

헬로, 안드로이드

4주차 - 사용자 인터페이스 디자인하기 (2)

강대기

동서대학교 컴퓨터정보공학부

학습 목표

- 메뉴를 추가하는 방법을 이해하고 실습할 수 있다.
- 프로그램의 기본 설정(settings)을 정의하는 방법을 알고 실습할 수 있다.
- 대화 상자를 여는 방법을 알고 실습할 수 있다.
- 로그 메시지로 디버깅하는 방법을 이해한다.
- 디버거로 디버깅하는 방법을 이해한다.

차례

- 메뉴 추가하기
- Settings 추가하기
- 새 게임 시작하기
- 로그 메시지로 디버깅하기
- 디버거로 디버깅하기
- 게임 끝내기
- 요약
- 퀴즈
- 연습문제

메뉴 추가하기

- 안드로이드가 지원하는 두 가지 메뉴
 - 메뉴 버튼을 눌렀을 때 나오는 메뉴
 - 화면을 손가락으로 누를 때 나오는 상황적 (contextual) 메뉴
- 메뉴 버튼을 눌렀을 때 나오는 메뉴를 만들려면
 1. 사용할 문자열 정의
 2. XML을 사용하여 메뉴 정의
 3. import 추가
 4. 메서드 오버라이드

사용할 문자열 정의

/res/values/strings.xml

```
<string
  name="settings_label">Settings...</string>
<string name="settings_title">Sudoku
settings</string>
<string name="settings_shortcut">s</string>
<string name="music_title">Music</string>
<string name="music_summary">Play
background music</string>
<string name="hints_title">Hints</string>
<string name="hints_summary">Show hints
during play</string>
```

XML을 사용하여 메뉴 정의

/res/menu/menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu
  xmlns:android="http://schemas.android.com/a
  pk/res/android">
  <item android:id="@+id/settings"
    android:title="@string/settings_label"

    android:alphabeticShortcut="@string/settings_
    shortcut" />
</menu>
```

import 추가 Sudoku.java

```
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.MenuItem;
```

메서드 오버라이드

(onCreateOptionsMenu in Sudoku.java)

@Override

```
public boolean onCreateOptionsMenu(Menu
menu) {
    super.onCreateOptionsMenu(menu);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu, menu);
    return true;
}
```


메서드 오버라이드 (onOptionsItemSelected in Sudoku.java)

@Override

```
public boolean onOptionsItemSelected(MenuItem  
item) {  
    switch (item.getItemId()) {  
        case R.id.settings:  
            startActivity(new Intent(this, Prefs.class));  
            return true;  
        // More items go here (if any) ...  
    }  
    return false;  
}
```

Settings 추가하기

- 기본 설정 정의
 - 예: `res/xml/settings.xml`
- Prefs 클래스 정의하여 PreferenceActivity 확장
 - `/src/org/example/sudoku/Prefs.java`
- Prefs 액티비티를 AndroidManifest.xml에 등록
 - `<activity android:name=".Prefs"`
 - `android:label="@string/settings_title">`
 - `</activity>`

res/xml/settings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen
  xmlns:android="http://schemas.android.com/apk/res/android">
  <CheckBoxPreference
    android:key="music"
    android:title="@string/music_title"
    android:summary="@string/music_summary"
    android:defaultValue="true" />
  <CheckBoxPreference
    android:key="hints"
    android:title="@string/hints_title"
    android:summary="@string/hints_summary"
    android:defaultValue="true" />
</PreferenceScreen>
```



```
/src/org/example/sudoku/Prefs.java
```

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

```
import android.os.Bundle;
```

```
import android.preference.PreferenceActivity;
```

```
public class Prefs extends PreferenceActivity {
```

```
    @Override
```

