VIZUALIZATION TECHNOLOGIES
CALSS 4 OCTOBER 9 2018

MOUSE

mouseX, mouseY -> current mouse coordinates
pmouseX, pmouseY -> mouse coordinates in the previous draw() call

Assignment in class 1
Use line(mouseX, mouseY, pmouseX, pmouseY) to continuously draw the movements of your mouse as a continuous line.

mouseIsPressed -> true or false
mouseButton, to only be checked when mouseIsPressed is true.
mouseButton can take the values LEFT, RIGHT, and CENTER.

Assignment in class 2
Make a circle that changes size when the mouse is pressed and resets to its size when the mouse is not pressed. Change the color of the circle depending if you click with the right mouse button, or the left mouse button.

Assignment in class 3
Use the dist() function to test if the mouse is inside a circle that oscillates in size. Use the dist() function to get the distance from the center of the circle to the cursor, then test to see if that distance is less than the radius of the circle.

LETTERS

Assignment in class 4

Draw letters on the canvas that you pressed by copying the following example.

```
function setup() {
	createCanvas(400, 400);
	textSize(64);
	textAlign(CENTER);
	fill(255);
}

function draw() {
	background(0);
	text(key, 60, 80);
}
```

The boolean variable keyIsPressed is true if any key is pressed and false if no keys are pressed.

You can test for a certain key by using the key variable. You can test for characters including punctuation:

```
	if(key == ‘a’) …
	if(key == ‘,’) …
```

If the pressed key is not an ASCII character, you have to use keyCode to test for it

```
	if(keyCode == BACKSPACE) …
	if(keyCode == RIGHT_ARROW) …
	if(keyCode == ENTER) …
```

Assignment in class 5

Implement a simple typewriter. As you type a key, it should appear on screen. You will need to use the following functions:

loop() -> activates the draw()

noLoop() -> pauses the draw()

textWidth() -> function that you will need to use in order to advance the type cursor.

keyTyped() -> a function that is asynchronously called when a key is typed: it ignores keys such as CTRL, SHIFT, ALT, etc.

keyPressed() -> a function that is asynchronously called when a key is typed. You will use this function to implement change of line.

STOCHASTIC FUNCTIONS AND PERLIN NOISE

```
random()
randomSeed(seed)
noise(x)
noise(x, y)
noise(x, y, z)
noiseSeed(seed)
```

Assignment in class 5

Create six charts on the same canvas that show the noise function and the random function. The first three are for random numbers for three different random seeds. For each horizontal pixel on the canvas, show a different random number. Use the map() function to have the [0, 1] values between [0, height]. Use another color for the other three charts, which should show noise values for certain intervals along the x. These intervals should increase as you move the mouse along the x. Use the map() function to obtain values that show a visual variance in for the noise chart. Normalize the noise chart to [0, height] by using the map() function as well.

Assignment in class 6

Create a grid of circles that change randomly in size. After this, make the size of the circles vary according with the noise function. The variation should happen in the size, but it is also dependent on the x and y positions of the circle. Therefore, you will have to use the noise(x, y, z). Try to obtain a similar effect to the one demonstrated in noise.mp4.

ARRAYS AND VECTORS

```
var array = [2002, 2003, 2004];
```

or

```
var array = [];
array[0] = 2002;
array[1] = 2003;
array[2] = 2004;
```

The content of array[2] is 2004.
The content of array[3] is undefined.
array.length is a property of any array and in this case is 3. You can use this property to add elements to the array or to iterate through its elements.

Vectors is an object with a direction and a magnitude. They can be defined in n-dimensions, but p5.js takes only 2 or 3. You can learn more on the operations that you can perform on vectors here: https://p5js.org/reference/#/p5.Vector.

ASSIGNMENT 4

Modify the vectors.js file as to use the vector.rotate() function instead of vector.set(). Replicate the effects on vectors.js for any number n of particles at the same time by using arrays.

READINGS FOR NEXT CLASS

Read Chapter 11 on arrays of “Make: Getting Started with p5.js”
Read Chapter 6 on arrays of “JavaScript: The Good Parts” by Douglas Crockford.