

10주차 정리

캔버스와 페인트

도형을 그릴 때 사용한다.



기본 메소드

android.graphics.Canvas 클래스의 점 찍는 메소드

```
public void drawPoint (float x, float y, Paint paint)
```

android.graphics.Paint 클래스에서 색상을 지정하는 메소드

```
public void setColor (int color)
```

캔버스와 페인트 코드

그래픽을 표현할 때는 View 클래스를 재정의하는 형태를 많이 사용함.

MainActivity 대신 우리가 캔버스로 사용하기 위해 재정의한 클래스 이름을 setContentView에 넣는다. → 안드로이드 화면 전체가 캔버스가 됨.

1. 커스텀 (재정의) 클래스에 extends View
2. onDraw 오버라이딩
3. setContentView에 재정의한 클래스 이름 넣기

```
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);  
        // 화면에 그려질 내용을 이곳에 코딩  
    }  
}
```

page 8

터치 이벤트

화면에 생성한 뷰를 터치하면 Touch 이벤트가 발생

View 클래스의 onTouchEvent() 메서드를 오버라이드해서 코딩

```
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 땔 때 할 일
            break;
        case MotionEvent.ACTION_CANCEL:
            // 터치가 취소될 때 할 일
            break;
        default:
            break;
    }
    return true;
}
```

MotionEvent.ACTION_DOWN - 화면 눌렀을 때

MotionEvent.ACTION_MOVE - 손가락을 움직일 때

MotionEvent.ACTION_UP - 화면에서 손 땔 때

MotionEvent.ACTION_CANCEL - 터치가 취소될 때

예제 9-4 Java 코드 3

```
1 private static class MyGraphicView extends View {
2     int startX = -1, startY = -1, stopX = -1, stopY = -1;
3     public MyGraphicView(Context context) {
4         super(context);
5     }
6     @Override
7     public boolean onTouchEvent(MotionEvent event) {
8         switch (event.getAction()) {
9             case MotionEvent.ACTION_DOWN:
10                startX = (int) event.getX();
11                startY = (int) event.getY();
12                break;
13             case MotionEvent.ACTION_MOVE:
14             case MotionEvent.ACTION_UP:
15                stopX = (int) event.getX();
16                stopY = (int) event.getY();
17                this.invalidate();
18                break;
19            }
20            return true;
21        }
22    }
23 }
```

this.invalidate(): 화면을 업데이트 함. 현재의 화면을 무효로 만들 → OnDraw가 실행되어 화면이 새롭게 그려진다.

paint.setAntiAlias(true); - 도형의 끝을 부드럽게 처리한다.

paint.setColor(Color.GREEN); - 한 번 지정하면 다른 색으로 지정할 때까지 유지한다.

paint.setStyle(Paint.Style.FILL); - 도형 내부를 채운다.

비트맵

비트맵 클래스는 캔버스에 이미지 파일을 보여주기 위해서 사용

BitmapFactory.decodeResource(getResources(), R.drawable.그림id); → /res/drawable 이미지 접속

BitmapFactory.decodeFile(파일 경로) → SD카드 이미지 접속

canvas.drawBitmap(picture, 시작x, 시작y, null)

/res/drawable 폴더에 있는 이미지 파일을 보여주는 **onDraw()** 메소드

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

    Bitmap picture = BitmapFactory.decodeResource(getResources(),
        R.drawable.그림id);
    canvas.drawBitmap(picture, 시작x, 시작y, null);
    picture.recycle();
}
```

이미지를 화면 중앙에 출력하기

```
시작x = (View 너비 - 이미지 너비) / 2
시작y = (View 높이 - 이미지 높이) / 2
```

실사용 코드

/res/drawable 사진을 올려놓는다.

