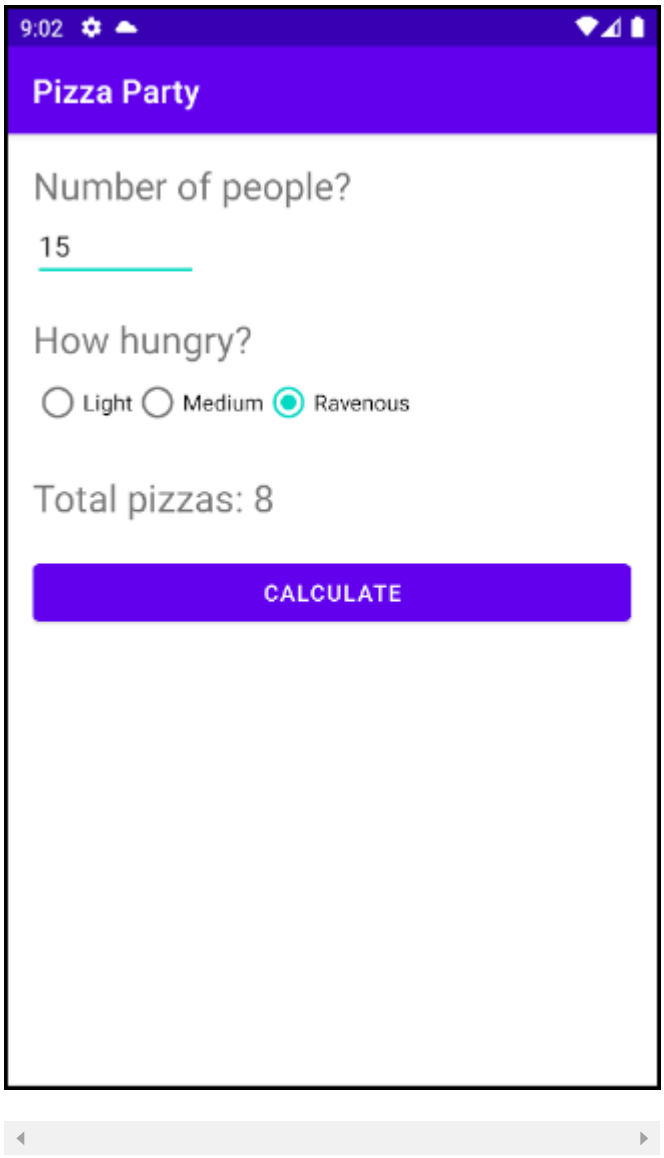
1.4 The Pizza Party app

Figure 1.4.1: Screenshot of Pizza Party app.



Feedback?

application name

An app's application name identifies the app in the Google Play Store.

