**[연습문제]**

1. 화면에 도형을 그릴 때 사용되는 클래스 중 ( 1 )는 도화지로, ( 2 )는 붓과 물감 개념과 비슷한다.

2. 다음 각각을 설명하는 클래스 및 메소드를 써라.

예) 색상을 설정 🡺 Paint.setColor()

(1) 선 두께를 설정 🡺 Paint.setSTrokeWidth()

(2) 폐합된 도형을 채울지 여부를 지정 🡺 Paint.setStyle()

(3) 원을 그림 🡺 Canvas.drawRect()

(4) 연속된 점을 연결해서 그림 🡺 Canvas.drawPath()

(5) 글자 크기를 지정 🡺 Paint.setTextSize()

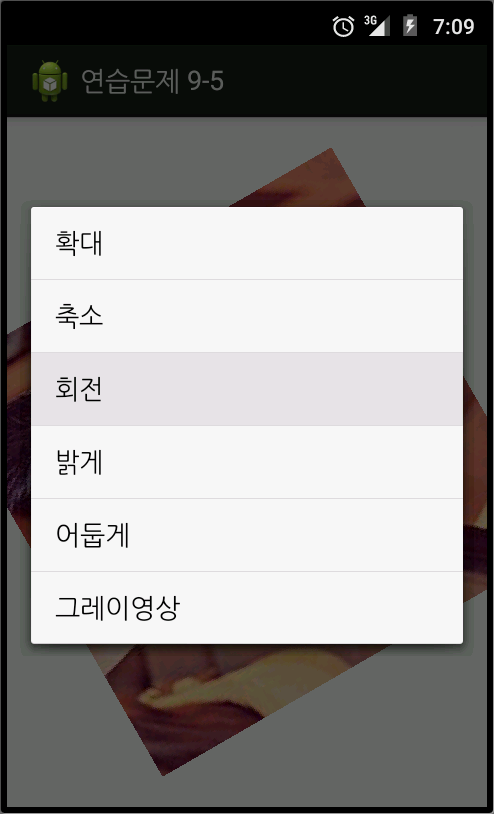
(6) 여러 개 점을 화면에 찍음 🡺 Canvas.drawPoints()

(7) 글자의 정렬방법을 지정 🡺 Paint.setTextAlign()

3. 리소스의 그림은 ( 1 ) 메소드를 이용하고, SD카드의 그림은 ( 2 ) 메소드를 이용해서 접근한다. 또 그림을 화면에 출력하는 것은 공통적으로 ( 3 ) 메소드를 사용한다. 비트맵 자원을 해제하기 위해서는 ( 4 ) 메소드를 사용한다..

4. 블러링 효과를 주기 위해서 ( 1 )클래스를, 엠보싱 효과를 주기 위해서는 ( 2 )를, 색상이나 밝기의 조절을 위해서는 ( 3 ), ( 4 )클래스를 사용한다. 특히, 칼라이미지를 그레이 이미지로 변환하려면 ( 5 ) 메소드를 사용한다.

5. <실습 9-2>에서 아이콘을 모두 없애고 컨텍스트 메뉴로 변경하라. 단, 별도의 메뉴 XML파일을 만들지 말고, Java 코드안에서 메뉴를 작성하라.



6. [직접 풀어보기 9-2]를 개선해서 이전에 그린 도형이 계속 화면에 남아 있도록 프로젝트를 작성하라.

힌트 : 도형 클래스를 정의하고, 동적 리스트로 도형 클래스가 추가되도록 한다.

(1) 도형 클래스의 예

private static class MyShape {

int shapeType; // 도형 종류

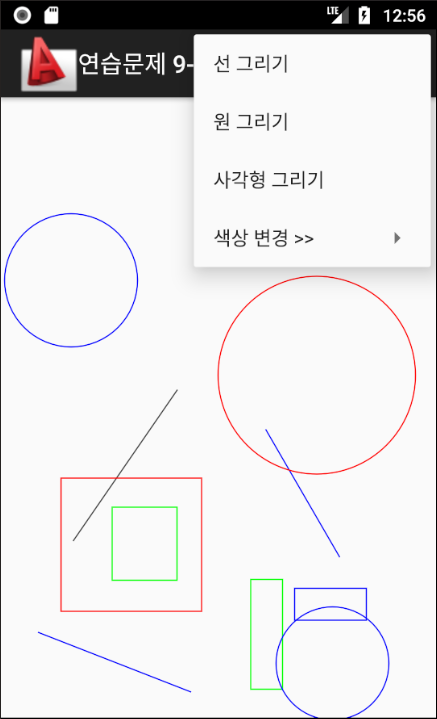
int startX, startY, stopX, stopY; // 도형의 2점

int color; // 도형 색상

}

(2) 동적 리스트 예

static List<MyShape> myshape = new ArrayList<MyShape>();



