Jacob Taggart
Week2_Day2_ResearchHW

Week 2 Day 2 Research HW

1. What is the difference in recyclerView and listView?

   A: First of all, RecyclerView reuses the same spaces when scrolling up and down. This is possible because it is the default way of writing the adapter in the RecyclerView while in ListView to do this one would need to implement the ViewHolder manually in the ListView adapter. By adapter, it’s meant that the adapter class controls how each item individually renders and what data is rendered. RecyclerView differentiates the list from its container so it can display list items easily at run time while scrolling, while ListView considers the list and container one entity when passed. Thus RecyclerView can delegate list items to certain layouts via the LayoutManager while ListView takes more explicit assignments from the developer to accomplish this. RecyclerView has three methods built in: onBindViewHolder, onCreateView, and getItemCount. It also contains an internal class. RecyclerView uses lazy loading which will be described below.

2. Define lazy loading.

   A: Lazy loading is the pattern of loading carried out by the RecyclerView which waits to initialize an object until it is needed in the activity. The opposite of lazy loading is called eager loading and causes a much lower level of efficiency as at runtime all of the objects are already initialized and loaded. Lazy loading is a much quicker and more efficient way to load objects.

3. What is an item decorator in RecyclerViews?

   A: Simply put, an item decorator in RecyclerViews is a way to separate each object in a list. Another way to put it is to say it is that by using item decorators we have a high level of convenience when putting dividers between items and visual groups. More broadly it’s a way to style in the margins between objects/items in a list.

4. What is the View Holder Pattern?

A: A ViewHolder pattern is a way of holding all the references to the view ID resources being used so that calls to the resource directly will not be required. Each time getView() or findViewById() is called, it makes for a heavy workload on the CPU which leads to slower performance. By using ViewHolders, the performance is greatly increased and this potentially makes for less coding from the developer.

5. How do you implement an item touch helper for the RecyclerView?

ItemTouchHelper is a subclass of RecyclerView.ItemDecorator which means that it’s extremely simple to put into most existing LayoutManagers and adapters. In the build gradle module file, the developer needs to ensure they have the implementation of this Android library, “com.android.support:recyclerview-v7:28.0.0”. To use the Helper, one needs to create an ItemTouchHelper.Callback which is a specific interface that allows the system to detect a move or a swipe event.This is also allows the developer to control what state the view is in that is selected. There are three callback methods that need to be overridden: getMovementFlags(), onMove(), and onSwiped(). There are also other optional helpers which can be overridden such as isLongPressDragEnabled() and isItemViewSwipeEnabled for more pinpoint control. getMovementFlags() gets the movements of drags and swipes of the user. onMove() detects any movement and onSwiped() detects for a swipe.