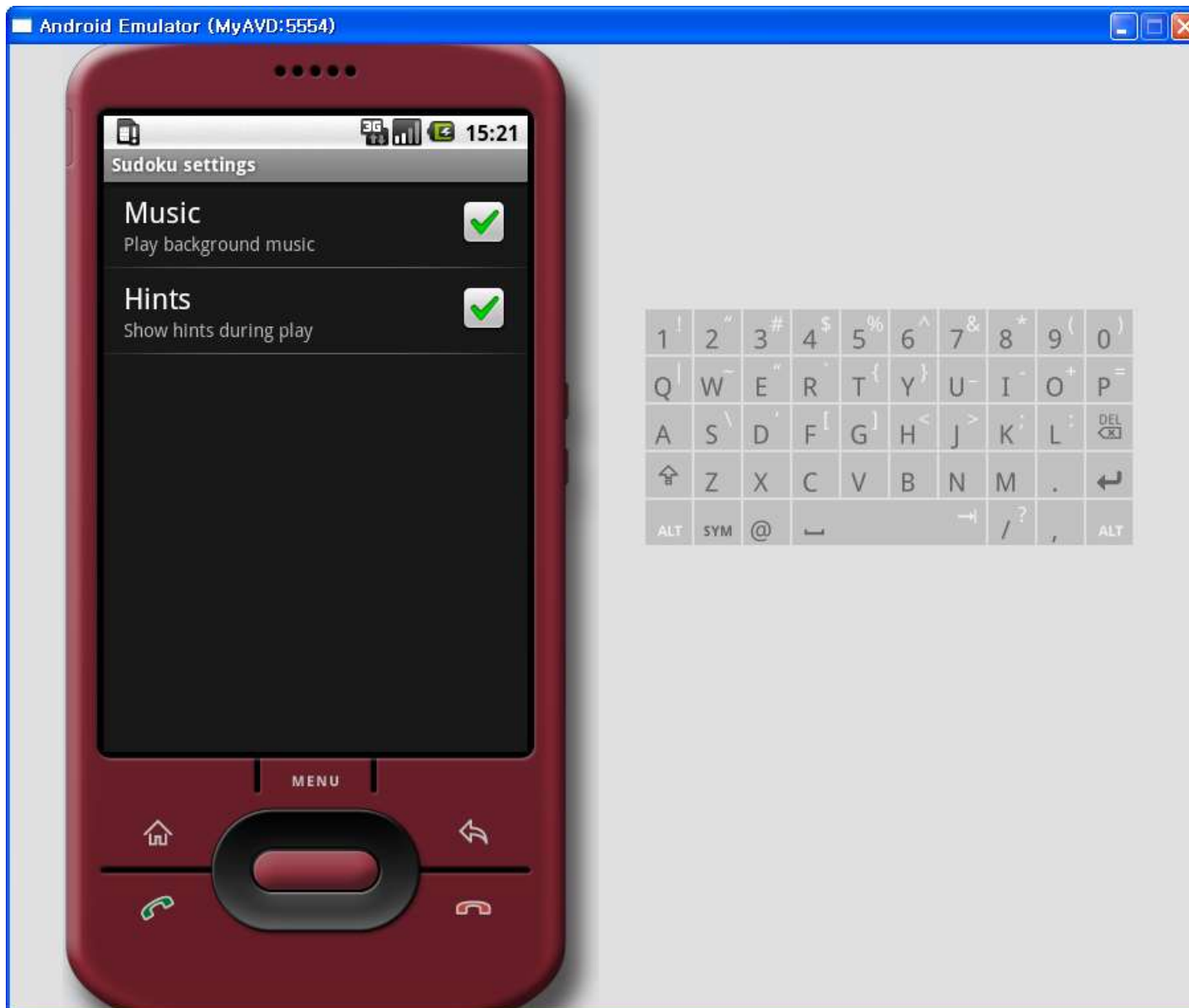
```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        addPreferencesFromResource(R.xml.settings);
```

```
    }
```

```
}
```



새 게임 시작하기

- 문자열 추가
- 배열 리소스 만듦
- import 추가
- “News Game” 버튼 클릭 처리
- 난이도 목록 사용자 인터페이스 만듦

/res/values/strings.xml

```
<string name="new_game_title">Difficulty</string>  
<string name="easy_label">Easy</string>  
<string name="medium_label">Medium</string>  
<string name="hard_label">Hard</string>
```

