



옥상훈

**Web Site: <http://okgosu.net>**

twitter @okgosu

---

# 그래픽 API

- 커스텀 뷰
- 드로잉 기본
- Path 드로잉
- 애니메이션
- SurfaceView
- 멀티터치 처리



# 그래픽 메카니즘

---

## □ 레이아웃

- 화면상의 각 뷰들을 공간 계산(Measurement)
  - UI프레임워크는 루트뷰를 찾고 그 뷰의 measure 메소드를 호출하여 자식의 크기 확인
  - 차례대로 자식의 크기를 확인
  - 부모는 자식의 크기를 바탕으로 자신의 크기를 계산
- onMeasure 메소드를 오버라이딩해서 필요한 공간 확보
  - 부모가 제공할 수 있는 공간에 대한 픽셀 단위 너비와 높이 전달
  - 뷰는 onMeasure에서 setMeasuredDimensions를 호출해 부모에게 어느 정도의 공간이 필요한지 알려줘야 함

## □ 드로잉

- 레이아웃 단계의 공간에 직접 그리는 단계
- 안드로이드 운영체제는 뷰가 수정될 때마다 onDraw() 메서드를 호출
- 애플리케이션 상태가 변경되어 현재 화면에 그려진 내용을 고쳐야 할 때, View의 invalidate 메소드 호출
  - 이벤트 큐에 화면을 다시 그리는 이벤트 추가
  - 이벤트가 처리되면 draw 메소드 호출
- draw는 onDraw호출
  - onDraw를 오버라이딩해 원하는 내용을 그림



# 그래픽 관련 주요 클래스

---

- ❑ 캔버스 (android.graphics.Canvas)
  - ❑ 그래픽을 그리기 위한 바탕
  - ❑ Activity > View > Canvas
  - ❑ onDraw 메소드에서 오버라이딩하여 캔버스에 그림
  
- ❑ 페인트 (android.graphics.Paint)
  - ❑ 캔버스에 그릴 때 색깔, 투명도, 붓 크기 제어
  - ❑ 캔버스에 글씨를 쓸 때 글꼴, 크기, 스타일 제어
  
- ❑ 비트맵 (android.graphics.Bitmap)
  - ❑ 이미지에 대한 픽셀 데이터를 제어



# 그래픽 관련 주요 클래스

---

## □ Color

- 색의 구성: 알파(alpha), 레드(red), 그린(green), 블루(blue)
- 32 비트 정수
- 색상의 표현
  - 예) `int color = Color.BLUE; // 파란 색`
  - 예) `color = Color.argb(127,255,0,255); // 반투명 보라`
- XML 리소스에서 색상 정의
  - `<resources>`
  - `<color name="mycolor">#3500ffff</color>`
  - `</resources>`
- 자바 코드 내에서 사용
  - `color = getResources().getColor(R.color.mycolor);`



# 그래픽 관련 주요 클래스

---

## □ Path 클래스

- 선, 사각형, 커브 등 벡터 그리기 명령어를 포함
- 예) 원형 패스 정의
  - `circle = new Path();`
  - `circle.addCircle(150,150,100,Direction.CW);`
- 예) 텍스트 추가
  - `private static final String myNote = "안드로이드";`
  - `canvas.drawPath(circle, cPaint);`
  - `canvas.drawTextOnPath(myNote, circle, 0, 20, cPaint);`



# 그래픽 관련 주요 클래스

---

## □ Drawable 클래스

- 비트맵, 단색 등과 같이 화면 표시용 시각적 요소에 사용됨
- 다른 그래픽에 결합시킬 수 있으며, 사용자 인터페이스 위젯(버튼이나 뷰의 배경)에 사용됨
- Drawable 의 형태
  - 비트맵(Bitmap) – PNG 또는 JPG 이미지
  - 나인패치(NinePatch) – 늘어나는 PNG 이미지
  - 모양(Shape) – 경로에 기초한 벡터 그리기 명령어, Scalable Vector Graphics (SVG) 의 축소판
  - 레이어 (Layer) – Z-order 순서에 따라 서로 덮어 그리는 하위 drawable을 포함하는 컨테이너
  - 상태 (State) – 상태(비트 마스크; bit mask)에 맞는 하나의 하위 drawable을 보여주는 컨테이너
  - 레벨 (Level) – 컨테이너
  - 스케일 (Scale) – 컨테이너



