



Girls Who Code At Home

Meteor Catcher Game: Part 4
Reference Guide

Meteor Catcher Game: Part 4 - Reference Guide



In this document you will find all of the answers to some of the questions in the activity. Follow along with the activity and when you see this icon, stop and check your ideas here.

Step 1: Add the Catcher

JAVASCRIPT

```
let meteorX = 200;
let meteorY = 0;
let meteorDiameter = 20;

let catcherDiameter = 40;

let speed = 0.5;

function setup() {
  createCanvas(400, 400);
}

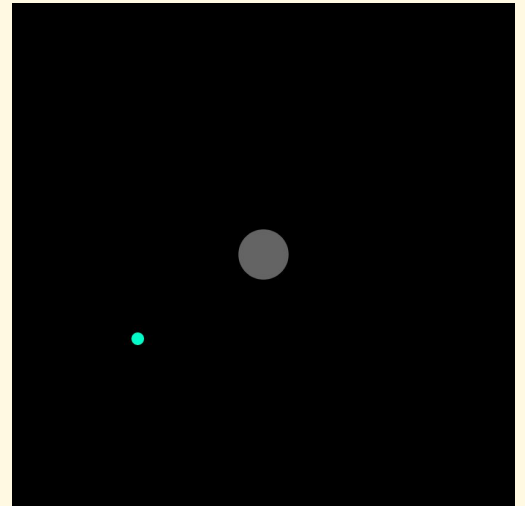
function draw() {
  background(0, 0, 0);
  noStroke();

  //Draw the meteor
  fill(0, 254, 202);
  ellipse(meteorX, meteorY, meteorDiameter,
    meteorDiameter);

  // Make the meteor fall
  meteorY = meteorY + speed;

  //Draw the catcher to follow the mouse
  fill(255, 255, 255, 100);
  ellipse(200, 200, catcherDiameter, catcherDiameter);
}
```

RESULT



Step 3: Test Your Code

JAVASCRIPT

```
let meteorX = 200;
let meteorY = 0;
let meteorDiameter = 20;

let catcherDiameter = 40; // Store diameter of catcher

let speed = 0.5;

function setup() {
  createCanvas(400, 400);
}

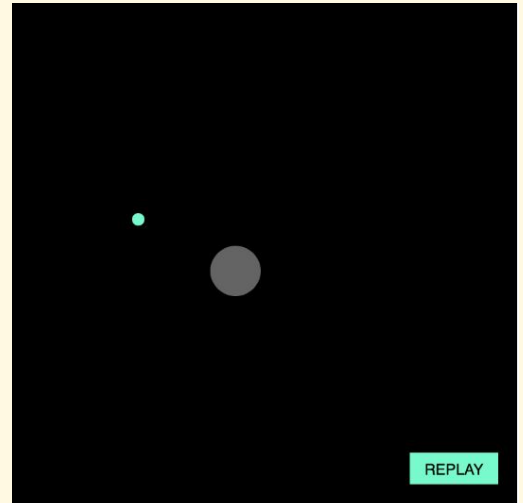
function draw() {
  background(0, 0, 0);
  noStroke();

  //Draw the meteor
  fill(0, 254, 202);
  ellipse(meteorX, meteorY, meteorDiameter,
    meteorDiameter);

  // Make the meteor fall
  meteorY = meteorY + speed;

  //Draw the catcher to follow the mouse
  fill(255, 255, 255, 100);
  ellipse(mouseX, mouseY, catcherDiameter,
    catcherDiameter);
}
```

RESULT



Note: In this [example sketch](#) we included a replay button so you can reset the meteor's behavior. If we did not include this, you would see only a black box once the meteor fell off the bottom of the screen. We will fix this in the next part with a conditional, but we're not there yet!

