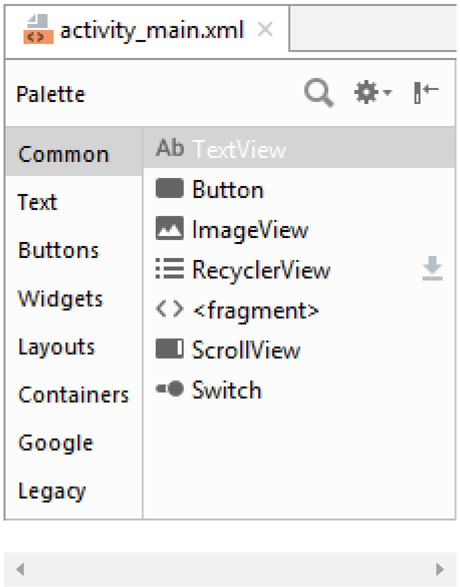
2.4 Widgets and event handling

Figure 2.4.1: Widgets displayed in the Layout Editor.



Feedback?

event

An event represents an action that occurs or is "triggered" because of the user's interaction with the widget.

event listener

An event listener is an interface in the `View` class that contains at least one callback method.

callback

A callback is a method that Android calls when an event is triggered.

toast

A toast is a short message that appears momentarily.

Toast.makeText()

The static method Toast.makeText() creates a `Toast` object using a `Context`, text message, and duration.

show()

The `Toast` method show() displays the toast.

Context

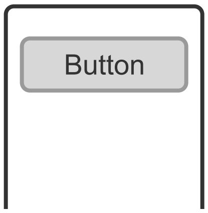
The Context class is an abstract class that provides global information about an application or app component. A number of important classes like `Activity`, `Application`, and `Service` subclass `Context`.

PARTICIPATION ACTIVITY 2.4.2: Callback method onClick() displays a toast.

Start □ 2x speed

```
public class MainActivity extends AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_main);

     Button button = findViewById(R.id.myButton);
     button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
           Toast.makeText(MainActivity.this, "Button click", Toast.LENGTH_LONG).show();
        }
     });
  }
}
```



Captions

1. findViewById() returns a Button with ID myButton, defined in MainActivity's layout file.
2. The Button class inherits the setOnClickListener() method from the View class. setOnClickListener() takes a View.OnClickListener argument.
3. An anonymous implementation of the OnClickListener interface implements the callback method onClick(). onClick's view parameter references the view (the button) that is clicked.
4. When the button is clicked, Android calls onClick(). makeText() creates a Toast object, and show() displays the toast for a few seconds.

Feedback?

android:onClick

An XML layout may use the android:onClick XML attribute to specify a widget's click callback without having to implement a `View.OnClickListener`.

Figure 2.4.2: Click callback named in layout.

Layout XML

```
<Button ...
   android:onClick="clickCallback" />
```

Callback in MainActivity

```
public void clickCallback(View view) {
   // button was clicked
}
```

Feedback?

lambda expression

A lambda expression is a code block with parameters and an optional return value.

Construct 2.4.1: Lambda expression

declaration.

```
(parameter1, parameter2) -> { code block }
```

Feedback?

PARTICIPATION ACTIVITY

2.4.5: Lambda expression implements onClick() callback method.

Start □ 2x speed

```
button.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View v) {
      Toast.makeText(MainActivity.this, "Button click", Toast.LENGTH_LONG).show();
   }
});
```

Lambda expression

```
button.setOnClickListener(
   (v) -> {
      Toast.makeText(MainActivity.this, "Button click", Toast.LENGTH_LONG).show()
   }
);
```

Simplified

```
button.setOnClickListener(
   v -> Toast.makeText(MainActivity.this, "Button click", Toast.LENGTH_LONG).show());
```

Captions

1. An anonymous implementation of the OnClickListener interface implements the callback method onClick().
2. A lambda expression implements the onClick() callback with much less code.
3. Parameter v is the View. The code block in { } executes when the button is clicked.
4. When a lambda expression has a single parameter, the ( ) around the parameter can be omitted.
5. When the lambda expression contains a single line of code, { } can be omitted from the code block.

Feedback?

This section does not contain presentation elements.

How was this section? 👍 👎 Provide feedback