08장. 그래픽과 애니메이션

01. 화면에 뭔가를 그리려면
02. 텍스트 그리기
03. 비트맵 다루기
04. 도형 다루기
05. 애니메이션 다루기

컴퓨터공학과 변영철 교수

Canvas와 Paint

- 화면에 뭔가를 그리려면 유효한 Canvas가 있어야 하며, 보통은 View로부터 상속받는 뷰의 onDraw 메소드를 구현하면 파라미터로 얻을 수 있음
- 새로 작성한 뷰 클래스 객체를 Activity에 설정함

```
public class MyActivity extends Activity
{
	@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(new MyView(this));
	}

	class MyView extends View {
		public MyView(Context context) {
		    super(context);
		}

		@Override
		protected void onDraw(Canvas canvas) {
		    canvas.drawColor(Color.BLUE);
		}
	}
}
```

Canvas와 Paint

- Canvas
  - 무언가를 그리는 공간
  - getHeight 메소드와 getWidth 메소드로 Canvas를 담는 컨테이너 뷰 크기를 알 수 있음
  - Canvas 객체는 Paint 객체를 이용하여 원 등을 그릴 수 있음 (drawCircle)
- Paint
  - 물감에 해당되며, 스타일과 색, 그리고 렌더링 정보 등을 표현
- Paint 안티앨리어싱(anti-aliasing)
  - 그래픽(도형이든, 글꼴이든)이 화면에 좀 더 매끄럽게 나타나도록 하는 기술
  - anti-aliasing Paint객체를 생성하는 예제

```
Paint aliasedPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
```

Canvas와 Paint

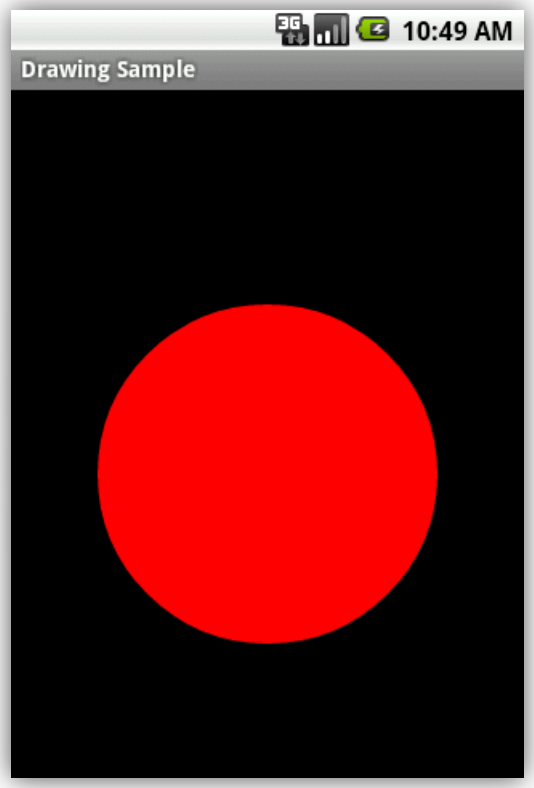
- Paint 스타일
  - Drawable 내부를 색으로 채우는 방식을 설정
  - STROKE 스타일 : 외곽선만 나타나고 색을 채우지 않음(기본)

```
Paint linePaint = new Paint();
linePaint.setStyle(Paint.Style.STROKE);
```

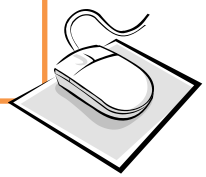
- Paint 그래디언트
  - 여러 색들이 매끄럽게 변화하면서 표현
  - 그래디언트 표현을 위한 클래스들 : android.graphics.Shader 에서 파생된 LinearGradient, RadialGradient, SweepGradient
  - 그래디언트들은 색상의 변화가 "흘러가는" 방향에서 차이
  - 모든 그래디언트는 적어도 두 개의 색(시작 색과 끝 색)을 요구
  - Paint 객체의 setShader 메소드 호출하여 설정

