

Movement

By Crystal Hess

<http://csed.fun>

Learning Goals

- Practice reading and then modifying code that you didn't write
- Use variables to store and manipulate an object's position
- Use conditionals to control when actions can and cannot occur

```

1 var monster_img;
2 var cookie_img;
3 var points;
4 var monster_x, monster_y;
5 var cookie_x, cookie_y;
6
7 function preload() {
8     monster_img = loadImage("assets/cookie_monster.png");
9     cookie_img = loadImage("assets/cookie.png");
10 }
11
12 function setup() {
13     createCanvas(720, 400);
14     monster_x = 150
15     monster_y = height-150;
16     cookie_x = 725;
17     cookie_y = random(350);
18     points = 0;
19 }
20
21 function draw() {
22     background(200);
23     displayPoints();
24
25     image(monster_img, monster_x, monster_y);
26     image(cookie_img, cookie_x, cookie_y);
27
28     moveCookie();
29     moveMonster();
30     checkForChomp();
31 }

```

Don't run the code yet! Just read it!

A look at the starter code

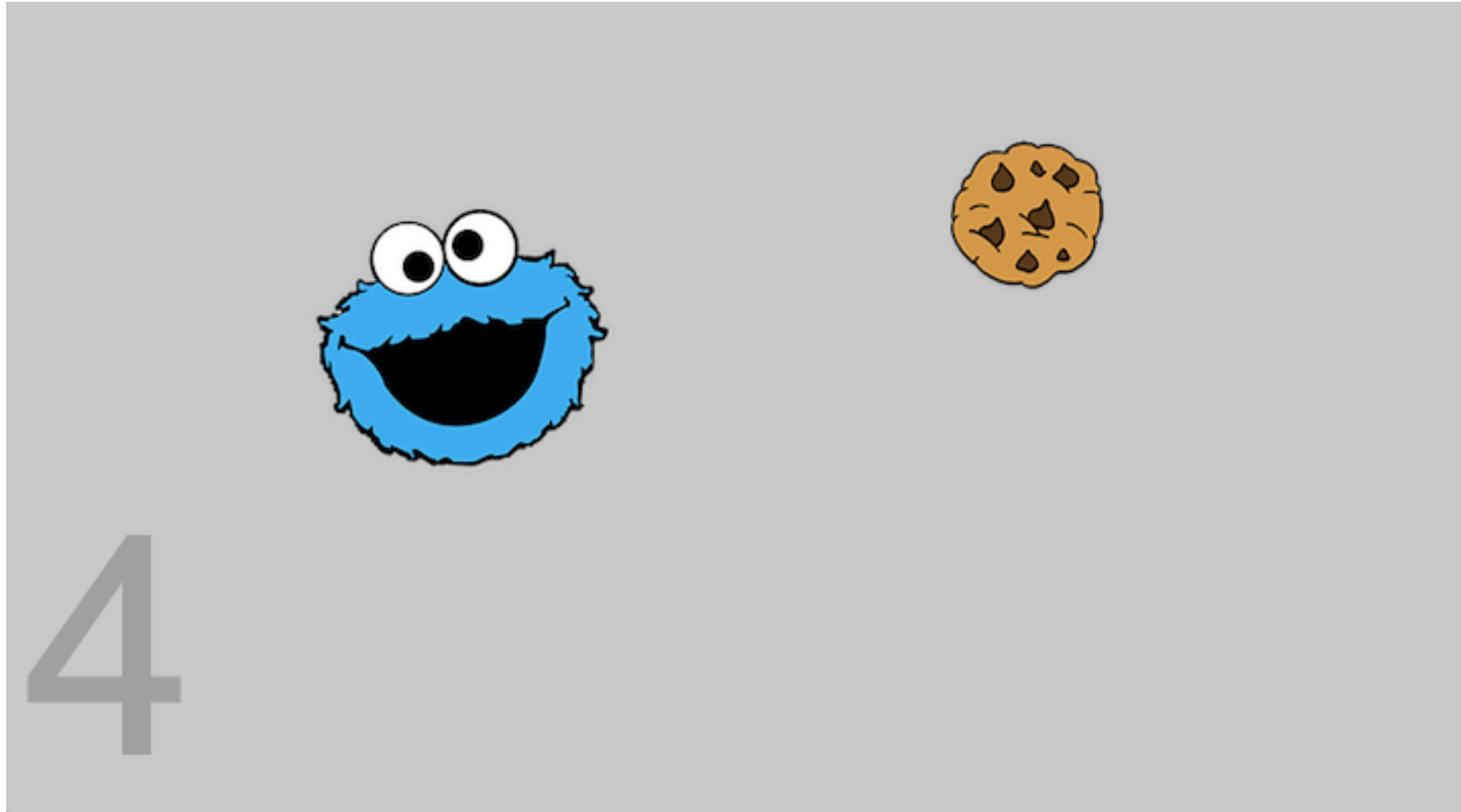
1. Lines 1-5: do you know what each of these variables will represent?
2. Lines 7-10: where can you find these images in your documents?
3. Lines 12-18: draw a loose sketch of that the setup of the canvas will look like.
4. Lines 21-31: explain in your own words what happens during each iteration of the **draw()** loop.

A look at the starter code

5. Lines 33-36: where do you expect to see the points displayed on the screen?
6. Lines 39-46: in which direction will the cookie move?
7. What happens when/if the cookie moves off screen? How does the program know the cookie has moved off screen?
8. Lines 48-53: How and when does the monster move?
9. Lines 55-62: When does a chomp occur? What happens when there's a chomp?

```
32
33 function displayPoints() {
34     fill(160);
35     textSize(150);
36     text(points,10,370);
37 }
38
39 function moveCookie() {
40     if(cookie_x < 0) {
41         cookie_x = 725;
42         cookie_y = random(350);
43     }
44     else
45         cookie_x -= 4;
46 }
47
48 function moveMonster() {
49     if(keyIsDown(UP_ARROW) && monster_y > 0)
50         monster_y -= 2;
51     if(keyIsDown(DOWN_ARROW) && monster_y < height-150)
52         monster_y += 2;
53 }
54
55 function checkForChomp() {
56     var d = dist(cookie_x, cookie_y, monster_x, monster_y);
57     if (d < 100) {
58         points += 1;
59         cookie_x = 725;
60         cookie_y = random(350);
61     }
62 }
```

Ok, now run the code...



Features to add (with a pair)

- Track the number of cookies that are missed and end the game after 3 misses
 1. Decide: how do you want to track misses? With a number or with a string?
 2. Decide: do you want to display the number of misses on the screen? How?
 3. Decide: how are you going to end the game? Place a big message blocking the screen?

Added Fun

- Decide: do you want the game to be able to be restarted? How would this work?
- Add a piece of pie and a piece of cake that randomly appear and move on the screen to be eaten. Make it so that eating the pie or cake count as a miss.
 1. Decide: what do you want the images to look like? Where do you need to save them?
 2. Decide: how fast do you want the pie and cake move?
 3. Decide: do you want it to be possible for all three items to appear at the same time? Or do you want them to appear one at a time?

Added Fun

- Decide: how can you prevent the food items from overlapping each other?

Fun Extensions

- Give Cookie Monster the ability to move in any direction, but not off the screen
- Make the cookies speed up every time you catch one

...Come up with your own!