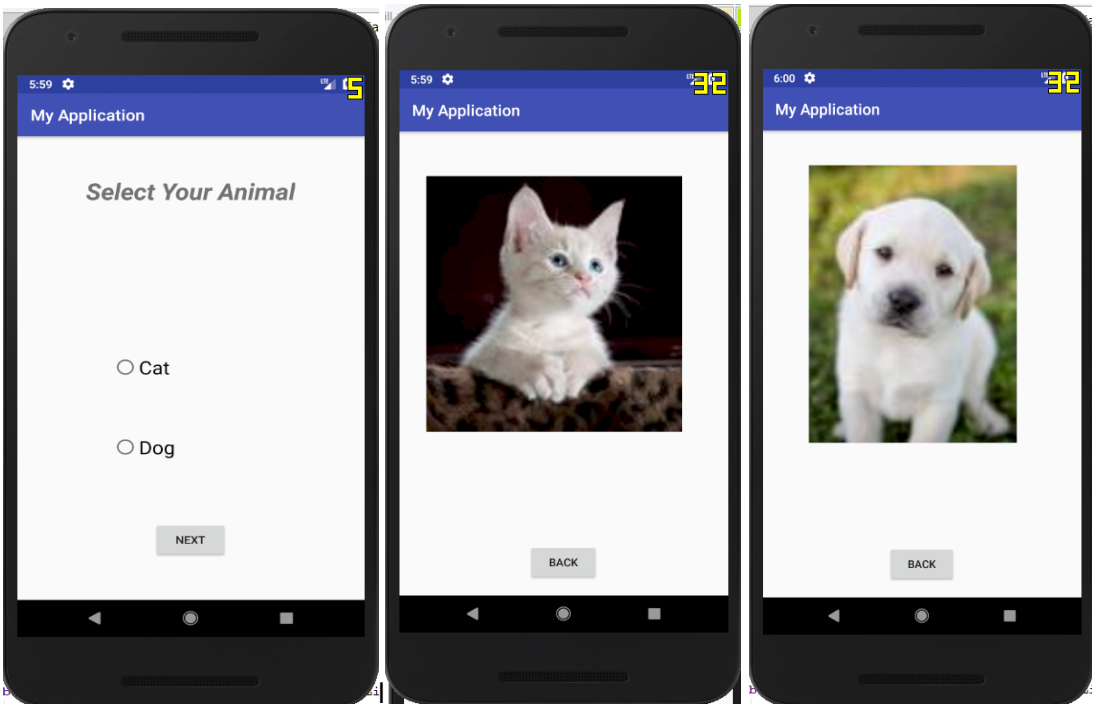
Introduction to User Interface Elements in Android

Objective: In this tutorial you will learn how to use different User Interface elements provided by Android Studio. At the end of the session you will be able to:

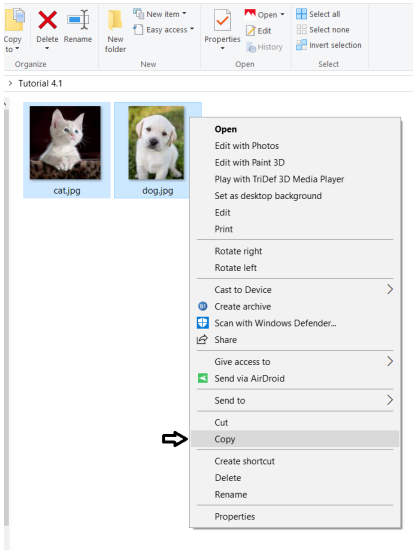
- Create radio buttons
- Create a progress bar
- Display images in your mobile app

In this session we are going to create an application in which the user will be able to choose an image to display. The user can select an image of either a cat or a dog using radio button and click next. If the option “Cat” is selected, a new screen will appear with a cat picture. If the option “Dog” is selected, a new screen will appear with a dog picture.

