

헬로, 안드로이드

3주차 - 사용자 인터페이스 디자인하기 (1)

강대기

동서대학교 컴퓨터정보공학부

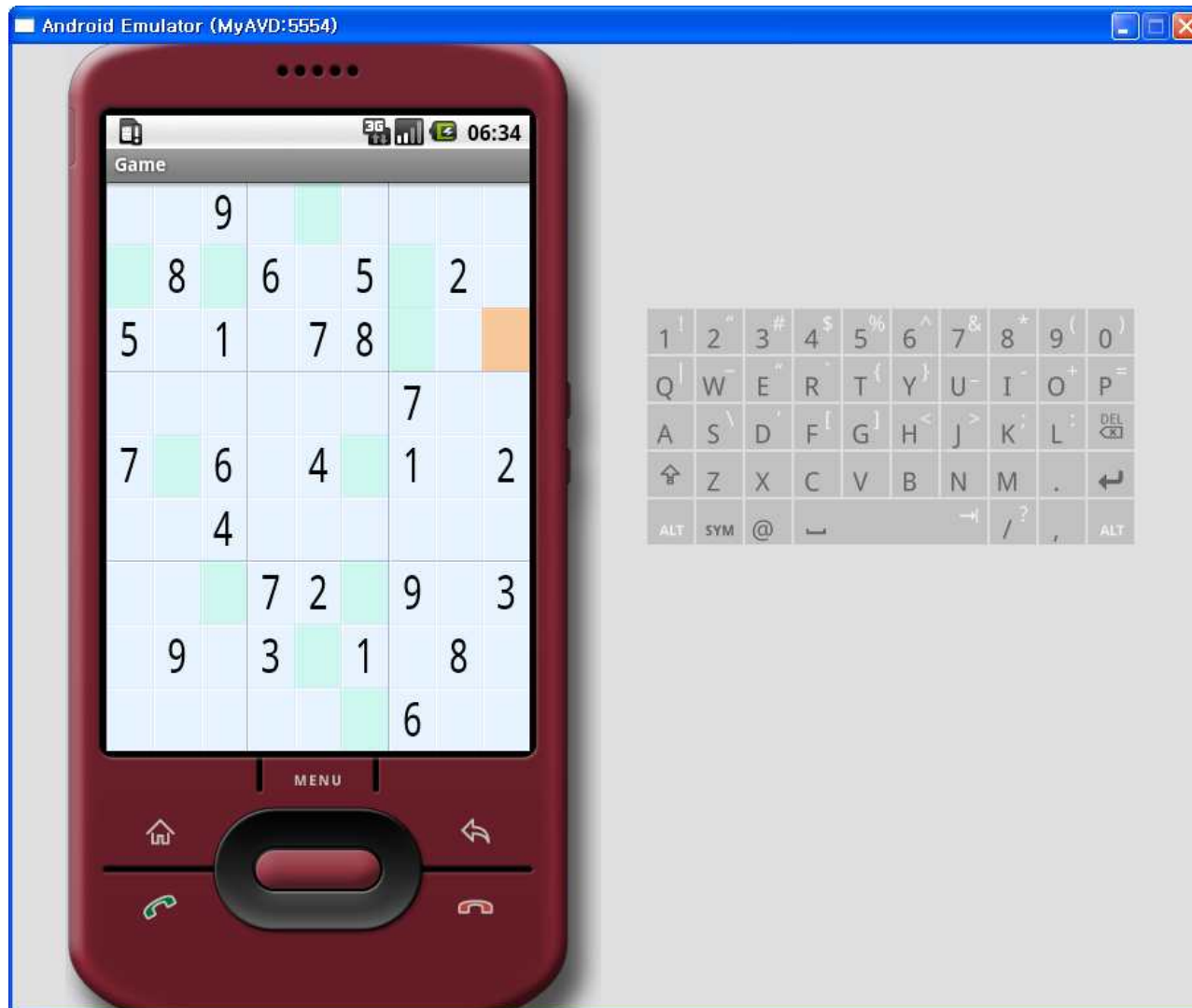
학습 목표

- 선언하여 디자인을 하는 방법을 이해하고, 실행할 수 있다.
- 시작화면을 만드는 방법과 대체 리소스를 사용하는 방법을 이해하고 실행할 수 있다.
- About 과 같은 상자를 구현하고, 테마를 적용하는 법을 이해하고 실행할 수 있다.

