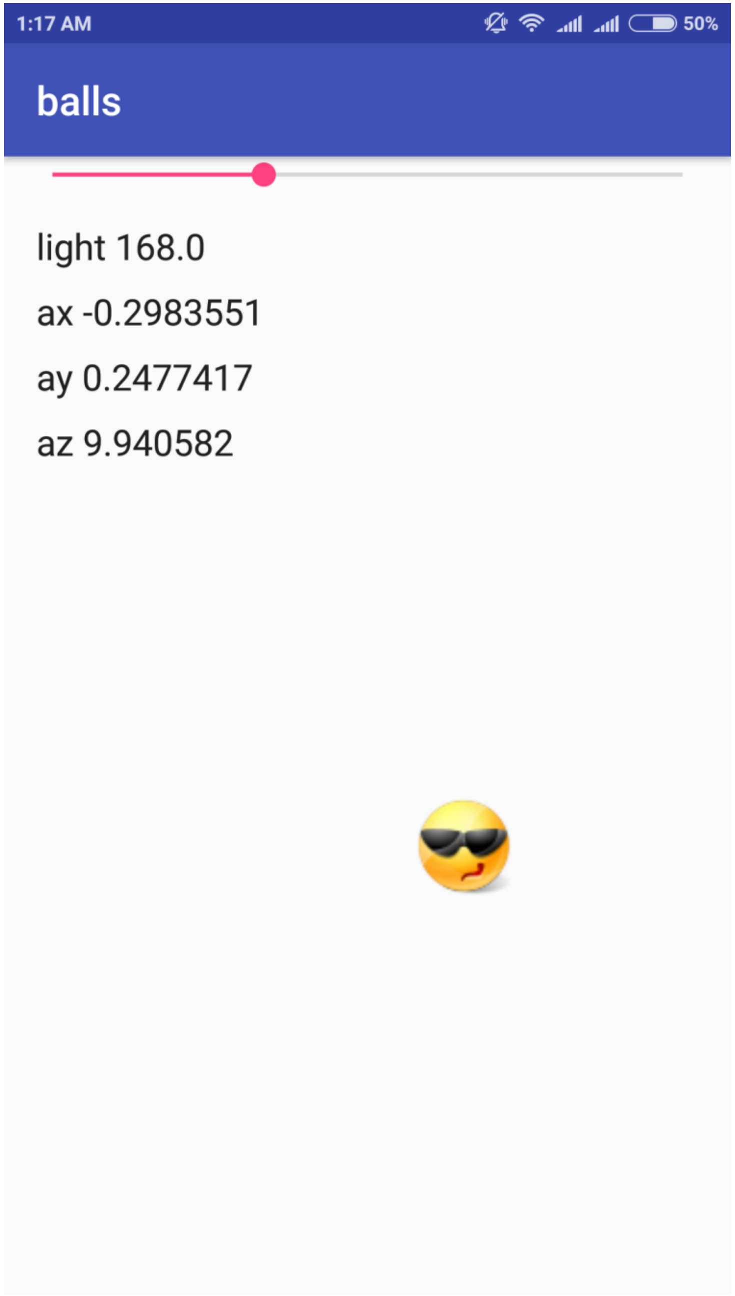
Mobile Programming

Lec 7. Using Sensors

Ewha Womans University

Layout

- Four Texts
  - show light level
  - show acceleration
- One images
  - one will show acceleration (It will roll later)
- One SeekBar
  - will indicates light level



