RhaRham
Interactive Inc.

센서 / 카메라 / 위치정보 활용

공개 SW 개발자 대회

㈜라람인터랙티브
박 유 태
@컨버전스(안드로이드 펍)

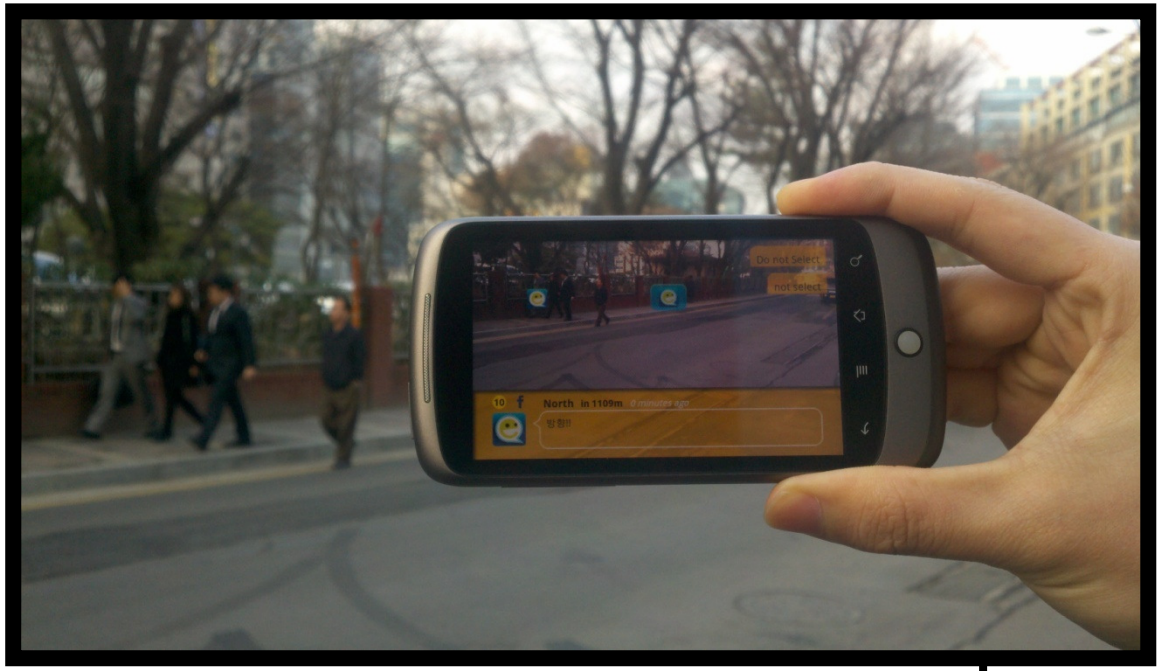
Android Camera

RhaRham
Interactive Inc.

Android Location

(GPS, NETWORK)

Android COMPASS



Android Camera

Android Location
(GPS, NETWORK)

Android COMPASS

AR

RhaRham Interactive Inc.

Subject Index

1. Augmented Reality
2. Camera
3. Location
4. Compass
5. OpenSource

What is AR?

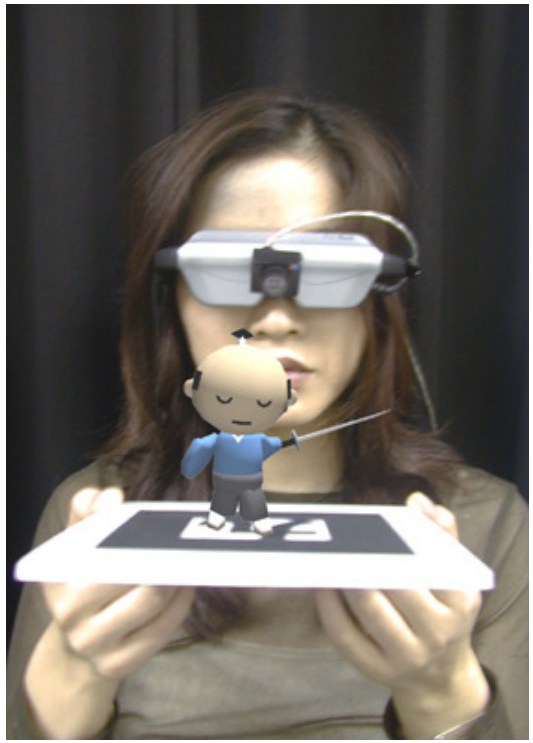
- 실제와 가상의 혼합
- Ronald Azuma의 정의
  - 현실(Real- world elements)의 이미지와 가상의 이미지를 결합한 것.
  - 실시간으로 인터랙션(interaction)이 가능한 것.
  - 3차원의 공간 안에 놓인 것.

