

Android Week Five

Yehyun (Joseph) Ryu
UMN App Developers

Outline

- More Customization for ListView
 - OnClick for Item
 - Background Color
- RecyclerView

WeekFourExamples
0
1
2
3
4
5
6
7

Add OnClick to Items

- setOnClickListener() to listItemView

```
listItemView.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Toast.makeText(getContext(), String.valueOf(number), Toast.LENGTH_SHORT).show();  
    }  
});
```

Intent Data Passing

- `Intent intent = new Intent(context, Activity.class);`
- `intent.putExtra(key, value);`

```
Intent intent = new Intent(getApplicationContext(), ColorActivity.class);  
intent.putExtra("name: Number", number);  
getApplicationContext().startActivity(intent);
```

Create ColorActivity

- Create ColorActivity
- Create TextView in activity_color.xml and bind view in ColorActivity.java
- getIntent()
 - Extract number
 - Set background and text

```
mNumberTextView = findViewById(R.id.color_textview);
```

```
Intent intent = getIntent();
```

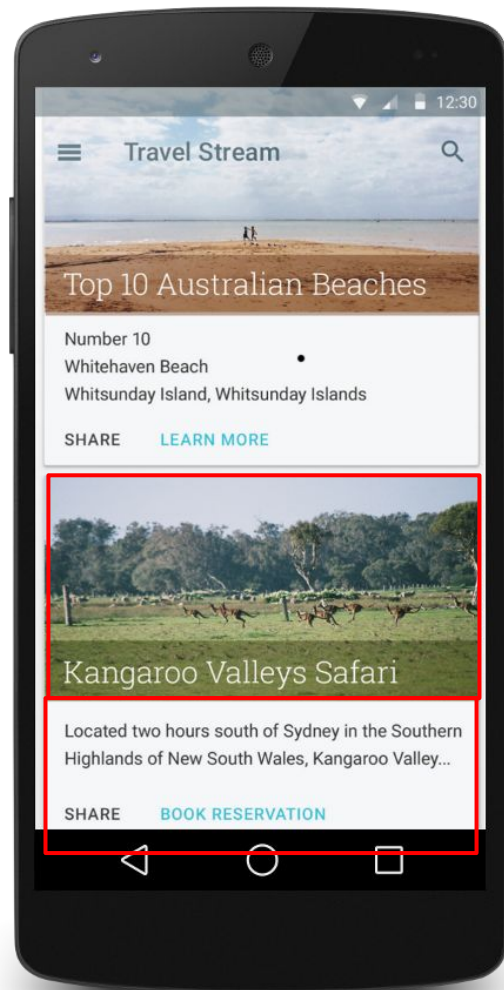
```
if(intent != null) {
```

```
    mNumber = intent.getIntExtra("Number", defaultValue: 0);
```

```
}
```

CardView

- ViewGroup with rounded corners and elevation.
- Elevation: vertical distance from screen
 - 0dp: no shadow
 - x dp > 0: shadow
- More distinction, more visible
- Looks better



Create CardView

- Add this line in build.gradle (app)
 - compile 'com.android.support:cardview-v7:26.1.0'
 - Sync gradle
- Create CardView in itemview.xml
 - Put textview within cardview

```
<android.support.v7.widget.CardView
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_margin="8dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent">

    <TextView
        android:id="@+id/itemview_text"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:text="TESTING"
        android:gravity="center"
        android:textSize="24sp"
        android:textColor="@android:color/black"
        android:fontFamily="sans-serif-medium" />

</android.support.v7.widget.CardView>
```

Finished!

