

# Chapter09

## 그래픽과 이미지



# 학습목표

---

- 캔버스에 도형을 그리는 방법을 익힌다.
- 이미지 파일을 처리하는 방식을 알아본다.
- 영상처리 앱을 작성한다.

# 차례

---

- 01 그래픽
- 02 이미지

# 1. 그래픽 ▶ 캔버스와 페인트 기본

## ■ 캔버스와 페인트

- 화면에 도형을 그릴 때 사용되는 Canvas와 Paint 클래스



그림 9-1 Canvas와 Paint 클래스

# 1. 그래픽 ▶ 캔버스와 페인트 기본

---

- 캔버스와 페인트

- android.graphics.Canvas 클래스의 점을 찍는 메소드의 원형

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

- android.graphics.Paint 클래스에서 색상을 지정하는 메소드의 원형

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

# 1. 그래픽 ▶ 캔버스와 페인트 기본

## ■ 캔버스와 페인트

- 그래픽을 표현할 때는 View 클래스를 재정의하는 형태를 많이 사용

```
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

## ■ 그래픽 기본 Java 코드

- 그래픽을 출력할 때는 주로 View.onDraw( ) 메소드를 오버라이딩해서 사용

예제 9-1 그래픽 기본의 Java 코드

```
1 public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(new MyGraphicView(this));
4 }
5
6 private static class MyGraphicView extends View {
7     public MyGraphicView(Context context) {
8         super(context);
9     }
10
11     @Override
12     protected void onDraw(Canvas canvas) {
13         super.onDraw(canvas);
14         Paint paint = new Paint();
15         paint.setAntiAlias(true);
16         paint.setColor(Color.GREEN);
17         canvas.drawLine(10, 10, 300, 10, paint);
18
19         paint.setColor(Color.BLUE);
```



# 1. 그래픽 ▶ 그래픽 처리 기본

---

```
20    paint.setStrokeWidth(5);
21    canvas.drawLine(10, 30, 300, 30, paint);
22
23    paint.setColor(Color.RED);
24    paint.setStrokeWidth(0);
25
26    paint.setStyle(Paint.Style.FILL);
27    Rect rect1 = new Rect(10, 50, 10+100, 50+100);
28    canvas.drawRect(rect1, paint);
29
30    paint.setStyle(Paint.Style.STROKE);
31    Rect rect2 = new Rect(130, 50, 130+100, 50+100);
32    canvas.drawRect(rect2, paint);
33
34    RectF rect3 = new RectF(250, 50, 250+100, 50+100);
35    canvas.drawRoundRect(rect3, 20, 20, paint);
36
37    canvas.drawCircle(60, 220, 50, paint);
38
39    paint.setStrokeWidth(5);
```



# 1. 그래픽 ▶ 그래픽 처리 기본

---

```
40    Path path1 = new Path();
41    path1.moveTo(10, 290);
42    path1.lineTo(10+50, 290+50);
43    path1.lineTo(10+100, 290);
44    path1.lineTo(10+150, 290+50);
45    path1.lineTo(10+200, 290);
46    canvas.drawPath(path1, paint);
47
48    paint.setStrokeWidth(0);
49    paint.setTextSize(30);
50    canvas.drawText("안드로이드", 10, 390, paint);
51 }
52 }
```

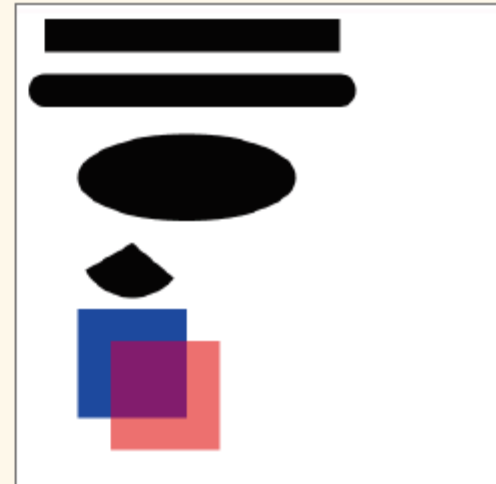
# 1. 그래픽 ▶ 그래픽 처리 기본

---

## ▶ 직접 풀어보기 9-1

그림과 같은 화면을 출력하도록 다음 메소드를 사용하여 Java를 코딩하라.

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



# 1. 그래픽 ▶ 터치 이벤트

---

## ■ 터치 이벤트

- 화면에 생성한 뷰를 터치하면 Touch 이벤트가 발생
- View 클래스의 onTouchEvent( ) 메소드를 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 뗄 때 할 일  
            break;  
        case MotionEvent.ACTION_CANCEL:  
            // 터치가 취소될 때 할 일  
            break;  
        default:  
            break;  
    }  
    return true;  
}
```