VR (Virtual Reality)

- 실제와 가상의 혼합
  - Virtual Reality

AR (Augmented Reality)

- 실제와 가상의 혼합
  - Augmented Reality



< ARToolkit >

Location- based AR

- AR 구현의 필수 고려 사항

특징!! + 현실

RhaRham Interactive Inc.

Location- based AR

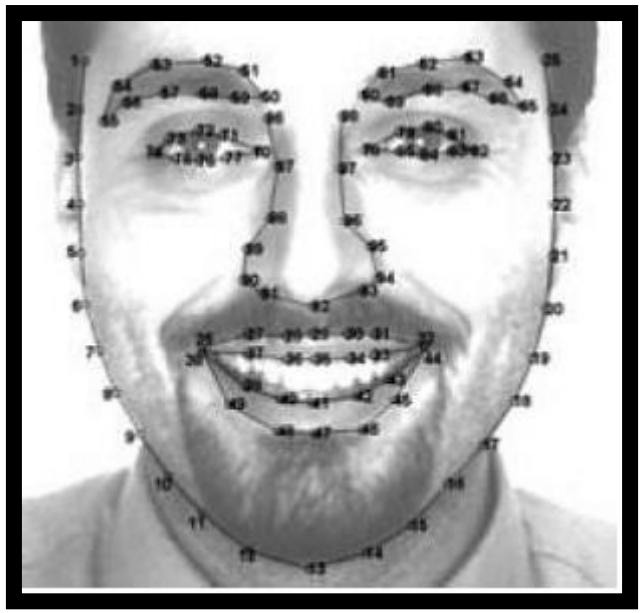
- 특징 추출
  - Marker, Markerless
  - Location

- But.

NO!!

