# Add dice images

In our app, we have simulated the dice, we are rolling in our game, with a random number generator. But to make the game more interesting, it would be great to show actual dice images to make the game more real. In Android, we can upload images and display them in our app. The widget we will use is Image View. This time though, we're going to create them entirely in code using XML tags to position and load our images in the app. First, we need to load our images. I have six images that are the various faces to the dice. They are located in the exercise files for this video in the Assets folder.

These files need to be imported into our project. In Android Studio, in the Project Panel, look for the res folder. Inside of this, you might have a folder called drawable. In my case I don't. The drawable folder is where we're going to be adding our images. I need to add the folder. So I'm going to right click on res and select New and then select Android resource directory. This will allow us to add a new, specific, Android resource directory that will then be linked with our project. I'm going to change the resource type and scroll to the top and select drawable.

Go ahead and click OK. This will create a new drawable folder that we can then add our images into. Go to the Finder or Explorer, select the images and then copy to the clipboard. Then we'll right click on the drawable folder and then we'll select paste. Click OK to confirm copying the files. With images added, we can now create our image view widgets and lay them out in the activity. Open content\_main.XML and we're going to use the text view to edit the XML code.

The content main XML file defines the layout of the white area in the center of our app. It is, actually, only part of the layout. There is another file called activity\_main.XML. Go ahead and open that to view it. If you go to layout and then co to acitivty\_main.XML. You'll see here that there are a number of other components including widgets for the app bar, the floating action button with the envelope icon and more. If you look carefully, you'll see a tag called include. This tag takes the layout that we're using for our app, content\_main and is inserting it into this section.

So the layout we have been working with is a section of the overall app. If you return to the content\_main.XML, we can add our image view widget. First, go to the bottom and add some space below the button widget and we'll add in our new tag, ImageView. If you press enter, it will automatically input some of the common properties of the widget. The first we're going to define is width. We'll set that to 100dp. Then, we'll go to the second one which is height. We'll set that to 100dp as well.

We need to give our ImageView an ID so we can refer to it using code. We'll add that using android:id and we'll set that equal to @+id/die1Image. As you've been entering these value in, you'll start to see the preview in the IDE update. Currently, it is displayed in the upper left corner of the content section of the layout. That is because, if you remember, the layout we are working with is included in the outer layout specified by activity\_main.XML.

Since we're working with a relative layout here, we can position it based on the other widgets. We can position it below the roll result text view widget. To do that we'll add in a new attribute, android:layout\_below and we'll set that equal to @id/rollResult. Now we need to add some space between the text view and the new image view. We can add a margin to the top of the widget. Let's set it to 25 device pixels.

Android:layout\_marginTop=25dp. Next, we need to make sure that it stays aligned to the left. We want this to be aligned to the left of the overall container, not to another widget. So we want to align it based on the parent, the outer layout container. We'll set this as android:layout\_alignParentLeft and we'll set that =true.

Last, we need to provide a reference to the image that we want to display. We will display the die with one with a single pip on it for all three. We will use an "at" reference to refer to our drawable folder in our project. So we'll set this as, android: source, spelled src=@drawable/die\_1. Notice that we don't include the .png in the reference here? That's it, we have our first image. So let's go ahead and create our second one.

We can take the code that we already created, highlight it, right click and select copy. Then we can go underneath and then paste. We will have to change a few things. First, we need to change the ID. We'll change this to die2Image. Then we want to change the alignment to place this in the center instead of on the left. So we'll change layout\_alignParentLeft to layout\_centerHorizontal. As you can see, the roll button overlaps with our image.

We'll fix this in a little bit. Now we can copy and past the code again for our third image. We'll change our ID to three and we'll change our alignment to layout\_alignParentRight. Now we need to fix the roll button. Remember, this is a relative layout so we want to move it down. It currently is positioned based on the roll result widget. We can change that reference to position it based on the second die image instead.

So if we scroll up in our text, we can find our button. We can see here that the layout below attribute is set to position us below roll result. We can change this to die2Image which then moves it down. That's it, our basic layout is created for our three dice and are filled in with images we imported into our project.