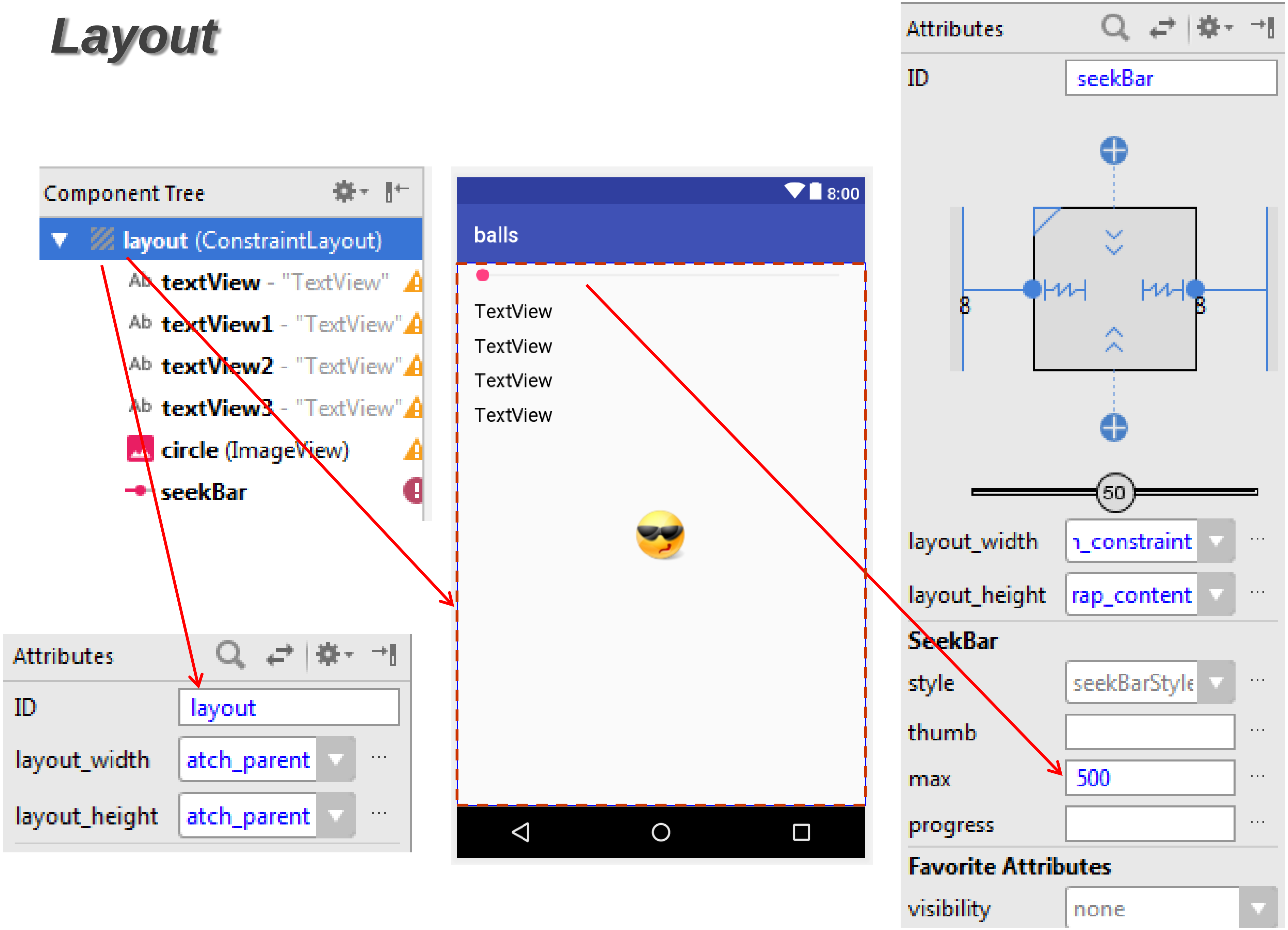
light 168.0

ax -0.2983551

ay 0.2477417

az 9.940582

Layout



Component Tree

- layout (ConstraintLayout)
  - textView - "TextView"
  - textView1 - "TextView"
  - textView2 - "TextView"
  - textView3 - "TextView"
  - circle (ImageView)
  - seekBar

Attributes

ID: layout

layout_width: atch_parent

layout_height: atch_parent

Attributes

ID: seekBar

layout_width: ı_constraint

layout_height: rap_content

SeekBar

style: seekBarStyle

thumb:

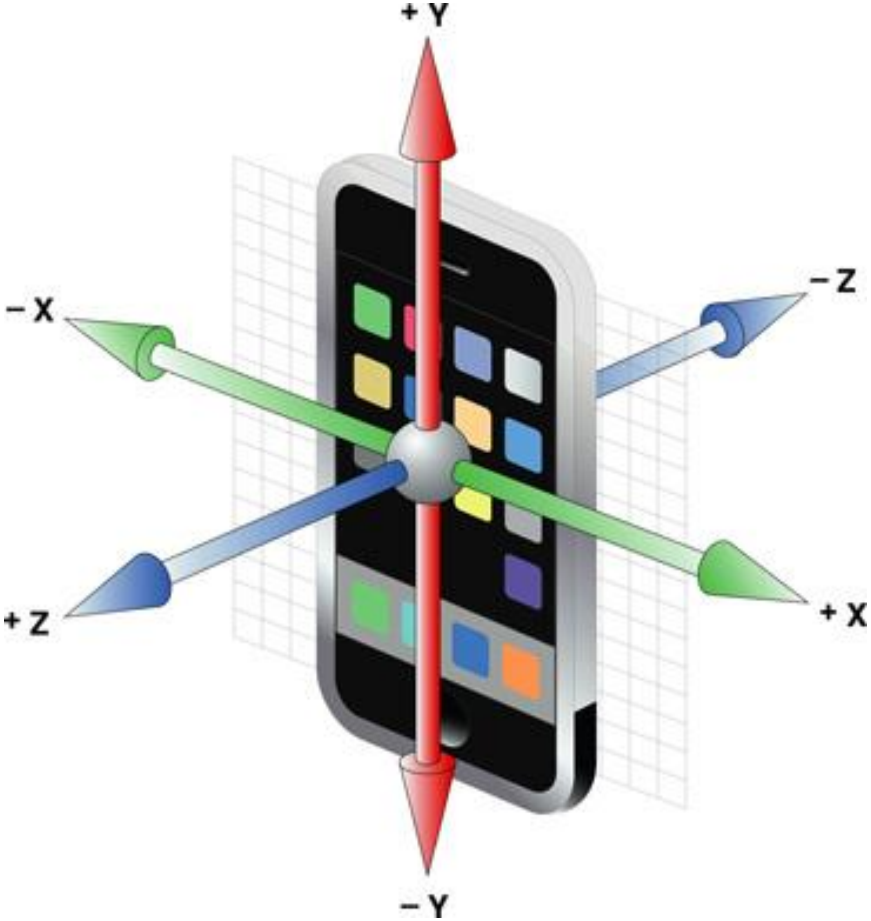
max: 500

progress:

