

# Ch09. 그래픽과 이미지01

모바일게임프로그래밍  
김지심 교수

# 학습목표

캔버스에 도형을 그릴 수 있다.  
이미지 파일을 처리할 수 있다.

# 목차

---

01 그래픽

02 이미지



01



그래픽



# 1. 그래픽 | 캔버스와 페인트 기본

## 2차원 그래픽 출력방식

- 1) 뷰를 레이아웃에 배치하고 뷰에서 제공하는 기능으로 그래픽을 출력하거나 애니메이션 효과를 내는 것(그리는 작업을 안드로이드 시스템이 담당).  
정적 구현에 적합
- 2) 안드로이드가 제공하는 2D API(Canvas 등)를 직접 출력하는 방식으로  
게임 등 화면이 수시로 변하는 동적 구현에 적합

# 1. 그래픽 | 캔버스와 페인트 기본

## 캔버스와 페인트

Canvas는 도화지, Paint는 붓의 개념

Paint 클래스로 색상 등의 그래픽 옵션을 지정한 후,  
Canvas 클래스의 메소드를 이용해 도형을 그리는 구조



그림 9-1 Canvas와 Paint 클래스



# 1. 그래픽 | 캔버스와 페인트 기본

## Canvas

CLASS	METHOD/DEFINITION
android.Graphics.Canvas	그래픽을 그리기 위한 클래스로 다양한 도형, 이미지, 텍스트 등의 그리기 API 제공 onDraw에 Canvas 객체를 넘겨주어 뷰를 화면에 그림
onDraw(Canvas canvas)	클래스가 생성되거나 화면이 무효화(invalidate)될 때 호출되어 이 메소드에서 그린 내용이 뷰 영역에 출력됨
void invalidate()	화면을 무효화하여(화면 갱신) onDraw() 호출 (동적 그리기에 반드시 필요)
void drawPoint(float x, float y, Paint paint)	x, y좌표에 지정된 paint를 적용하여 점을 찍음
void drawLine(float startX, float startY, float stopX, float stopY, Paint paint)	시작, 끝점의 x,y좌표로 선 그림
new Rect(int left, int top, int right, int bottom)	왼쪽 상단, 오른쪽 하단 좌표로 사각형 생성
void drawRect(Rect r, Paint paint)	사각형 정보를 Rect에 담아 그림
void drawRoundRect(RectF r, float rx, float ry, Paint paint)	축반지름인 rx, ry으로 부드러운 사각형을 그림
void drawCircle(float cx, float cy, float radius, Paint paint)	원점의 x, y 좌표와 반지름으로 원을 그림
Rect, RectF	int형, float형

# 1. 그래픽 | 캔버스와 페인트 기본

## Paint

CLASS	METHOD/DEFINITION
android.Graphics.Paint	그리기 옵션, 도형, 텍스트, bitmap의 속성을 설정
	void setColor(int color) 한번 지정하면 변경할 때까지 유지됨
	void setAntiAlias(Boolean aa) 도형의 끝을 부드럽게 처리
	void setStyle(Paint.Style style) 도형 내부를 채울지 여부를 결정, 한번 지정하면 변경할 때까지 유지됨
	void setStrokeWidth(float width) 테두리 두께(0: 1px)
	void setStrokeCap(Paint.Cap cap) 선의 끝부분에 대한 효과 지정
	void setTextSize(float textSize)

<https://developer.android.com/reference/android/graphics/Paint.html>



# 1. 그래픽 | 캔버스와 페인트 기본

## Path

CLASS	METHOD/DEFINITION
Path	연결된 여러 점을 갖는 클래스
	void moveTo(float x, float y)      해당 점으로 이동
	void lineTo(float x, float y)      점을 추가
	void drawPath(Path path, Paint paint)      지정된 path로 그림