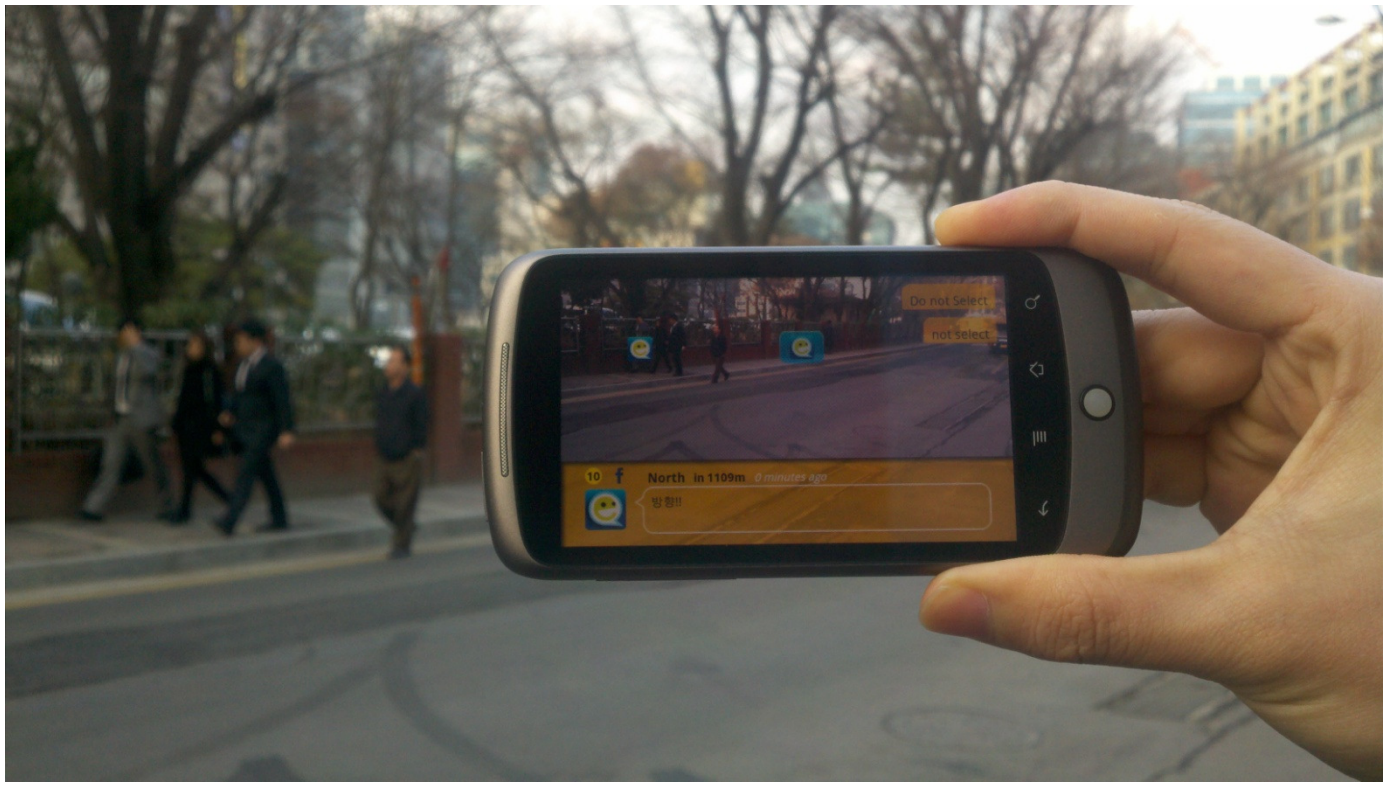
Pseudo- AR

- 가짜 AR ??
  - Location도 하나의 특징
  - Real AR ? Pseudo- AR? 은 소모적인 논쟁
  - 헤게모니.
  - 특징에 대한 인식



Location- based AR

- 실제와 가상의 혼합
  - Augmented Reality



Location- based AR in Android

- 필요 조건

Android Camera

SurfaceView

Camera

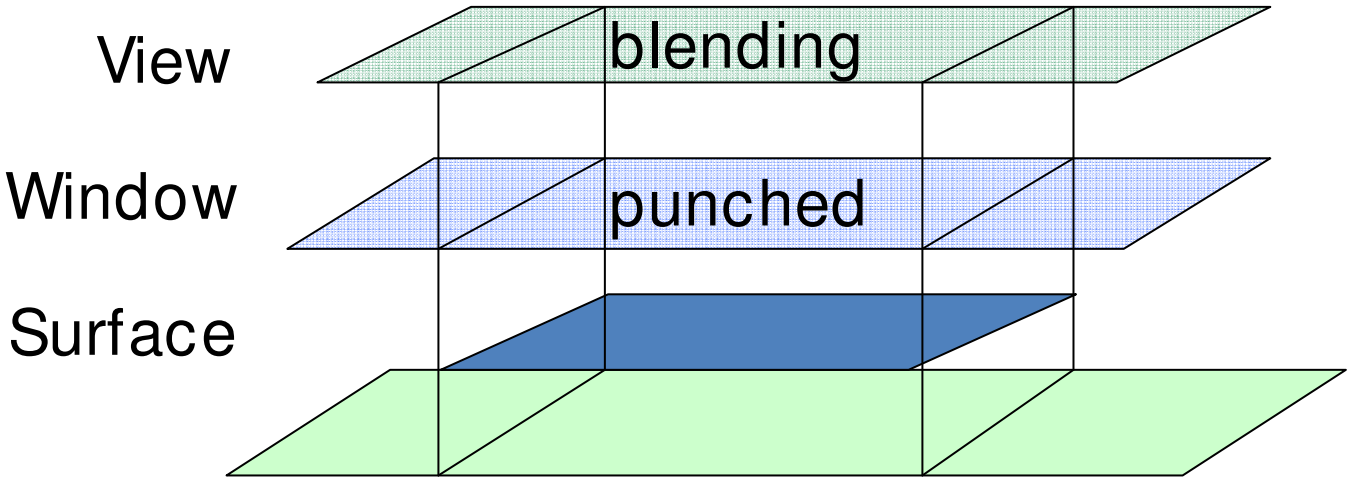
- 권한 설정
  - <uses-permission android:name="android.permission.CAMERA" />

