Menus

Upon completion of this module, a student will be able to

- understand and explain where menus are used
- define a menu in XML
- add an options menu to the action bar
- respond to options menu selections
- define a navigation drawer in XML
- enable and work with a navigation drawer in Java

Assignment

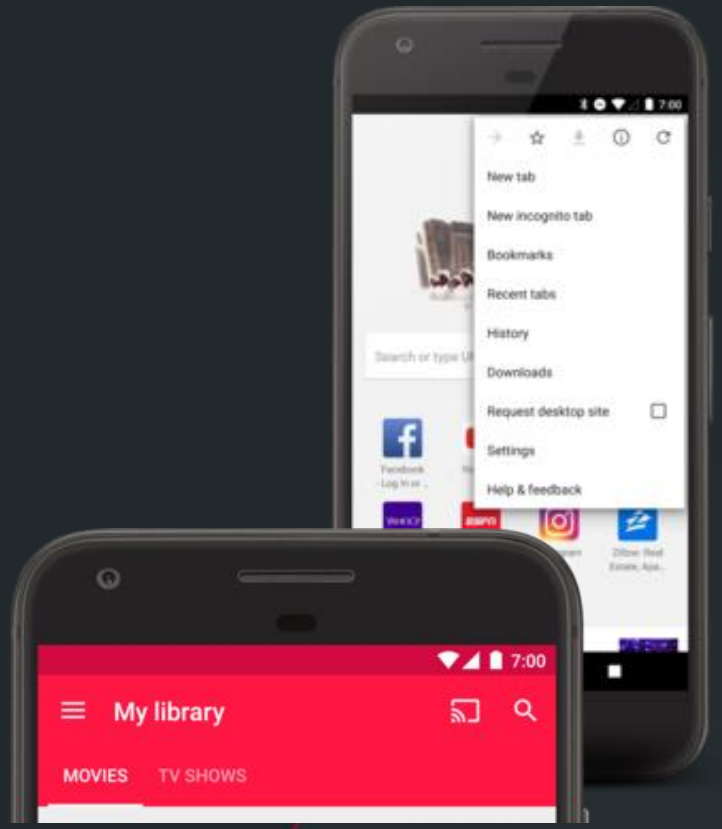
- Task
  - For this project, you'll design an app and then implement the navigation drawer and action bar menu for that app.
- Repo
  - https://github.com/LambdaSchool/AndroidMenusDesign
- Submission
  - Fork on github and submit pull request

A Student Can

understand and explain where menus are used

Menus

- Options menu
- Context Menu
- Popup Menu



A Student Can

define a menu in XML

Menu vs Layout

- Menu resource
  - Used in multiple contexts
  - Lays out differently in each
- Layout
  - Used in Activities or Fragments
  - More strict in structure and look

Elements and Attributes

- Menu
  - Container for items
- Item
  - MenuItem
- Group
  - Container of items

- icon
  - Menu icon image
- showAsAction
  - How it should appear in the action bar
- checkable
- checkableBehavior

A Student Can

add an options menu to the action bar

Adding Options Menu

- Override onCreateOptionsMenu
- Menu Inflater
- Inflate menu resource

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.game_menu, menu);
    return true;
}
```

A Student Can

respond to options menu selections

Handling Click

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
        case R.id.new_game:
            newGame();
            return true;
        case R.id.help:
            showHelp();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

- Override onOptionsItemSelected
- Perform action based on item selected

A Student Can

define a navigation drawer in XML

Layout

- DrawerLayout – Parent (2 Children)
  1. Activity content
  2. Navigation drawer layout

```
<android.support.v4.widget.DrawerLayout
    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:id="@+id/drawer_layout"
    ...
    tools:context=".MainActivity">

    <!-- Layout to contain contents of main body of screen (drawer will slide over this) -->
    <android.support.constraint.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <!-- Activity Content -->

    </android.support.constraint.ConstraintLayout>

    <!-- Container for contents of drawer - use NavigationView to make configuration easier -->
    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        ...
        android:fitsSystemWindows="true" />

</android.support.v4.widget.DrawerLayout>
```

Nav Drawer Content

- Menu resource

```
<android.support.design.widget.NavigationView
    android:id="@+id/nav_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:fitsSystemWindows="true"
    app:menu="@menu/options_menu" />
```

Header

- app:headerLayout="@layout/nav_header"

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    ...
    android:layout_height="@dimen/nav_header_height"
    android:background="@drawable/side_nav_bar"
    android:gravity="bottom"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <ImageView
        android:id="@+id/imageView"
        ...
        android:contentDescription="@string/nav_header_desc"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        app:srcCompat="@mipmap/ic_launcher_round" />

    <TextView
        ...
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:text="@string/nav_header_title"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1" />

    <TextView
        android:id="@+id/textView"
        ...
        android:text="@string/nav_header_subtitle" />

</LinearLayout>
```

A Student Can

enable and work with a navigation drawer in Java

Toolbar

```
<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.AppBarOverlay">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/AppTheme.PopupOverlay" />

</android.support.design.widget.AppBarLayout>
```

- setSupportActionBar(toolbar);

Add Toggle Button

- Create toggle object
- Add toggle to toolbar
- Bind drawer and toggle

```
DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
        this, drawer, toolbar,
        R.string.openDrawerContentDescRes,
        R.string.closeDrawerContentDescRes);
drawer.addDrawerListener(toggle);
toggle.syncState();
```

Handle Click Events

```
NavigationView navigationView = findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(
    new NavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelected(MenuItem menuItem) {
            // set item as selected to persist highlight
            menuItem.setChecked(true);
            // close drawer when item is tapped
            mDrawerLayout.closeDrawers();

            // Add code here to update the UI based on the item selected
            // For example, swap UI fragments here

            return true;
        }
    });
```

- OnNavigationItemSelectedListener