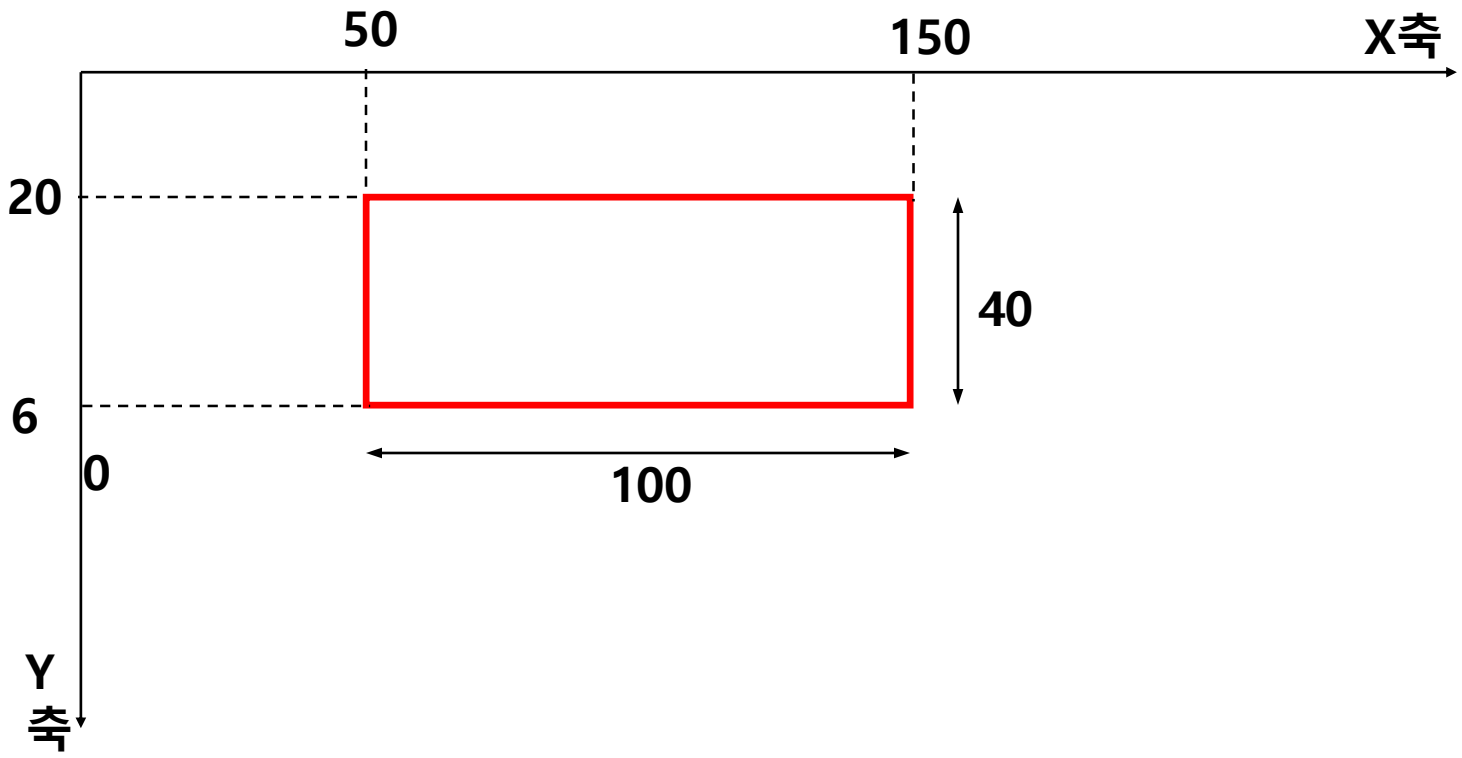



```
drawLine (10, 20, 150, 50;
```

혹은

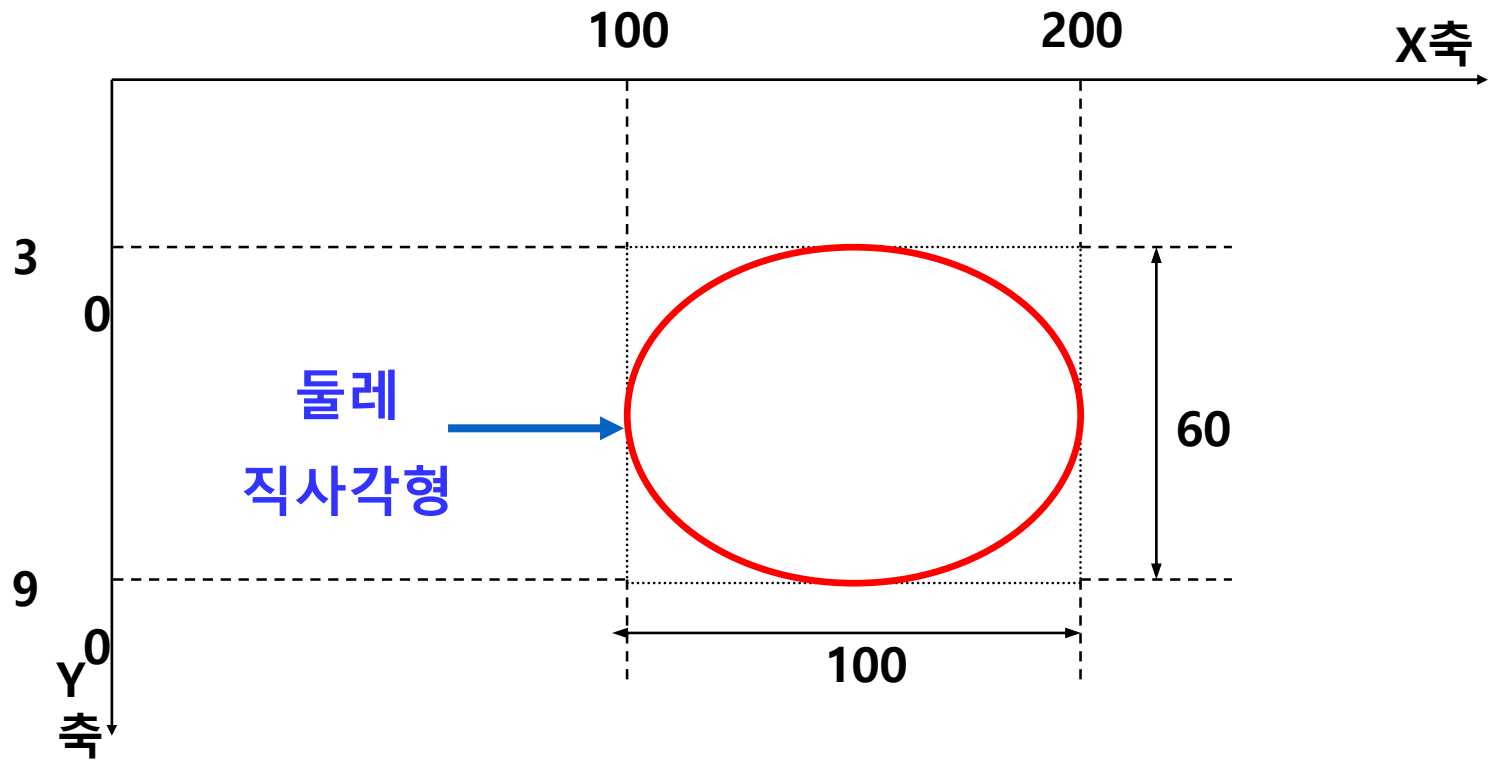
```
drawLine (150, 50, 10, 20);
```

직사각형 그리기



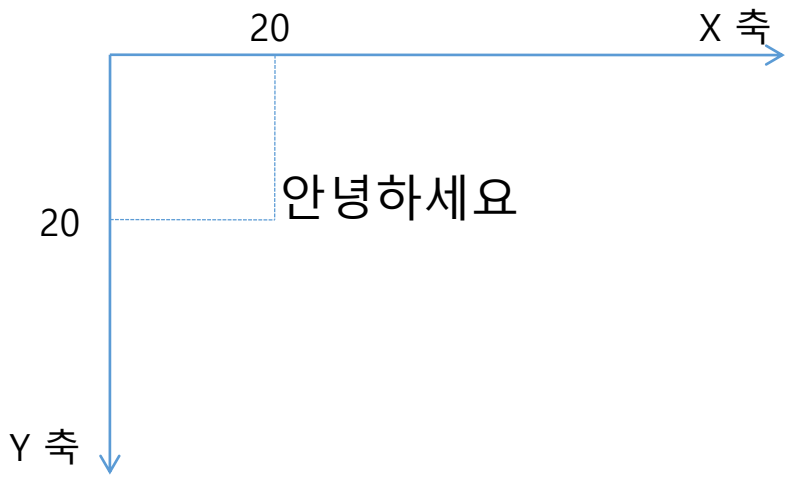

```
drawRect(50, 20, 100, 40);
```

타원그리기




```
drawOval(100, 30, 100, 60);
```

문자열 쓰기




```
drawString("안녕하세요", 20, 20);
```

■ 폰트 지정 : Font 클래스

● 생성자

```
// 지정한 폰트 이름, 폰트 스타일, 폰트 크기를 사용해 폰트 객체를 생성한다.
Font(String name, int style, int size)
```

● Font 설정

```
void setFont(Font font)
```

● Font 클래스가 제공하는 상수

상수		설명
폰트 이름	String DIALOG	대화상자에서 주로 사용하는 폰트이다.
	String MONOSPACED	고정 폭을 가지는 폰트이다.
	String SERIF	삐침이 있는 가변 폭의 폰트이다.
	String SANS_SERIF	삐침이 없는 가변 폭의 폰트이다.
폰트	int BOLD	굵은체이다.
스타일	int ITALIC	이탤릭체이다.
	int PLAIN	일반 폰트이다.

도형 그리기와 칠하기

도형 그리기

- 선, 타원, 사각형, 둥근 모서리 사각형, 원호, 폐 다각형

Graphics의 메소드

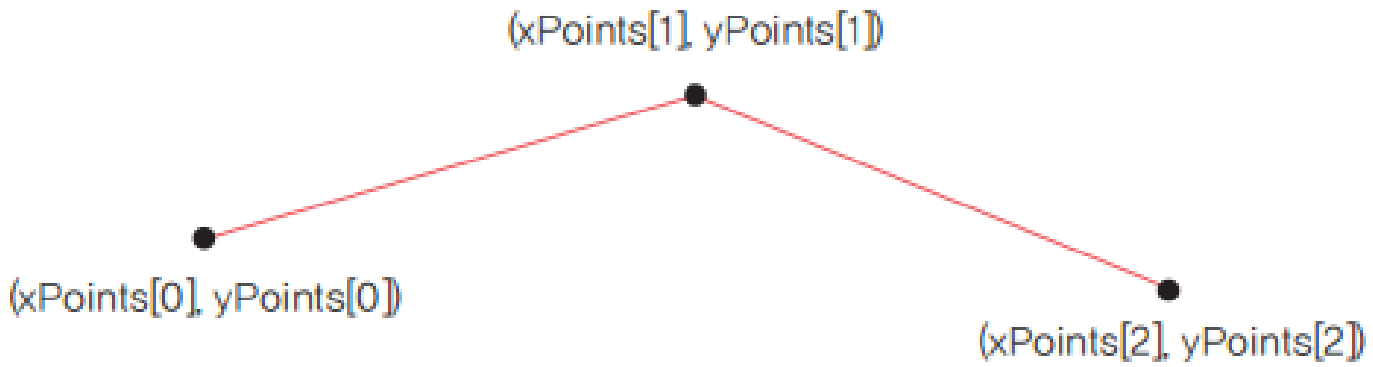
```
void drawLine(int x1, int y1, int x2, int y2)
    (x1, y1)에서 (x2, y2)까지 선을 그린다.
void drawOval(int x, int y, int w, int h)
    (x, y)에서 w x h 크기의 사각형에 내접하는 타원을 그린다.
void drawRect(int x, int y, int w, int h)
    (x, y)에서 w x h 크기의 사각형을 그린다.
void drawRoundRect(int x, int y, int w, int h, int arcWidth, int arcHeight)
• arcWidth: 모서리 원의 수평 반지름
• arcHeight: 모서리 원의 수직 반지름
    (x, y)에서 w x h 크기의 사각형을 그리되, 4개의 모서리는 arcWidth와 arcHeight를 이용하
    여 원호로 그린다.
```

그래픽 그리기

도형 그리기

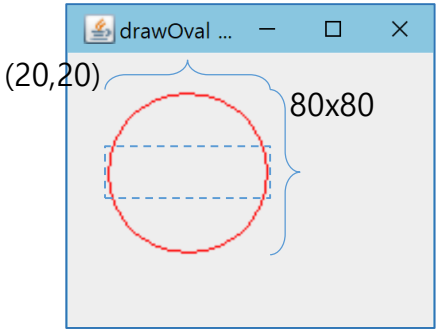
- 직선

```
void drawLine(int x1, int y1, int x2, int y2)
void drawPolyline(int[] xPoints, int[] yPoints, int nPoints)
```

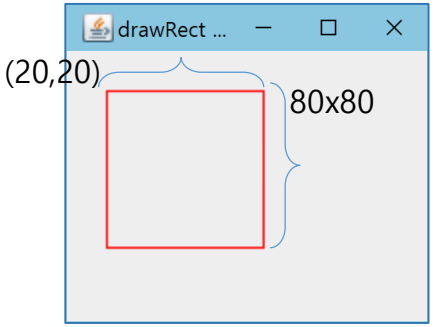


다른 도형 그리기 사례

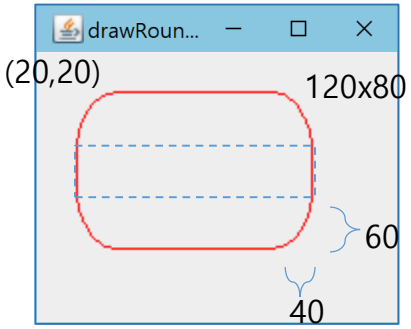

```
class MyPanel extends JPanel {
  public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setColor(Color.RED);
    g.drawOval(20,20,80,80);
  }
}
```




```
class MyPanel extends JPanel {
  public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setColor(Color.RED);
    g.drawRect(20,20,80,80);
  }
}
```

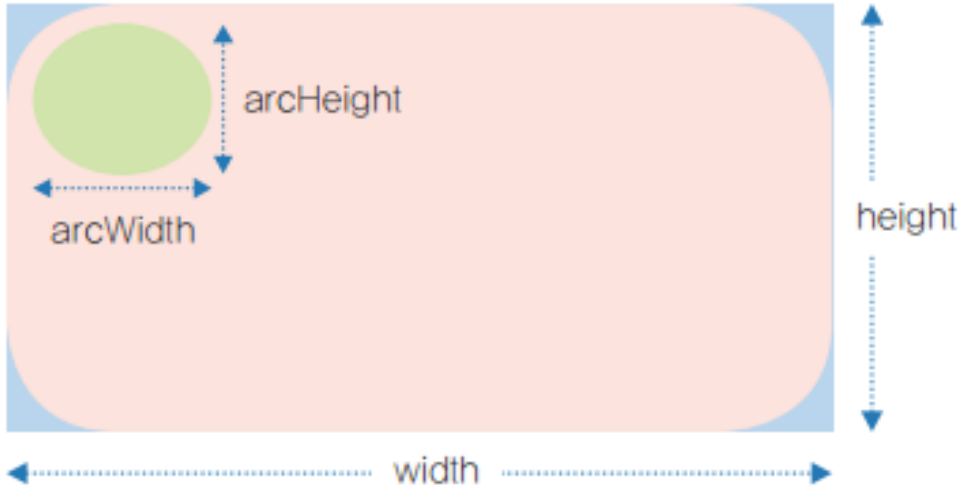



```
class MyPanel extends JPanel {
  public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setColor(Color.RED);
    g.drawRoundRect(20,20,120,80,40,60);
  }
}
```



그래픽 그리기


```
void drawRect(int x, int y, int width, int height)
void drawRoundRect(int x, int y, int width, int height, int arcWidth,
                   int arcHeight)
void draw3DRect(int x, int y, int width, int height, boolean raised)

void fillRect(int x, int y, int width, int height)
void fillRoundRect(int x, int y, int width, int height, int arcWidth,
                   int arcHeight)
void fill3DRect(int x, int y, int width, int height, boolean raised)
```



원호와 폐다각형 그리기

원호와 폐다각형 그리는 Graphics 메소드

void drawArc(int x, int y, int w, int h, int startAngle, int arcAngle)

- startAngle: 원호의 시작 각도
- arcAngle: 원호 각도

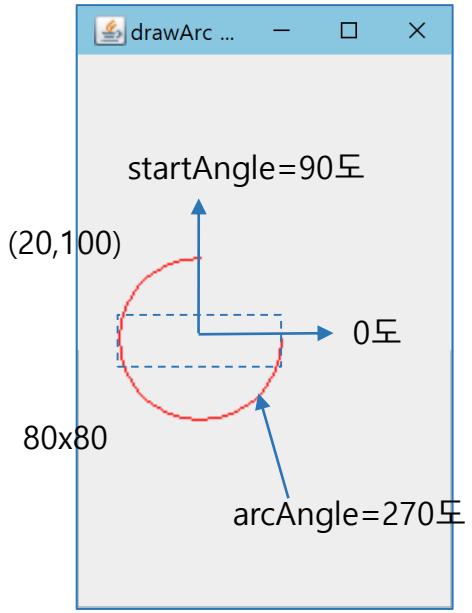
(x, y)에서 w x h 크기의 사각형에 내접하는 원호를 그린다. 3시 방향이 0도 기점이다. startAngle 지점에서 arcAngle 각도만큼 원호를 그린다. arcAngle이 양수이면 반시계 방향, 음수이면 시계 방향으로 그린다.

void drawPolygon(int [] x, int [] y, int n)

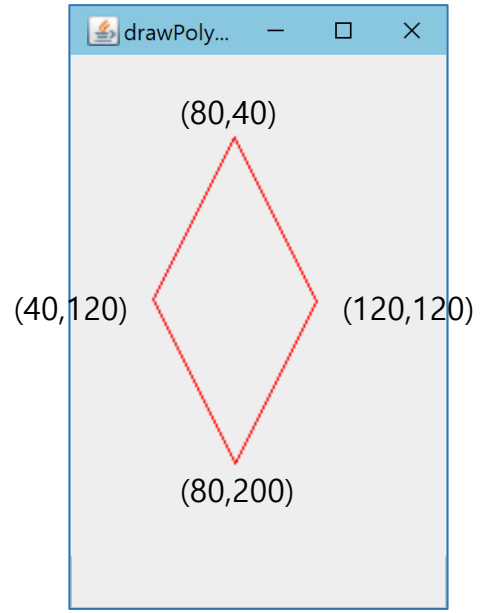
x, y 배열에 저장된 점들 중 n개를 연결하는 폐다각형을 그린다. (x[0], y[0]), (x[1], y[1]), ... , (x[n-1], y[n-1]), (x[0], y[0])의 점들을 순서대로 연결한다.

