

Agenda

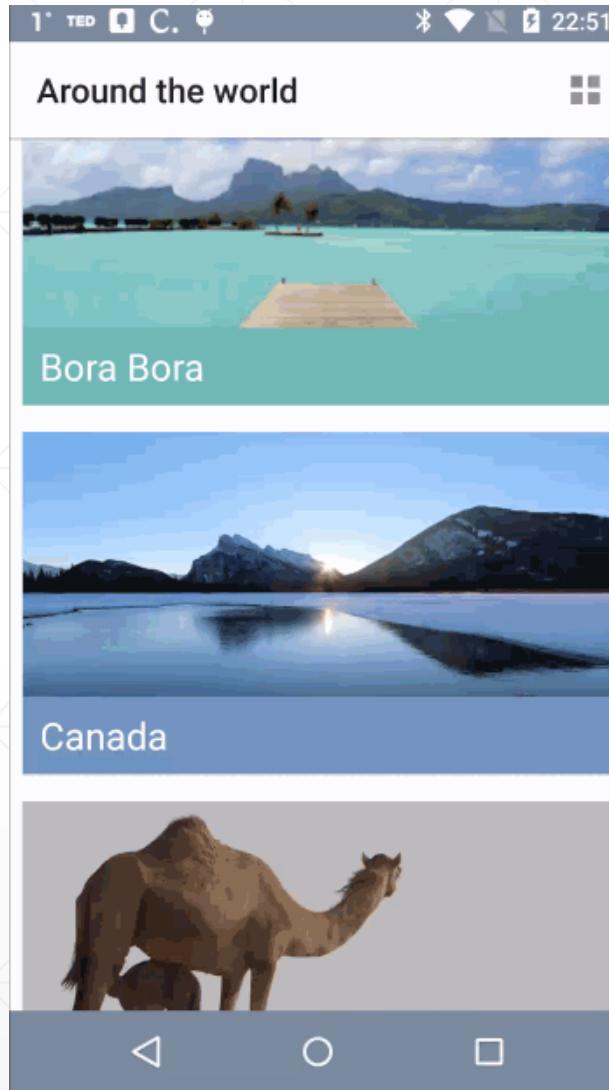
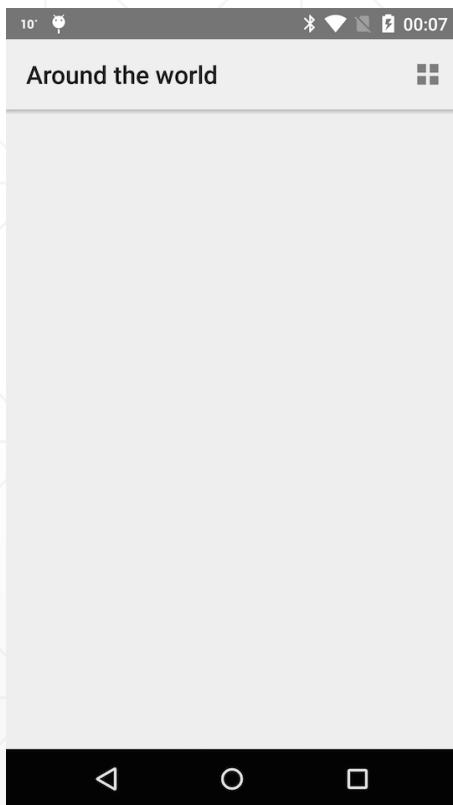
- Material Design
 - <https://www.raywenderlich.com/103367/material-design>



Bullet Points

- Implement the material theme;
 - Build dynamic views using new widgets like RecyclerView and CardView;
 - Use Palette API to generate color schemes that you can use for text or background views;
 - Create interactions using the animation APIs.
-

Before and After



build.gradle

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    compile 'com.android.support:recyclerview-v7:21.+'  
    compile 'com.android.support:cardview-v7:21.+'  
    compile 'com.android.support:palette-v7:21.0.0'  
    compile 'com.squareup.picasso:picasso:2.5.0'  
}
```

style.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <style name="AppTheme" parent="android:ThemeOverlay.Material.Light">
        <item name="android:colorPrimary">@color/primary</item>
        <item name="android:colorPrimaryDark">@color/primary_dark</item>
        <item name="android:colorAccent">@color/accent</item>
        <item name="android:navigationBarColor">@color/primary_dark</item>
        <item name="android:windowActionBar">false</item>
        <item name="android:windowNoTitle">true</item>
        <item name="android:displayOptions">disableHome</item>
    </style>
</resources>
```



Around the world



style.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <style name="AppTheme" parent="android:ThemeOverlay.Material.Light">
        <item name="android:colorPrimary">@color/primary</item>
        <item name="android:colorPrimaryDark">@color/primary_dark</item>
        <item name="android:colorAccent">@color/accent</item>
        <item name="android:navigationBarColor">@color/primary_dark</item>
        <item name="android:windowActionBar">false</item>
        <item name="android:windowNoTitle">true</item>
        <item name="android:displayOptions">disableHome</item>
    </style>
</resources>
```



RecyclerView

- Implementing a Recycler View in XML
- Initializing a Recycler View and Applying a Layout Manager



Implementing a Recycler View in XML