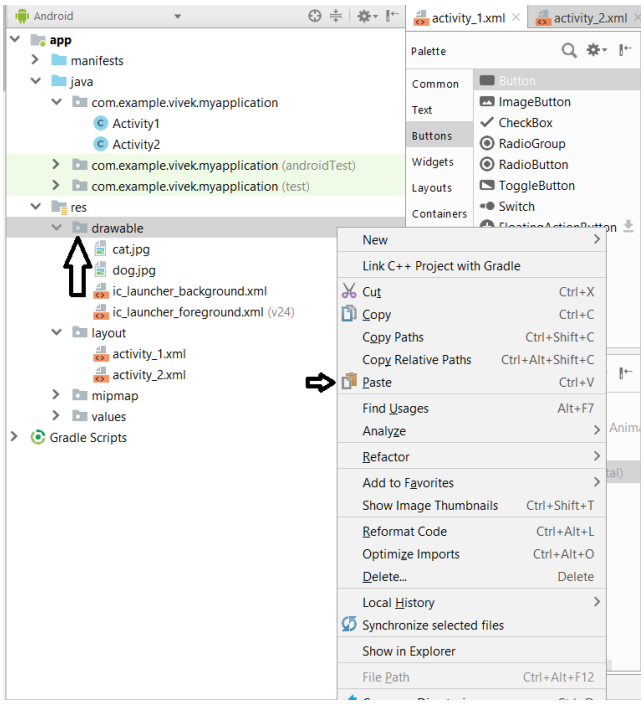


Part 1: Adding Images in Android Project

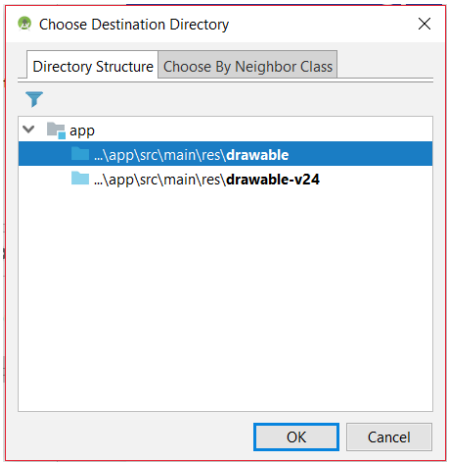
1. Your first task is to find and download two images, one of a cat and the other is a dog. Each image should be less than 500kb in size.
2. Rename the images to ‘cat.jpg’ and ‘dog.jpg’ respectively.
3. Copy the images from where they are stored on your computer.

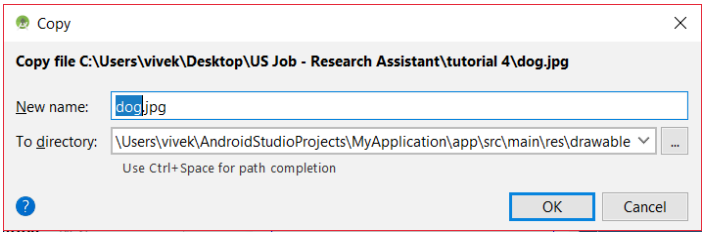


4. Open the project you have worked with in previous tutorial. The images of your project will be stored in a folder named 'drawable' which is located inside 'res' folder. Expand it in project window.
5. Right click on 'drawable' folder and click on paste.



6. Click OK on next windows. Now you will be able to see the image names listed in drawable folder. Refer to the below images.