# 1. 그래픽 | 캔버스와 페인트 기본

## 캔버스와 페인트

### 형식 예시

그래픽을 표현할 때는 주로 View 클래스를 상속받아 View.onDraw(Canvas)를 재정의

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(new 재정의한 클래스 이름(this));
}

private static class 재정의한 클래스 이름 extends View {
    public 재정의한 클래스 이름(Context context) {
        super(context);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        // 여기에 화면에 그려질 내용을 코딩
    }
}
```

# 1. 그래픽 | 그래픽 처리 기본

예제9-1: Ex09\_01

onDraw() 메소드를 오버라이딩하여 그리기

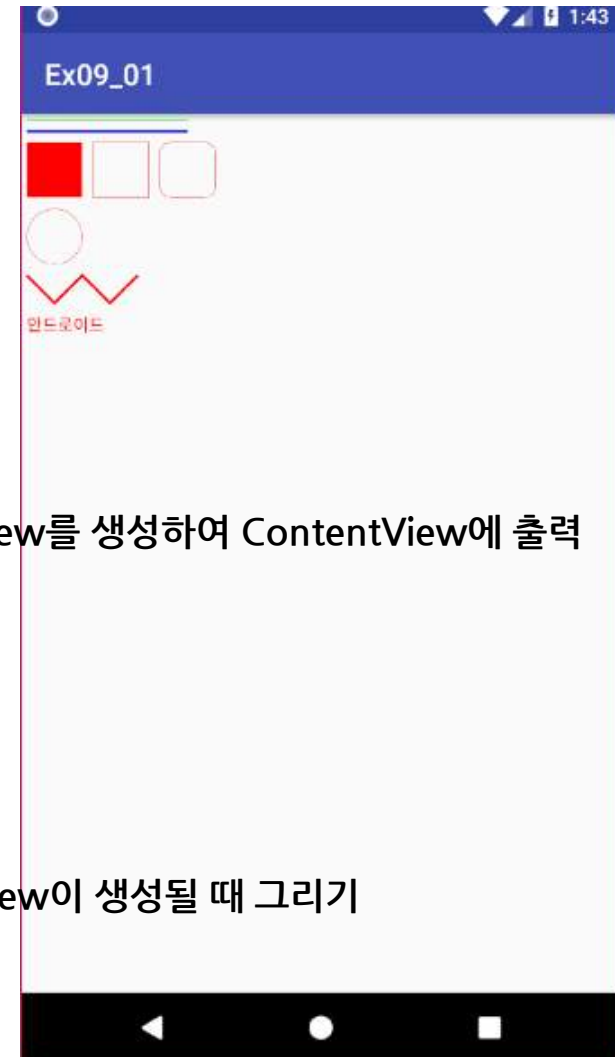
activity\_main.xml 삭제

```
+ MainActivity extends AppCompatActivity
- fields
- constructors
+ methods
# onCreate (savedInstanceState: Bundle?, bundle: Bundle?): void
```

→ MyGraphicView를 생성하여 ContentView에 출력

```
+ MainActivity.MyGraphicView extends View
- fields
+ constructors
+ MyGraphicView (context: Context)
+ methods
# onDraw (canvas: Canvas): void
```

→ MyGraphicView이 생성될 때 그리기



# 1. 그래픽 | 그래픽 처리 기본

## 예제9-1: Ex09\_01 MainActivity.java

```
14  public class MainActivity extends AppCompatActivity {  
15  
16      @Override  
17      protected void onCreate(Bundle savedInstanceState) {  
18          super.onCreate(savedInstanceState);  
19          setContentView(new MyGraphicView(this));  
20      }
```