```
<android.support.v7.widget.RecyclerView  
    android:id="@+id/list"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="#f8f8f8"  
    android:divider="@null"  
    android:listSelector="@android:color/transparent"/>
```

Initializing a Recycler View

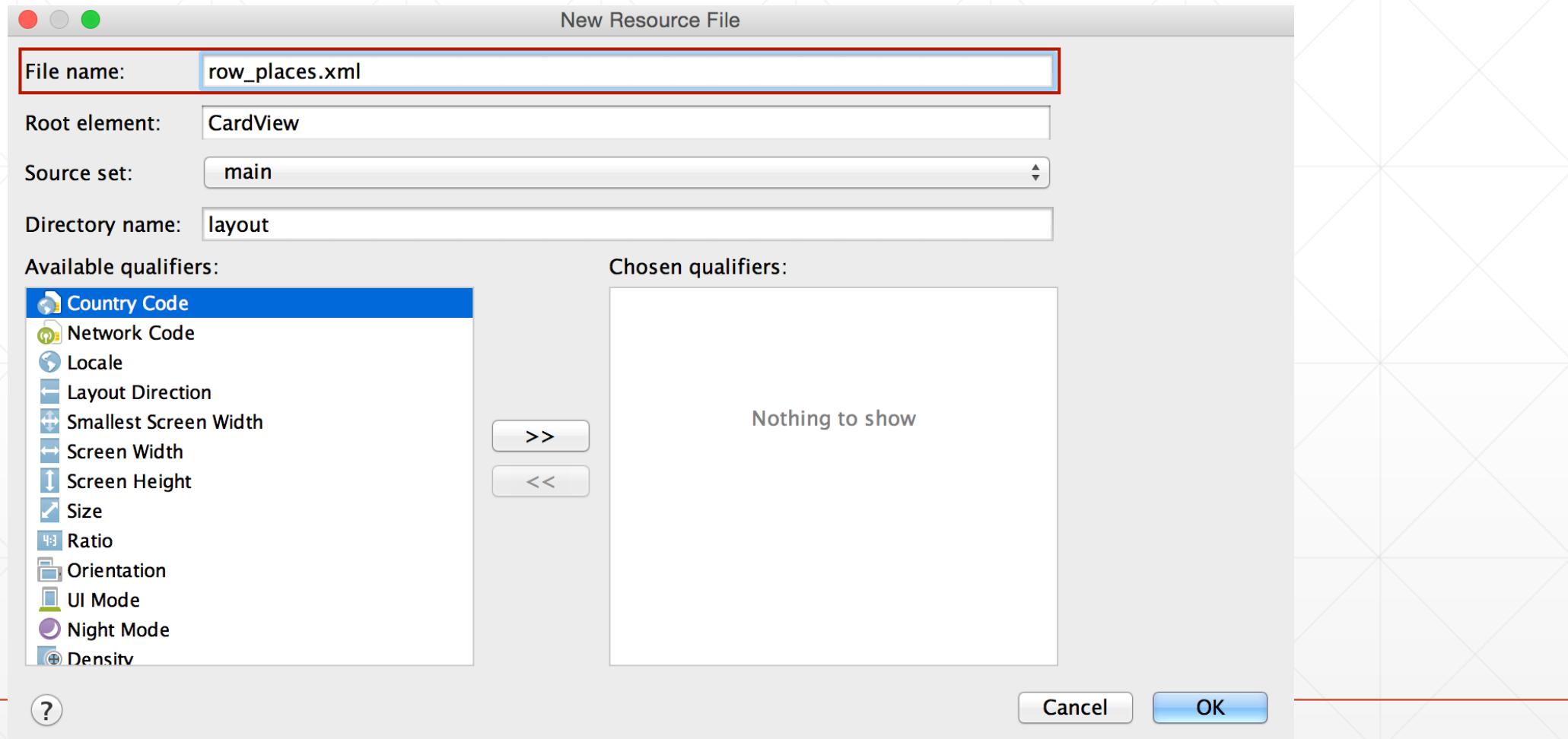
```
private RecyclerView mRecyclerView;  
private StaggeredGridLayoutManager mStaggeredLayoutManager;
```



Applying a Layout Manager

```
mStaggeredLayoutManager = new StaggeredGridLayoutManager(1, StaggeredGridLayoutManager.VERTICAL);  
  
mRecyclerView = (RecyclerView) findViewById(R.id.list);  
mRecyclerView.setLayoutManager(mStaggeredLayoutManager);
```

Creating Rows and Cells Using a Card View



```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    android:id="@+id/placeCard"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    card_view:cardCornerRadius="0dp"
    card_view:cardElevation="1dp">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="200dp">

        <ImageView
            android:id="@+id/placeImage"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:scaleType="centerCrop"
            android:transitionName="tImage" />

        <LinearLayout
            android:id="@+id/mainHolder"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="?android:selectableItemBackground"
            android:orientation="horizontal" />

        <LinearLayout
            android:id="@+id/placeholderNameHolder"
            android:layout_width="match_parent"
            android:layout_height="45dp"
            android:layout_gravity="bottom"
            android:orientation="horizontal"
            android:transitionName="tNameHolder">

            <TextView
                android:id="@+id/placeName"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="center_vertical"
                android:gravity="left"
                android:paddingLeft="10dp"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:textColor="@android:color/white" />

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

Implementing an Adapter for a Recycler View

```
// 1
public class TravelListAdapter extends RecyclerView.Adapter<TravelListAdapter.ViewHolder> {

    Context mContext;

    // 2
    public TravelListAdapter(Context context) {
        this.mContext = context;
    }

    // 3
    public class ViewHolder extends RecyclerView.ViewHolder {
        public LinearLayout placeHolder;
        public LinearLayout placeNameHolder;
        public TextView placeName;
        public ImageView placeImage;

        public ViewHolder(View itemView) {
            super(itemView);
            placeHolder = (LinearLayout) itemView.findViewById(R.id.mainHolder);
            placeName = (TextView) itemView.findViewById(R.id.placeName);
            placeNameHolder = (LinearLayout) itemView.findViewById(R.id.placeNameHolder);
            placeImage = (ImageView) itemView.findViewById(R.id.placeImage);
        }
    }
}
```

```
// 1
public class TravelListAdapter extends RecyclerView.Adapter<TravelListAdapter.ViewHolder> {

    Context mContext;

    // 2
    public TravelListAdapter(Context context) {
        this.mContext = context;
    }

    // 3
    public class ViewHolder extends RecyclerView.ViewHolder {
        public LinearLayout placeHolder;
        public LinearLayout placeNameHolder;
        public TextView placeName;
        public ImageView placeImage;

        public ViewHolder(View itemView) {
            super(itemView);
            placeHolder = (LinearLayout) itemView.findViewById(R.id.mainHolder);
            placeName = (TextView) itemView.findViewById(R.id.placeName);
            placeNameHolder = (LinearLayout) itemView.findViewById(R.id.placeNameHolder);
            placeImage = (ImageView) itemView.findViewById(R.id.placeImage);
        }
    }
}
```

TravelListAdapter.java

```
// 1
@Override
public int getItemCount() {
    return new PlaceData().placeList().size();
}

// 2
@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.row_places, parent, false);
    return new ViewHolder(view);
}

// 3
@Override
public void onBindViewHolder(final ViewHolder holder, final int position) {
    final Place place = new PlaceData().placeList().get(position);
    holder.placeName.setText(place.name);
    Picasso.with(mContext).load(place.getImageResourceId(mContext)).into(holder.placeImage);
}
```



