
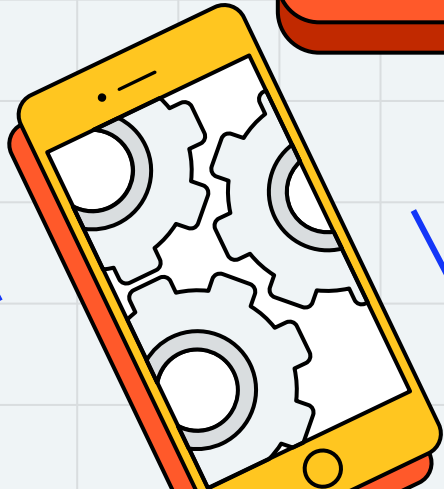




Application

Programming




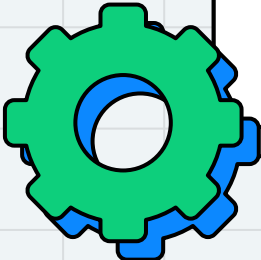
Hend Alkittawi





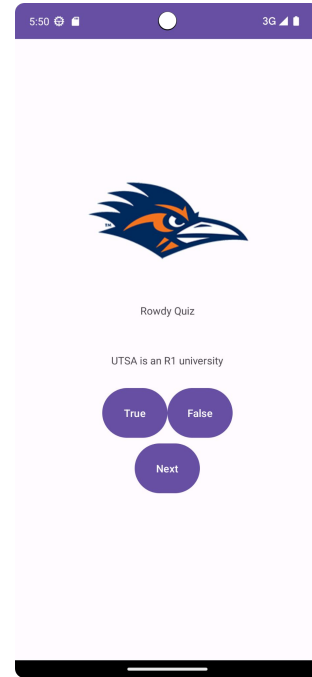
Android Development

Building a Multi-screen App



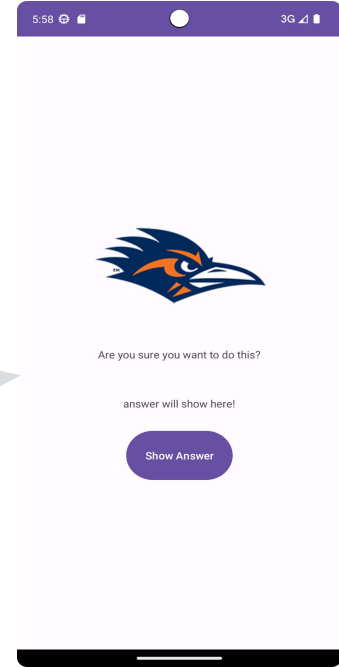
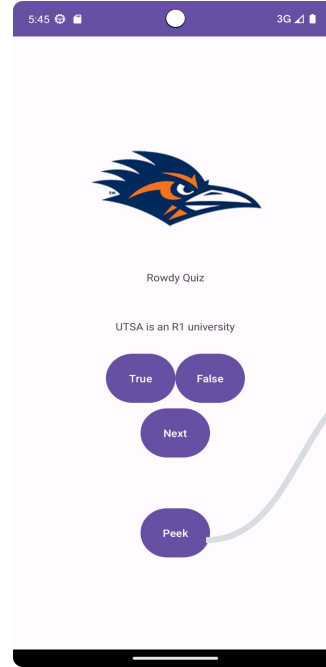
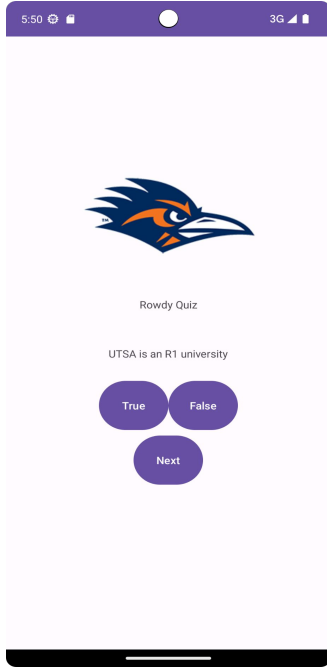
ROWDY QUIZ APP

- We created the Rowdy Quiz app to read questions from a data file and display them to the user.
- We also added a logo to our screen



