Creating variation in the balls

Random colors

We want to make each ball have a random color. To do this open up your `ball.py` file. Locate the line inside the `__init__` function that has to do with color. Notice that this line has 3 numbers in it. In programming, these are the red, green, and blue values that you want to set the pixel colors to. Experiment with changing these numbers and re-running your program. Can you get your balls to be all green? How about all blue? Can you figure out how you would make them yellow? Try different numeric combinations to see what colors the balls change to. (Hint: The number always has to be within the range 0-255 inclusive).

That was rather exciting. Now, lets make each ball we create a random color. Comment out the line you have been changing above (by placing a `#` at the front of it). Right after that line, let’s add the code:

```python
self.set_color()
```

This will call a function to set a color for us. We don’t have that function yet, so we will now create it. At the very bottom of `ball.py` create a new function that looks like this:

```python
def set_color(self):
    r = random.randrange(0,255)
    g = random.randrange(0,255)
    b = random.randrange(0,255)
    self.color = [r,g,b]
```

Notice that we have set those red, green, and blue color values to take on a random value between 0-255. Make sure your code runs!

Random radii

You should be able to follow the above logic and have each ball have a random radius. We could simply change the existing line in `__init__` (where it sets the radius) to the following:

```python
self.radius = random.randrange(self.BALL_MIN_RADIUS, self.BALL_MAX_RADIUS)
```

Now we are picking a random radius between those 2 values. I think that 100 is a bit much for the max, so you might want to adjust it. I ended up setting my `BALL_MIN_RADIUS` to 1 and the `BALL_MAX_RADIUS` to 5.

Random direction

If you look in the `__init__` function, you will see 2 variables that determine the balls rate of motion, `dx` and `dy`. By default they are both set at 1, meaning for each new frame the ball will move 1 unit to the right and 1 unit down (at least until it collides with the window edge). Let’s make them move in random directions.

Change those lines to something like this:

```python
self.dx = random.choice([1,-1])
self.dy = random.choice([1,-1])
```

Now, the balls will have random directions, but they are all moving at the same rate of speed. Test your code at this point to see if you like it.

Random speeds

If we wanted them to have more random speeds, we would modify the above `dx` and `dy` values that set the direction and rate of motion. Something like this should work:

```python
self.dx = random.randrange(1,5) * random.choice([1,-1])
self.dy = random.randrange(1,5) * random.choice([1,-1])
```