# 1. 그래픽 ▶ 터치 이벤트

## 실습 9-1 간단 그림판 앱 만들기

- 안드로이드 프로젝트 생성
  - (1) 프로젝트 이름 : Project9\_1
  - (2) 패키지 이름 : com.cookandroid.project9\_1
- 화면 디자인 및 편집
  - Java로만 코드 작성.
  - activity\_main.xml 파일 삭제해도 무방함

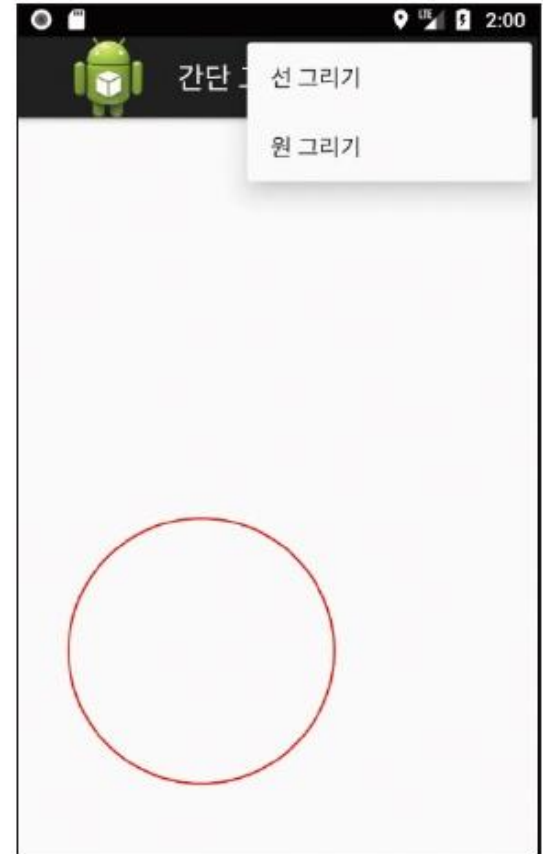


그림 9-2 간단 그림판 앱 결과 화면

# 1. 그래픽 ▶ 터치 이벤트

- Java 코드 작성 및 수정

- (1) View 클래스의 상속을 받는 MyGraphicView 클래스를 만들기

예제 9-2 간단 그림판의 Java 코드 1

```
1  ~~~~ 중간 생략(import문) ~~~~
2  public class MainActivity extends AppCompatActivity {
3      final static int LINE = 1, CIRCLE = 2;
4      static int curShape = LINE;
5
6      @Override
7      public void onCreate(Bundle savedInstanceState) {
8          super.onCreate(savedInstanceState);
9          setContentView(new MyGraphicView(this));
10         setTitle("간단 그림판");
11     }
12
13     private static class MyGraphicView extends View {
14         public MyGraphicView(Context context) {
15             super(context);
16         }
17
18     }
19 }
```

# 1. 그래픽 ▶ 터치 이벤트

---

- Java 코드 작성 및 수정

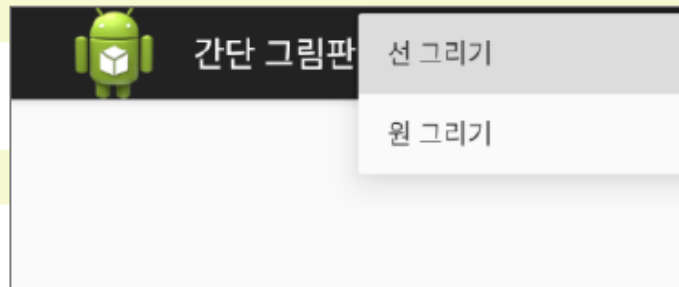
- (2) 옵션 메뉴 작성

- 선 그리기, 원 그리기 옵션 메뉴 만들기
    - 항목을 클릭하면 curShape 변수에 선택한 전역상수 대입
    - onCreate( )와 같은 레벨로 onCreateOptionsMenu( )와 onOptionsItemSelected( ) 메소드 자동 완성