**[연습문제-답안]**

1. (1) Canvas (2) Paint

2. (1) Paint.setSTrokeWidth()

(2) Paint.setStyle()

(3) Canvas.drawRect()

(4) Canvas.drawPath()

(5) Paint.setTextSize()

(6) Canvas.drawPoints()

(7) Paint.setTextAlign()

3.

(1) BitmapFactory.decodeResource()

(2) BitmapFactory.decodeFile()

(3) Canvas.drawBitmap()

(4) Bitmap.recycle()

4.

(1) BlurMaskFilter

(2) EmbosMaskFilter

(3) ColorMatrix

(4) ColorMatrixColorFilter

(5) ColorMatrix.setSaturation()

5.

\*\*\* activity\_main.xml \*\*\*

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<LinearLayout

android:id="@+id/pictureLayout"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:gravity="center"

android:orientation="horizontal" >

</LinearLayout>

</LinearLayout>

\*\*\* Java 코드 \*\*\*

package com.cookandroid.exercise09\_5;

import android.app.Activity;

import android.content.Context;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.graphics.Canvas;

import android.graphics.ColorMatrix;

import android.graphics.ColorMatrixColorFilter;

import android.graphics.Paint;

import android.os.Bundle;

import android.view.ContextMenu;

import android.view.ContextMenu.ContextMenuInfo;

import android.view.MenuItem;

import android.view.View;

import android.widget.ImageButton;

import android.widget.LinearLayout;

public class MainActivity extends Activity {

ImageButton ibZoomin, ibZoomout, ibRotate, ibBright, ibDark, ibGray;

MyGraphicView graphicView;

static float scaleX = 1, scaleY = 1;

static float angle = 0;

static float color = 1;

static float satur = 1;

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

setTitle("연습문제 9-5");

LinearLayout pictureLayout = (LinearLayout) findViewById(R.id.pictureLayout);

graphicView = (MyGraphicView) new MyGraphicView(this);

pictureLayout.addView(graphicView);

registerForContextMenu(graphicView);

}

@Override

public void onCreateContextMenu(ContextMenu menu, View v,

ContextMenuInfo menuInfo) {

// TODO Auto-generated method stub

if (v == graphicView) {

menu.add(0, 1, 0, "확대");

menu.add(0, 2, 0, "축소");

menu.add(0, 3, 0, "회전");

menu.add(0, 4, 0, "밝게");

menu.add(0, 5, 0, "어둡게");

menu.add(0, 6, 0, "그레이영상");

}

}

@Override

public boolean onContextItemSelected(MenuItem item) {

// TODO Auto-generated method stub

switch (item.getItemId()) {

case 1:

scaleX = scaleX + 0.2f;

scaleY = scaleY + 0.2f;

graphicView.invalidate();

return true;

case 2:

scaleX = scaleX - 0.2f;

scaleY = scaleY - 0.2f;

graphicView.invalidate();

return true;

case 3:

angle = angle + 20;

graphicView.invalidate();

return true;

case 4:

color = color + 0.2f;

graphicView.invalidate();

return true;

case 5:

color = color - 0.2f;

graphicView.invalidate();

return true;

case 6:

if (satur == 0)

satur = 1;

else

satur = 0;

graphicView.invalidate();

return true;

}

return super.onContextItemSelected(item);

}

private static class MyGraphicView extends View {

public MyGraphicView(Context context) {

super(context);

}

@Override

protected void onDraw(Canvas canvas) {

super.onDraw(canvas);

int cenX = this.getWidth() / 2;

int cenY = this.getHeight() / 2;

canvas.scale(scaleX, scaleY, cenX, cenY);

canvas.rotate(angle, cenX, cenY);

Paint paint = new Paint();

float[] array = { color, 0, 0, 0, 0, 0, color, 0, 0, 0, 0, 0,

color, 0, 0, 0, 0, 0, 1, 0 };

