Add some Buttons

We will add our buttons below the drum images
Play, Record, Stop

We have 3 new images for these buttons
https://github.com/shawnlg/flutter_ball/tree/09_buttons

Add the new images into the assets folder with the other drum images
Add them to pubspec.yaml and to main.dart

As you get more familiar with Dart, Flame, and our game, I will be giving less details on the things you have done many times. If you get stuck, you can always go to GitHub.

The tool Beyond Compare by Scooter Software
You can easily compare two Flutter projects to see what is different

Sprite vs. Sprite Component

Our Drum game component was a SpriteComponent.

All of the features of a Component plus a built-in Sprint image

A Sprite is just the image itself. If your component needs to draw more than one image, you can have it render Sprites directly.

Our DrumController will render the 3 buttons as images.

It will check if the user taps on any of the buttons during its update()

Render will either show the Play and Record buttons or the Stop button depending on what we are doing.

Add Button Sprites

We start by importing the Sprite class in our drum controller file

```
import 'package:flame/sprite.dart';
```

We add instance variables, creating the 3 Sprint classes, one for each button

```
Sprite recordButton = Sprite('drum/record.jpg');
Sprite playButton = Sprite('drum/play.jpg');
Sprite stopButton = Sprite('drum/stop.jpg');
```

We define where we want the buttons to be – in 2 rectangles

```
Rect button1Rect;   // where to place the first button
Rect button2Rect;   // where to place the second button
```

Add Button Sprites

We start by importing the Sprite class in our drum controller file

```
import 'package:flame/sprite.dart';
```

We add instance variables, creating the 3 Sprint classes, one for each button

```
Sprite recordButton = Sprite('drum/record.jpg');
Sprite playButton = Sprite('drum/play.jpg');
Sprite stopButton = Sprite('drum/stop.jpg');
```

We define where we want the buttons to be – in 2 rectangles

```
Rect button1Rect;   // where to place the first button
Rect button2Rect;   // where to place the second button
```

We now put the logic in our render() method to display the right buttons depending on what we are doing

```
void render(Canvas c) {
  // what we do depends on the state of the drum machine
  switch (state) {
    case State.TAP:  // show the record and play buttons
      recordButton.renderRect(c, button1Rect);
      playButton.renderRect(c, button2Rect);
      break;
    case State.RECORD:  // show the stop button
      stopButton.renderRect(c, button2Rect);
      break;
    case State.PLAY:  // show the stop button
      stopButton.renderRect(c, button2Rect);
      break;
    default:
  }
}
```

You can see how easy it is to draw an image on the screen.

The Sprite has a renderRect() method where you pass it the canvas and a rectangle where the image goes.

We only draw buttons when we are in 3 of the states

TAP, RECORD, PLAY

One last thing – calculate where the 2 button rectangles will be placed on the screen

Done in resize() when we get the screen size

Add this to resize()

```
// save the button locations
button1Rect = Rect.fromLTWH(pctX*10, pctY*80, pctX*20, pctX*20);
button2Rect = Rect.fromLTWH(pctX*40, pctY*80, pctX*20, pctX*20);
```

We use percentages again to put the buttons near the bottom.

Run your game. You should see the Play and Record button images

Nothing happens when you tap them

Next up, how we detect button taps and use them to control the state of the drum machine

Remember to compare your code with that in GitHub if you have any problems.