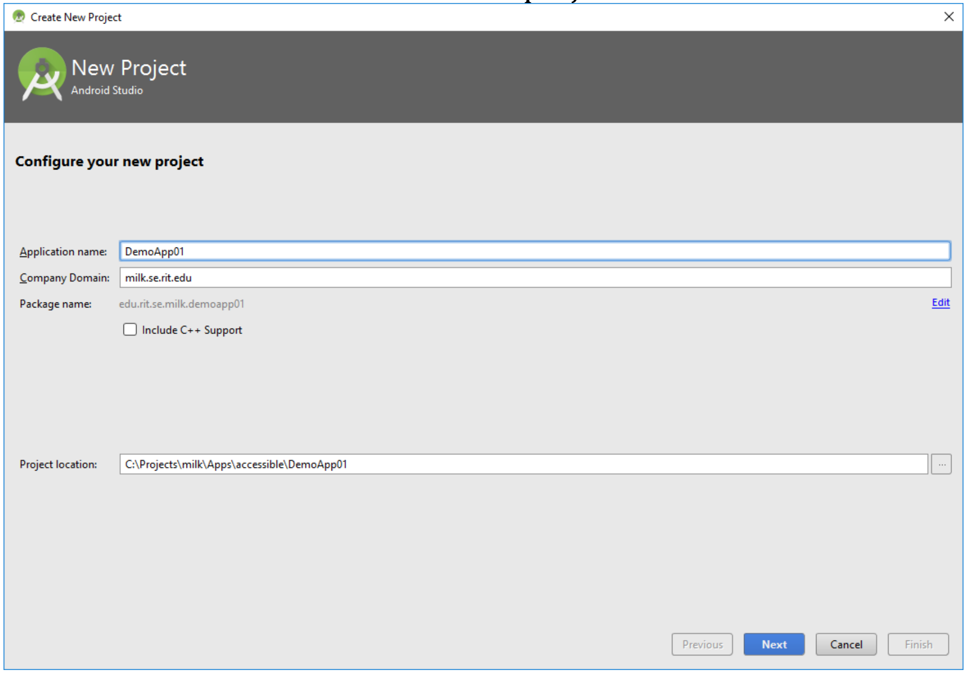
Introduction to contentDescription

Background

- What is some background information on the module

Activity Instructions

1. Project Creation
    a. Follow the screens below to create a new project:



Configure your new project

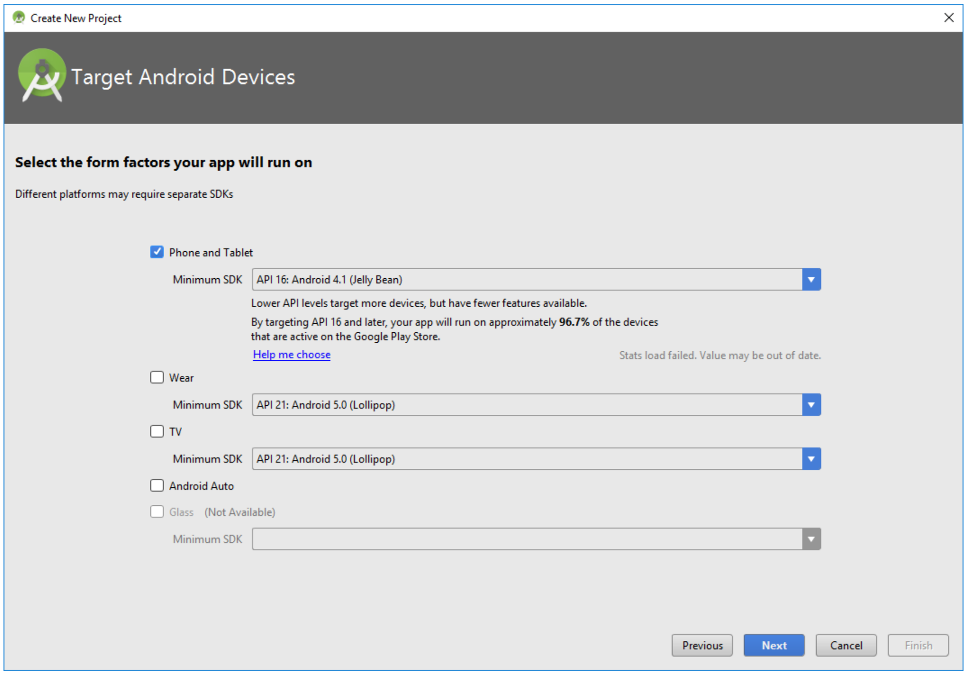
Application name: DemoApp01

Company Domain: milk.se.rit.edu

Package name: edu.rit.se.milk.demoapp01 Edit

□ Include C++ Support

Project location: C:\Projects\milk\Apps\accessible\DemoApp01



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 16: Android 4.1 (Jelly Bean)

Lower API levels target more devices, but have fewer features available.

