



학생 여러분 반갑습니다.  
다른 친구들이 입장할 때까지  
조금 기다려 주십시오.

곧 모바일 프로그래밍 수업을  
시작합니다.

음소거(🔇)가 되었는지 확인 바랍니다.

모바일 프로그래밍  
화목(1,2교시)/ 화목(3,4교시)  
정윤현 (AI/소프트웨어학부)



# Mobile Programming

## Android Programming

Chap 9. Multimedia & Location based Service

Prof. Younhyun Jung

Email) [younhyun.jung@gachon.ac.kr](mailto:younhyun.jung@gachon.ac.kr)

# Android multimedia framework



- The Android multimedia framework includes support for playing variety of common media types, so that you can easily integrate audio, video and images into your applications.  
↳ 가능성이 있음?
  - You can play audio or video from media files stored in your application's resources (raw resources), from standalone files in the filesystem, or from a data stream arriving over a network connection, all using MediaPlayer APIs.
- Must read : <https://developer.android.com/guide/topics/media/mediaplayer.html>

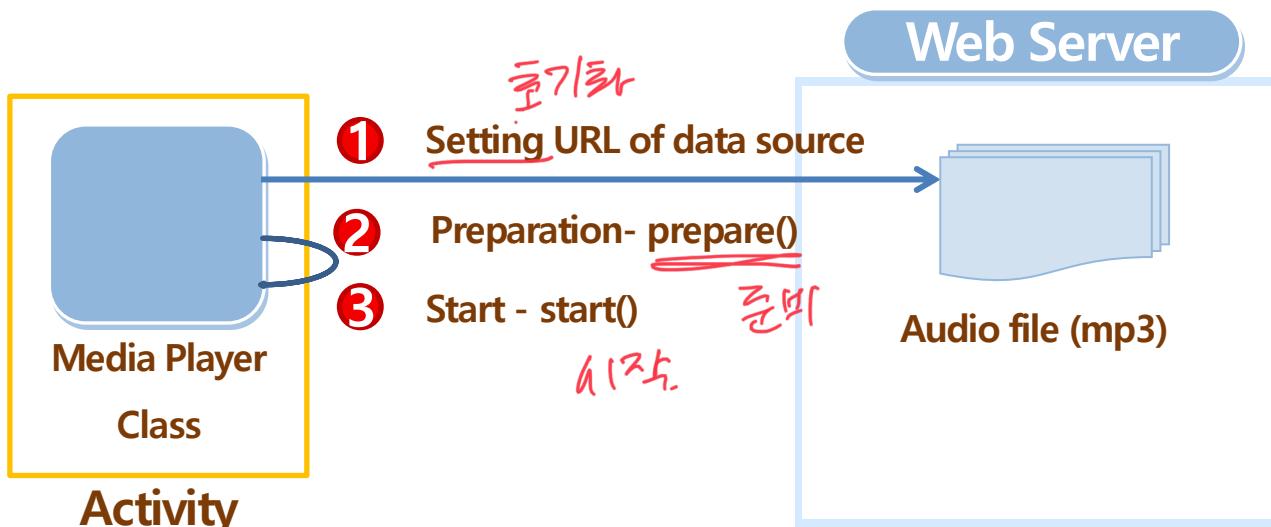
# Using MediaPlayer



- One of the most important components of the media framework is the **MediaPlayer** class. An instance of this class can fetch, decode, and play both audio and video with minimal setup. It supports several different media sources such as:  
⇒ 다양한 타입 지원!
- Local resources : internal/external storages ✓
- External URLs (streaming) ✓
- Etc..



# Using MediaPlayer



- Must read : <https://developer.android.com/guide/topics/media/mediaplayer.html>

Try to make an Audio Player

# Exercise



- **activity\_main.xml**

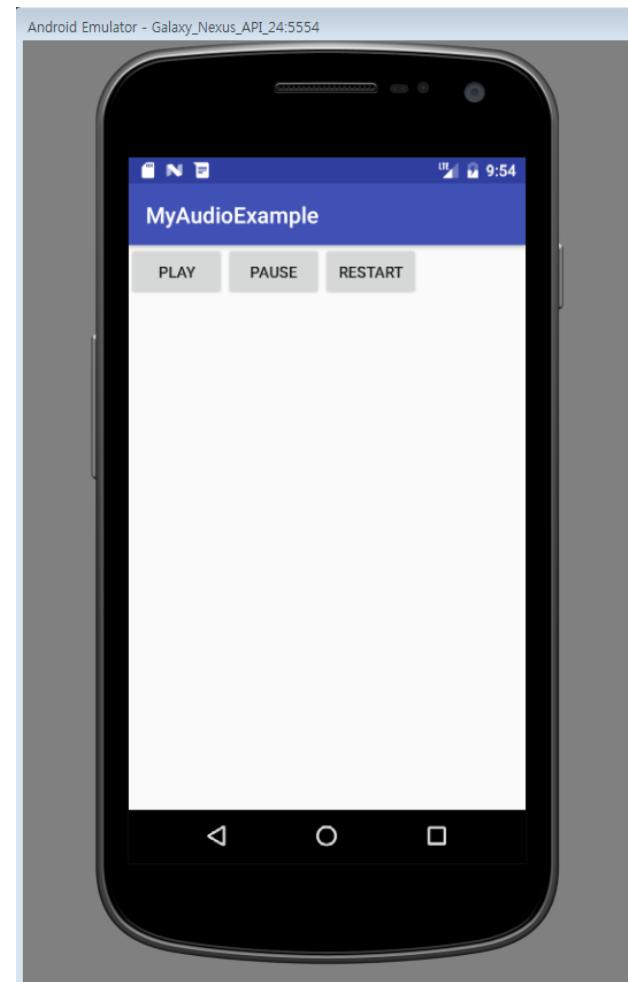
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <Button
        android:id="@+id/plyBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Play" />

