

# Drawable Animation

Drawable animation lets you load a series of Drawable resources one after another to create an animation. This is a traditional animation in the sense that it is created with a sequence of different images, played in order, like a roll of film. The [AnimationDrawable](/reference/android/graphics/drawable/AnimationDrawable.html) (</reference/android/graphics/drawable/AnimationDrawable.html>) class is the basis for Drawable animations.

While you can define the frames of an animation in your code, using the [AnimationDrawable](/reference/android/graphics/drawable/AnimationDrawable.html) (</reference/android/graphics/drawable/AnimationDrawable.html>) class API, it's more simply accomplished with a single XML file that lists the frames that compose the animation. The XML file for this kind of animation belongs in the `res/drawable/` directory of your Android project. In this case, the instructions are the order and duration for each frame of the animation.

The XML file consists of an `<animation-list>` element as the root node and a series of child `<item>` nodes that each define a frame: a drawable resource for the frame and the frame duration. Here's an example XML file for a Drawable animation:

```
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="true">
    <item android:drawable="@drawable/rocket_thrust1" android:duration="200" />
    <item android:drawable="@drawable/rocket_thrust2" android:duration="200" />
    <item android:drawable="@drawable/rocket_thrust3" android:duration="200" />
</animation-list>
```

This animation runs for just three frames. By setting the `android:oneshot` attribute of the list to `true`, it will cycle just once then stop and hold on the last frame. If it is set `false` then the animation will loop. With this XML saved as `rocket_thrust.xml` in the `res/drawable/` directory of the project, it can be added as the background image to a View and then called to play. Here's an example Activity, in which the animation is added to an [ImageView](/reference/android/widget/ImageView.html) (</reference/android/widget/ImageView.html>) and then animated when the screen is touched:

```
AnimationDrawable rocketAnimation;

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    ImageView rocketImage = (ImageView) findViewById(R.id.rocket_image);
    rocketImage.setBackgroundResource(R.drawable.rocket_thrust);
    rocketAnimation = (AnimationDrawable) rocketImage.getBackground();
}

public boolean onTouchEvent(MotionEvent event) {
    if (event.getAction() == MotionEvent.ACTION_DOWN) {
        rocketAnimation.start();
        return true;
    }
    return super.onTouchEvent(event);
}
```

It's important to note that the `start()` method called on the `AnimationDrawable` cannot be called during the `onCreate()` method of your Activity, because the `AnimationDrawable` is not yet fully attached to the window. If you want to play the animation immediately, without requiring interaction, then you might want to call it from the `onWindowFocusChanged()` ([/reference/android/app/Activity.html#onWindowFocusChanged\(boolean\)](/reference/android/app/Activity.html#onWindowFocusChanged(boolean))) method in your Activity, which will get called when Android brings your window into focus.

For more information on the XML syntax, available tags and attributes, see [Animation Resources](/guide/topics/resources/animation-resource.html) (</guide/topics/resources/animation-resource.html>).

