그림판 이어그리기

public class MainActivity extends AppCompatActivity {

MyView myview;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

myview = new MyView(this);

setContentView(myview);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

menu.add(0,1,0,"선그리기");

menu.add(0,2,0,"원그리기");

menu.add(0,3,0,"글자 넣기");

menu.add(0,4,0,"네모 그리기");

menu.add(0,8,0,"연필");

SubMenu smenu = menu.addSubMenu("색상변경>>");

smenu.add(0,5,0,"빨강");

smenu.add(0,6,0,"초록");

smenu.add(0,7,0,"파랑");

return super.onCreateOptionsMenu(menu);

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

if(item.getItemId() == 1){

myview.shape = 1;

}

else if(item.getItemId() == 2){

myview.shape = 2;

}

else if(item.getItemId() == 3){

myview.shape = 3;

}

else if(item.getItemId() == 4){

myview.shape = 4;

}

else if(item.getItemId() == 5){

myview.color = 1;

}

else if(item.getItemId() == 6){

myview.color = 2;

}

else if(item.getItemId() == 7){

myview.color = 3;

}

else if(item.getItemId() == 8){

myview.shape = 8;

}

return super.onOptionsItemSelected(item);

}

}

public class MyView extends View {

int shape = 1; //1;선그리기, 2:원그리기

int color = 1;

int start\_x = -1, start\_y = -1, stop\_x = -1, stop\_y = -1;

ArrayList<Point> points = new ArrayList<Point>();

ArrayList<ShapeInfo> shapelist = new ArrayList<ShapeInfo>();

public MyView(Context context) {

super(context);

}

@Override

protected void onDraw(Canvas canvas) {

super.onDraw(canvas);

Paint paint = new Paint();

paint.setAntiAlias(true);

paint.setStrokeWidth(3);

if (color == 1) {

paint.setColor(Color.RED);

} else if (color == 2) {

paint.setColor(Color.GREEN);

} else if (color == 3) {

paint.setColor(Color.BLUE);

}

if (shape == 1) {

paint.setStrokeWidth(3);

canvas.drawLine(start\_x, start\_y, stop\_x, stop\_y, paint);

} else if (shape == 2) {

paint.setStrokeWidth(0);

int radius = (int) Math.sqrt(Math.pow(stop\_x - start\_x, 2) + Math.pow(stop\_y - start\_y, 2));

canvas.drawCircle(start\_x, start\_y, radius, paint);

} else if (shape == 3) {

paint.setStrokeWidth(0);

paint.setTextSize(30);

canvas.drawText("Hi!!", start\_x, start\_y, paint);

} else if (shape == 4) {

paint.setStyle(Paint.Style.STROKE);

Rect rect = new Rect(start\_x, start\_y, stop\_x, stop\_y);

canvas.drawRect(rect, paint);

} else if (shape == 8) {

paint.setStrokeWidth(3);

}

for (int i = 1; i < points.size(); i++) {

if (!points.get(i).isDraw) continue;

canvas.drawLine(points.get(i - 1).x, points.get(i - 1).y, points.get(i).x, points.get(i).y, paint);

}

for (int i = 1; i < shapelist.size(); i++) {

canvas.drawLine(shapelist.get(i).x1, shapelist.get(i).y1, shapelist.get(i).x2, shapelist.get(i).y2, paint);

}

}

@Override

public boolean onTouchEvent(MotionEvent event) {

switch (event.getAction()){

case MotionEvent.ACTION\_UP:

if(shape == 1){

shapelist.add(new ShapeInfo(start\_x, start\_y, stop\_x, stop\_y, 1));

}

case MotionEvent.ACTION\_MOVE:

if(shape == 8) {

points.add(new Point(event.getX(), event.getY(), true));

}

stop\_x = (int)event.getX();

stop\_y = (int)event.getY();

this.invalidate();

break;

case MotionEvent.ACTION\_DOWN:

start\_x = (int)event.getX();

start\_y = (int)event.getY();

if(shape == 8) {

points.add(new Point(start\_x, start\_y, false));

}

invalidate();

break;

}

return true;

}

}

class Point {

float x;

float y;

boolean isDraw;

public Point(float x, float y, boolean isDraw) {

this.x = x;

this.y = y;

this.isDraw = isDraw;

}

}

class ShapeInfo {

float x1 = 0, x2 = 0, y1 = 0, y2 = 0;

int shapeType = 1; //1 : line

public ShapeInfo(float x1, float y1,float x2, float y2, int shapeType) {

this.x1 = x1;

this.y1 = y1;

this.x2 = x2;

this.y2 = y2;

this.shapeType = shapeType;

}

}

