Ms. Boland
Software Engineering

Animation in p5

Key Objectives:

- ❏ I can create an animation program that runs without error in order to demonstrate my knowledge of coding.
- ❏ My program includes an algorithm that uses sequencing, selection, and loops in order to achieve object motion/animation.
- ❏ With the use of comments, I am able to identify/highlight where sequencing, selection, and loops are necessary in my program.

ESSENTIAL QUESTION:

How can code be used as a creative and expressive medium?

SKILLS

ellipse()
rect()
line()
point()
triangle()
quad()
arc()
beginShape()
fill()
noFill()
stroke()
noStroke()
strokeWeight()
mousePressed()

Sequencing algorithms

Conditionals

DO NOW

- Click on web editor link (provided in google classroom & Zoom chat)
- Log into "p5 editor account" (Create an account if you haven't already)
- Create a circle centered on your canvas

CODE ALONG: During this exploration, you will discover how to move objects in p5.

Independent Work

Part 1: Animation in p5

Task: You will use motion via coding to create a unique animated screensaver. Your program will be free of error, and include clear comments that highlight key functions and sequencing necessary for object motion.

Key Attributes:

- Use motion via coding to create a unique animated object.
- Add comments to different functions used in order to identify/explain the sequencing of your code.
- Use fill and stroke to change the color of your object in order to enhance/create depth in the detail of your animation.
- Move your object across/around the canvas as a means of demonstrating my knowledge of motion in p5, and your understanding of the coordinate plane in p5.
- Objects in screensaver move vertically, horizontally, diagonally, bounces; include varied start/stop points with repetitive motion; and, include more than one shape (at least 5).

Ms. Boland
Software Engineering

Part 2: Reflection Questions
Directions: Create a google doc and answer the following questions using complete sentences.

1. How can you make images move in p5? Give an example.
2. Describe the process you used to create your unique animation
3. Why is it important to pay attention to the order/sequencing of code in p5?.
4. Why are comments an important part of the coding process?

[p5 reference sheet](#) | *Official documentation of the p5 library.*	[p5 web editor](#) | *Online IDE for p5 projects - it comes with the p5 library linked in the HTML and will give p5 specific errors.*

Animation

```
//display coordinates of the mouse at coordinate (20,20)
  text(mouseX + ", " + mouseY, 20, 20)
```

This is a helpful chunk of code that can help you find mouse coordinates when drawing shapes. Use it!

In p5, each time the **draw()** loop runs, it **draws an image**. Our sketches are like digital flip books: each image rendered by draw() is like a new page in the book.

When p5 runs, it calls draw() again and again, drawing image after image, as if it were flipping the pages of a book. Each 'page' p5 draws is called a **frame**.

To create smooth motion, p5.js tries to run the code inside draw() at 60 frames each second. This is what is called the **frame rate**.

When you want a shape to be animated, you have to look at what is changing about the shape itself.

```
var x; // Start by creating a
```

HORIZONTAL MOTION:

To make the ball move, we need to change the x position of the ball every time we draw *until* it reaches the right edge.

```
variable called x, this value needs
to change every frame

function setup() {
  createCanvas(600, 120);
  background(0);
  x = 0; // Start at the left edge
of the canvas

function draw() {
  x = x + 1; // Move slightly to the
right every frame by changing the
value of x by 1

 ellipse(x, 60, 20, 20);
}
```

```
var x;
var y;

function setup() {
  createCanvas(600, 600);
  x = 0;
  y = 0;
}

function draw() {
  background(0);
  x = x + 1;
  y = y + 1;
  ellipse(x, y, 20, 20);
}
```

DIAGONAL MOTION:

If we include both the x and y variable, we can make our ball move in both directions at the same time, resulting in diagonal motion.

Think back to learning about slope in your Algebra courses!

```
x = x + random(-1, 1);
y = y + random(-1, 1);
```

```
var min = -3;
var max = 5;
```

MOVE AT RANDOM:

In our sketches, to move our ball along a line, we would add a given number to its x and y position, on each turn of the draw loop.

THE random() FUNCTION:

We can assign random “x” and “y” values to the position of

```
function draw(){

var result = random(min, max);

//output random values between -3 and 5
console.log(result);
}
```

our shapes by using the **random()** function.

This function can accept two arguments: a **minimum** and a **maximum.**

If we supply one argument, the **random()** function has a **default minimum of 0.**

```
x = x + 0.2; or x = x + 15;
```

MOVING FASTER OR SLOWER

Until now, we have been adding 1 pixel to the position of the ball each time the draw loop is called. Try using other numbers instead like

←

```
x = x + speed;
```

As you can see, the number we add to x in each frame determines the **"speed"** of the ball. So let's replace the "1" with a variable called speed so that x = x + 1; becomes

←

x = x+1	**can be written as**	x += 1
x = x-1		x -= 1
x = x*1		x *= 1
x = x/1		x /= 1

ASSIGNMENT OPERATORS

Because expressions like x = x + 1 are so common, there is a shorthand for them: x += 1

This is a common shorthand, you will most likely encounter it in the future, and it also works for other operations.