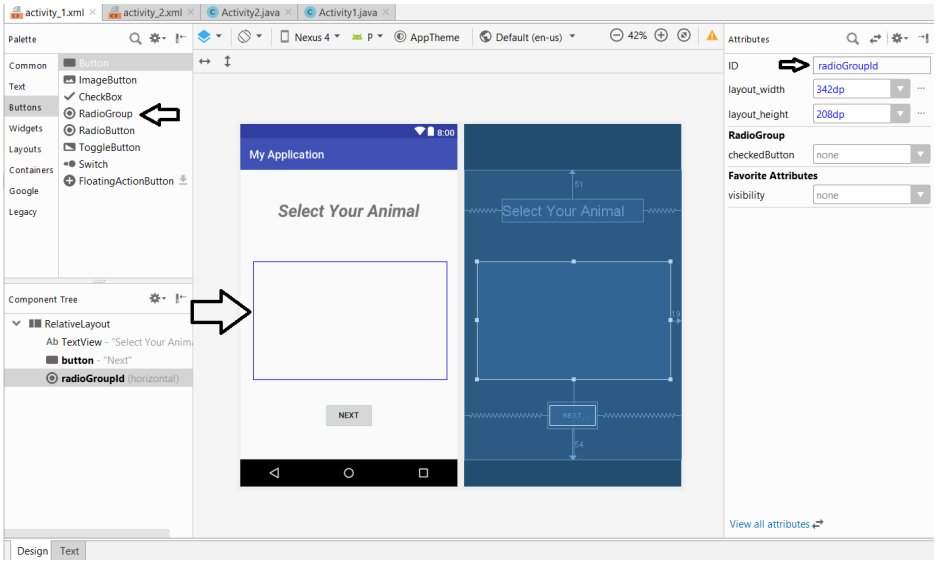




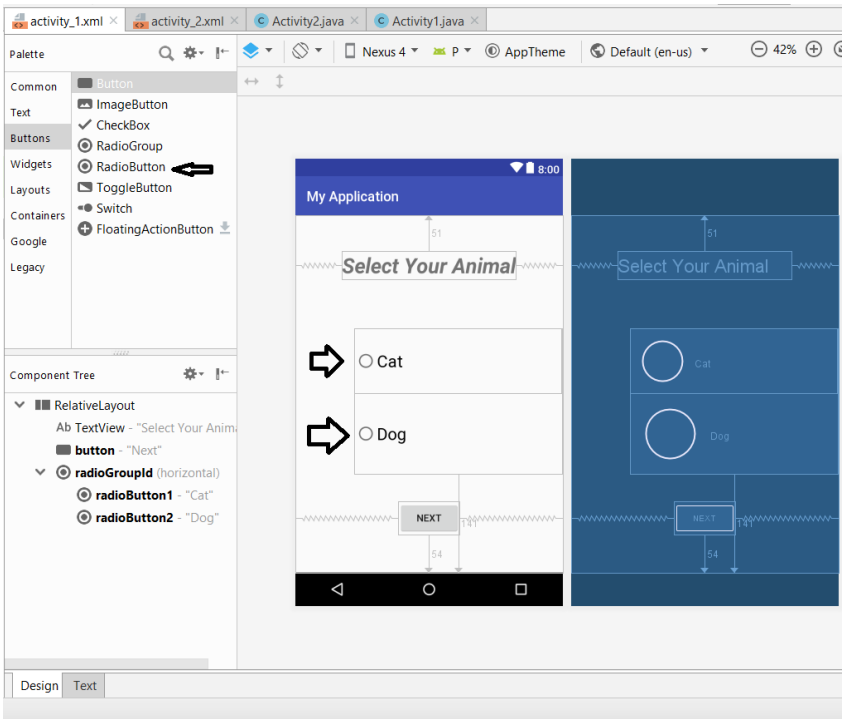
#End_of_Part_1

Part 2: Create Radio Buttons

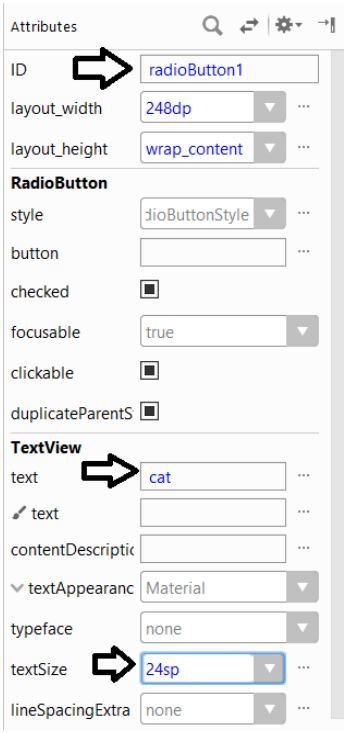
1. Open activity_youruserid1.
2. Change the text to ‘Select your Animal’ and change it’s position and size accordingly.
3. Select the ‘RadioGroup’ from Palette -> Buttons.



4. Adjust the size and position of RadioGroup. Remember, this area will contain both radio buttons.
5. Change the ID of RadioGroup to ‘radioGroupId’ as shown in the above screenshot.
6. Add two radio buttons and position them as shown in image.



7. Change the text, ID and textSize of ‘cat’ radiobutton as shown in the image below.



8. Make the same changes for ‘dog’ radio button.

#End_of_Part_2

Part 3: Adding the Functionality of Radio Button

1. Activity_Youruserid1.java code initially, is shown in the screenshot below.

saActivity.java | activity_sa.xml | SpoorthyActivity.java | activity_spoorthy.xml

```
package com.example.spoorthy.spoorthyapplication;

import android.content.Context;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class SpoorthyActivity extends AppCompatActivity {
    private Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_spoorthy);
    button = (Button) findViewById(R.id.button);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick (View v){
        openActivitysa();
    }
    });
}
   public void openActivitysa() {
        Intent intent = new Intent( packageContext: this, saActivity.class);
        startActivity(intent);
    }
}
```

2. Let's add the code step by step now. To use radiogroup, import statements are used to invoke the packages which contains RadioGroup widget.A RadioGroup class is used for a set of radio buttons. If we check one radio button that belongs to a radio group, it automatically unchecks any previously checked radio button within the same group.

saActivity.java | activity_sa.xml | SpoorthyActivity.java | activity_spoorthy.xml