- SurfaceView for Camera Capture Frame

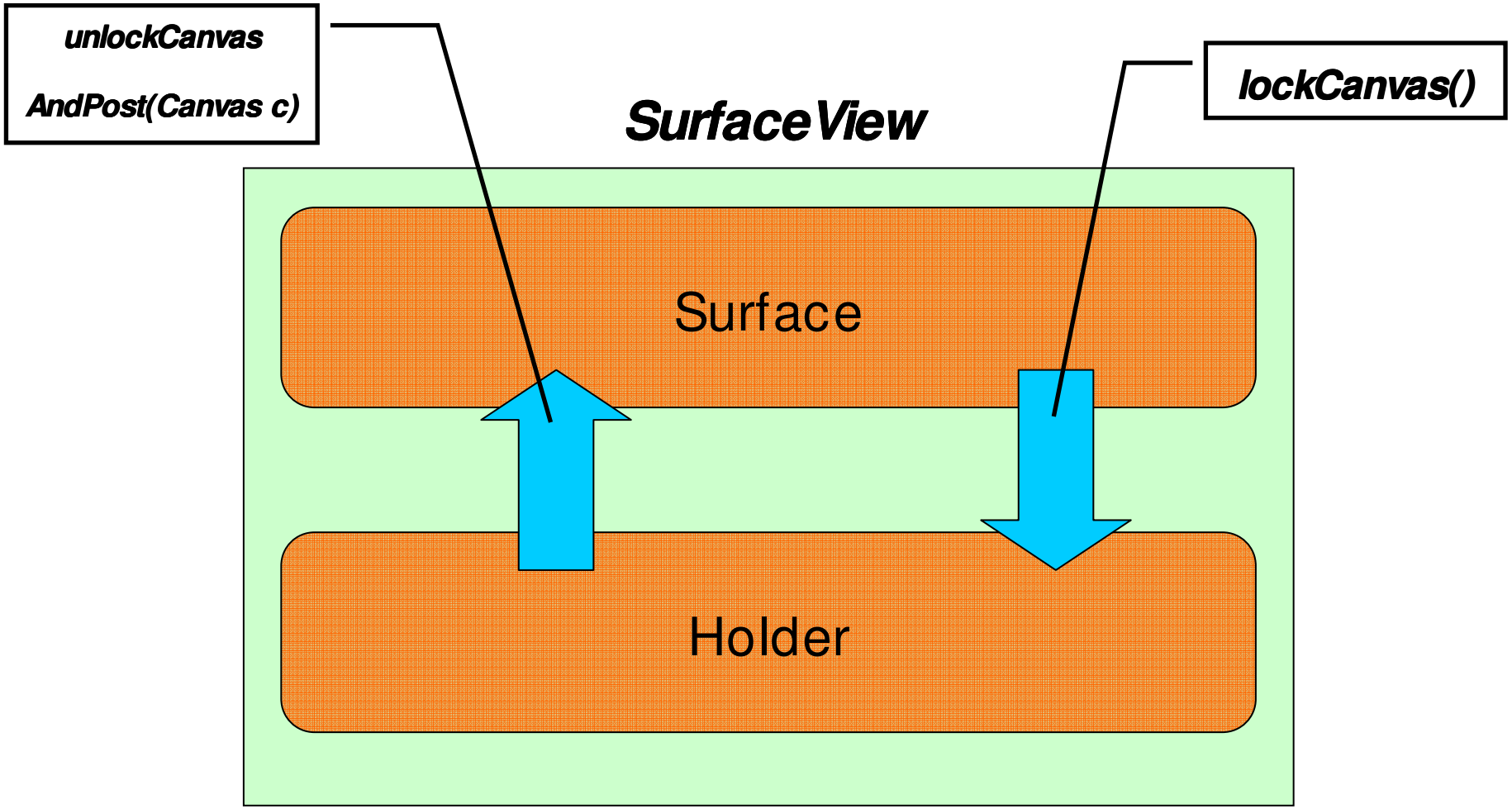
```
public void onCreate(Bundle savedInstanceState) {
        try{
                super.onCreate(savedInstanceState);
                cv = new CustomCameraView( this.getApplicationContext());
                FrameLayout fl = new FrameLayout( this.getApplicationContext());
                setContentView(fl);
                rl.addView(cv);
        }
        catch(Exception e){}
}
```

Camera

- What is SurfaceView?
  - UI Thread가 아닌 다른 Thread에서 UI 작업 : 성능!
  - The surface is Z ordered so that it is behind the window holding its SurfaceView
  - the SurfaceView punches a hole in its window to allow its surface to be displayed.



