04_Mouse_Movement_Lab_Lesson_Plan

Description

In this lesson, students will learn additional ways they can use their mouse to cause interactions within their p5 play programs. Students will learn how to make a collision through a code along and then will read through some simplified documentation with a partner.

Objectives

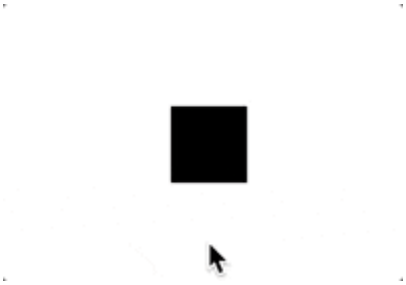
- I can use mouse movement to change aspects of my program
- I can control my sprite with mouse movement.

Standards

- **7-8.CT.7 Algorithms && Programming:** Design or remix a program that uses a variable to maintain the current value of a key piece of information.

Brain-Starter + Hook (10 min)

Prompt:
Look at the gif below. Turn and talk with your partner and describe what is happening in the gif.



Mini-Discussion: have students share what their partner saw or said.

Code Along (7-10 min)

Say: “We have already seen a few ways that we can control sprites using our mouse (like with `.collides`). Today we are going to examine other ways your mouse can interact with sprites. Let’s try it together! Open up the starter code.”

Please note that the starter code is an empty editor with p5 play linked.

```
var sprite1

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

```
  colorMode(HSB)

  sprite1 = new Sprite(200,200,50,80) //new square sprite in the middle of the canvas

}

function draw() {
  background(220);

 //Let's make a square sprite in the middle of the canvas.

  //When the mouse is hovering over the sprite, let's get it's color to change.

  if (sprite1.mouse.hovering()) { //checks IF the mouse is currently hovering over the
sprite
    console.log("Mouse hovering over sprite1")

    sprite1.color = color(300,50,100) //changes the color to pink

    sprite1.width = 100 //changes the width of the sprite to 100
  }
  else { // all other situations -- not hovering.
    sprite1.color = color(100,50,100) //changes the color back to green

    sprite1.width = 50 //changes the width of the sprite back to 50
  }


  //Let's also get it's width to increase (still when the mouse is hovering over the
sprite)



}
```

Questions to ask during code along:

1. Where do we make variables?
2. Where do we initialize variables?
3. What is the difference between the draw function and the setup function?
4. What do you think the difference is between “hovering”, “hovered”, and “hovers”?

Pause frequently during code along to take questions and comments.

Lab Exploration (20-25 min)

Regroup students into pairs. Students will be working together to investigate how to use different mouse movement interactions.

Create a document (ideally a google document or PDF) with below information. Please note that there are some code hints hidden in white writing. Give each pair of students access to the Mouse Movement Documentation, students must use it to complete the Mouse Movement Lab.

Mouse Movement Documentation

Hover	Hover means your mouse is on top of a sprite but it is not clicking the sprite. You can have `.hovered()`, `.hovering()`, `.hovers()`
Press	Press means your mouse is on top of a sprite AND it is clicking the sprite. You can have `.pressed()`, `.pressing()`, `.presses()`
Drag	Drag means your mouse is on top of a sprite and clicking and dragging it around your canvas. You can have `.dragged()`, `.dragging()`, `.drags()`
Release	Release means your mouse was clicking the sprite but has stopped. You can have `.releasing()`, `.released()`

Lab Instructions for students

```
function setup() {
```

```
  createCanvas(400, 400);


}

function draw() {
  background(220);

////////////////  MILD  ////////////////

  //1. Create a sprite called sprite1 and make its height change when your mouse is
hovering over it.



  //2. Create a sprite called sprite2 and make its color change when you click it.



  //3. Make a sprite called sprite3, change its location by clicking and dragging it to
a new location on the canvas.



  ////////////////  SPICY  ////////////////

  //Create a sprite called sprite4, make the sprite get larger AFTER it has been
clicked.


}
```

Students should submit their code at the end of class to help identify comprehension and identify common mistakes to inform the next lesson.

Debrief (7 min)

Say: “We have now learned so many different ways you can engage with your sprites using the mouse. Lets see what we learned.”

Ask the students to answer the following question as an exit slip.

- What is the difference between “hover” and “press”?

Have students share out at their tables and then ask for popcorn responses.