원호와 폐다각형 그리기 사례


```
class MyPanel extends JPanel {
  public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setColor(Color.RED);

    g.drawArc(20,100,80,80,90,270);
  }
}
```




```
class MyPanel extends JPanel {
  public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setColor(Color.RED);

    int []x = {80,40,80,120};
    int []y = {40,120,200,120};
    g.drawPolygon(x, y, 4);
  }
}
```

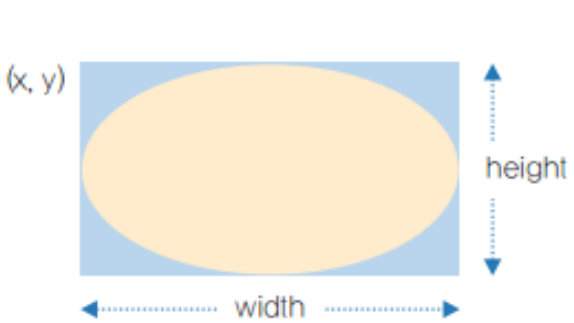


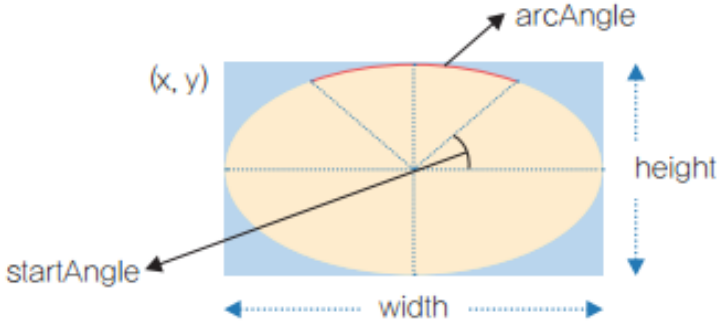
그래픽 그리기

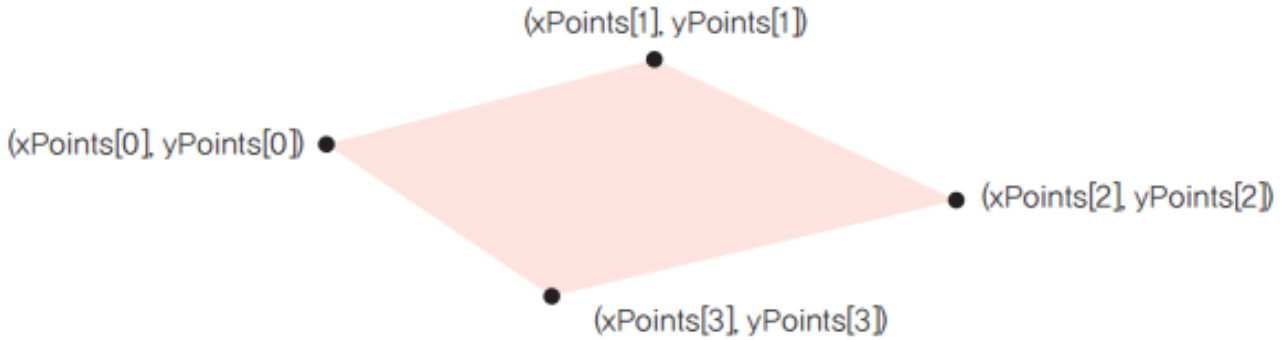
도형 그리기

- 타원, 호, 다각형

```
void drawOval(int x, int y, int width, int height)
void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)
void drawPolygon(int[] xPoints, int[] yPoints, int nPoints)

void fillOval(int x, int y, int width, int height)
void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle)
void fillPolygon(int[] xPoints, int[] yPoints, int nPoints)
```







도형 칠하기

도형 칠하기

- 도형을 그리고 내부를 칠하는 기능
- 도형의 외곽선과 내부를 따로 칠하는 기능은 없다.
- 도형 칠하기를 위한 메소드
  - 도형 그리기 메소드 명에서 draw 를 fill로 대치하면 된다. 인자는 동일
  - 예) `drawRect() -> fillRect(), drawArc() -> fillArc()`

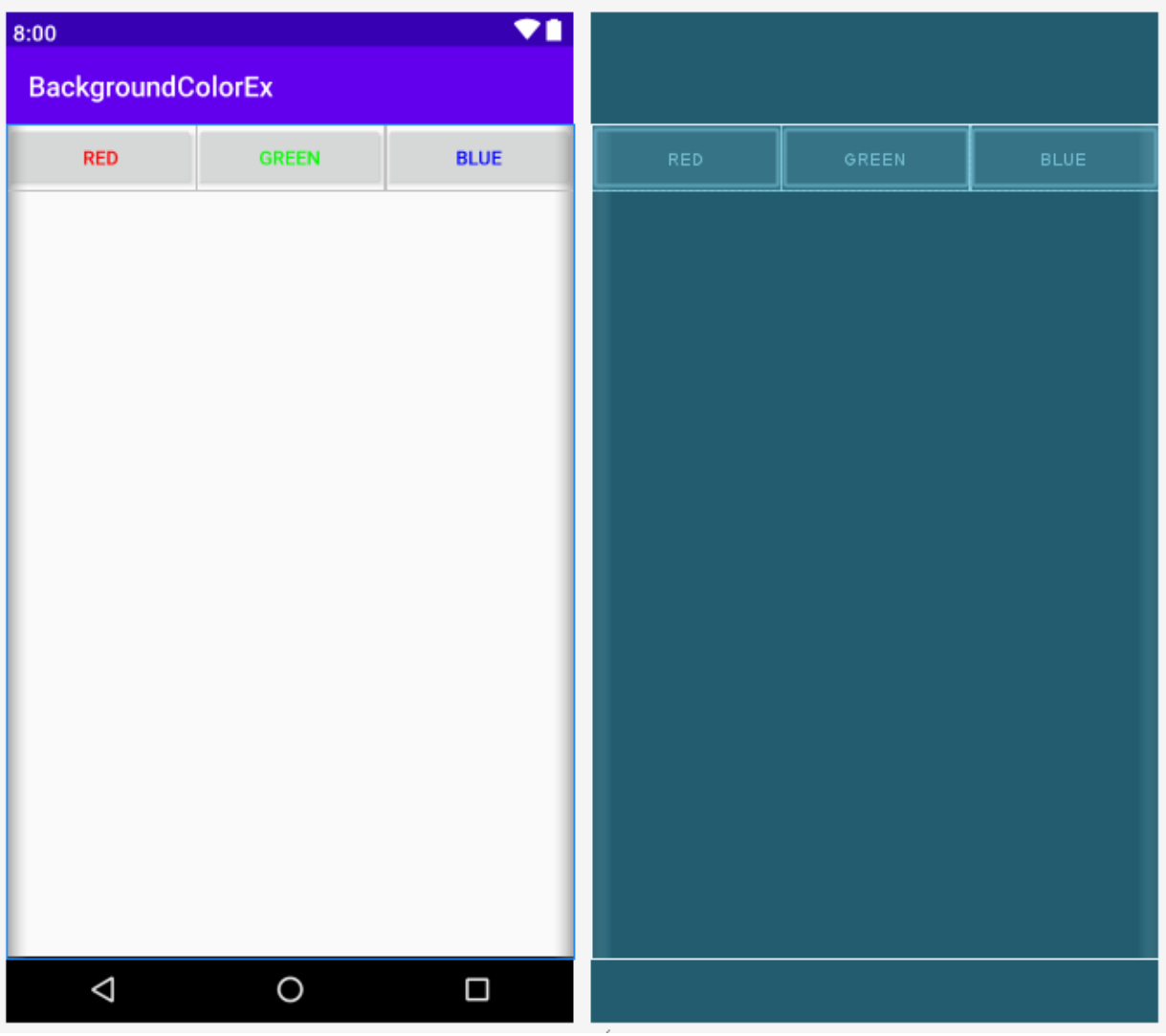
칠하기 메소드

- `void fillOval(int x1, int y1, int w, int h)`
- `void fillRect(int x1, int y1, int w, int h)`
- `void fillRoundRect(int x1, int y1, int w, int h, int arcWidth, int arcHeight)`
- `void fillArc(int x, int y, int w, int h, int startAngle, int arcAngle)`
- `void fillPolygon(int []x, int []y, int n)`

버튼클릭시 화면색 변경

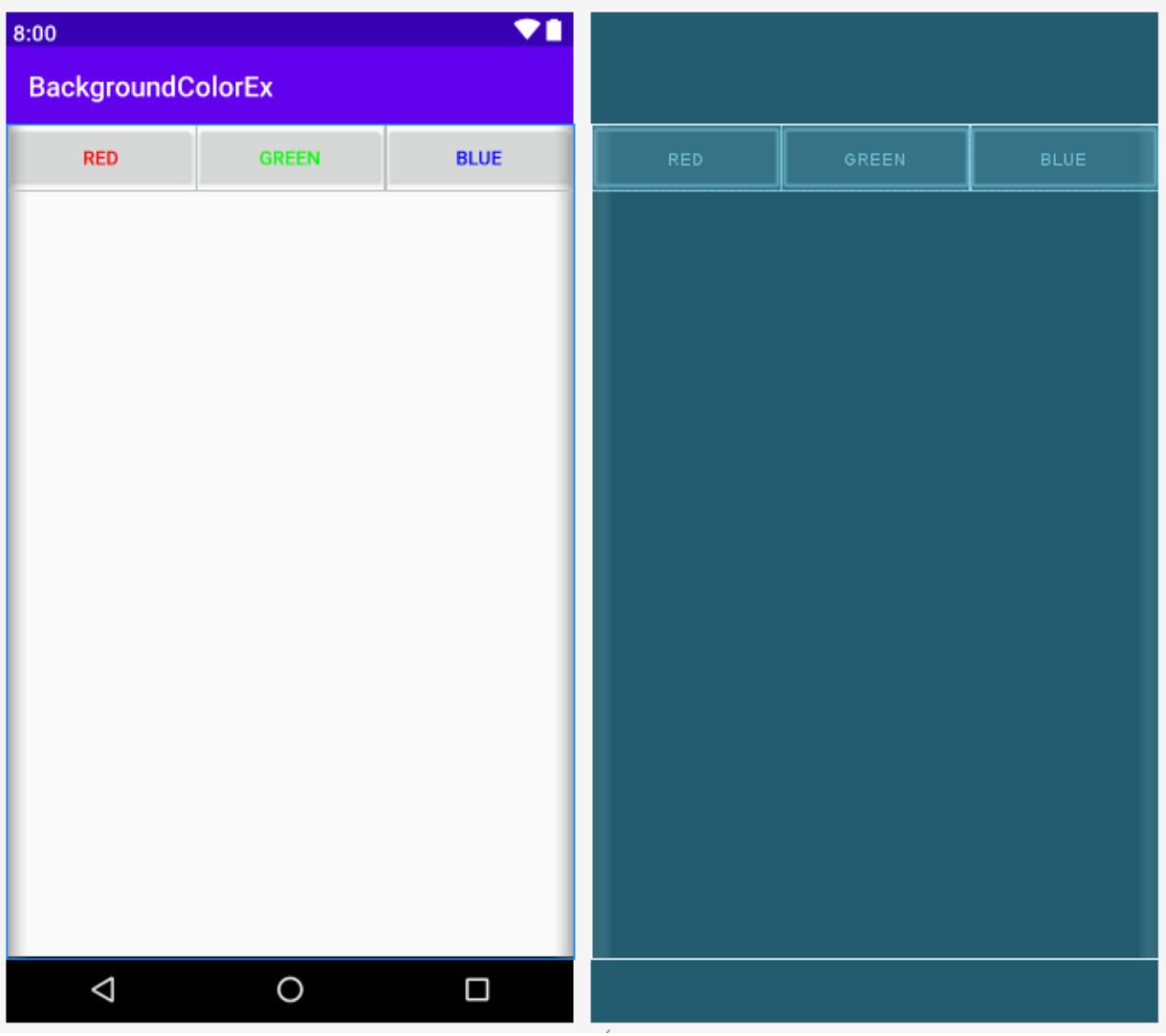

6:47
BackgroundColorEx
RED
GREEN
BLUE
6:47
BackgroundColorEx
RED
GREEN
BLUE
6:47
BackgroundColorEx
RED
GREEN
BLUE
6:47
BackgroundColorEx
RED
GREEN
BLUE

activity_main.xml




```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="horizontal"
    android:id="@+id/main"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btnRED"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="RED"
        android:onClick="onClick"
        android:textColor="#FF0000" />

    <Button
        android:id="@+id/btnGREEN"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="GREEN"
        android:onClick="onClick"
        android:textColor="#00FF00"/>

    <Button
        android:id="@+id/btnBLUE"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="BLUE"
        android:onClick="onClick"
        android:textColor="#0000FF" />
</LinearLayout>
```

activity_main.xml




```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="horizontal"
    android:id="@+id/main"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btnRED"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="RED"
        android:onClick="onClick"
        android:textColor="#FF0000" />

    <Button
        android:id="@+id/btnGREEN"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="GREEN"
        android:onClick="onClick"
        android:textColor="#00FF00"/>

    <Button
        android:id="@+id/btnBLUE"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="BLUE"
        android:onClick="onClick"
        android:textColor="#0000FF" />
</LinearLayout>
```

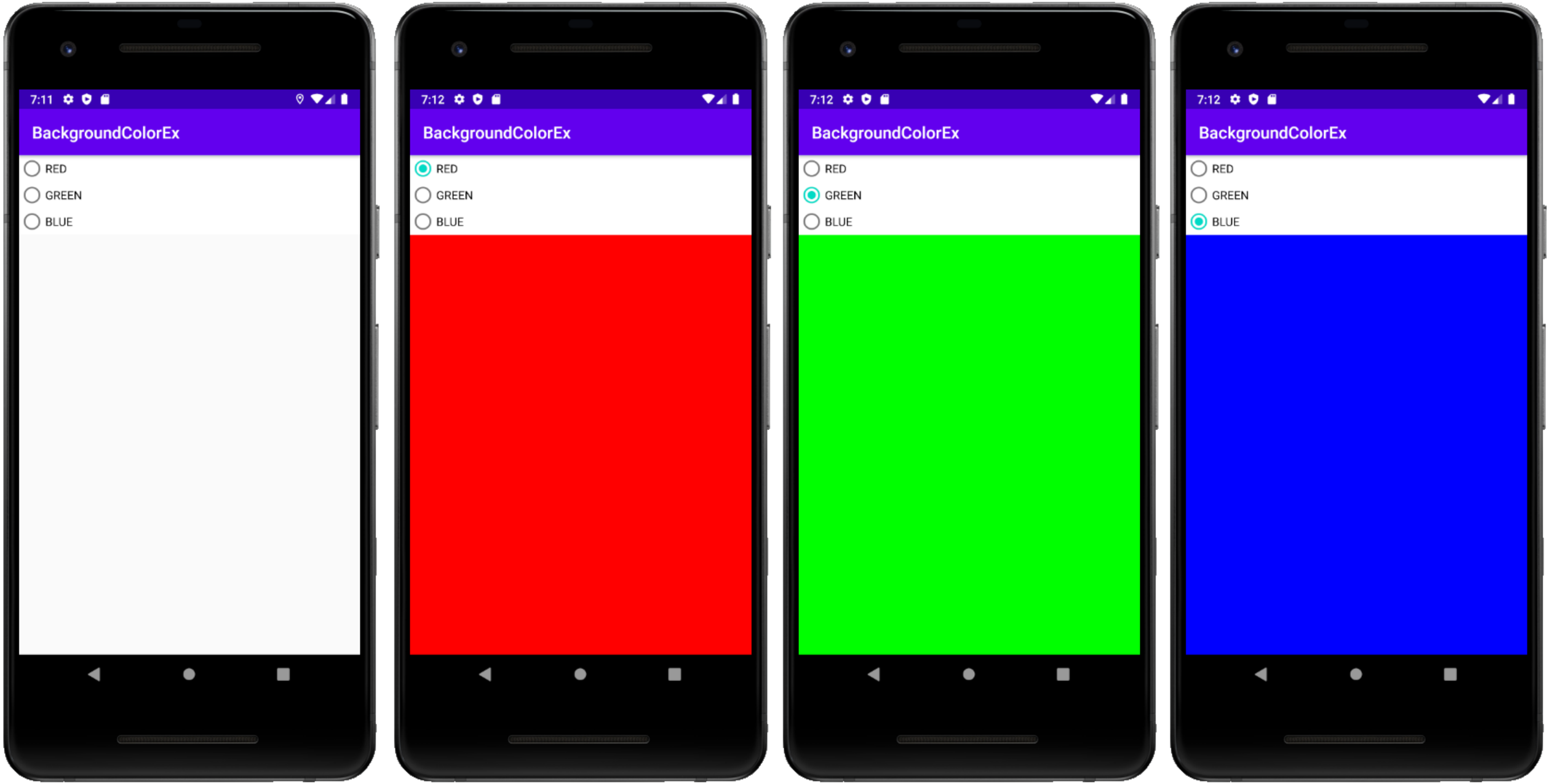
MainActivity.java


```
public class MainActivity extends AppCompatActivity {
    LinearLayout layout;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        layout = findViewById(R.id.main); //layout id 확인
    }

    public void onClick(View view) {
        switch (view.getId()) {
            case R.id.btnRED:
                layout.setBackgroundColor(Color.RED);
                break;
            case R.id.btnGREEN:
                layout.setBackgroundColor(Color.GREEN);
                break;
            case R.id.btnBLUE:
                layout.setBackgroundColor(Color.BLUE);
                break;
        }
    }
}
```