# 애니메이션

---

- ❑ 프레임별 애니메이션
  - ❑ 조금씩 다른 이미지들을 화면에 빠르게 표시
  - ❑ 코드로 정의하기에 복잡한 그래픽 변형들에 적합
  - ❑ 비트맵 자원 객체를 생성하고 AnimationDrawable객체에 추가
  
- ❑ 트위닝 애니메이션
  - ❑ 비트맵, 도형, 텍스트 등의 변환 방식을 개발자가 지정하면 시스템이 적절히 프레임을 만들어서 애니메이션 효과를 보여줌
  - ❑ 알파, 회전, 확대, 축소, 이동과 같은 변환을 지원
  - ❑ XML을 이용한 트위닝 애니메이션 설정





# 멀티터치

---

- ❑ 터치 이벤트 구현

- ❑ **View에서** onTouchEvent(MotionEvent event) 함수 오버라이딩

- ❑ 멀티 터치 여부: event.getPointerCount()해서 2면 멀티터치

- ❑ 안드로이드 2.1

- ❑ event.getAction()이 MotionEvent.ACTION\_DOWN 인 경우

- ❑ 첫번째 지점: event.getX(0), event.getY(0);

- ❑ 두번째 지점 : event.getX(1), event.getY(1);

- ❑ 안드로이드 2.2

- ❑ event.getAction() && MotionEvent.ACTION\_MASK 의 결과 값이 MotionEvent.ACTION\_POINTER\_DOWN 인 경우에

- ❑ 첫번째 지점: event.getX(0), event.getY(0);

- ❑ 두번째 지점 : event.getX(1), event.getY(1);



# 멀티터치

---

- ❑ 터치 관련 동작
  - ❑ MotionEvent.ACTION\_DOWN: 화면을 터치
  - ❑ MotionEvent.ACTION\_MOVE: 터치한 상태에서 이동
  - ❑ MotionEvent.ACTION\_UP: 터치 종료
  
- ❑ 줌 컨트롤
  - ❑ 무브전 이전 지점 좌표 저장
  - ❑ 이동하는 지점간의 거리를 계산



---

# 폰 서비스

- 구글맵
- 센서 활용
- 네트워크 상태 조회



# 위치정보

---

- ❑ Global Positioning System (GPS)
  - ❑ 기본적으로 10~25m 정도의 오차를 가짐
  - ❑ 내비게이션은 자체 알고리즘으로 이를 보정
    - ❑ 인근 휴대폰 기지국
    - ❑ Wifi 핫스팟
  - ❑ 한국에서 위치정보를 제공하는 사업자에게는 국가 기관이 위치 정보 내역을 요구할 수 있음
- ❑ 안드로이드에서는 Dalvik Debug Monitor Service (DDMS) 를 이용하여 위치정보를 에뮬레이션함



# 위치정보 관련 클래스

---

## ❑ LocationManager

- ❑ 위치기반 서비스 제어하며 LocationProvider 값을 추출

- ❑ `LocationManager locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);`

## ❑ LocationProvider

- ❑ 장치의 현재 위치를 결정하는데 사용되는 서로 다른 각각의 위치검색 기술을 표현 예) GPS
- ❑ LocationProvider를 통해 현재 위치 정보를 업데이트함
- ❑ LocationProvider는 다음과 같은 방법으로 획득 가능
  - ❑ `getProvider` 호출
    - ❑ `String providerName = LocationManager.GPS_PROVIDER;`
    - ❑ `LocationProvider gpsProvider = locationManager.getProvider(providerName);`



# 위치정보 관련 클래스

---

- ❑ 이용할 수 있는 공급자 찾기
  - ❑ `List<String> providers = locationManager.getProviders(True);`
- ❑ Criteria 를 이용하여 특정 조건으로 찾기
  - ❑ `setAccuracy(Criteria.ACCURACY_COARSE / Criteria.ACCURACY_FINE)` : 정확도
  - ❑ `setPowerRequirement(Criteria.POWER_LOW / )` : 전력소비량
  - ❑ `setAltitudeRequired(true / false)` : 고도데이터 사용 유무
  - ❑ `setBearingRequired(true / false)` : 방위데이터 사용 유무
  - ❑ `setSpeedRequired(true / false)` : 속도데이터 사용 유무
  - ❑ `setCostAllowed(true / false)` : 비용 들지 여부
- ❑ `LocationManager.getBestProvider(criteria, {true|false})`
  - ❑ 요구사항에 가장 부합하는 위치공급자를 리턴
- ❑ `LocationManager.getProvider(criteria, {true|false})`
  - ❑ 요구사항에 부합하는 위치공급자를 리스트형태로 리턴



