

Ch09. 그래픽과 이미지

모바일게임프로그래밍
김지심 교수

학습목표

캔버스에 도형을 그릴 수 있다.
이미지 파일을 처리할 수 있다.

목차

01 그래픽
02 이미지



02



이미지



2. 이미지 | 비트맵 기본

비트맵(Bitmap)

CLASS	METHOD/DEFINITION
Bitmap	<p>그래픽 출력의 최종 결과인 픽셀 정보를 가진 클래스 캔버스에 이미지 파일을 보여주기 위해서 주로 사용</p> <p>Bitmap BitmapFactory. decodeResource(Resources res, int id) /res/drawable 폴더에 있는 이미지 파일을 보여줌(onDraw() 메소드에서 사용)</p> <p>Bitmap BitmapFactory. decodeFile(String pathName) SD카드의 이미지 파일을 보여줌(onDraw() 메소드에서 사용)</p> <p>Canvas void drawBitmap(Bitmap bitmap, float left, float top, Paint paint) 비트맵을 출력</p> <p>void recycle() 비트맵 리소스를 해제 * (출력물이 사라지는 게 아니라,) 이미지를 메모리에서 해제할 수 있도록 참조를 해제하여 GC대상이 됨</p>
Bitmap Factory	여러 리소스로부터 Bitmap 객체를 얻어 옴

2. 이미지 | 비트맵 기본

비트맵(Bitmap)

형식 예시

```
protected void onDraw(Canvas canvas) {  
    super.onDraw(canvas);  
  
    Bitmap picture = BitmapFactory.decodeResource(getResources(),  
R.drawable.그림id);  
    canvas.drawBitmap(picture, 시작x, 시작y, null);  
    picture.recycle();  
}
```

```
protected void onDraw(Canvas canvas) {  
    super.onDraw(canvas);  
  
    Bitmap picture = BitmapFactory.decodeFile("파일경로 및 파일");  
    canvas.drawBitmap(picture, 시작x, 시작y, null);  
    picture.recycle();  
}
```

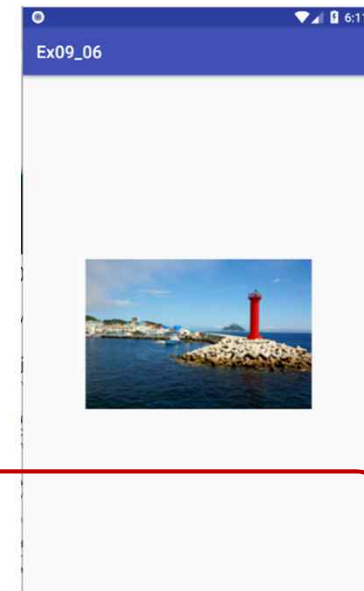
2. 이미지 | 비트맵 기본

예제9-6: Ex09_06

화면 중앙에 이미지 파일을 출력
jeju14.jpg를 /res/drawable에 복사

MainActivity.java

```
11 public class MainActivity extends AppCompatActivity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(new MyPictureView(this));
17     }
18
19     private class MyPictureView extends View {
20         public MyPictureView(Context context) { super(context); }
21
22
23
24         @Override
25
26         protected void onDraw(Canvas canvas) {
27             super.onDraw(canvas);
28             Bitmap picture = BitmapFactory.decodeResource(getResources(), R.drawable.jeju14);
29
30             int cenX = this.getWidth() / 2;
31             int cenY = this.getHeight() / 2;
32
33             int picX = (this.getWidth() - picture.getWidth()) / 2;
34             int picY = (this.getHeight() - picture.getHeight()) / 2;
35
36             canvas.drawBitmap(picture, picX, picY, null);
37             picture.recycle();
38         }
39     }
40 }
```



2. 이미지 | 이미지의 기하학적 변환

기하학적 변환

이미지가 아닌 캔버스에 기하학적 변환을 주고 이미지 파일을 변환된 캔버스에 출력(Canvas 클래스 사용)