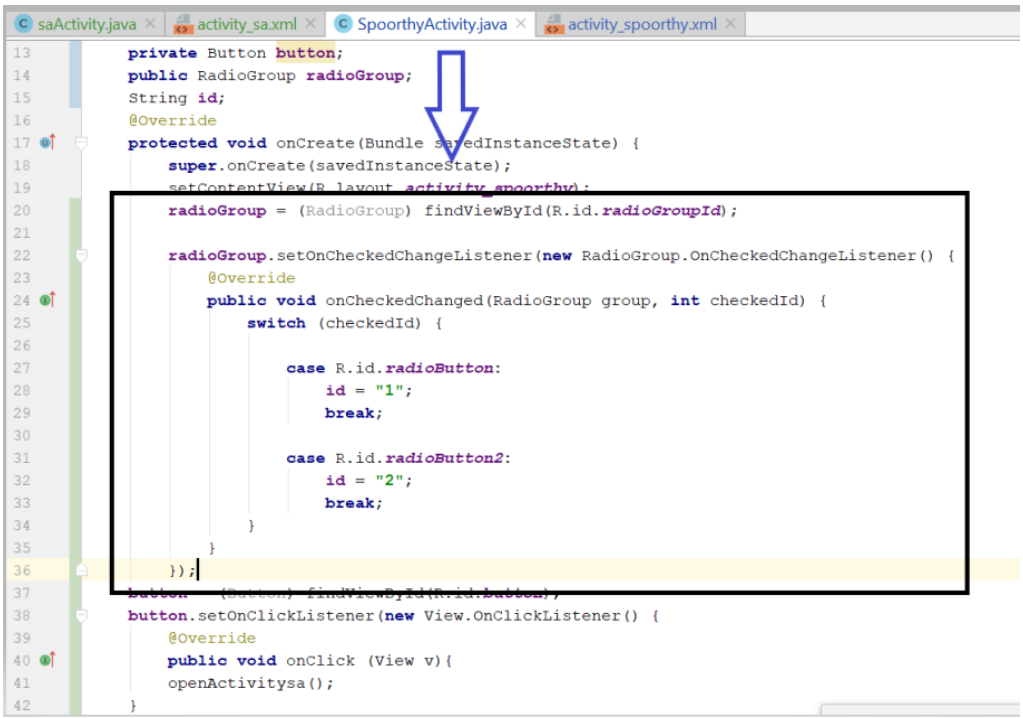
```
package com.example.spoorthy.spoorthyapplication;

import android.content.Context;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.RadioGroup;

public class SpoorthyActivity extends AppCompatActivity {
    private Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_spoorthy);
    button = (Button) findViewById(R.id.button);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick (View v){
        openActivitysa();
    }
    });
}
   public void openActivitysa() {
        Intent intent = new Intent( packageContext: this, saActivity.class);
        startActivity(intent);
    }
}
```

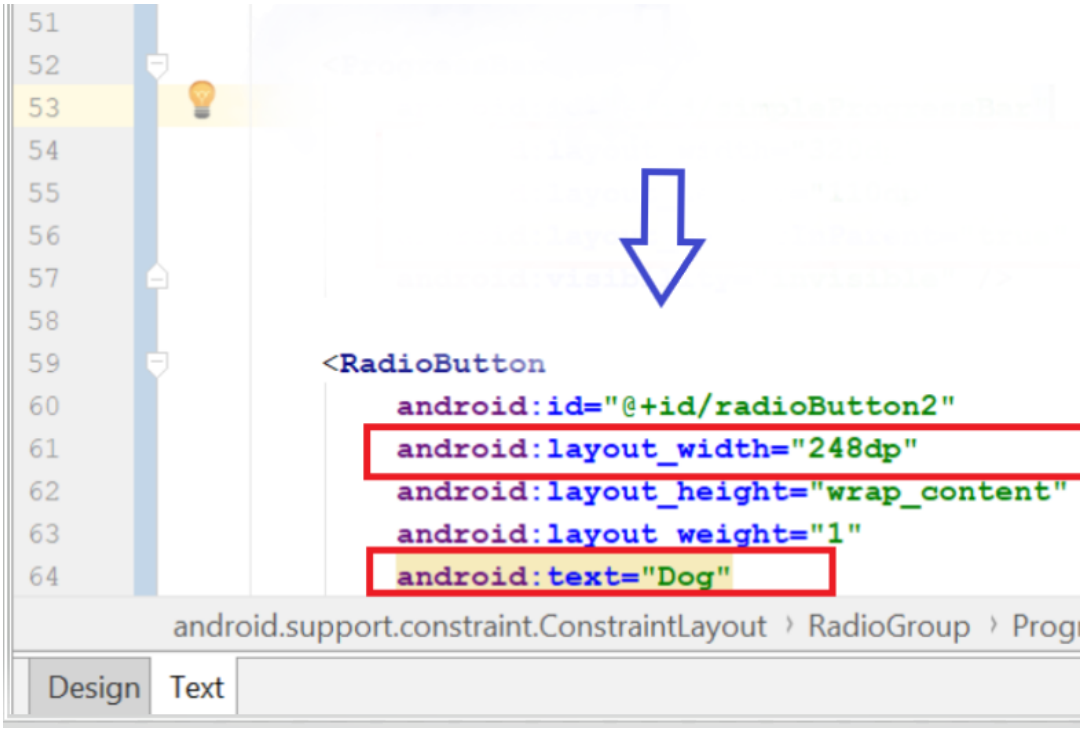
3. Declare a variable of type RadioGroup. It is initialized to the value of 'radioGroupId' that we included in our layout file (Part 2, Step 5). A variable 'id' of String type is created.