    <Button
        android:id="@+id/pauseBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pause" />

    <Button
        android:id="@+id/restartBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Restart" />

</LinearLayout>
```





# Exercise

- MainActivity.java

```
package org.androidtown.listview.myaudioexample;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity {
    static final String AUDIO_URL = "http://sites.google.com/site/ubiaccessmobile/sample_audio.amr";
    private MediaPlayer mediaPlayer;
    private int playbackPosition = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button startBtn = (Button) findViewById(R.id.plyBtn);
        Button pauseBtn = (Button) findViewById(R.id.pauseBtn);
        Button restartBtn = (Button) findViewById(R.id.restartBtn);

        startBtn.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {
                try{
                    playAudio(AUDIO_URL); ✓
                    Toast.makeText(getApplicationContext(),"음악 파일 재생 시작됨.", Toast.LENGTH_LONG).show();
                } catch (Exception e){e.printStackTrace();}
            }
        });
    }
}
```

decratatic = true.



# Exercise

- MainActivity.java

```
pauseBtn.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v) {
        if (mediaPlayer != null){
            playbackPosition = mediaPlayer.getCurrentPosition();
            mediaPlayer.pause();
            Toast.makeText(getApplicationContext(),"음악 파일 재생 중지됨.", Toast.LENGTH_LONG).show();
        }
    }
});

restartBtn.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v) {
        if(mediaPlayer != null && !mediaPlayer.isPlaying()){
            mediaPlayer.start();
            mediaPlayer.seekTo(playbackPosition);
            Toast.makeText(getApplicationContext(),"음악 파일 재생 재시작됨.", Toast.LENGTH_LONG).show();
        }
    }
});
```

→ 턴제 미션입니다.



# Exercise

- MainActivity.java

```
private void playAudio(String url) throws Exception {  
    killMediaPlayer();  
  
    mediaPlayer=new MediaPlayer();  
    mediaPlayer.setDataSource(url);  
    mediaPlayer.prepare();  
    mediaPlayer.start();  
}  
  
protected void onDestroy(){  
    super.onDestroy();  
    killMediaPlayer();  
}  
  
private void killMediaPlayer(){  
    if(mediaPlayer!=null){  
        try{mediaPlayer.release();  
        } catch (Exception e){  
            e.printStackTrace();  
        }  
    }  
}
```

?



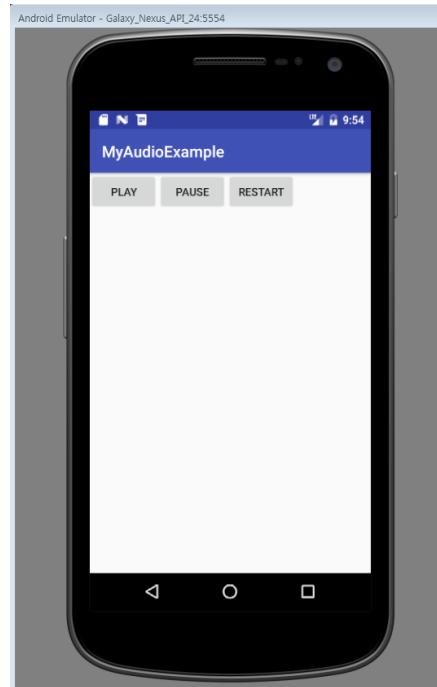
# Exercise

- Add Permission in manifest file