[Reference] Methods

`void rotate(float degrees, float px, float py)` 회전

`void translate(float dx, float dy)` 이동

`void scale(float sx, float sy, float px, float py)` 확대/축소

`void skew(float sx, float sy)` 기울이기



(a) 회전



(b) 이동



(c) 확대



(d) 기울이기

2. 이미지 | 이미지의 기하학적 변환

예제9-7: Ex09-06의 OnDraw()메소드를 수정하여 테스트해보자.

```
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51
```

```
@Override  
protected void onDraw(Canvas canvas) {  
    super.onDraw(canvas);  
    Bitmap picture = BitmapFactory.decodeResource(getResources(), R.drawable.jeju14);  
  
    int cenX = this.getWidth() / 2;  
    int cenY = this.getHeight() / 2;  
  
    int picX = (this.getWidth() - picture.getWidth()) / 2;  
    int picY = (this.getHeight() - picture.getHeight()) / 2;  
  
    canvas.rotate(45, cenX, cenY);  
    canvas.drawBitmap(picture, picX, picY, null);  
  
    canvas.translate(-150, 200);  
    canvas.drawBitmap(picture, picX, picY, null);  
  
    canvas.scale(2, 2, cenX, cenY);  
    canvas.drawBitmap(picture, picX, picY, null);  
  
    canvas.skew(0.3f, 0.3f);  
    canvas.drawBitmap(picture, picX, picY, null);  
  
    picture.recycle();  
}
```

한 블럭씩 실행하여 테스트하기

2. 이미지 | 이미지 효과

블러링(Blurring)

이미지를 뿌옇게 만드는 것으로 BlurMaskFilter 클래스 사용(가장자리 처리)

CLASS	METHOD/DEFINITION
BlurMaskFilter	블러링 효과 지원 클래스
BlurMaskFilter(float radius, BlurMaskFilter.Blur style)	생성자(영역, 스타일 설정)
MaskFilter setMaskFilter (MaskFilter maskfilter)	Paint객체에 효과가 설정된 마스크필터 설정

블러 효과(ex. BlurMaskFilter.Blur.NORMAL)



2. 이미지 | 이미지 효과

* 이미지 효과 적용시 유의사항

hardwareAccelerated 속성을 false로 변경해야 효과가 적용됨

(API14부터 기본값이 true로 설정되어 있음)

(Here is the official word from the Android graphics team: "BlurMaskFilter is not supported with hardware acceleration." (As of July 10, 2012)
It renders correctly if you set android:hardwareAccelerated="false" on your Activity in AndroidManifest.xml.)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="kjs.mjc.ackr.ex09_10">
4
5     <application
6         android:hardwareAccelerated="false"
7         android:allowBackup="true"
8         android:icon="@mipmap/ic_launcher"
9         android:label="Ex09_10"
10        android:roundIcon="@mipmap/ic_launcher_round"
11        android:supportsRtl="true"
12        android:theme="@style/AppTheme">
13         <activity android:name=".MainActivity">
14             <intent-filter>
15                 <action android:name="android.intent.action.MAIN" />
16
17                 <category android:name="android.intent.category.LAUNCHER" />
18             </intent-filter>
19         </activity>
20     </application>
21 </manifest>
```

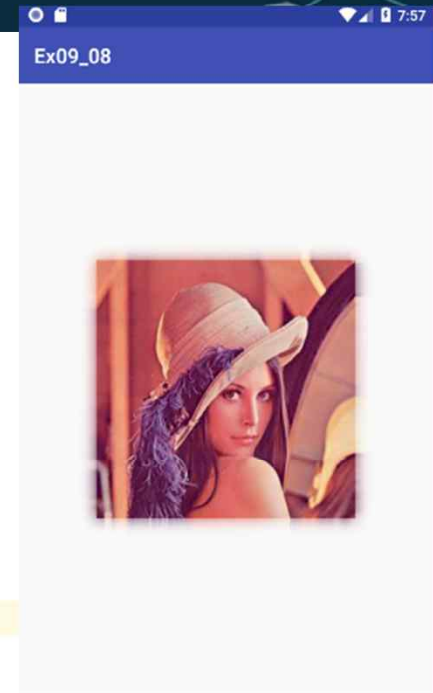
2. 이미지 | 이미지 효과

예제9-10:Ex09_08

블러링 효과 적용(without activity_main.xml)

MainActivity.java