4. Add the following code. The setOnCheckedChangeListener() method will get triggered whenever the user selects one of the radio buttons. The onCheckedChange() method will handle the operations to be done when the radiobutton is selected. If radiobutton1 which is of 'cat' is selected then 'id' is initialized to 1. If radiobutton2 which is of 'dog' is selected then 'id' is initialized to 2.
*Note:The Java **switch statement** is a control statement which executes one statement from multiple conditions.*



5. The final code should look like this.

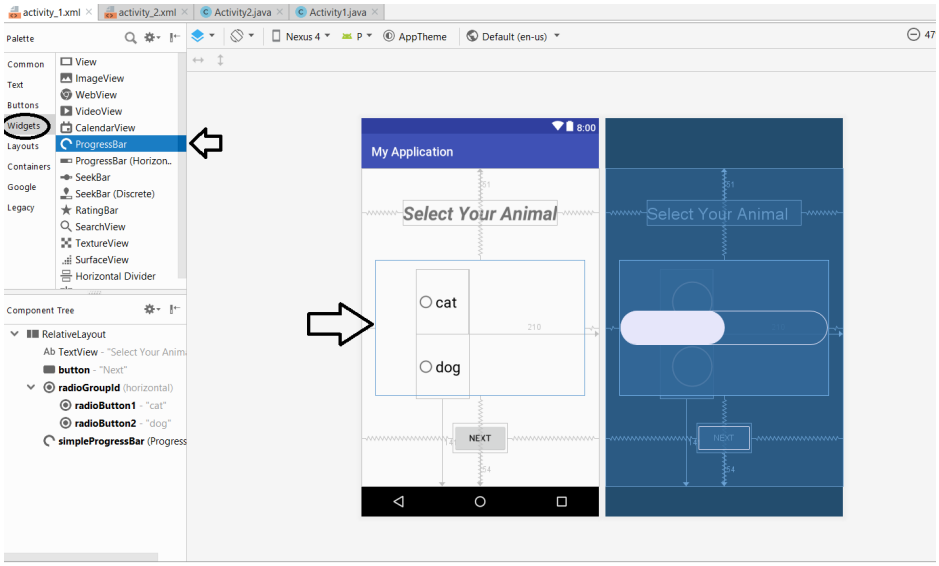
6. Now, open the text tab and make the following changes in the xml file.