```
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
```

```
    android:usesCleartextTraffic="true"
```

- RUN!!



Try to make a VideoView

# Exercise



- activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

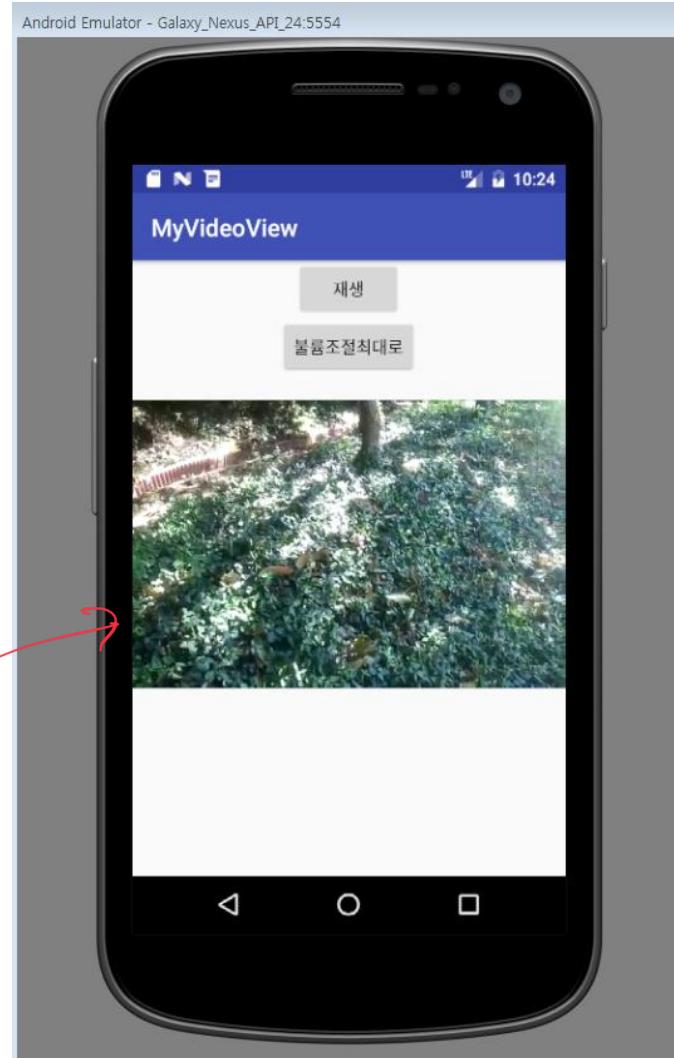
    <Button
        android:id="@+id/startBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="재생"
        android:layout_gravity="center"/>

    <Button
        android:id="@+id/volumeBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="볼륨조절최대로"
        android:layout_gravity="center"/>

    <VideoView
        android:id="@+id/videoView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="20dp"/>
</LinearLayout>
```

gachon

제작되었습니다!





# Exercise

- MainActivity.java

```
package org.androidtown.listview.multimedia;

import android.media.AudioManager;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.MediaController;
import android.widget.VideoView;
public class MainActivity extends AppCompatActivity {
    static final String VIDEO_URL = "http://sites.google.com/site/ubiaccessmobile/sample_video.mp4";
    private VideoView videoView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        videoView = (VideoView) findViewById(R.id.videoView);
        MediaController mc = new MediaController(this);
        videoView.setMediaController(mc);
        videoView.setVideoURI(Uri.parse(VIDEO_URL));
    }
}
```

✓

이미 컨트롤러와  
연결되어 있음!



# Exercise

- MainActivity.java

```
Button startBtn = (Button) findViewById(R.id.startBtn);
    Button volumeBtn = (Button) findViewById(R.id.volumeBtn);
    startBtn.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
            videoView.seekTo(0);
            videoView.start();
        }
    });
    volumeBtn.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
            AudioManager mAudioManager = (AudioManager) getSystemService(AUDIO_SERVICE);
            int maxVolume = mAudioManager.getStreamMaxVolume(AudioManager.STREAM_MUSIC);
            mAudioManager.setStreamVolume(AudioManager.STREAM_MUSIC, maxVolume, AudioManager.FLAG_SHOW_UI);
        }
    });
}
```

시작Btn 클릭 시 비디오 재생 및 초기화  
volumeBtn 클릭 시 음량 조절

- Add Permission in manifest file & RUN!

```
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
```

```
    android:usesCleartextTraffic="true"
```

# Location Based Services (LBS)



- One of the unique features of mobile applications is location awareness
  - adding location awareness to your app offers users a more contextual experience
  - That is, Location-based services or **LBS**, became today's one of Killer applications!
- A location-based service (LBS) is an information system driven by the ability of the 'central' system to detect the geographical position of the mobile devices.  
Manager를 사용하여  
고장이?  
→ 개인적인 위치를 제공합니다!
- LBS apps track your location, and may offer additional services such as locating amenities nearby, as well as offer suggestions for route planning, and so on.  
시작점.  
navigate.