Around the world

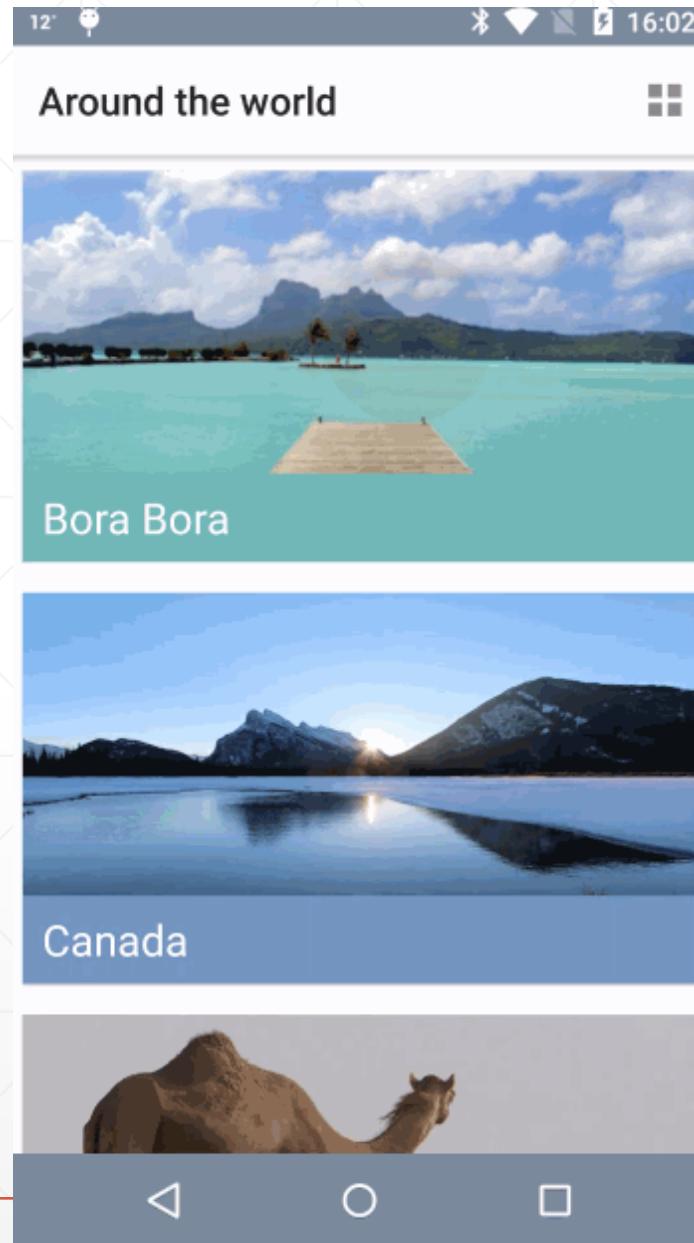


MainActivity.java

```
Private TravelListAdapter mAdapter;
```

```
mAdapter = new TravelListAdapter(this);  
mRecyclerView.setAdapter(mAdapter);
```

Implementing a Click Interface for Each Cell



Implementing a Click Interface for Each Cell

OnItemClickListener mItemClickListener;

```
public class ViewHolder extends RecyclerView.ViewHolder implements  
    View.OnClickListener {
```

```
@Override  
public void onClick(View v) {  
}
```

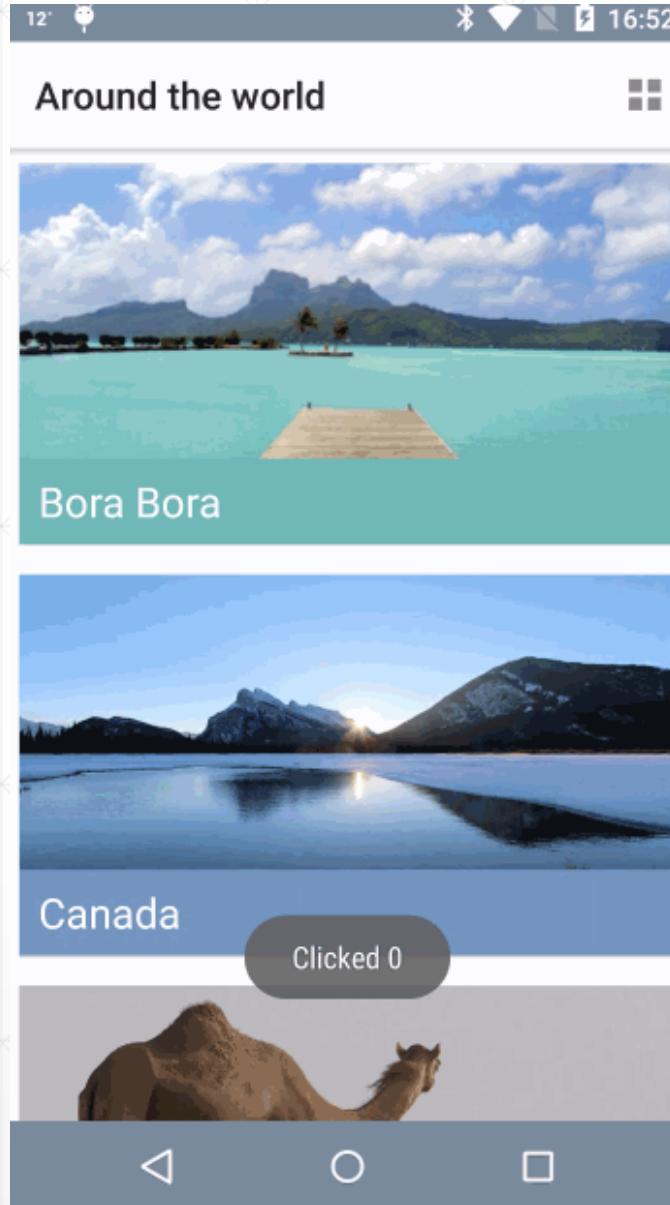
```
placeHolder.setOnClickListener(this);
```

```
public interface OnItemClickListener {  
    void onItemClick(View view, int position);  
}  
  
public void setOnItemClickListener(final OnItemClickListener mItemClickListener) {  
    this.mItemClickListener = mItemClickListener;  
}  
  
@Override  
public void onClick(View v) {  
    if (mItemClickListener != null) {  
        mItemClickListener.onItemClick(itemView, getPosition());  
    }  
}
```