```
13  <> public class MainActivity extends AppCompatActivity {  
14  
15      @Override  
16      public void onCreate(Bundle savedInstanceState) {  
17          super.onCreate(savedInstanceState);  
18          setContentView(new MyGraphicView(this));  
19      }  
20  
21      private static class MyGraphicView extends View {  
22          public MyGraphicView(Context context) { super(context); }  
23  
24  
25  
26      @Override  
27      protected void onDraw(Canvas canvas) {  
28          super.onDraw(canvas);  
29          Bitmap picture = BitmapFactory.decodeResource(getResources(), R.drawable.lena256);  
30
```



2. 이미지 | 이미지 효과

예제9-10:Ex09_08
MainActivity.java

```
31      int picX = (this.getWidth() - picture.getWidth()) / 2;
32      int picY = (this.getHeight() - picture.getHeight()) / 2;
33
34      Paint paint = new Paint();
35
36
37      BlurMaskFilter bMask;
38
39      //
40      //
41      //
42
43      // bMask = new BlurMaskFilter(30, BlurMaskFilter.Blur.NORMAL);
44      // paint.setMaskFilter(bMask);
45      // canvas.drawBitmap(picture, picX, picY, paint);
46      // picture.recycle();
47      //
48      // bMask = new BlurMaskFilter(30, BlurMaskFilter.Blur.OUTER);
49      // paint.setMaskFilter(bMask);
50      // canvas.drawBitmap(picture, picX, picY, paint);
51      // picture.recycle();
52      //
53      bMask = new BlurMaskFilter(30, BlurMaskFilter.Blur.SOLID);
54      paint.setMaskFilter(bMask);
55      canvas.drawBitmap(picture, picX, picY, paint);
56      picture.recycle();
57  }
58
59 }
```

한 블럭씩 실행하여 테스트하기

2. 이미지 | 이미지 효과

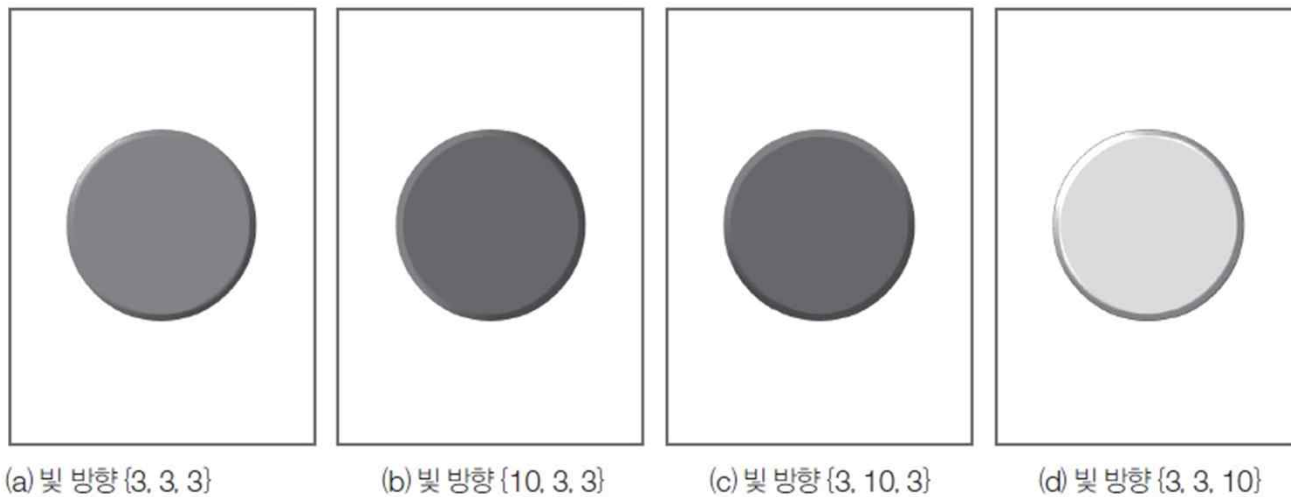
엠보싱(Embossing)

이미지가 볼록하게 튀어나와 보이는 효과로, EmbosMaskFilter 클래스 사용

[Reference] Methods

EmbossMaskFilter(float[] direction, float ambient, float specular, float blurRadius) 생성자(빛의 xyz 방향, 빛의 밝기, 반사계수, 블러링 크기)

엠보싱 효과



2. 이미지 | 이미지 효과

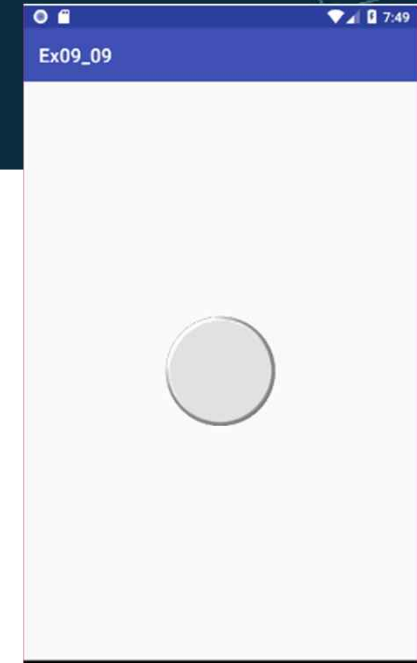
예제9-10:Ex09_09

엠보싱 효과 적용(without activity_main.xml)

매니페스트에 가속화 false 설정

MainActivity.java