# 1. 그래픽 ▶ 터치 이벤트

## 예제 9-3 간단 그림판의 Java 코드 2

```
1 public boolean onCreateOptionsMenu(Menu menu) {
2     super.onCreateOptionsMenu(menu);
3     menu.add(0, 1, 0, "선 그리기");
4     menu.add(0, 2, 0, "원 그리기");
5     return true;
6 }
7 public boolean onOptionsItemSelected(MenuItem item) {
8     switch (item.getItemId()) {
9         case 1:
10            curShape = LINE; // 선
11            return true;
12        case 2:
13            curShape = CIRCLE; // 원
14            return true;
15    }
16    return super.onOptionsItemSelected(item);
17 }
```



# 1. 그래픽 ▶ 터치 이벤트

## ■ Java 코드 작성 및 수정

- (3) MyGraphicView 클래스에 터치와 관련된 메소드를 완성
  - MyGraphicView의 전역변수 시작x, 시작y, 끝x, 끝y 및 반지름 변수를 선언
  - onTouchEvent( ) 메소드 자동 완성 후 자동 완성 외의 코드 완성

### 예제 9-4 간단 그림판의 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;
```



# 1. 그래픽 ▶ 터치 이벤트

---

```
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 }
```

# 1. 그래픽 ▶ 터치 이벤트

---

- Java 코드 작성 및 수정
  - (4) 실제로 화면에 도형이 그려질 onDraw( ) 메소드 완성
    - MyGraphicView의 내부에 onDraw( )를 자동 완성하고 나머지 코딩
    - 페인트에 선의 두께, 채우기 여부, 선의 색상 지정
    - switch( )~case문으로 메뉴에서 선택한 내용에 따라 선 또는 원 그림

# 1. 그래픽 ▶ 터치 이벤트

## 예제 9-5 간단 그림판의 Java 코드 4

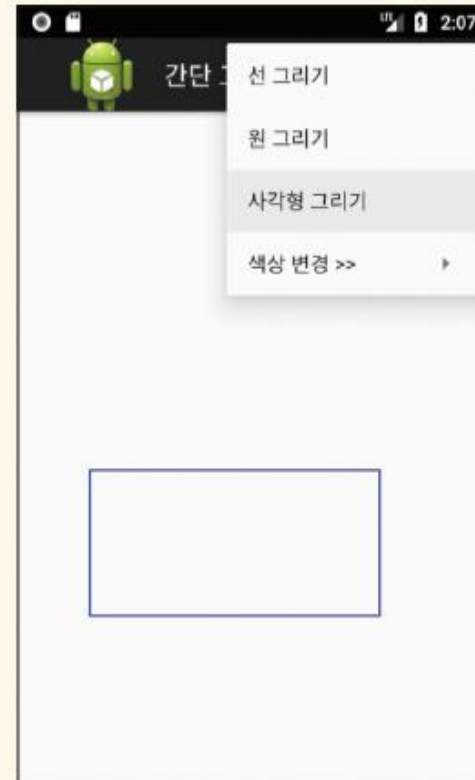
```
1  protected void onDraw(Canvas canvas) {
2      super.onDraw(canvas);
3      Paint paint = new Paint();
4      paint.setAntiAlias(true);
5      paint.setStrokeWidth(5);
6      paint.setStyle(Paint.Style.STROKE);
7      paint.setColor(Color.RED);
8
9      switch (curShape) {
10         case LINE:
11             canvas.drawLine(startX, startY, stopX, stopY, paint);
12             break;
13         case CIRCLE:
14             int radius = (int) Math.sqrt(Math.pow(stopX - startX, 2)
15                 + Math.pow(stopY - startY, 2));
16             canvas.drawCircle(startX, startY, radius, paint);
17             break;
18     }
19 }
```

# 1. 그래픽 ▶ 터치 이벤트

## ▶ 직접 풀어보기 9-2

[실습 9-1]을 다음과 같이 수정하라.

- 클릭한 두 점을 끝점으로 하는 사각형이 추가로 그려지게 한다.
- 색상이 옵션 메뉴에서 선택되게 한다. 색상은 서브 메뉴로 나오게 하고 빨강, 초록, 파랑만 사용한다.



## 2. 이미지 ▶ 비트맵의 기본

---

### ■ 비트맵(Bitmap)

- 비트맵(Bitmap) 클래스는 캔버스에 이미지 파일을 보여주기 위해서 사용
- /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();  
}
```