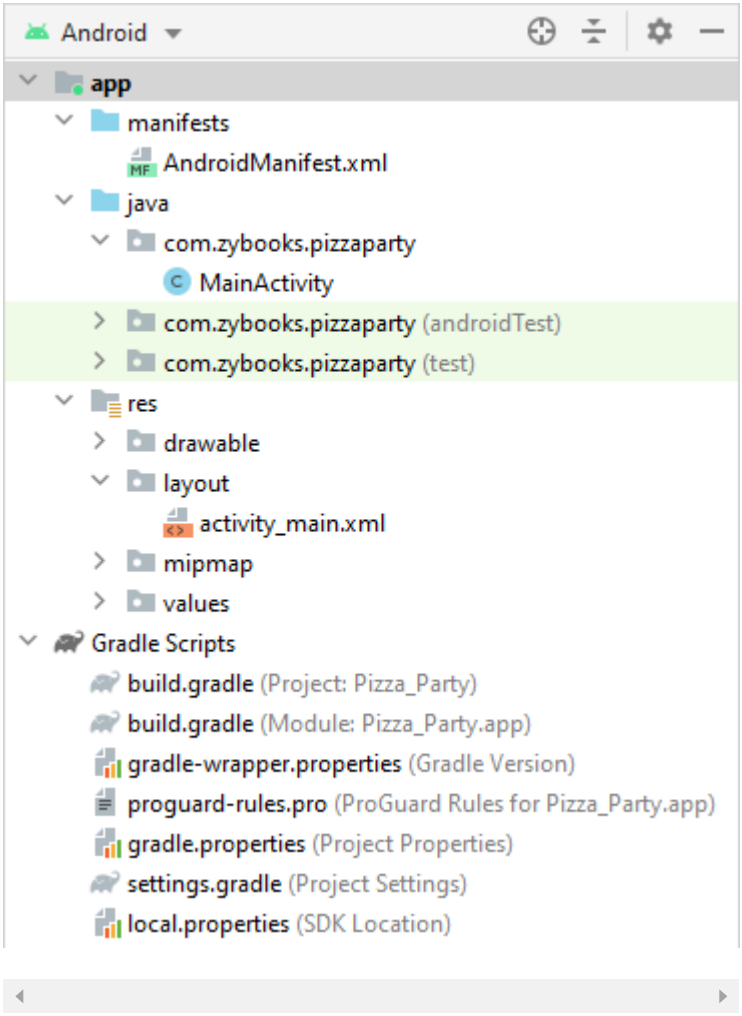
package name

The package name names a group of related Java classes.

minimum SDK

The minimum SDK is the earliest version of Android that the app supports; the lower the version, the more Android devices that are capable of running the app.

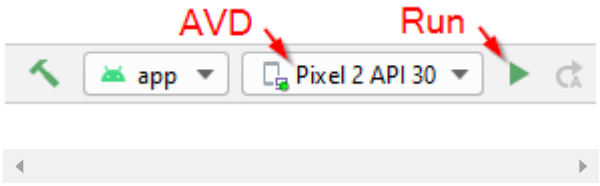
Figure 1.4.2: Android project structure.



Feedback?

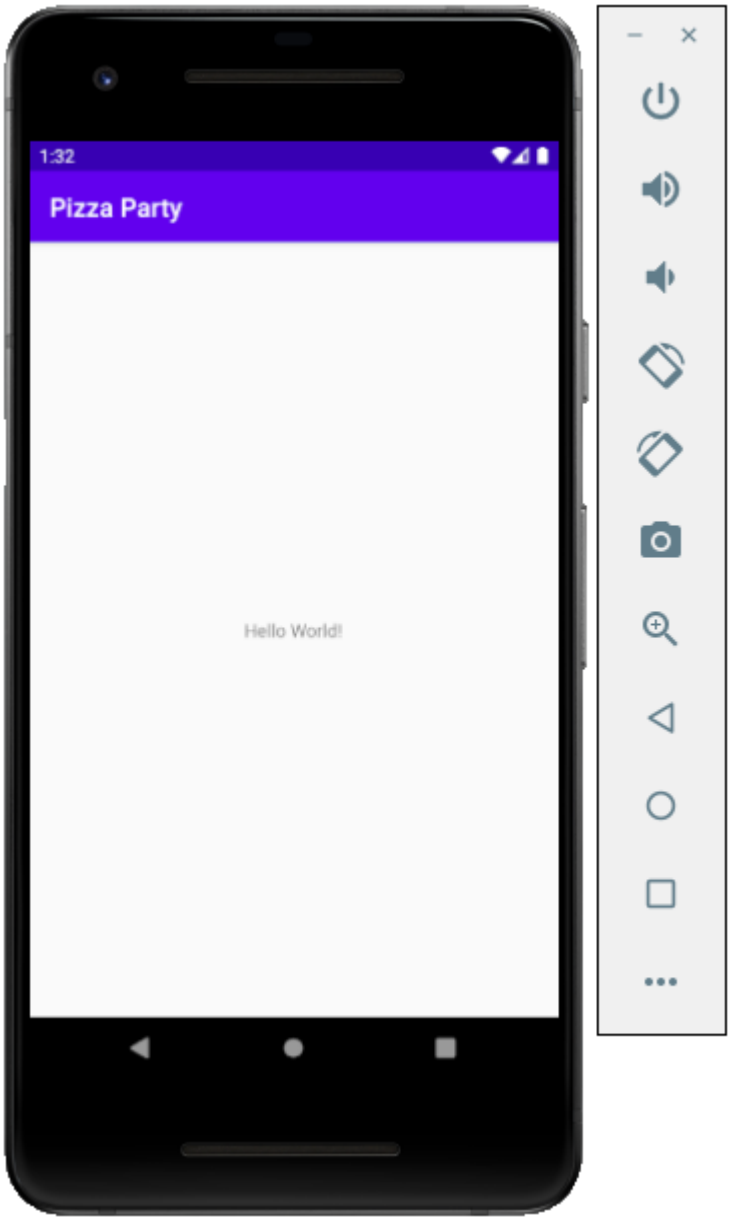
Figure 1.4.3: Run button in Android Studio's

toolbar.