```
12  public class MainActivity extends AppCompatActivity {  
13  
14      @Override  
15      public void onCreate(Bundle savedInstanceState) {  
16          super.onCreate(savedInstanceState);  
17          setContentView(new MyGraphicView(this));  
18      }  
19  
20      private static class MyGraphicView extends View {  
21          public MyGraphicView(Context context) { super(context); }  
22  
23  
24  
25      @Override  
26      protected void onDraw(Canvas canvas) {  
27          super.onDraw(canvas);  
28  
29          int cenX = this.getWidth() / 2;  
30          int cenY = this.getHeight() / 2;  
31
```



2. 이미지 | 이미지 효과

예제9-10:Ex09_09
MainActivity.java

```
36 // eMask = new EmbossMaskFilter(new float[] { 3, 3, 3 }, 0.5f, 5, 10);
37 // paint.setMaskFilter(eMask);
38 // canvas.drawCircle(cenX, cenY, 150, paint);
39
40 // eMask = new EmbossMaskFilter(new float[] {10,3,3}, 0.5f, 5, 10);
41 // paint.setMaskFilter(eMask);
42 // canvas.drawCircle(cenX, cenY, 150, paint);
43
44 // eMask = new EmbossMaskFilter(new float[] {3,10,3}, 0.5f, 5, 10);
45 // paint.setMaskFilter(eMask);
46 // canvas.drawCircle(cenX, cenY, 150, paint);
47
48
49
50 eMask = new EmbossMaskFilter(new float[] {3,3,10}, 0.5f, 5, 10);
51 paint.setMaskFilter(eMask);
52 canvas.drawCircle(cenX, cenY, 150, paint);
53
54 }
55
56 }
```

한 블록씩 실행하여 테스트하기

2. 이미지 | 이미지 효과

컬러매트릭스

색상, 밝기를 조절하기 위해 `ColorMatrix`, `ColorMatrixColorFilter` 클래스 사용

[Reference] Class

`ColorMatrix` 4*5 매트릭스(1차원배열 형태)로 색상과 투명도를 조정

`ColorMatrixColorFilter` `ColorMatrix`를 적용하여 컬러필터 정보를 담음

[Reference] Method

`ColorFilter setColorFilter(ColorFilter filter)` 컬러필터 설정(`ColorMatrix`가 적용된 `ColorMatrixColorFilter` 객체를 `Paint`객체에 설정)

2. 이미지 | 이미지 효과

컬러매트릭스 형식 예시

