Chap09. 그래픽과 이미지

01 그래픽

02 이미지

01. 그래픽

◆ 그래픽

- 화면에 점, 선, 원, 사각형 등의 도형을 그리는 방식
  - 좌표 직접 입력(그래픽 처리 기본)
  - 화면을 손가락으로 터치(터치 이벤트)
- 그래픽 처리 방식 잘 활용
  - 그림판 같은 앱

01. 그래픽

◆ 캔버스(Canvas)와 페인트(Paint)

- 화면에 도형을 그릴 때 사용되는 클래스

Canvas 클래스 **도화지**		•점, 선, 원, 사각형 그리기 •텍스트 쓰기 •이미지 출력
Paint 클래스 **붓**		•색상 선택 •스타일 선택 •펜 두께 선택 •글꼴 선택

01. 그래픽

◆ android.graphics.Canvas 클래스

- 점 찍는 메소드 원형

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

◆ android.graphics.Paint 클래스

- 색상 지정 메소드 원형

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

01. 그래픽

◆ android.graphics.Canvas 클래스

void drawPoint(float x, float y, Paint paint) : 설정한 좌표에 점을 출력한다.

void drawLine(float startX, float startY, float stopX, float stopY, Paint paint) : 설정한 두 좌표를 연결하는 선분을 출력한다.

void drawCircle(float cx, float cy, float radius, Paint paint) : (cx, cy)를 중심으로 하고 radius 를 반지름으로 하는 원을 출력한다.

void drawRect(float left, float top, float right, float bottom, Paint paint)
void drawRect(Rect r, Paint paint)
void drawRect(RectF rect, Paint paint) : 지정한 직사각형을 출력한다.

void drawText(String text, float x, float y, Paint paint) : (x, y)에 text를 출력한다.

void drawRoundRect(RectF rect, float rx, float ry, Paint paint) : rect의 모서리를 rx, ry의 크기를 갖는 타원의 모양으로 만들어 출력한다.

void drawOval(RectF oval, Paint paint) : oval 에 내접하는 타원을 출력한다.

void drawArc(RectF oval, float startAngle, float sweepAngle, boolean useCenter, Paint paint) : oval 내에서 startAngle의 각도로부터 출발하여 sweepAngle 만큼 휩쓴 원호를 출력한다. useCenter 사용시 도형의 중심과 이어지며, 사용하지 않으면 startAngle의 위치와 sweepAngle을 지난 끝지점이 연결된다.

void drawLines(float[] pts, Paint paint) : 배열에 저장된 선을 순서대로 그린다. 좌표 4개씩 한 선분을 구성한다.

void drawPoints(float[] pts, int offset, int count, Paint paint) : 배열에 저장된 좌표중 offset부터 시작하여 count만큼의 점들을 찍는다.

01. 그래픽

◆ android.graphics.Paint 클래스

void setColor(int Color)
void setARGB(int a, int r, int g, int b) : 그리기에 사용할 색상을 지정한다.

void setAntiAlias(boolean aa) : 안티안리어싱 사용을 설정한다.
- Paint paint = new Paint(Paint.ANTI_ALIAS_FLAG); 와 같이 생성자에서 설정할 수 도 있다.

int getColor() : 지정되어있는 색상을 16진수로 받아들인다.

int getAlpha() : 지정되어있는 알파값을 받아들인다.

void setStrokeWidth(float width) : 선의 굵기를 width로 지정한다.

01. 그래픽

◆ android.graphics.Paint 클래스

void setStrokeCap(Paint.Cap cap) : 선의 끝 모양을 지정한다.
- BUTT : 지정한 좌표에서 선이 끝난다. (Default)
- ROUND : 둥근 모양으로 끝이 장식된다.
- SQUARE : 사각형 모양이되 지정한 좌표보다 조금 더 그어진다.

void setStrokeJoin(Paint.Join join) : 선이 만나 각지는 곳의 모양을 어떻게 처리할지를 지정한다.
- MITER : 90도로 각진 형태를 그린다. (Default)
- BEVEL : 깎은 모양으로 그린다.
- Round : 둥근 모양으로 그린다.

void setStrokeMiter(float miter) : MITER 조인 적용시 어느 각도 이상을 뾰족한 것으로 인식할 것인가를 지정한다.