MainActivity.java

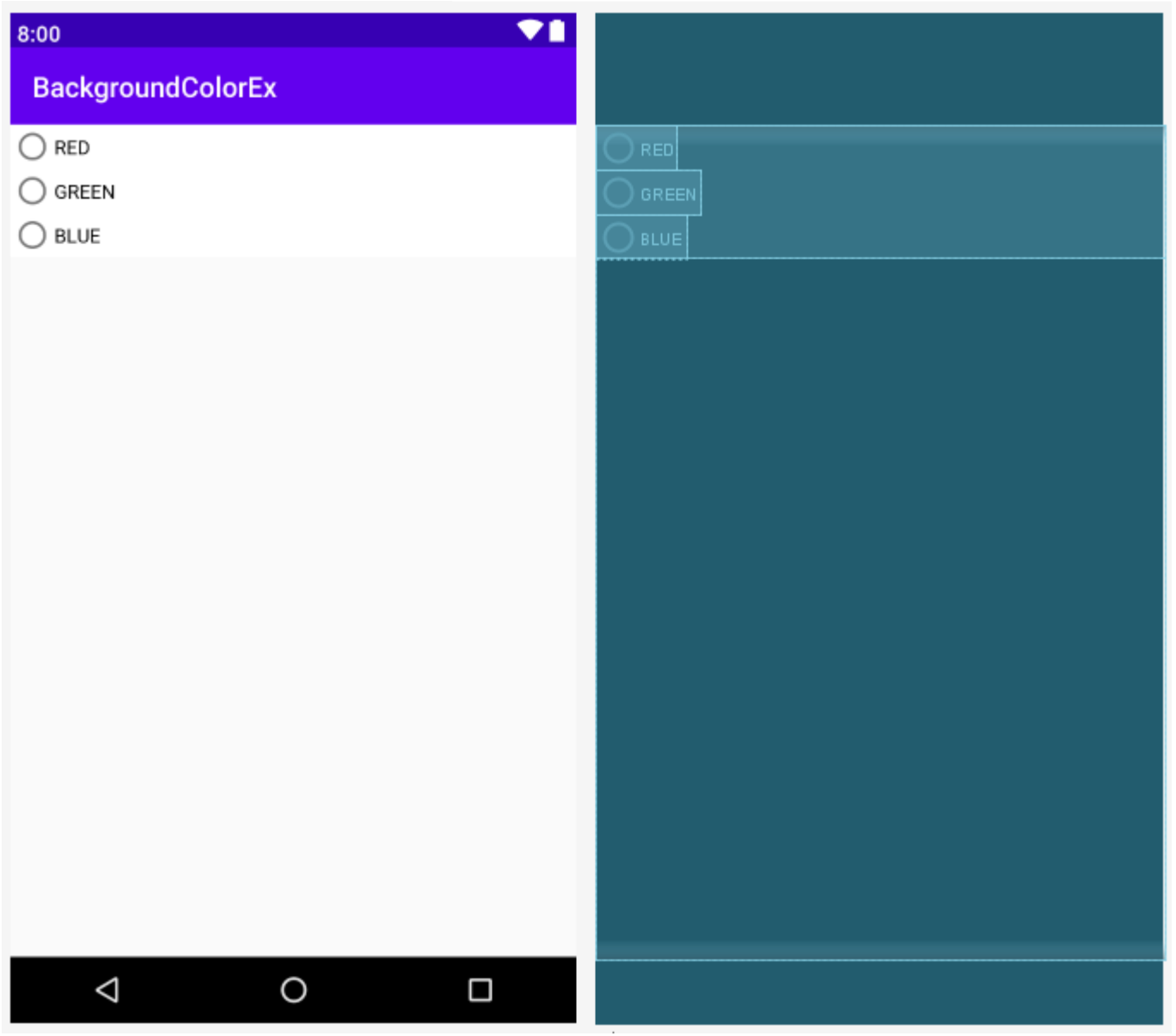

```
public class MainActivity extends AppCompatActivity {
    LinearLayout layout;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        layout = findViewById(R.id.main); //layout id 확인
    }

    public void onClick(View view) {
        switch (view.getId()) {
            case R.id.btnRED:
                layout.setBackgroundColor(Color.RED);
                break;
            case R.id.btnGREEN:
                layout.setBackgroundColor(Color.GREEN);
                break;
            case R.id.btnBLUE:
                layout.setBackgroundColor(Color.BLUE);
                break;
        }
    }
}
```

라디오버튼으로 화면색 변경


7:11
BackgroundColorEx
RED
GREEN
BLUE
7:12
BackgroundColorEx
RED
GREEN
BLUE
7:12
BackgroundColorEx
RED
GREEN
BLUE
7:12
BackgroundColorEx
RED
GREEN
BLUE

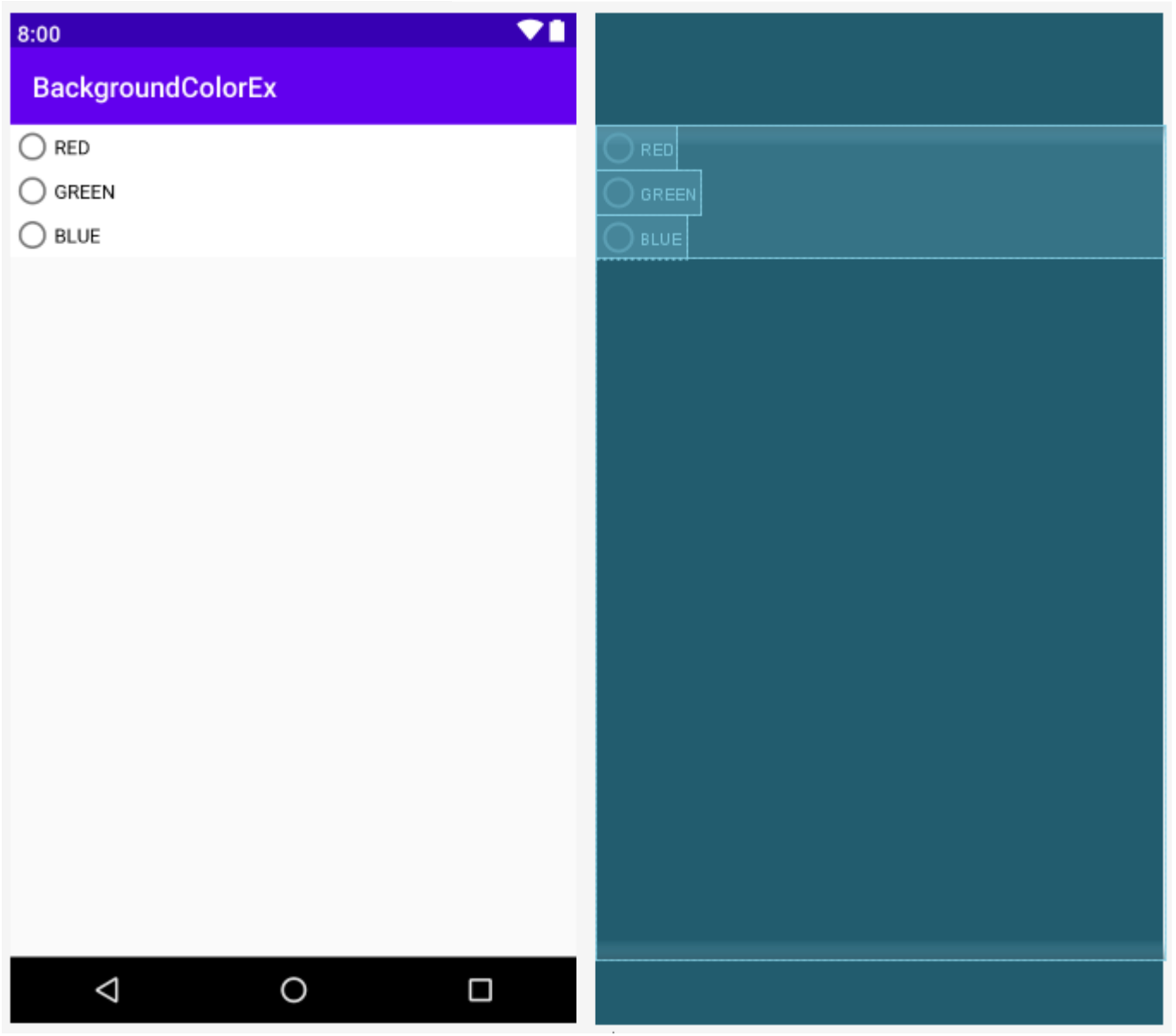
main_activity.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:id="@+id/main"
    tools:context=".MainActivity">

    <RadioGroup
        android:id="@+id/radiogroup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#FFFFFF">
        <RadioButton
            android:id="@+id/radioRed"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="RED" />
        <RadioButton
            android:id="@+id/radioGreen"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="GREEN" />
        <RadioButton
            android:id="@+id/radioBlue"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="BLUE" />
    </RadioGroup>
</LinearLayout>
```



main_activity.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:id="@+id/main"
    tools:context=".MainActivity">

    <RadioGroup
        android:id="@+id/radiogroup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#FFFFFF">
        <RadioButton
            android:id="@+id/radioRed"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="RED" />
        <RadioButton
            android:id="@+id/radioGreen"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="GREEN" />
        <RadioButton
            android:id="@+id/radioBlue"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="BLUE" />
    </RadioGroup>
</LinearLayout>
```



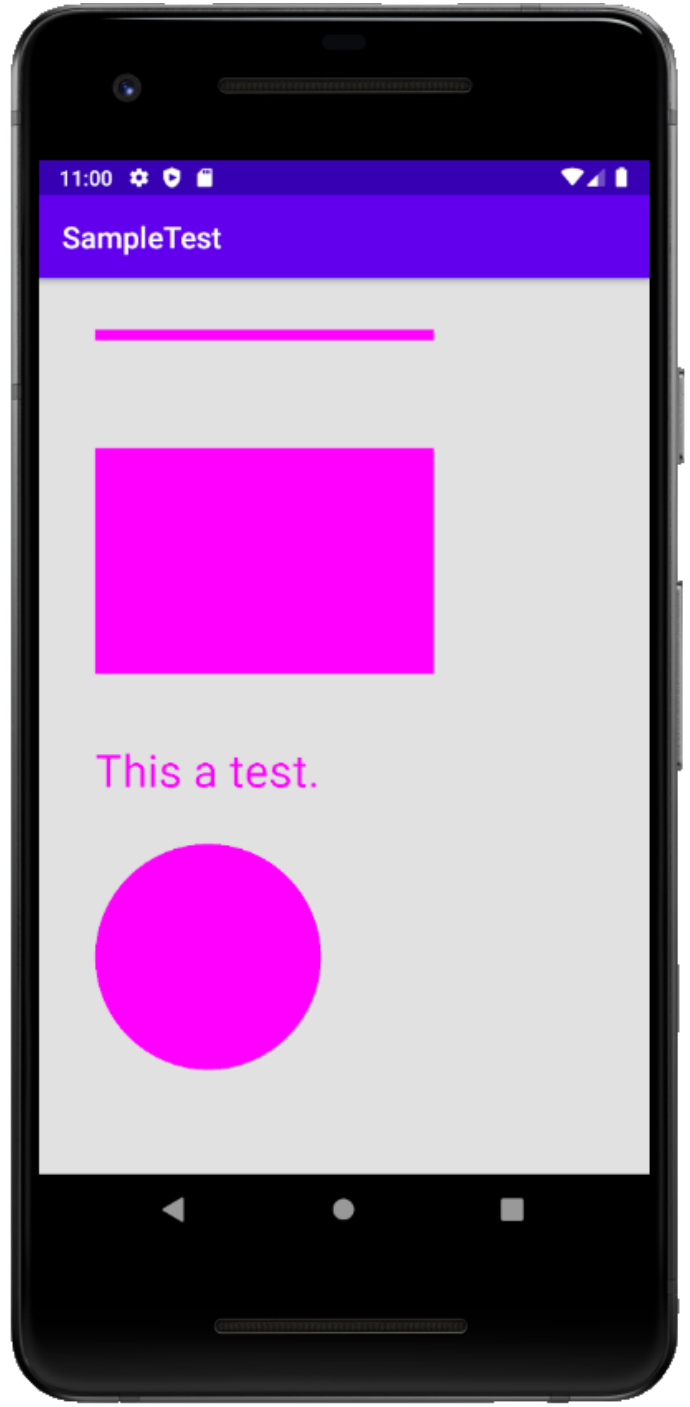
MainActivity.java


```
public class MainActivity extends AppCompatActivity {

    LinearLayout layout;
    RadioGroup radioGroup;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        layout = findViewById(R.id.main); //layout id 확인
        radioGroup = findViewById(R.id.radiogroup);
        radioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup group, int checkedId) {
                switch (checkedId){
                    case R.id.radioRed:
                        layout.setBackgroundColor(Color.RED);
                        break;
                    case R.id.radioGreen:
                        layout.setBackgroundColor(Color.GREEN);
                        break;
                    case R.id.radioBlue:
                        layout.setBackgroundColor(Color.BLUE);
                        break;
                }
            }
        });
    }
}
```

MainActivity.java


```
public class MainActivity extends AppCompatActivity {

    LinearLayout layout;
    RadioGroup radioGroup;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        layout = findViewById(R.id.main); //layout id 확인
        radioGroup = findViewById(R.id.radiogroup);
        radioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup group, int checkedId) {
                switch (checkedId){
                    case R.id.radioRed:
                        layout.setBackgroundColor(Color.RED);
                        break;
                    case R.id.radioGreen:
                        layout.setBackgroundColor(Color.GREEN);
                        break;
                    case R.id.radioBlue:
                        layout.setBackgroundColor(Color.BLUE);
                        break;
                }
            }
        });
    }
}
```

도형 그리기
(MainActivity)


11:00
SampleTest
This a test.