도형 그리기

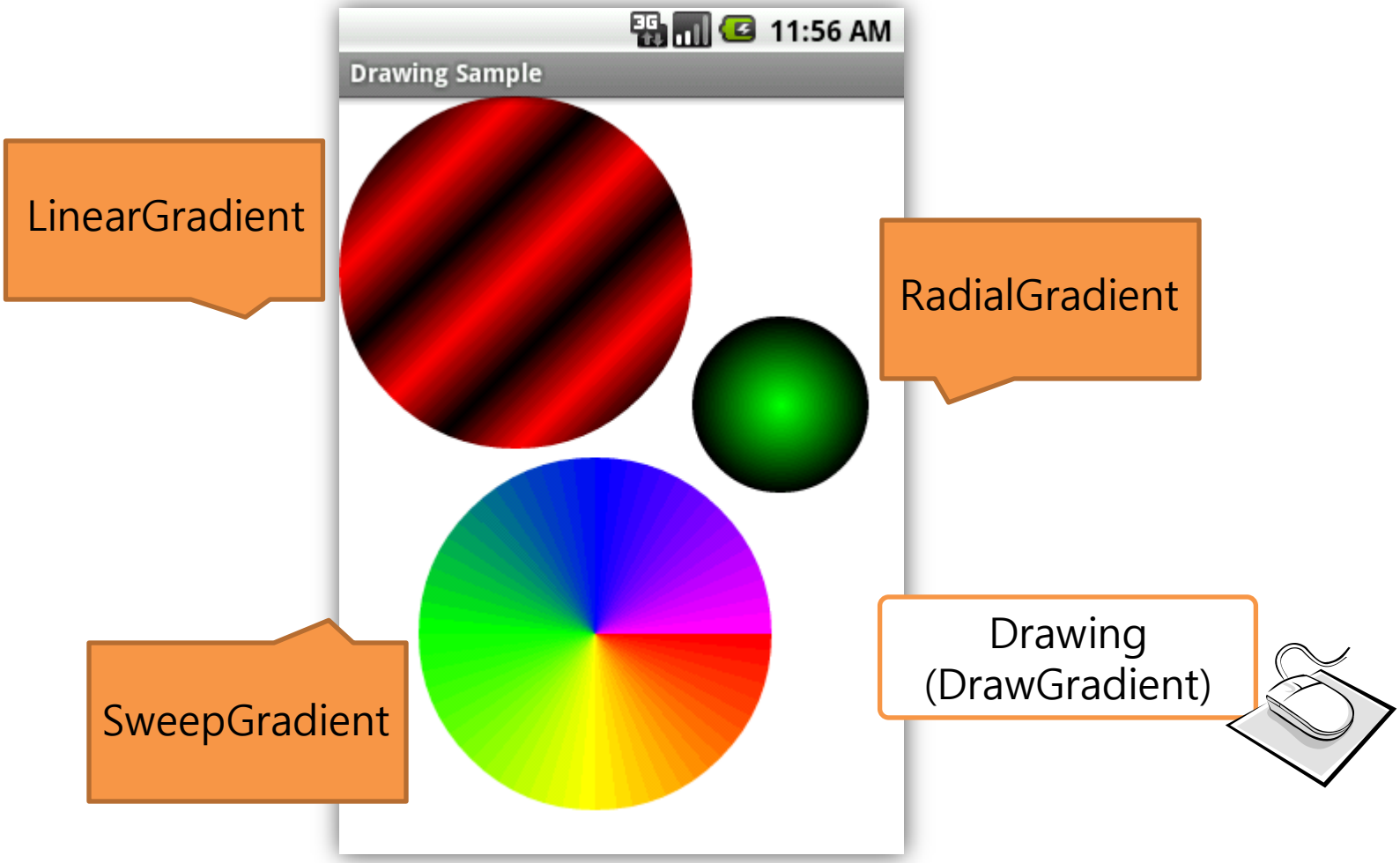
```
public class ShapeActivity extends OptionMenuActivity
{
	@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
			setContentView(new MyView(this));
	}

	private class MyView extends View
	{
		public MyView(Context context) {
			super(context);
		}

		@Override
		protected void onDraw(Canvas canvas) {
			canvas.drawColor(Color.BLACK);

			Paint p = new Paint(Paint.ANTI_ALIAS_FLAG);
			p.setColor(Color.RED);

			canvas.drawCircle(canvas.getWidth()/2, canvas.getHeight()/2, canvas.getWidth()/3, p);
		}
	}
}
```



Drawing



그래디언트


11:56 AM
Drawing Sample
LinearGradient
RadialGradient
SweepGradient
Drawing
(DrawGradient)

폰트와 스타일 적용

- 안드로이드 기본 폰트는 산스세리프(sans serif)
- 다른 폰트를 사용하고자 한다면 Typeface 인스턴스를 생성
- 고정폭(monospace) 폰트의 Typeface를 생성하는 예

```
Typeface tf= Typeface.create(Typeface.MONOSPACE, Typeface.NORMAL);
mPaint.setTypeface(tf); canvas.drawText(...);
```