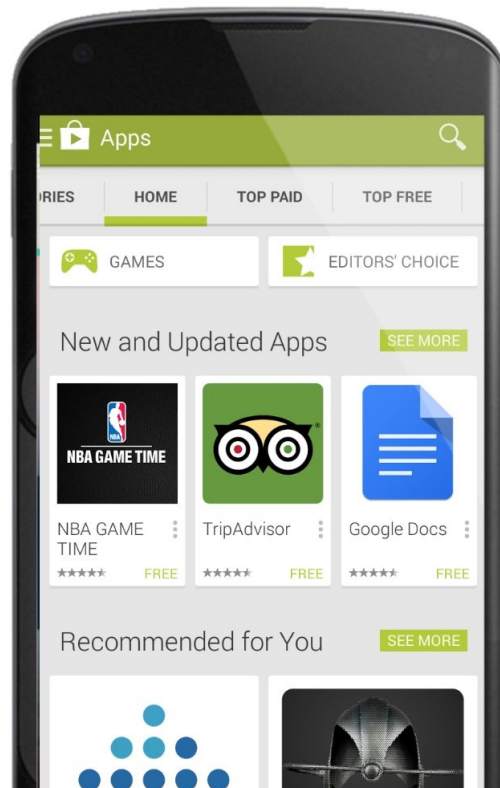
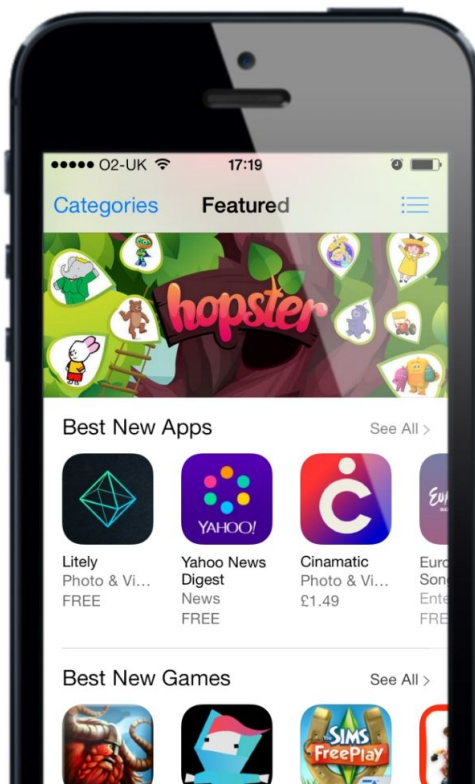
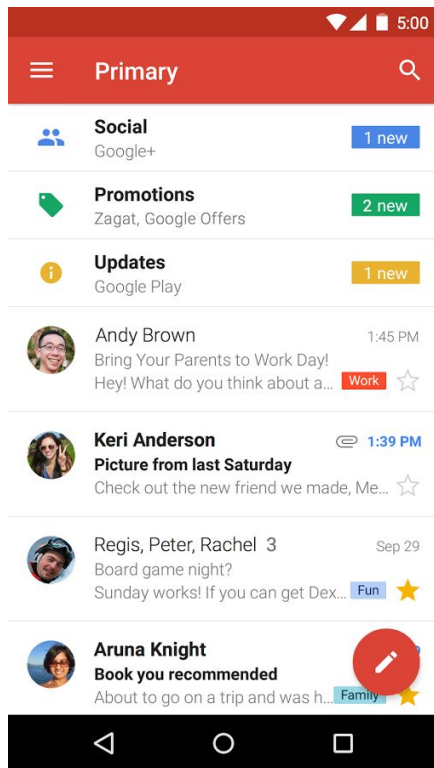
WeekFourExamples
0
1
2
3
4
5
6
7