```
10  @Override
11  protected void onDraw(Canvas canvas) {
12      super.onDraw(canvas);
13      Bitmap picture = BitmapFactory.decodeResource(getResources(),
14          R.drawable.jeju14);
15      int picX = (this.getWidth() - picture.getWidth()) / 2;
16      int picY = (this.getHeight() - picture.getHeight()) / 2;
17      canvas.drawBitmap(picture, picX, picY, null);
18      picture.recycle();
19  }
```

picture.recycle - 비트맵 리소스를 해제한다.

이미지의 기하학적 변환

회전: rotate()

확대/축소: scale()

이동: translate()

기울이기: skew()

이미지 블러링

이미지를 **뿌옇게** 만드는 것으로 BlurMaskFilter 클래스 사용
NORMAL, INNER, OUTER, SOLID

```
BlurMaskFilter(반지름, 스타일);
```

이미지 엠보싱

이미지 **볼록하게** 튀어나와 보이는 효과, EmbossMaskFilter 클래스 제공, 그림자 효과

32p의 코드를 따라 쳐도 엠보싱이 적용되지 않는데 이때 아래 코드를 **Manifest**파일에 추가하자.

Manifest파일에 **<application android:hardwareAccelerated = "false"/>**를 지정해주어야 화면에 엠보싱 효과가 나타난다. (하드웨어 가속 false)

```
EmbossMaskFilter(빛의 xyz 방향 일차 배열, 빛의 밝기, 반사 계수, 블러링 크기);
```

컬러 매트릭스

ColorMatrix는 canvas.setColorFilter(new ColorMatrixColorFilter(cm)); 을 사용한다.
setColor 아님

- 색상, 밝기 조절 위해 ColorMatrix와 ColorMatrixColorFilter클래스 사용

```
Paint paint = new Paint();
float[] array = { 4 x 5 배열 };
ColorMatrix cm = new ColorMatrix(array);
paint.setColorFilter(new ColorMatrixColorFilter(cm));
canvas.drawBitmap(...);
```

- ColorMatrix에 사용할 배열(Array)의 각 위치의 값

Red (1)	0	0	0	Brightness(0)
0	Green (1)	0	0	Brightness(0)
0	0	Blue (1)	0	Brightness(0)
0	0	0	Alpha(1)	0

```
10 Paint paint = new Paint();
11 float[] array = { 2 , 0 , 0 , 0 , -25 ,
12                  0 , 2 , 0 , 0 , -25 ,
13                  0 , 0 , 2 , 0 , -25 ,
14                  0 , 0 , 0 , 1 , 0  };
15 ColorMatrix cm = new ColorMatrix(array);
16 paint.setColorFilter(new ColorMatrixColorFilter(cm));
17 canvas.drawBitmap(picture, picX, picY, paint);
18 picture.recycle();
19 }
```

확대 or 축소

Canvas.scale(scaleX, scaleY, cenX, cenY)

Canvas.rotate(angle, cenX, cenY)

ColorMatrix.setSaturation() - 흑백 ex cm.setSaturRation(0) or (1)

setColorFilter 예시

예제 9-15 Java 코드 4

```
1 ~~~~ 생략 ~~~~
2 static float color=1;
3
4 ~~~~ 생략 ~~~~
5 ibBright = (ImageButton) findViewById(R.id.ibBright);
6 ibBright.setOnClickListener(new View.OnClickListener() {
7     public void onClick(View v) {
8         color = color + 0.2f;
9         graphicView.invalidate();
10    }
11 });
```

```
12
13 ~~~~ 생략 ~~~~
14 Paint paint = new Paint();
15 float[] array = { color , 0 , 0 , 0 , 0 ,
16                  0 , color , 0 , 0 , 0 ,
17                  0 , 0 , color , 0 , 0 ,
18                  0 , 0 , 0 , 1 , 0  };
19 ColorMatrix cm = new ColorMatrix(array);
20 paint.setColorFilter(new ColorMatrixColorFilter(cm));
21
22 ~~~~ 생략 ~~~~
23 canvas.drawBitmap(picture, picX, picY, paint);
```



생략된 코드

Paint.setColorFilter ~~

이미지를 화면에 표시하는 코드

리소스를 정리하는 코드

p36

3. 이미지 활용

- 실습 9-2 미니 포토샵 앱 만들기
- 2 화면 디자인 및 편집
 - (2) AndroidManifest.xml에 아이콘으로 사용할 그림 파일의 id로 변경 및 하드웨어 가속기 기능을 끄

```
android:icon="@drawable/그림 파일 아이디"
android:hardwareAccelerated="false"
```