SurfaceView



SurfaceView

- CallBack 메커니즘
  - surfaceCreated
    - Surface 생성시 호출.
  - surfaceChanged
    - Surface 변경시 호출(화면 크기...)
  - surfaceDestroyed
    - Surface 제거시 호출, 할당된 자원 반납.
- Holder를 통한 관리
  - getHolder();

SurfaceView

```
Camera camera;
SurfaceHolder previewHolder;

public CustomCameraView(Context ctx)
{
  super(ctx);
  previewHolder = this.getHolder();
  previewHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
  previewHolder.addCallback(surfaceHolderListener);
}
```

Surface Create

```
SurfaceHolder.Callback surfaceHolderListener = new
SurfaceHolder.Callback() {
        public void surfaceCreated(SurfaceHolder holder) {
            camera=Camera.open();
            try
            {
              camera.setPreviewDisplay(previewHolder);
            }
            catch (Throwable t) {

            }
        }
```

Surface Change

```
public void surfaceChanged(SurfaceHolder
surfaceHolder, int format, int w, int h)
{
   Parameters params = camera.getParameters();
   params.setPreviewSize(320, 480);
   params.setPictureFormat(PixelFormat.JPEG);
   camera.setParameters(params);
   camera.startPreview();
}
```

RhaRham Interactive Inc.

Surface Destroy

```
public void surfaceDestroyed(SurfaceHolder arg0)
{
    // TODO Auto-generated method stub
    camera.stopPreview();
    camera.release();
}
```

Android Location

GPS Provider
Network Provider

Location

- 현재 위치의 자세한 정보가 필요해!!

- 권한 설정
  - <uses- permission android:name="android.permission.LOCATION"/>
  - <uses- permission android:name="android.permission.ACCESS_FINE_LOCATION" />
  - <uses- permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

Location

- Location Manager

```
locMan =
(LocationManager)ctx.getSystemService(Context.
LOCATION_SERVICE);
```

```
locMan.requestLocationUpdates(LocationManager.
GPS_PROVIDER, 100, 1, this);
```

public void ***requestLocationUpdates*** (String provider, long minTime, float minDistance, LocationListener listener)

Providers

- GPS Provider
  - ACCESS_FINE_LOCATION Permission.
  - Cold Start : 2분~ 15분 소요.
  - 환경적 요인에 민감.

- Network Provider
  - ACCESS_COARSE_LOCATION Permission.
  - 위치확인 가능한 Network 연결 시 바로 작동
  - 오차범위가 크다.

Location

```
public void onLocationChanged(Location location)
{
      if(curLocation == null) {
         curLocation = location;
         ARView.Location = location;
         locationChanged = true;
      }
      else if(curLocation.getLatitude() == location.getLatitude() &&
              curLocation.getLongitude() == location.getLongitude()){
          locationChanged = false;
      }
      else{
          locationChanged = true;
      }
      curLocation = location;
      postInvalidate();
}
```

Android Compass

Sensor

SensorEvent, SensorManager

Compass

넌 대체 누굴 보고 있는 거야?

- 시스템 서비스 획득

```
sensorMan =
(SensorManager)ctx.getSystemService(Context.SENSOR_SERVICE);

sensorMan.registerListener
(this, sensorMan.getDefaultSensor(Sensor.TYPE_ORIENTATION),
SensorManager.SENSOR_DELAY_FASTEST);

sensorMan.registerListener
(this, sensorMan.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
 SensorManager.SENSOR_DELAY_FASTEST);
```