# Location Services

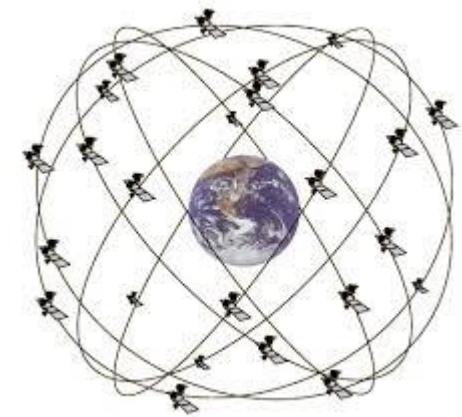
- Location Based Services are used in a variety of situations,
  - such as ***commercial, entertainment, emergency, health, work, personal life, etc.***
- Application Examples:
  - Provide transportation information on how to go from ‘here’ to ‘there’.
    - E.g., Naver Map, Daum Map, ...
  - Locate the nearest bank, restaurant, gas station, hotel, golf course, hospital, police station, etc.
  - Social networking is used to locate and reach events, friends and family members.
    - E.g., 1 km



# How the Global Positioning System (GPS) Works ?



- GPS (Global Positioning System)
- a satellite based navigation system that can be used to locate positions anywhere on earth
- Designed and operated by the U.S. Department of Defense (DOD) under the name NAVSTAR (Navigation System for Timing and Ranging)
- consists of
  - satellites, control and monitor stations, and receivers

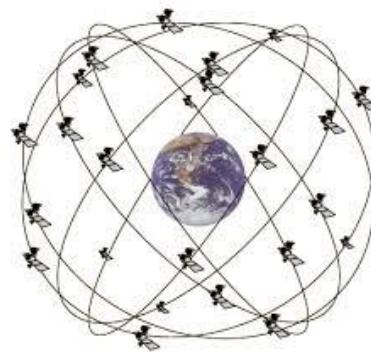


[http://www.faa.gov/about/office\\_org/headquarters\\_offices/ato/service\\_units/techops/navservices/gnss/gps/howitworks/](http://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/howitworks/)



# GPS Principle

- GPS (Global Positioning System)
  - The Global Positioning System(GPS) consists of 27 Earth-orbiting satellites (24 in operation and three extras in case one fails).
    - Developed by the USA as a military navigation system, but soon it opened to other civilian uses.

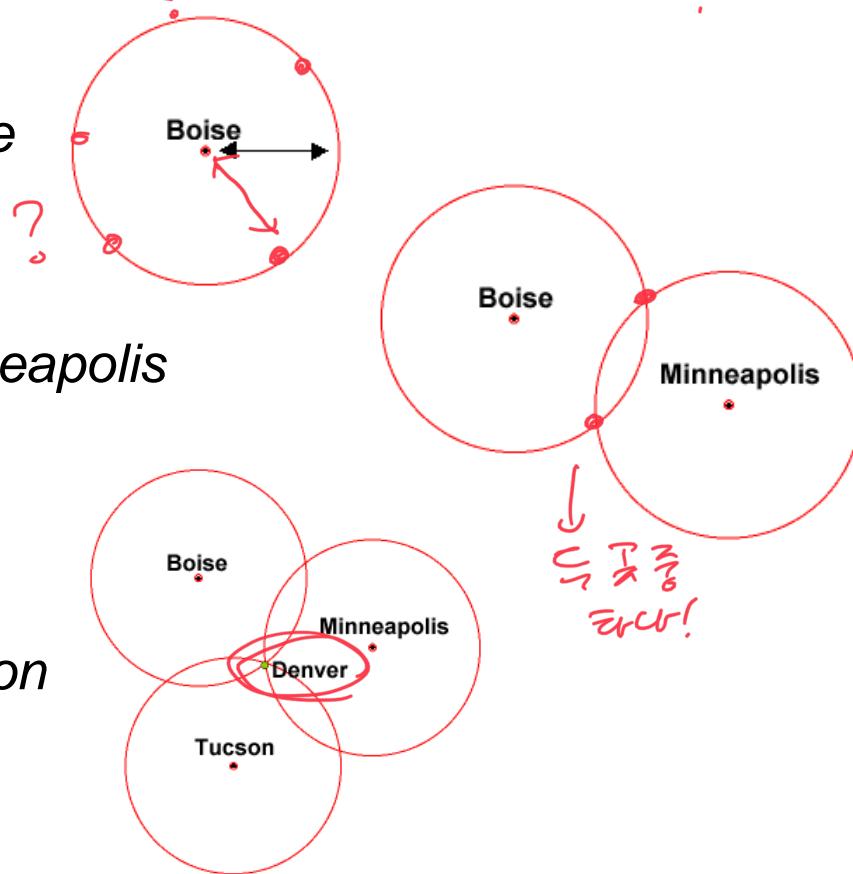


- solar-powered satellites circles the globe at about 12,000 miles (19,300 km), making **two complete rotations** every day
- The orbits are arranged so that at any time, anywhere on Earth, there are **at least four satellites** "visible" *in the sky*.