By targeting API 16 and later, your app will run on approximately **96.7%** of the devices that are active on the Google Play Store.

Help me choose

Stats load failed. Value may be out of date.

☐ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

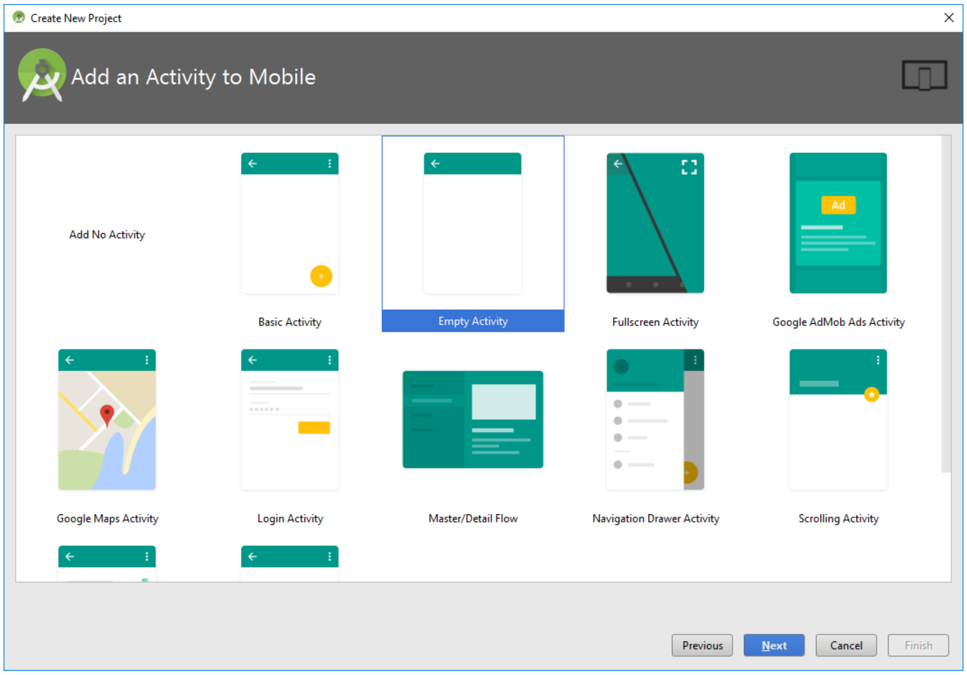
☐ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Android Auto

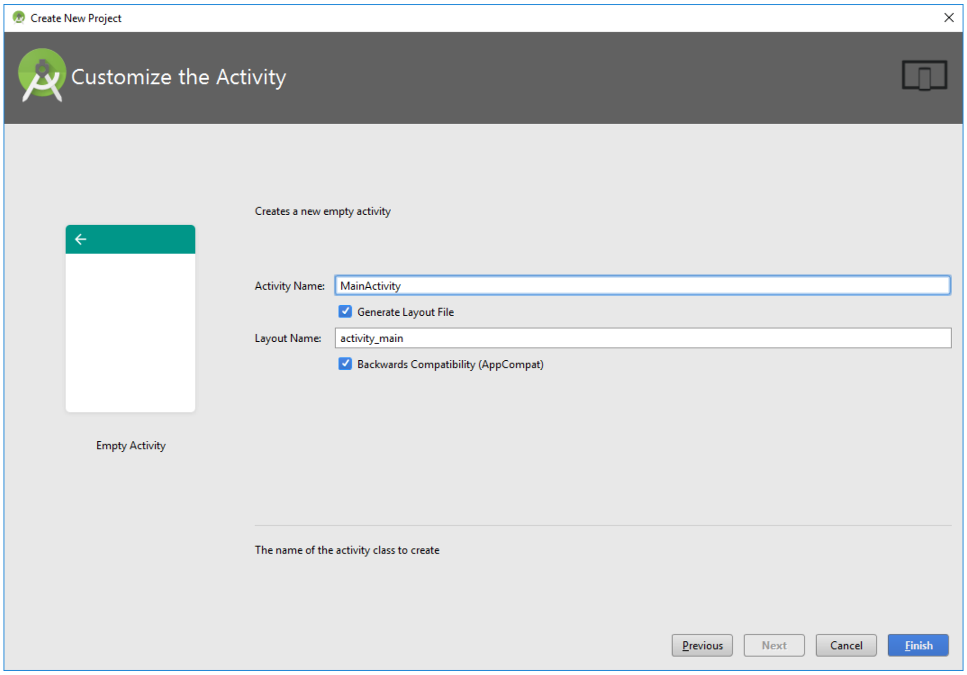
☐ Glass (Not Available)

