

Lesson 15: Collision Detection

Lesson aim:

To learn how to tell when two objects have hit each other. This is called Collision Detection.

Why:

Collision detection is used all the time in games. We need to know when two objects collide: the player hitting power ups, bullets hitting players, racers colliding with the edge of the track, and many other examples.

How to do collision detections

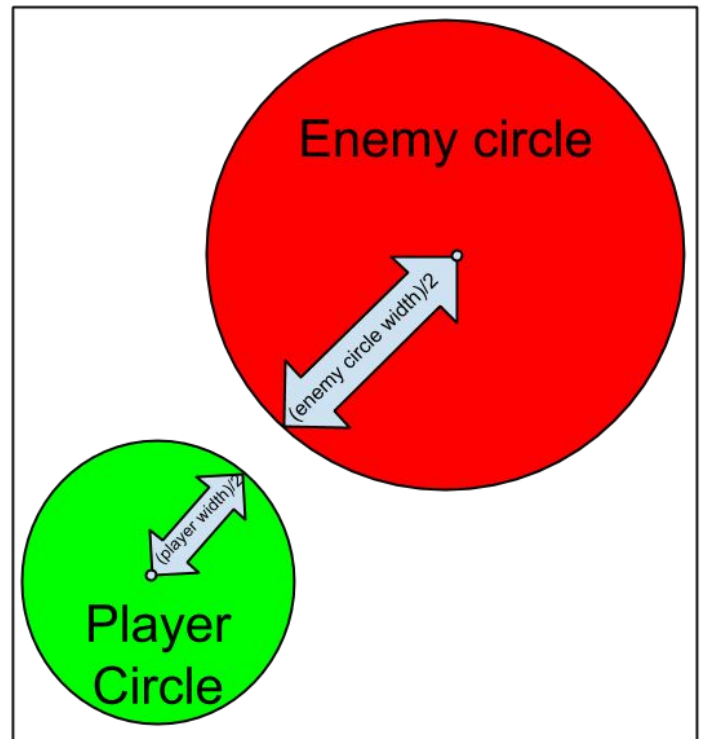
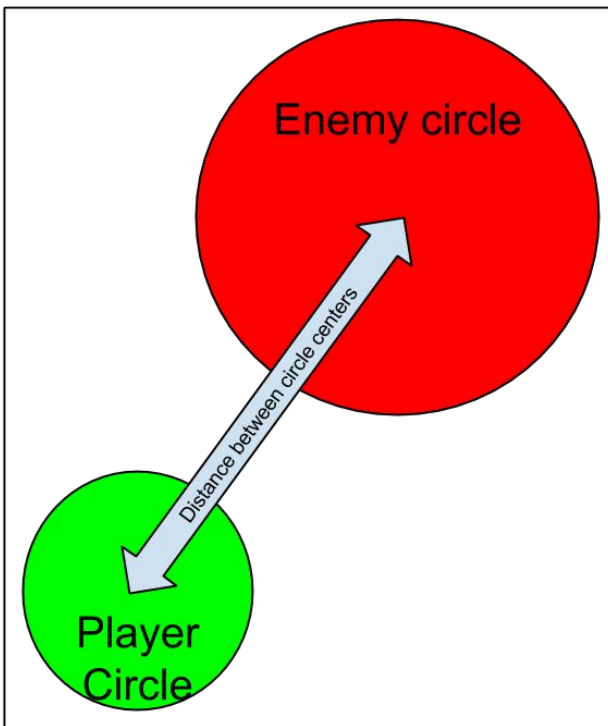
We need to check when two circles overlap.

That's when the collision has happened.

We need to know **two** pieces of information:

1. The distance between the centres of the two circles.
2. The distance between the centre of each circle and the edge, added together. The two radiuses added together.

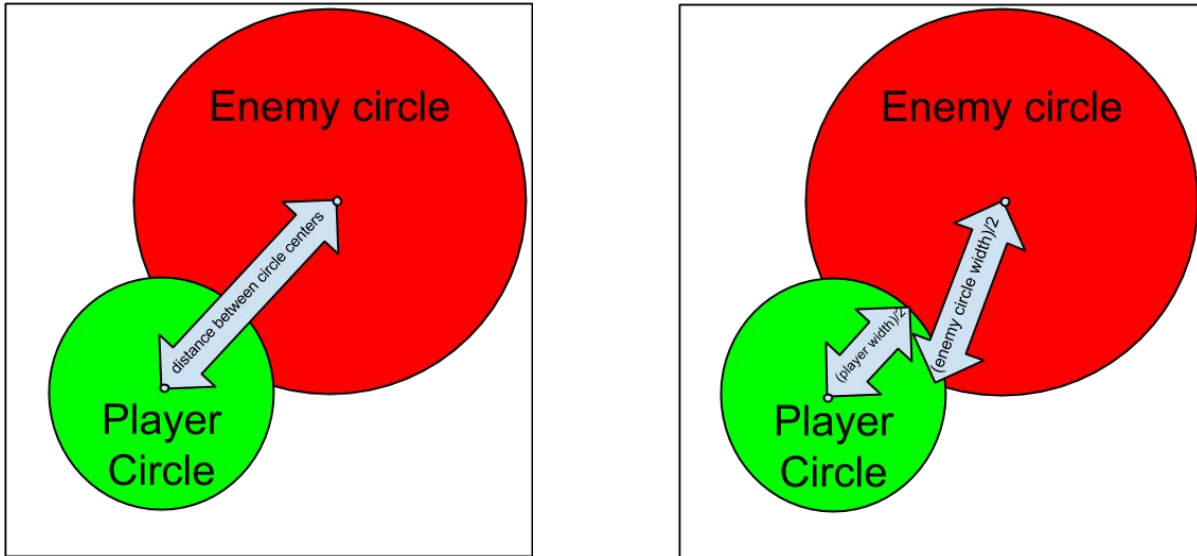
If the **length of the two radiuses added together** is *bigger* than the **distance between the two centres**, the circles have collided.



See the distance between the circles (left) is greater than the two radius' added together (show on right)

So: $(\text{playerRadius} + \text{enemyRadius}) < \text{distanceBetweenCentres}$

Scenario where the two circles are colliding:



In the scenario above: compare the length of the *distance between the centres* (**LEFT**) and the length of the two radiuses added together (**RIGHT**).

$(\text{playerRadius} + \text{enemyRadius}) > \text{distanceBetweenCentres}$

How do we get these three values?

We **already know** the radius of the enemy circle and the radius of the player circle. It is half the width and height of a circle ellipse. (This is much more complicated for an oval - forget about that and let's stick with circles!) For the below circle the radius is $120 / 2 = 60$:

```
ellipse(mouseX,mouseY,120,120);
```

To get the distance **between two circle centres** we use the `dist` function. This takes 4 point (horizontal point 1, vertical point 1, horizontal point 2, vertical point 2) and returns the distance between the two points.

```
ellipse(mouseX,mouseY,120,120); //player circle
ellipse(400,400,500,500); //enemy circle
text("distance between circle centers is: " +
dist(mouseX,mouseY,400,400),50,25); //show the distance on the screen
text("radius player circle is: " + 60, 50,50); //120/2 =60
text("radius enemy circle is: " + 250,50,75); //500/2 =250
```

Task:

1. Use your knowledge to extend the above program so that **if** the circles collide (the circle radiuses added together is **greater than** the distance between the centers) the screen changes colour.
2. Extend the program so you are controlling the player circle using the keyboard.
3. Further extend the program so the enemy ball is dropping down the screen.