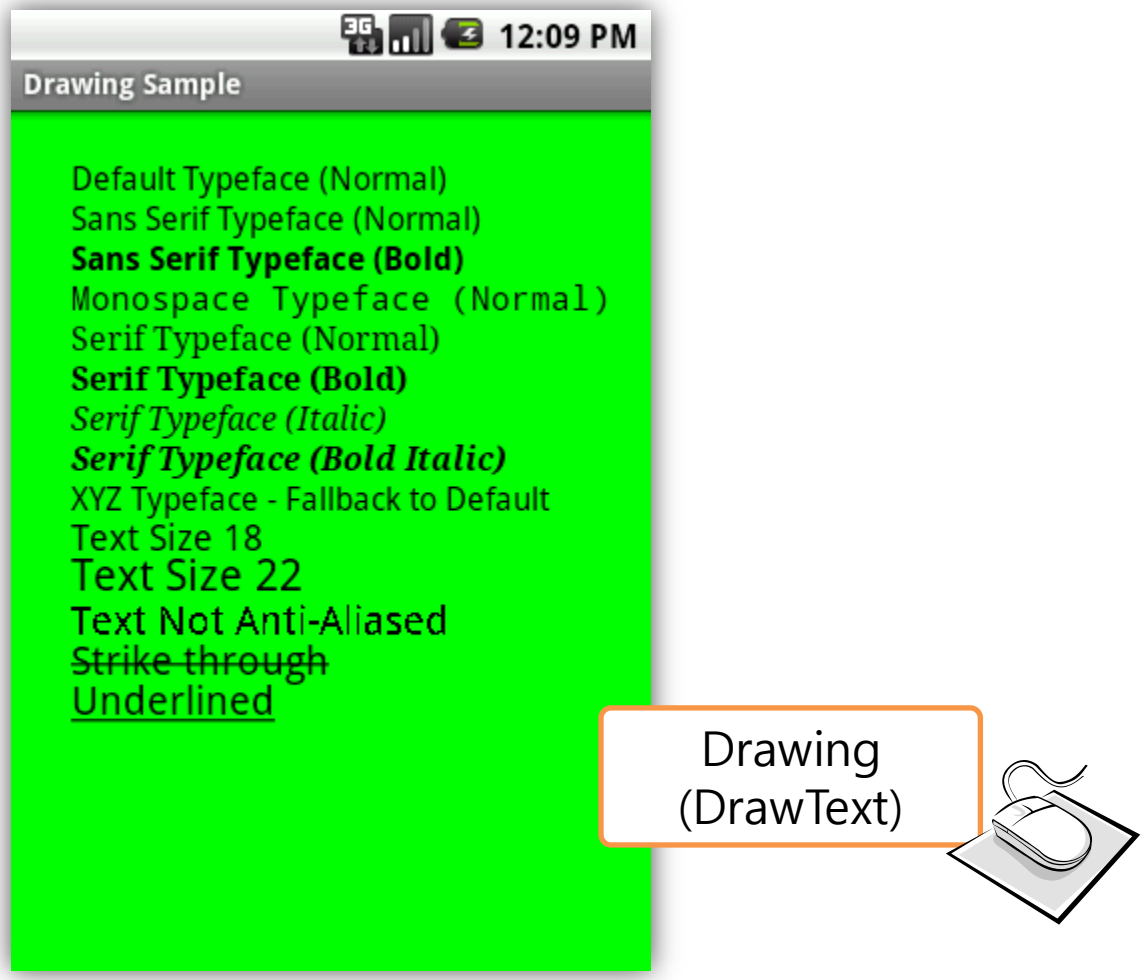
- Typeface 생성 시 서체의 스타일 여부도 설정할 수 있음

```
Typeface tf = Typeface.create(Typeface.SERIF, Typeface.ITALIC);
mPaint.setTypeface(tf); canvas.drawText(...);
```

- 안티앨리어싱, 밑줄, 취소선 등의 기타 속성들은 Paint 객체의 setFlags 메소드로 설정

```
mPaint.setFlags(Paint.UNDERLINE_TEXT_FLAG)
```