Sensor List

- Sensor Class
  - TYPE_ACCELEROMETER
    - A constant describing an accelerometer sensor type.
  - TYPE_GYROSCOPE
    - A constant describing a gyroscope sensor type
  - TYPE_LIGHT
    - A constant describing an light sensor type.
  - TYPE_MAGNETIC_FIELD
    - A constant describing a magnetic field sensor type.
  - TYPE_ORIENTATION
    - deprecated. use SensorManager.getOrientation() instead.
  - TYPE_PRESSURE
    - A constant describing a pressure sensor type
  - TYPE_PROXIMITY
    - A constant describing an proximity sensor type.
  - TYPE_TEMPERATURE
    - A constant describing a temperature sensor type

public final float[] **values** Since: API Level 3

The length and contents of the values array vary depending on which sensor type is being monitored (see also SensorEvent for a definition of the coordinate system used):

Sensor.TYPE_ORIENTATION:

All values are angles in degrees.

values[0]: Azimuth, angle between the magnetic north direction and the Y axis, around the Z axis (0 to 359). 0=North, 90=East, 180=South, 270=West

values[1]: Pitch, rotation around X axis (-180 to 180), with positive values when the z-axis moves **toward** the y-axis.

values[2]: Roll, rotation around Y axis (-90 to 90), with positive values when the x-axis moves **away** from the z-axis.

Note: This definition is different from **yaw, pitch and roll** used in aviation where the X axis is along the long side of the plane (tail to nose).

Note: It is preferable to use getRotationMatrix() in conjunction with remapCoordinateSystem() and getOrientation() to compute these values; while it may be more expensive, it is usually more accurate.

Sensor.TYPE_ACCELEROMETER:

All values are in SI units (m/s^2) and measure the acceleration applied to the phone minus the force of gravity.

values[0]: Acceleration minus Gx on the x-axis

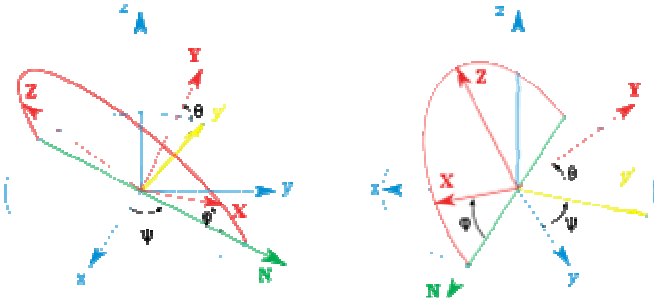
values[1]: Acceleration minus Gy on the y-axis

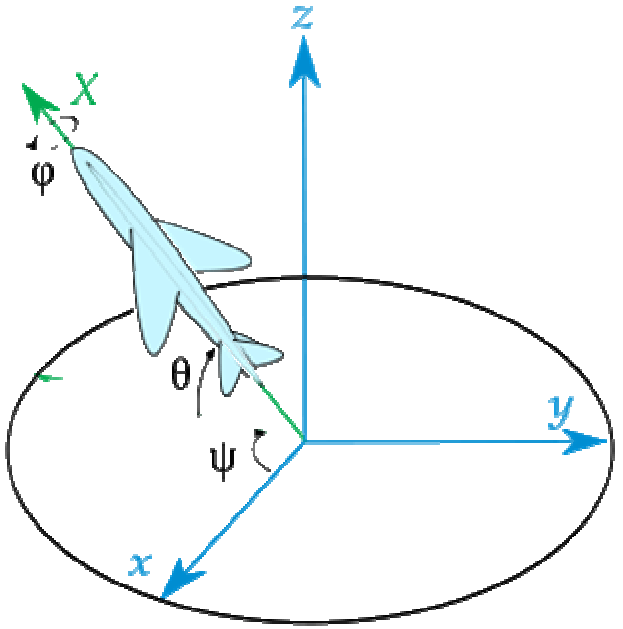
values[2]: Acceleration minus Gz on the z-axis

Examples:

- When the device lies flat on a table and is pushed on its left side toward the right, the x acceleration value is positive.
- When the device lies flat on a table, the acceleration value is +9.81, which correspond to the acceleration of the device (0 m/s^2) minus the force of gravity (-9.81 m/s^2).
- When the device lies flat on a table and is pushed toward the sky with an acceleration of A m/s^2, the acceleration value is equal to A+9.81 which correspond to the acceleration of the device (+A m/s^2) minus the force of gravity (-9.81 m/s^2).