Step 4: Check for Understanding

Describe how this line of code would change the behavior of our catcher:

```
ellipse(200, 200, mouseX, mouseY);
```

Instead of changing the position of the ellipse, the width and height of the ellipse would change based on the mouse position. To see what this looks like in action, try changing the line of code. Just be sure to change it back to the original code with the right variable:

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

Step 6: Calculating Distance

JAVASCRIPT

```
let meteorX = 200; // Store the X position of the meteor
let meteorY = 0; // Store the Y position of the meteor

let meteorDiameter = 20; // Store diameter of the meteor
let catcherDiameter = 40; // Store diameter of catcher

let speed = 0.5; // Store speed of the meteor

function setup() {
  createCanvas(400, 400);
}

function draw() {
  background(0, 0, 0);
  noStroke();

  //Draw the meteor
  fill(0, 254, 202);
  ellipse(meteorX, meteorY, meteorDiameter, meteorDiameter);

  // Make the meteor fall
  meteorY = meteorY + speed;

  //Draw the catcher to follow the mouse
  fill(255, 255, 255, 100);
  ellipse(mouseX, mouseY, catcherDiameter, catcherDiameter);

  // Determine the distance between meteor and the catcher
  distance = dist(meteorX, meteorY, mouseX, mouseY);
}
```

Step 8: Set up catcher conditional

Plan the catcher conditional

There are different ways you can write this pseudocode. Here are a couple:

- If the **distance** value is less than 15 pixels, set the **meteorY** value to 0 at the top of the canvas.
- If the **distance** value is less than 15 pixels, reassign the variable for the y position of the meteor to 0.

Add catcher conditional

JAVASCRIPT

```
// Determine the distance between meteor and the catcher
distance = dist(meteorX, meteorY, mouseX, mouseY);

// Test to see if meteor and catcher have intersected
if (distance < 15) {
  // Redraw meteor to top of screen at a random location on x-axis
  meteorY = 0;
}
```

Step 9: Set up screen bottom conditional

There are different ways you can write this pseudocode. Here are a couple:

- If the **meteorY** value is greater than 400 pixels, set the **meteorY** value to 0.
- If the **meteorY** value is greater than the height of the canvas, assign the variable for the y position of the meteor to 0.

Step 10: Test Your Code

JAVASCRIPT

```
let meteorX = 200; // Store the X position of the meteor
let meteorY = 0; // Store the Y position of the meteor

let meteorDiameter = 20; // Store diameter of the meteor
let catcherDiameter = 40; // Store diameter of catcher

let speed = 0.5; // Store speed of the meteor
let distance; // Store distance between meteor and catcher

function setup() {
  createCanvas(400, 400);
}

function draw() {
  background(0, 0, 0);
  noStroke();

  // Draw the meteor
  fill(0, 254, 202);
  ellipse(meteorX, meteorY, meteorDiameter,
meteorDiameter);

  // Make the meteor fall
  meteorY = meteorY + speed;

  // Draw the catcher
  fill(255, 255, 255, 100);
  ellipse(mouseX, mouseY, catcherDiameter,
catcherDiameter);

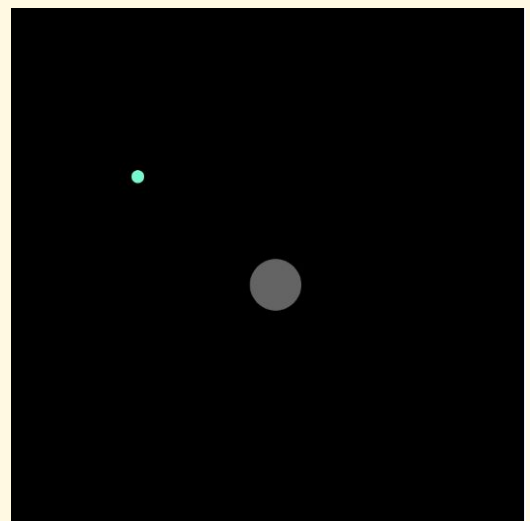
  // Determine the distance between meteor and the catcher
  distance = dist(meteorX, meteorY, mouseX, mouseY);

  // Print the value of distance
  print('Distance = ' + distance);

  // Test to see if meteor and catcher have intersected
  if (distance < 15) {
    // Redraw meteor to top of screen at a random location
    on x-axis
    meteorY = 0;
  }

  // Test to see if meteor has intersected with screen bottom
  if(meteorY > height) {
    meteorY = 0;
  }
}
```

RESULT



Click [here](#) to run the example sketch.

Step 11: Check for Understanding

Let's say you want to "catch" the meteor when the catcher barely touches the outer edge of the meteor. Would you increase or decrease the value in the expression of your first conditional statement?

You would *increase* the value. This would allow the statement to still evaluate as true with more distance between the center of the catcher and center of the meteor.