Minimum SDK:



Create New Project

Add an Activity to Mobile



Create New Project

Customize the Activity

Creates a new empty activity

Activity Name: MainActivity

☑ Generate Layout File

Layout Name: activity_main

☑ Backwards Compatibility (AppCompat)

Empty Activity

The name of the activity class to create

2. Construct User Interface
   a. From the Palette tool window, add the following UI controls into the screen layout.
      i. Update the following properties of the existing Relative Layout:
         - `layout_width="match_parent"`
         - `layout_height="match_parent"`

      ii. Within the existing Relative Layout add:
         1. **Switch**
            - Update the following properties:
               - `text="Alternate Rendering"`
               - `layout_width="match_parent"`
               - `layout_height="wrap_content"`
               - `id="@+id/switchAccessibility"`
               - `focusable="false"`
         2. **Relative Layout**:
            - Update the following properties:
               - `layout_width="match_parent"`
               - `layout_height="wrap_content"`
               - `id="@+id/layoutContents"`
               - `layout_weight="100"`
            - Add the following UI controls:
               i. **Button**:
                  - Update the following properties:
                     - `text="&#60;&#60;--"`
                     - `layout_width="wrap_content"`
                     - `layout_height="wrap_content"`
                     - `layout_alignParentTop="true"`

        - layout_alignParentLeft="true"
        - layout_alignParentStart="true"
        - layout_marginTop="146dp"
        - id="@+id/buttonLeft"
        - background="@android:drawable/btn_default"
        - gravity="center"
        - layout_gravity="left|center"
        - layout_marginLeft="50dp"

ii. **TextView**

- Update the following properties:
    - text="Tap the Back or Next button to proceed"
    - layout_width="match_parent"
    - layout_height="wrap_content"
    - id="@+id/textView"
    - textAppearance="@android:style/TextAppearance.DeviceDefault.Medium"
    - layout_above="@+id/buttonRight"
    - layout_marginBottom="36dp"
    - textStyle="normal|bold"
    - textAlignment="center"
    - layout_alignParentLeft="false"
    - layout_alignParentStart="false"
    - layout_alignParentRight="false"
    - layout_alignParentEnd="false"
    - gravity="center_horizontal"

iii. **Button**

- Update the following properties:
    - text="--&#62;&#62;"
    - layout_width="wrap_content"
    - layout_height="wrap_content"
    - id="@+id/buttonRight"
    - layout_gravity="right"
    - background="@android:drawable/btn_default"
    - gravity="center"
    - layout_alignTop="@+id/buttonLeft"
    - layout_alignParentRight="true"
    - layout_alignParentEnd="true"
    - layout_marginRight="50dp"