# 2-D Trilateration (삼변 측량)



- You are 625 miles from Boise
- You are 690 miles from Minneapolis
- you are 615 miles from Tucson

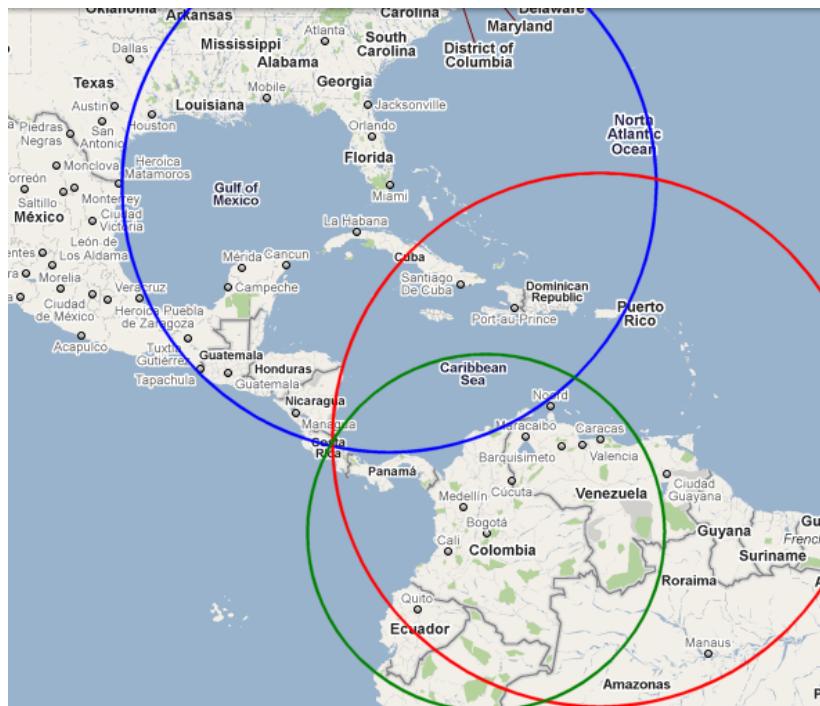


You now know exactly where you are -- Denver, Colorado!

# Global Positioning System (GPS)



- A GPS receiver's job is to *locate three or more of these satellites*, figure out the distance to each, and use this information to deduce its own location. This operation is based on a mathematical principle called **trilateration**.



--- Miami 1795 km  
--- Caracas 1874 km  
--- Bogota 1251 km

The actual location is:  
San Jose, Costa Rica.

$$거리 = 전파 속도 \times 전파 도달 시간$$

$$r=ct$$

r : 거리

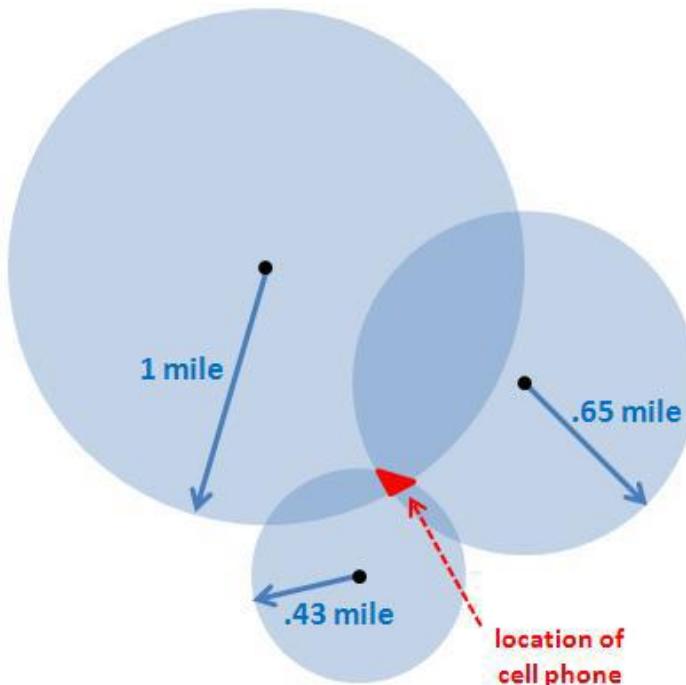
c : 전파의 속도

T : 인공위성으로부터 전파가 발사되어 GPS수신기까지 도달하는데 필요한 시간



# Cell Tower Triangulation

- An alternative method to determine the location of a cell phone is to estimate its distance to three nearby cell towers.

