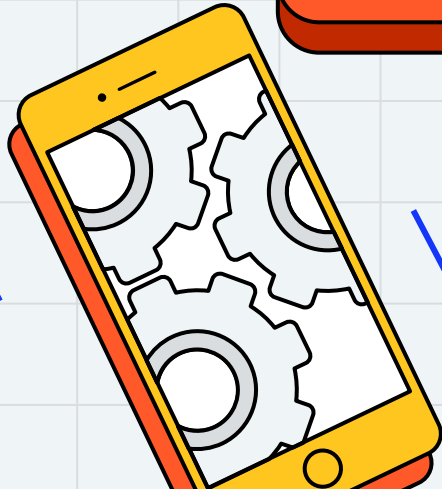


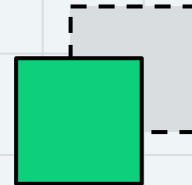


Application

Programming



Hend Alkittawi





Leaderboard App

An App to Demo File IO, Exception
Handling, Scroll Views and Dynamic
Image Loading

ANDROID APP DEVELOPMENT

- Roadmap
 - Gather **requirements**
 - Design **views**
 - (layout xml files - setup views ids)
 - Design **model** classes
 - (classes to represent app data)
 - Design **controller**
 - (classes to connect the views and model classes with methods to handle user interactions and to manage the data flow and update views)
 - Create your **data files**

ANDROID APP DEVELOPMENT

- At this point we have most of the **tools** needed to create an application.
- To create an application, start by **gathering requirements** based on which **tasks and subtasks can be identified**.
- Important questions to ask
 - What is the main purpose of the application?
 - How will the user interact with it?
 - What technology will we use?

WRITING DATA TO A FILE

- The Assets folder is read-only, we cannot write data to the files in the Assets folder.
- What if we need to modify data within a file located in the Assets folder or create a file for both reading and writing data?
 - A work around this is to create a copy of the file from the Assets to the AVD memory, then use the "copy file" for reading and writing data.
- To **write** data to a file, create a file in the **AVD memory**.
- Once created, the file can be located in the AVD memory by navigating to View > Tool Windows > Device Explorer then data > user > 0 > edu.utsa.cs3443.*projectName* > files

SCROLL VIEWS

- A ScrollView is a view group that **allows the view hierarchy placed within it to be scrolled.**
- A ScrollView may have only one direct child placed within it.
- To add multiple views within the ScrollView, make the direct child you add a view group, for example LinearLayout, and place additional views within that LinearLayout.

```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</ScrollView>
```

IMAGE VIEWS

- An ImageView can display an image from the drawable folder
 - statically (hard-code)

```
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/rowdylogo"/>
```

- dynamically (xml and controller class)

```
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/dynamic_image" />
```

```
public void displayImage(String filename){  
    ImageView dynamicImage = (ImageView) findViewById(R.id.dynamic_image);  
    int imageResource = getResources().getIdentifier(filename, "drawable", getPackageName());  
    dynamicImage.setImageResource(imageResource);  
}
```

DYNAMIC LOADING OF VIEWS

- MainActivity can dynamically place buttons in its layout (screen)

```
private void dynamicSetupButton(String player) {  
    // create a layout object  
    LinearLayout rootLayout = (LinearLayout) findViewById(R.id.root_layout);  
    Button myButton = new Button(this);  
    myButton.setText(player);  
    myButton.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) { Log.i(TAG, "Clicked on " + myButton.getText()); }  
    });  
    // setup the attributes for the button  
    LinearLayout.LayoutParams buttonAttributes  
        = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,  
        LinearLayout.LayoutParams.WRAP_CONTENT);  
    // add the button to the layout  
    rootLayout.addView(myButton, buttonAttributes);  
}
```




THANK

YOU!



DO YOU HAVE ANY QUESTIONS?



hend.alkittawi@utsa.edu



By Appointment



Online