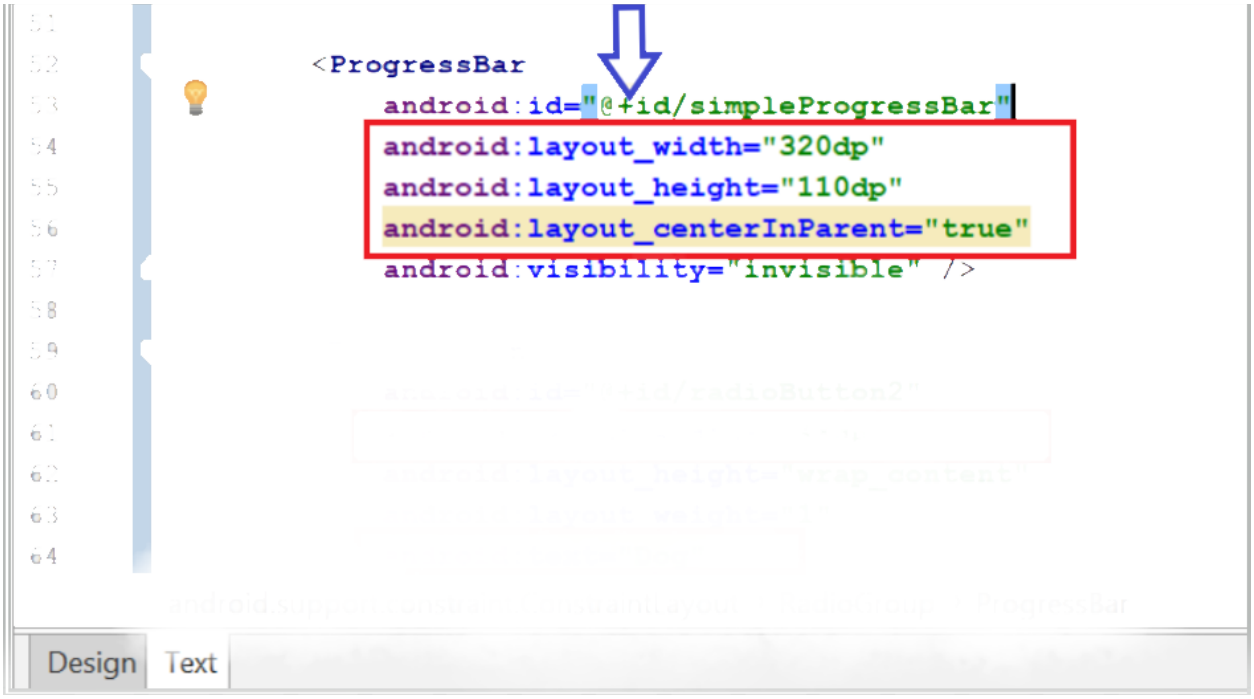
#End_Of_Part_3

Part 4: Create A Progress Bar

1. Create a button and change its label to ‘Next’.
2. Add a circular progressbar from ‘widgets’. Make sure it covers more space than the radio group as shown in the image. Give it ID as ‘simpleProgressBar’.



3. Now, open the text tab and make the following changes in the xml file.



#End_of_Part4

Part 5: Adding the Functionality of Progress Bar

1. To use progressbar, import statements are used to invoke the packages which contains ProgressBar widget.

```
package com.example.spoorthy.spoorthyapplication;

import android.content.Context;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;

public class SpoorthyActivity extends AppCompatActivity {
    private Button button;
    String id;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_spoorthy);
```

2. ProgressBar type variable 'simpleProgressBar' is initiated with the ID of progressbar used in the layout file. Since we want to see the progressbar while the second screen is being loaded, we are setting it to 'visible' state using setVisibility() method in onClick() function. Activity_Youruserid2.java is the class responsible to display the image. The decision to select an image to display depends on the value of 'id' variable. But this variable is in Activity_Youruserid1.java. Activity_Youruserid2.java will have to use this variable which is not present in its scope, to do so putExtra() method is used where 'id' can be sent to next activity which is Activity_Youruserid2.java in this case. Here 'id' is just renamed to 'SelectedRadioButton'.

```
        final ProgressBar simpleProgressBar = (ProgressBar) findViewById(R.id.simpleProgressBar);
        button = (Button) findViewById(R.id.button);

        button.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {

                simpleProgressBar.setVisibility(View.VISIBLE);
                openActivity2();
            }
        });
    }

    public void openActivity2(){

        Intent intent = new Intent( packageContext: this, Activity2.class);

        intent.putExtra( name: "SelectedRadioButton", id);
        startActivity(intent);
    }
}
```

Now we need to make changes in code of Activity_Youruserid2.java. The code looks like the one in the screenshot below initially.

saActivity.java | activity_sa.xml | SpoorthyActivity.java | activity_spoorthy.xml

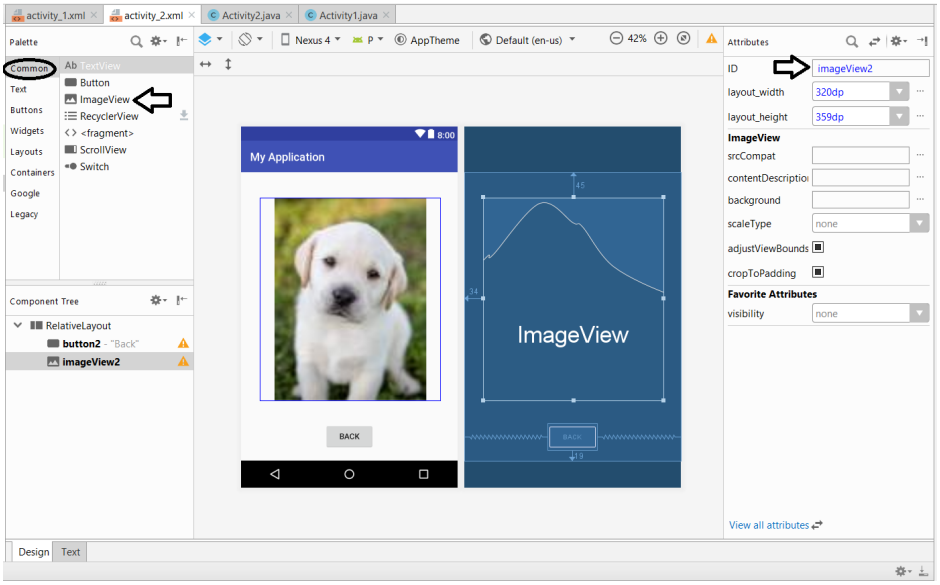
```
1  package com.example.spoorthy.spoorthyapplication;
2
3  import android.content.Intent;
4  import android.support.v7.app.AppCompatActivity;
5  import android.os.Bundle;
6  import android.view.View;
7  import android.widget.Button;
8
9
10 public class saActivity extends AppCompatActivity {
11     private Button button2;
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_sa);
17
18         button2 = findViewById(R.id.button2);
19         button2.setOnClickListener((v) → { openSpoorthyActivity(); });
25     }
26
27     public void openSpoorthyActivity() {
28         Intent intent = new Intent( packageContext: this, SpoorthyActivity.class);
29         startActivity(intent);
30     }
31
32 }
33
34
```

saActivity › openSpoorthyActivity()