Feedback?

Figure 1.4.4: Pizza Party app running on an emulator.



Feedback?

View

A View object occupies a rectangular region of the screen and is usually a widget.

widget

A widget is a UI component like a button, checkbox, radio button, slider, etc.

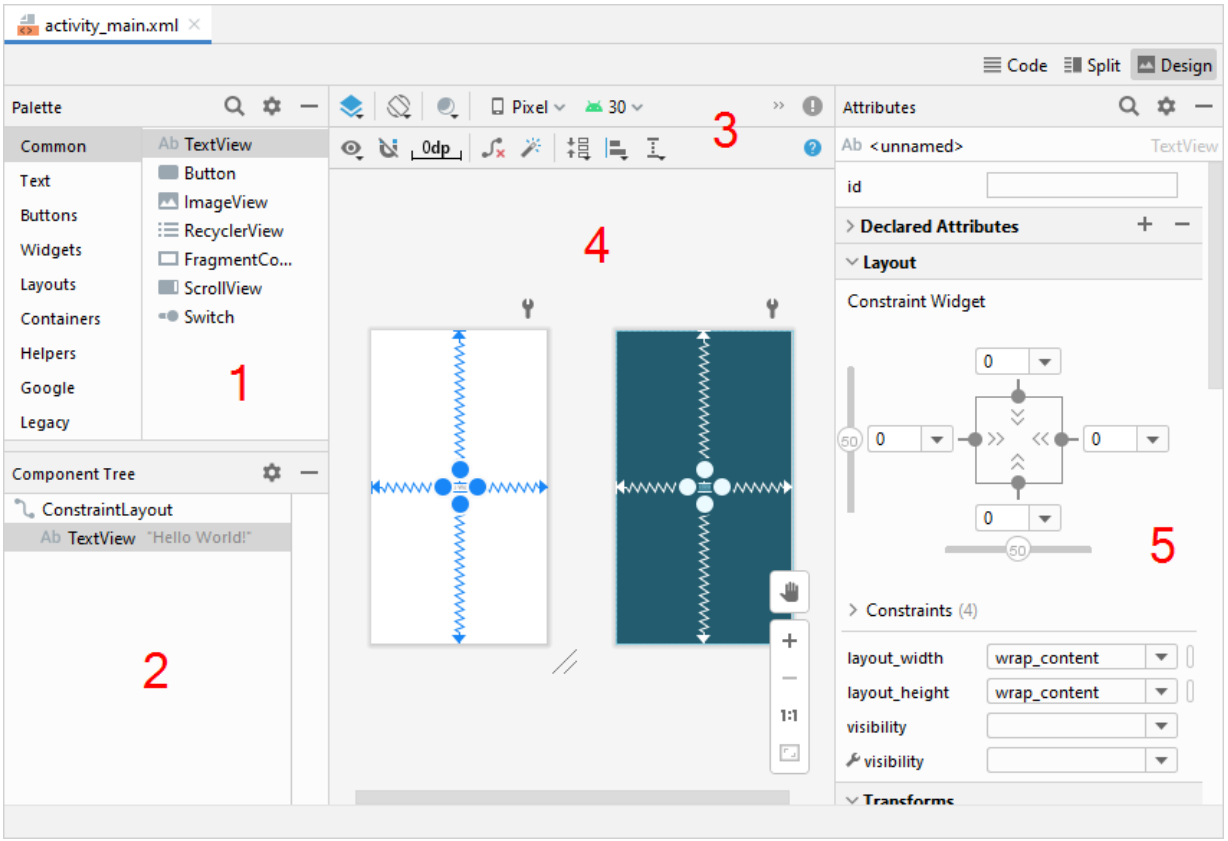
ViewGroup

A ViewGroup is a special type of View that is invisible and determines how Views should be displayed. Ex: A ViewGroup might display Views in a vertical list, in a grid, or in a fixed location.

Layout Editor

The Layout Editor allows an XML layout file to be edited.

Figure 1.4.5: Layout Editor showing a layout file in Design mode.



Feedback?