폰트와 스타일 적용

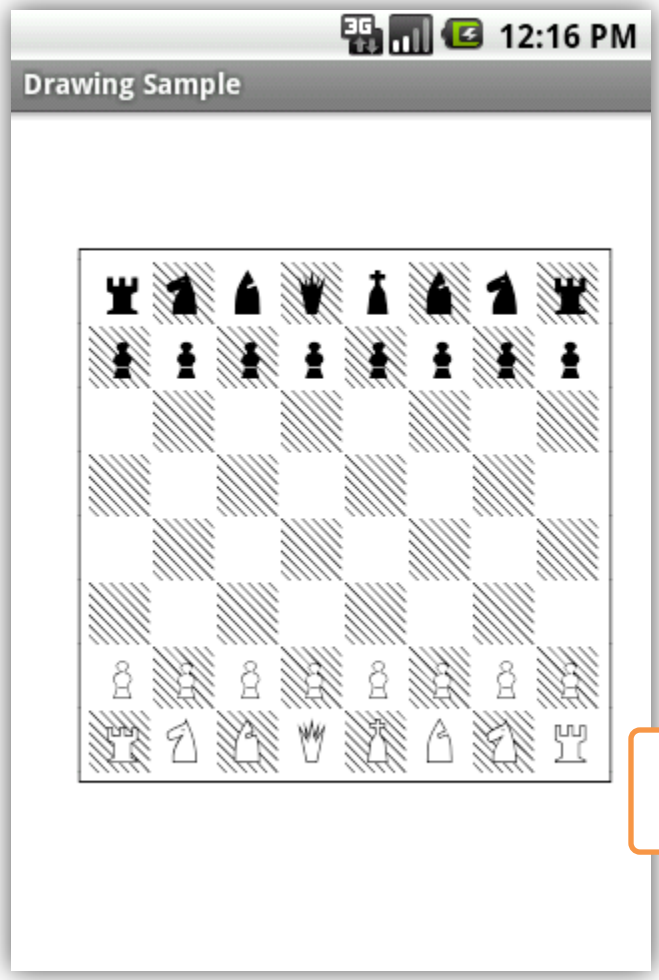

12:09 PM
Drawing Sample
Default Typeface (Normal)
Sans Serif Typeface (Normal)
Sans Serif Typeface (Bold)
Monospace Typeface (Normal)
Serif Typeface (Normal)
Serif Typeface (Bold)
Serif Typeface (Italic)
Serif Typeface (Bold Italic)
XYZ Typeface - Fallback to Default
Text Size 18
Text Size 22
Text Not Anti-Aliased
Strike-through
Underlined
Drawing
(DrawText)

커스텀 폰트

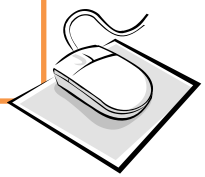
- 기본 폰트 이외의 폰트를 사용하고자 하면 원하는 폰트 파일(.ttf)을 리소스에 넣은 후 실행 시점에서 불러와서 적용
- 커스텀 폰트의 예
  - Chess Utrecht라는 체스판과 여러 체스말들의 "기호"를 담은 공개 체스 폰트
  - http://www.chessvariants.com/d.font/utrecht.html
  - 내려 받은 chess1.ttf 파일을 /assets/fonts/ 폴더에 넣음
- 사용 방법
  - createFromAsset 메소드로 Chess Utrecht 폰트가 적용된 Typeface 객체를 생성

```
Paint mPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
Typeface mType = Typeface.createFromAsset(getContext().getAssets(), "fonts/chess1.ttf");
```

커스텀 폰트



Drawing



비트맵 이미지

- 비트맵 그리기

```
Bitmap bmp= BitmapFactory.decodeResource(getResources(), R.drawable.bluejay);
canvas.drawBitmap(bmp, 0, 0, null);
```

- 확대·축소

```
Bitmap sm = Bitmap.createScaledBitmap(pic, 50, 75, false);
```

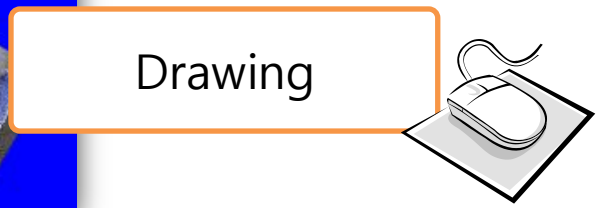
- 변환
  - Matrix 클래스로 Bitmap을 다양한 방식으로 변환 가능
  - 회전, 좌우 대칭, 상하 대칭 등
  - 반사(mirror) 변환 행렬을 적용해서, 좌우가 뒤집힌 Bitmap 객체를 생성하는 예

```
Matrix mirrorMatrix = new Matrix(); mirrorMatrix.preScale(-1, 1);
Bitmap mirrorPic =
    Bitmap.createBitmap(pic, 0, 0, pic.getWidth(), pic.getHeight(), mirrorMatrix, false);
```