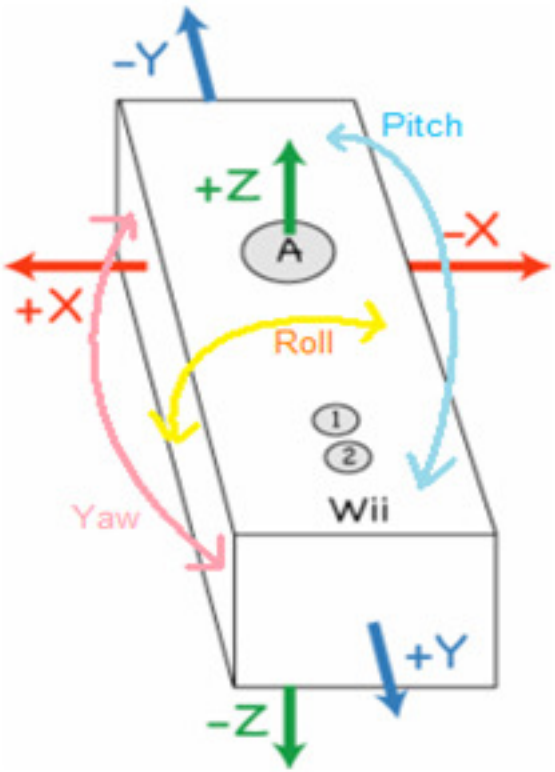
- Yaw,Pitch and Roll

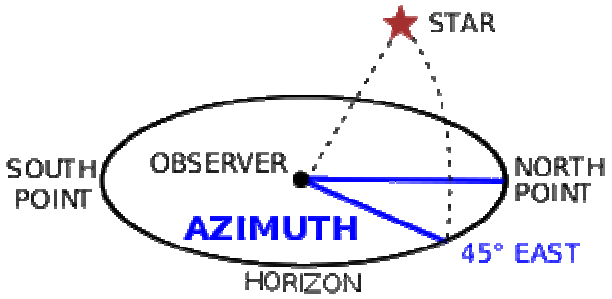




Sensor

- Azimuth,Pitch and Roll





Sensor

- Device Orientation
  - Orientation Sensor directly – Legacy
  - Accelerometers + Magnetic field
- According to Android Doc
  - SLOW…
  - But increased accuracy, ability to modify

Note: It is preferable to use `getRotationMatrix()` in conjunction with `remapCoordinateSystem()` and `getOrientation()` to compute these values; while it may be more expensive, it is usually more accurate.

Sensor - Legacy

```
   public static volatile float direction = (float) 0;
   public static volatile float inclination;
   public static volatile float rollingZ = (float)0;


   public static volatile float kFilteringFactor =
(float)0.05;
   public static float aboveOrBelow = (float)0;


   public void onAccuracyChanged(Sensor arg0, int
arg1){}


   public void onSensorChanged(SensorEvent evt)
   {
      float vals[] = evt.values;


      if(evt.sensor.getType() ==
Sensor.TYPE_ORIENTATION)
      {
         float rawDirection = vals[0];


         direction =(float) ((rawDirection *
kFilteringFactor) +
            (direction * (1.0 - kFilteringFactor)));
```

```
inclination =
         (float) ((vals[2] * kFilteringFactor) +
         (inclination * (1.0 - kFilteringFactor)));

if(aboveOrBelow > 0)
          inclination = inclination * - 1;


       if(evt.sensor.getType() ==
Sensor.TYPE_ACCELEROMETER)
       {
          aboveOrBelow =
            (float) ((vals[2] * kFilteringFactor) +
            (aboveOrBelow * (1.0 -
kFilteringFactor)));
       }
    }
  }
```