## 2. 이미지 ▶ 비트맵의 기본

- 비트맵(Bitmap)

- SD 카드의 이미지 파일을 보여주는 onDraw( ) 메소드

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

- 이미지를 화면 중앙에 출력하기

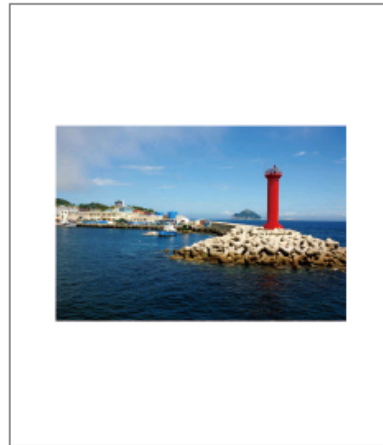
```
시작x = (View 폭 - 이미지 폭) / 2  
시작y = (View 높이 - 이미지 높이) / 2
```

## 2. 이미지 ▶ 비트맵의 기본

- jeju14.jpg를 /res/drawable에 미리 복사

예제 9-6 화면 중앙에 이미지 파일을 출력하는 Java 코드

```
1 public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(new MyGraphicView(this));
4 }
5
6 private static class MyGraphicView extends View {
7     public MyGraphicView(Context context) {
8         super(context);
9     }
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    }
20 }
```



## 2. 이미지 ► 이미지의 기하학적 변환

- 많이 사용되는 Canvas 클래스의 기하학적 메소드
  - 회전: `rotate( )`
  - 확대/축소: `scale( )`
  - 이동: `translate( )`
  - 기울이기: `skew( )`



(a) 회전



(b) 이동



(c) 확대



(d) 기울이기

그림 9-3 기하학적 변환 결과



## 2. 이미지 ► 이미지의 기하학적 변환

- [그림 9-3]을  
만드는 예제

예제 9-7 기하학적 변환의 Java 코드

```
1  protected void onDraw(Canvas canvas) {
2      super.onDraw(canvas);
3      Bitmap picture = BitmapFactory.decodeResource(getResources(), R.drawable.
        small);
4
5      int cenX = this.getWidth() / 2;
6      int cenY = this.getHeight() / 2;
7      int picX = (this.getWidth() - picture.getWidth()) / 2;
8      int picY = (this.getHeight() - picture.getHeight()) / 2;
9
10     canvas.rotate(45, cenX, cenY);
11     canvas.drawBitmap(picture, picX, picY, null);
12
13     canvas.translate(-150, 200);
14     canvas.drawBitmap(picture, picX, picY, null);
15
16     canvas.scale(2, 2, cenX, cenY);
17     canvas.drawBitmap(picture, picX, picY, null);
18
19     canvas.skew(0.3f, 0.3f);
20     canvas.drawBitmap(picture, picX, picY, null);
21
22     picture.recycle();
23 }
```

## 2. 이미지 ► 이미지 활용

### ■ 블러링

- 이미지를 뿌옇게 만드는 것으로 BlurMaskFilter 클래스 사용

```
BlurMaskFilter(반지름, 스타일);
```



(a) NORMAL



(b) INNER



(c) OUTER



(d) SOLID

그림 9-4 블러링 효과

## 2. 이미지 ► 이미지 활용

### 예제 9-8 블러링 효과의 Java 코드

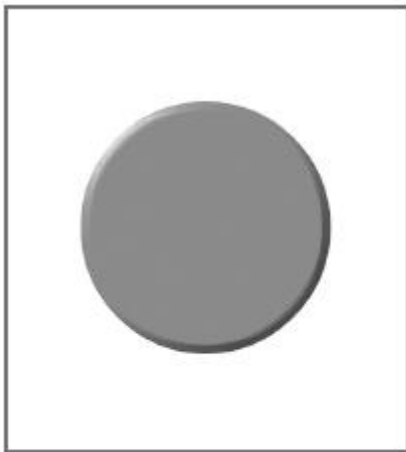
```
1  protected void onDraw(Canvas canvas) {
2      super.onDraw(canvas);
3      Bitmap picture = BitmapFactory.decodeResource(getResources(),
4          R.drawable.lena256);
5
6      int picX = (this.getWidth() - picture.getWidth()) / 2;
7      int picY = (this.getHeight() - picture.getHeight()) / 2;
8
9      Paint paint = new Paint();
10     BlurMaskFilter bMask;
11
12     bMask = new BlurMaskFilter(30, BlurMaskFilter.Blur.NORMAL);
13     paint.setMaskFilter(bMask);
14     canvas.drawBitmap(picture, picX, picY, paint);
15     picture.recycle();
16     ~~~~ 중간 생략(INNER, OUTER, SOLID 스타일) ~~~~
17 }
```