```

20 }
21
22 public class MyGraphicView extends View {
23     public MyGraphicView(Context context) { super(context); }
24
25
26
27     @Override
28     protected void onDraw(Canvas canvas) {
29         super.onDraw(canvas);
30
31         Paint paint = new Paint();
32         paint.setAntiAlias(true);
33         paint.setColor(Color.GREEN);
34         canvas.drawLine(10, 10, 300, 10, paint);
35
36         paint.setColor(Color.BLUE);
37         paint.setStrokeWidth(5);
38         canvas.drawLine(10, 30, 300, 30, paint);
39
40         paint.setColor(Color.RED);
41         paint.setStrokeWidth(0);
42
43         paint.setStyle(Paint.Style.FILL);
44         Rect rect1 = new Rect(10, 50, 10 + 100, 50 + 100);
45         canvas.drawRect(rect1, paint);
46
47         paint.setStyle(Paint.Style.STROKE);
48         Rect rect2 = new Rect(130, 50, 130 + 100, 50 + 100);
49         canvas.drawRect(rect2, paint);
50
51         RectF rect3 = new RectF(250, 50, 250 + 100, 50 + 100);
52         canvas.drawRoundRect(rect3, 20, 20, paint);
53
54         canvas.drawCircle(60, 220, 50, paint);
55
56         paint.setStrokeWidth(5);
57
58         Path path1 = new Path();
59         path1.moveTo(10, 290);
60         path1.lineTo(10 + 50, 290 + 50);
61         path1.lineTo(10 + 100, 290);
62         path1.lineTo(10 + 150, 290 + 50);
63         path1.lineTo(10 + 200, 290);
64         canvas.drawPath(path1, paint);
65
66         paint.setStrokeWidth(0);
67         paint.setTextSize(30);
68         canvas.drawText("안드로이드", 10, 390, paint);
69     }
70 }
71
72

```

Paint(붓)을 설정하고 Canvas(도화지)에 그림

직접 JAVA 코드로 그리기 시에는 생성자 하나만 정의  
\* xml에 정의하여 사용 시엔 모든 생성자를 정의해주어야 함

## 예제9-1: Ex09\_01 MainActivity.java

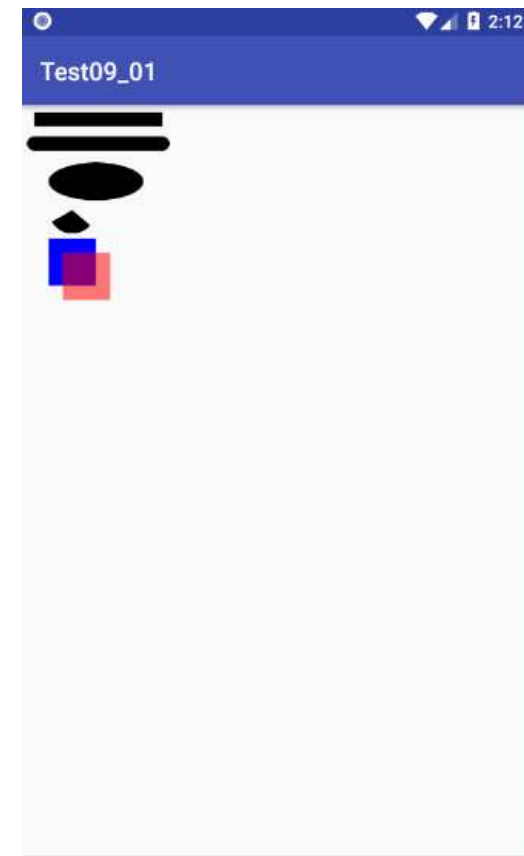
MainActivity 내부에 View를 상속받아  
MyGraphicView 클래스 정의

# 1. 그래픽 | 그래픽 처리 기본

## 직접 풀어보기: Test09\_01

그림과 같은 화면을 출력하도록 다음 메소드를 사용해 Java를 코딩해보자.

- `Paint.setStrokeCap()`
- `Canvas.drawOval()`
- `Paint.setColor(Color.argb())`





# 1. 그래픽 | 터치 이벤트

## 터치 이벤트

화면에 생성한 뷰를 터치하면 Touch 이벤트가 발생하므로 이를 처리

CLASS		METHOD/DEFINITION
View	View 클래스	
	<code>boolean onTouchEvent(MotionEvent event)</code>	MotionEvent 객체로 터치 이벤트를 처리
	<code>int getAction()</code>	터치한 동작을 얻음  MotionEvent.ACTION_DOWN 손가락을 누를 때 MotionEvent.ACTION_MOVE 터치한 상태에서 드래그 MotionEvent.ACTION_UP 손가락을 떼를 때
	<code>float getX(), getY()</code>	이벤트 좌표 리턴

# 1. 그래픽 | 터치 이벤트

## 터치 이벤트 형식 예시

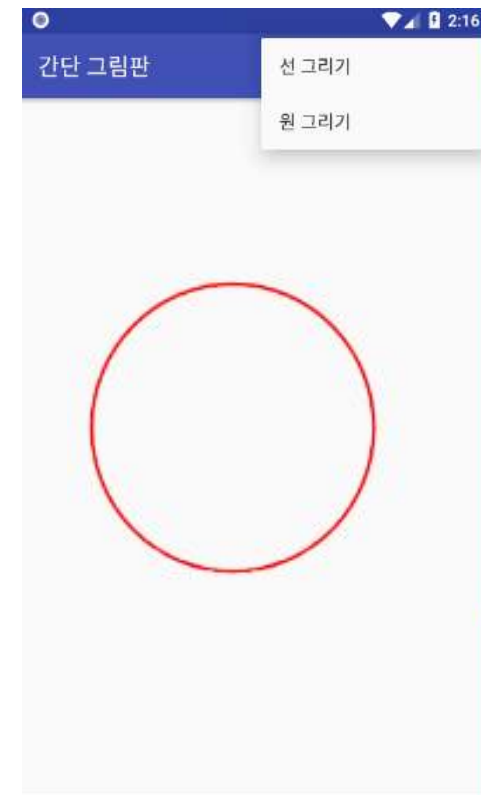
```
public boolean onTouchEvent(MotionEvent event) {  
    switch (event.getAction()) {  
        case MotionEvent.ACTION_DOWN:  
            // 손가락으로 화면을 누르기 시작했을 때 할 일  
            break;  
        case MotionEvent.ACTION_MOVE:  
            // 터치 후 손가락을 움직일 때 할 일  
            break;  
        case MotionEvent.ACTION_UP:  
            // 손가락을 화면에서 뗄 때 할 일  
            break;  
        case MotionEvent.ACTION_CANCEL:  
            // 터치가 취소될 때 할 일  
            break;  
        default:  
            break;  
    }  
    return true;  
}
```

# 1. 그래픽 | 터치 이벤트

## 실습9-1: Poject09\_01

옵션메뉴에서 선택하여 도형 그리기

activity\_main.xml 삭제



# 1. 그래픽 | 터치 이벤트

## 실습9-1: Poject09\_01 MainActivity.java

