Task 5.1P

Elementary, said he.

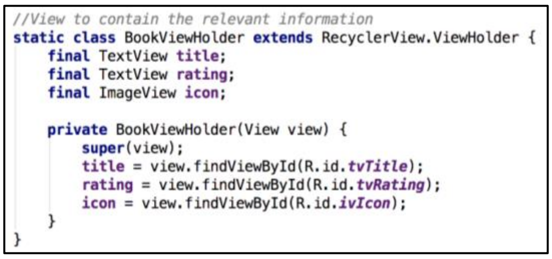
1. A simple booklist

As usual, Screenshots may be found in the Appendix.

Code Snippet – Row Adapter

//Following the tutorial from https://www.androidhive.info/2016/01/android-working-with-recycler-view/ (last accessed 4/10/18)

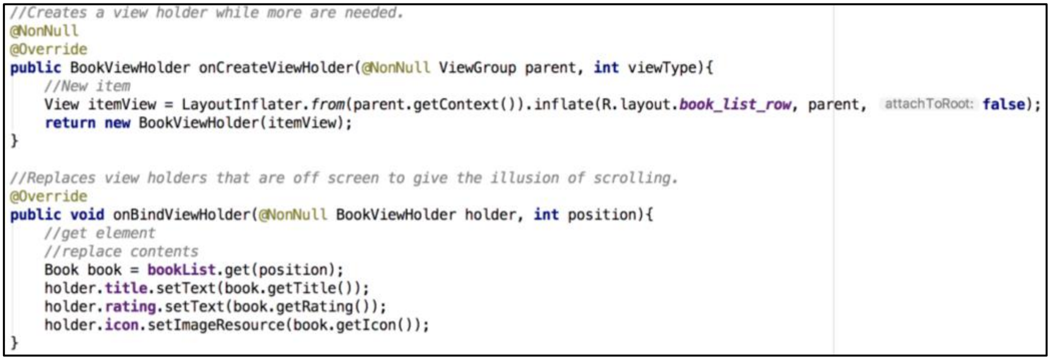
Central to the Adapter is the view holder, used to contain the information for each 'item' in the recycler view's list.



```
//View to contain the relevant information
static class BookViewHolder extends RecyclerView.ViewHolder {
    final TextView title;
    final TextView rating;
    final ImageView icon;

    private BookViewHolder(View view) {
        super(view);
        title = view.findViewById(R.id.tvTitle);
        rating = view.findViewById(R.id.tvRating);
        icon = view.findViewById(R.id.ivIcon);
    }
}
```

These are then created (to fill the area size) or manipulated (when scrolled offscreen) as necessary.



```
//Creates a view holder while more are needed.
@NonNull
@Override
public BookViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType){
    //New item
    View itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.book_list_row, parent,  attachToRoot: false);
    return new BookViewHolder(itemView);
}

//Replaces view holders that are off screen to give the illusion of scrolling.
@Override
public void onBindViewHolder(@NonNull BookViewHolder holder, int position){
    //get element
    //replace contents
    Book book = bookList.get(position);
    holder.title.setText(book.getTitle());
    holder.rating.setText(book.getRating());
    holder.icon.setImageResource(book.getIcon());
}
```