Favorite Attributes

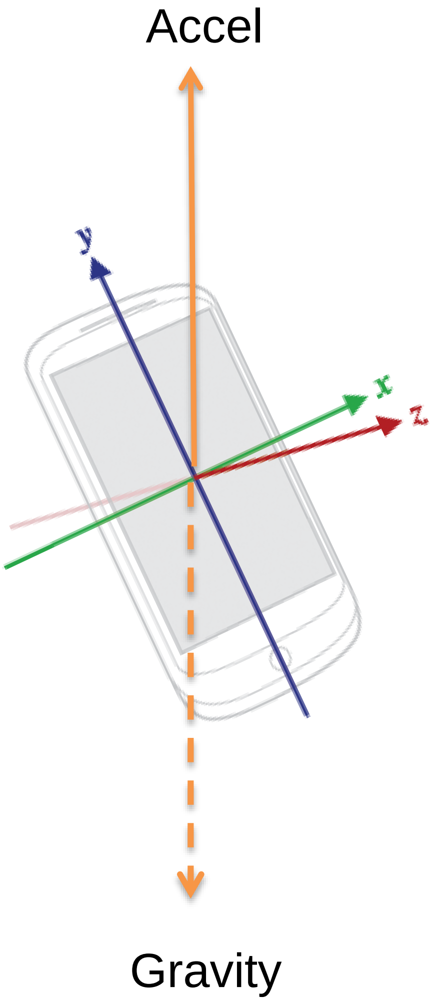
visibility: none

Accelerometer



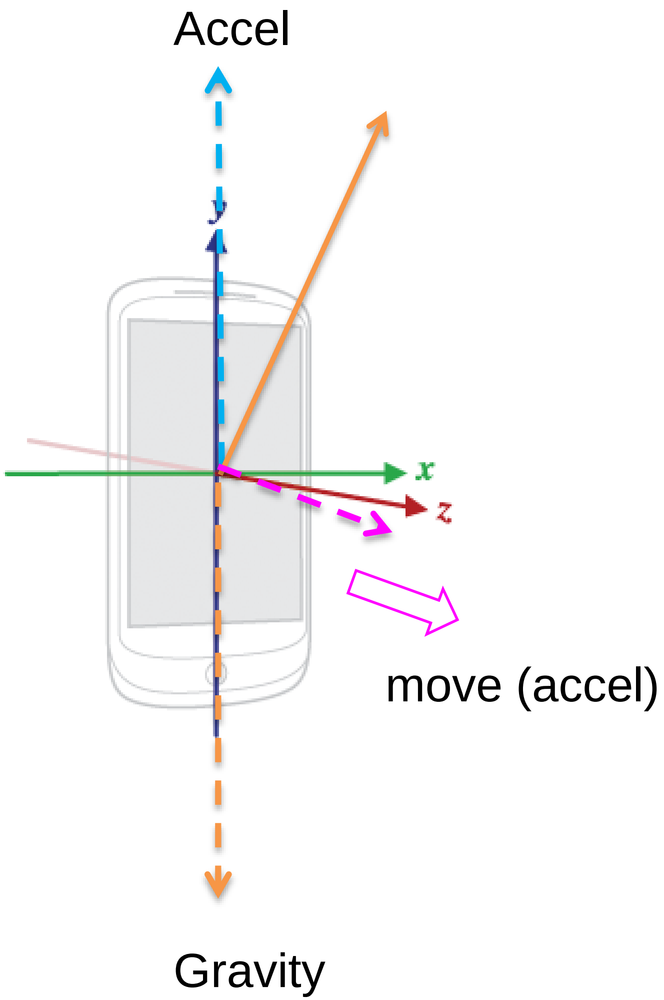
Accelerometer

- Gravity is acceleration too.   Indistinguishable.



Accel

Gravity

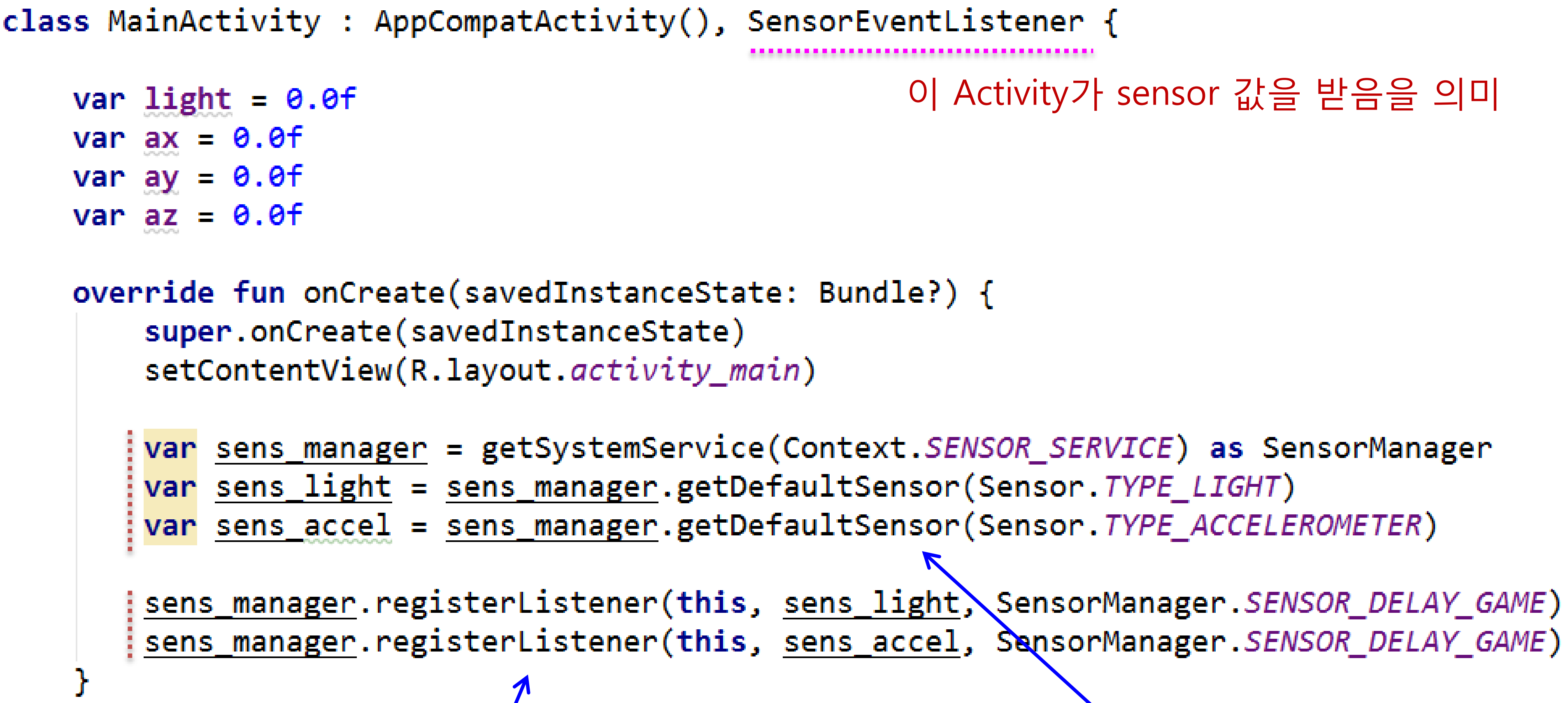


Accel

move (accel)