void setStyle(Paint.Style style) : 외곽선과 내부의 어느 쪽을 그릴 것인가를 지정한다.
- FILL : 채우기O 외곽선X
- FILL_AND_STROKE _ 채우기O 외곽선O
- STROKE : 채우기X 외곽선O

01. 그래픽

◆ 그래픽을 표현할 때 View 클래스 재정의

- 자동 완성되거나 고정된 내용(굵게 표시된 부분만 변경)

```
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
    }
}
```

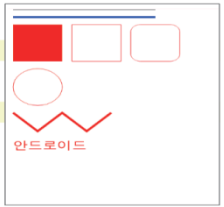
01. 그래픽

◆ 그래픽 기본의 Java 코드(p.356~357 예제 9-1)

- 그래픽 출력(View.onDraw( ) 메소드 오버라이딩)

예제 9-1 그래픽 기본의 Java 코드

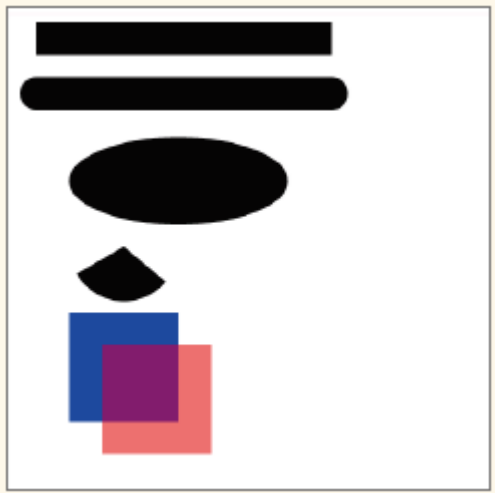
```
1   public void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(new MyGraphicView(this));
4   }
5
6   private static class MyGraphicView extends View {
7     public MyGraphicView(Context context) {
8       super(context);
9     }
10
11    @Override
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);
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);
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  }
```



직접 풀어보기 9-1(p.358)

◆ 그림과 같은 화면을 출력하도록 다음 메소드를 사용하여 Java를 코딩하여라.

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



01. 그래픽

◆ 터치 이벤트

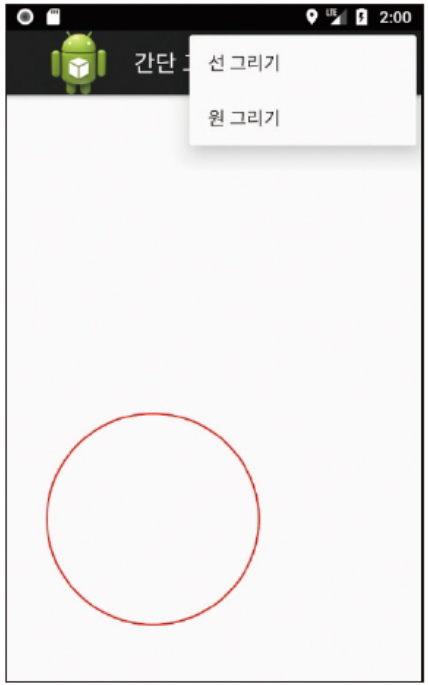
- onTouchEvent( ) 메소드 오버라이딩

```
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;
}
```

간단 그림판 앱 만들기(실습 9-1)

◆ 안드로이드 프로젝트 생성

- [01] 프로젝트 이름(Project9_1)
- 패키지 이름(com.cookandroid.project9_1)
- [실습 2-4]의 1~4(p.87~89)



간단 그림판 앱 만들기(실습 9-1)

2. 화면 및 디자인 편집

- [02] Java 코드로만 작성(activity_main.xml 삭제해도 무방)

간단 그림판 앱 만들기(실습 9-1)

3. Java 코드 작성 및 수정(MainActivity.java)

- [03] [MainActivity.java] 코딩(p.360 예제 9-2)
  - View 클래스의 상속을 받는 MyGraphicView 클래스 생성