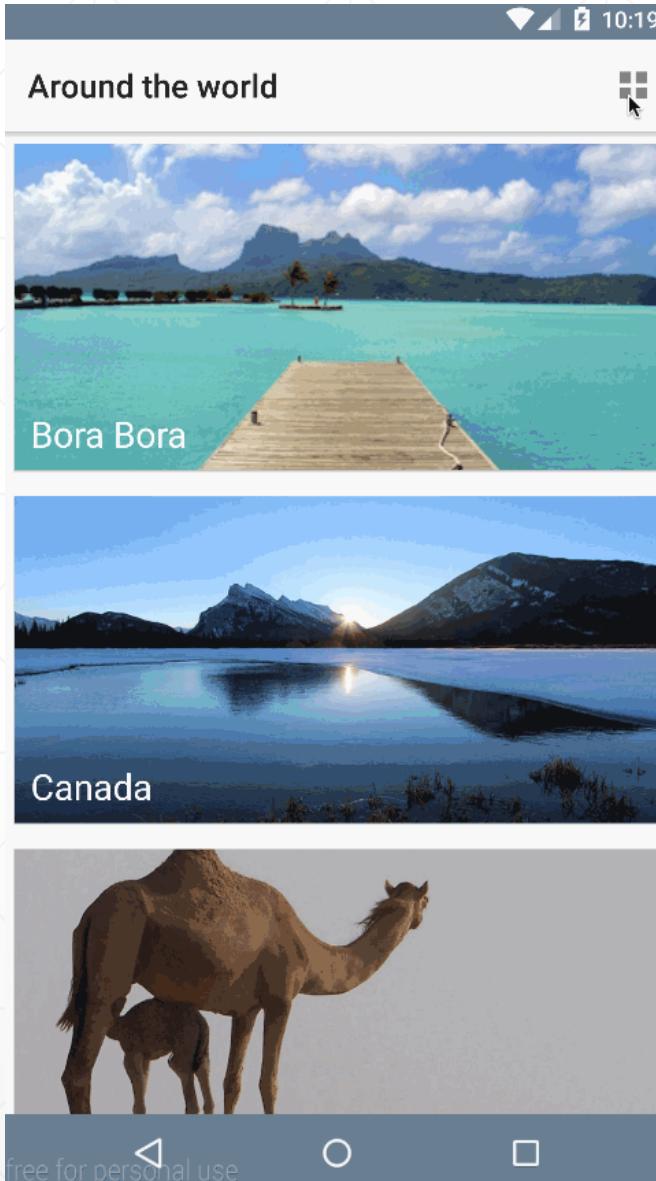
```
TravelListAdapter.OnItemClickListener onItemClickListener = new
TravelListAdapter.OnItemClickListener() {
    @Override
    public void onItemClick(View v, int position) {
        Toast.makeText(MainActivity.this, "Clicked " + position,
                Toast.LENGTH_SHORT).show();
    }
};

mAdapter.setOnItemClickListener(onItemClickListener);
```

Toast notifications



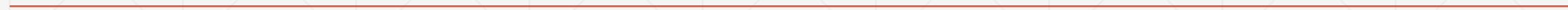
From List to Grid



From List to Grid

```
mStaggeredLayoutManager.setSpanCount(2);
```

```
mStaggeredLayoutManager.setSpanCount(1);
```



```
private void toggle() {  
    MenuItem item = menu.findItem(R.id.action_toggle);  
    if (isListView) {  
        mStaggeredLayoutManager.setSpanCount(2);  
        item.setIcon(R.drawable.ic_action_list);  
        item.setTitle("Show as list");  
        isListView = false;  
    } else {  
        mStaggeredLayoutManager.setSpanCount(1);  
        item.setIcon(R.drawable.ic_action_grid);  
        item.setTitle("Show as grid");  
        isListView = true;  
    }  
}
```

Using the Palette API in the list



Around the world



Using the Palette API in the list

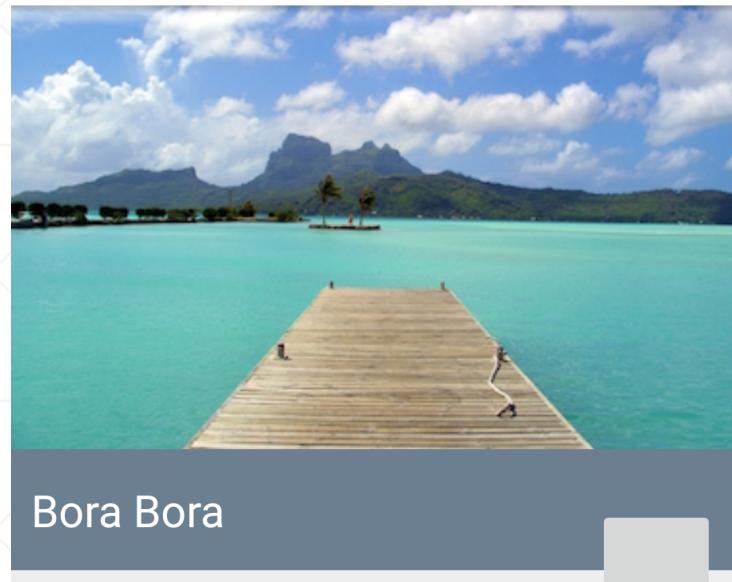
```
Bitmap photo = BitmapFactory.decodeResource(mContext.getResources(), place.getImageResourceId(mContext));

Palette.generateAsync(photo, new Palette.PaletteAsyncListener() {
    public void onGenerated(Palette palette) {
        int mutedLight = palette.getMutedColor(mContext.getResources().getColor(android.R.color.black));
        holder.placeNameHolder.setBackgroundColor(mutedLight);
    }
});
```

Palette API Color Profiles

- Vibrant
 - Vibrant Dark
 - Vibrant Light
 - Muted
 - Muted Dark
 - Muted Light
-

Around the world



MY ACTIVITIES

Material APIs

```
Intent intent = new Intent(MainActivity.this, DetailActivity.class);
intent.putExtra(DetailActivity.EXTRA_PARAM_ID, position);
startActivity(intent);
```

Adding a Floating Action Button with Ripple

```
<?xml version="1.0" encoding="utf-8"?>
<ripple xmlns:android="http://schemas.android.com/apk/res/android"
    android:color="?android:colorControlHighlight">
</ripple>
```