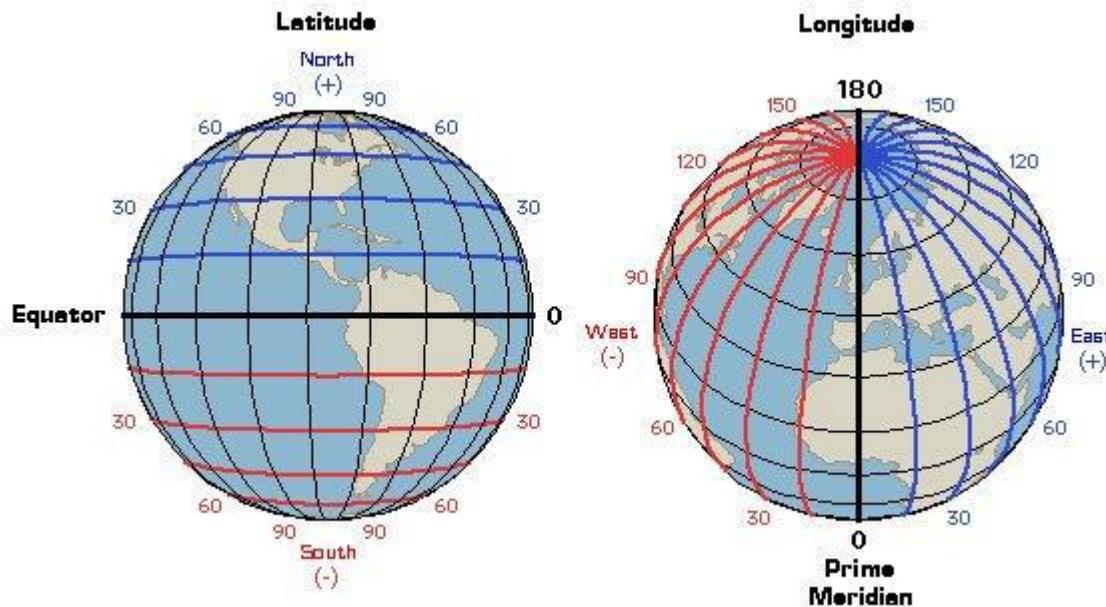


*Triangulation - cell phone detected within a certain radius of each of 3 cell towers – the area where each cell tower overlaps the phone is where it is pinpointed.*



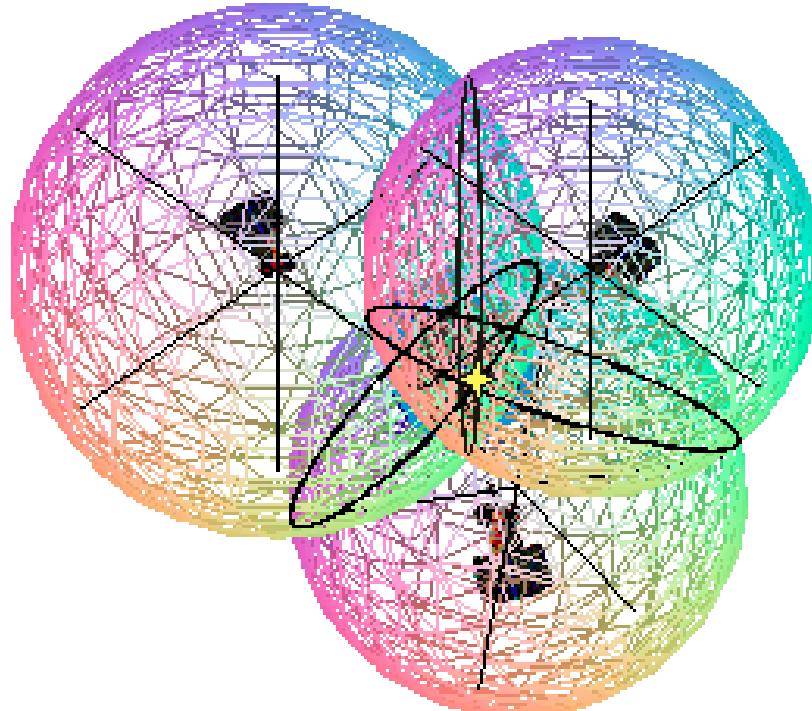
# Basic : Latitude & Longitude

- **Latitude** (위도) in GPS-Decimal notation: +90.000000 (North) to -90.000000 (South)
- **Longitude** (경도) in GPS-Decimal notation: +180.000000 (East) to -180.000000 (West)