차례

- 스도쿠 예제 소개하기
- 선언하여 디자인하기
- 시작화면 만들기
- 대체 리소스 사용하기
- About 상자 구현하기
- 테마 적용하기
- 요약
- 퀴즈
- 연습문제

스도쿠 예제 소개하기



선언하여 디자인하기

- 사용자 인터페이스를 디자인하는 방법
 - 절차적 방법 – C나 Java 프로그램 코드로
 - 선언적 방법 – HTML 언어를 통해 표현하여
- 안드로이드의 경우, 둘 다 지원
 - 절차적 방법 – Java 코드
 - 선언적 방법 – XML 표현
- 추천하는 방법은 XML 표현

시작화면 만들기

- Program name – Sudoku
- Package name – org.example.sudoku
- Activity name – Sudoku
- Application name – Sudoku
- 안드로이드 에뮬레이터는 항상 열어놓음
- 게임의 오프닝 화면 구성
- 액티비티 – Sudoku.java
- 리소스 – R.java
- 레이아웃 – main.xml
 - ADT의 레이아웃 에디터는 별로 좋지 않다

Sudoku.java

```
package org.example.sudoku;

import android.app.Activity;
import android.os.Bundle;

public class Sudoku extends Activity implements OnClickListener {

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

R.java

```
/* AUTO-GENERATED FILE. DO NOT MODIFY.
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found. It
 * should not be modified by hand.
 */