```
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

+: 밝게, -: 어둡게

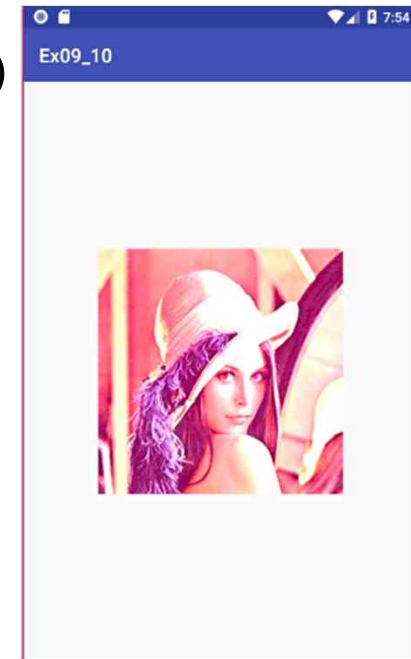
클수록 대비 커짐

2. 이미지 | 이미지 효과

예제9-10:Ex09_10

RGB 색상 대비를 2배로 변경(without activity_main.xml)
MainActivity.java

```
17 public class MainActivity extends AppCompatActivity {  
18  
19     @Override  
20     public void onCreate(Bundle savedInstanceState) {  
21         super.onCreate(savedInstanceState);  
22         setContentView(new MyGraphicView(this));  
23     }  
24  
25     private static class MyGraphicView extends View {  
26         public MyGraphicView(Context context) {  
27             super(context);  
28         }  
29  
30     @Override  
31     protected void onDraw(Canvas canvas) {  
32         super.onDraw(canvas);  
33     }  
34 }
```



2. 이미지 | 이미지 효과

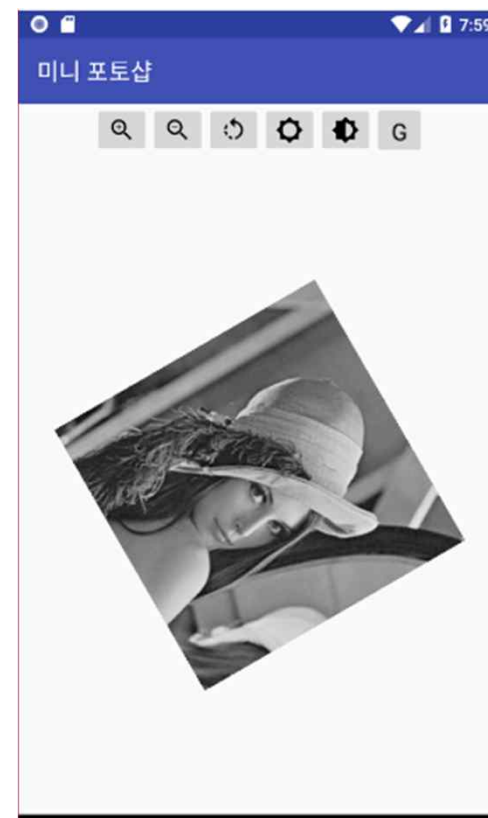
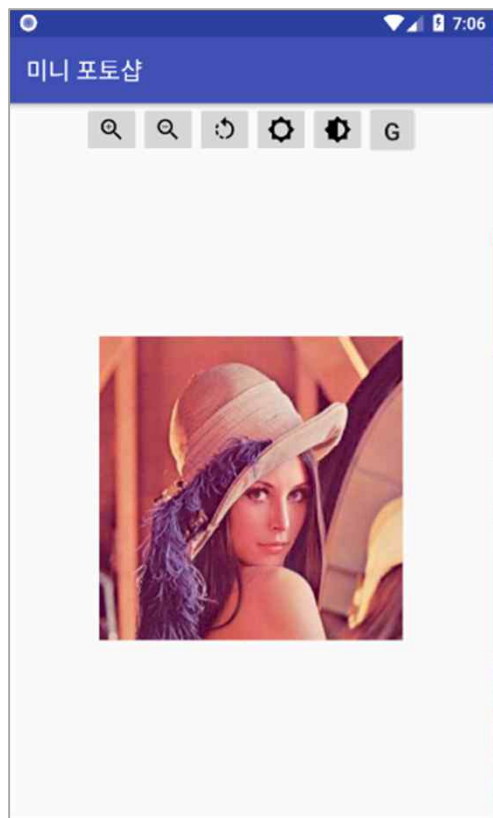
예제9-10:Ex09_10 MainActivity.java

```
34      Bitmap picture = BitmapFactory.decodeResource(getResources(), R.drawable.lena256);
35
36      int picX = (this.getWidth() - picture.getWidth()) / 2;
37      int picY = (this.getHeight() - picture.getHeight()) / 2;
38
39      Paint paint = new Paint();
40
41      float[] array = {2, 0, 0, 0, -25,
42                      0, 2, 0, 0, -25,
43                      0, 0, 2, 0, -25,
44                      0, 0, 0, 1, 0};
45      ColorMatrix cm = new ColorMatrix(array);
46      paint.setColorFilter(new ColorMatrixColorFilter(cm));
47      canvas.drawBitmap(picture, picX, picY, paint);
48      picture.recycle();
49  }
50  }
51  }
```