## 2. 이미지 ▶ 이미지 활용

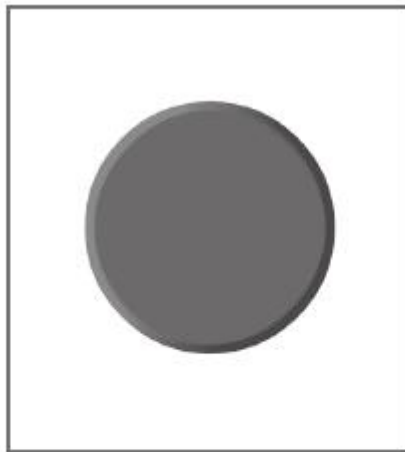
### ■ 엠보싱(Embossing)

- 이미지가 볼록하게 튀어나와 보이는 효과로, EmbossMaskFilter 클래스 제공

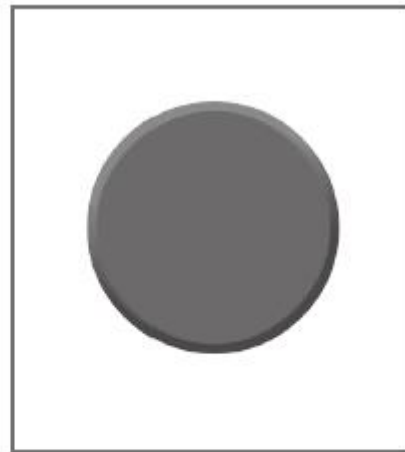
```
EmbossMaskFilter(빛의 xyz 방향 1차 배열, 빛의 밝기, 반사 계수, 블러링 크기);
```



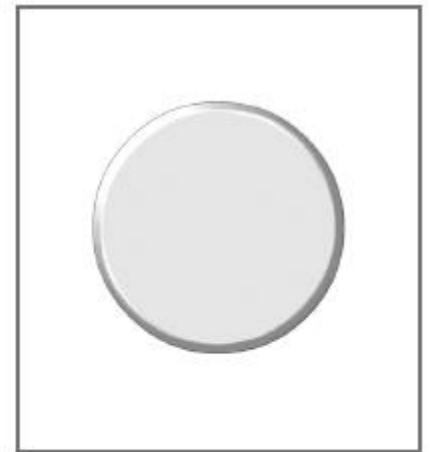
(a) 빛 방향 {3, 3, 3}



(b) 빛 방향 {10, 3, 3}



(c) 빛 방향 {3, 10, 3}



(d) 빛 방향 {3, 3, 10}

그림 9-5 엠보싱 효과

## 2. 이미지 ► 이미지 활용

### 예제 9-9 엠보싱 효과의 Java 코드

```
1  protected void onDraw(Canvas canvas) {
2      super.onDraw(canvas);
3
4      int cenX = this.getWidth() / 2 ;
5      int cenY = this.getHeight() / 2;
6
7      Paint paint = new Paint();
8      paint.setColor(Color.GRAY);
9      EmbossMaskFilter eMask;
10
11     eMask = new EmbossMaskFilter(new float[] {3,3,3}, 0.5f, 5, 10);
12     paint.setMaskFilter(eMask);
13     canvas.drawCircle(cenX, cenY, 150, paint);
14     ~~~~ 중간 생략(빛의 방향을 바꾼 세 가지) ~~~~
15 }
```