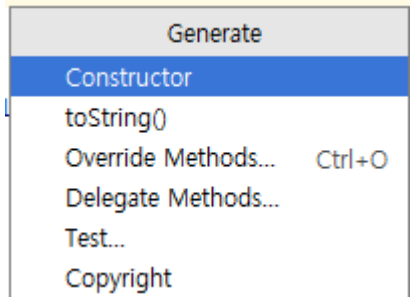
MainActivity.java

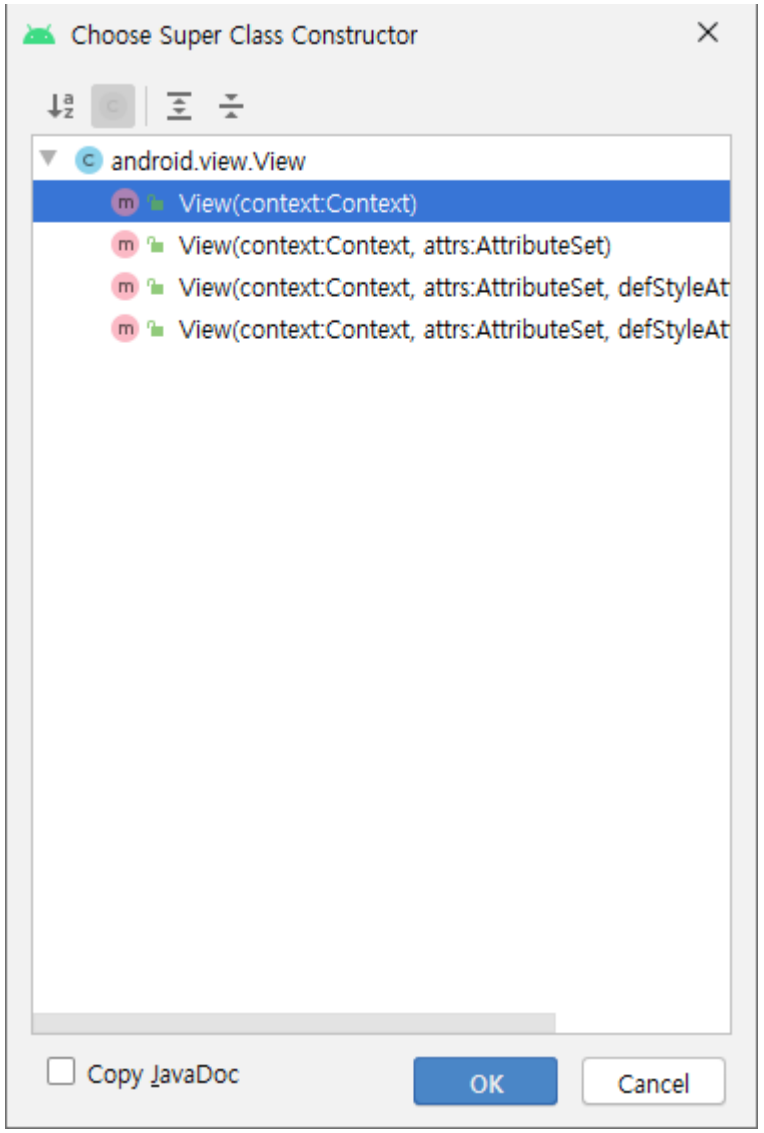

```
class CustomView extends View {
}

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        CustomView customView = new CustomView(this);
        setContentView(customView);
    }
}
```

MainActivity.java


```
11  class CustomView extends View {
12  }
13
14  public class MainActivity extends AppCompatActivity {
15
16      @Override
17      protected void onCreate(Bundle savedInstanceState) {
18          super.onCreate(savedInstanceState);
19          setContentView(R.layout.activity_main);
20          CustomView customView = new CustomView(this);
21          setContentView(customView);
22      }
23  }
```

단축키 : alt+Insert






```
import android.content.Context;
import android.view.View;

class CustomView extends View {
    public CustomView(Context context) {
        super(context);
    }
}
```

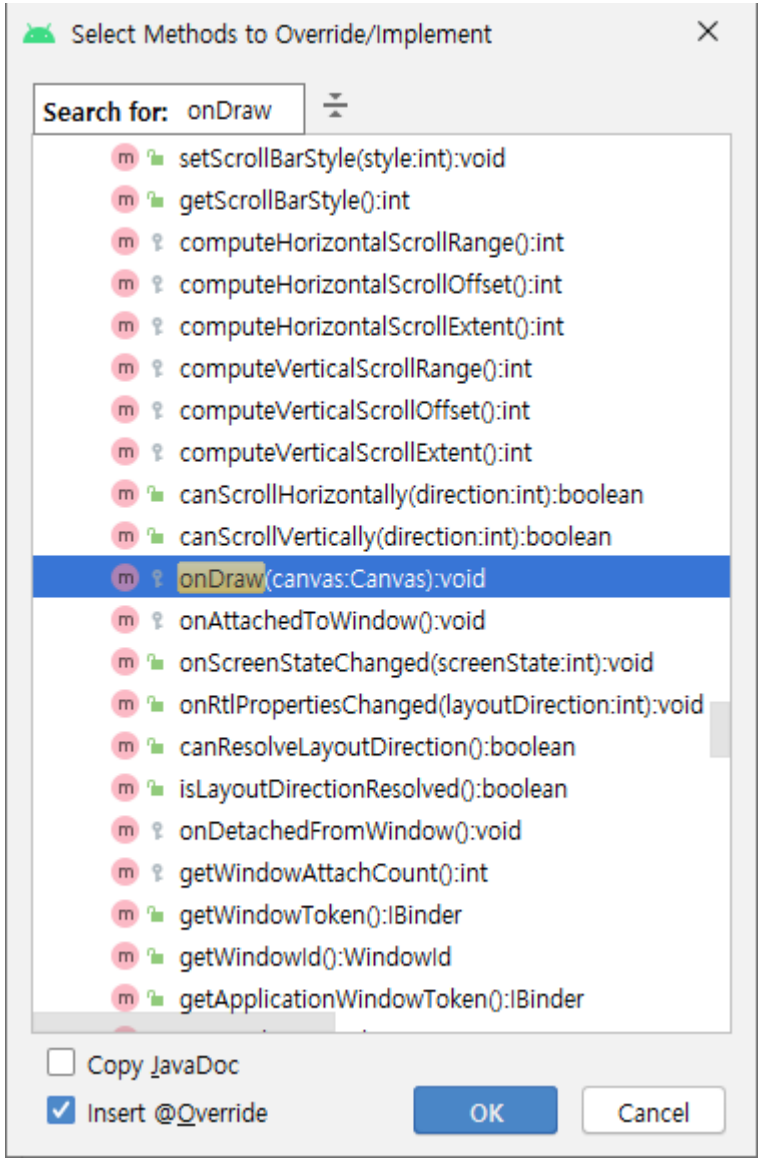
MainActivity.java


```
class CustomView extends View {
    private Paint paint;
    public CustomView(Context context) {
        super(context);
        setBackgroundColor(Color.rgb(225,225,225));
    }
}

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        CustomView customView = new CustomView(this);
        setContentView(customView);
    }
}
```

onDraw() – override

- 단축키 : ctrl + o

```
class CustomView extends View {
    public CustomView(Context context) {
        super(context);
        setBackgroundColor(Color.rgb(225,225,225));
    }

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
    }
}
```

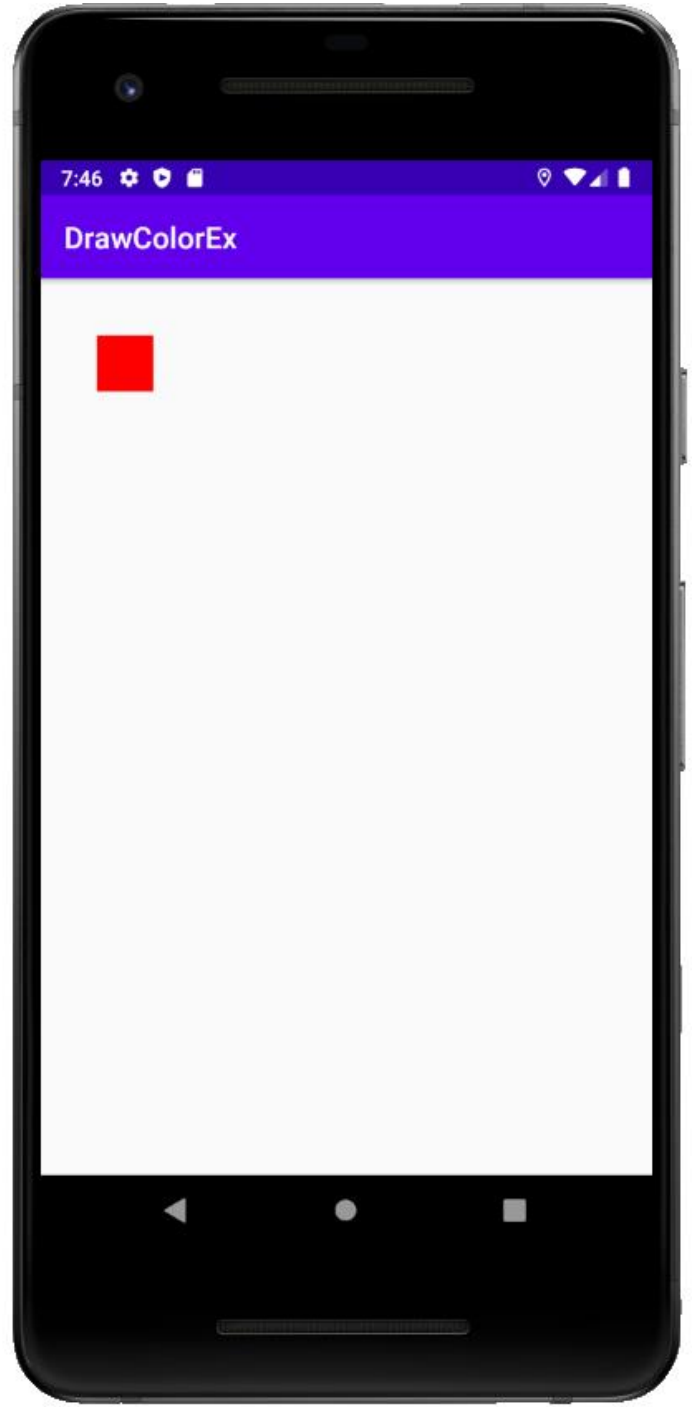


```
class CustomView extends View {
    private Paint paint;
    public CustomView(Context context) {
        super(context);
        paint = new Paint();
        setBackgroundColor(Color.rgb(225,225,225));
    }
    @Override
    protected void onDraw(Canvas canvas) {
        paint.setColor(Color.MAGENTA);
        paint.setStrokeWidth(20);
        canvas.drawLine(100, 100, 700, 100, paint);
        canvas.drawRect(100, 300, 700, 700, paint);
        canvas.drawCircle(300, 1200, 200, paint);
        paint.setTextSize(80);
        canvas.drawText("This a test.", 100, 900, paint);
    }
}

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        CustomView customView = new CustomView(this);
        setContentView(customView);
    }
}
```

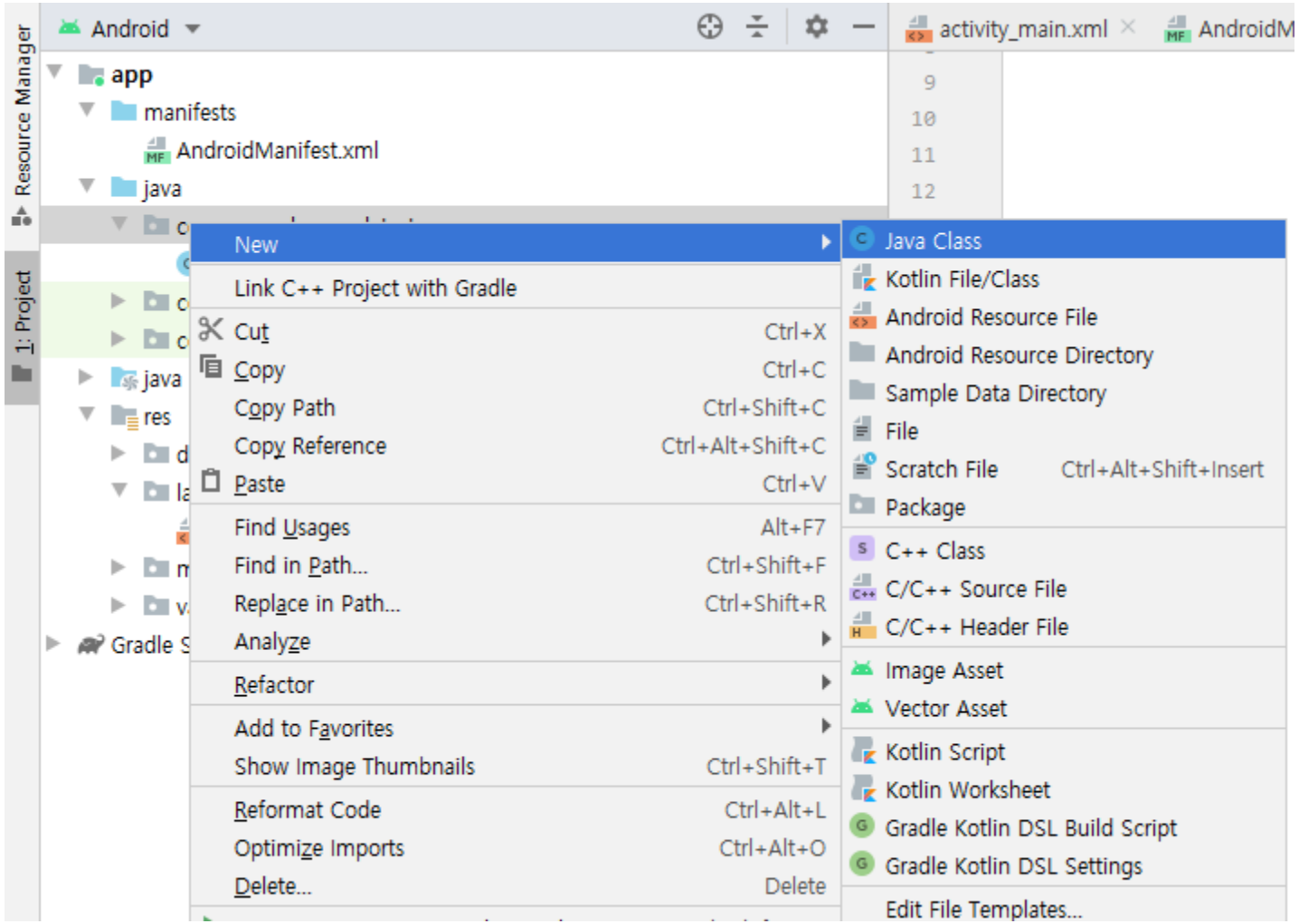
MainActivity.java


```
class CustomView extends View {
    private Paint paint;
    public CustomView(Context context) {
        super(context);
        paint = new Paint();
        setBackgroundColor(Color.rgb(225,225,225));
    }
    @Override
    protected void onDraw(Canvas canvas) {
        paint.setColor(Color.MAGENTA);
        paint.setStrokeWidth(20);
        canvas.drawLine(100, 100, 700, 100, paint);
        canvas.drawRect(100, 300, 700, 700, paint);
        canvas.drawCircle(300, 1200, 200, paint);
        paint.setTextSize(80);
        canvas.drawText("This a test.", 100, 900, paint);
    }
}

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        CustomView customView = new CustomView(this);
        setContentView(customView);
    }
}
```

빨간색 사각형 그리기
(CustomActivity)

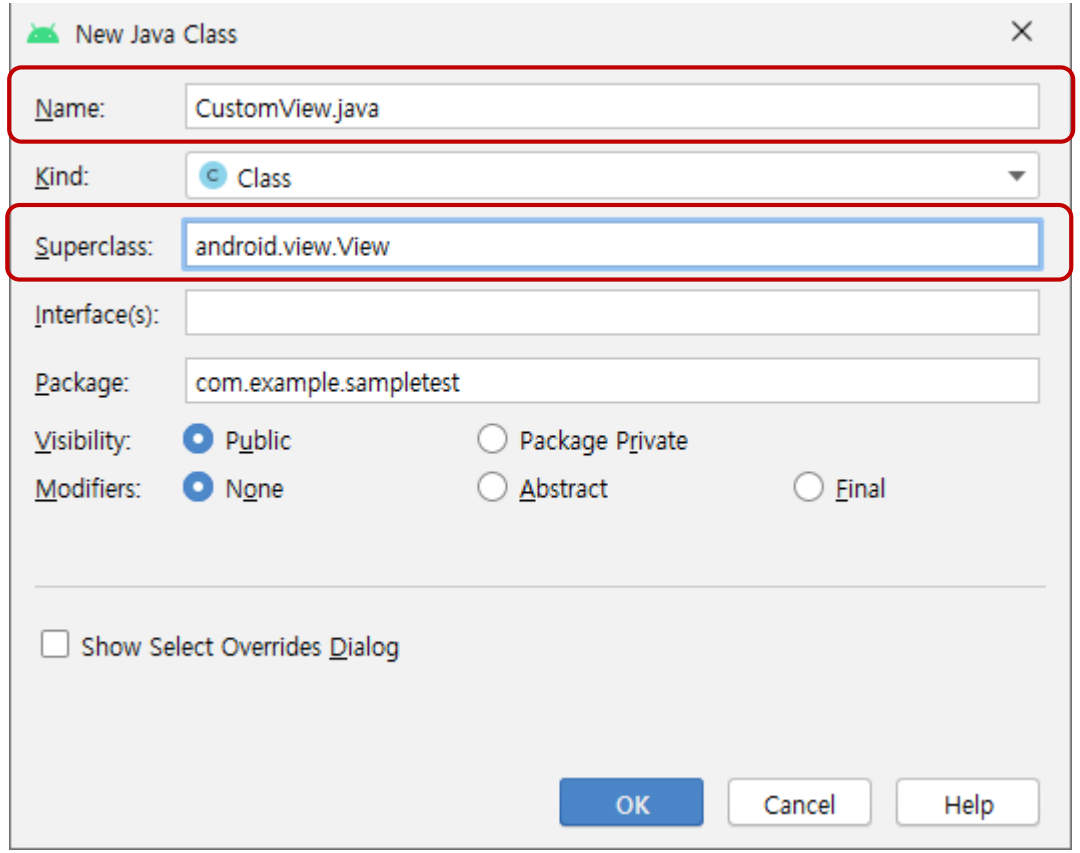

7:46
DrawColorEx

새로운 클래스 생성하기

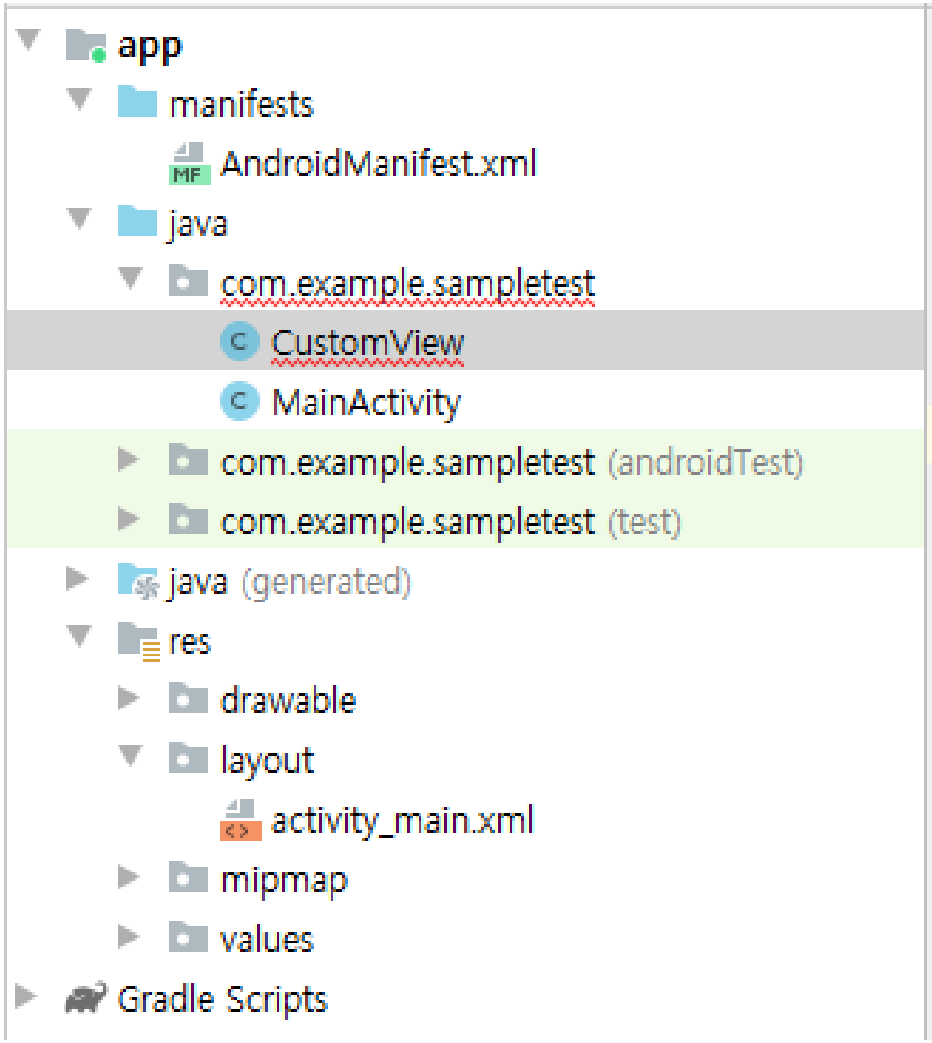

Android
app
manifests
AndroidManifest.xml
java
activity_main.xml
AndroidM
Resource Manager
1: Project
New
Link C++ Project with Gradle
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy Reference Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Show Image Thumbnails Ctrl+Shift+T
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Java Class
Kotlin File/Class
Android Resource File
Android Resource Directory
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Package
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset
Kotlin Script
Kotlin Worksheet
Gradle Kotlin DSL Build Script
Gradle Kotlin DSL Settings
Edit File Templates...