2. 이미지 | 이미지 효과

실습9-2: Project09_02

미니 포토샵(효과 버튼을 클릭하여 이미지에 효과 주기)



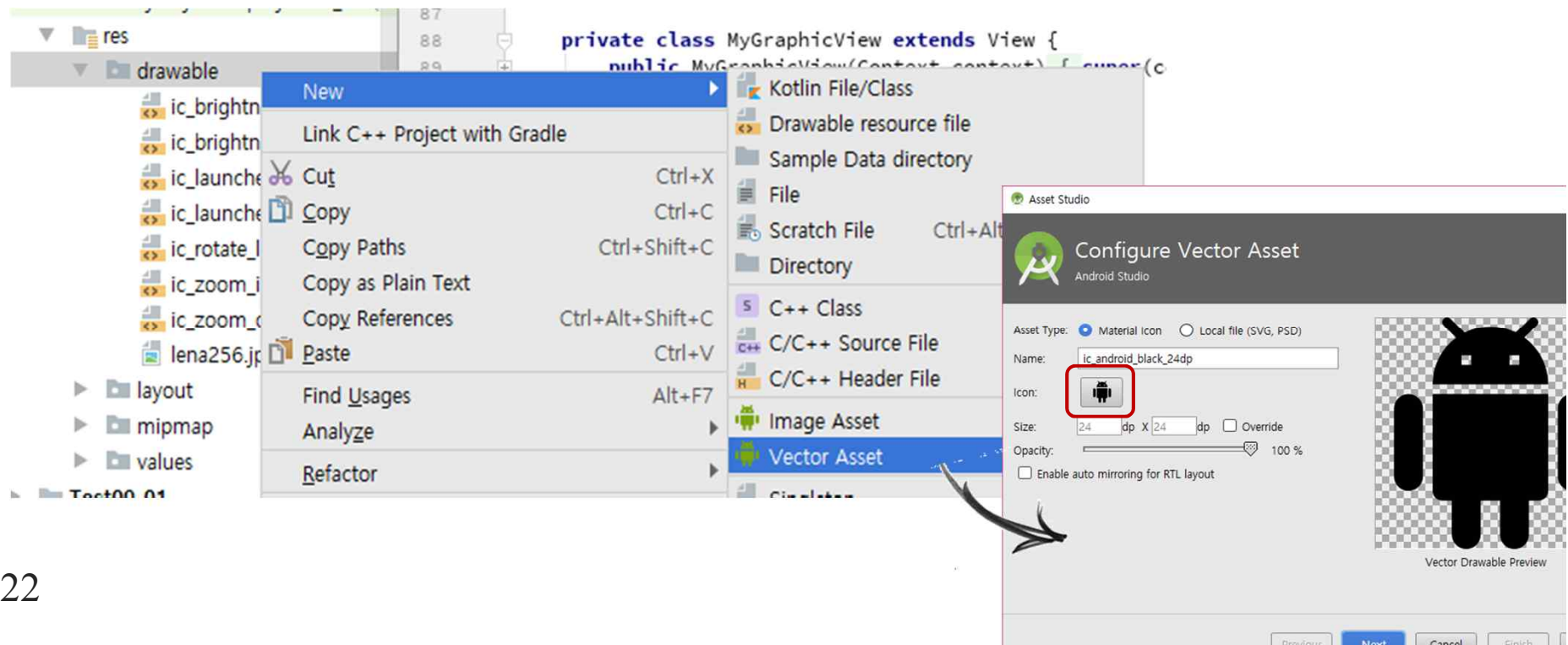
2. 이미지 | 이미지 효과

실습9-2: Project09_02

매니페스트에 가속화 false 설정

activity_main.xml

ImageButton에 안드로이드 스튜디오의 Vector asset을 이용하기 위해,
아래와 같이 필요한 벡터 이미지 추가

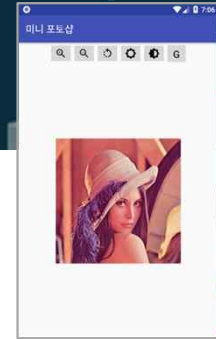


2. 이미지 | 이미지 효과

실습9-2: Project09_02

activity_main.xml

커스텀뷰(MyGraphicView)를 'pictureLay' 레이아웃에 설정할 예정



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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center">
        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/ibZoomin"
            android:src="@drawable/ic_zoom_in_black_24dp"/>
        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/ibZoomout"
            android:src="@drawable/ic_zoom_out_black_24dp"/>
        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/ibRotate"
            android:src="@drawable/ic_rotate_left_black_24dp"/>

    </LinearLayout>
```

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/ibBright"
    android:src="@drawable/ic_brightness_5_black_24dp"/>
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/ibDark"
    android:src="@drawable/ic_brightness_6_black_24dp"/>
<Button
    android:layout_width="44dp"
    android:layout_height="44dp"
    android:id="@+id/ibGray"
    android:textSize="20dp"
    android:text="G"/>
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:id="@+id/pictureLay"
    android:gravity="center"/>
</LinearLayout>
```


2. 이미지 | 이미지 효과

실습9-2: Project09_02 MainActivity.java

```
17 public class MainActivity extends AppCompatActivity {  
18  
19     MyGraphicView graphicView;  
20  
21     static float scaleX = 1, scaleY = 1; //스케일링변수 (Zoom)  
22     static float angle = 0; //Rotate 변수  
23     static float color = 1;  