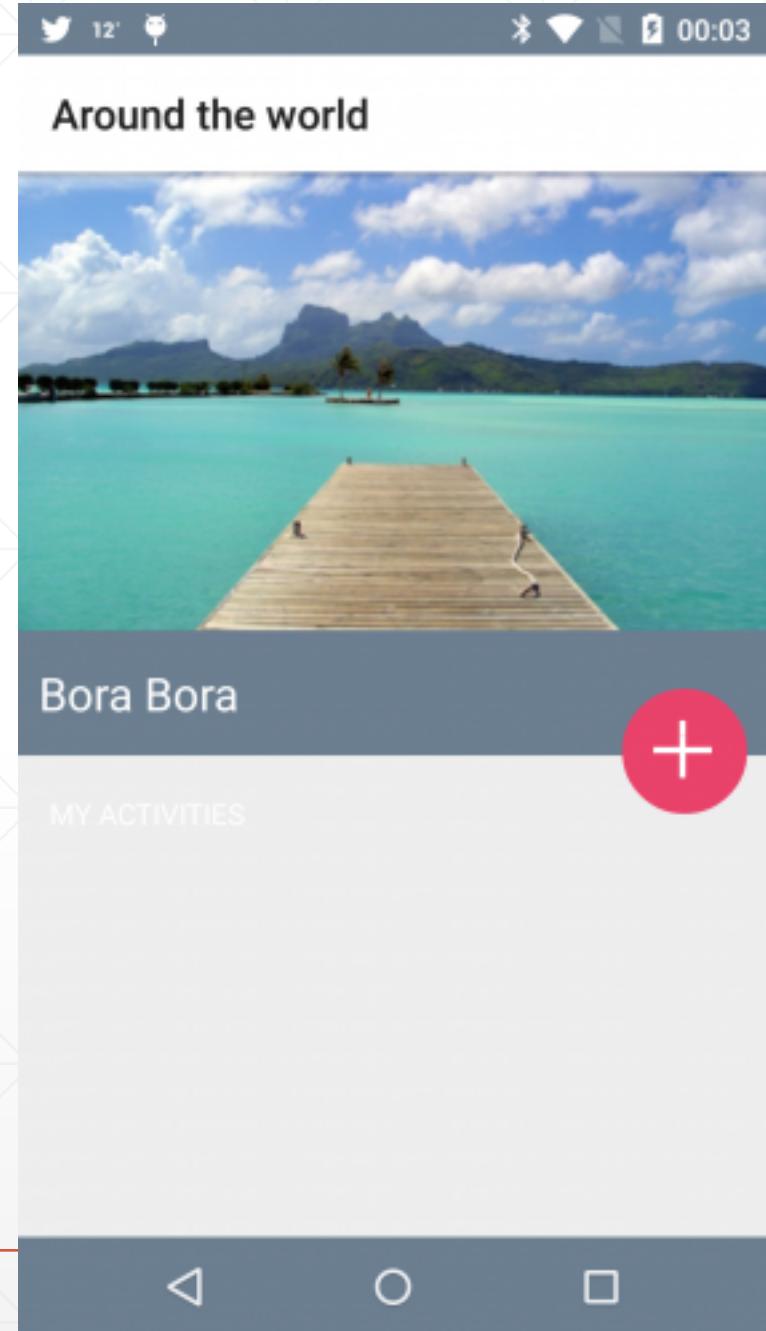
Adding a Floating Action Button with Ripple

```
<?xml version="1.0" encoding="utf-8"?>
<ripple xmlns:android="http://schemas.android.com/apk/res/android"
    android:color="?android:colorControlHighlight">
    <item>
        <shape android:shape="oval">
            <solid android:color="?android:colorAccent" />
        </shape>
    </item>
</ripple>
```



Adding a Floating Action Button with Ripple

```
<ImageButton  
    android:id="@+id	btn_add"  
    android:layout_width="@dimen/floating_button_size"  
    android:layout_height="@dimen/floating_button_size"  
    android:layout_gravity="end"  
    android:layout_marginEnd="10dp"  
    android:layout_marginTop="28dp"  
    android:alpha="0.0"  
    android:background="@drawable/btn_background"  
    android:contentDescription="@null"  
    android:elevation="@dimen/button_elevation"/>
```



Adding a Reveal Animation



—

Adding a Reveal Animation

- `revealEditText()`
- `hideEditText()`



```
private void revealEditText(LinearLayout view) {  
    int cx = view.getRight() - 30;  
    int cy = view.getBottom() - 60;  
    int finalRadius = Math.max(view.getWidth(), view.getHeight());  
    Animator anim = ViewAnimationUtils.createCircularReveal(view, cx, cy, 0, finalRadius);  
    view.setVisibility(View.VISIBLE);  
    isEditTextVisible = true;  
    anim.start();  
}
```

```
private void hideEditText(final LinearLayout view) {
    int cx = view.getRight() - 30;
    int cy = view.getBottom() - 60;
    int initialRadius = view.getWidth();
    Animator anim = ViewAnimationUtils.createCircularReveal(view, cx, cy, initialRadius, 0);
    anim.addListener(new AnimatorListenerAdapter() {
        @Override
        public void onAnimationEnd(Animator animation) {
            super.onAnimationEnd(animation);
            view.setVisibility(View.INVISIBLE);
        }
    });
    isEditTextVisible = false;
    anim.start();
}
```

Around the world

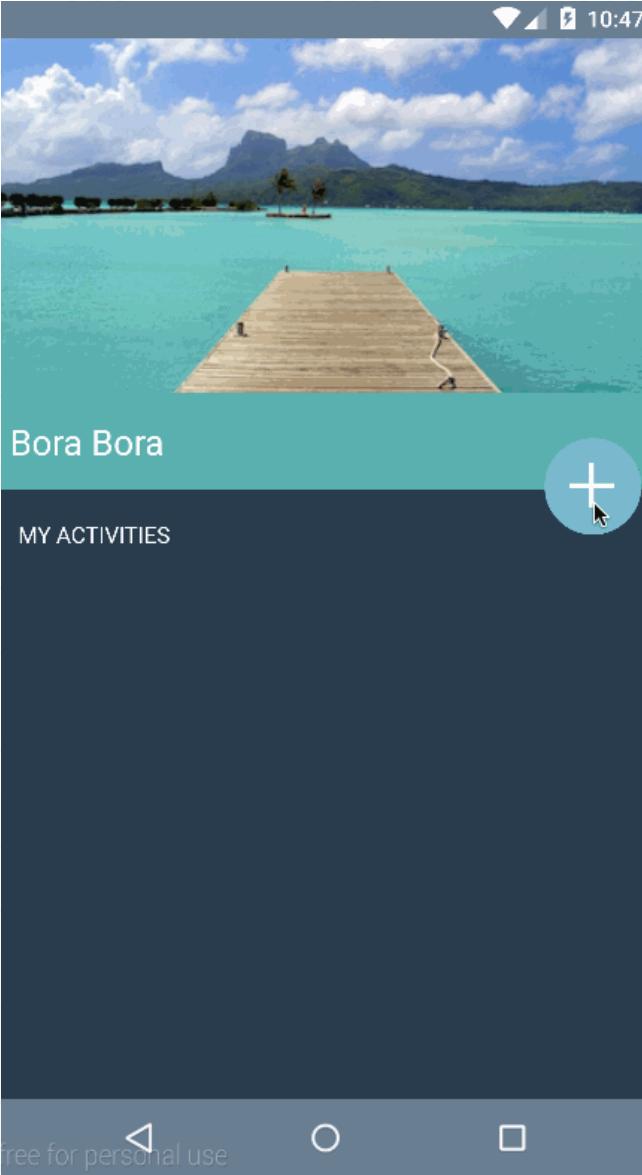


Bora Bora

MY ACTIVITIES



Morphing FAB



Morph FAB

```
<?xml version="1.0" encoding="utf-8"?>
<animated-vector
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:drawable="@drawable/icn_add">
</animated-vector>
```