Gravity

(1) Deliver sensor values to Activity

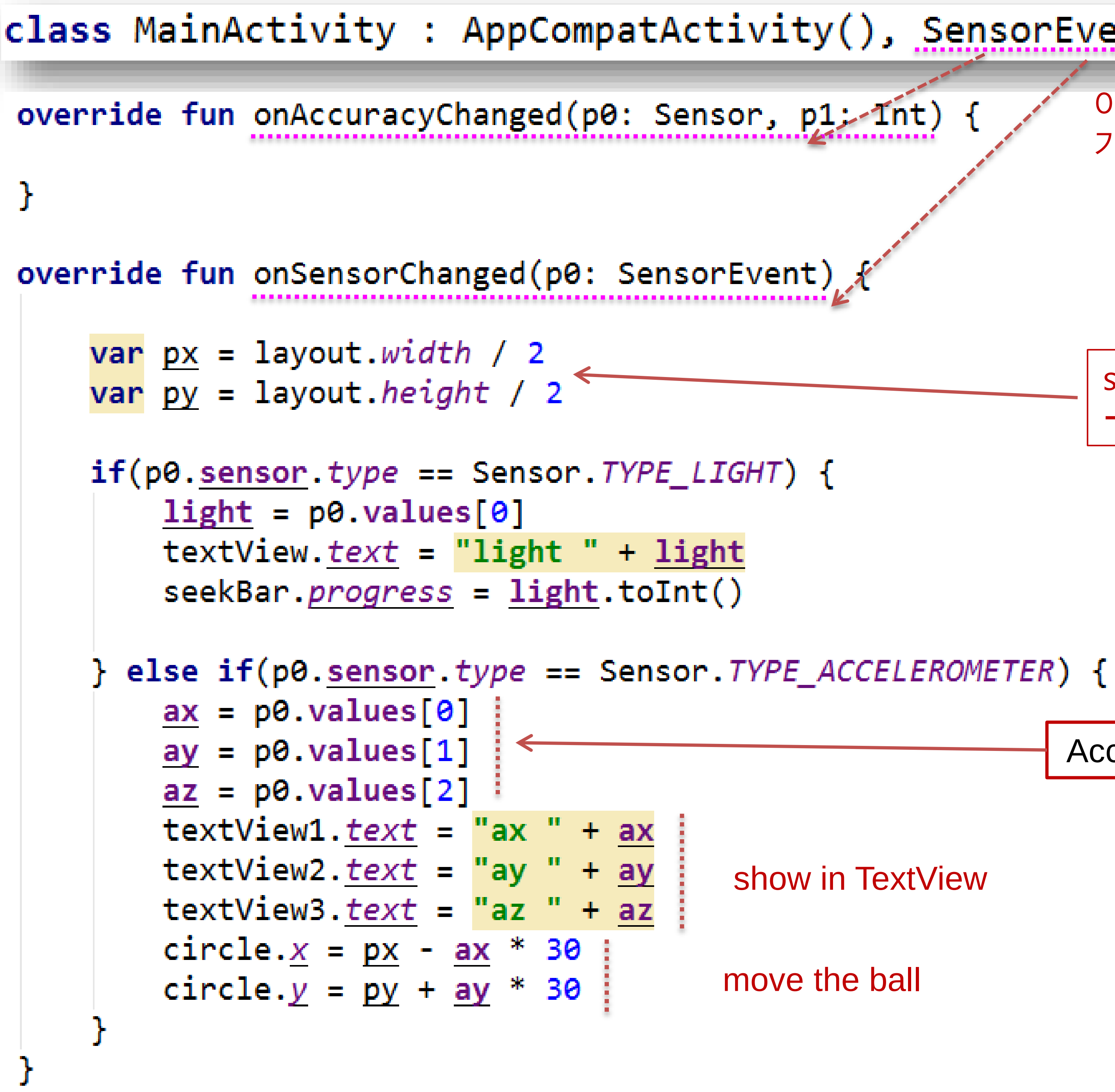


```
class MainActivity : AppCompatActivity(), SensorEventListener {

    var light = 0.0f
    var ax = 0.0f
    var ay = 0.0f
    var az = 0.0f

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        var sens_manager = getSystemService(Context.SENSOR_SERVICE) as SensorManager
        var sens_light = sens_manager.getDefaultSensor(Sensor.TYPE_LIGHT)
        var sens_accel = sens_manager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER)

        sens_manager.registerListener(this, sens_light, SensorManager.SENSOR_DELAY_GAME)
        sens_manager.registerListener(this, sens_accel, SensorManager.SENSOR_DELAY_GAME)
    }
```

이 Activity가 sensor 값을 받음을 의미

Start to read sensor

Get sensor manager,
Get light sensor
Get acceleration sensor
(not working yet)



```
class MainActivity : AppCompatActivity(), SensorEventListener {
override fun onAccuracyChanged(p0: Sensor, p1: Int) {

}

override fun onSensorChanged(p0: SensorEvent) {

    var px = layout.width / 2
    var py = layout.height / 2

    if(p0.sensor.type == Sensor.TYPE_LIGHT) {
        light = p0.values[0]
        textView.text = "light " + light
        seekBar.progress = light.toInt()

    } else if(p0.sensor.type == Sensor.TYPE_ACCELEROMETER) {
        ax = p0.values[0]
        ay = p0.values[1]
        az = p0.values[2]
        textView1.text = "ax " + ax
        textView2.text = "ay " + ay
        textView3.text = "az " + az
        circle.x = px - ax * 30
        circle.y = py + ay * 30
    }
}
```