package org.example.sudoku;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int icon=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030002;
    }
    public static final class string {
        public static final int app_name=0x7f090000;
    }
}
```


레이아웃

- 상위 객체 테두리 안에서 하나 이상의 하위 객체와 이들의 위치를 화면에 설정하는 동작을 포함하는 컨테이너 (교재 32쪽)
 - `FrameLayout`
 - `LinearLayout`
 - `RelativeLayout`
 - `TableLayout`
- 컨테이너 – 객체들을 담는 객체

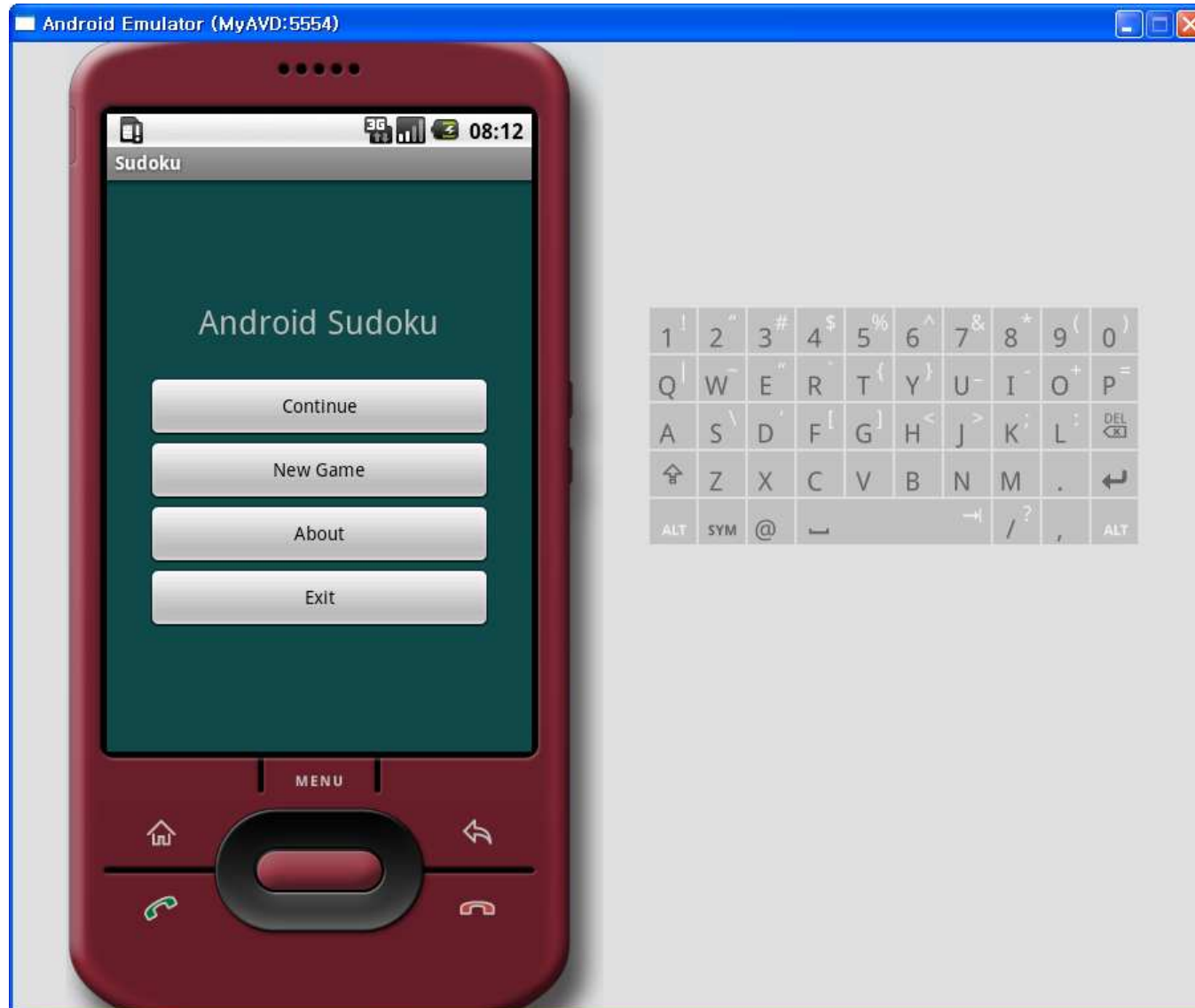
@+id/resid 구문

- 리소스 아이디를 정의하여 이를 통해 참조할 수 있음.
 - `android:id="@+id/continue_button"`
 - `android:id="@+id/new_button"`
 - `android:id="@+id/about_button"`
 - `android:id="@+id/exit_button"`

main.xml

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:background="@color/background"
  android:layout_height="fill_parent" android:layout_width="fill_parent"
  android:padding="30dip" android:orientation="horizontal">
  <LinearLayout android:orientation="vertical"
    android:layout_height="wrap_content" android:layout_width="fill_parent"
    android:layout_gravity="center">
    <TextView android:text="@string/main_title"
      android:layout_height="wrap_content" android:layout_width="wrap_content"
      android:layout_gravity="center" android:layout_marginBottom="25dip"
      android:textSize="24.5sp" />
    <Button android:id="@+id/continue_button"
      android:layout_width="fill_parent" android:layout_height="wrap_content"
      android:text="@string/continue_label" />
    <Button android:id="@+id/new_button"
      android:layout_width="fill_parent" android:layout_height="wrap_content"
      android:text="@string/new_game_label" />
    <Button android:id="@+id/about_button"
      android:layout_width="fill_parent" android:layout_height="wrap_content"
      android:text="@string/about_label" />
    <Button android:id="@+id/exit_button"
      android:layout_width="fill_parent" android:layout_height="wrap_content"
      android:text="@string/exit_label" />
  </LinearLayout>
</LinearLayout>
```

시작 화면



대체 리소스 사용하기

- 예를 들어, 가로 방향을 위한 레이아웃 지정
 - `res/layout-land/main.xml`
- 이외에 모든 리소스의 대체 버전을 명시하는 데에도 디렉터리 이름에 리소스 접미사를 사용함
 - 언어, 지역, 화소 밀도, 해상도, 입력 방법 등