24     static float satur = 1;  
25  
26     @Override  
27     protected void onCreate(Bundle savedInstanceState) {  
28         super.onCreate(savedInstanceState);  
29         setContentView(R.layout.activity_main);  
30         setTitle("미니 포토샵");  
31  
32         LinearLayout pictureLayout = findViewById(R.id.pictureLay);  
33         graphicView = new MyGraphicView(this);  
34         pictureLayout.addView(graphicView);  
35  
36         cilckIcons(); 사용자 정의 메소드  
37     }  
38 }
```

MyGraphicView를 리니어레이아웃에 추가

2. 이미지 | 이미지 효과

실습9-2: Project09_02

MainActivity.java

이벤트 처리를 위해
clickIcons()
메소드를 정의

```
39  
40  
43  
44  
45  
46  
47  
49  
52  
53  
54  
55  
57  
60  
61  
62  
64  
67  
68  
69  
71  
74  
75  
76  
78  
81  
82  
83  
84  
86  
private void clickIcons() {  
    findViewById(R.id.ibZoomin).setOnClickListener((view) -> {  
        scaleX = scaleX + 0.2f;  
        scaleY = scaleY + 0.2f;  
        graphicView.invalidate();  
    });  
    findViewById(R.id.ibZoomout).setOnClickListener((view) -> {  
        scaleX = scaleX - 0.2f;  
        scaleY = scaleY - 0.2f;  
        graphicView.invalidate();  
    });  
    findViewById(R.id.ibRotate).setOnClickListener((view) -> {  
        angle = angle + 20;  
        graphicView.invalidate();  
    });  
    findViewById(R.id.ibBright).setOnClickListener((view) -> {  
        color = color + 0.2f;  
        graphicView.invalidate();  
    });  
    findViewById(R.id.ibDark).setOnClickListener((view) -> {  
        color = color - 0.2f;  
        graphicView.invalidate();  
    });  
    findViewById(R.id.ibGray).setOnClickListener((view) -> {  
        if (satur == 0) satur = 1;  
        else satur = 0;  
        graphicView.invalidate();  
    });  
}
```