CustomView.java클래스 생성

- **View클래스 상속**



CustomView.java



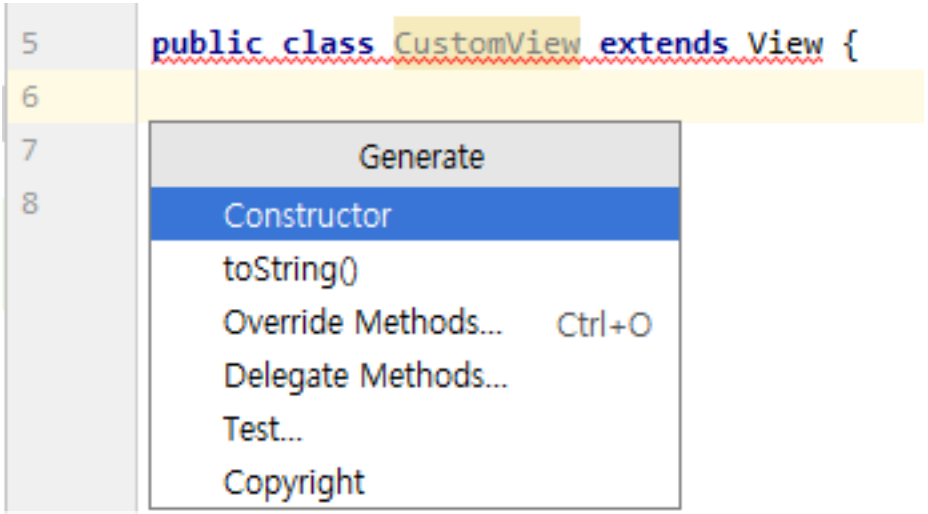

```
1  package com.example.sampletest;
2
3  import android.view.View;
4
5  public class CustomView extends View {
6
7  }
8
```

```
5  public class CustomView extends View {
6
7
8
```

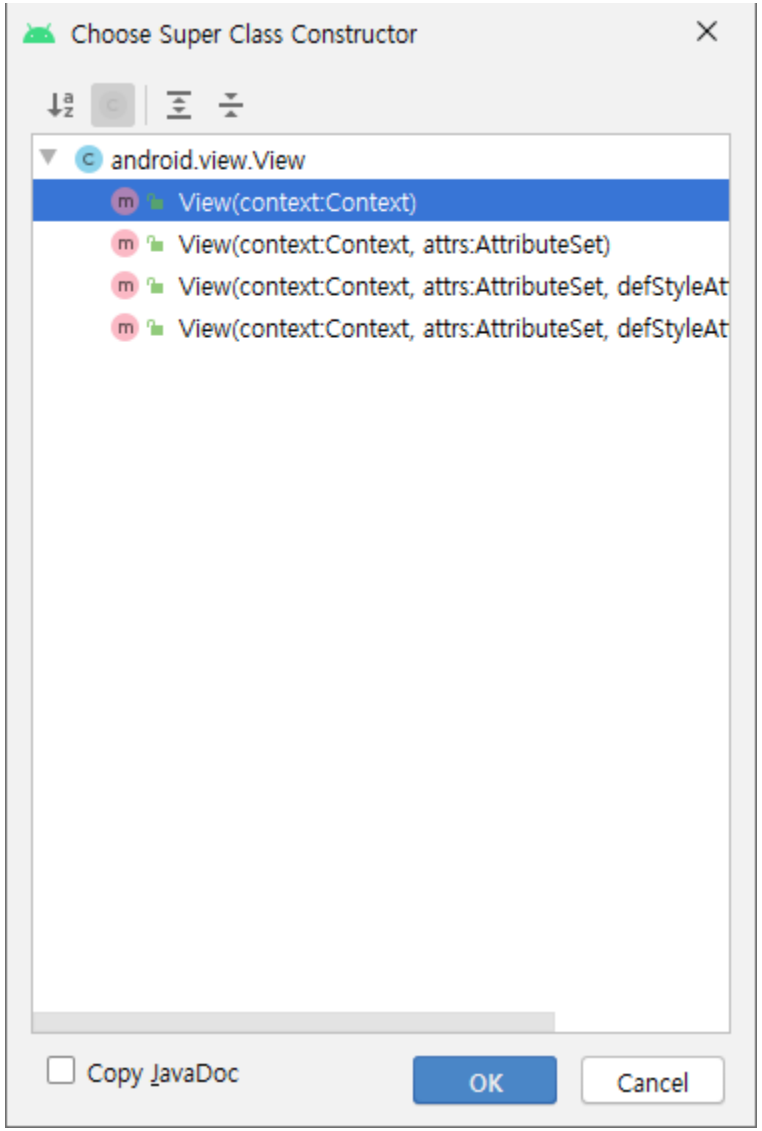
Generate
Constructor
toString()
Override Methods... Ctrl+O
Delegate Methods...
Test...
Copyright

단축키 : alt + Insert

CustomView.java



단축키 : alt+Insert




```
import android.content.Context;
import android.view.View;

public class CustomView extends View {
    public CustomView(Context context) {
        super(context);
    }
}
```

CustomView.java

페인트 객체 초기화하기

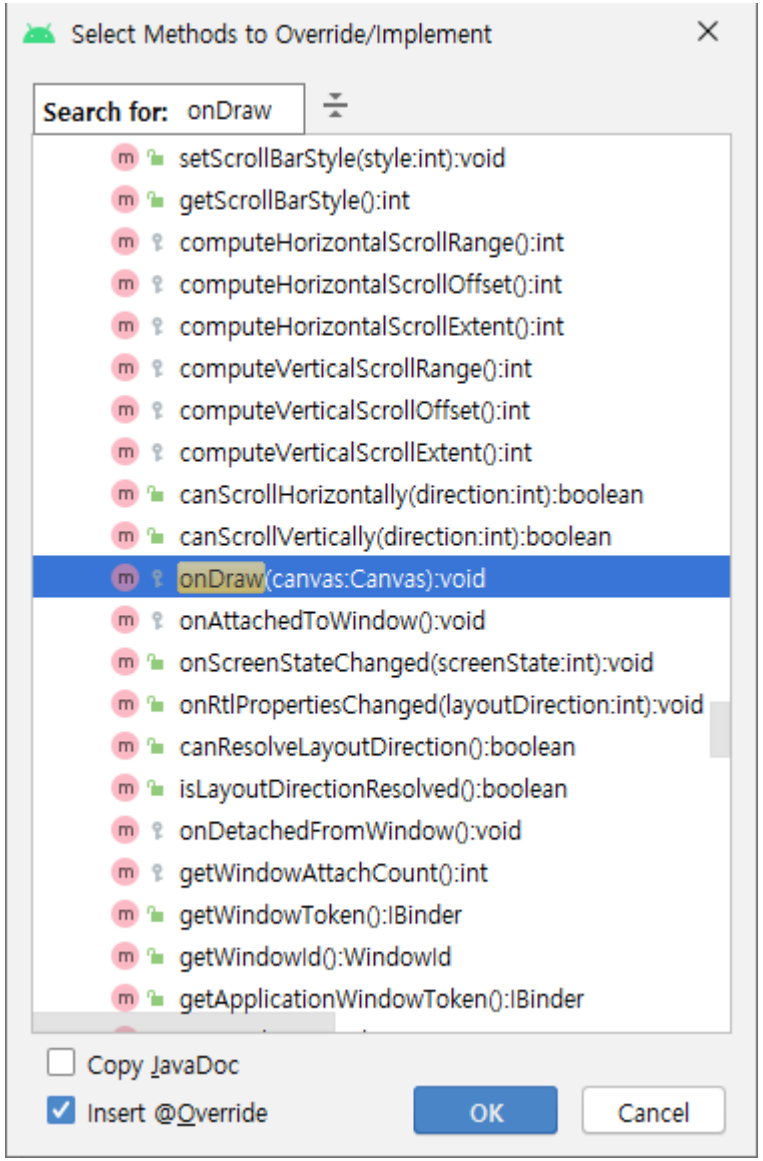
- paint객체는 그래픽 속성을 가지고 있음
- paint객체 생성후 색상 설정

```
public class CustomView extends View {
    Paint paint;
    public CustomView(Context context) {
        super(context);
        paint = new Paint();
        paint.setColor(Color.RED);
    }
}
```

CustomView.java

- **onDraw() – override**
  - 단축키 : ctrl + o

```
public class CustomView extends View {
    Paint paint;
    public CustomView(Context context) {
        super(context);
        paint = new Paint();
        paint.setColor(Color.RED);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
    }
}
```



CustomView.java


```
public class CustomView extends View {
    Paint paint;
    public CustomView(Context context) {
        super(context);
        paint = new Paint();
        paint.setColor(Color.RED);
    }
    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        canvas.drawRect(100, 100, 200, 200, paint);
    }
}
```

CustomView.java


```
public class CustomView extends View {
    Paint paint;
    public CustomView(Context context) {
        super(context);
        paint = new Paint();
        paint.setColor(Color.RED);
    }
    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        canvas.drawRect(100, 100, 200, 200, paint);
    }
}
```

MainActivity.java

- MainActivity와 연동

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        CustomView customView = new CustomView(this);
        setContentView(customView);
    }
}
```

MainActivity.java


```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        CustomView customView = new CustomView(this);
        setContentView(customView);
    }
}
```

Graphic Font

Fields	
public static final Typeface	DEFAULT The default NORMAL typeface object
public static final Typeface	DEFAULT_BOLD The default BOLD typeface object.
public static final Typeface	MONOSPACE The NORMAL style of the default monospace typeface.
public static final Typeface	SANS_SERIF The NORMAL style of the default sans serif typeface.
public static final Typeface	SERIF The NORMAL style of the default serif typeface.

https://developer.android.com/reference/android/graphics/Typeface?hl=en

MainActivity.java


```
class CustomView extends View {
    public CustomView(Context context) {
        super(context);
    }

    @Override
    protected void onDraw(Canvas canvas) {
    }
}

public class MainActivity extends AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        CustomView customView = new CustomView(this);
        setContentView(customView);
    }
}
```

MainActivity.java


```
class CustomView extends View {
    public CustomView(Context context) {
        super(context);
        setBackgroundColor(Color.YELLOW);
    }
```

MainActivity.java

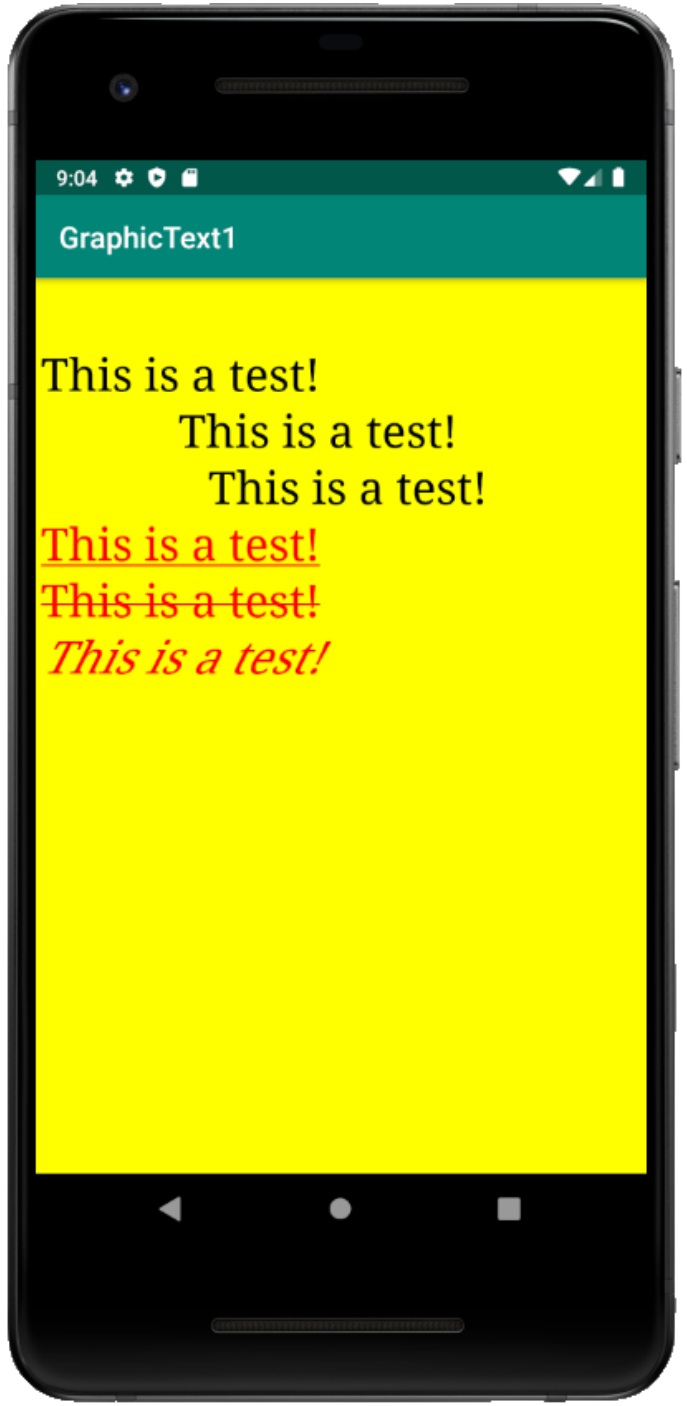

```
@Override
protected void onDraw(Canvas canvas) {
    Paint paint = new Paint();
    paint.setAntiAlias(true);
    paint.setTextSize(100);
    Typeface t;

    t = Typeface.create(Typeface.DEFAULT, Typeface.NORMAL);
    paint.setTypeface(t);
    canvas.drawText("DEFAULT 폰트", 10, 400, paint);

    t = Typeface.create(Typeface.DEFAULT_BOLD, Typeface.NORMAL);
    paint.setTypeface(t);
    canvas.drawText("DEFAULT_BOLD 폰트", 10, 600, paint);

    t = Typeface.create(Typeface.MONOSPACE, Typeface.NORMAL);
    paint.setTypeface(t);
    canvas.drawText("MONOSPACE 폰트", 10, 800, paint);

    t = Typeface.create(Typeface.SERIF, Typeface.NORMAL);
    paint.setTypeface(t);
    canvas.drawText("SERIF 폰트", 10, 1000, paint);

    t = Typeface.create(Typeface.SANS_SERIF, Typeface.NORMAL);
    paint.setTypeface(t);
    canvas.drawText("SANS_SERIF 폰트", 10, 1200, paint);
}
```

MainActivity.java