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 }
```

간단 그림판 앱 만들기(실습 9-1)

3. Java 코드 작성 및 수정(MainActivity.java)

- [04] 옵션 메뉴 작성(p.361 예제 9-3)
  - 선 그리기, 원 그리기 옵션 메뉴 생성
  - 항목 클릭(curShape 변수에 선택한 전역상수 대입)
  - onCreateOptionsMenu( )와 onOptionsItemSelected( ) 자동 완성
    - 메뉴 → [Code] → [Override Methods]

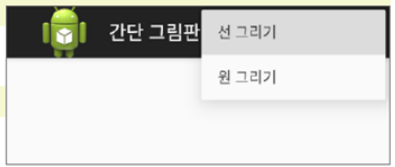
간단 그림판 앱 만들기(실습 9-1)

3. Java 코드 작성 및 수정(MainActivity.java)

- [04] 옵션 메뉴 작성(p.361 예제 9-3)

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 }
```



간단 그림판 앱 만들기(실습 9-1)

3. Java 코드 작성 및 수정(MainActivity.java)

- [05] 터치와 관련된 메소드 완성(p.362 예제 9-4)
  - MyGraphicView 클래스
    - MyGraphicView의 전역변수 시작x, 시작y, 끝x, 끝y 및 반지름 변수 선언
    - onTouchEvent( ) 메소드 자동 완성 후 자동 완성 외의 코드 완성
      - 메뉴 → [Code] → [Override Methods]

간단 그림판 앱 만들기(실습 9-1)

3. Java 코드 작성 및 수정(MainActivity.java)

- [05] 터치와 관련된 메소드 완성(p.362 예제 9-4)

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;
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    }
```

간단 그림판 앱 만들기(실습 9-1)

3. Java 코드 작성 및 수정(MainActivity.java)

- [06] 도형이 그려질 onDraw() 메소드 완성(p.363 예제 9-5)
  - MyGraphicView 내부에 onDraw( ) 자동 완성하고 나머지 코딩
  - 페인트에 선의 두께, 채우기 여부, 선의 색상 지정
  - switch( )~case문으로 메뉴 선택(선 또는 원)

간단 그림판 앱 만들기(실습 9-1)

3. Java 코드 작성 및 수정(MainActivity.java)

- [06] 도형이 그려질 onDraw() 메소드 완성(p.363 예제 9-5)

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  }
```

간단 그림판 앱 만들기(실습 9-1)

4. 프로젝트 실행 및 결과 확인

- [07] 저장 및 실행
  - 저장 – 메뉴 → [File] → [Save All] 클릭
  - 실행 – 메뉴 → [Run] → [Run ‘app’] 클릭

5. 안드로이드 애플리케이션 개발 완료

- [08] 결과 확인 및 초기화면으로 돌아감

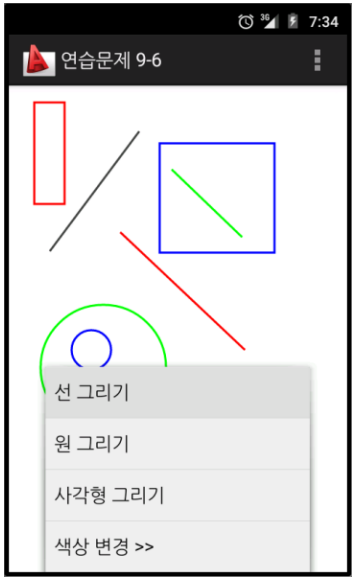
직접 풀어보기 9-2(p.364)

◆ [실습 9-1]을 다음과 같이 수정하여라.

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

과제9-1 연습문제6(p.387~388)

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



과제9-1 연습문제6(p.387~388)

◆ 도형 클래스의 예

```
private static class MyShape {
                int shapeType; // 도형 종류
                int startX, startY, stopX, stopY; // 도형의 2점
                int color; // 도형 색상 }
```

◆ 동적 리스트 예

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

과제9-1 연습문제6(p.387~388)

◆ 추가 사항

```
// 도형을 저장할 리스트
static List<MyShape> myshape = new ArrayList<MyShape>();
static boolean isFinish = false; // ACTION_UP 여부

// 도형 클래스
private static class MyShape {
    int shapeType; // 도형 종류
    int startX, startY, stopX, stopY; // 도형의 두점
    int color; // 도형 색상
}
```

과제9-1 연습문제6(p.387~388)

◆ 변경 사항

```
@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;
}
```

과제9-1 연습문제6(p.387~388)

```
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;
        }
    }
}
```