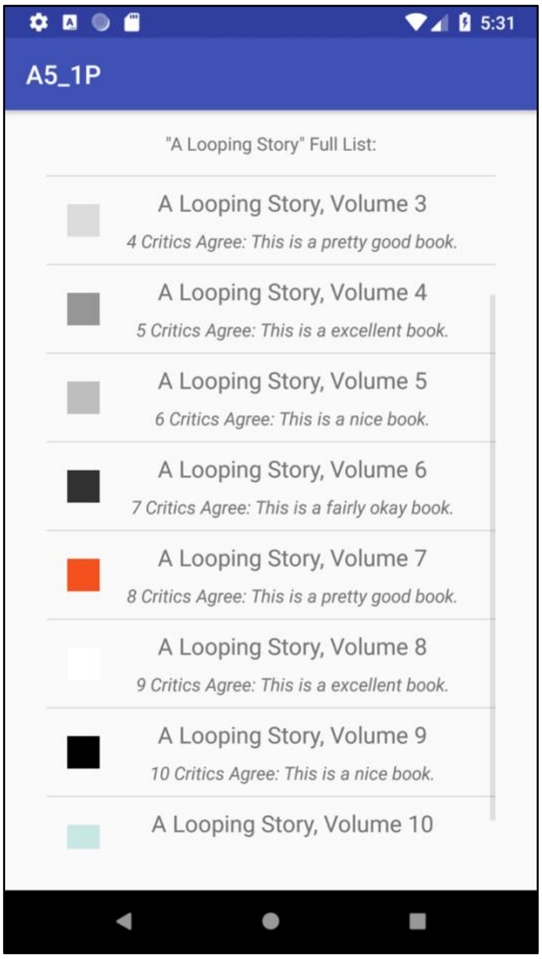
2. Activities and Fragments

While both Activities and Fragments may display information, only Activities can be used to display views, while Fragments must be contained within an activity. Fragments, however, are modular and once placed in an activity may be swapped out as necessary depending on things like screen size or other requirements. (Refer below graphic)

'FragmentTransaction's are used to detach, (re-)attach, show, hide and do various other tasks with Fragments as a whole. (Data passing is still managed with Intents.) To govern this, a 'FragmentManager' is used by the Activity.

One particular use for Fragments is taking advantage of larger screen sizes. (eg Tablets). The perennial example is the Gmail app, where half the screen may be a fragment for selecting an item from the inbox, while the other half may be for viewing the chosen message.

Appendix – Screenshots and Graphics



A5_1P

"A Looping Story" Full List:

A Looping Story, Volume 3

4 Critics Agree: This is a pretty good book.

A Looping Story, Volume 4

5 Critics Agree: This is a excellent book.

A Looping Story, Volume 5

6 Critics Agree: This is a nice book.

A Looping Story, Volume 6

7 Critics Agree: This is a fairly okay book.

A Looping Story, Volume 7

8 Critics Agree: This is a pretty good book.

A Looping Story, Volume 8

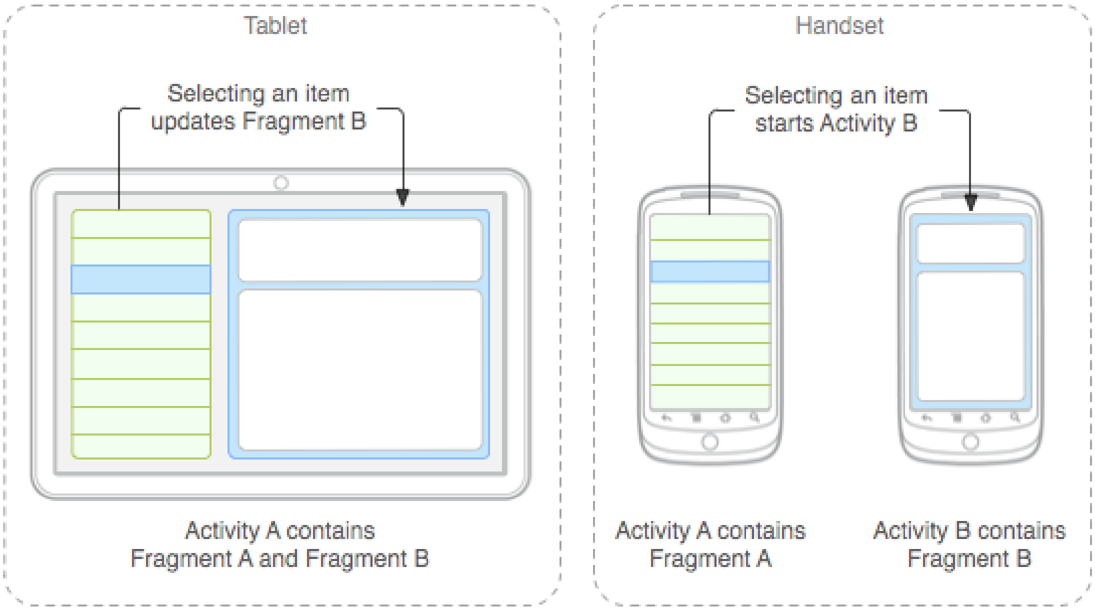
9 Critics Agree: This is a excellent book.

A Looping Story, Volume 9

10 Critics Agree: This is a nice book.

A Looping Story, Volume 10

Using the latest in publishing technology, this series is procedurally generated!



Tablet
Selecting an item updates Fragment B
Activity A contains Fragment A and Fragment B
Handset
Selecting an item starts Activity B
Activity A contains Fragment A
Activity B contains Fragment B

Graphic depicting relationship between Activities and Fragments via https://developer.android.com/guide/components/fragments 4/10/18