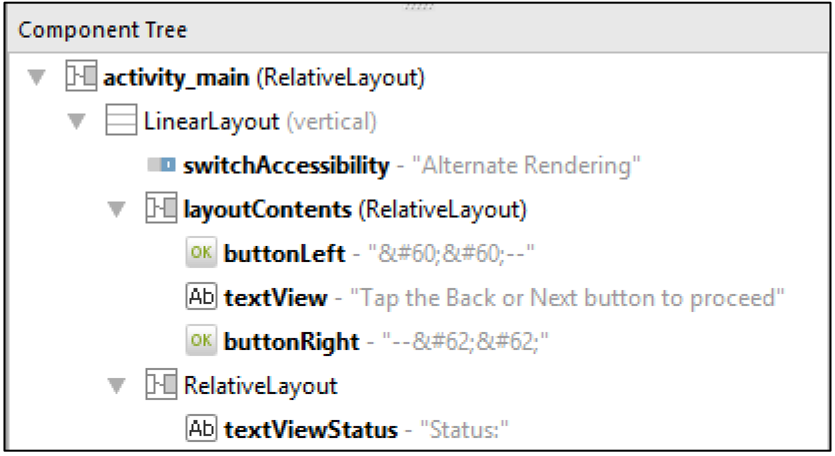
3. **Relative Layout**

- Update the following properties:
    - layout_width="match_parent"
    - layout_height="wrap_content"
    - layout_weight="1"
- Add the following UI controls:

i. **TextView**:

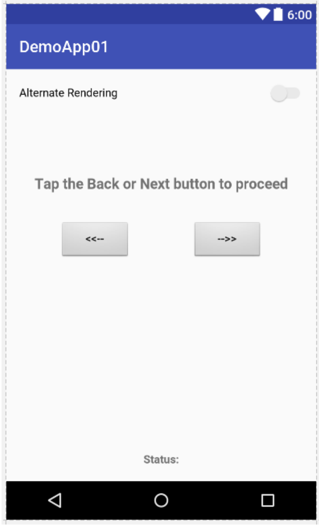
- `Update the following properties:`
- `layout_width="match_parent"`
- `layout_height="wrap_content"`
- `id="@+id/textViewStatus"`
- `layout_alignParentTop="true"`
- `layout_alignParentLeft="true"`
- `layout_alignParentStart="true"`
- `textAlignment="center"`
- `textStyle="normal|bold"`
- `layout_alignParentRight="true"`
- `layout_alignParentEnd="true"`
- `text="Status:"`
- `gravity="bottom"`

Following is the hierarchical layout of the controls on the screen:



```
Component Tree
▼ activity_main (RelativeLayout)
    ▼ LinearLayout (vertical)
          switchAccessibility - "Alternate Rendering"
        ▼ layoutContents (RelativeLayout)
              buttonLeft - "&#60;&#60;--"
              textView - "Tap the Back or Next button to proceed"
              buttonRight - "--&#62;&#62;"
        ▼ RelativeLayout
              textViewStatus - "Status:"
```

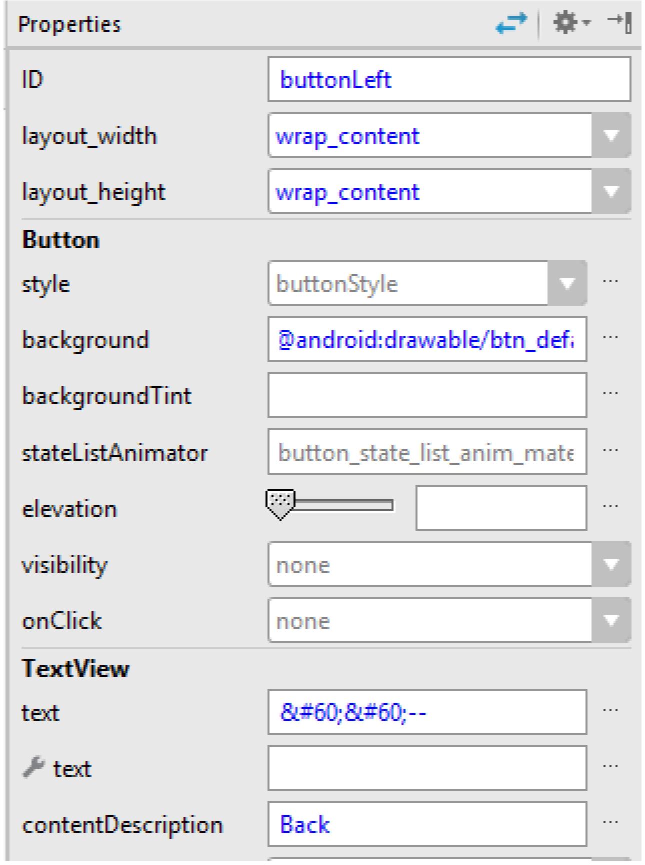
Following is the rendering of controls on the screen:



3. Set Content Description

The “contentDescription” property of a control is utilized for accessibility purposes. The text associated with this property defines the content of this control and is used by Android’s TalkBack feature.