# 3D-Trilateration



Three spheres

- <http://electronics.howstuffworks.com/gadgets/travel/gps2.htm>



# Android Location Classes

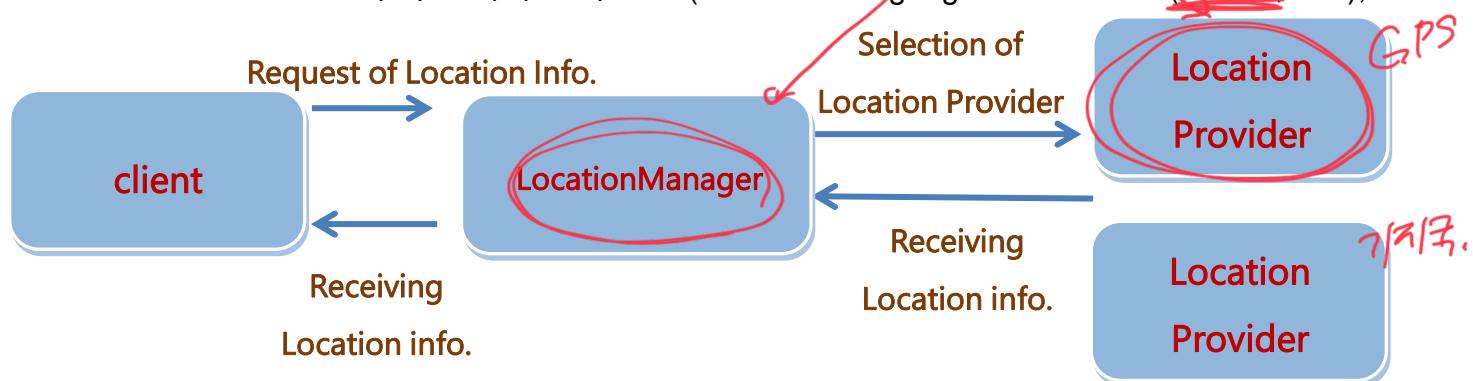
- The Android API provides Location data based on a variety of methods including:
  - Cell Tower Triangulation,
  - (most commonly) GPS chip readings
- Important Classes for LBS
  - Location Manager
    - The class provides access to the system location services.
  - Location Classes
    - Classes (objects) representing a geographic location sensed at a particular time.
  - LocationListener class

위치 변경 등을  
제작



# Overall Procedure of LBS

- LocationManager
- LocationProvider : LocationManager는 위치정보를 받도록 도와주는/중개자 역할만 수행. 실제 위치 정보는 LocationProvider가 제공
  - e.g.,Network/GPS/Passive
  - Location Provider 직접 선택 방법 (LocationManager.GPS\_PROVIDER)
  - Location Provider를 조건에 따른 최적 선택 방법 (locationManager.getBestProvider(criteria, true);



- getLastKnownLocation(string provider) : 마지막 확인된 위치 정보 획득

```
LocationManager locationManager =  
        (LocationManager) getSystemService(Context.LOCATION_SERVICE);  
Location lastLocation =  
        locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);  
Double latitude = lastLocation.getLatitude();
```



# Overall Procedure of LBS

- requestLocationUpdate() methods : After selecting Location Provider, we need to request Location update from it!
    - Implementing LocationListener
  - RemoveUpdates(LocationListener listener) : stop location updates
- OS가 Feedback  
↳ 수동으로 처리?



# Location Manager class

The class provides access to the system location services.

These services allow applications

1. To obtain *periodic updates* of the device's geographical location,  
    주기적 업데이트 or ?
2. to fire an application-specified Intent when  
the device enters the proximity of a given geographical location.

- You retrieve the Android location manager through
  - **Context.getSystemService(Context.LOCATION\_SERVICE);**

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



# Location class

- A class representing a geographic location sensed at a particular time.
  - Each class object contains a particular location information.
- A location consists of a latitude and longitude, a local timestamp and optionally information on altitude, speed, and direction (bearing).
  - Methods:
    - getLatitude(), getLongitude(), getTime(), ...

```
private class GPSListener implements LocationListener {  
    public void onLocationChanged(Location location) {  
        // capture location data sent by current provider  
        double latitude = location.getLatitude();  
        double longitude = location.getLongitude();
```



# LocationListener class

- Used for receiving notifications from the **LocationManager** when the location has changed.
- These methods are called if the **LocationListener** has been *registered* with the location manager service using the method:

**requestLocationUpdates (Provider, minTime, minDistance, LocationListener)**

최소 시간      최소 변동거리.

이 User가  
어디에 있나?



# Example

- LocationManager, LocationListener

```
LocationManager lm =  
    (LocationManager) getSystemService(Context.LOCATION_SERVICE);  
  
// This listener will catch and disseminate location updates  
myLocationListener= new GPSListener();  
longminTime= 10000; // frequency update: 10 seconds  
floatminDistance= 50; // frequency update: 50 meter Som 빈동 AI  
Listener update.  
lm.requestLocationUpdates(//request GPS updates  
    LocationManager.GPS_PROVIDER,  
    minTime,  
    minDistance,  
    myLocationListener);
```

*GPS Provider*



# Example

- LocationListener

GPS Listener 은  
무엇인가요?

```
String provider = LocationManager.GPS_PROVIDER;  
int t = 5000; // milliseconds  
int distance = 5; // meters
```

```
LocationListener myLocationListener = new LocationListener() {
```

```
    public void onLocationChanged(Location location) {
```

Called when the location has changed.

```
        // Update application based on new location.  
    }
```

```
    public void onProviderDisabled(String provider){
```

Called when the provider is disabled/enabled by the user.

```
        // Update application if provider disabled.  
    }
```

```
    public void onProviderEnabled(String provider){
```

```
        // Update application if provider enabled.  
    }
```

```
    public void onStatusChanged(String provider, int status, Bundle extras){
```

Called when the provider status changes.

```
        // Update application if provider hardware status changed.  
    }
```

```
};
```

```
locationManager.requestLocationUpdates(provider, t, distance, myLocationListener);
```



# Location Provider

- LocationProvider Class

- An abstract superclass for location providers.