ROWDY QUIZ APP

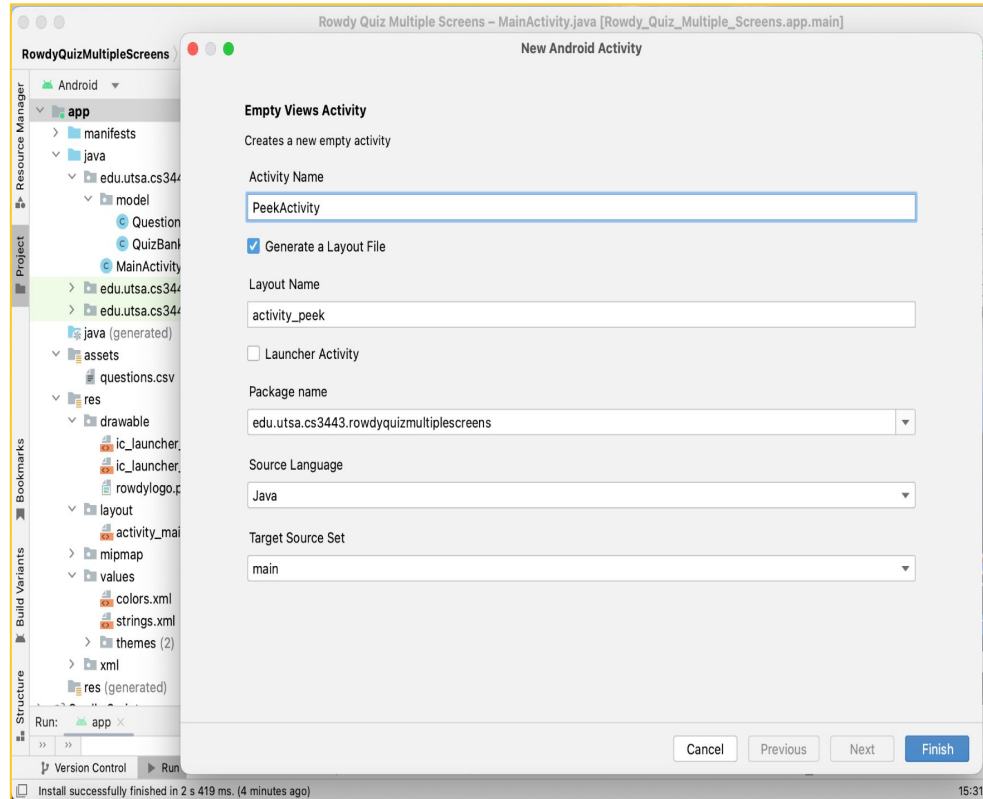
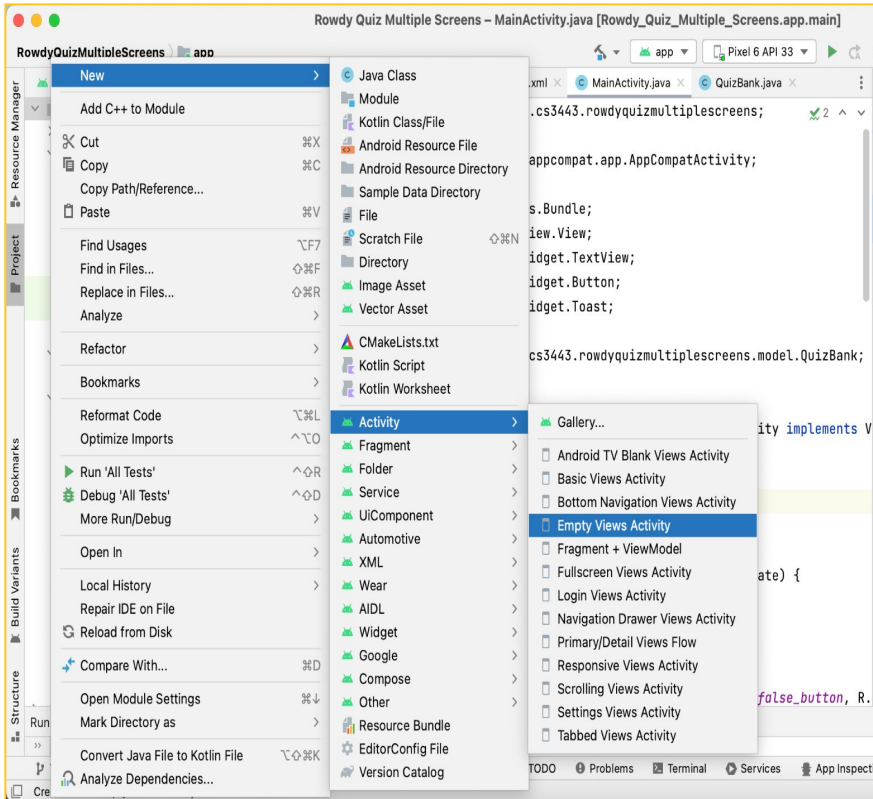
- We will modify the app to allow the user to peek at the answer



