

Lab 6 - Collisions

Before you start:

Copy the following code into your python file:

```
from turtle import *
import random

class Ball(Turtle):
    def __init__(self, radius, color, speed):
        Turtle.__init__(self)
        self.shape("circle")
        self.shapesize(radius/10)
        self.radius = radius
        self.color(color)
        self.speed(speed)
```

And... now what?

Alright, as you just learned ball collisions, we will be working on that in python now!

To start off, create two ball objects with different radii.

Now, add a new function inside the code with the name `check_collision` that takes two balls as parameters. Your code should look something like this.

```
from turtle import *
import random

class Ball(Turtle):
    def __init__(self, radius, color, speed):
        Turtle.__init__(self)
        self.shape("circle")
        self.shapesize(radius/10)
        self.radius = radius
        self.color(color)
        self.speed(speed)

# Code to create two ball objects here

def check_collision(ball1, ball2):
    # Logic to check if the balls collide.
```

Inside this function, check if the two balls collide. If they do, make them do something crazy, like, change colors, or anything of your choice.

Alright, I finished my code...

Did you try the code? Does it work?

I did it! It works!

Are you sure it works? Did you try the edge cases?

Wait what...? What is an edge case?

An edge case is a case where we are given a value that requires special handling in the code. In this case, for example, edge cases are colliding the two balls from all the different sides (Positive Y, Negative Y, Positive X, Negative X)

If you have checked the edge cases and they all work, move onto the next part of the lab.

Alright... so what do I do now?

Easy, tiger. Since you already did the collisions method and know how to apply it in the code, it's time for something more challenging.

In your code, create a new list that contains the two balls (or how much balls you want in the code).

Create a new function inside the code (not inside any class) that takes the list of balls and checks if any of the balls collide with each other (collision of pairs). If they do collide, check which ball has the smaller radius and make it teleport! (Use `random.randint` for the teleportation co-ordinates)

Are you done early?

Alright, it's time to push your code to Github!

Check out www.tinyurl.com/YL1-Github for instructions.

Scroll down to the last 6 slides for the code uploading part!