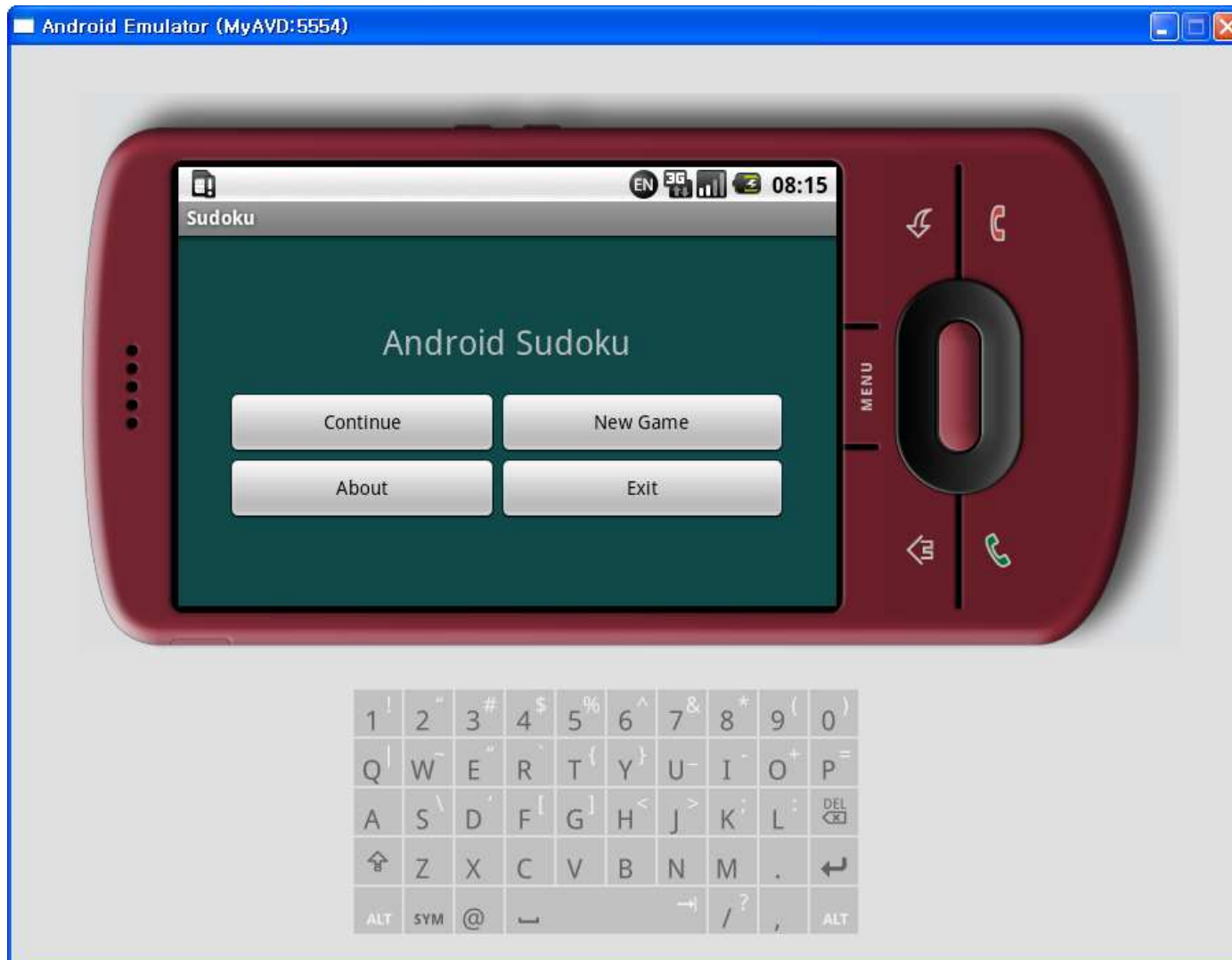
dip 와 sp

- 단순히 픽셀 단위로 거리를 지정하면, 640x480 화면과 1920x1200 화면에서 다르게 보일 것
- 안드로이드의 단위 지정
 - px – pixel 픽셀
 - in – inch 인치
 - mm – millimeter 밀리미터
 - pt – point (인치의 1/72)
 - dp (density independent pixel) – 밀도에 독립적인 화소, 1 인치 당 160개의 점이 있는 디스플레이에서 1 dp 는 1 px 과 같음
 - dip – dip
 - sp (scale independent pixel) – 스케일에 독립적인 화소, dp와 유사하나 사용자의 글꼴 크기 설정에 의해 측정됨