2. 이미지 | 이미지 효과

실습9-2: Project09_02

MainActivity.java

내부에 MyGraphicView를 정의하고 onDraw로 이미지 파일 출력 처리

```
88 private class MyGraphicView extends View {
89     public MyGraphicView(Context context) { super(context); }
90
91
92
93     @Override
94     protected void onDraw(Canvas canvas) {
95         super.onDraw(canvas);
96
97         int cenX = this.getWidth() / 2;
98         int cenY = this.getHeight() / 2;
99
100         canvas.scale(scaleX, scaleY, cenX, cenY);
101         canvas.rotate(angle, cenX, cenY);
102
103         Paint paint = new Paint();
104
105         float[] array = {color, 0, 0, 0, 0,
106                         0, color, 0, 0, 0,
107                         0, 0, color, 0, 0,
108                         0, 0, 0, 1, 0};
109         ColorMatrix cm = new ColorMatrix(array);
110
111         if (satur == 0) cm.setSaturation(satur);
112
113         paint.setColorFilter(new ColorMatrixColorFilter(cm));
114
115         Bitmap picture = BitmapFactory.decodeResource(getResources(), R.drawable.lena256);
116
117         int picX = (this.getWidth() - picture.getWidth()) / 2;
118         int picY = (this.getHeight() - picture.getHeight()) / 2;
119
120         canvas.drawBitmap(picture, picX, picY, paint);
121         picture.recycle();
122     }
123 }
124
125
```

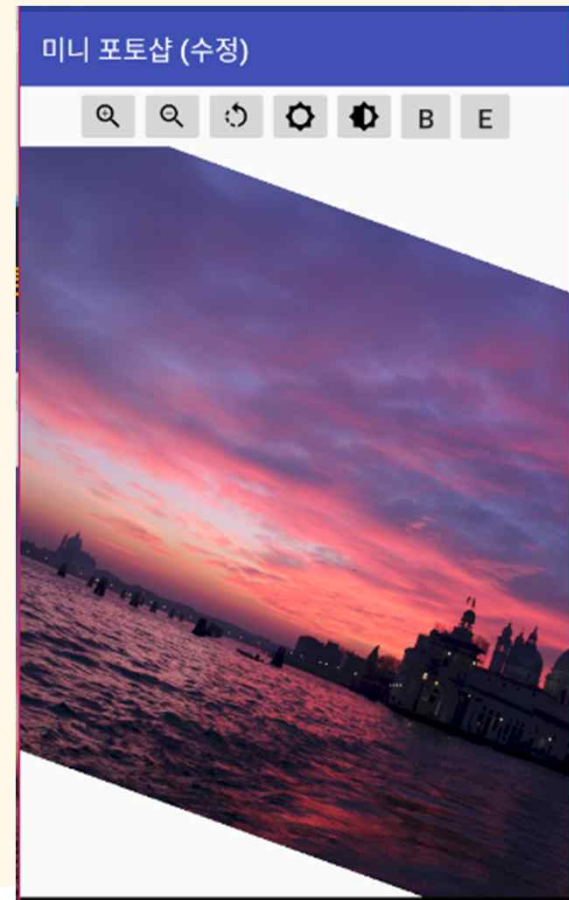
2. 이미지 | 이미지 효과

직접 풀어보기: Test09_03

▶ 직접 풀어보기 9-3

[실습 9-2]를 다음과 같이 수정한다.

- 회색 영상 버튼을 없앤다.
- 밝게 하기를 클릭하면 채도가 증가하고, 어둡게 하기를 클릭하면 채도가 감소하도록 코드를 수정한다.
- 블러링, 엠보싱 아이콘도 추가하고 코딩한다. 클릭하면 블러링 또는 엠보싱 기능이 온/오프 되도록 한다.



참고문헌

천인국(2015), 그림으로 쉽게 설명하는 안드로이드 프로그래밍(개정3판), 생능출판사.