```
15 public class MainActivity extends AppCompatActivity {  
16  
17     final static int LINE = 1, CIRCLE = 2;  
18     static int curShape = LINE;  
19  
20     @Override  
21     protected void onCreate(Bundle savedInstanceState) {  
22         super.onCreate(savedInstanceState);  
23         setContentView(new MyGraphicView(this));  
24         setTitle("간단 그림판");  
25     }
```

# 1. 그래픽 | 터치 이벤트

실습9-1: Poject09\_01

MainActivity.java

옵션 메뉴('선 그리기', '원 그리기') 생성 및 항목 선택 이벤트 처리

```
26  
27     @Override  
28  
29  
30  
31  
32  
33  
34     }  
35  
36     @Override  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51
```

옵션메뉴를 생성하는 onCreateOptionsMenu(~)  
완성

메뉴의 항목 이벤트를 처리하는  
onOptionsItemSelected 완성

- 각 항목 선택 시, curShape에 적절한 상수 대입
- 선/원 그리기

# 1. 그래픽 | 터치 이벤트

## 실습9-1: Project09\_01

### MainActivity.java

MainActivity 내부에 View 클래스의 상속을 받는 MyGraphicView 클래스를 작성  
onTouchEvent() 메소드를 자동완성 후 나머지를 코딩

```
52  
53 public class MyGraphicView extends View {  
54  
55     int startX = -1, startY = -1, stopX = -1, stopY = -1;  
56  
57     public MyGraphicView(Context context) { super(context); }  
60  
61     @Override  
62     public boolean onTouchEvent(MotionEvent event) {  
63         switch (event.getAction()) {  
64             case MotionEvent.ACTION_DOWN:  
65                 startX = (int) event.getX();  
66                 startY = (int) event.getY();  
67                 break;  
68             case MotionEvent.ACTION_MOVE:  
69             case MotionEvent.ACTION_UP:  
70                 stopX = (int) event.getX();  
71                 stopY = (int) event.getY();  
72                 this.invalidate(); onDraw() 호출됨  
73                 break;  
74         }  
75         return true;  
76     }  
}
```



# 1. 그래픽 | 터치 이벤트

## 실습9-1: Poject09\_01

### MainActivity.java

MyGraphicView에 onDraw( )로 선, 원 그리기

페인트에 선의 두께, 채우기 여부, 선의 색상을 지정

switch( )~case문으로 메뉴에서 선택한 내용에 따라 선 또는 원을 그림

```
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
  
@Override  
protected void onDraw(Canvas canvas) { MyGraphicView가 생성될 때 호출되어, Paint 설정,  
                                         TouchEvent에서 invalidate()로 호출되어 draw  
    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;  
    }  
}
```

# 1. 그래픽 | 터치 이벤트

## 직접 풀어보기9-1: Test09\_02

[실습 9-1]을 다음과 같이 수정한다.

- 색상이 옵션 메뉴에서 선택되도록 한다. 색상은 서브 메뉴로 나오게 하고 빨강, 초록, 파랑 세 가지만 사용한다.
- 사각형 그리기를 추가한다.