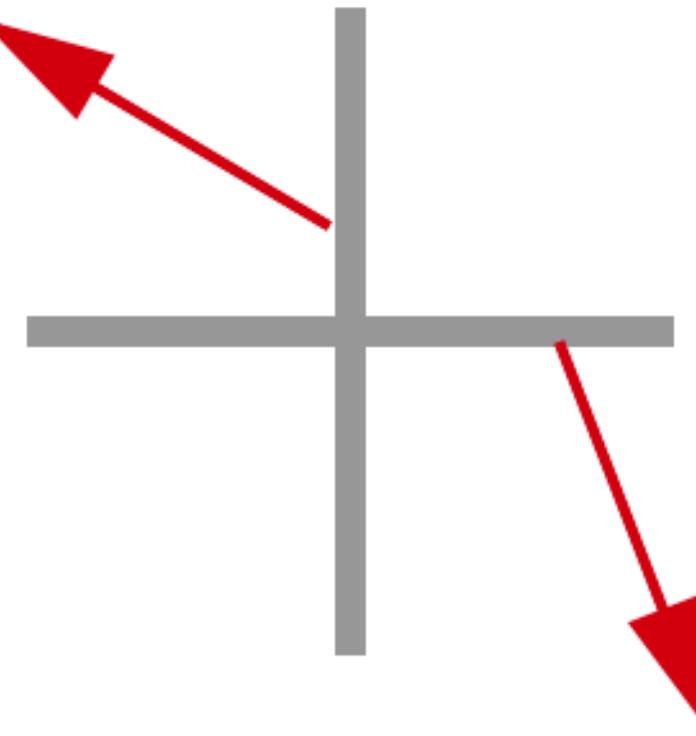
Morph FAB

```
<?xml version="1.0" encoding="utf-8"?>
<animated-vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:drawable="@drawable/icn_add">

    <target
        android:name="sm_vertical_line"
        android:animation="@anim/path_morph" />
    <target
        android:animation="@anim/path_morph_lg"
        android:name="lg_vertical_line" />
    <target
        android:animation="@anim/fade_out"
        android:name="horizontal_line" />

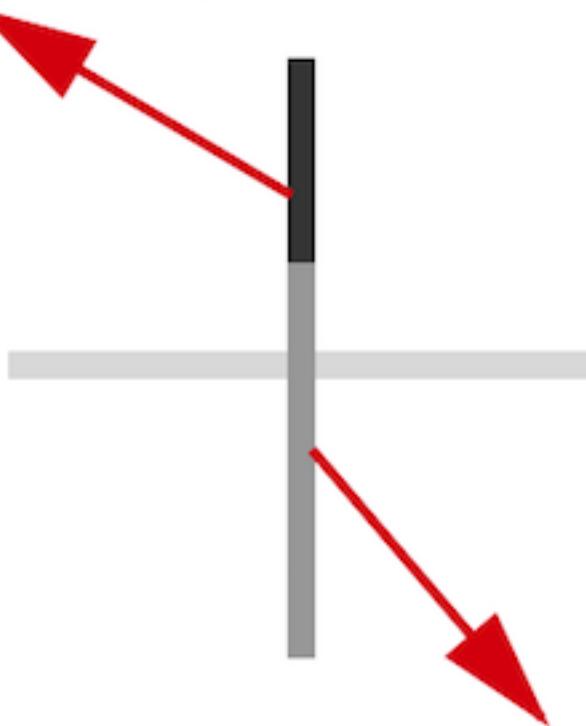
</animated-vector>
```

*Morph the vertical line into a
checkmark*



Fade horizontal line

Smaller vertical line



Larger vertical line

Morph FAB

```
<?xml version="1.0" encoding="utf-8"?>
<animated-vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:drawable="@drawable/icn_add">

    <target
        android:animation="@anim/path_morph_reverse"
        android:name="sm_vertical_line"/>
    <target
        android:animation="@anim/path_morph_lg_reverse"
        android:name="lg_vertical_line" />
    <target
        android:animation="@anim/fade_in"
        android:name="horizontal_line" />

</animated-vector>
```

Morph FAB

Animatable mAnimatable;

```
mAddButton.setImageResource(R.drawable.icn_morp);
mAnimatable = (Animatable) (mAddButton).getDrawable();
mAnimatable.start();
```

```
mAddButton.setImageResource(R.drawable.icn_morph_reverse);
mAnimatable = (Animatable) (mAddButton).getDrawable();
mAnimatable.start();
```

3°



01:46

Around the world



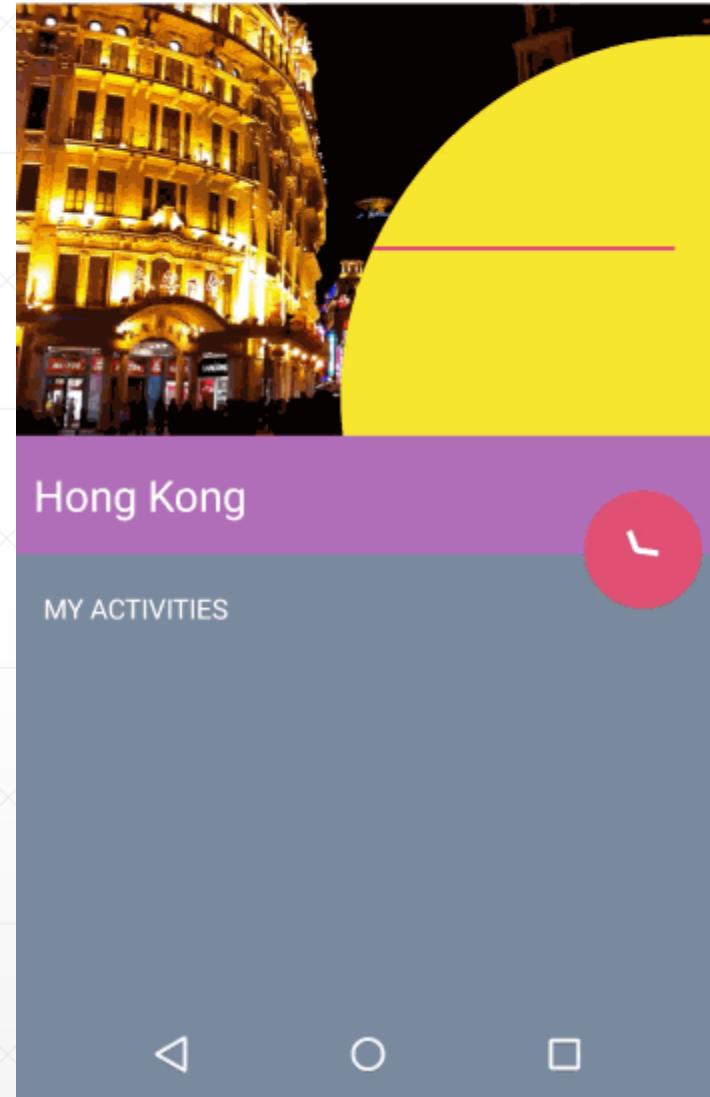
London



Adding Dynamic Colors to Views Using Palette API



Around the world



Adding Dynamic Colors to Views Using Palette API

```
private void colorize(Bitmap photo) {  
    mPalette = Palette.generate(photo);  
    applyPalette();  
}
```

