



# Girls Who Code At Home

流星キャッチャーゲーム: Part 4

リファレンスガイド

## 流星キャッチャーゲーム：パート4 – リファレンスガイド



このドキュメントには、アクティビティで出題されるいくつかの質問に対するすべての回答が掲載されています。アクティビティに沿って進み、このアイコンを見たら、立ち止まってここで自分の考えを確認してください。

### ステップ 1: キャッチャーを追加する

#### 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();

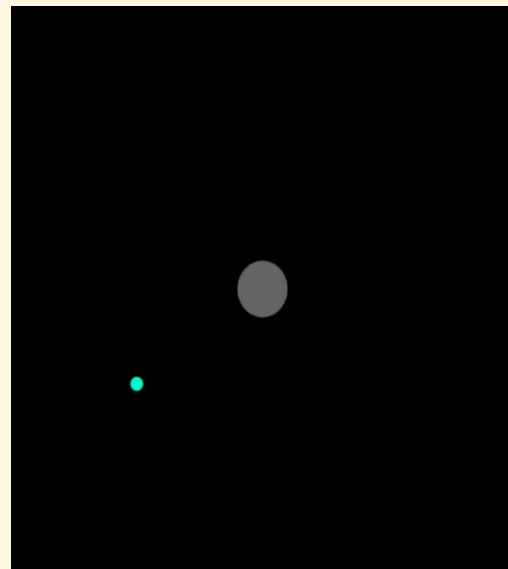
  //流星を描画する

  fill(0,254,202);
  ellipse(meteorX, meteorY, meteorDiameter,
    meteorDiameter);

  // 流星を落下させる
  meteorY = meteorY + speed;

  //マウスに追従するようにキャッチャーを描画する
  fill(255, 255, 255, 100);
  ellipse(200, 200, catcherDiameter, catcherDiameter);
}
```

#### RESULT



## ステップ3：コードのテスト

### JAVASCRIPT

```
let meteorX = 200;
let meteorY = 0;
let meteorDiameter = 20;
let catcherDiameter = 40; // キャッチャーの直径を格納
let speed = 0.5;
function setup() {
  createCanvas(400,400);
}

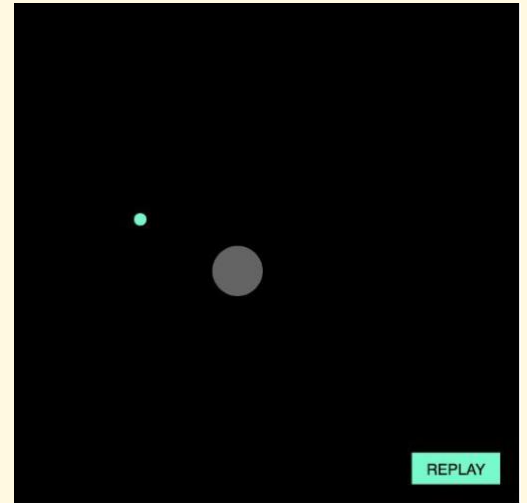
function draw() {
  background(0,0,0);
  noStroke();

  //流星を描画
  fill(0,254,202);
  ellipse(meteorX, meteorY, meteorDiameter,
    meteorDiameter);

  // 流星を落下させる
  meteorY = meteorY + speed;

  ///マウスに追従するようにキャッチャーを描画する fill(255,
  255, 255, 100);
  ellipse(mouseX, mouseY, catcherDiameter,
    catcherDiameter);
}
```

### RESULT



**Note:** 注： [このスケッチ](#)の例では、流星の動作をリセットできるようにリプレイボタンを付けました。もし、これを入れなければ、流星が画面の下から落ちると、黒いボックスだけが表示されることになります。次のパートでは、これを条件付きで修正しますが、今回はまだです

## ステップ4：理解度の確認

このコードによって、キャッチャーの動作がどのように変わるかを説明します。：

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

楕円（ellipse）の位置を変える代わりに、楕円（ellipse）の幅と高さをマウスの位置に応じて変更するようにしました。実際にどのように見えるかを見るには、コードの行を変更してみてください。ただし、変数を正しく指定して、元のコードに戻してください。：

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

## ステップ6：距離の計算

### JAVASCRIPT

```
let meteorX = 200; // 流星の X 位置を格納
let meteorY = 0; // 流星の Y 位置を格納

let meteorDiameter = 20; // 流星の直径を格納
let catcherDiameter = 40; // キャッチャーの直径を格納
let speed = 0.5; // 流星の速度を格納する