LocationProvider	Description
Network	Uses the <u>mobile network</u> or <u>Wi-Fi</u> to determine the best location. Might have a higher precision in closed rooms than GPS. → 실내 특위 가능.
GPS	Use the GPS receiver in the Android device to determine the best location via satellites. Usually better precision than network.
passive	Allows to participate in location of updates of other components to save energy (다른 앱에서 사용한 위치 측정 정보를 재활용)

- A location provider *supplies periodic reports on the geographical location of the device*.
- They may also have *different battery consumption characteristics or monetary costs to the user*.
- Selecting LocationProvider via Criteria
  - The **Criteria** class allows providers to be selected based on user-specified criteria.



# Choose Location Provider

- The user can specify the location provider explicitly in the code using a number of constants:
  - LocationManager.GPS\_PROVIDER
  - LocationManager.NETWORK\_PROVIDER
  - LocationManager.PASSIVE\_PROVIDER
- We can use Criteria.

```
Criteria criteria = new Criteria();
criteria.setAccuracy(Criteria.ACCURACY_FINE); // 정확도 설정
criteria.setPowerRequirement(Criteria.POWER_LOW); // 전원 소모량 허용 정도
criteria.setAltitudeRequired(false); // 고도 정보 필요 여부?
criteria.setBearingRequired(false); // 방위각 정보 필요 여부?
criteria.setSpeedRequired(false); // 속도 정보 필요 여부?
criteria.setCostAllowed(true); // 추가 요금부가 허용?
```

```
String provider = locationManager.getBestProvider(criteria, true);
```



# Exercise

- activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="55dp"
        android:text="내위치 확인하기" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="92dp"
        android:text="내 위치"
        android:textSize="18dp"/>
</RelativeLayout>
```





# Exercise

- MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textView = (TextView) findViewById(R.id.textView);

        Button button = (Button) findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                startLocationService();
            }
        });
        requestPermissions(new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
            0);
    }
}
```

```
package org.androidtown.listview.mylibs;
import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
```

uses-permission  
코드로 퍼미션  
가능하다!



# Exercise

```
/**  
 * 위치 정보 확인을 위해 정의한 메소드  
 */  
private void startLocationService() {  
    LocationManager manager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);  
  
    GPSListener gpsListener = new GPSListener();  
    long minTime = 10000;  
    float minDistance = 0;  
  
    try {  
        manager.requestLocationUpdates(  
            LocationManager.GPS_PROVIDER,  
            minTime,  
            minDistance,  
            gpsListener);  
  
        // 위치 확인이 안되는 경우에도 최근에 확인된 위치 정보 먼저 확인  
        Location lastLocation = manager.getLastKnownLocation(LocationManager.GPS_PROVIDER);  
        if (lastLocation != null) {  
            Double latitude = lastLocation.getLatitude();  
            Double longitude = lastLocation.getLongitude();  
            위치 정보 있을 때  
            ↳ 표시?  
            textView.setText("내 위치 : " + latitude + ", " + longitude);  
            Toast.makeText(getApplicationContext(), "Last Known Location : " + "Latitude : " + latitude + "\nLongitude:"  
+ longitude, Toast.LENGTH_LONG).show();  
        }  
    } catch(SecurityException ex) {  
        ex.printStackTrace();  
    }  
  
    Toast.makeText(getApplicationContext(), "위치 확인이 시작되었습니다. 로그를 확인하세요.", Toast.LENGTH_SHORT).show();  
}
```

GPS 사용자  
Event handler.

Distance 조건 엔진.



# Exercise

```
/**  
 * 리스너 클래스 정의  
 */  
private class GPSListener implements LocationListener {  
    /**  
     * 위치 정보가 확인될 때 자동 호출되는 메소드  
     */  
    public void onLocationChanged(Location location) {  
        Double latitude = location.getLatitude();  
        Double longitude = location.getLongitude();  
  
        String msg = "Latitude : " + latitude + "\nLongitude:" + longitude;  
        Log.i("GPSListener", msg);  
  
        textView.setText("내 위치 : " + latitude + ", " + longitude);  
        Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();  
    }  
  
    public void onProviderDisabled(String provider) {}  
  
    public void onProviderEnabled(String provider) {}  
  
    public void onStatusChanged(String provider, int status, Bundle extras) {}  
}
```

실제 GPS 정보가 있을 때

# Exercise

- Add Permission in

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

- RUN!!

- You may send GPS value generated from emulator!

이치 ⇒ 학교 건물 중 탐색!