# 현재 위치 확인

---

- ❑ Location Manager 접근
  - ❑ `LocationManager locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);`
- ❑ Location 객체
  - ❑ `String provider = LocationManager.GPS_PROVIDER;`
  - ❑ `Location location = locationManager.getLastKnownLocation(provider);`
- ❑ 움직임 추적
  - ❑ **public void** `onLocationChanged(Location location)` 구현
  - ❑ `locationManager.requestLocationUpdates(provider, 밀리초, 거리, LocationListener)`
  - ❑ `locationManager.removeUpdates(LocationListener)`



## 근접 경보 사용

---

- ❑ 사용자가 특정 위치로 이동하거나 그 위치에서 벗어날 때 어 플리케이션이 반응하도록함
- ❑ 근접 경보 발생시 intent 호출
  - ❑ `Intent intent = new Intent();`
  - ❑ `PendingIntent pendingIntent = PendingIntent.getBroadcast(this, -1, intent, 0);`





# Geocoder

---

- ❑ 주소와 경도/위도 맵 좌표간의 정보를 서로 변환.
  - ❑ Forward Geocoding : 특정 주소의 위도와 경도를 확인
  - ❑ Reverse Geocoding : 주어진 위도와 경도에 대한 주소 확인
- ❑ `Geocoder goecoder = new Geocoder(getApplicationContext(), Locale.getDefault());`



# 구글맵 사용

---

## ❑ MapView

### ❑ 구글맵이 표시되는 View

- ❑ `mapView.setSatellite(true)` : 위성 뷰
- ❑ `mapView.setStreetView(true)` : street뷰
- ❑ `mapView.setTraffic(true)` : 예상 교통량

## ❑ MapActivity

### ❑ 맵 뷰를 포함하는 activity

## ❑ ItemizedOverlay, OverlayItem

### ❑ 맵에 표시되는 마커와 마커의 내용

- ❑ `List<Overlay> overlays = mapView.getOverlay();`
- ❑ `MyItemizedOverlay markers = new MyItemizedOverlay(R.drawable.img);`
- ❑ `overlays.add(markers);`



# 구글맵 사용

---

## ❑ MapController

- ❑ 맵을 제어. 중심 위치와 확대 단계를 설정
- ❑ 맵 컨트롤러 표시
  - ❑ `View zoomControls = mapView.getZoomControls();`
  - ❑ `mapView.addView(zoomControls, MapView.LayoutParams);`
  - ❑ `mapView.displayZoomControls(true);`
- ❑ Location객체에 저장된 위도 경도 값을 이용하려면 GeoPoint에 백만단위로 변환한 뒤 저장
  - ❑ `MapViewController mc = MapView.getController();`
  - ❑ `Double lat = 37.123 * 1E6;`
  - ❑ `Double lng = 123.123 * 1E6;`
  - ❑ `GeoPoint point = new GeoPoint(lat.intValue(), lng.intValue());`
- ❑ MapView 중심을 다시 잡거나 확대하려면
  - ❑ `mc.setCenter(point);`
  - ❑ `mc.setZoom(1); : 1~21`



# 구글맵 사용

---

## ❑ MyLocationOverlay

- ❑ 현재 위치와 방향을 MapView위에 보여주도록 설계된 오버레이
- ❑ 사용방법
  - ❑ `List<Overlay> overlays = mapView.getOverlays();`
  - ❑ `MyLocationOverlay mylocationoverlay = new MyLocationOverlay(this, mapView);`
  - ❑ `overlays.add(mylocationoverlay);`
- ❑ 나침반과 마커 모두를 활성화
  - ❑ `mylocationoverlay.enableCompass();`
  - ❑ `mylocationoverlay.enableMyLocation(mapView.getMapController());`



# 센서

---

## ❑ SensorManager

- ❑ `SensorManager mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);`
- ❑ 센서 서비스는 값이 변경될 때마다 `onSensorChanged()`를 호출함
- ❑ 모든 센서는 부동 소수점 명령을 반환함
- ❑ 반환되는 배열의 크기는 센서의 특성에 따라 다름

