HW5:

App Bar layout

Martin Mudenda Bbela

2582912

Overview:

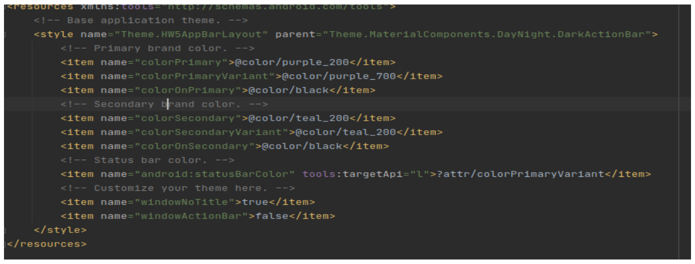
Modify the ActionBar app by by altering the behavior when each action item is clicked in the following ways:

- When item1 through item4 is clicked, a TextView should be displayed at the upper-left, upper-right, bottom-left, and buttom-right on the screen, respectively
- When item5 is clicked, all existing TextViews that have been added to the screen should be cleared

Method:

Manifest

set windows action bar to False in both Day and night themes

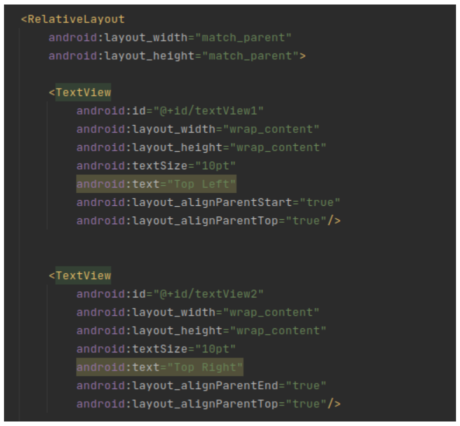


```xml
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.HW5AppBarLayout" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/purple_200</item>
        <item name="colorPrimaryVariant">@color/purple_700</item>
        <item name="colorOnPrimary">@color/black</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/teal_200</item>
        <item name="colorSecondaryVariant">@color/teal_200</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
        <item name="windowNoTitle">true</item>
        <item name="windowActionBar">false</item>
    </style>
</resources>
```

Main Activity

layout

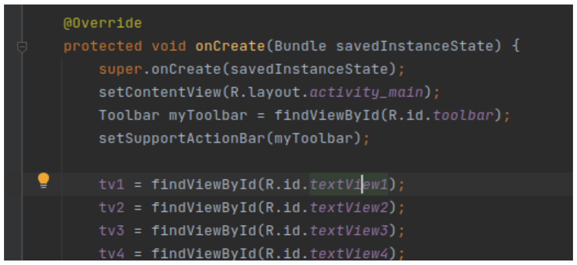
The Layout has 4 *TextView* objects anchored to each corner of the Relative Layout. Each object is named after the corner they are anchored to.



```xml
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="10pt"
        android:text="Top Left"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"/>


    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="10pt"
        android:text="Top Right"
        android:layout_alignParentEnd="true"
        android:layout_alignParentTop="true"/>
```

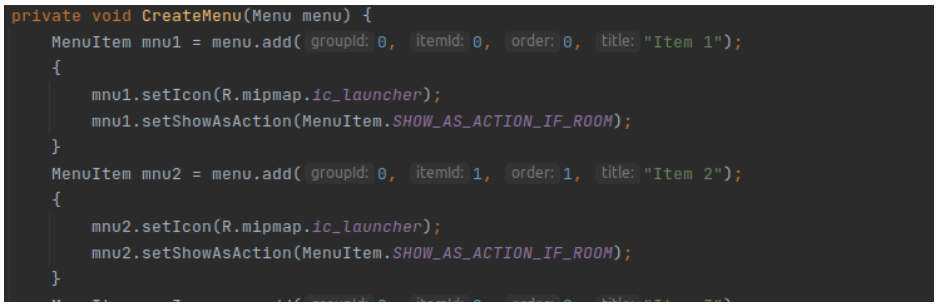
Activity

Wire in the TextViews in onCreate and initialize the custom Toolbar. Setup the custom toolbar in onCreateOptionsMenu().



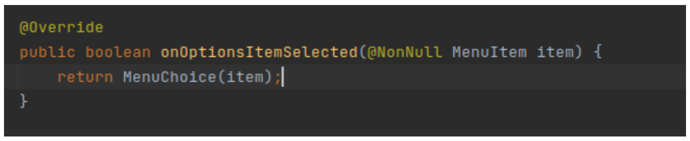
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar myToolbar = findViewById(R.id.toolbar);
    setSupportActionBar(myToolbar);