```
class CustomView extends View {
    public CustomView(Context context) {
        super(context);
        setBackgroundColor(Color.YELLOW);
    }
    @Override
    protected void onDraw(Canvas canvas) {
        Paint paint = new Paint();
        paint.setAntiAlias(true);
        paint.setTextSize(100);
        Typeface t;

        t = Typeface.create(Typeface.DEFAULT, Typeface.NORMAL);
        paint.setTypeface(t);
        canvas.drawText("DEFAULT 폰트", 10, 400, paint);

        t = Typeface.create(Typeface.DEFAULT_BOLD, Typeface.NORMAL);
        paint.setTypeface(t);
        canvas.drawText("DEFAULT_BOLD 폰트", 10, 600, paint);

        t = Typeface.create(Typeface.MONOSPACE, Typeface.NORMAL);
        paint.setTypeface(t);
        canvas.drawText("MONOSPACE 폰트", 10, 800, paint);

        t = Typeface.create(Typeface.SERIF, Typeface.NORMAL);
        paint.setTypeface(t);
        canvas.drawText("SERIF 폰트", 10, 1000, paint);

        t = Typeface.create(Typeface.SANS_SERIF, Typeface.NORMAL);
        paint.setTypeface(t);
        canvas.drawText("SANS_SERIF 폰트", 10, 1200, paint);
    }
}
```

MainActivity.java


```java
public class MainActivity extends AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        CustomView customView = new CustomView(this);
        setContentView(customView);
    }
}
```



setTextAlign

Added in API level 1

```
public void setTextAlign (Paint.Align align)
```

Set the paint's text alignment. This controls how the text is positioned relative to its origin. LEFT align means that all of the text will be drawn to the right of its origin (i.e. the origin specifieds the LEFT edge of the text) and so on.

Parameters	
`align`	**`Paint.Align`**: set the paint's Align value for drawing text.

https://developer.android.com/reference/android/graphics/Paint

MainActivity.java


```
class CustomView extends View {
    public CustomView(Context context) {
        super(context);
    }

    @Override
    protected void onDraw(Canvas canvas) {
    }
}

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        CustomView customView = new CustomView(this);
        setContentView(customView);
    }
}
```

MainActivity.java


```
public CustomView(Context context) {
    super(context);
    setBackgroundColor(Color.YELLOW);
}
```

MainActivity.java


```
    @Override
    protected void onDraw(Canvas canvas) {
        Paint paint = new Paint();
        paint.setAntiAlias(true);
        paint.setTextSize(80);
        Typeface t;
        t = Typeface.create(Typeface.SERIF, Typeface.NORMAL);
        paint.setTypeface(t);
        paint.setTextAlign(Paint.Align.LEFT);
        canvas.drawText("This is a test!", 10, 200, paint);

        paint.setTextAlign(Paint.Align.CENTER);
        canvas.drawText("This is a test!", 500, 300, paint);

        paint.setTextAlign(Paint.Align.RIGHT);
        canvas.drawText("This is a test!", 800, 400, paint);

        paint.setTextAlign(Paint.Align.LEFT);
        paint.setColor(Color.RED);

        paint.setUnderlineText(true);
        canvas.drawText("This is a test!", 10, 500, paint);
        paint.setUnderlineText(false);

        paint.setStrikeThruText(true);
        canvas.drawText("This is a test!", 10, 600, paint);
        paint.setStrikeThruText(false);

        paint.setTextSkewX(-0.5f);
        canvas.drawText("This is a test!", 10, 700, paint);
        paint.reset();
    }
}
```

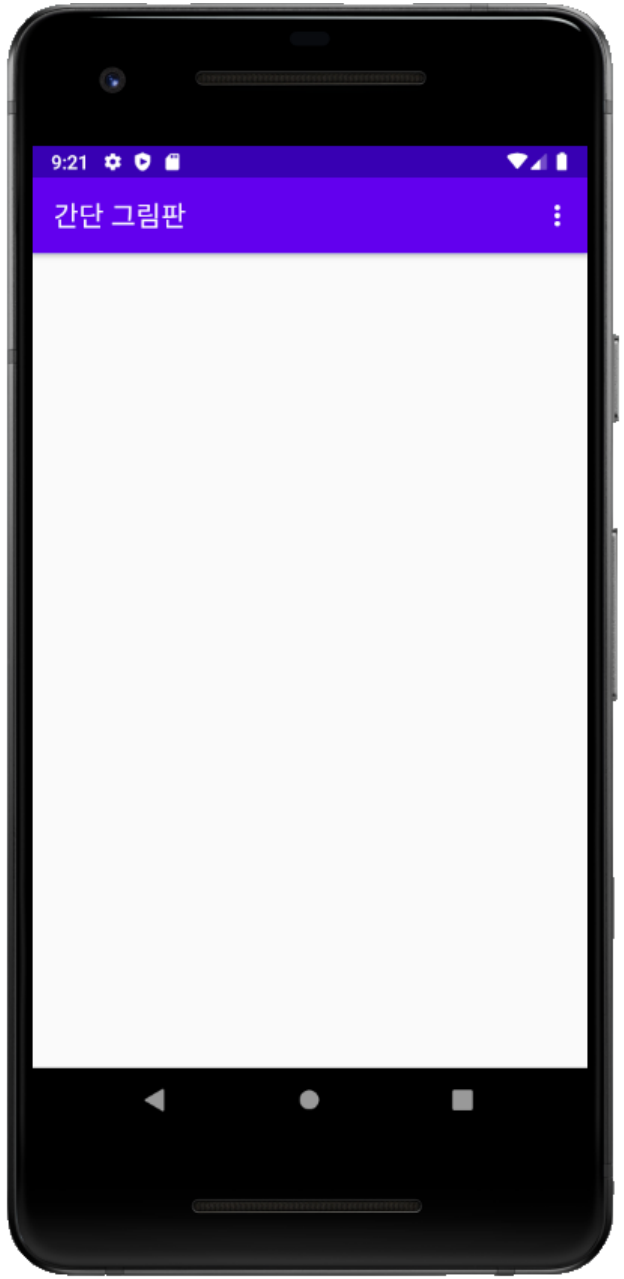
MainActivity.java


```
class CustomView extends View {
    public CustomView(Context context) {
        super(context);
        setBackgroundColor(Color.YELLOW);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        Paint paint = new Paint();
        paint.setAntiAlias(true);
        paint.setTextSize(80);

        Typeface t;
        t = Typeface.create(Typeface.SERIF, Typeface.NORMAL);
        paint.setTypeface(t);
        paint.setTextAlign(Paint.Align.LEFT);
        canvas.drawText("This is a test!", 10, 200, paint);

        paint.setTextAlign(Paint.Align.CENTER);
        canvas.drawText("This is a test!", 500, 300, paint);

        paint.setTextAlign(Paint.Align.RIGHT);
        canvas.drawText("This is a test!", 800, 400, paint);

        paint.setTextAlign(Paint.Align.LEFT);
        paint.setColor(Color.RED);
```

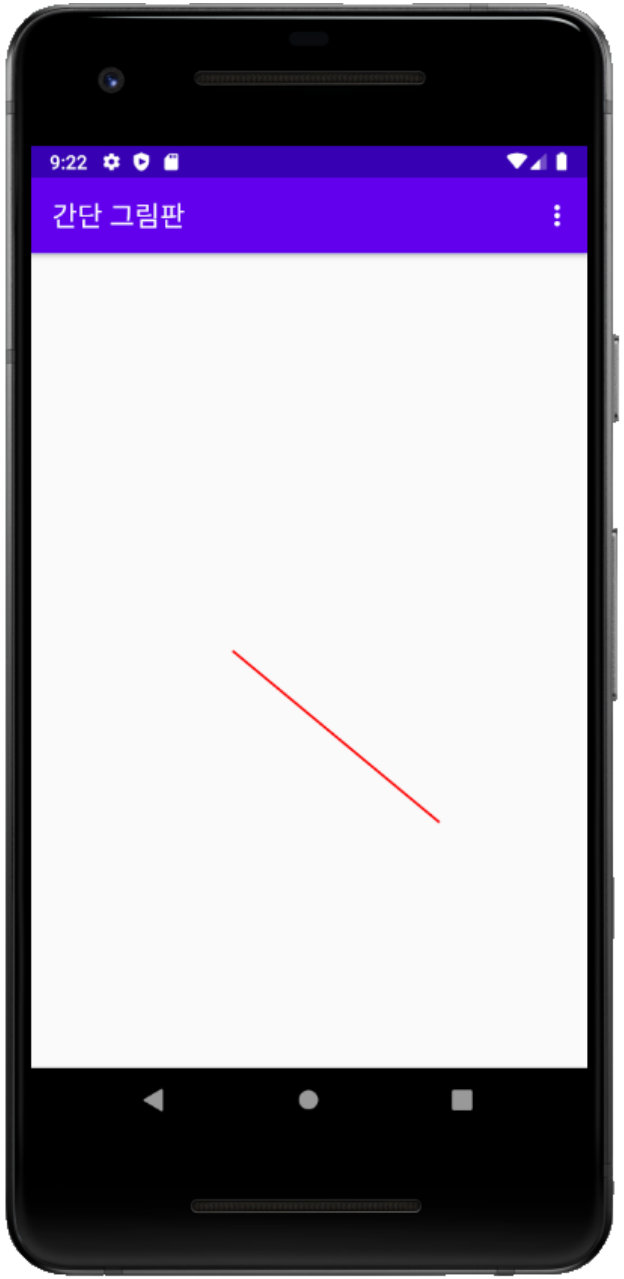
MainActivity.java

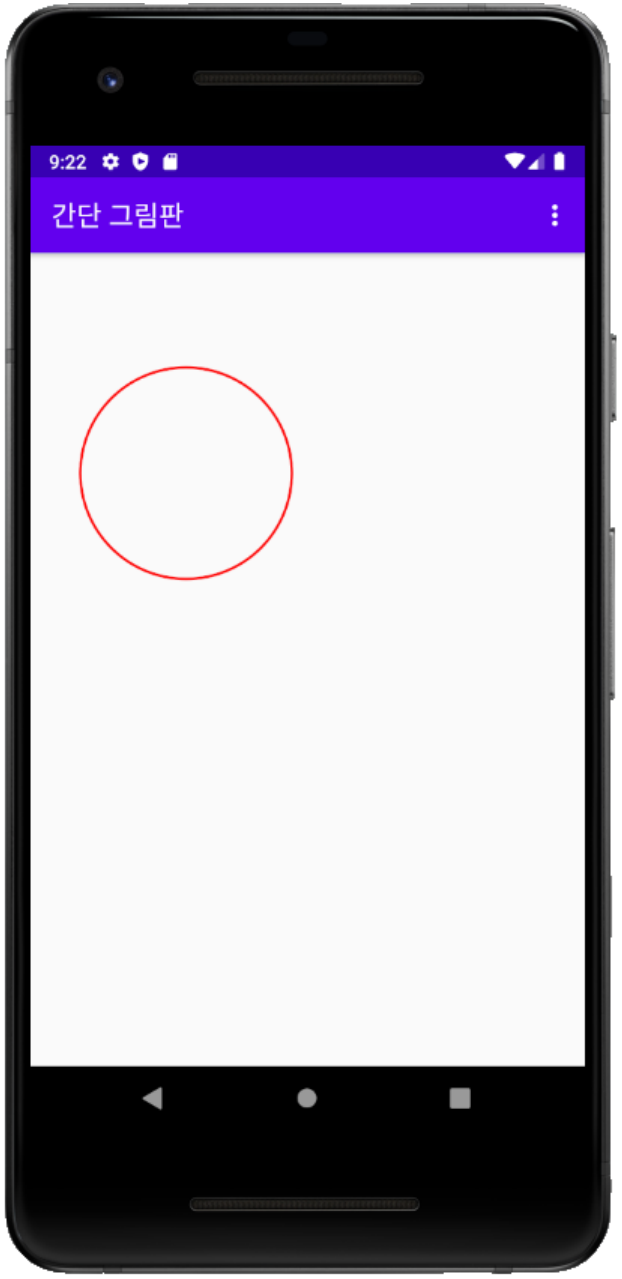

```
        paint.setUnderlineText(true);
        canvas.drawText("This is a test!", 10, 500, paint);
        paint.setUnderlineText(false);

        paint.setStrikeThruText(true);
        canvas.drawText("This is a test!", 10, 600, paint);
        paint.setStrikeThruText(false);

        paint.setTextSkewX(-0.5f);
        canvas.drawText("This is a test!", 10, 700, paint);
        paint.reset();
    }
}

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        CustomView customView = new CustomView(this);
        setContentView(customView);
    }
}
```

간단 그림판 만들기


9:21
간단 그림판


9:22
간단 그림판
선 그리기
원 그리기


9:22
간단 그림판


9:22
간단 그림판

MainActivity.java

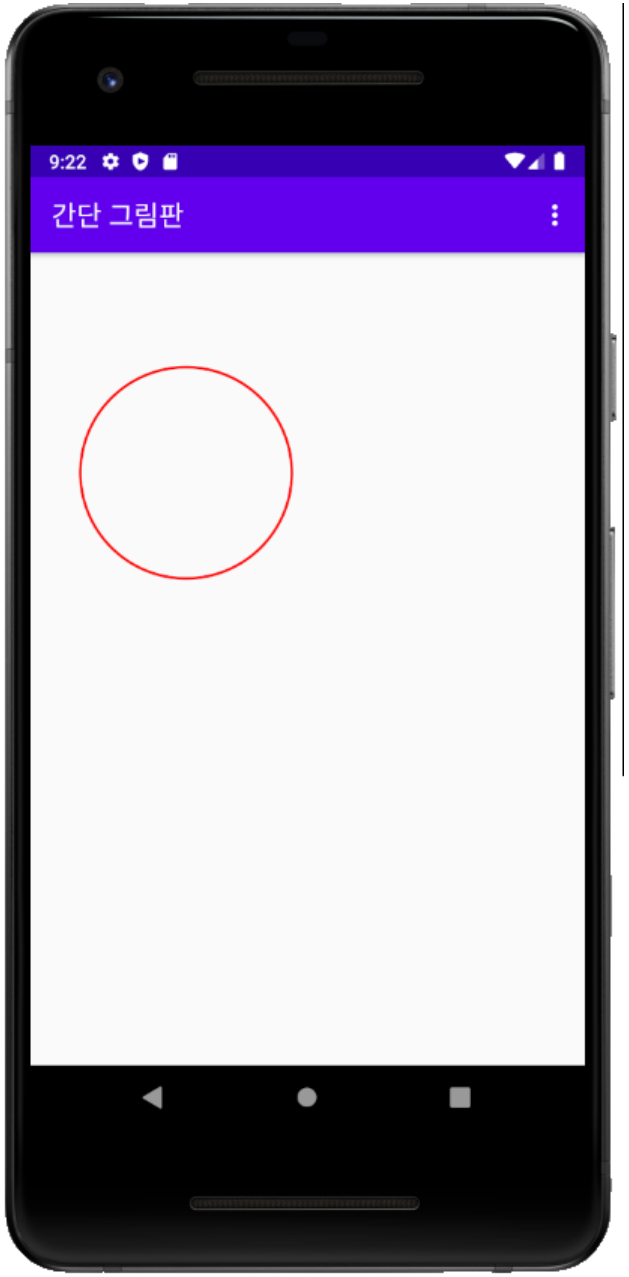

```
public class MainActivity extends AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new MyGraphicView (this));
    }


    private static class MyGraphicView extends View {
        public MyGraphicView(Context context) {
            super(context);
        }
        @Override
        protected void onDraw(Canvas canvas) {
            super.onDraw(canvas);
        }
    }
}
```

MainActivity.java


```
public class MainActivity extends AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new MyGraphicView (this));
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        return super.onOptionsItemSelected(item);
    }
}
```

MainActivity.java


```
public class MainActivity extends AppCompatActivity {
    private static class MyGraphicView extends View {
        public MyGraphicView(Context context) {
            super(context);
        }
        @Override
        public boolean onTouchEvent(MotionEvent event) {
            return super.onTouchEvent(event);
        }

        @Override
        protected void onDraw(Canvas canvas) {
            super.onDraw(canvas);
        }
    }
}
```

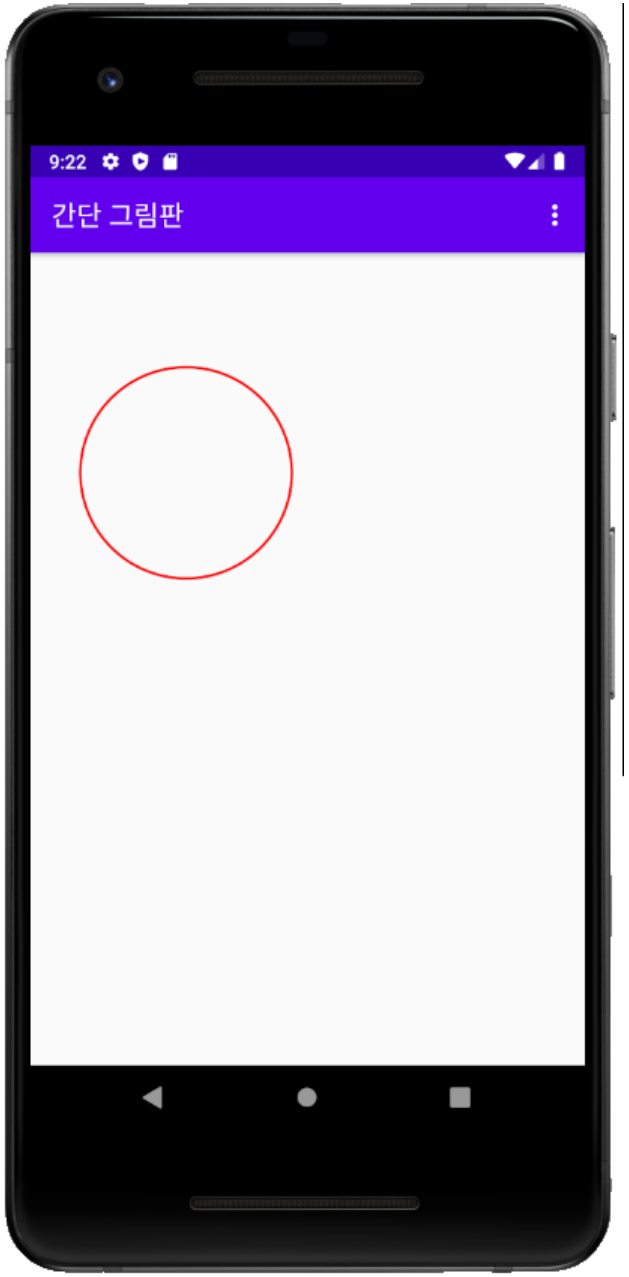


MainActivity.java