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

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

  //流星を描画する fill(0,254, 202);
  ellipse(meteorX, meteorY, meteorDiameter, meteorDiameter);

  // 流星を落下させる meteorY = meteorY + speed;

  //マウスに追従するようにキャッチャーを描画する
  fill(255, 255, 100);
  ellipse(mouseX, mouseY, catcherDiameter, catcherDiameter);

  // 流星とキャッチャーの距離の決定
  distance = dist(meteorX, meteorY, mouseX, mouseY);
}
```

## ステップ8：キャッチャーコンディションの設定

### キャッチャーコンディションを計画する

この疑似コードの書き方は様々です。ここではそのいくつかを紹介します。

- **distance** の値が 15 ピクセル以下の場合、キャンバスの上部にある **meteorY** の値を 0 に設定します。
- **distance** の値が 15 ピクセル以下の場合、流星の y 位置を表す変数を 0 に再設定する。

### キャッチャーの条件を追加しよう

#### JAVASCRIPT

```
// 流星とキャッチャーの距離を決定する
distance = dist(meteorX, meteorY, mouseX, mouseY);

// 流星とキャッチャーが交差したかどうかをテストする
if (distance < 15) {
    // Redraw meteor to top of screen at a random location on x-axis meteorY = 0;
}
```

## ステップ9：画面下部の条件を設定する

この疑似コードの書き方は様々です。ここではそのいくつかを紹介します：

- **meteorY** の値が 400 ピクセルより大きい場合、0 を設定する。
- **meteorY** の値がキャンバスの高さより大きい場合、meteor の y 位置を表す変数に 0 を代入します。

## ステップ10:コードのテスト

### JAVASCRIPT

```
let meteorX = 200; // 流星の X 位置を格納する。
let meteorY = 0; // 流星の Y 位置を格納する。

let meteorDiameter = 20; // 流星の直径を格納する。
let catcherDiameter = 40; // キャッチャーの直径を格納する

let speed = 0.5; // 流星の速度を格納する。
let distance; // 流星とキャッチャーの間の距離を格納する。

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

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

  // 流星を描画する
  fill(0, 254, 202);
  ellipse(meteorX, meteorY, meteorDiameter, meteorDiameter);

  // 流星を落下させる
  meteorY = meteorY + speed;

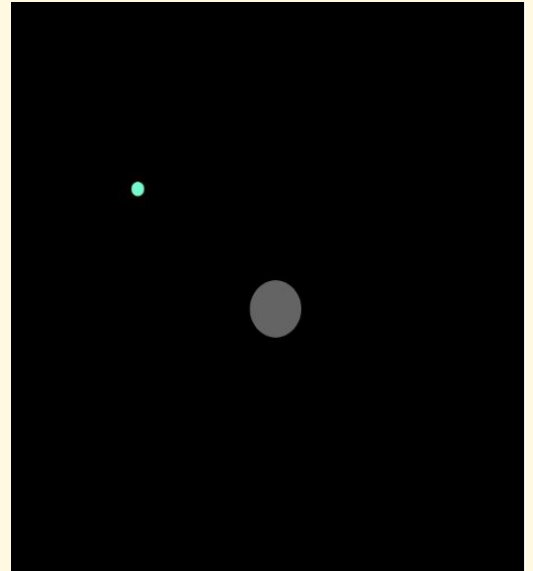
  // キャッチャーを描画する
  fill(255, 255, 255, 100);
  ellipse(mouseX, mouseY, catcherDiameter, catcherDiameter);

  // 流星とキャッチャーの距離の決定
  distance = dist(meteorX, meteorY, mouseX, mouseY);

  // 距離の値を表示する
  print('Distance = ' + distance);
  // 流星とキャッチャーが交差したかどうかのテスト
  if (距離 < 15) {
    // 流星を画面上部の x 軸上のランダムな位置に再描画する。
    meteorY = 0;
  }

  // 流星と画面下部が交差しているかどうかのテスト
  if(meteorY > height) {
    meteorY = 0;
  }
}
```

### RESULT



[ここ](#) をクリックして実行。

## ステップ11：理解度の確認

例えば、流星の外縁にかろうじてキャッチャーが触れたときに流星を「キャッチ」したいとしましょう。このとき、最初の条件文の式中の値を増やしますか？減らしますか？

このときは、値を増やします。そうすれば、キャッチャーの中心と流星の中心の距離が離れていても、この文は真と評価されるようになります。