BUILDING A MULTI-SCREEN APP

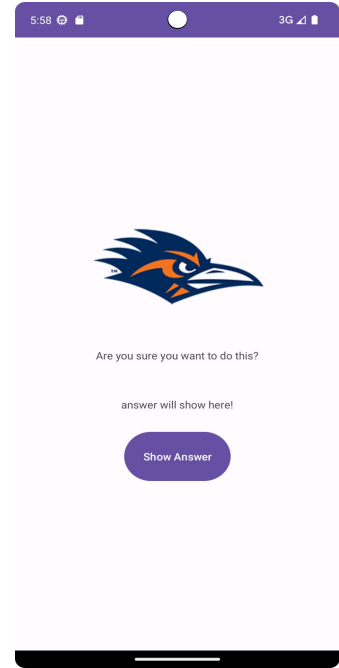
- To add a new screen to an Android app
 1. create a new activity and a new layout using the new activity wizard
 - Right click on app > New > Activity > Empty Views Activity
 - this will **create a new layout** file and **update the AndroidManifest.xml** file
 - the manifest is an xml file that contains metadata that describes your application to the Android OS
 - every activity in an application must be declared in the manifest so that the OS can access it
 2. add code to start the new activity

CREATING A NEW ACTIVITY



CREATING A NEW ACTIVITY

- Create a new activity and a new layout using the [new activity wizard](#)
 - The layout for the new screen has
 - LinearLayout
 - ImageView
 - TextView
 - TextView
 - Button



STARTING A NEW ACTIVITY

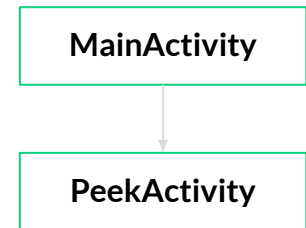
- The simplest way to start another activity is with the `startActivity()` method

```
Intent intent = new Intent(currentActivity, newActivity.class);
startActivity(intent)
```

- For our app, MainActivity will start the PeekActivity in response to a click on the *Peek* button

- Add the following method to MainActivity.java and call it as appropriate!

```
private void launchActivity() {
    Intent intent = new Intent(this, PeekActivity.class);
    startActivity(intent);
}
```



STARTING A NEW ACTIVITY

- Sometimes it is necessary to **pass data** between the starting activity (parent) and the started activity (child).
- When starting an activity, the **starting (parent) activity** can add an **extra** to the intent with the **putExtra()** method
 - extras are arbitrary data that the calling activity can include with an intent
 - an extra is a structured **key-value pair**

```
Intent intent = new Intent(currentActivity, newActivity.class);  
intent.putExtra(name, value);  
startActivity(intent)
```

STARTING A NEW ACTIVITY

- Sometimes it is necessary to **pass data** between the starting activity (parent) and the started activity (child)
 - to retrieve the extra from the intent use the **get[type]Extra()** method

```
value = getIntent().get[type]Extra(name, default_val);
```

- available methods
 - getBooleanExtra()
 - getCharExtra()
 - getIntExtra()
 - getStringExtra() ... etc.