비트맵 이미지

- 여러 변환(반사, 30도 회전)을 적용한 예

```
Matrix m = new Matrix();
m.preRotate(30);
m.preScale(-1, 1);
```



Drawable 자원 정의하기

- /res/drawable/green_rect.xml 자원 정의

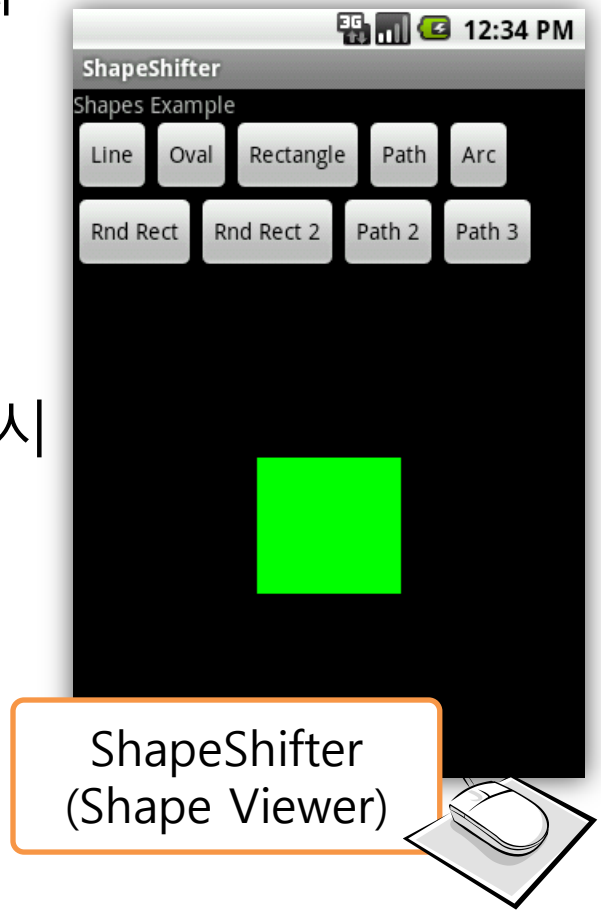
```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android=
    "http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid android:color="#0f0"/>
</shape>
```

- 자원을 적재하여 ImageView 위젯에 표시

```
ImageView iView = (ImageView)findViewById(R.id.ImageView1);
iView.setImageResource(R.drawable.green_rect);
```

- 코드에서 직접 자원 정의하는 예

```
ShapeDrawable rect = new ShapeDrawable(new RectShape());
rect.getPaint().setColor(Color.GREEN);
```



여러 가지 도형들

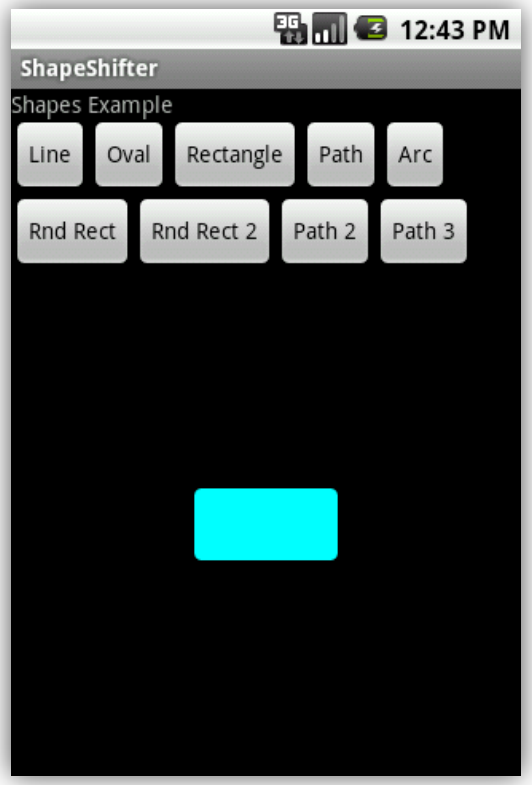
- android.graphics.drawable.shapes 패키지 제공 도형
  - 직사각형(정사각형)
  - 모서리가 둥근 직사각형
  - 타원(원)
  - 원호와 직선들
  - 그 외의 경로 기반 무정형 도형들

  -> 레이아웃의 있는 ImageView 위젯 안에 표시(가령 ImageView의 setImageDrawable 메소드 호출)할 수도 있고, Canvas의 여러 메소드들을 이용해서 화면에 직접 그릴 수도 있음