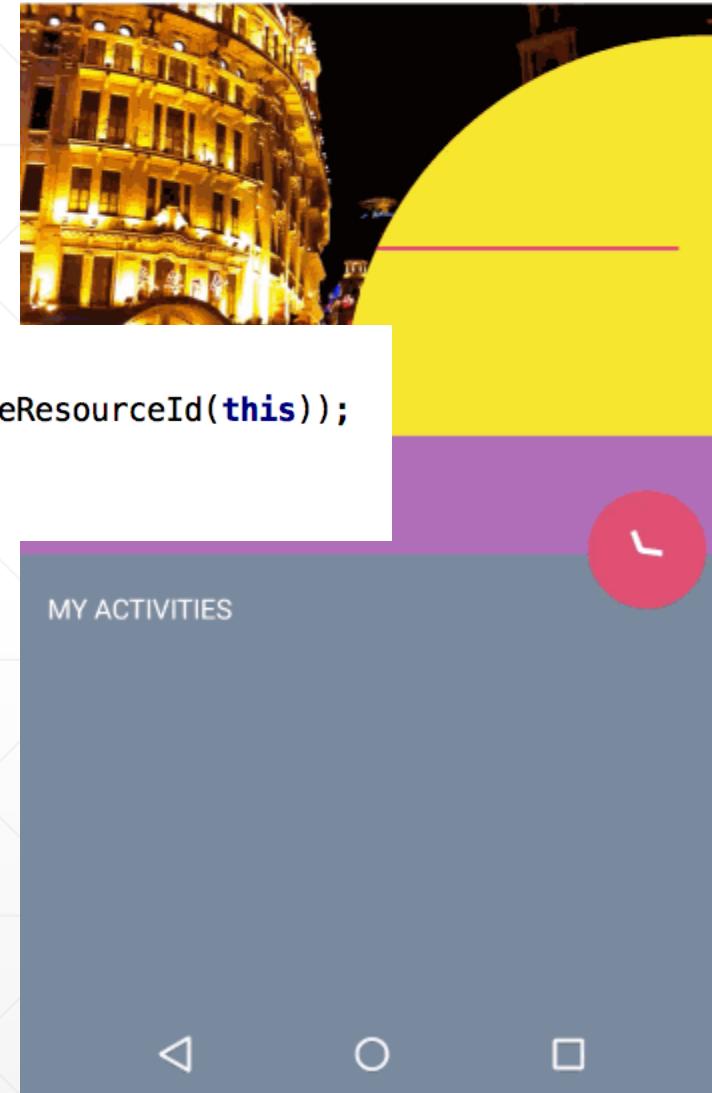
Adding Dynamic Colors to Views Using Palette API

```
private void applyPalette() {  
    getWindow().setBackgroundDrawable(new ColorDrawable(mPalette.getDarkMutedColor(defaultColorForRipple)));  
    mTitleHolder.setBackgroundColor(mPalette.getMutedColor(defaultColorForRipple));  
    mRevealView.setBackgroundColor(mPalette.getLightVibrantColor(defaultColorForRipple));  
}
```

Adding Dynamic Colors to Views Using Palette API

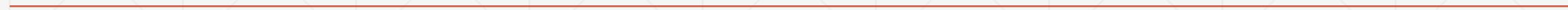
```
private void getPhoto() {  
    Bitmap photo = BitmapFactory.decodeResource(getResources(), mPlace.getImageResourceId(this));  
    colorize(photo);  
}
```

Around the world



Activity Transitions With Shared Elements

- The image of the place;
- The title of the place;
- The background area of the title



Activity Transitions With Shared Elements

```
<ImageView  
    android:id="@+id/placeImage"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:scaleType="centerCrop"  
    android:transitionName="tImage" />
```

Activity Transitions With Shared Elements

```
<LinearLayout  
    android:id="@+id/placeNameHolder"  
    android:layout_width="match_parent"  
    android:layout_height="45dp"  
    android:layout_gravity="bottom"  
    android:orientation="horizontal"  
    android:transitionName="tNameHolder">
```

Activity Transitions With Shared Elements

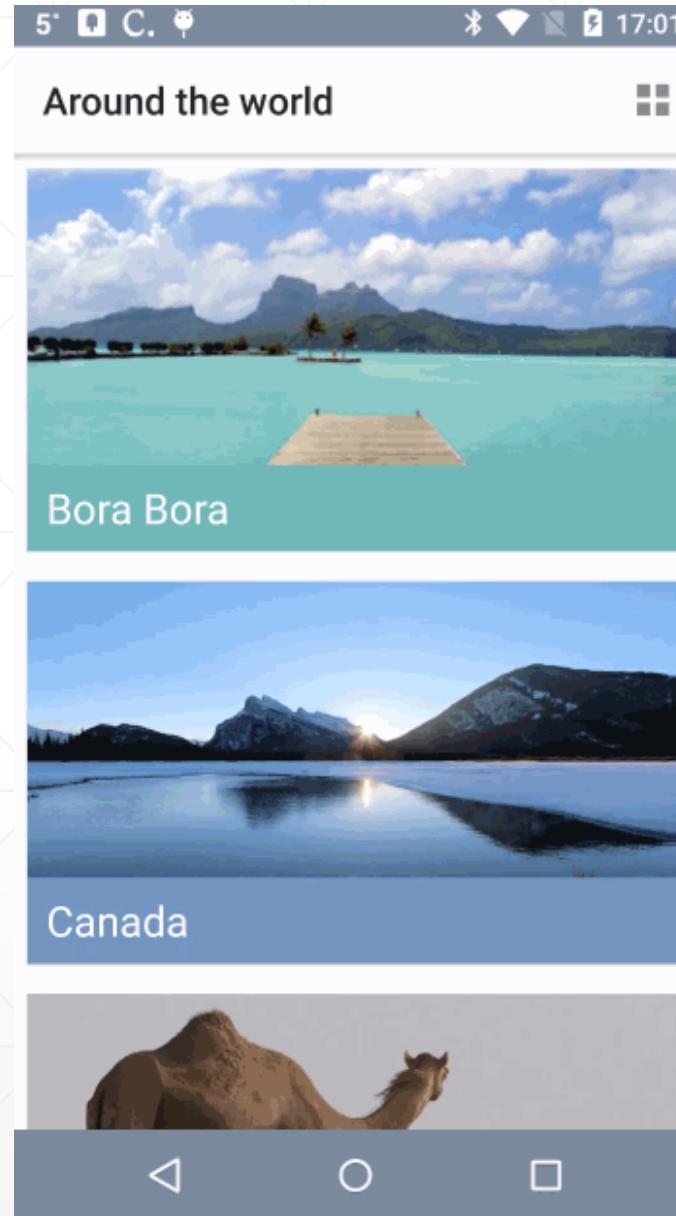
```
<ImageView  
    android:id="@+id/placeImage"  
    android:layout_width="match_parent"  
    android:layout_height="220dp"  
    android:scaleType="centerCrop"  
    android:transitionName="tImage" />
```