## ❑ 센서 종류

- ❑ `SENSOR_ACCELEROMETER` – x,y,z 좌표의 가속 측정
- ❑ `SENSOR_LIGHT` – 주위 영역의 밝기를 알려줌
- ❑ `SENSOR_MAGNETIC_FIELD` – x,y,z, 축에 전자 극을 반환함
- ❑ `SENSOR_ORIENTATION` – 장치가 한쪽으로 쏠리고, 던져지고, 돌려지는 것을 측정함
- ❑ `SENSOR_PROXIMITY` – 센서와 객체 사이의 거리를 알려줌
- ❑ `SENSOR_TEMPERATURE` – 주위의 온도를 측정함
- ❑ `SENSOR_TRICORDER` – 장치를 영화 스타트렉의 트라이코더와같이 만듦



# 안드로이드 기기의 네트워크 상태 조회

---

## ❑ ConnectivityManager 클래스

### ❑ 네트워크 연결 상태 점검

### ❑ Context의 getSystemService() 메서드를 호출해서 객체 리턴

### ❑ getNetworkInfo()

#### ❑ 네트워크의 상태를 알수있음

```
ConnectivityManager cm = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo ni =
    cm.getNetworkInfo(ConnectivityManager.TYPE_WIFI);
boolean isWifiAvail = ni.isAvailable();
boolean isWifiConn = ni.isConnected();
ni = cm.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);
boolean isMobileAvail = ni.isAvailable();
boolean isMobileConn = ni.isConnected();

status.setText("WiFi\nAvail = "+ isWifiAvail +
    "\nConn = " + isWifiConn +
    "\nMobile\nAvail = "+ isMobileAvail +
    "\nConn = " + isMobileConn);
```



# 전화기 상태 정보 조회

---

## □ 개요

- 통화중 상태, 단말기 전화번호, 기기 식별자, 일련 번호 추출
- 메니 페스트 파일에 READ\_PHONE\_STATE 권한 설정 필요
- TelephonyManager 객체를 이용해서 전화기 상태 및 단말기 전화 번호 조회 가능

## □ 구현

- 전화기 음성 통화 상태 조회 : getCallState() 메서드 이용
- 통화 상태 변경시 리스너의 구현과 등록
  - PhoneStateListener 인터페이스 구현해서 리스너 구현
  - TelephonyManager의 listen() 메서드를 이용 리스너 등록
- 기타 정보
  - 망운영 회사 정보 조회: getNetworkOperarion() 메서드
  - 이동 통신사 조회: getSimOperatorName() 메서드
  - 전화 번호 조회: getLine1Number() 메서드



# 통화 상태 조회

---

- TelephonyManager 객체를 이용해서 전화기 상태 및 단말기 전화 번호 조회

```
TelephonyManager telManager = (TelephonyManager)
    getSystemService(Context.TELEPHONY_SERVICE);
```

- 전화기 음성 통화 상태 조회

- getCallState() 메서드 이용

```
int callStatus = telManager.getCallState();
String callState = null;

switch (callStatus) {
    case TelephonyManager.CALL_STATE_IDLE:
        callState = "Phone is idle.";
        break;
    case TelephonyManager.CALL_STATE_OFFHOOK:
        callState = "Phone is in use.";
        break;
    case TelephonyManager.CALL_STATE_RINGING:
        callState = "Phone is ringing!";
        break;
}

Log.i("telephony", callState);
```





# 통화 상태 조회

---

- 통화 상태 변경시 리스너의 구현과 등록
  - PhoneStateListener 인터페이스 구현해서 리스너 구현
  - TelephonyManager의 listen() 메서드를 이용 리스너 등록

```
telManager.listen(new PhoneStateListener() {  
    public void onCallStateChanged(  
        int state, String incomingNumber) {  
        String newState = getCallStateString(state);  
        if (state == TelephonyManager.CALL_STATE_RINGING) {  
            Log.i("telephony", newState +  
                " number = " + incomingNumber);  
        } else {  
            Log.i("telephony", newState);  
        }  
    }  
}, PhoneStateListener.LISTEN_CALL_STATE);
```



# 이동통신 서비스 정보 조회

---

- ❑ 망운영 회사 정보 조회
  - ❑ getNetworkOperatorName() 메서드
- ❑ 이동 통신사 조회
  - ❑ getSimOperatorName() 메서드
- ❑ 전화 번호 조회
  - ❑ getLine1Number() 메서드

```
String opName = telManager.getNetworkOperatorName();  
Log.i("telephony", "operator name = " + opName);
```

```
String phoneNumber = telManager.getLine1Number();  
Log.i("telephony", "phone number = " + phoneNumber);
```

```
String providerName = telManager.getSimOperatorName();  
Log.i("telephony", "provider name = " + providerName);
```



---

수고하셨습니다 ^^/

