Department of Computer Technology and Information Systems

## CTIS 487 - Mobile Application Development

# **Lab Guide Extra Splash Screen**

|  |
| --- |
| **OBJECTIVES :** Splash Screen + Recycler View |
| **Instructor :** Neşe ŞAHİN ÖZÇELİK  **Assistant :**  Leyla SEZER |

Create the following design SplashActivty which will be launcher activity duration for 5 seconds.

****

**import** android.os.Handler;  
…  
**public class** SplashActivity **extends** AppCompatActivity {  
 Intent **intent**;  
 *// Splash screen timer* **private static int** *SPLASH\_TIME\_OUT* = 5000;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
  
 *// Hiding title bar using code* getSupportActionBar().hide();  
  
 setContentView(R.layout.***activity\_splash***);  
  
 *// Hiding the status bar* getWindow().setFlags(WindowManager.LayoutParams.***FLAG\_FULLSCREEN***, WindowManager.LayoutParams.***FLAG\_FULLSCREEN***);  
  
 *// Locking the orientation to Portrait* setRequestedOrientation(ActivityInfo.***SCREEN\_ORIENTATION\_PORTRAIT***);  
  
 */\*  
 \* postDelayed() method causes the Runnable object to be added to the  
 \* message queue to be run after the specified amount of time elapses.  
 \* The runnable will be run on the thread to which this handler is  
 \* attached.  
 \*/* **new** Handler().postDelayed(**new** Runnable() {  
 */\*  
 \* Showing splash screen with a timer. This will be useful when you  
 \* want to show your application logo or company logo  
 \*/* @Override  
 **public void** run() {  
 *// This method will be executed once the timer is over  
 // Start your application's main activity which is a ListView* **intent** = **new** Intent(SplashActivity.**this**, MainActivity.**class**);  
  
 *// Close this activity* finish();  
  
 startActivity(**intent**);  
 }  
 }, *SPLASH\_TIME\_OUT*);  
 }  
}

**Step1: Create a House class**

|  |
| --- |
| **House** |
| -name: String  -words: String  -logo: int |
| +House(String,String,int)  +getName():String  +setName(String)  +getWords():String  +setWords(String)  +getLogo():String  +setLogo(String)  +toString():String |

**Step2: Create empty MainActivity and its layout**

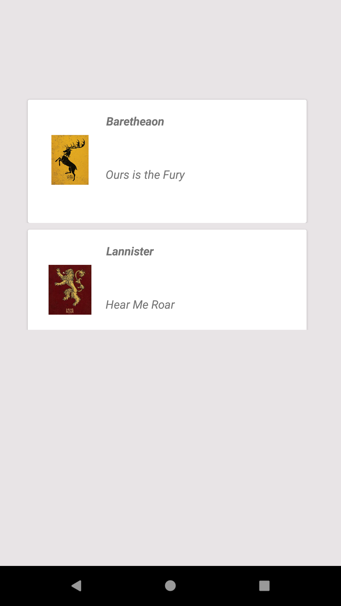
<**android.support.v7.widget.RecyclerView  
 android:id="@+id/recyclerGot"  
 android:layout\_width="0dp"  
 android:layout\_height="286dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="112dp"  
 android:layout\_marginEnd="16dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"** />

**Step3: Create recycler\_layout**

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout 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"  
 android:id="@+id/constLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**>  
  
 <**android.support.v7.widget.CardView  
 android:layout\_width="337dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="8dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"**>  
  
  
 <**android.support.constraint.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="149dp"**>  
  
 <**ImageView  
 android:id="@+id/rec\_img"  
 android:layout\_width="70dp"  
 android:layout\_height="60dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="4dp"  
 android:layout\_marginBottom="8dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@mipmap/ic\_launcher"** />  
  
 <**TextView  
 android:id="@+id/rec\_tv"  
 android:layout\_width="125dp"  
 android:layout\_height="44dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="16dp"  
 android:text="TextView"  
 android:textStyle="bold|italic"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.008"  
 app:layout\_constraintStart\_toEndOf="@+id/rec\_img"  
 app:layout\_constraintTop\_toTopOf="parent"** />  
  
 <**TextView  
 android:id="@+id/rec\_exp"  
 android:layout\_width="125dp"  
 android:layout\_height="44dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginBottom="25dp"  
 android:text="TextView"  
 android:textStyle="italic"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/rec\_img"  
 app:layout\_constraintTop\_toBottomOf="@+id/rec\_tv"  
 app:layout\_constraintVertical\_bias="0.0"** />  
  
  
 </**android.support.constraint.ConstraintLayout**>  
  
 </**android.support.v7.widget.CardView**>  
</**android.support.constraint.ConstraintLayout**>

**STEP 4: To create custom recycler view**

* As customized recycler item house image and next to it house name and the house words will be displayed, get the recycler\_layout.xml and import it to your project.



**To create custom RecylerView**

* In MainActivity class, create an ArrayList to hold House objects and add House objects by calling prepareData() method. Think about this cities can be taken from a database or json file through network connection.