Activity Transitions With Shared Elements

```
// 1  
ImageView placelImage = (ImageView) v.findViewById(R.id.placeImage);  
LinearLayout placeNameHolder = (LinearLayout) v.findViewById(R.id.placeNameHolder);  
  
// 2  
Pair<View, String> imagePair = Pair.create((View) placelImage, "tImage");  
Pair<View, String> holderPair = Pair.create((View) placeNameHolder, "tNameHolder");  
  
// 3  
ActivityOptionsCompat options =  
ActivityOptionsCompat.makeSceneTransitionAnimation(MainActivity.this,  
        imagePair, holderPair);  
ActivityCompat.startActivity(MainActivity.this, intent, options.toBundle());
```

Activity Transitions With Shared Elements



Activity Transitions With Shared Elements

```
getWindow().getEnterTransition().addListener(new TransitionAdapter() {  
  
    @Override  
  
    public void onTransitionEnd(Transition transition) {  
  
        mAddButton.animate().alpha(1.0f);  
  
        getWindow().getEnterTransition().removeListener(this);  
  
    }  
  
});
```

Activity Transitions With Shared Elements

```
<ImageButton
    android:id="@+id	btn_add"
    android:layout_width="@dimen/floating_button_size"
    android:layout_height="@dimen/floating_button_size"
    android:layout_gravity="end"
    android:layout_marginEnd="10dp"
    android:layout_marginTop="28dp"
    android:alpha="0.0"
    android:background="@drawable/btn_background"
    android:contentDescription="@null"
    android:elevation="@dimen/button_elevation"/>
```



Action Bar

```
<item name="android:windowActionBar">false</item>
<item name="android:windowNoTitle">true</item>

<b><include android:id="@+id/toolbar" layout="@layout/toolbar" /></b>
```



Action Bar

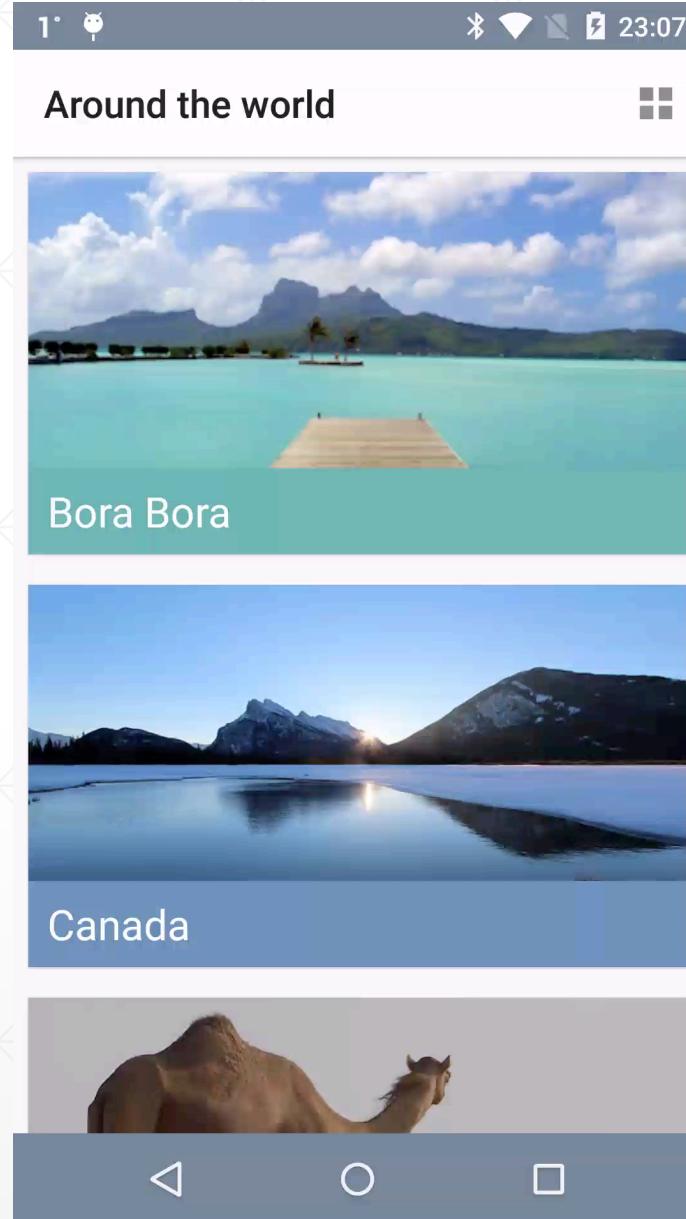
```
private Toolbar toolbar;  
  
toolbar = (Toolbar) findViewById(R.id.toolbar);  
setUpActionBar();  
  
private void setUpActionBar() {  
    if (toolbar != null) {  
        setActionBar(toolbar);  
        getSupportActionBar().setDisplayHomeAsUpEnabled(false);  
        getSupportActionBar().setDisplayShowTitleEnabled(true);  
        getSupportActionBar().setElevation(7);  
    }  
}
```



```
TravelListAdapter.OnItemClickListener onItemClickListener = new TravelListAdapter.OnItemClickListener() {
    @Override
    public void onItemClick(View v, int position) {
        // 1
        Intent transitionIntent = new Intent(MainActivity.this, DetailActivity.class);
        transitionIntent.putExtra(DetailActivity.EXTRA_PARAM_ID, position);
        ImageView placeImage = (ImageView) v.findViewById(R.id.placeImage);
        LinearLayout placeNameHolder = (LinearLayout) v.findViewById(R.id.placeNameHolder);
        // 2
        View navigationBar = findViewById(android.R.id.navigationBarBackground);
        View statusBar = findViewById(android.R.id.statusBarBackground);

        Pair<View, String> imagePair = Pair.create((View) placeImage, "tImage");
        Pair<View, String> holderPair = Pair.create((View) placeNameHolder, "tNameHolder");
        // 3
        Pair<View, String> navPair = Pair.create(navigationBar,
            Window.NAVIGATION_BAR_BACKGROUND_TRANSITION_NAME);
        Pair<View, String> statusPair = Pair.create(statusBar, Window.STATUS_BAR_BACKGROUND_TRANSITION_NAME);
        Pair<View, String> toolbarPair = Pair.create((View)toolbar, "tActionBar");
        // 4
        ActivityOptionsCompat options = ActivityOptionsCompat.makeSceneTransitionAnimation(MainActivity.this,
            imagePair, holderPair, navPair, statusPair, toolbarPair);
        ActivityCompat.startActivity(MainActivity.this, transitionIntent, options.toBundle());
    }
};
```

Material Design



Bullet Points

- Implement the material theme;
 - Build dynamic views using new widgets like RecyclerView and CardView;
 - Use Palette API to generate color schemes that you can use for text or background views;
 - Create interactions using the animation APIs.
-