직사각형

- 너비가 100픽셀, 높이가 2픽셀, 수평 직선처럼 보이는 자홍색 직사각형을 표시

```
ShapeDrawable rect = new ShapeDrawable(new RectShape());
rect.setIntrinsicHeight(2);
rect.setIntrinsicWidth(100);
rect.getPaint().setColor(Color.MAGENTA);

ImageView iView = (ImageView)findViewById(R.id.ImageView1);
iView.setImageDrawable(rect);
```



- 둥근 모서리 지정
  - ShapeDrawable 객체 생성 시 RoundRectShape 객체 지정

```
ShapeDrawable rndrect =
   new ShapeDrawable(new RoundRectShape( new float[] { 5, 5, 5, 5, 5, 5, 5, 5 },
```

타원과 원

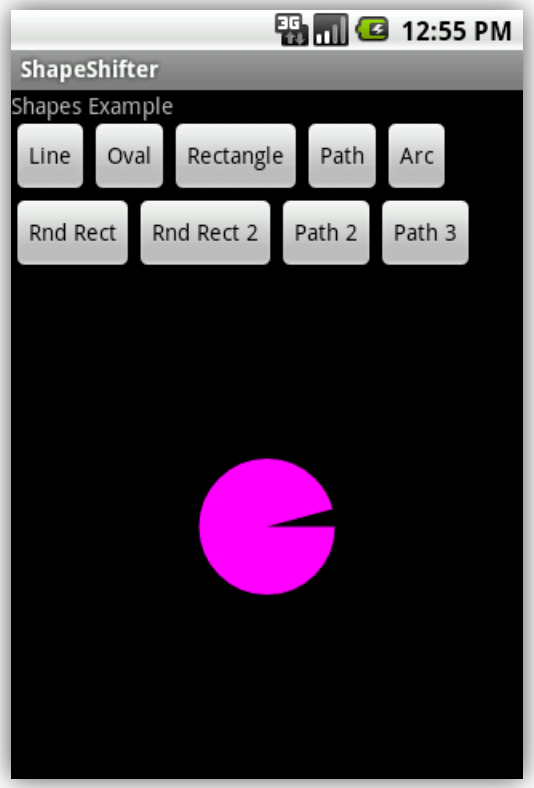
- 너비가 100픽셀이고 높이가 40픽셀인, 플라스틱 원반처럼 보이는 빨간 타원

```
ShapeDrawable oval = new ShapeDrawable(new OvalShape());
oval.setIntrinsicHeight(40);
oval.setIntrinsicWidth(100);
oval.getPaint().setColor(Color.RED);

ImageView iView = (ImageView)findViewById(R.id.ImageView1);
iView.setImageDrawable(oval);
```

원호

- 그래프 같은 원호(arc) 도형은 ArcShape 객체를 인수로 주어서 생성한 ShapeDrawable

```
ShapeDrawable pacMan = new ShapeDrawable(new ArcShape(0, 345));
pacMan.setIntrinsicHeight(100);
pacMan.setIntrinsicWidth(100);
pacMan.getPaint().setColor(Color.MAGENTA);

ImageView iView = (ImageView)findViewById(R.id.ImageView1);
iView.setImageDrawable(pacMan);
```



경로 도형

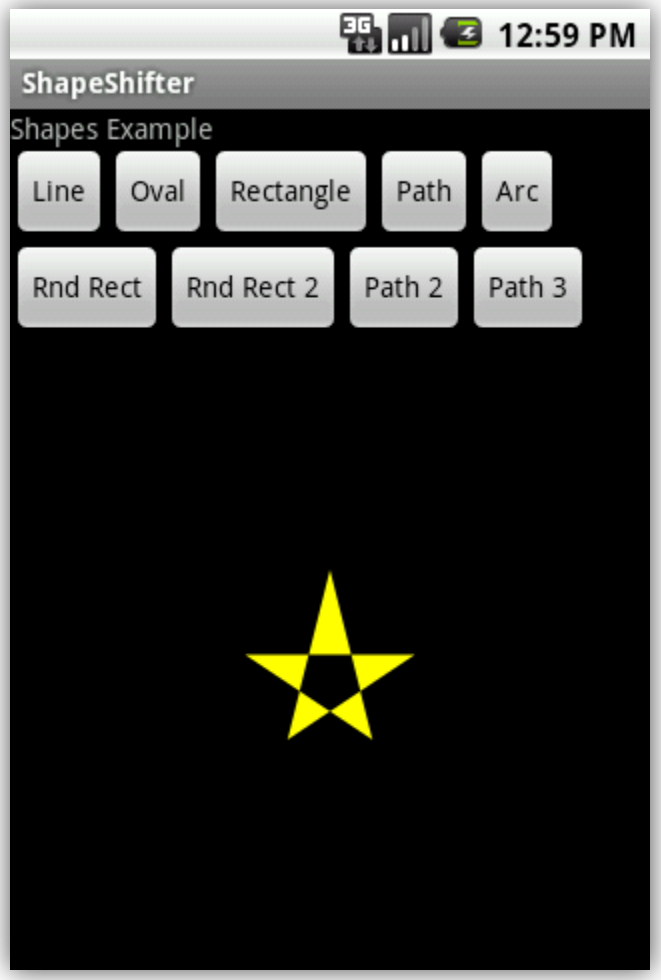
- 일련의 직선, 곡선 선분들로 도형의 외곽선을 직접 정의하면 자유 도형을 만들 수 있음

