

# C2 Bouncing Ball

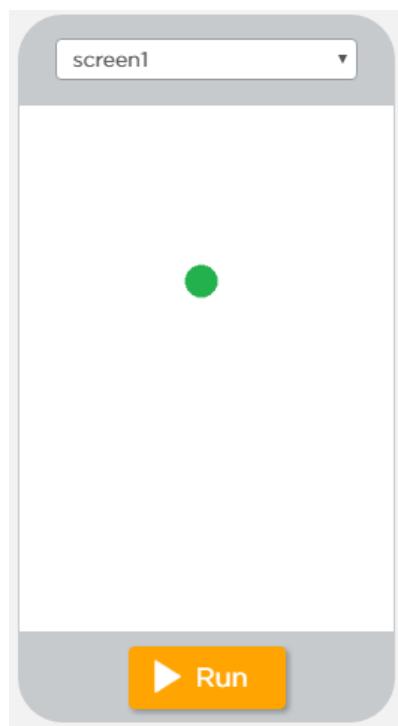
Media required:

- ball.png
- ball2.png

## Design view

Create this screen.

It is only one image on it – it uses the ball.png file. The image's id is ball.



Code view.

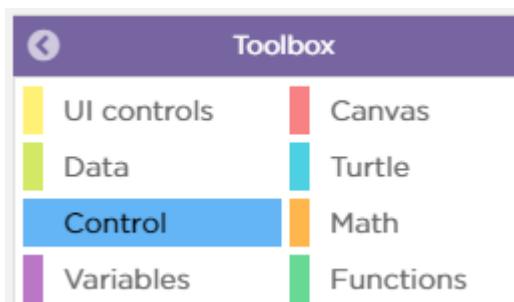
Begin with these three variables.

```
var x = 126;  
var y = 98;  
var dir = "up";
```

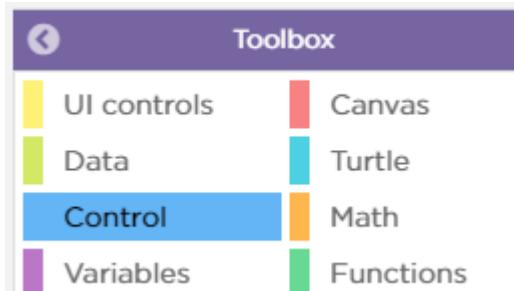
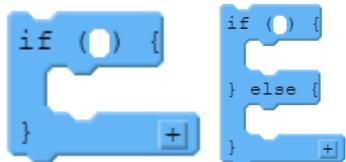
Then, because we have no buttons, we need no onEvent blocks. Instead all of our other code will go in a new block.

Under the control tab in the toolbox, select setInterval. Inside of an event occurring when a button is clicked, this event will occur all the time.

```
setInterval(callback, ms)
```



Inside it, we are going to use two types of if blocks. Don't mix them up.



```

setInterval( function() {
  if (dir == "up") {
    y = y - 3;
  } else {
    y = y + 3;
  }
  if (y <= 0) {
    dir = "down";
  }
  if (y >= 450) {
    dir = "up";
  }
  setPosition("ball", x, y, 32, 32);
}, 20);

```

The first block is an if/else block. It checks the direction and changes the y variable accordingly.

The second block is only an if block. It checks if we hit the top wall and switches the direction.

The third block is also an if block. It checks if we hit the bottom wall and switches the direction.

The last block moves the ball to it's new x,y spot.

The 20 is the millisecs between movements.

Run the code. The ball should bounce up and down.

Now that is working, we will add the right and left.

```

1 var x = 126;
2 var y = 98;
3 var dir = "up";
4 var side = "right";

```

First, add a new variable.

Then, move the ball right and left by switching the x variable.

```
setInterval( function() {
    if (dir == "up") {
        y = y - 3;
    } else {
        y = y + 3;
    }
    if (y <= 0) {
        dir = "down";
    }
    if (y >= 450) {
        dir = "up";
    }
    setPosition("ball", x, y, 32, 32);
}, 20);
```

A Scratch script using the `setInterval` block to move a ball. Inside the loop, there are two parallel `if` blocks: one for moving up (decreasing y) and one for moving down (increasing y). After these, another `if` block checks if the ball has reached the top or bottom boundaries (y <= 0 or y >= 450), changing its direction. Finally, the `setPosition` block moves the ball to its new position. A black arrow points from the `x` variable in the `setPosition` block to the `x` variable in the second `if` block, indicating where it should be placed.

Then, add the ifs to check if the walls were hit and to change the directions accordingly.

```
setInterval( function() {
    if (dir == "up") {
        y = y - 3;
    } else {
        y = y + 3;
    }
    if (side == "left") {
        x = x - 3;
    } else {
        x = x + 3;
    }
    if (y <= 0) {
        dir = "down";
    }
    if (y >= 450) {
        dir = "up";
    }
    if (x <= 0) {
        side = "right";
    }
    if (x >= 320) {
        side = "left";
    }
    setPosition("ball", x, y, 32, 32);
}, 20);
```

A Scratch script using the `setInterval` block to move a ball. It includes the logic from the previous script, plus additional `if` blocks to check if the ball has hit the left or right walls (x <= 0 or x >= 320). If a wall is hit, the `side` variable is set to "right" or "left" respectively, and the ball's direction is reversed. A black arrow points from the `x` variable in the `if (x <= 0)` block to the `x` variable in the second `if` block, indicating where it should be placed.

Finally, use the above code to help you add a NEW ball to the screen. There should be two balls bouncing around when you are finished.

You will need to make a second set of variables:

```
var x2 = 200;  
var y2 = 4;  
var dir2 = "down";  
var side2 = "left";
```

Start the new ball in a different location and with a different starting direction.