    tv1 = findViewById(R.id.textView1);
    tv2 = findViewById(R.id.textView2);
    tv3 = findViewById(R.id.textView3);
    tv4 = findViewById(R.id.textView4);
```

To Create the Options Menu we create a MenuItem and set a default Icon, then set it to only show as action if there is room.

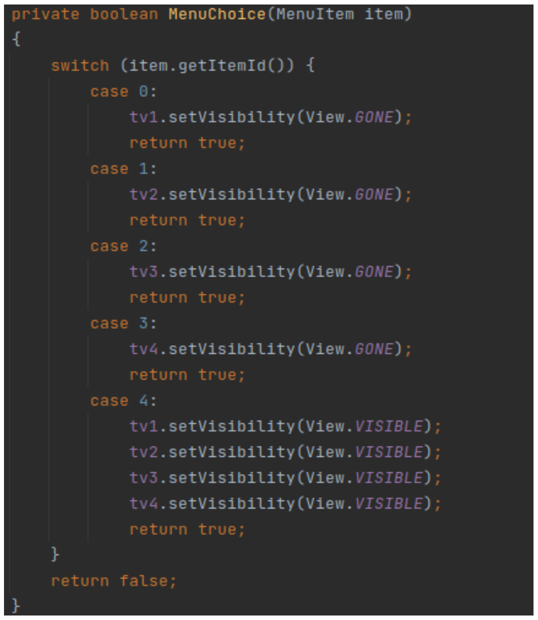


```
private void CreateMenu(Menu menu) {
    MenuItem mnu1 = menu.add( groupId: 0,  itemId: 0,  order: 0,  title: "Item 1");
    {
        mnu1.setIcon(R.mipmap.ic_launcher);
        mnu1.setShowAsAction(MenuItem.SHOW_AS_ACTION_IF_ROOM);
    }
    MenuItem mnu2 = menu.add( groupId: 0,  itemId: 1,  order: 1,  title: "Item 2");
    {
        mnu2.setIcon(R.mipmap.ic_launcher);
        mnu2.setShowAsAction(MenuItem.SHOW_AS_ACTION_IF_ROOM);
    }
```

Now that the options have been created. We wire an on *MenuChoice* method into *onOptionItemSelected*. This will set the visibility of the TextViews off depending on which option Item is selected. It will also finally set the last option which will make all of the TextViews Visible.



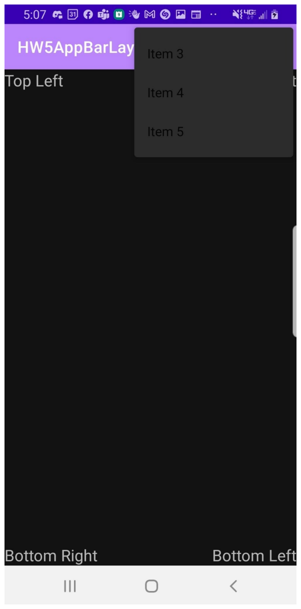
```
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    return MenuChoice(item);
}
```

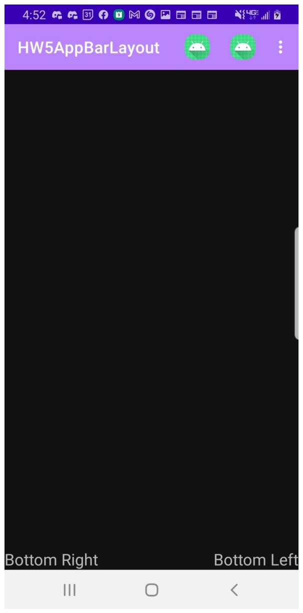


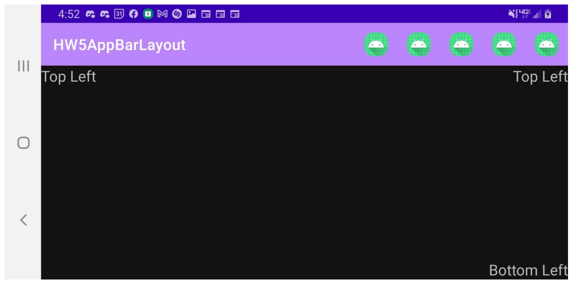
```
private boolean MenuChoice(MenuItem item)
{
    switch (item.getItemId()) {
        case 0:
            tv1.setVisibility(View.GONE);
            return true;
        case 1:
            tv2.setVisibility(View.GONE);
            return true;
        case 2:
            tv3.setVisibility(View.GONE);
            return true;
        case 3:
            tv4.setVisibility(View.GONE);
            return true;
        case 4:
            tv1.setVisibility(View.VISIBLE);
            tv2.setVisibility(View.VISIBLE);
            tv3.setVisibility(View.VISIBLE);
            tv4.setVisibility(View.VISIBLE);
            return true;
    }
    return false;
}
```

Result:

The AppBar will only show the MenuItems as Icons when they are available. Each MenuItem will make one of the TextViews invisible except the *Item 5* option which will make them all visible again.


5:07
HW5AppBarLay
Item 3
Item 4
Item 5
Top Left
Bottom Right
Bottom Left


4:52
HW5AppBarLayout
Bottom Right
Bottom Left


4:52
HW5AppBarLayout
Top Left
Top Left
Bottom Left