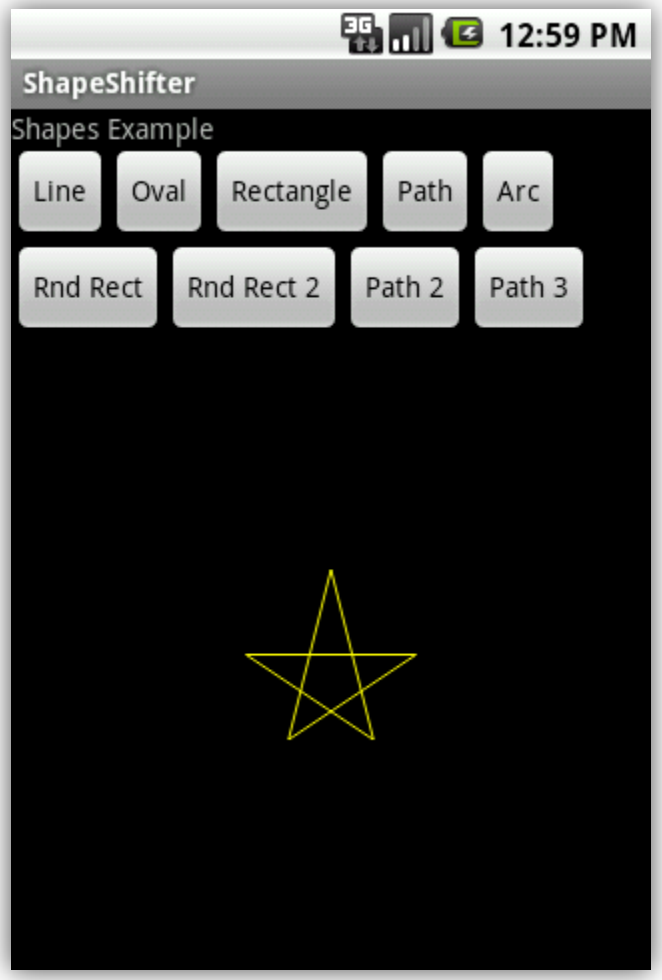
```
Path p = new Path();
p.moveTo(50, 0);
p.lineTo(25,100);
p.lineTo(100,50);
p.lineTo(0,50);
p.lineTo(75,100);
p.lineTo(50,0);
```

- 사용하려면 Path 객체를 담은 PathShape 객체를 생성하고 그것으로 ShapeDrawable 객체를 생성해야 함

```
ShapeDrawable star = new ShapeDrawable(new PathShape(p, 100, 100));
star.setIntrinsicHeight(100);
star.setIntrinsicWidth(100);
star.getPaint().setColor(Color.YELLOW);
star.getPaint().setStyle(Paint.Style.STROKE);
```

경로 도형


12:59 PM
ShapeShifter
Shapes Example
Line
Oval
Rectangle
Path
Arc
Rnd Rect
Rnd Rect 2
Path 2
Path 3


12:59 PM
ShapeShifter
Shapes Example
Line
Oval
Rectangle
Path
Arc
Rnd Rect
Rnd Rect 2
Path 2
Path 3