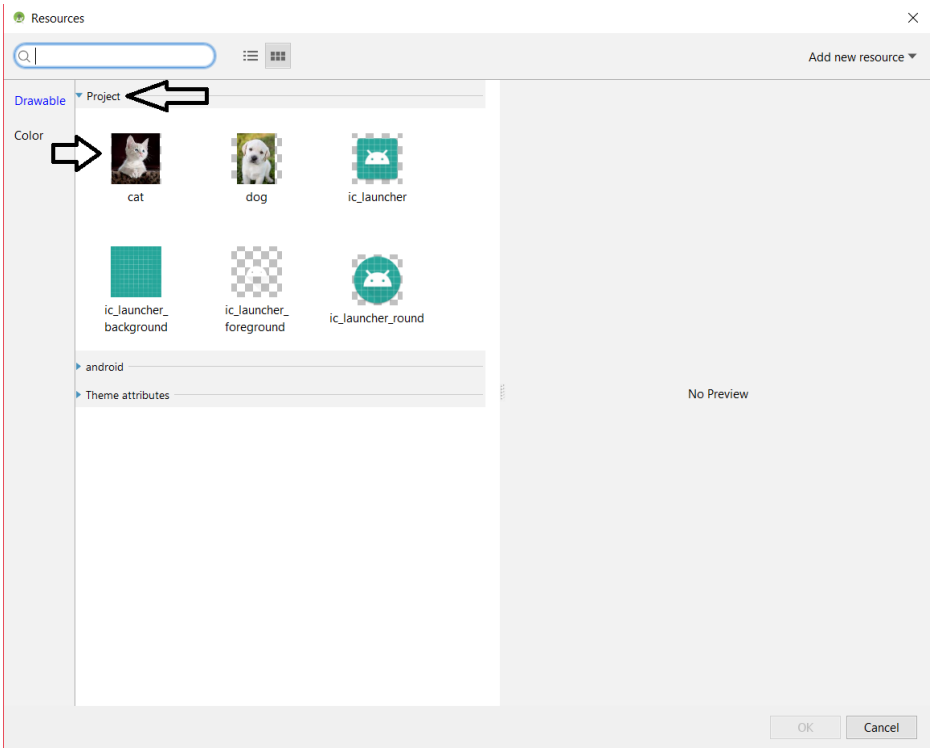
#End_Of_Part5

Part 6: Create An Image View

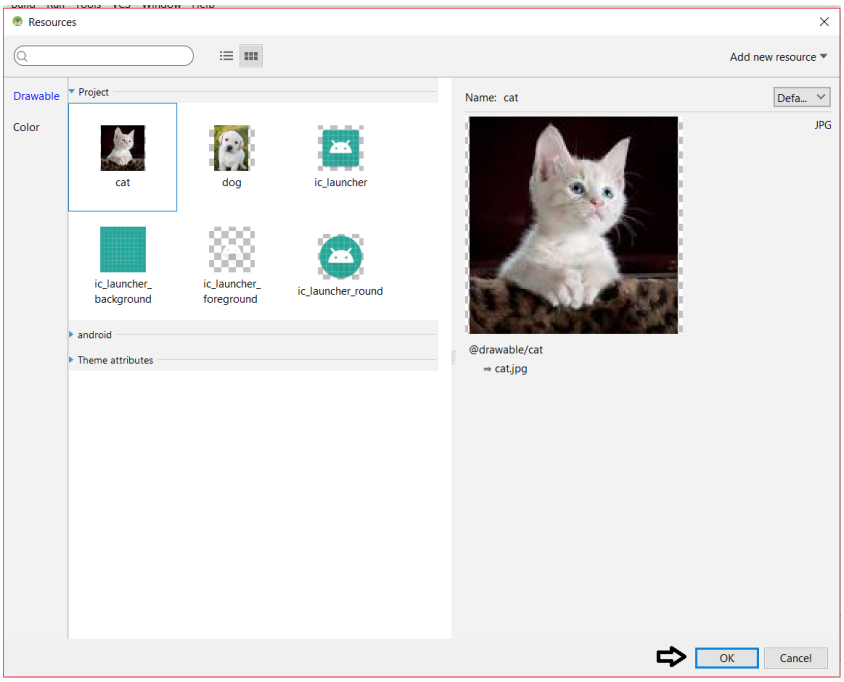
1. Now, go to your second activity ‘activity_youruserid2’.
2. Go to Palette -> Common.