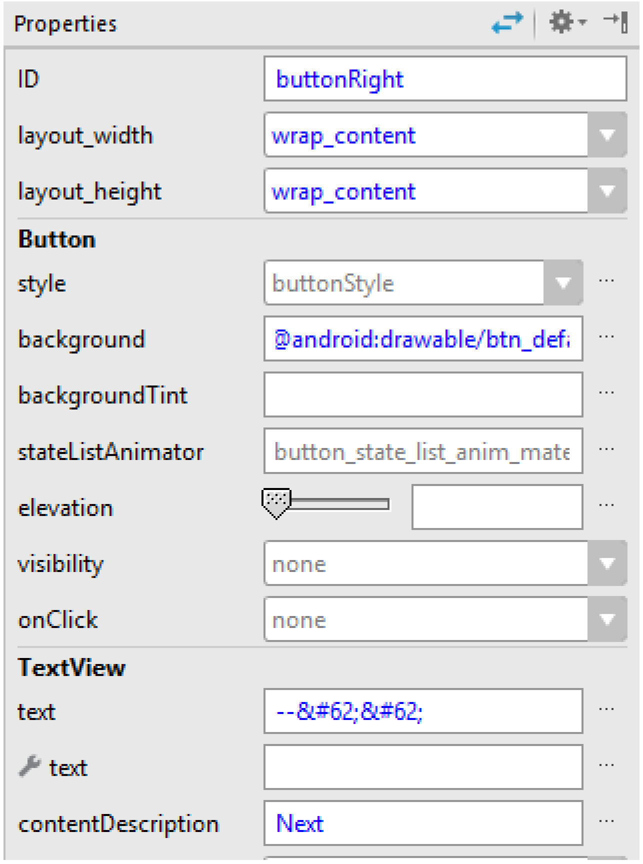
Select the Button with id “buttonLeft”. Set contentDescription to “Back”



Properties

ID	buttonLeft
layout_width	wrap_content
layout_height	wrap_content
Button	
style	buttonStyle
background	@android:drawable/btn_def
backgroundTint	
stateListAnimator	button_state_list_anim_mate
elevation	
visibility	none
onClick	none
TextView	
text	&#60;&#60;--
text	
contentDescription	Back

Select the Button with id “buttonRight”. Set contentDescription to “Next”



Properties

ID	buttonRight
layout_width	wrap_content
layout_height	wrap_content
Button	
style	buttonStyle
background	@android:drawable/btn_def
backgroundTint	
stateListAnimator	button_state_list_anim_mate
elevation	
visibility	none
onClick	none
TextView	
text	--&#62;&#62;
text	
contentDescription	Next

