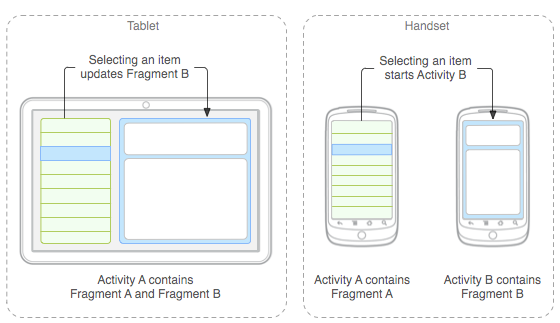
**Topic: Fragments**  
(Previous topic: UI Layout and Orientation)

Intro

* Reminders:
  + Lab 4 beta was due Saturday, code reviews are due tomorrow (Tuesday)
    - Questions about the lab? Bugs you’re trying to solve?
* Show where we are in the syllabus, week 6, next week we cover fragments

Fragment intro

* Fragments allow our apps to adapt to different screen sizes, densities, and aspect ratios.
* Code in an activity determines at runtime which fragments to load
* 
* Create a fragment by subclassing Fragment
  + The UI layout can be defined in AXML
  + Add the layout to the fragment in the onCreateView method
    - Use the LayoutInflator object which takes three arguments
      * Resource ID to inflate
      * The ViewGroup to be the parent of the inflated layout
      * A boolean indicating whether the inflated layout should be attached to the ViewGroup
  + public static class ExampleFragment extends Fragment {  
        @Override  
        public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                                 Bundle savedInstanceState) {  
            // Inflate the layout for this fragment  
            return inflater.inflate(R.layout.example\_fragment, container, false);  
        }  
    }
* Fragments can be added to an activity using AXML or can be loaded programmatically
  + Using AXML:  
      
    <?xml version="1.0" encoding="utf-8"?>  
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
        android:orientation="horizontal"  
        android:layout\_width="match\_parent"  
        android:layout\_height="match\_parent">  
        <fragment android:name="com.example.news.ArticleListFragment"  
                android:id="@+id/list"  
                android:layout\_weight="1"  
                android:layout\_width="0dp"  
                android:layout\_height="match\_parent" />  
        <fragment android:name="com.example.news.ArticleReaderFragment"  
                android:id="@+id/viewer"  
                android:layout\_weight="2"  
                android:layout\_width="0dp"  
                android:layout\_height="match\_parent" />  
    </LinearLayout>
  + The android:name attribute in the <fragment> specifies the Fragment class to instantiate in the layout.
* Programmatically
  + To make fragment transactions in your activity (such as add, remove, or replace a fragment), you must use APIs from [FragmentTransaction](http://developer.android.com/reference/android/app/FragmentTransaction.html). You can get an instance of [FragmentTransaction](http://developer.android.com/reference/android/app/FragmentTransaction.html) from your [Activity](http://developer.android.com/reference/android/app/Activity.html)like this:  
      
    FragmentManager fragmentManager = [getFragmentManager()](http://developer.android.com/reference/android/app/Activity.html#getFragmentManager());  
    FragmentTransaction fragmentTransaction = fragmentManager.[beginTransaction()](http://developer.android.com/reference/android/app/FragmentManager.html#beginTransaction());

Practice:

1. Create a fragment UI layout for each fragment.
   1. In addition to the existing main, create two fragment UIs in AXML, put them in the Resources, layout folder
2. Create a class for each fragment
   1. Add new files using the Android, Fragment option
   2. Add the code to each OnCreateView method to inflate the layout  
      *return inflater.Inflate(Resource.Layout.Fragment1UI, container, false);*
3. Add a fragment element to the main axml layout
   1. Add an android:name attribute with the name of the fragment class