Figure 1.4.6: Layout Editor in Code mode.

activity_main.xml

Code Split Design

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Feedback?

TextView

A TextView is a widget that displays text.

Figure 1.4.7: res/layout/activity_main.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.zybooks.pizzaparty.MainActivity">
```

```
<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Number of people?"
   android:textSize="24sp"
   android:labelFor="@id/num_attend_edit_text" />

<EditText
   android:id="@+id/num_attend_edit_text"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:inputType="number"
   android:ems="5"
   android:importantForAutofill="no"
   android:hint="10" />

<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_marginTop="20dp"
   android:text="How hungry?"
   android:textSize="24sp"
   android:labelFor="@id/hungry_radio_group" />

<RadioGroup
   android:id="@+id/hungry_radio_group"
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:orientation="horizontal">
   <RadioButton
      android:id="@+id/light_radio_button"
      android:text="Light"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content" />
   <RadioButton
      android:id="@+id/medium_radio_button"
      android:text="Medium"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:checked="true" />
   <RadioButton
      android:id="@+id/ravenous_radio_button"
      android:text="Ravenous"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content" />
</RadioGroup>

<TextView
   android:id="@+id/num_pizzas_text_view"
   android:text="Total pizzas: ?"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_marginTop="20dp"
   android:textSize="24sp"/>

<Button
   android:id="@+id/calc_button"
   android:text="Calculate"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
```

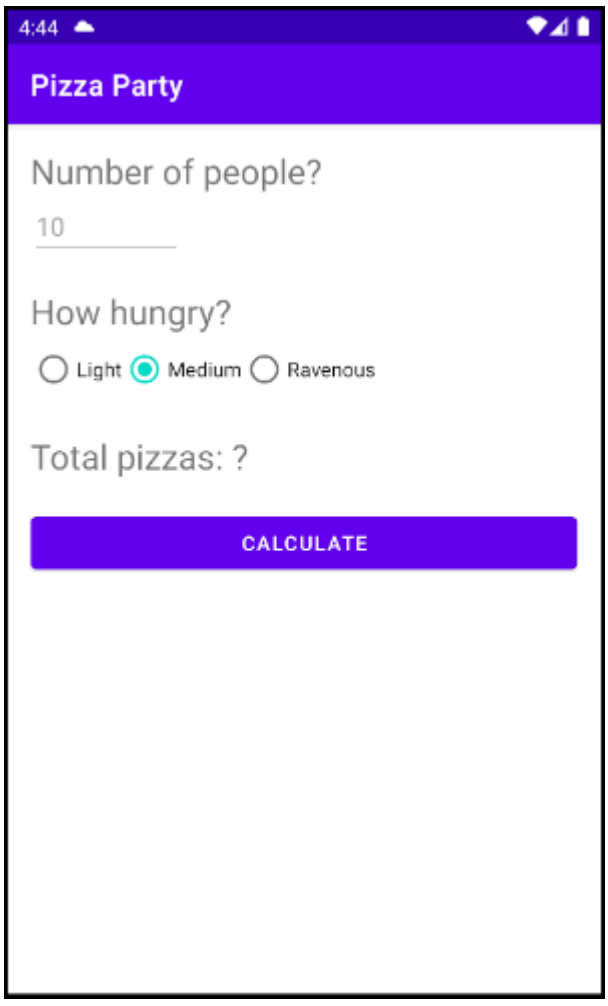
```
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:onClick="calculateClick" />
</LinearLayout>
```

Feedback?

LinearLayout

LinearLayout is a ViewGroup that displays all child Views vertically or horizontally using the `android:orientation` attribute.

Figure 1.4.8: Pizza Party app running on an emulator.



Feedback?

AppCompatActivity

AppCompatActivity is the superclass for all activities and provides the latest Android functionality for devices running older versions of Android.

Activity

`AppCompatActivity` extends the Activity class, which provides methods that Android calls to create, start, stop, and destroy the activity.

onCreate()

The `Activity` method onCreate(), the first method called when the activity starts, loads the activity's XML layout and performs other initialization logic.

setContentView()

The `Activity` method setContentView() sets the activity's content to the given layout file.

findViewById()

The `Activity` method findViewById() returns a View from the layout file that matches the given ID.

PARTICIPATION ACTIVITY 1.4.4: Running the MainActivity.

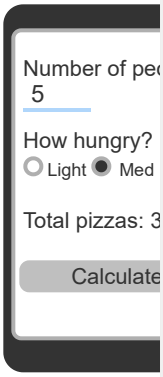
Start 2x speed

