Android XII

Animations
Kinds of Animation in Android SDK

- Property animation
  - allows the changing of view properties through time

- View animation
  - Tween animation that can perform a series of simple transformations (position, size, rotation, and transparency) on the contents of a View object
  - Deprecated

- Drawable animation
  - Drawable animation lets you load a series of Drawable resources one after another to create an animation

- OpenGL ES: Low level coding for max flexibility, gaming

- Outside the SDK: Material Design Animations, LibGDX(gaming), Unity(gaming)--- code the whole app in Unity!
**Property Animation**

- Use ObjectAnimator (a subclass of ValueAnimator) to define duration, value, property name, repeat, repeatMode, etc for one or two properties at a time
  - Can define at runtime or with xml resource file:
    ```xml
    <objectAnimator xmlns:android="http://schemas.android.com/apk/res/android"
        android:duration="1000"
        android:valueTo="200"
        android:valueType="floatType"
        android:propertyName="y"
        android:repeatCount="1"
        android:repeatMode="reverse"/>
    ```

- Or use a ValueAnimator for more control
- Both involve an Interpolator: a class that defines the rate of change
- See demo for coding details
Properties that can be animated

- translationX and translationY - controls the location of the view. This value is added to the left or top coordinates of the view which are set by the layout.
- rotation, rotationX and rotationY - used to control the 2D and 3D rotation of the view.
- scaleX and scaleY - control the 2D scaling of a view.
- pivotX and pivotY - control the location of the view’s pivot point.
- x and y - these are utility properties describing the final location of the view in its container (by summing the left and top values and translationX and translationY values).
- alpha - controls the transparency of a view.

credit: https://www.101apps.co.za/articles/a-property-animation-tutorial.html
Animation Options

- `setRepeatCount` – specifies the number of times the animation should repeat after the initial run.
- `setRepeatMode` – defines what this animation should do when it reaches the end.
- `setDuration` – specifies the animation’s total duration.

https://www.raywenderlich.com/128105/introduction-android/animations
Animator Sets and Listeners

- Animations can be bundled together in an AnimatorSet.
- Animation listeners can be registered to listen for events like animation starts, ends, repeats, etc.
Drawable Animation

- Takes a series of images and displays them sequentially
- Define the animation in your resources folder (res/drawables)
- Get a ref to it in your view and run

```xml
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="true">
    <item android:drawable="@drawable/mario1" android:duration="200" />
    <item android:drawable="@drawable/mario2" android:duration="200" />
    <item android:drawable="@drawable/mario3" android:duration="200" />
    <item android:drawable="@drawable/mario4" android:duration="200" />
    <item android:drawable="@drawable/mario5" android:duration="200" />
    <item android:drawable="@drawable/mario6" android:duration="200" />
    <item android:drawable="@drawable/mario7" android:duration="200" />
</animation-list>
```