STARTING A NEW ACTIVITY

- For our app, MainActivity should pass the answer of the current question to the PeekActivity so that the answer is displayed to the user when the Show Answer button is clicked

```

// some code is omitted .. new code is highlighted here
// refer to full code on Github

public class MainActivity extends AppCompatActivity {

    private static final String intentKey = "answer";

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

        Button peekButton = findViewById(R.id.peek_button);
        peekButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                launchActivity();
            }
        });
    }

    private void launchActivity() {
        Intent intent = new Intent(this, PeekActivity.class);
        intent.putExtra(intentKey, String.valueOf(getAnswer()));
        startActivity(intent);
    }

    public static String decodeIntent(){
        return intentKey;
    }
}

```

```

// some code is omitted
// refer to full code on Github

public class PeekActivity extends AppCompatActivity {

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

        Button button = findViewById(R.id.show_answer_button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String answer =
                    getIntent().getStringExtra(MainActivity.decodeIntent());

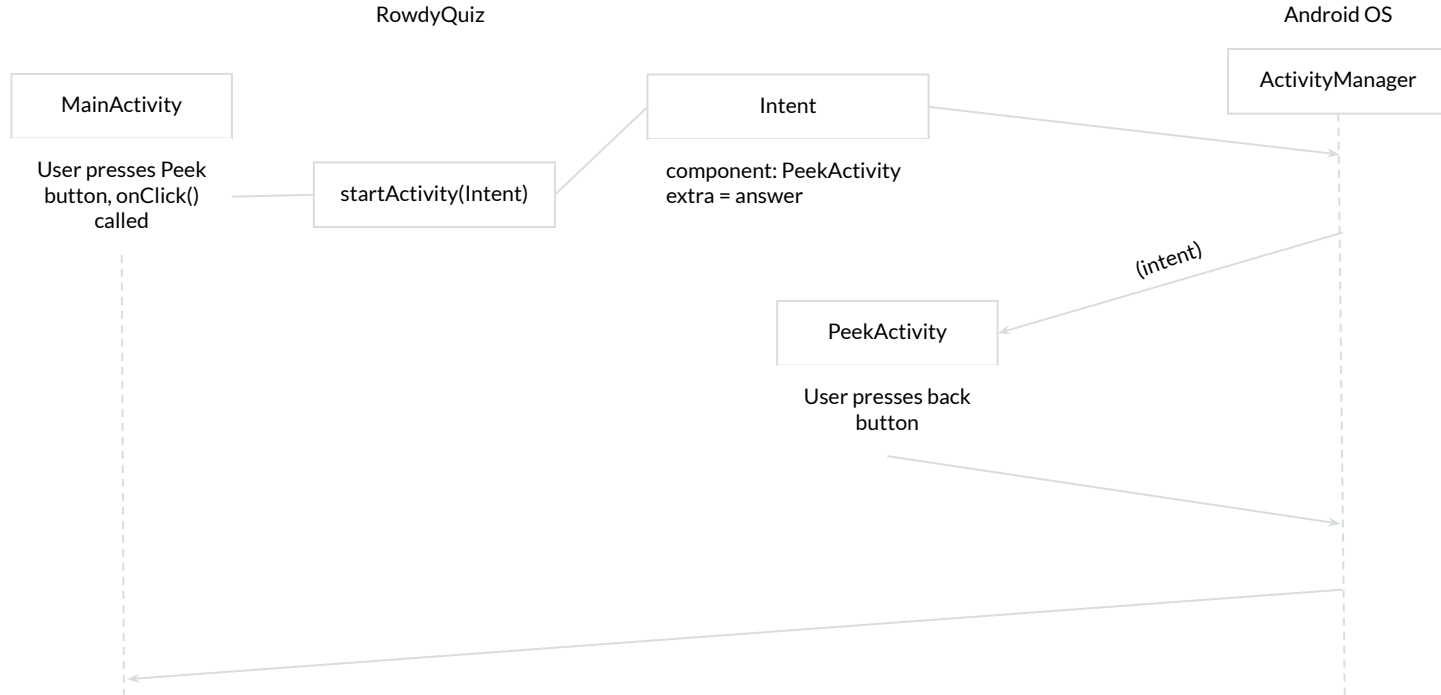
                getIntent().getStringExtra("answer");

                TextView answerTextView = findViewById(R.id.shown_answer);
                answerTextView.setText(answer);
            }
        });
    }
}

```

BUILDING A MULTI-SCREEN APP

- Rowdy Quiz Sequence Diagram





THANK

YOU!



DO YOU HAVE ANY QUESTIONS?



hend.alkittawi@utsa.edu



By Appointment



Online