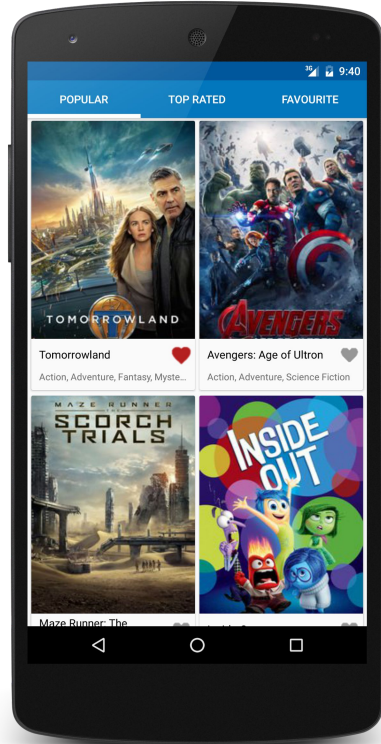
Only One Activity

WeekFourExamples
3

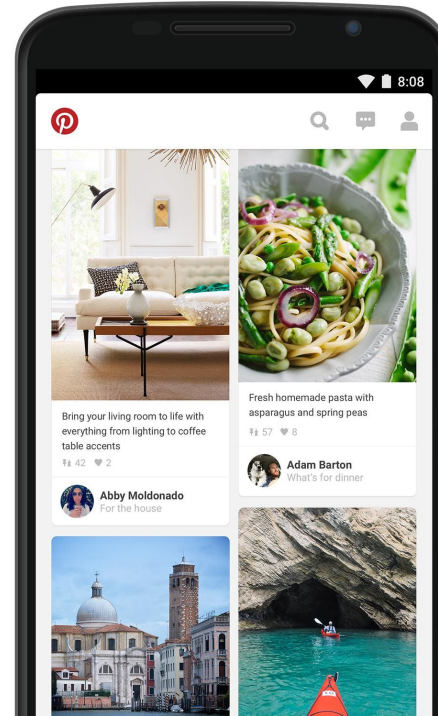
RecyclerView



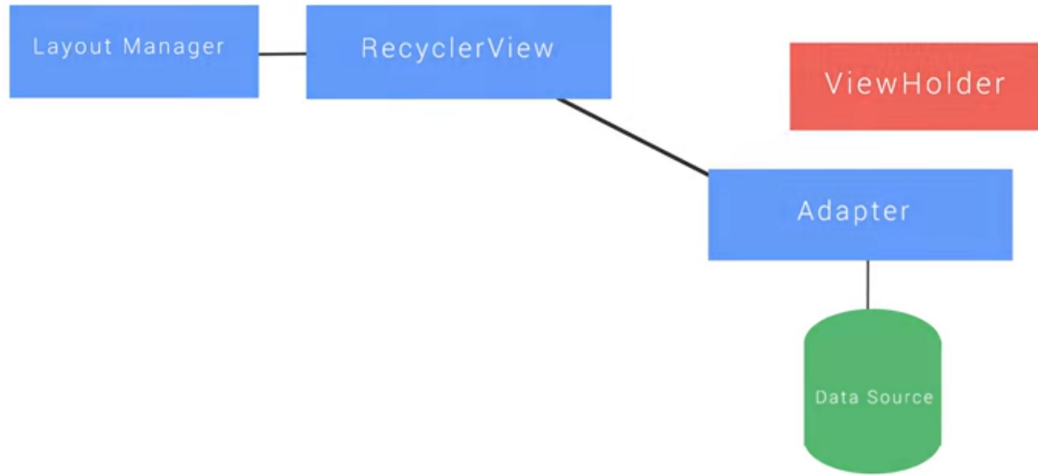
RecyclerView



Discover and save creative ideas
for any project or interest



How RecyclerView Works



Udacity

How RecyclerView Works

- Adapter
 - Provides new view when needed
 - Bind data from data source to the view
- Data Source
 - Data to represent
- ViewHolder
 - Receives view from adapter
 - Contains reference to root view
 - Cache views in viewgroup
 - Less findViewById() calls
- LayoutManager
 - Displays views in a certain orientation or style (Linear, Horizontal, Grid, Staggered)

ListView vs RecyclerView

RecyclerView is better because ...

1. Free animations for item insertion and deletion
2. Easy to filter
3. Flexible layout
4. Less expensive

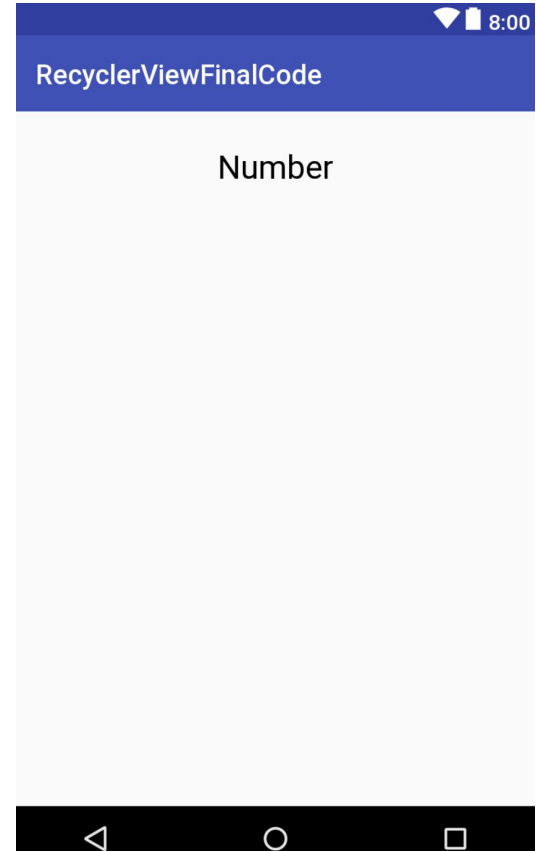
ListView doesn't force ...

1. View recycling
2. View caching to reduce findViewById calls

How to Implement RecyclerView

1. Create Item Layout File

- Create layout resource file called itemview.xml
- Add TextView with ID



2. Create Adapter

- Create Java file called RecyclerViewAdapter.java
- Extend RecyclerView.Adapter<VH>
- Create ViewHolder class within Adapter class
- Override onCreateViewHolder, onBindViewHolder, getItemCount
- Create constructors
- Implement the functions

3. Attach Adapter and LayoutManager

- LinearLayoutManager in this case
- setAdapter()
- setLayoutManager()

```
ArrayList<Integer> numbers = new ArrayList<>();  
for(int i = 0; i < 15; i++) {  
    numbers.add(i);  
}  
mRecyclerView = (RecyclerView) findViewById(R.id.main_recyclerview);  
RecyclerViewAdapter adapter = new RecyclerViewAdapter(context: this, numbers);  
mRecyclerView.setAdapter(adapter);  
LinearLayoutManager layoutManager = new LinearLayoutManager(context: this, LinearLayoutManager.VERTICAL, reverseLayout: false);  
mRecyclerView.setLayoutManager(layoutManager);
```

Next Week

- Fragments
- Activity Lifecycle
- Fragment Lifecycle