```java
public class MainActivity extends AppCompatActivity {
    final static int LINE = 1, CIRCLE = 2;
    static int curShape = LINE;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new MyGraphicView (this));
        setTitle("간단 그림판");
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);
        menu.add(0, 1, 0, "선 그리기");
        menu.add(0, 2, 0, "원 그리기");
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case 1:
                curShape = LINE; // 선
                return true;
            case 2:
                curShape = CIRCLE; // 원
                return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

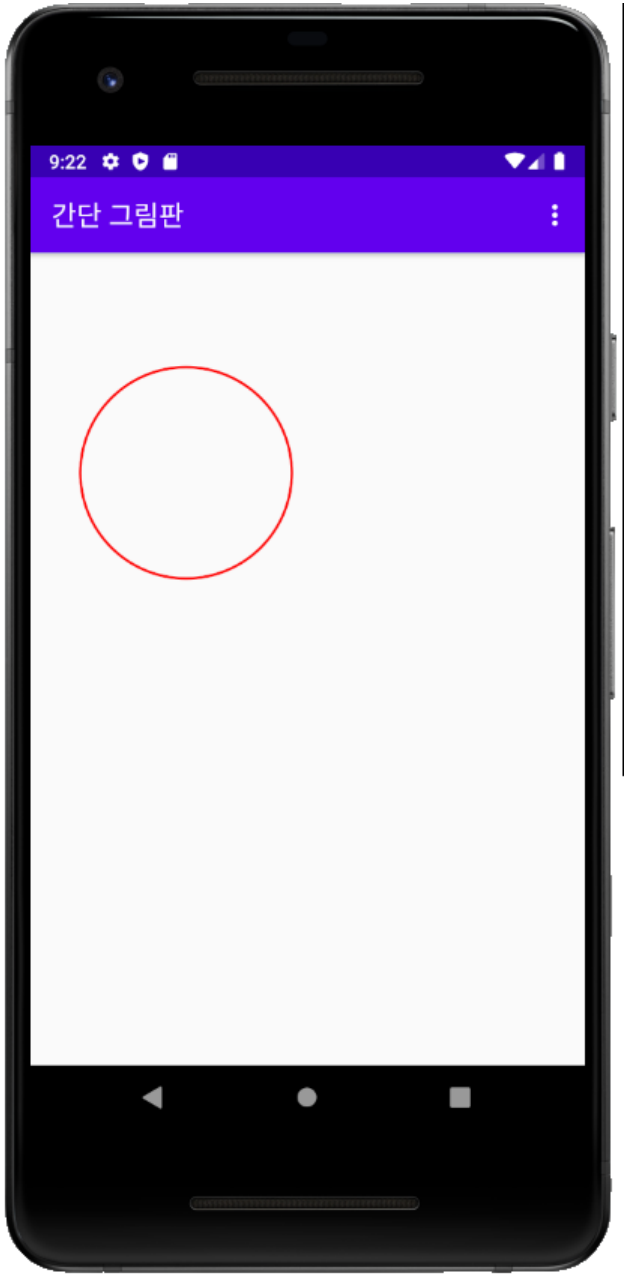
MainActivity.java


```
private static class MyGraphicView extends View {
    int startX = -1, startY = -1, stopX = -1, stopY = -1;
    public MyGraphicView(Context context) {
        super(context);
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                startX = (int) event.getX();
                startY = (int) event.getY();
                break;
            case MotionEvent.ACTION_MOVE:
            case MotionEvent.ACTION_UP:
                stopX = (int) event.getX();
                stopY = (int) event.getY();
                this.invalidate();
                break;
        }
        return true;
    }
}
```



MainActivity.java


```
private static class MyGraphicView extends View {
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        Paint paint = new Paint();
        paint.setAntiAlias(true);
        paint.setStrokeWidth(5);
        paint.setStyle(Paint.Style.STROKE);
        paint.setColor(Color.RED);

        switch (curShape) {
            case LINE:
                canvas.drawLine(startX, startY, stopX, stopY, paint);
                break;
            case CIRCLE:
                int radius = (int) Math.sqrt(Math.pow(stopX - startX, 2)
                        + Math.pow(stopY - startY, 2));
                canvas.drawCircle(startX, startY, radius, paint);
                break;
        }
    }
}
```

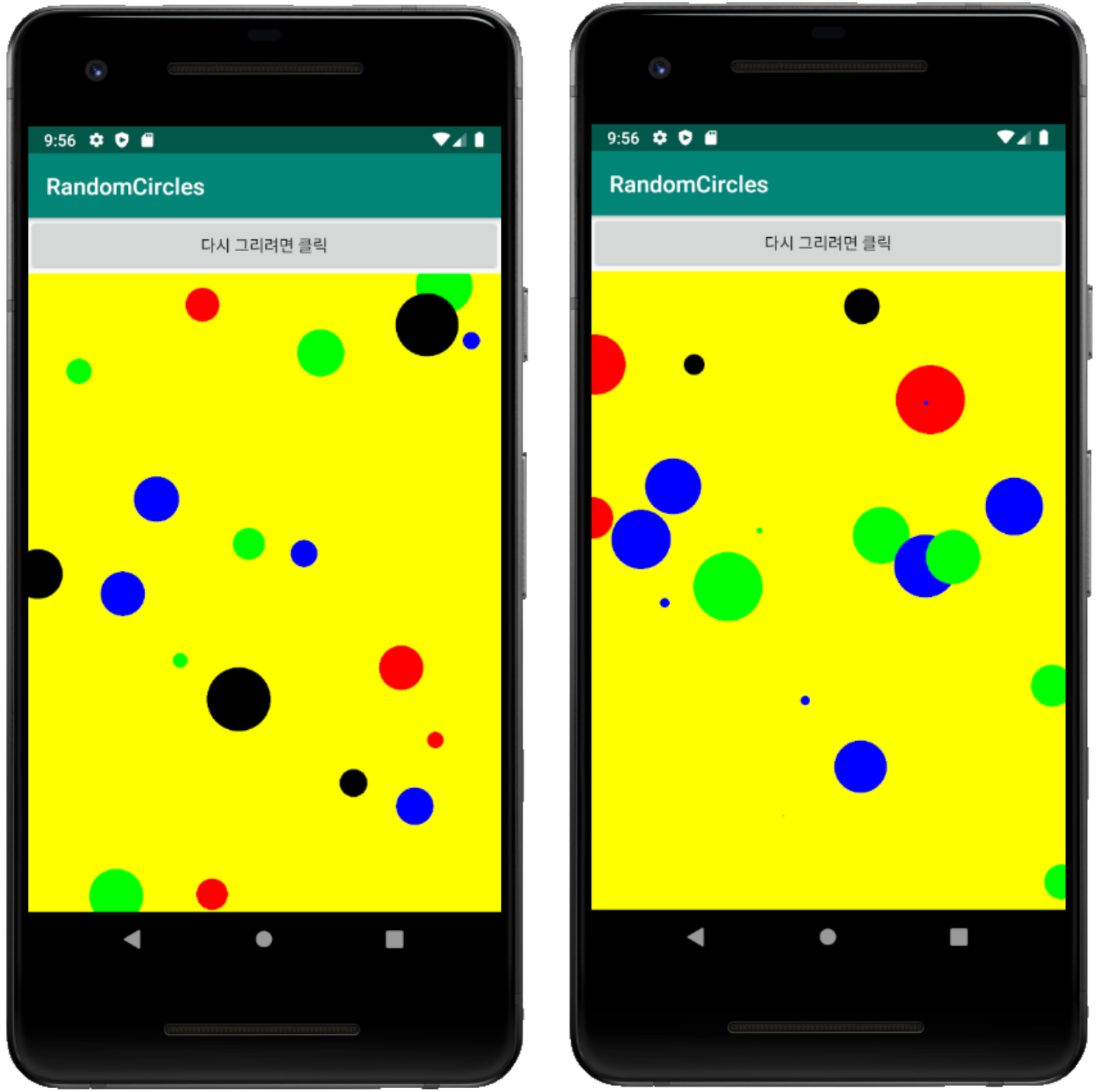


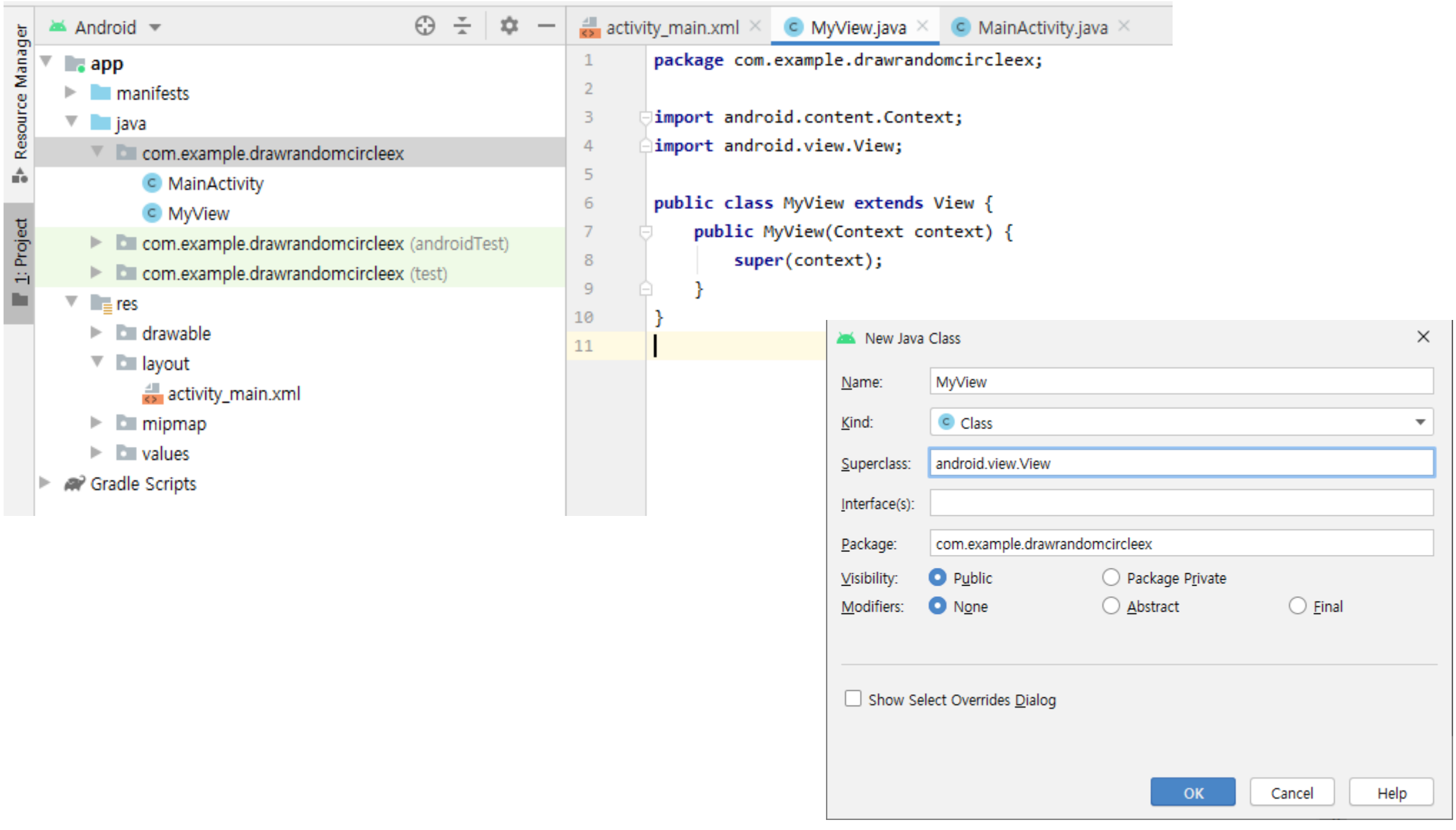
MainActivity.java –full code

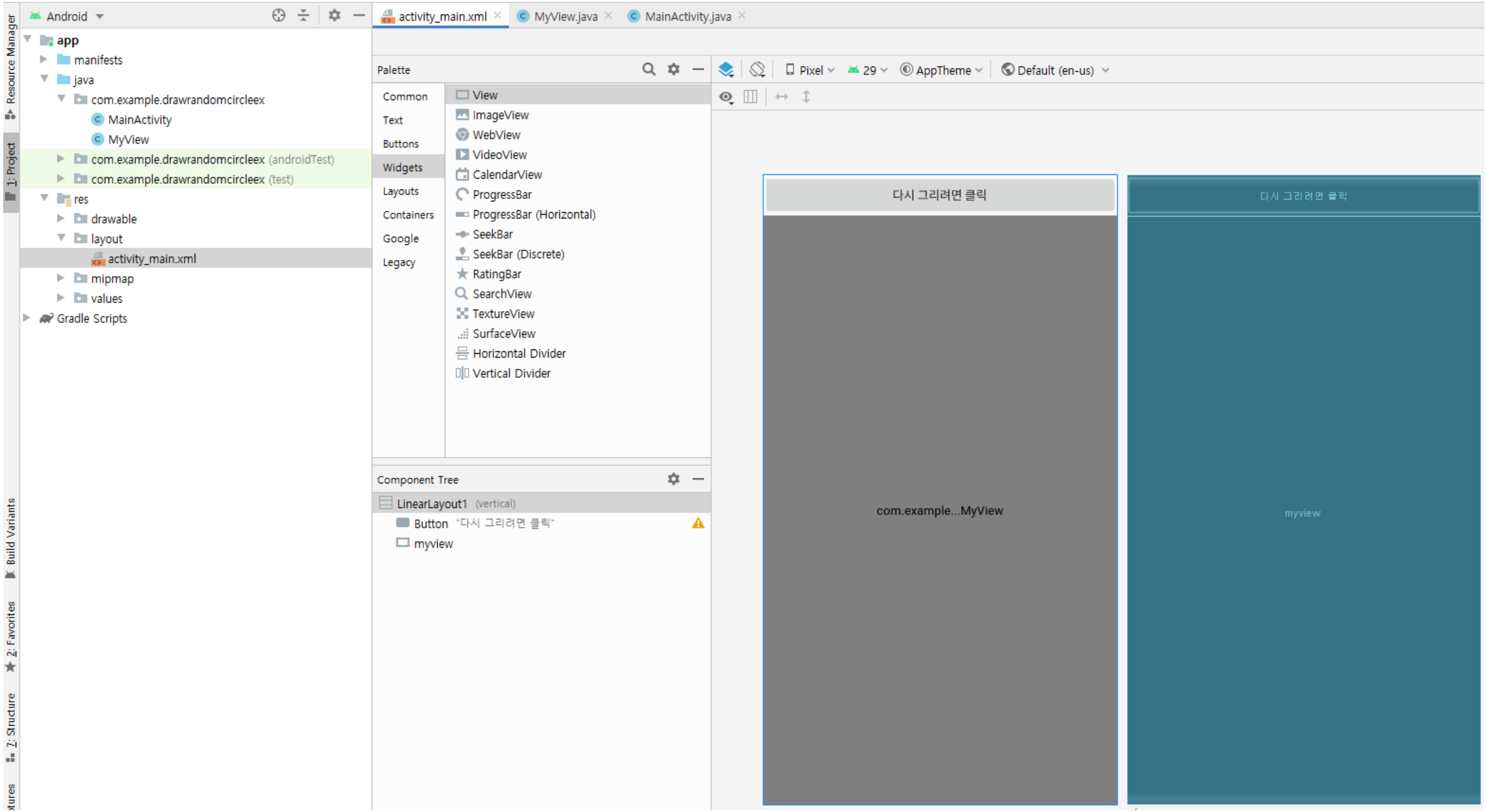

```
public class MainActivity extends AppCompatActivity {
    final static int LINE = 1, CIRCLE = 2;
    static int curShape = LINE;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new MyGraphicView (this));
        setTitle("간단 그림판");
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);
        menu.add(0, 1, 0, "선 그리기");
        menu.add(0, 2, 0, "원 그리기");
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case 1:
                curShape = LINE; // 선
                 return true;
            case 2:
                curShape = CIRCLE; // 원
                 return true;
        }
        return super.onOptionsItemSelected(item);
    }
```

