

Pygame Assignment - Bouncing Ball Edit

In class you saw a demo in pyGame with movement:

a) Bouncing Ball - which displayed a ball bouncing in a window, following the law of reflection.

For this assignment, you'll edit the bouncing ball program.

Edit A - Wrap Around Ball

Edit the code so that the ball "wraps around the screen". So, for example, if the ball was at $x = 300$, $y = 600$, at the bottom of the screen three tenths of the way over from the left to the right, the ball should reappear, moving in the same direction, at $x = 300$, $y = 0$. Similarly, if the ball was at $x = 1000$, $y = 200$, the ball should wrap around, reappearing at $x = 0$, $y = 200$ and continue moving in the same direction.

Edit B - Wrap Around or Bounce?

Edit the code from part A so that now, instead of always wrapping around (you'll see this is a bit boring after a while), the ball wraps around 10 times and then bounces 10 times and goes back and forth between bouncing and wrapping around. By wrapping around 10 times, we mean that the ball must hit some boundary and reappear on the other side 10 times.

Edit C - Two Balls

Edit your code so that there are two balls of the same size with two different colors. Have the balls moving at different speeds but bouncing around. Don't worry about intersections, just let them happen. In the future we'll learn how to detect these and then take some action based on them.

Edit D - One Ball Following Another

Take your solution to part C and change the second ball so it always follows the previous ball. Basically, have both balls follow the same steps, but just start the second ball a fraction of a second after the first.

Edit D - Planetary motion (optional)

For this one, you may want to start from scratch instead of editing the bouncing ball program. Create a scene where the sun (a filled yellow circle) is in the middle of the screen and two planets are orbiting the sun at different radii with different rates of speed. This edit is probably the easiest to do with sin and cos. If you are unfamiliar with these functions, you can skip this part.