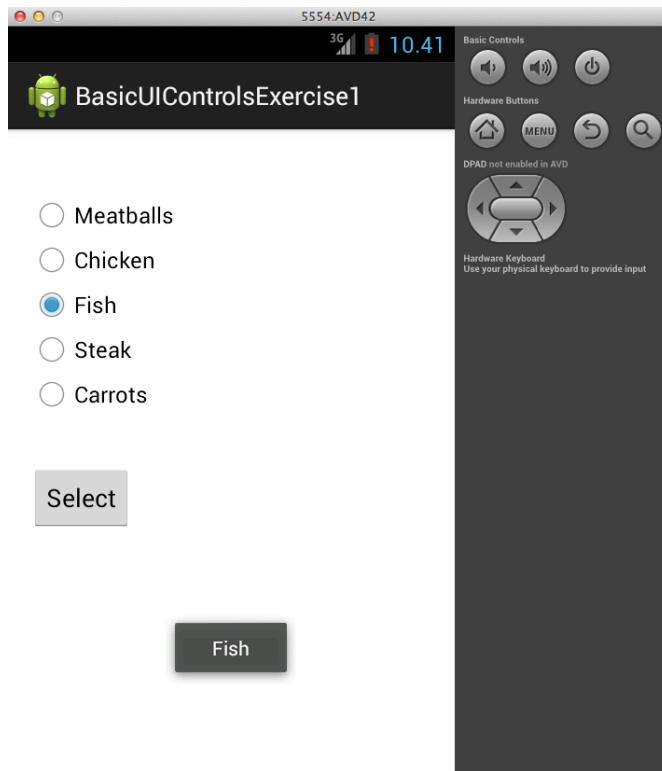


Basic UI Controls - 1

Create application with RadioButtons and use Toast to show selected RadioButton Text.



Layout XML file:

- use RadioGroup to populate RadioButtons
- create a few RadioButtons in main.xml layout
- add Select Button
- remember add food texts to string resources

`<RadioGroup`

```
android:id="@+id/myRadioGroup"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:layout_marginTop="29dp" >
```

`<RadioButton`

```
android:id="@+id/radio0"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:checked="true"
```

```
        android:text="@string/meatballs_text" />
    ...

```

Test your application in Emulator. You will see that your application is working but nothing happens when you click Select Button.

Open your main Activity class and implement following method

```
public void selectButtonClicked(View view) {
    // find radiogroup
    RadioGroup rg = (RadioGroup) findViewById(R.id.myRadioGroup);
    int id = rg.getCheckedRadioButtonId();
    // find button
    RadioButton rb = (RadioButton) findViewById(id);
    // get radiobutton text
    String text = (String) rb.getText();
    // toast message to screen
    Toast.makeText(getApplicationContext(), text,
        Toast.LENGTH_SHORT).show();
}

```

Remember call above method from Select button (from layout file)

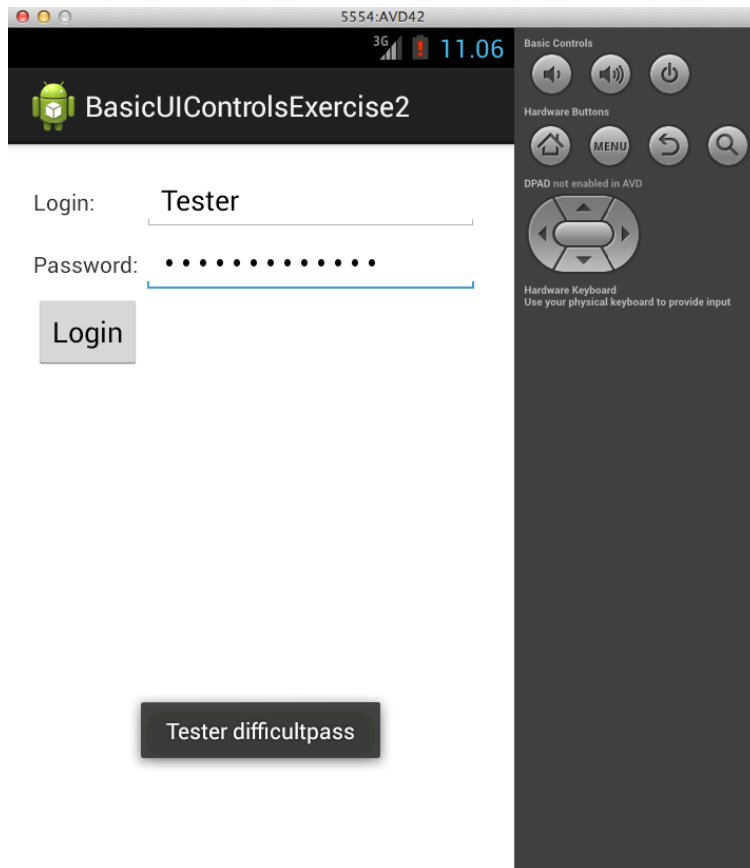
```
android:onClick="selectButtonClicked"

```

Test your application. Now it should Toast selected food!

Basic UI Controls - 2

Create application with TextView, EditText and AutoCompleteTextView Controls to demonstrate login page.



Layout XML file

- first create layout as above screen shows
- use `TableLayout` to align objects
- use `AutoCompleteTextView` in login
- use `Password` style in `EditText` in password
- use `Button` to toast login and password strings

Populate `AutoCompleteTextView` in `onCreate()`-method

```
// AutoCompleteTextView
AutoCompleteTextView actv = (AutoCompleteTextView)
    findViewById(R.id.login); // add stings to control
ArrayAdapter<String> aa = new ArrayAdapter<String>(this,
    android.R.layout.simple_dropdown_item_1line,
    new String[]
        {"Pasi", "Juha", "Kari", "Jouni", "Esa", "Hannu"});
actv.setAdapter(aa);
```

Test your application. Type for example: Ka in your `AutoCompleteTextView` as a login and it should show Kari word for you.