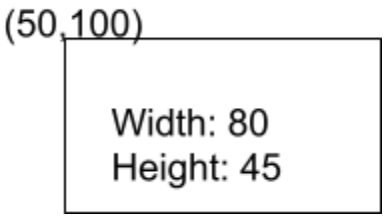
Collision Logic && mouseIsPressed

<table>
<tr>
<td>

```
if(mouseIsPressed){
   //code that happens on a click
}
```

</td>
<td>The mouseIsPressed variable returns a true or false value based on if the mouse is being clicked anywhere on the canvas. It can be used as a part of a conditional.</td>
</tr>
<tr>
<td>

MOUSE ON RECTANGLE?



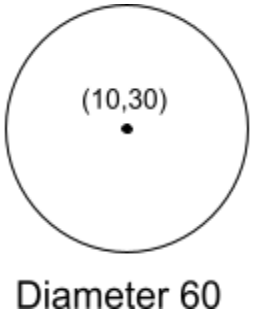

```
if(mouseX > 50 && mouseX < 50 + 80 &&
mouseY > 100 && mouseY < 100 + 45){
   console.log("Mouse on rectangle.")
}
```

</td>
<td>To figure out if a mouse is on the rectangle, you need to check if:

The mouse's x position is bigger than the leftmost edge of the rectangle (<code>mouseX > xPosition</code>)
The mouses' x position is less than the rightmost edge of the rectangle (<code>mouseX < xPosition + width</code>)
The mouse's y position is bigger than the topmost edge of the rectangle (<code>mouseY > yPosition</code>)
The mouse's y position is less than the bottommost edge of the rectangle (<code>mouseY < yPosition + height</code>)

</td>
</tr>
<tr>
<td>

MOUSE ON ELLIPSE?




```
var distanceFromCenter =
dist(mouseX,mouseY,10,30)
```

</td>
<td>To determine if the mouse is on the ellipse, you need to:

Calculate how far the mouse is from the center of the circle (<code>dist(mouseX,mouseY,circX,circY)</code>)
Determine if the mouse is less than a radius distance away from the center by dividing the diameter by 2.

</td>
</tr>
</table>

`if(distanceFromCenter<60/2){`
`  console.log("Mouse on circle.")`
`}`
	(`distanceFromCenter<60/2`)
	• If it is, then the mouse is on the circle!

Link to Collide2d Library Reference: https://github.com/bmoren/p5.collide2D

What is Collide2d?

Collide is a pre-built library for p5 that has a ton of custom functions figured out by other programmers. These functions all test to see if two things are *colliding*, or touching!

You can check the library for what options are available. All of them look at two separate shapes and, when the function is called, will return true if the objects are touching and false if they are not. This makes them perfect to use in **conditional statements** that require a true/false statement!

MOUSE PRESSED

```
function setup(){
   createCanvas(400,400)
   //things for start of program
}

function draw(){
   background(220)
   //things that repeat in program
}

function mousePressed(){
   //whatever happens when the mouse is
pressed!
}
```

This is a major callback function like setup and draw. It runs separately from each and runs just one time whenever the mouse is clicked on the canvas.

← It would look like this.

Links and Resources

P5.js random() reference	https://p5js.org/reference/#/p5/random
Coding Train: Let vs Var	https://thecodingtrain.com/Tutorials/16-javascript-es6/16.1-let-vs-var.html
JavaScript if else and else if	https://www.w3schools.com/js/js_if_else.asp

Ms. Boland
Software Engineering

Good to Know! *There is SO MUCh going on in p5, and we don't have time to explicitly cover all of it during our remote PD. The below topics may be useful as you complete your remote tasks for today, but are **not required** - so feel free to skip this part if you feel good just working with what you know.*		
Collide && Click Youtube Video Link *Learn about using **collide2D** && **mouse clicks***	**Conditionals on the Canvas** Youtube Video Link *Learn about using **if/else if/else***	**Intro to Animation** Youtube Video Link *Learn about animating your shapes on the screen.*
Intro to Collide2D with Ellipse Youtube Video Link *Learn about using the **collide2D** library with an ellipse.*	**Collide2D with Other Shapes** Youtube Video Link *Learn about using **collide2D** && other shapes.*	**Making things bounce** Youtube Video Link *Learn how to make your shapes bounce all over the screen*

NOTES

Animation in p5	