Sensor - conjunction

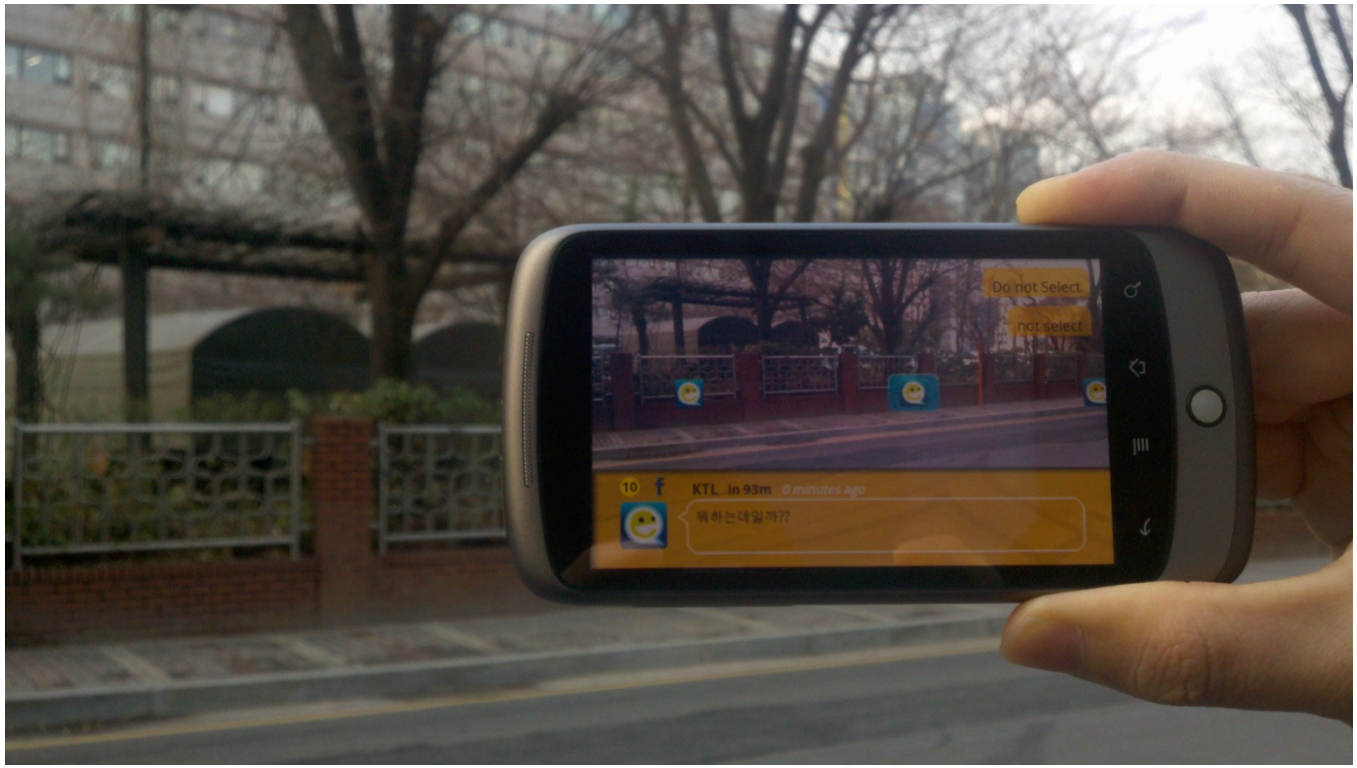
```
float[] values = new float[3];
float[] R = new float[9];

SensorManager.getRotationMatrix(R,
null,accelerometerValues,magneticFieldValues);

SensorManager.getOrientation(R, values);

// Convert from radians to degrees.
values[0] = (float) Math.toDegrees(values[0]);
values[1] = (float) Math.toDegrees(values[1]);
values[2] = (float) Math.toDegrees(values[2]);
```

Sensor

- `SensorManager.getRotationMatrix(R, null, aValues, mValues);`
- `float[] outR = new float[9];`
- `SensorManager.remapCoordinateSystem(R,SensorManager.AXIS_X,SensorManager.AXIS_Z, outR);`
- `SensorManager.getOrientation(outR, values);`
- `// Convert from radians to degrees.`
- `values[0] = (float) Math.toDegrees(values[0]);`
- `values[1] = (float) Math.toDegrees(values[1]);`
- `values[2] = (float) Math.toDegrees(values[2]);`

RhaRham
Interactive Inc.

Location- based AR in Android



AR OpenSource

- ***NyARToolkit***
  - http://nyatla.jp/nyartoolkit/wiki/index.php?FrontPage.en
- ***AndAR***
  - http://code.google.com/p/andar/
- ***mixare***
  - http://www.mixare.org/

참고자료

- 안드로이드 펍 – www.androidpub.com
- Professional Android2 – Reto meier, Wrox
- Wikipedia
  - Android
  - AR
  - Wii Remote
  - Azimuth,Pitch,Roll

Thanks!!

박유태
㈜라람인터랙티브