```
package com.zybooks.pizzaparty;

import android.os.Bundle;
```

activity_mair

```
import android.view.View;
import android.widget.EditText;
import android.widget.RadioGroup;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

   public final int SLICES_PER_PIZZA = 8;

   private EditText mNumAttendEditText;
   private TextView mNumPizzasTextView;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);
       mNumAttendEditText = findViewById(R.id.num_attend_edit_text);
       mNumPizzasTextView = findViewById(R.id.num_pizzas_text_view);
   }

   public void calculateClick(View view) {
       String numAttendStr = mNumAttendEditText.getText().toString();
       int numAttend = Integer.parseInt(numAttendStr);

       int slicesPerPerson = 4;
       int totalPizzas = (int) Math.ceil(numAttend * slicesPerPerson /
          (double) SLICES_PER_PIZZA);
       mNumPizzasTextView.setText("Total pizzas: " + totalPizzas);
   }
}
```

```
<LinearLay
  <TextVie
  <EditTex
etc...
  <TextVie
</LinearLa
```



Captions

1. MainActivity uses the package com.zybooks.pizzaparty and imports various classes from the Android API. MainActivity extends the AppCompatActivity class, which displays the UI and processes user input.
2. The onCreate() method is called when MainActivity first starts, and setContentView() sets the MainActivity's content to the layout in activity_main.xml
3. findViewById() returns an object representing the widget from activity_main.xml that matches the given ID.
4. When the user types a number and presses the Calculate button, calculateClick() is called.
5. getText() returns the editable text, and toString() converts the editable text into a string. Integer.parseInt() converts the string into an integer, and setText() changes the text of the TextView.

Feedback?

Figure 1.4.9: Handling the radio button selection.

```
package com.zybooks.pizzaparty;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.RadioGroup;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

   public final static int SLICES_PER_PIZZA = 8;

   private EditText mNumAttendEditText;
   private TextView mNumPizzasTextView;
   private RadioGroup mHowHungryRadioGroup;

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

       // Assign the widgets to fields
       mNumAttendEditText = findViewById(R.id.num_attend_edit_text);
       mNumPizzasTextView = findViewById(R.id.num_pizzas_text_view);
       mHowHungryRadioGroup = findViewById(R.id.hungry_radio_group);
   }

   public void calculateClick(View view) {

       // Get the text that was typed into the EditText
       String numAttendStr = mNumAttendEditText.getText().toString();

       // Convert the text into an integer
       int numAttend = Integer.parseInt(numAttendStr);

       // Determine how many slices on average each person will eat
       int slicesPerPerson = 0;
       int checkedId = mHowHungryRadioGroup.getCheckedRadioButtonId();
       if (checkedId == R.id.light_radio_button) {
           slicesPerPerson = 2;
       }
       else if (checkedId == R.id.medium_radio_button) {
           slicesPerPerson = 3;
       }
       else if (checkedId == R.id.ravenous_radio_button) {
           slicesPerPerson = 4;
       }

       // Calculate and show the number of pizzas needed
       int totalPizzas = (int) Math.ceil(numAttend * slicesPerPerson /
(double) SLICES_PER_PIZZA);
       mNumPizzasTextView.setText("Total pizzas: " + totalPizzas);
   }
}
```

Feedback?

Try 1.4.1: Project quick fix.

Android Studio has over 100 keyboard shortcuts to improve developer productivity. One of the most commonly used keyboard shortcuts is the ***project quick fix***: Alt+Enter on Windows or Option+Enter on a Mac.

Try the following:

1. Delete the import statement below from MainActivity.java:

   ```
   import android.widget.RadioGroup;
   ```

2. The line declaring `mHowHungryRadioGroup` should now display `RadioGroup` in red because the import statement for the `RadioGroup` class is missing.
3. Click on the red `RadioGroup`, and a hint should appear.
4. Press Alt+Enter on Windows or Option+Enter on a Mac, and the import statement will automatically re-appear.

The project quick fix shortcut is helpful when adding code from this material or code found online when the import statement is not given. If Android Studio finds more than one possible import statement, the developer is prompted to select which import statement to add.

Feedback?

This section does not contain presentation elements.

How was this section? Provide feedback