아래 2개의 functions을 가짐을 의미

size of screen(layout)
→ center

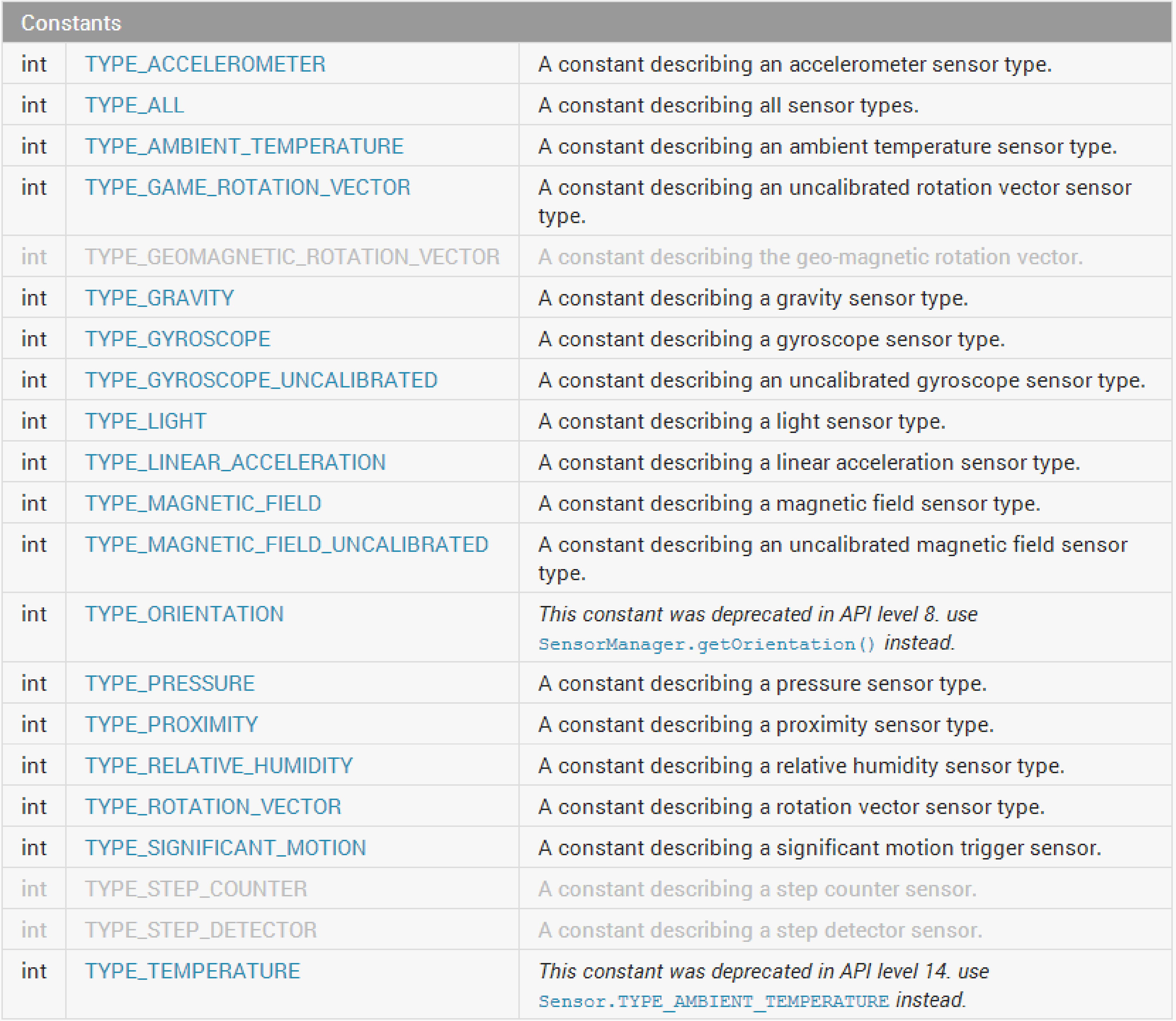
Acceleration in m/sec^2

show in TextView

move the ball

Sensor Delay

- SENSOR_DELAY_NORMAL
  - a delay of 200,000 microseconds.
- SENSOR_DELAY_GAME
  - 20,000 microsecond delay
- SENSOR_DELAY_UI
  - 60,000 microsecond delay
- SENSOR_DELAY_FASTEST
- 0 microsecond delay
- Or, specify the delay as an absolute value (in microseconds).