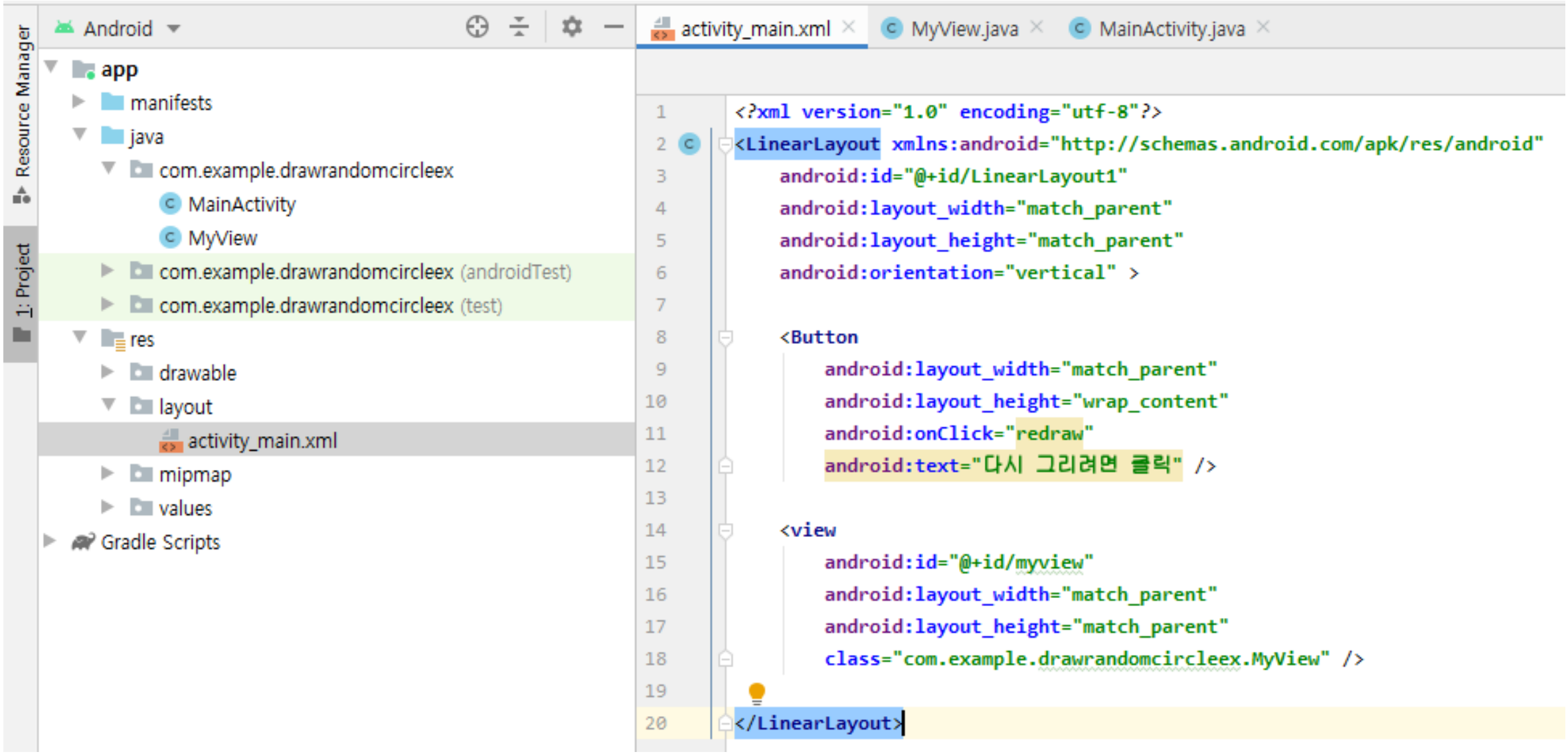
```
    private static class MyGraphicView extends View {
        int startX = -1, startY = -1, stopX = -1, stopY = -1;
        public MyGraphicView(Context context) {
            super(context);
        }
        @Override
        public boolean onTouchEvent(MotionEvent event) {
            switch (event.getAction()) {
                case MotionEvent.ACTION_DOWN:
                    startX = (int) event.getX();
                    startY = (int) event.getY();
                    break;
                case MotionEvent.ACTION_MOVE:
                case MotionEvent.ACTION_UP:
                    stopX = (int) event.getX();
                    stopY = (int) event.getY();
                    this.invalidate();
                    break;
            }
            return true;
        }
        protected void onDraw(Canvas canvas) {
            super.onDraw(canvas);
            Paint paint = new Paint();
            paint.setAntiAlias(true);
            paint.setStrokeWidth(5);
            paint.setStyle(Paint.Style.STROKE);
            paint.setColor(Color.RED);
            switch (curShape) {
                case LINE:
                    canvas.drawLine(startX, startY, stopX, stopY, paint);
                    break;
                case CIRCLE:
                    int radius = (int) Math.sqrt(Math.pow(stopX - startX, 2)
                            + Math.pow(stopY - startY, 2));
                    canvas.drawCircle(startX, startY, radius, paint);
                    break;
            }
        }
    }
}
```

랜덤 원 그리기


9:56
RandomCircles
다시 그리려면 클릭
9:56
RandomCircles
다시 그리려면 클릭

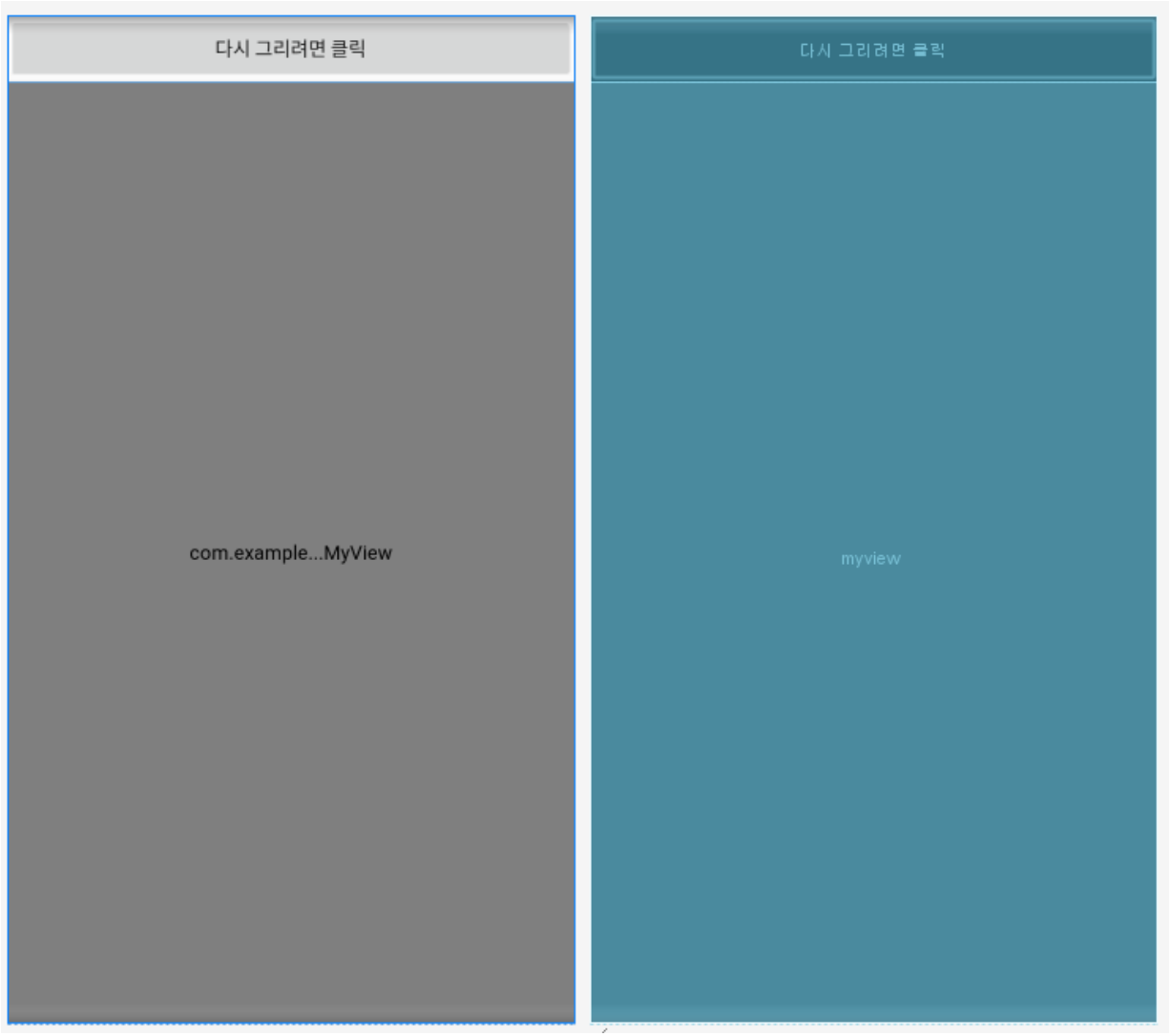

Android
app
manifests
java
com.example.drawrandomcircleex
MainActivity
MyView
com.example.drawrandomcircleex (androidTest)
com.example.drawrandomcircleex (test)
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
Resource Manager
1: Project
activity_main.xml
MyView.java
MainActivity.java
package com.example.drawrandomcircleex;
import android.content.Context;
import android.view.View;
public class MyView extends View {
public MyView(Context context) {
super(context);
}
}
New Java Class
Name: MyView
Kind: Class
Superclass: android.view.View
Interface(s):
Package: com.example.drawrandomcircleex
Visibility: Public Package Private
Modifiers: None Abstract Final
Show Select Overrides Dialog
OK
Cancel
Help


Android
app
manifests
java
com.example.drawrandomcircleex
MainActivity
MyView
com.example.drawrandomcircleex (androidTest)
com.example.drawrandomcircleex (test)
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
activity_main.xml
MyView.java
MainActivity.java
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
View
ImageView
WebView
VideoView
CalendarView
ProgressBar
ProgressBar (Horizontal)
SeekBar
SeekBar (Discrete)
RatingBar
SearchView
TextureView
SurfaceView
Horizontal Divider
Vertical Divider
Component Tree
LinearLayout1 (vertical)
Button "다시 그리려면 클릭"
myview
Pixel
29
AppTheme
Default (en-us)
다시 그리려면 클릭
com.example...MyView
다시 그리려면 클릭
myview
Resource Manager
1: Project
Build Variants
2: Favorites
7: Structure


Resource Manager
1: Project
Android
app
manifests
java
com.example.drawrandomcircleex
MainActivity
MyView
com.example.drawrandomcircleex (androidTest)
com.example.drawrandomcircleex (test)
res
drawable
layout
activity_main.xml
mipmap
values
Gradle Scripts
activity_main.xml
MyView.java
MainActivity.java
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3 android:id="@+id/LinearLayout1"
4 android:layout_width="match_parent"
5 android:layout_height="match_parent"
6 android:orientation="vertical" >
7
8 <Button
9 android:layout_width="match_parent"
10 android:layout_height="wrap_content"
11 android:onClick="redraw"
12 android:text="다시 그리려면 클릭" />
13
14 <view
15 android:id="@+id/myview"
16 android:layout_width="match_parent"
17 android:layout_height="match_parent"
18 class="com.example.drawrandomcircleex.MyView" />
19
20 </LinearLayout>

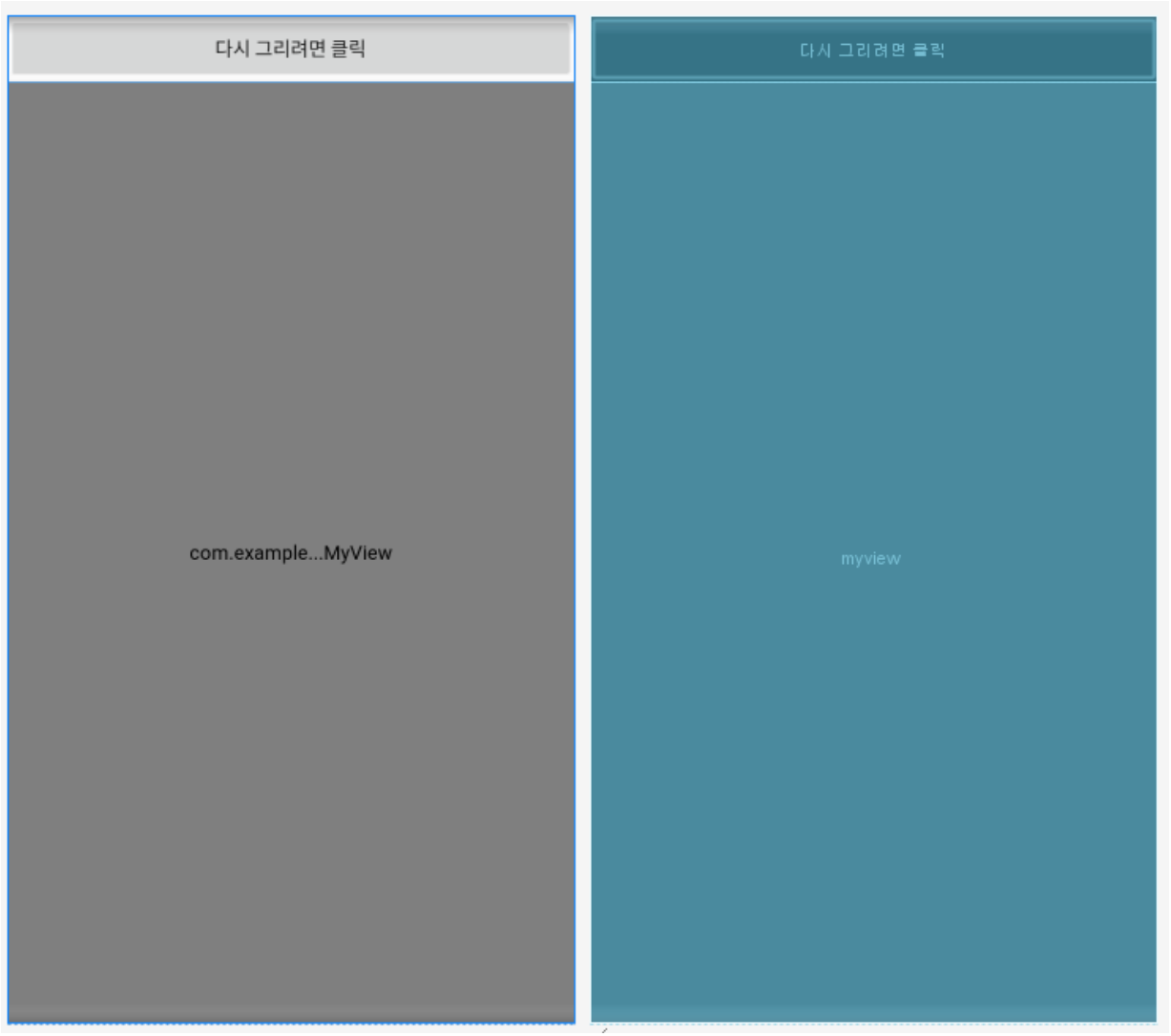
activity_main.xml


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="redraw"
        android:text="다시 그리려면 클릭" />

    <view
        android:id="@+id/myview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        class="com.example.drawrandomcircleex.MyView" />

</LinearLayout>
```



activity_main.xml


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="redraw"
        android:text="다시 그리려면 클릭" />

    <view
        android:id="@+id/myview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        class="com.example.drawrandomcircleex.MyView" />

</LinearLayout>
```



MainActivity.java


```
public class MainActivity extends AppCompatActivity {
    MyView mMyView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mMyView = (MyView) findViewById(R.id.myview);
    }

    public void redraw(View v) {
        mMyView.invalidate();
    }
}
```

MainActivity.java


```
public class MainActivity extends AppCompatActivity {
    MyView mMyView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mMyView = (MyView) findViewById(R.id.myview);
    }

    public void redraw(View v) {
        mMyView.invalidate();
    }
}
```

MyView.java


```
public class MyView extends View {

   public MyView(Context context) {
      super(context);
   }

   public MyView(Context context, AttributeSet attrs) {
      super(context, attrs);
   }

   @Override
   protected void onDraw(Canvas canvas) {
   }

}
```

MyView.java


```
public class MyView extends View {
   private Paint[] mForegrounds = {
         makePaint(Color.BLACK), makePaint(Color.BLUE), makePaint(Color.GREEN), makePaint(Color.RED) };
   private static Random r = new Random();

   public MyView(Context context) {
      super(context);
   }

   public MyView(Context context, AttributeSet attrs) {
      super(context, attrs);
   }

   private Paint makePaint(int color) {
      Paint p = new Paint();
      p.setColor(color);
      return (p);
   }
}
```

MyView.java


```
public class MyView extends View {

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        canvas.drawColor(Color.YELLOW);
        int width = getWidth();
        int height = getHeight();
        for (int i = 0; i < 20; i++) {
            float x = r.nextInt(width);
            float y = r.nextInt(height);
            float radius = r.nextInt(80);
            Paint circleColor = mForegrounds[r.nextInt(mForegrounds.length)];
            canvas.drawCircle(x, y, radius, circleColor);
        }
    }
}
```

MyView.java


```
public class MyView extends View {
   private Paint[] mForegrounds = { makePaint(Color.BLACK), makePaint(Color.BLUE), makePaint(Color.GREEN), makePaint(Color.RED) };
   private static Random r = new Random();

   public MyView(Context context) {
      super(context);
   }

   public MyView(Context context, AttributeSet attrs) {
      super(context, attrs);
   }

   @Override
   protected void onDraw(Canvas canvas) {
      super.onDraw(canvas);
      canvas.drawColor(Color.YELLOW);
      int width = getWidth();
      int height = getHeight();
      for (int i = 0; i < 20; i++) {
         float x = r.nextInt(width);
         float y = r.nextInt(height);
         float radius = r.nextInt(80);
         Paint circleColor = mForegrounds[r.nextInt(mForegrounds.length)];
         canvas.drawCircle(x, y, radius, circleColor);
      }
   }

   private Paint makePaint(int color) {
      Paint p = new Paint();
      p.setColor(color);
      return (p);
   }
}
```