ColorMatrix cm = new ColorMatrix(array);

if (satur == 0)

cm.setSaturation(satur);

paint.setColorFilter(new ColorMatrixColorFilter(cm));

Bitmap picture = BitmapFactory.decodeResource(getResources(),

R.drawable.lena256);

int picX = (this.getWidth() - picture.getWidth()) / 2;

int picY = (this.getHeight() - picture.getHeight()) / 2;

canvas.drawBitmap(picture, picX, picY, paint);

picture.recycle();

}

}

}

6.

package com.cookandroid.exercise09\_6;

import java.util.ArrayList;

import java.util.List;

import android.app.Activity;

import android.content.Context;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.graphics.Rect;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;

import android.view.MotionEvent;

import android.view.SubMenu;

import android.view.View;

public class MainActivity extends Activity {

final static int LINE = 1, CIRCLE = 2, RECTANGLE = 3;

static int curShape = LINE;

static int curColor = Color.DKGRAY;

// 도형을 저장할 리스트

static List<MyShape> myshape = new ArrayList<MyShape>();

static boolean isFinish = false; // ACTION\_UP 여부

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(new MyGraphicView(this));

setTitle("연습문제 9-6");

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// TODO Auto-generated method stub

super.onCreateOptionsMenu(menu);

menu.add(0, 1, 0, "선 그리기");

menu.add(0, 2, 0, "원 그리기");

menu.add(0, 3, 0, "사각형 그리기");

SubMenu sMenu = menu.addSubMenu("색상 변경 >>");

sMenu.add(0, 4, 0, "빨강");

sMenu.add(0, 5, 0, "초록");

sMenu.add(0, 6, 0, "파랑");

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// TODO Auto-generated method stub

switch (item.getItemId()) {

case 1:

curShape = LINE; // 선

return true;

case 2:

curShape = CIRCLE; // 원

return true;

case 3:

curShape = RECTANGLE; // 사각형

return true;

case 4:

curColor = Color.RED;

return true;

case 5:

curColor = Color.GREEN;

return true;

case 6:

curColor = Color.BLUE;

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();

isFinish = false;

break;

case MotionEvent.ACTION\_MOVE:

stopX = (int) event.getX();

stopY = (int) event.getY();

isFinish = false;

this.invalidate();

break;

case MotionEvent.ACTION\_UP:

// 마우스가 Up되면 최종 적으로 배열에 넣는다.

MyShape shape = new MyShape();

shape.shapeType = curShape;

shape.startX = startX;

shape.startY = startY;

shape.stopX = stopX;

shape.stopY = stopY;

shape.color = curColor;

myshape.add(shape);

isFinish = true;

this.invalidate();

break;

}

return true;

}

protected void onDraw(Canvas canvas) {

super.onDraw(canvas);

Paint paint = new Paint();

paint.setAntiAlias(true);

paint.setStrokeWidth(3);

paint.setStyle(Paint.Style.STROKE);

// 일단 배열의 도형을 모두 그린다.

for (int i = 0; i < myshape.size(); i++) {

MyShape shape = myshape.get(i);

paint.setColor(shape.color);

switch (shape.shapeType) {

case LINE:

canvas.drawLine(shape.startX, shape.startY, shape.stopX,

shape.stopY, paint);

break;

case CIRCLE:

int radius = (int) Math.sqrt(Math.pow(shape.stopX

- shape.startX, 2)

+ Math.pow(shape.stopY - shape.startY, 2));

canvas.drawCircle(shape.startX, shape.startY, radius, paint);

break;

case RECTANGLE:

Rect rect = new Rect(shape.startX, shape.startY,

shape.stopX, shape.stopY);

canvas.drawRect(rect, paint);

break;

}

}

// 그림이 아직 진행 중이면 (= 화면을 터치 중이면) 현재 진행중이 도형을 그린다.

if (isFinish == false) {

paint.setColor(curColor);

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;

case RECTANGLE:

Rect rect = new Rect(startX, startY, stopX, stopY);

canvas.drawRect(rect, paint);

break;

}

}

}

}

// 도형 클래스

private static class MyShape {

int shapeType; // 도형 종류

int startX, startY, stopX, stopY; // 도형의 2점

int color; // 도형 색상

}

}