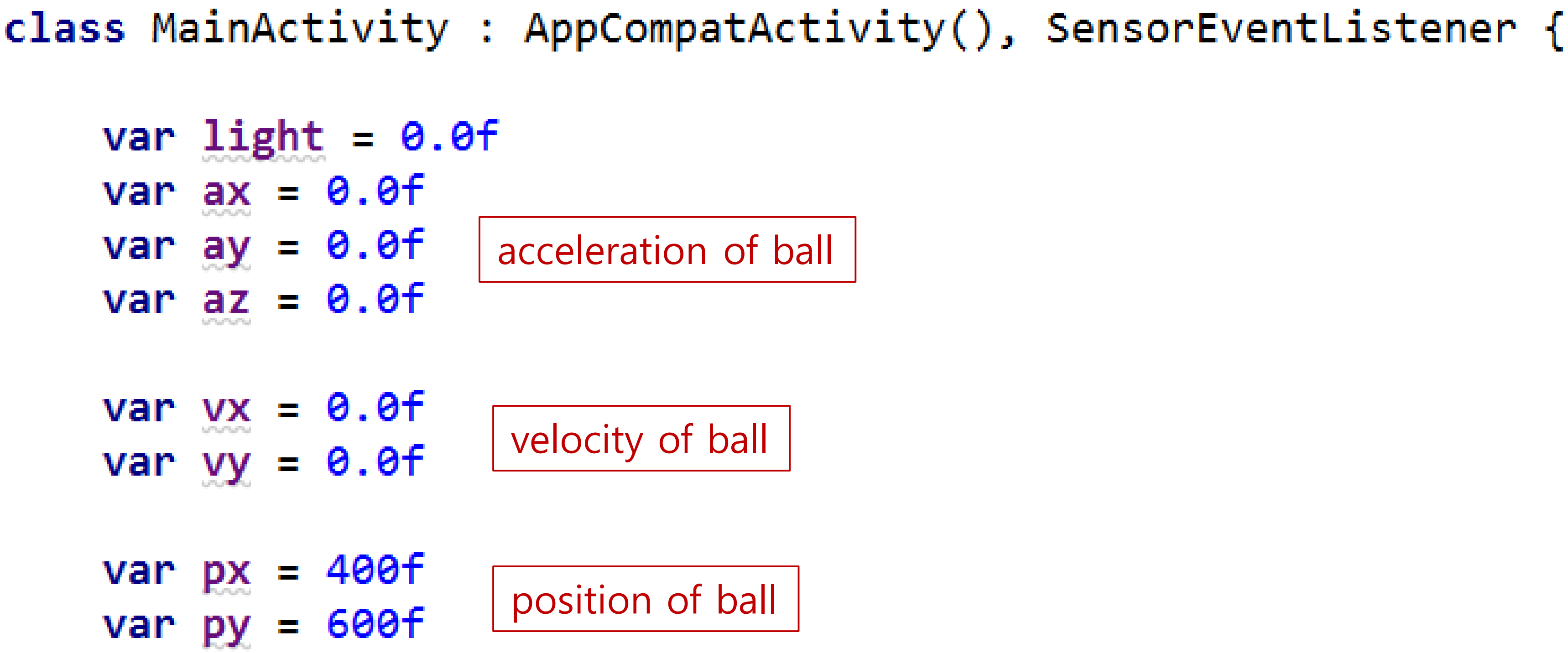


Constants		
int	TYPE_ACCELEROMETER	A constant describing an accelerometer sensor type.
int	TYPE_ALL	A constant describing all sensor types.
int	TYPE_AMBIENT_TEMPERATURE	A constant describing an ambient temperature sensor type.
int	TYPE_GAME_ROTATION_VECTOR	A constant describing an uncalibrated rotation vector sensor type.
int	TYPE_GEOMAGNETIC_ROTATION_VECTOR	A constant describing the geo-magnetic rotation vector.
int	TYPE_GRAVITY	A constant describing a gravity sensor type.
int	TYPE_GYROSCOPE	A constant describing a gyroscope sensor type.
int	TYPE_GYROSCOPE_UNCALIBRATED	A constant describing an uncalibrated gyroscope sensor type.
int	TYPE_LIGHT	A constant describing a light sensor type.
int	TYPE_LINEAR_ACCELERATION	A constant describing a linear acceleration sensor type.
int	TYPE_MAGNETIC_FIELD	A constant describing a magnetic field sensor type.
int	TYPE_MAGNETIC_FIELD_UNCALIBRATED	A constant describing an uncalibrated magnetic field sensor type.
int	TYPE_ORIENTATION	*This constant was deprecated in API level 8. use* `SensorManager.getOrientation()` *instead.*
int	TYPE_PRESSURE	A constant describing a pressure sensor type.
int	TYPE_PROXIMITY	A constant describing a proximity sensor type.
int	TYPE_RELATIVE_HUMIDITY	A constant describing a relative humidity sensor type.
int	TYPE_ROTATION_VECTOR	A constant describing a rotation vector sensor type.
int	TYPE_SIGNIFICANT_MOTION	A constant describing a significant motion trigger sensor.
int	TYPE_STEP_COUNTER	A constant describing a step counter sensor.
int	TYPE_STEP_DETECTOR	A constant describing a step detector sensor.
int	TYPE_TEMPERATURE	*This constant was deprecated in API level 14. use* `Sensor.TYPE_AMBIENT_TEMPERATURE` *instead.*

More about sensors

- Detail Information on Various Sensor values
  - http://developer.android.com/reference/android/hardware/SensorEvent.html#values

(2) Roll the ball

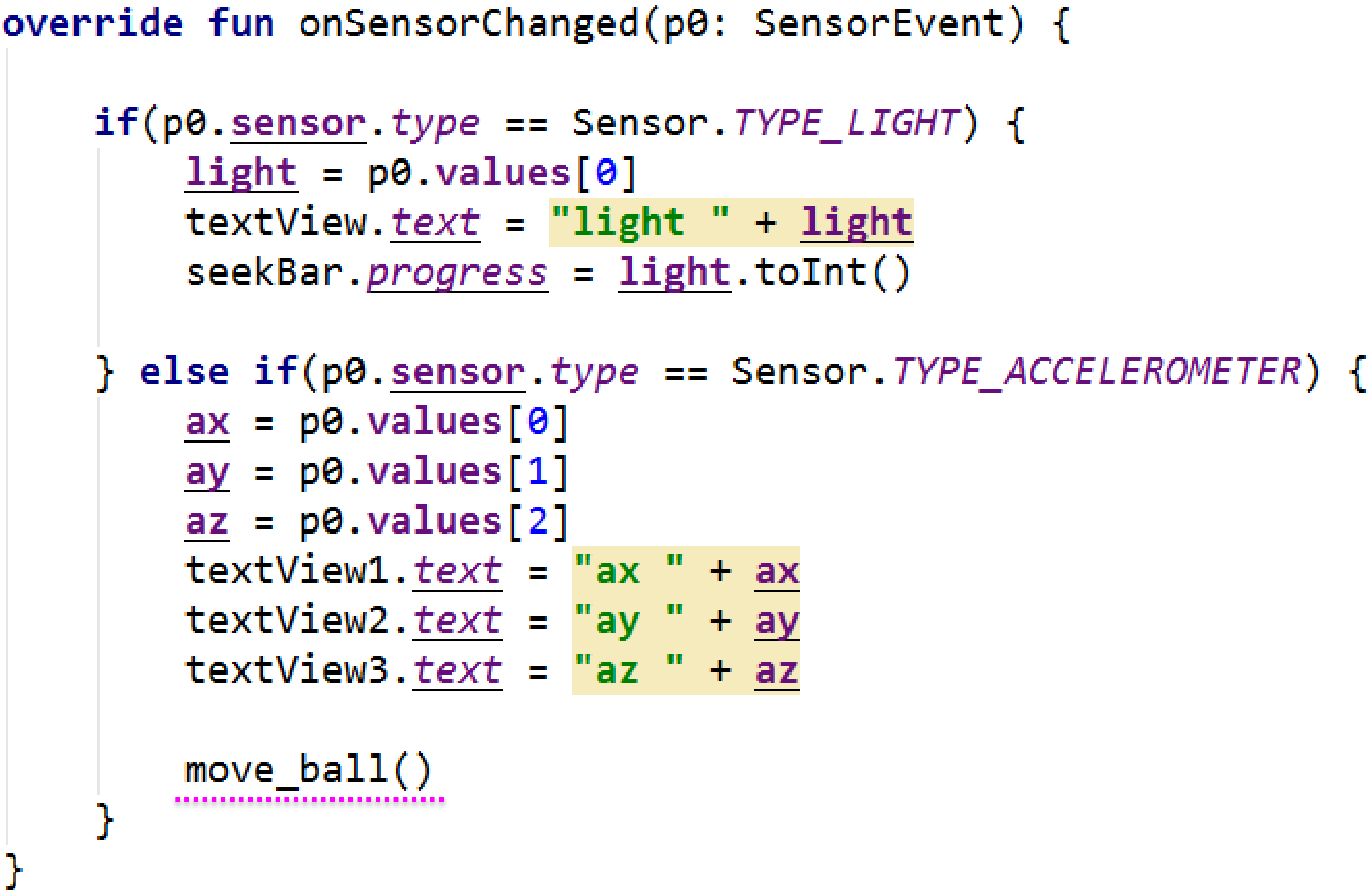


```
class MainActivity : AppCompatActivity(), SensorEventListener {

    var light = 0.0f
    var ax = 0.0f
    var ay = 0.0f
    var az = 0.0f

    var vx = 0.0f
    var vy = 0.0f

    var px = 400f
    var py = 600f
```