프레임 애니메이션

- 조금씩 다른 이미지들을 차례로 빠르게 화면에 표시함으로써 사물이 움직이는 듯한 환상을 만들어 내는 것
- 애니메이션 방법

  [1] ImageView 노드를 갖는 레이아웃 작성 및 설정

  [2] 일련의 비트맵 자원 로드

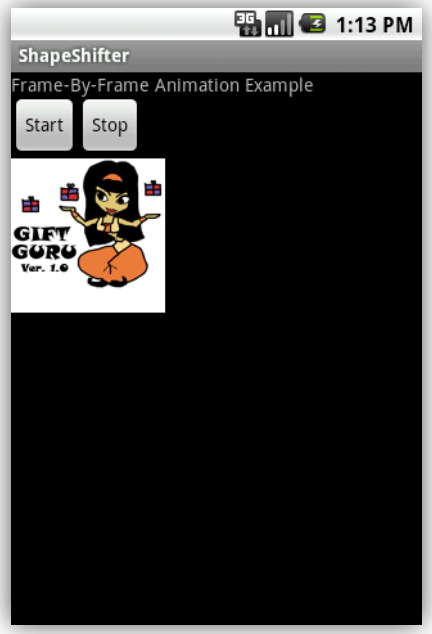
  [3] AnimationDrawable 객체(anim) 정의 및 프레임(비트맵) 추가

  [4] anim drawable 객체를 ImageView 객체의 배경으로 설정(setBackgroundDrawable)

  [5] 애니메이션 시작

프레임 애니메이션

```
ImageView imageview = (ImageView)findViewById(R.id.ImageView_Juggle);

//일련의 이미지 생성
BitmapDrawable bmp1 = BitmapDrawable)getResources().getDrawable(R.drawable.splash1);
BitmapDrawable bmp2 = BitmapDrawable)getResources().getDrawable(R.drawable.splash2);
BitmapDrawable bmp3 = BitmapDrawable)getResources().getDrawable(R.drawable.splash3);

int duration = 250;

//애니메이션 drawable 생성 및 프레임 설정
anim = new AnimationDrawable();
anim.setOneShot(false); // loop continuously
anim.addFrame(bmp1, duration);
anim.addFrame(bmp2, duration);
anim.addFrame(bmp3, duration);

//anim drawable 객체를 뷰 위젯의 배경으로 설정
imageview.setBackgroundDrawable(anim);

//애니메이션 시작
anim.setVisible(true,true);
anim.start();
```



트위닝 애니메이션

- 트위닝 애니메이션 정의
  - 시작과 끝을 지정하면 사이(between) 애니메이션이 자동으로 수행됨
  - /res/anim/ 폴더에 XML 자원으로 정의
  - 간단한 5초짜리 회전 애니메이션의 예 (Spin)

```
<?xml version="1.0" encoding="utf-8" ?>
<set xmlns:android = "http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <rotate
        android:fromDegrees="0"
        android:toDegrees="360"
        android:pivotX="50%”
        android:pivotY="50%"
        android:duration="5000" />
</set>
```

트위닝 애니메이션

- 애니메이션 불러오기
  - /res/anim/grow.xml에서 불러온 애니메이션을 ImageView에 적용하는 예

```
ImageView iView = (ImageView)findViewById(R.id.ImageView1);
iView.setImageResource(R.drawable.green_rect);
Animation an = AnimationUtils.loadAnimation(this, R.anim.grow);
iView.startAnimation(an);
```

- 이벤트 다루기

```
class MyListener implements Animation.AnimationListener {
    public void onAnimationEnd(Animation animation) {   }
    public void onAnimationRepeat(Animation animation) {   }
    public void onAnimationStart(Animation animation) {   }
}

an.setAnimationListener(new MyListener());
```

트위닝 변환

- 투명도 변경(AlphaAnimation)
  - 대상을 5초간 “페이드인”하는(완전 투명에서 완전 불투명으로) 투명도 변경 애니메이션

```
<alpha
    android:fromAlpha="0.0"
    android:toAlpha="1.0"
    android:duration="5000">
</alpha>
```

- 회전(RotateAnimation)
  - 중심을 회전축으로 하여 5초간 대상을 시계 방향으로 한 바퀴 돌리는 회전

```
<rotate
    android:fromDegrees="0"
    android:toDegrees="360"
    android:pivotX="50%"
    android:pivotY="50%"
    android:duration="5000" />
```

트위닝 변환

- 크기 변환(ScaleAnimation)
  - 5초간 대상의 크기를 두 배로 늘리는 비례 변환 애니메이션

```
<scale android:pivotX="50%"
    android:pivotY="50%"
    android:fromXScale="1.0"
    android:fromYScale="1.0"
    android:toXScale="2.0"
    android:toYScale="2.0"
    android:duration="5000" />
```

- 이동(Translate)
  - 5초간 대상을 Y축을 따라 위쪽으로(음의 방향) 100픽셀 이동하는 이동 변환 애니메이션

```
<translate android:toYDelta="-100"
    android:fillAfter="true"
    android:duration="2500" />
```