/res/values/arrays.xml

```
<resources>  
  <array name="difficulty">  
    <item>@string/easy_label</item>  
    <item>@string/medium_label</item>  
    <item>@string/hard_label</item>  
  </array>  
</resources>
```




```
/src/org/example/sudoku/Sudoku.java
```

```
import 추가
```

```
import android.app.AlertDialog;
```

```
import android.content.DialogInterface;
```

```
import android.util.Log;
```

/src/org/example/sudoku/Sudoku.java

“News Game” 버튼 클릭 처리

```
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.continue_button:
            startGame(Game.DIFFICULTY_CONTINUE);
            break;
            // ...

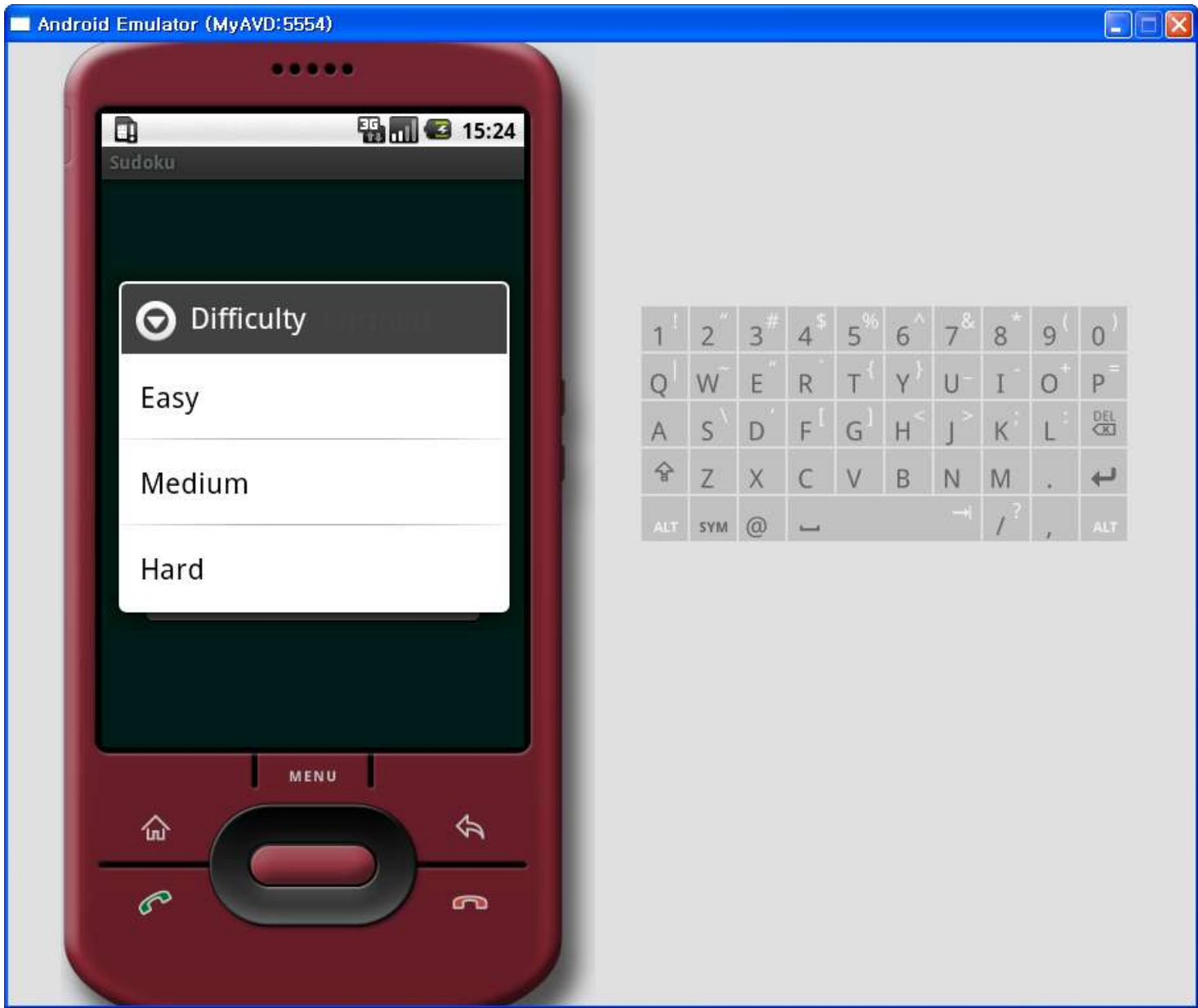
        case R.id.about_button:
            Intent i = new Intent(this, About.class);
            startActivity(i);
            break;
            // More buttons go here (if any) ...
        case R.id.new_button:
            openNewGameDialog();
            break;
        case R.id.exit_button:
            finish();
            break;
    }
}
```