## 2. 이미지 ► 이미지 활용

### ■ 컬러매트릭스

- 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

## 2. 이미지 ► 이미지 활용

### ■ RGB 색상 대비를 2배로 변경하는 Java 코드

예제 9-10 컬러매트릭스의 Java 코드

```
1  protected void onDraw(Canvas canvas) {
2
3      super.onDraw(canvas);
4
5      Bitmap picture = BitmapFactory.decodeResource(getResources(),
6      R.drawable.lena256);
7
8      int picX = (this.getWidth() - picture.getWidth()) / 2;
9      int picY = (this.getHeight() - picture.getHeight()) / 2;
10
11     Paint paint = new Paint();
12     float[] array = { 2 , 0 , 0 , 0 , -25 ,
13                       0 , 2 , 0 , 0 , -25 ,
14                       0 , 0 , 2 , 0 , -25 ,
15                       0 , 0 , 0 , 1 , 0  };
16     ColorMatrix cm = new ColorMatrix(array);
17     paint.setColorFilter(new ColorMatrixColorFilter(cm));
18     canvas.drawBitmap(picture, picX, picY, paint);
19     picture.recycle();
20 }
```



## 2. 이미지 ► 이미지 활용

### 실습 9-2 미니 포토샵 앱 만들기

- 안드로이드 프로젝트 생성
  - (1) 프로젝트 이름 : Project9\_2
  - (2) 패키지 이름 : com.cookandroid.project9\_2
- 화면 디자인 및 편집
  - (1) 사용할 그림 파일들을 /res/drawable에 복사
  - (2) AndroidManifest.xml에 아이콘으로 사용할 그림 파일의 id로 변경 및 하드웨어 가속기 기능 끄

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



그림 9-6 미니 포토샵 앱 결과 화면



## 2. 이미지 ► 이미지 활용

---

### ■ 화면 디자인 및 편집

- (3) activity\_main.xml 수정
  - 바깥 리니어레이아웃 안에 2개의 리니어레이아웃 생성
  - 두 리니어레이아웃의 layout\_weight는 1:9 정도로 설정
  - 위쪽 리니어레이아웃에 이미지 버튼 6개 생성
  - 위젯의 id를 다음과 같이 선언
    - 리니어레이아웃 : iconLayout, pictureLayout
    - 이미지버튼 : ibZoomin, ibZoomout, ibRotate, ibBright, ibDark, ibGray

## 2. 이미지 ▶ 이미지 활용

예제 9-11 activity\_main.xml

```
1 <LinearLayout>
2     <LinearLayout
3         android:id="@+id/iconLayout"
4         android:layout_width="fill_parent"
5         android:layout_height="0dip"
6         android:layout_weight="1"
7         android:gravity="center"
8
9         <ImageButton
10             android:id="@+id/ibZoomin"
11             android:src="@drawable/zoom_in" />
12
13         ~~~~ 중간 생략(이미지버튼 5개) ~~~~
14     </LinearLayout>
15
16     <LinearLayout
17         android:id="@+id/pictureLayout"
18         android:layout_width="fill_parent"
19         android:layout_height="0dip"
20         android:layout_weight="9"
21         android:gravity="center" >
22     </LinearLayout>
23 </LinearLayout>
```



## 2. 이미지 ► 이미지 활용

### ■ Java 코드 작성 및 수정

#### ■ (1) MainActivity.java 코딩

- 이미지버튼에 대응할 6개 위젯 변수 선언
- MyGraphicView 클래스 변수를 선언
- MyGraphicView 정의 : 그림 파일을 중앙에 비트맵으로 출력
- pictureLayout을 인플레이트한 후 MyGraphicView를 추가

#### 예제 9-12 미니 포토샵의 Java 코드 1

```
1 public class MainActivity extends AppCompatActivity {  
2     ImageButton ibZoomin, ibZoomout, ibRotate, ibBright, ibDark, ibGray;  
3     MyGraphicView graphicView;  
4  
5     @Override  
6     public void onCreate(Bundle savedInstanceState) {  
7         super.onCreate(savedInstanceState);  
8         setContentView(R.layout.activity_main);  
9         setTitle("미니 포토샵");  
10    }
```