res/layout-land/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android" android:background="@color/background"
    android:layout_height="fill_parent" android:layout_width="fill_parent" android:padding="15dip"
    android:orientation="horizontal">
    <LinearLayout
        android:orientation="vertical" android:layout_height="wrap_content" android:layout_width="fill_parent"
        android:layout_gravity="center" android:paddingLeft="20dip" android:paddingRight="20dip">
        <TextView android:text="@string/main_title"
            android:layout_height="wrap_content" android:layout_width="wrap_content"
            android:layout_gravity="center" android:layout_marginBottom="20dip" android:textSize="24.5sp" />
        <TableLayout android:layout_height="wrap_content" android:layout_width="wrap_content"
            android:layout_gravity="center"
            android:stretchColumns="*">
            <TableRow>
                <Button android:id="@+id/continue_button" android:text="@string/continue_label" />
                <Button android:id="@+id/new_button" android:text="@string/new_game_label" />
            </TableRow>
            <TableRow>
                <Button android:id="@+id/about_button" android:text="@string/about_label" />
                <Button android:id="@+id/exit_button" android:text="@string/exit_label" />
            </TableRow>
        </TableLayout>
    </LinearLayout>
</LinearLayout>
```

가로 방향 레이아웃



About 상자 구현하기

- 스도쿠에 대한 정보가 있는 창이 나옴
- 구현 방법
 - 새 액티비티를 정의하고 시작시킴
 - AlertDialog 클래스를 사용해 보여줌
 - 하위 클래스인 Dialog 클래스를 새 뷰로 팽창시켜 보여줌
- 새 액티비티를 정의함 – About 액티비티
- 이를 위해 새 레이아웃 파일을 정의 –
res/layout/about.xml

res/layout/about.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView

    xmlns:android="http://schemas.android.com/apk/res/andro
id"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10dip">
    <TextView
        android:id="@+id/about_content"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/about_text" />
</ScrollView>
```

res/values/strings.xml

```
<string name="about_title">About Android  
Sudoku</string>
```

```
<string name="about_text">\
```

Sudoku is a logic-based number placement puzzle. Starting with a partially completed 9x9 grid, the objective is to fill the grid so that each row, each column, and each of the 3x3 boxes (also called *blocks*) contains the digits 1 to 9 exactly once.

```
</string>
```

About.java

```
package org.example.sudoku;
```

```
import android.app.Activity;  
import android.os.Bundle;
```

```
public class About extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState)  
    {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.about);  
    }  
}
```

Sudoku.java

```
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;

...

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    // Set up click listeners for all the buttons
    View continueButton = findViewById(R.id.continue_button);
    continueButton.setOnClickListener(this);
    View newButton = findViewById(R.id.new_button);
    newButton.setOnClickListener(this);
    View aboutButton = findViewById(R.id.about_button);
    aboutButton.setOnClickListener(this);
    View exitButton = findViewById(R.id.exit_button);
    exitButton.setOnClickListener(this);
}
```

Sudoku.java

```
public class Sudoku extends Activity implements OnClickListener {
```

```
...
```

```
public void onClick(View v) {  
    switch (v.getId()) {  
        case R.id.continue_button:  
            startGame(Game.DIFFICULTY_CONTINUE);  
            break;  
            // ...  
  
        case R.id.about_button:  
            Intent i = new Intent(this, About.class);  
            startActivity(i);  
            break;  
            // More buttons go here (if any) ...  
        case R.id.new_button:  
            openNewGameDialog();  
            break;  
        case R.id.exit_button:  
            finish();  
            break;  
    }  
}
```

AndroidManifest.xml에 다음을 추가

```
<activity android:name=".About"  
    android:label="@string/about_title" >  
</activity>
```

테마 적용하기

```
<activity android:name=".About"  
    android:label="@string/about_title"  
  
    android:theme="@android:style/Theme.Dialog"  
>  
    </activity>
```


요약

- 선언하여 디자인을 하는 방법을 설명하고, 실제 예를 보였다.
- 시작화면을 만드는 방법과 대체 리소스를 사용하는 방법을 설명하고, 실제 예를 보였다.
- About 과 같은 상자를 구현하고, 테마를 적용하는 법을 설명하고, 실제 예를 보였다.

퀴즈

- 사용자 인터페이스를 디자인하는 방법들은 무엇이 있는가?
- 자동으로 생성되며 리소스와 관련이 있는 클래스는 무엇인가?
- 레이아웃은 무엇인가?
- 컨테이너란 무엇인가?
- 리소스 아이디는 어떻게 정의되는가?
- 대체 리소스를 사용해야 하는 경우는 언제인가?
- 새로운 액티비티는 어디에 등록해야 하는가?

연습문제

- 프로그램을 실행했을 때, 초기 화면을 선언해 보자. 이를 위해 xml 파일을 정의하고, 액티비티의 메서드를 수정하자.
- 선언한 초기 화면에 대해, 가로로 회전했을 때를 위한 레이아웃을 정의해 보자.
- 새로운 액티비티를 만들어 보자. 어떠한 일들을 해야 하는가?