4. <u>Code</u>

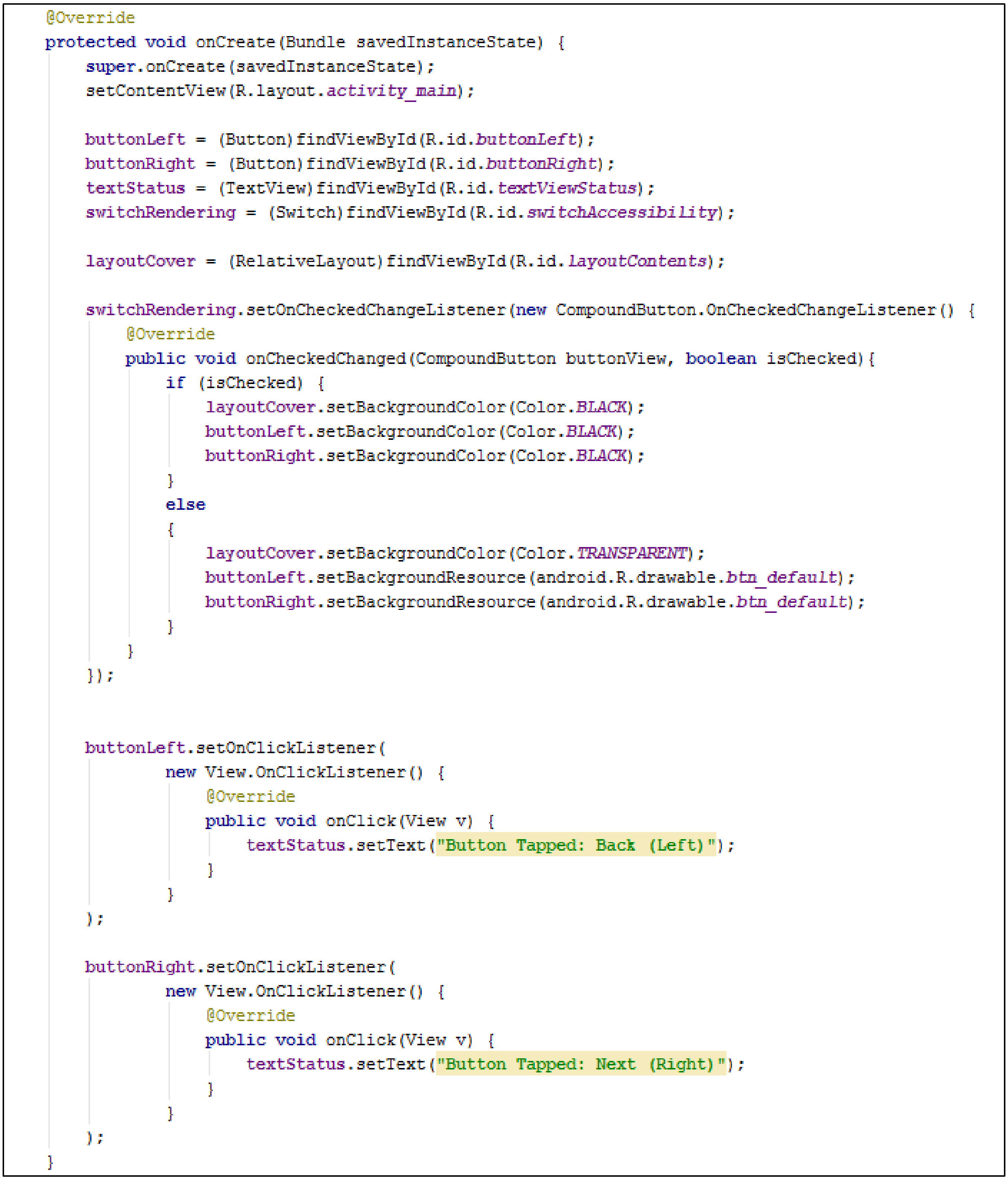
Open MainActivity.java and add the following code:

a. Declare the following variables:

```
public class MainActivity extends AppCompatActivity {

    Button buttonLeft, buttonRight;
    TextView textStatus;
    Switch switchRendering;
    RelativeLayout layoutCover;
```

b. Add the following code inside the **onCreate** method:



```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    buttonLeft = (Button)findViewById(R.id.buttonLeft);
    buttonRight = (Button)findViewById(R.id.buttonRight);
    textStatus = (TextView)findViewById(R.id.textViewStatus);
    switchRendering = (Switch)findViewById(R.id.switchAccessibility);

    layoutCover = (RelativeLayout)findViewById(R.id.layoutContents);

    switchRendering.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean isChecked){
            if (isChecked) {
                layoutCover.setBackgroundColor(Color.BLACK);
                buttonLeft.setBackgroundColor(Color.BLACK);
                buttonRight.setBackgroundColor(Color.BLACK);
            }
            else
            {
                layoutCover.setBackgroundColor(Color.TRANSPARENT);
                buttonLeft.setBackgroundResource(android.R.drawable.btn_default);
                buttonRight.setBackgroundResource(android.R.drawable.btn_default);
            }
        }
    });


    buttonLeft.setOnClickListener(
            new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    textStatus.setText("Button Tapped: Back (Left)");
                }
            }
    );

    buttonRight.setOnClickListener(
            new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    textStatus.setText("Button Tapped: Next (Right)");
                }
            }
    );
}
```

The above code achieves the following:

1. Handles the onCheckedChange event of the Switch control to:
    a. Set the color of all the controls to Black so that none of the controls are visible when the switch is checked (i.e. set to “On”)
    b. Sets the colors of all the controls to their original colors when the switch is unchecked (i.e. set to “Off”)