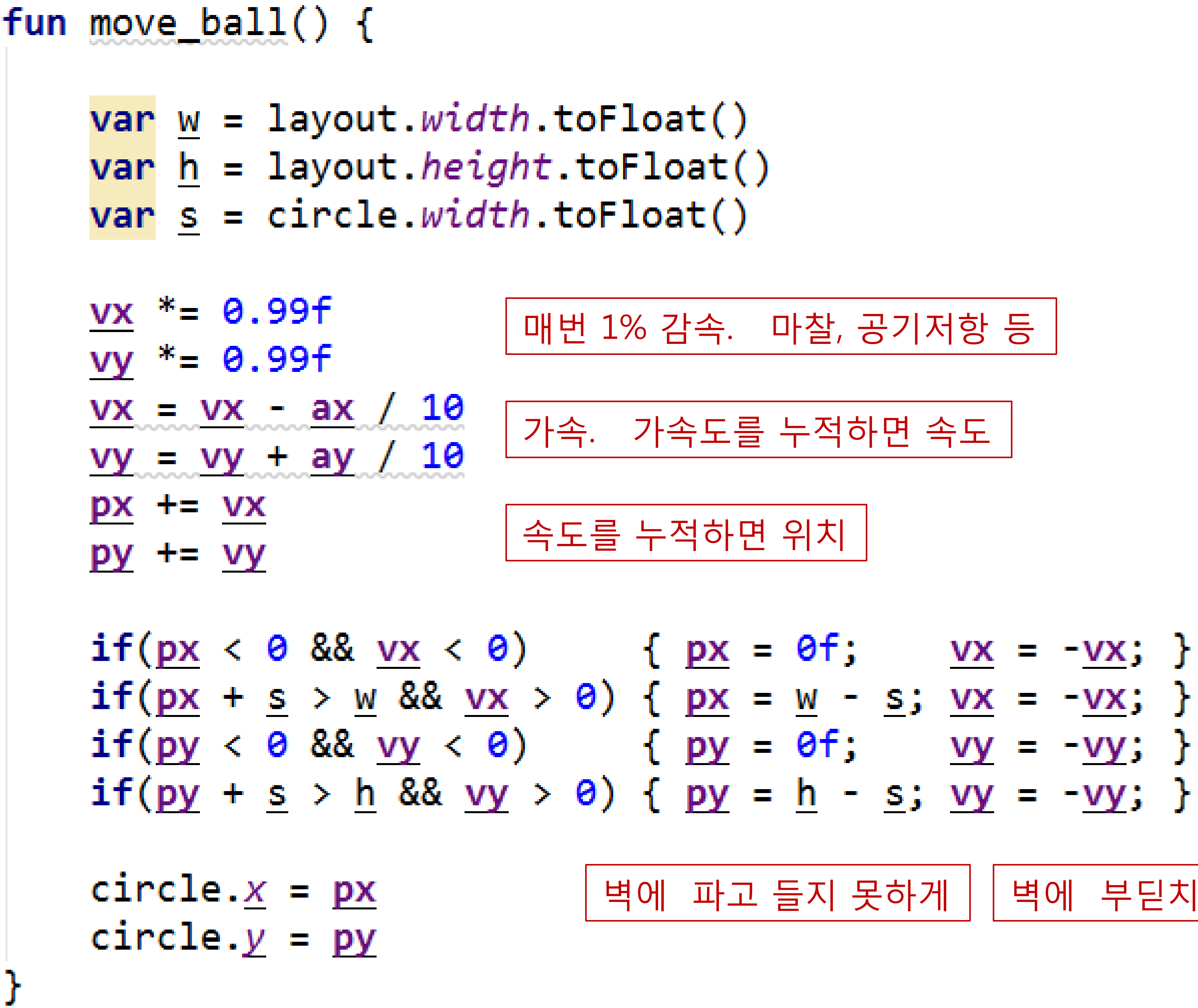
acceleration of ball

velocity of ball

position of ball



```
override fun onSensorChanged(p0: SensorEvent) {

    if(p0.sensor.type == Sensor.TYPE_LIGHT) {
        light = p0.values[0]
        textView.text = "light " + light
        seekBar.progress = light.toInt()

    } else if(p0.sensor.type == Sensor.TYPE_ACCELEROMETER) {
        ax = p0.values[0]
        ay = p0.values[1]
        az = p0.values[2]
        textView1.text = "ax " + ax
        textView2.text = "ay " + ay
        textView3.text = "az " + az

        move_ball()
    }
}
```



```
fun move_ball() {

    var w = layout.width.toFloat()
    var h = layout.height.toFloat()
    var s = circle.width.toFloat()

    vx *= 0.99f
    vy *= 0.99f
    vx = vx - ax / 10
    vy = vy + ay / 10
    px += vx
    py += vy

    if(px < 0 && vx < 0)     { px = 0f;    vx = -vx; }
    if(px + s > w && vx > 0) { px = w - s; vx = -vx; }
    if(py < 0 && vy < 0)     { py = 0f;    vy = -vy; }
    if(py + s > h && vy > 0) { py = h - s; vy = -vy; }

    circle.x = px
    circle.y = py
}
```