3. Drag and drop the 'ImageView' on the page. You will be asked to select a resource.
4. To select the resource, expand 'Project'.



5. Select image of cat or dog.
6. Click ok.

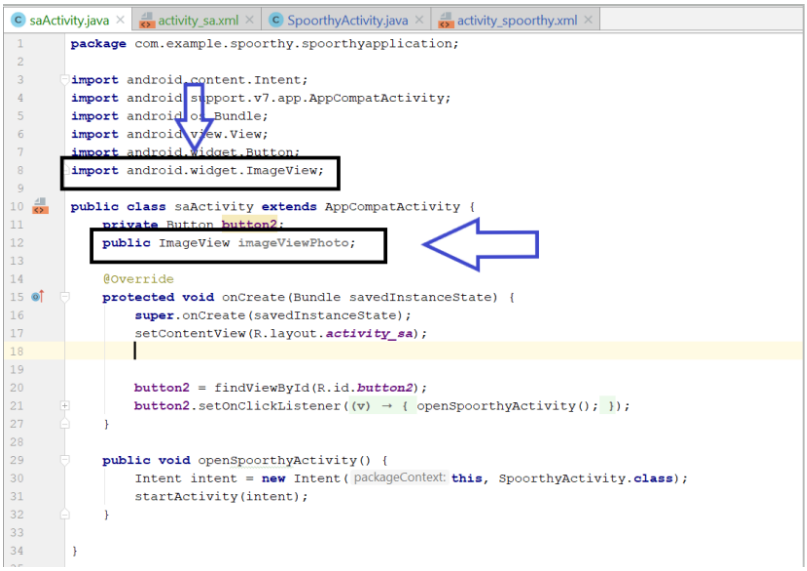


7. Resize it appropriately and set ID as 'imageView'.

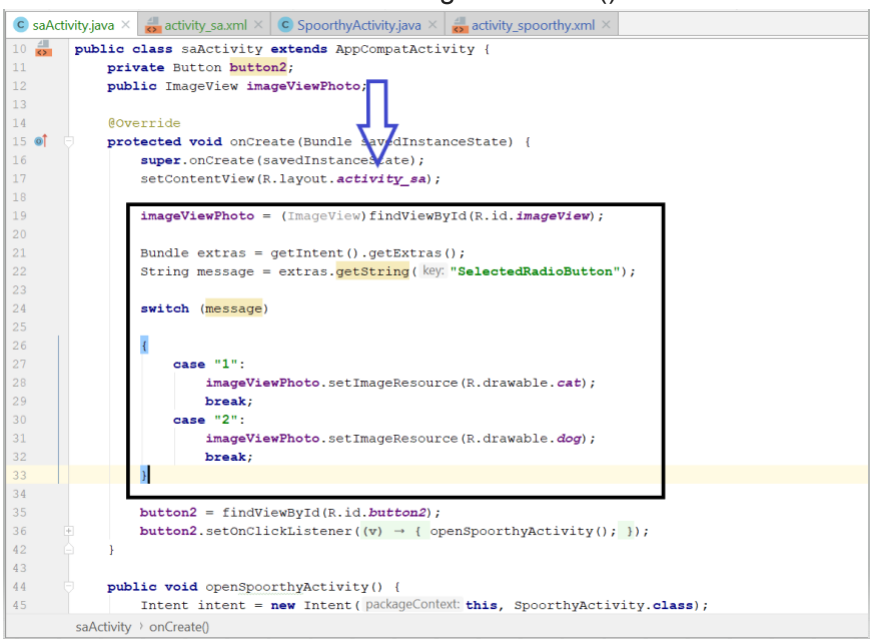
#End_Of_Part_6

Part 7: Adding the Functionality of Image View

1. Now we add the import statement for ImageView package and variable ‘imageViewPhoto’ of its type is declared.



2. The variable ‘imageViewPhoto’ is storing the id of ‘imageView2’ which we used in the Activity_Youruserid2 layout file. To use any variable from calling activity (Activity_Youruserid2), we need to use getIntent() and getExtra() methods. Variable ‘extras’ is of type ‘Bundle’ which is used to pass information between activities and other application components. Variable message stores the value of ‘SelectedRadioButton’. Here, switch statement sets the image resource for the imageView which depends on the value of selected radio button. You can see that photo of cat or dog is selected from ‘drawable’ folder in setImageResource() method.



3. Now check the IDs of imageView, radio button and progress bar by looking up at the element name in all the “findViewByID(R.id.’elementid’)” statements and the height and width of them and make sure all the changes are saved.
4. Run the application and check the application functionality.
5. Push the changes to your Github’s respective repository.

#End_Of_Part_7
#End_Of_Tutorial_6