**private** ArrayList<House> **recyclervalues** = **new** ArrayList<>();

…onCreate(…)

{

prepareData();

}

**public void** prepareData() {  
 Collections.*addAll*(**recyclervalues**, **new** House(R.drawable.***baratheon***, **"Baretheaon"**, **"Ours is the Fury"**),  
 **new** House(R.drawable.***lannister***, **"Lannister"**, **"Hear Me Roar"**),  
 **new** House(R.drawable.***martell***, **"Martell"**, **"Unbowed,Unbent,Unbroken"**),  
 **new** House(R.drawable.***stark***, **"Stark"**, **"Winter is Coming"**),  
 **new** House(R.drawable.***targeryan***, **"Targeryan"**, **"Fire and Blood"**),  
 **new** House(R.drawable.***tyrell***, **"Tyrell"**, **"Growing Strong"**));  
 */\*\*\*\*\*\*\*\*\*\*or \*\*\*\*\*\*\*\*\*\*\*\*/  
 /\*  
 recyclervalues.add(new House(R.drawable.lannister, "Lannister", "Hear Me Roar"));  
 ...  
 //This data can be taken from file, database or network.\*/*}

* Create recycler object in MainActivity

**private** RecyclerView **recyclerHouses**;

* Create MyRecyclerViewAdapter object and set the adapter to recycler.

…onCreate(…)

{

**recyclerHouses** = findViewById(R.id.***recyclerGot***);

**layoutManager** = **new** LinearLayoutManager(**this**);  
  
*//layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);***layoutManager**.setOrientation(LinearLayoutManager.***VERTICAL***);  
**recyclerHouses**.setLayoutManager(**layoutManager**);  
**recyclerHouses**.hasFixedSize();  
  
**adapter** = **new** MyRecyclerViewAdapter(**this**, **recyclervalues**);  
  
**recyclerHouses**.setAdapter(**adapter**);

}

* Create MyRecyclerViewAdapter class
  + **MyRecyclerViewItemHolder:** That method needs to construct a [RecyclerView.ViewHolder](https://developer.android.com/reference/android/support/v7/widget/RecyclerView.ViewHolder.html) and set the view it uses to display its contents. The type of the ViewHolder must match the type declared in the Adapter class signature. Typically, it would set the view by inflating an XML layout file. Because the view holder is not yet assigned to any particular data, the method does not actually set the view's contents.
  + The layout manager then binds the view holder to its data. It does this by calling the adapter's **onBindViewHolder()** method, and passing the view holder's position in the RecyclerView. The onBindViewHolder() method needs to fetch the appropriate data, and use it to fill in the view holder's layout.

**public class** MyRecyclerViewAdapter **extends** RecyclerView.Adapter<MyRecyclerViewAdapter.**MyRecyclerViewItemHolder**> {  
 **private** Context **context**;  
 **private** ArrayList<House> **recyclerItemValues**;  
  
 **public** MyRecyclerViewAdapter(Context context, ArrayList<House> values) {  
 **this**.**context** = context;  
 **this**.**recyclerItemValues** = values;  
 }  
  
 @Override  
 **public MyRecyclerViewItemHolder** **onCreateViewHolder**(ViewGroup viewGroup, **int** i) {  
 LayoutInflater inflator = LayoutInflater.*from*(viewGroup.getContext());  
  
 View itemView = inflator.inflate(R.layout.***recycler\_layout***, viewGroup, **false**);  
  
 MyRecyclerViewItemHolder mViewHolder = **new** MyRecyclerViewItemHolder(itemView);  
 **return** mViewHolder;  
 }  
  
 @Override  
 **public void** onBindViewHolder(MyRecyclerViewItemHolder myRecyclerViewItemHolder, **int** i) {  
  
 **final** House sm = **recyclerItemValues**.get(i);  
  
 myRecyclerViewItemHolder.**name**.setText(sm.getName());  
 myRecyclerViewItemHolder.**words**.setText(sm.getWords());  
 myRecyclerViewItemHolder.**img**.setImageResource(sm.getLogo());  
  
 myRecyclerViewItemHolder.**parentLayout**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Toast.*makeText*(**context**, **"Clicked on item "** + sm.getName(), Toast.***LENGTH\_LONG***).show();  
 }  
 });  
  
 myRecyclerViewItemHolder.**img**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Toast.*makeText*(**context**, **"Clicked on "** + sm.getName() + **" Image of item"**, Toast.***LENGTH\_LONG***).show();  
 }  
 });  
  
 myRecyclerViewItemHolder.**name**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Toast.*makeText*(**context**, **"Clicked on "** + sm.getName() + **" TextView of item"**, Toast.***LENGTH\_LONG***).show();  
 }  
 });  
  
  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return recyclerItemValues**.size();  
 }  
  
 **class** MyRecyclerViewItemHolder **extends** RecyclerView.ViewHolder {  
 TextView **name**, **words**;  
 ImageView **img**;  
 ConstraintLayout **parentLayout**;  
  
 **public** MyRecyclerViewItemHolder(@NonNull View itemView) {  
 **super**(itemView);  
 **name** = itemView.findViewById(R.id.***rec\_tv***);  
 **words** = itemView.findViewById(R.id.***rec\_exp***);  
 **img** = itemView.findViewById(R.id.***rec\_img***);  
 **parentLayout** = itemView.findViewById(R.id.***constLayout***);  
 }  
 }  
}

* Run the application, observe it. Opening the list scroll down-up.
* When click to the members of the Recycler List, there will be a toast message as shown.