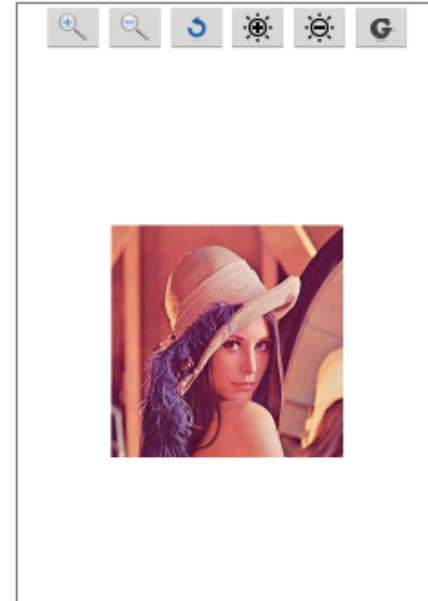
## 2. 이미지 ► 이미지 활용

---

```
11    LinearLayout pictureLayout = (LinearLayout) findViewById(R.id.pictureLayout);
12    graphicView = (MyGraphicView) new MyGraphicView(this);
13    pictureLayout.addView(graphicView);
14
15 }
16
17 private static class MyGraphicView extends View {
18     public MyGraphicView(Context context) {
19         super(context);
20     }
21     @Override
22     protected void onDraw(Canvas canvas) {
23         super.onDraw(canvas);
24
25         Bitmap picture = BitmapFactory.decodeResource(getResources(),
26                                     R.drawable.lena256);
27         int picX = (this.getWidth() - picture.getWidth()) / 2;
28         int picY = (this.getHeight() - picture.getHeight()) / 2;
```

## 2. 이미지 ► 이미지 활용

```
29     canvas.drawBitmap(picture, picX, picY, null);
30
31     picture.recycle();
32 }
33 }
34 }
```



## 2. 이미지 ► 이미지 활용

### ■ Java 코드 작성 및 수정

#### ■ (2) 확대 아이콘 코딩

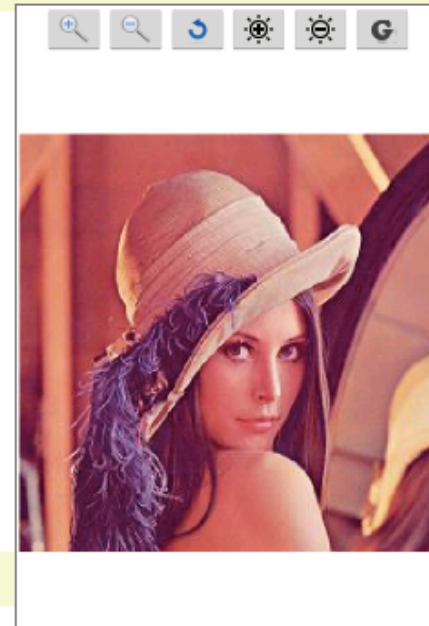
- 축척에 사용될 전역변수를 선언
- clickIcons( ) 메소드를 정의하고 확대 아이콘 클릭 리스너를 생성
- clickIcons( ) 메소드를 호출
- onDraw( )에 Canvas.scale( ) 메소드를 추가

#### 예제 9-13 미니 포토샷의 Java 코드 2

```
1  ~~~~ 중간 생략 ~~~~
2  static float scaleX=1, scaleY=1;
3
4  ~~~~ 중간 생략 ~~~~
5  private void clickIcons() {
6      ibZoomin = (ImageButton) findViewById(R.id.ibZoomin);
7      ibZoomin.setOnClickListener(new View.OnClickListener() {
8          public void onClick(View v) {
```

## 2. 이미지 ► 이미지 활용

```
9         scaleX = scaleX + 0.2f;
10        scaleY = scaleY + 0.2f;
11        graphicView.invalidate();
12    }
13    });
14
15 }
16 ~~~~ 중간 생략 ~~~~
17 clickIcons();
18
19 ~~~~ 중간 생략 ~~~~
20 int cenX = this.getWidth() / 2 ;
21 int cenY = this.getHeight() / 2;
22 canvas.scale(scaleX, scaleY, cenX, cenY);
23
```



## 2. 이미지 ► 이미지 활용

---

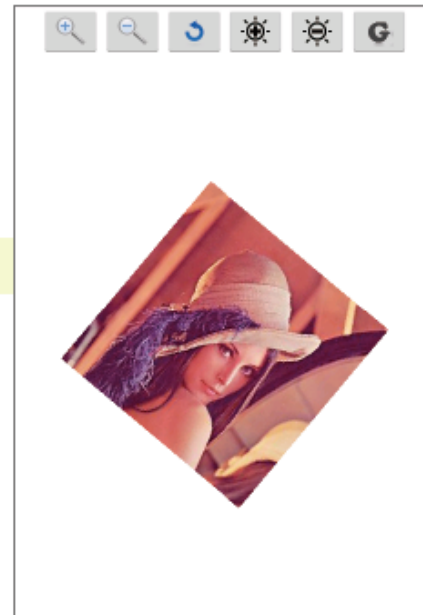
- Java 코드 작성 및 수정
  - (3) 회전 아이콘 코딩
    - 회전에 사용될 전역변수 선언
    - 회전 아이콘 클릭 리스너 생성
    - onDraw( )에 Canvas.rotate( ) 메소드를 추가