매번 1% 감속. 마찰, 공기저항 등

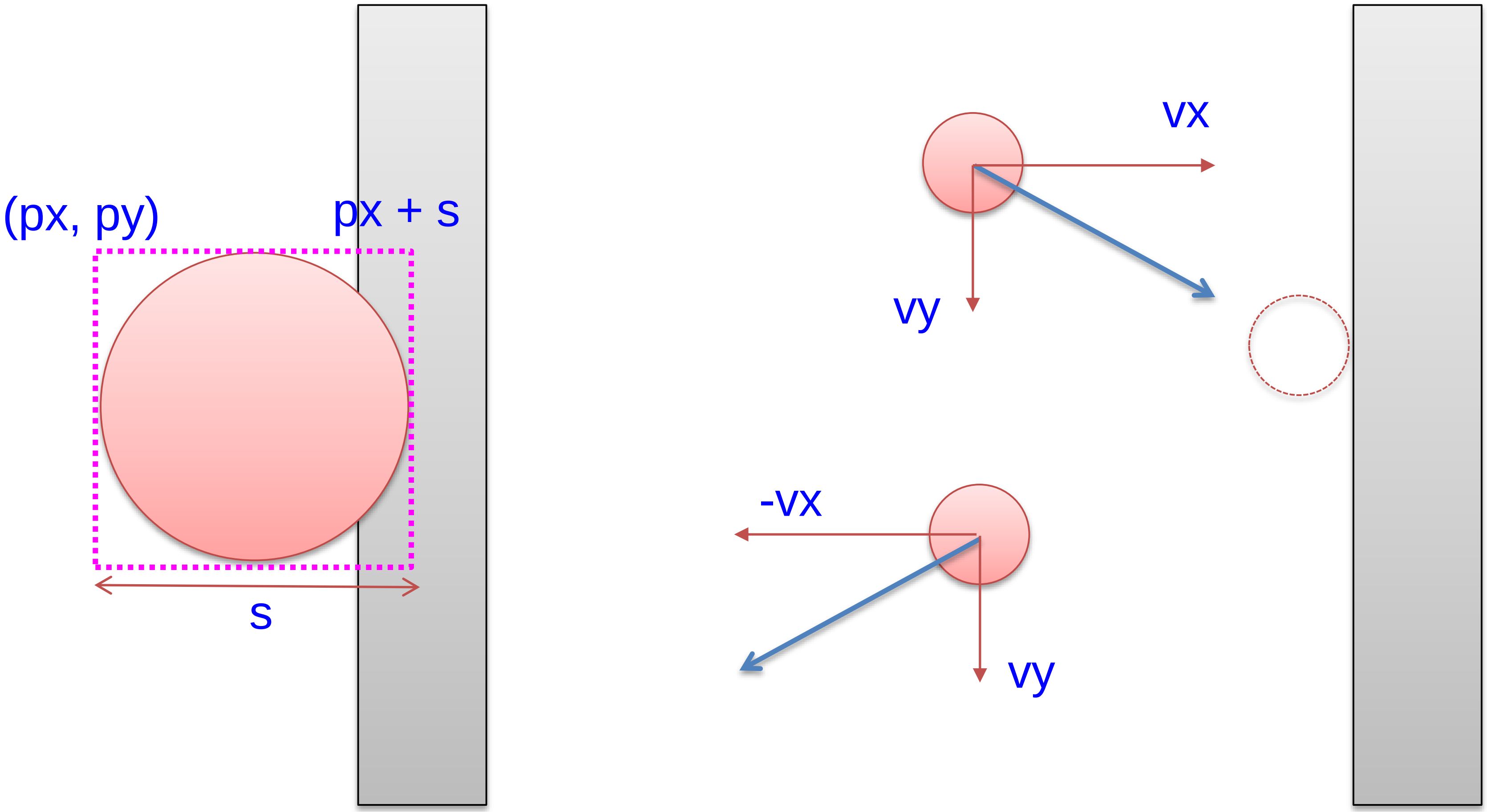
가속. 가속도를 누적하면 속도

속도를 누적하면 위치

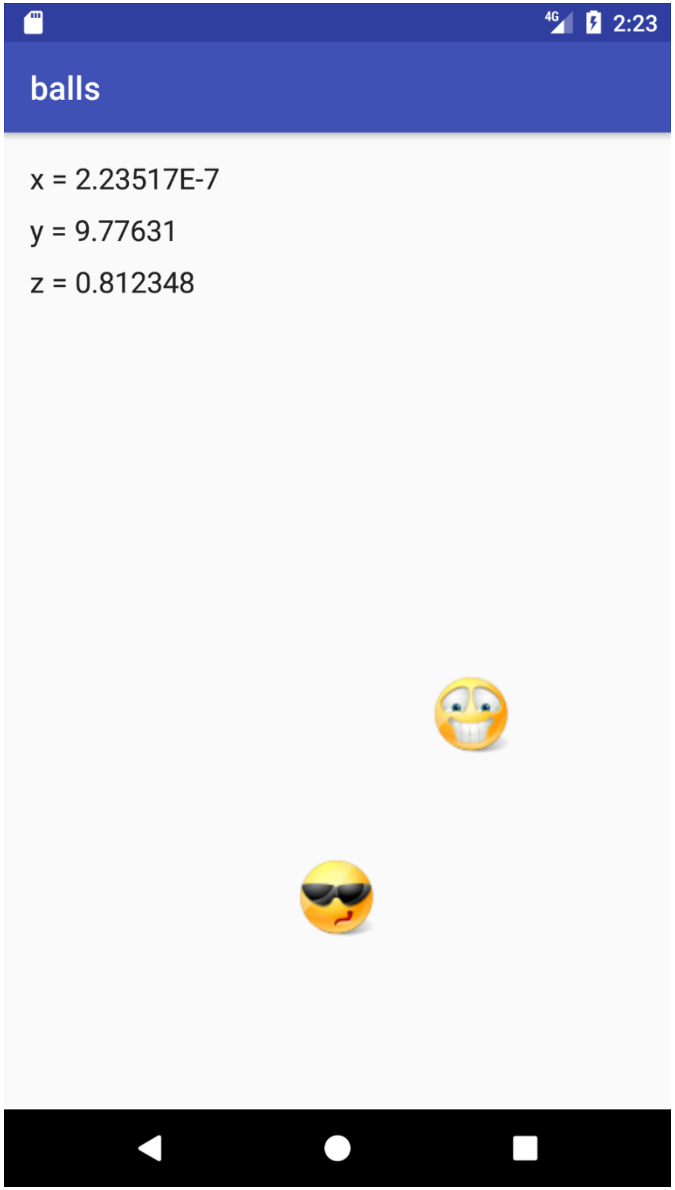
벽에 파고 들지 못하게

벽에 부딛치면 속도 반대

Checking collision. Velocity after.



(3) Two balls (homework)



```
x = 2.23517E-7
y = 9.77631
z = 0.812348
```