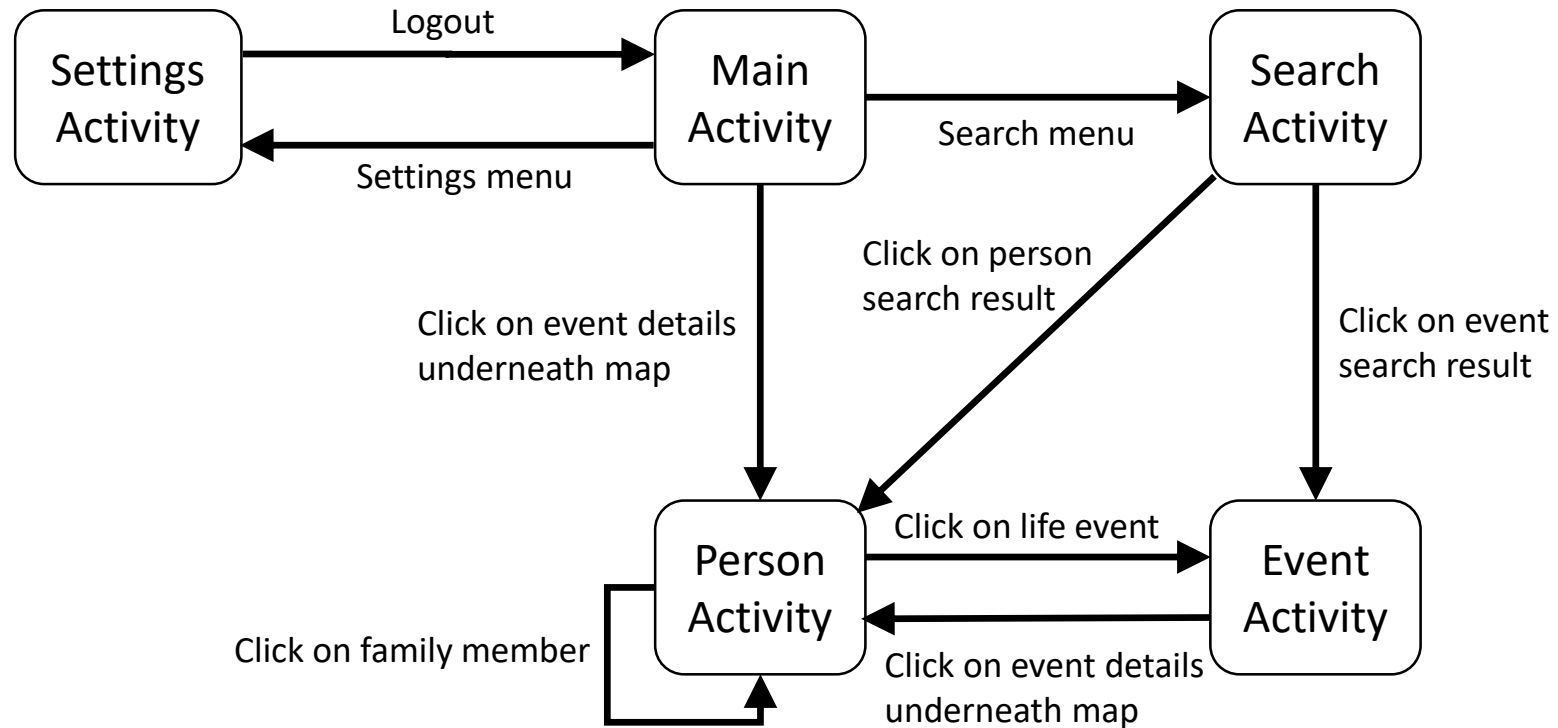


Navigation, Toolbar, Menus

Navigation in Family Map Client

Back button can always be used to return to the previous activity

Up button can always be used to return all the way back to Main Activity

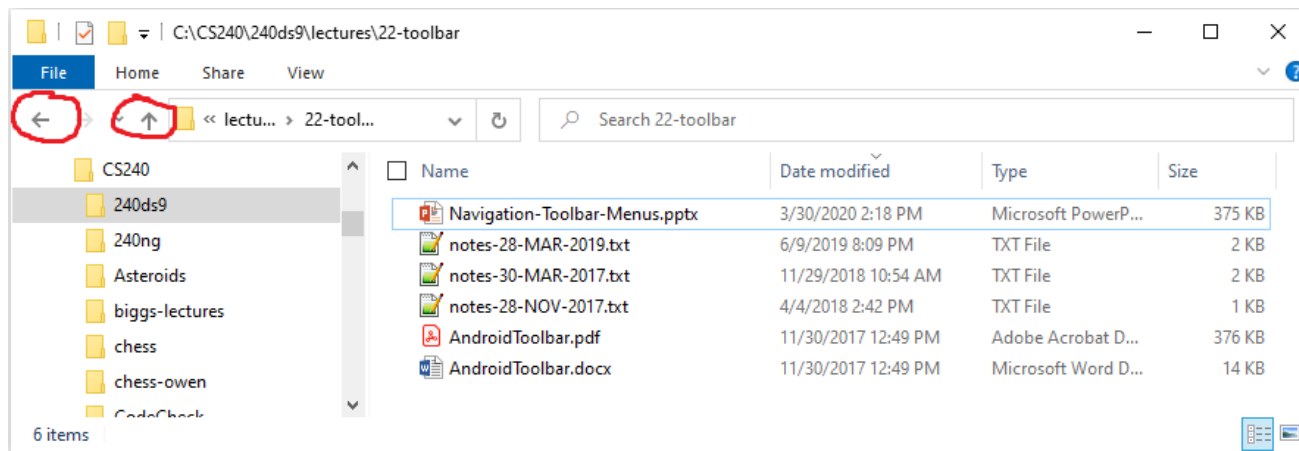


Navigation in Family Map Client

- REVIEW: Calling one activity from another
 - Create Intent object specifying which activity to call
 - Add activity parameters to Intent (putExtra)
 - Call new activity by calling
 - startActivity(Intent)
 - startActivityForResult(Intent)
 - Optionally, return values to calling activity
 - Create Intent object
 - Add return values to Intent (putExtra)
 - Call setResult(Intent)
 - Return control to calling activity
 - Call finish() method to terminate the current activity and return to the previous activity
- EXAMPLE: GeoQuiz with Cheating in Chapter 5 of textbook
 - Code is on 240 web site at “Lecture Files”, “Dr. Rodham”, “18-android-activities”, “code”, “05_SecondActivity_AndroidX.zip”

The Android Toolbar

- Chapter 13 in textbook, “The Toolbar”
- Menus
- Up button
 - Back button implements temporal navigation
 - Goes back to most recent activity
 - Up button implements structural navigation
 - Goes back to “parent” activity



Menus

- Create menu XML resource file
- Override onCreateOptionsMenu(...) method
- Override onOptionsItemSelected(...) method



Menus - Create menu XML resource file

- In “res” folder, right-click on “menu” folder and select “New” -> “Menu Resource File”

```
<?xml version="1.0" encoding="utf-8"?>
<menu 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"
      tools:context="edu.byu.cs240.familymap.ui.PersonActivity">

    <item android:id="@+id/action_jump"
          android:title="@string/action_jump"
          app:showAsAction="ifRoom"/>

    <item android:id="@+id/action_run"
          android:title="@string/action_run"
          app:showAsAction="ifRoom"/>

    <item android:id="@+id/action_scream"
          android:title="@string/action_scream"
          app:showAsAction="ifRoom"/>

</menu>
```

Menus - Override onCreateOptionsMenu(...) method

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    super.onCreateOptionsMenu(menu);  
    new MenuInflater(this).inflate(R.menu.menu_person, menu);  
    return true;  
}
```

Menus - Override onOptionsItemSelected(...) method

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_jump:
            Toast.makeText(this, "Jump!", Toast.LENGTH_LONG).show();
            return true;
        case R.id.action_run:
            Toast.makeText(this, "Run!", Toast.LENGTH_LONG).show();
            return true;
        case R.id.action_scream:
            Toast.makeText(this, "Scream!", Toast.LENGTH_LONG).show();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```


Up Button

- For activities that have an Up button, in the onCreate(...) method call:

```
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
```

- For each activity, define the “parent activity” in the AndroidManifest.xml file

```
<activity  
    android:name=".ui.EventActivity"  
    android:label="@string/title_activity_map"  
    android:parentActivityName=".ui.MainActivity">
```



Family Map Navigation Issues

- When the user returns to Main Activity or Event Activity using Back or Up, the map should be in the same state as when the user left
 - Same event should be selected, lines drawn properly, same scrolling position
 - If your code is done correctly, this should happen naturally and require no extra effort

Family Map Navigation Issues

- Up button should return back to Main Activity
 - Make Main Activity the parent activity for all the other activities
 - By default, when the user presses Up, Android will restart the parent activity (Main Activity in Family Map)
 - To prevent this, mark the parent activity as “single top” in the AndroidManifest.xml file

```
<activity  
    android:name=".ui.MainActivity"  
    android:label="@string/app_name"  
    android:launchMode="singleTop">
```

Family Map Navigation Issues

- When the user logs in or out, Main Activity needs to swap fragments and menus:
 - LOGGED OUT => LoginFragment and NO menus
 - LOGGED IN => MapFragment and Search/Settings menus
 - How is this swapping done?
- We already learned how to swap fragments in an activity
 - EXAMPLE: FragmentDemo
 - Code is on 240 web site at “Lecture Files”, “Dr. Rodham”, “19-android-fragments”, “code”, “FragmentDemo_AndroidX.zip”
- There are three times when fragments/menus need to be swapped
 - When the app starts
 - When the user logs in
 - When the user logs out

Family Map Navigation Issues

- When the app starts
 - In Main Activity onCreate(...) method:
 - Check to see if user is logged in
 - Make sure the appropriate fragment is installed
 - Call invalidateOptionsMenu() to force onCreateOptionsMenu(...) to be called again
- When the user logs in, follow the same steps as onCreate(...)
- When the user logs out,
 - In Settings Activity, call finish() to return to Main Activity
 - In Main Activity, override the onPostResume(...) lifecycle method. This will be called anytime the user returns to Main Activity using Back, Up, or Logout
 - In onPostResume(...) follow the same steps as onCreate(...)
 - NOTE: In onPostResume(...) you should only swap fragments/menus if the wrong ones are currently installed. Otherwise, leave them alone. This way, Main Activity won't lose its state when Back or Up is pressed.

Family Map Navigation Issues

- Create two different menu XML resource files (one with NO menus and one with Search/Settings menus)
- In Main Activity onCreateOptionsMenu(...) method,
 - Check to see if user is logged in
 - Install the appropriate menu