## 2. 이미지 ▶ 이미지 활용

예제 9-14 미니 포토샵의 Java 코드 3

```
1  ~~~~ 중간 생략 ~~~~
2  static float angle=0;
3
4  ~~~~ 중간 생략 ~~~~
5      ibRotate = (ImageButton) findViewById(R.id.ibRotate);
6      ibRotate.setOnClickListener(new View.OnClickListener() {
7          public void onClick(View v) {
8              angle = angle + 20;
9              graphicView.invalidate();
10         }
11     });
12
13 ~~~~ 중간 생략 ~~~~
14 canvas.rotate(angle, cenX, cenY);
```



## 2. 이미지 ▶ 이미지 활용

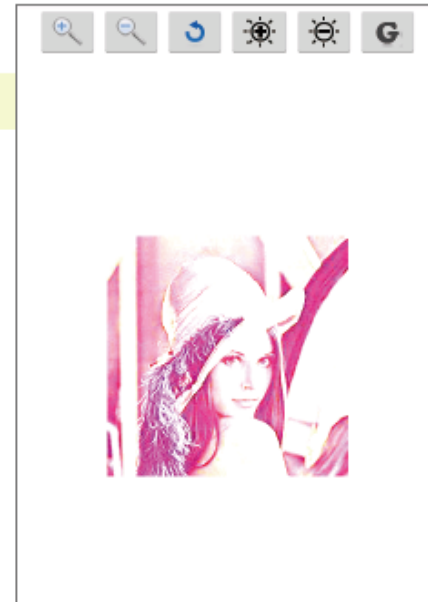
- Java 코드 작성 및 수정
  - (4) 밝게 하기 아이콘 코딩
    - 화면 밝기에 사용될 전역변수 선언
    - 밝게 하기 아이콘 클릭 리스너 생성
    - onDraw( )에 컬러매트릭스 적용

### 예제 9-15 미니 포토샵의 Java 코드 4

```
1  ~~~~ 중간 생략 ~~~~
2  static float color=1;
3
4  ~~~~ 중간 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     });
```

## 2. 이미지 ► 이미지 활용

```
13 ~~~~ 중간 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 ~~~~ 중간 생략 ~~~~
23         canvas.drawBitmap(picture, picX, picY, paint);
```



## 2. 이미지 ► 이미지 활용

---

- Java 코드 작성 및 수정
  - (5) 회색 영상으로 이미지를 변경하는 아이콘 코딩
    - 채도에 사용될 전역변수 선언
    - 회색 영상 아이콘 클릭 리스너 생성
    - `onDraw()`에 채도 설정 적용

## 2. 이미지 ▶ 이미지 활용

예제 9-16 미니 포토샵의 Java 코드 5

```
1  ~~~~ 중간 생략 ~~~~
2  static float satur=1;
3
4  ~~~~ 중간 생략 ~~~~
5      ibGray = (ImageButton) findViewById(R.id.ibGray);
6      ibGray.setOnClickListener(new View.OnClickListener() {
7          public void onClick(View v) {
8              if (satur == 0) satur = 1;
9              else satur = 0;
10             graphicView.invalidate();
11         }
12     });
13
14 ~~~~ 중간 생략 ~~~~
15 if (satur == 0) cm.setSaturation(satur);
```



## 2. 이미지 ▶ 이미지 활용

### ▶ 직접 풀어보기 9-3

[실습 9-2]를 다음과 같이 수정하라.

- 회색 영상 아이콘을 없앤다.
- 밝게 하기 아이콘을 클릭하면 채도가 높아지고, 어둡게 하기 아이콘을 클릭하면 채도가 낮아지도록 코드를 수정한다.
- 블러링, 엠보싱 아이콘을 추가하고, 클릭하면 블러링 또는 엠보싱 기능이 온/오프되게 한다.