/src/org/example/sudoku/Sudoku.java

난이도 목록 사용자 인터페이스 만들기

```
/** Ask the user what difficulty level they want */
private void openNewGameDialog() {
    new AlertDialog.Builder(this)
        .setTitle(R.string.new_game_title)
        .setItems(R.array.difficulty,
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialoginterface,
                    int i) {
                    startGame(i);
                }
            })
        .show();
}
```

```
/** Start a new game with the given difficulty level */
private void startGame(int i) {
    Log.d(TAG, "clicked on " + i);
    Intent intent = new Intent(Sudoku.this, Game.class);
    intent.putExtra(Game.KEY_DIFFICULTY, i);
    startActivity(intent);
}
```



로그 메시지로 디버깅하기

- Log.e() – 오류 (error)
 - Log.w() – 경고 (warning)
 - Log.i() – 정보 (information)
 - Log.d() – 디버깅 (debugging)
 - Log.v() – 세부 정보 (verbose)
-
- Window > Show View > Other > Android > LogCat

디버거로 디버깅하기

- 프로젝트 디버깅 활성화
- AndroidManifest.xml 에서
android:debuggable="true"
 - `<application android:icon="@drawable/icon"
android:label="@string/app_name"
android:debuggable="true">`
- Right click > Debug As > Android Application

게임 끝내기

- 대부분의 스마트폰 응용 프로그램에서 끝내기 버튼은 필요치 않음 - 안드로이드의 경우, 이전 버튼이나 홈 키를 누르면 됨

- Activity 를 종료하는 방법

```
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.exit_button:
            finish();
            break;
    }
}
```

요약

- 메뉴를 추가하는 방법을 설명하였다.
- 프로그램의 기본 설정(settings)을 정의하는 방법을 설명하였다.
- 대화 상자를 여는 방법을 설명하였다.
- 로그 메시지로 디버깅하는 방법을 소개하였다.
- 디버거로 디버깅하는 방법을 소개하였다.

퀴즈

- 안드로이드가 지원하는 두 가지 메뉴는 무엇인가?
- 기본 설정을 정의하기 위해서는 무슨 클래스를 확장해야 하는가?
- New Game 버튼을 클릭하면 나오는 대화 상자는 어떤 클래스인가?
- 로그 메시지로 디버깅하는 종류는 몇 가지가 있고 무엇 무엇인가?
- 프로젝트의 디버깅을 활성화하려면 어떻게 해야 하는가?
- 액티비티를 종료하려면 어떻게 하면 되는가?

연습문제

- 메뉴 버튼을 눌렀을 때 나오는 메뉴에 자신의 이름이 나오도록 바꾸어 보라.
- 기본 설정에 체크 박스를 3 개 더 정의해 보라.
- HelloMenu 라는 안드로이드 프로그램 프로젝트를 만들어서 메뉴 버튼 눌렀을 때 나오는 메뉴 아이템을 "설정"이라고 하고, "설정"이라는 아이템을 누르면 "음악"과 "힌트"에 대한 설정 화면(액티비티)를 실행하는 프로그램을 만들어 보자.
- `openNewGameDialog` 을 보면 한 줄로 구성되어 있다. 이를 세 줄 이상으로 나누어 보라.
- `openNewGameDialog` 메서드의 `DialogInterface.OnClickListener` 를 매개변수가 아닌 외부 클래스로 정의해 보라.