Hashemi_Yashar_LabReport2 9/20/17

Title: A Program for a moving object using Perlin Noise

Author: Yashar Hashemi

Background/Introduction:

In this lab I am designing a cancer cell that displays on screen and moves using Perlin Noise 60% of the time and with my own movement algorithm 40% of the time. Additionally, when the mouse is pressed the cancer cell will move with varied step sizes. The purpose of this lab is to learn about Perlin noise, movement algorithms, creating objects in Processing and the usefulness of randomness in programming.

Methods:

- Create a Cancer class
- Create two vectors: one to keep track of the location, the other for the x and y time parameters for Perlin Noise
- I created a Cancer() constructor and initialized the Cancer() object at the starting location of the Perlin Noise function and a Perlin noise x time of 0 and a y time of 10000. This is to prevent the x and y value generated from Perlin noise from being the same.
- create a noisestep(tx, ty) function which takes an x and y time parameter for Perlin Noise and maps the output of Perlin Noise to the height and width of the window.
- I created a step() function that moves the cancer cell using Perlin Noise 60% of the time and using my own algorithm(location.y+=9) 40% of the time by means of the random() function in the Processing library:
  float n = random(0,1);
  if (n<.6) {Perlin Noise} else { location.y +=9}
- Additionally, the step() function changes when the mouse is pressed:
  if (mousePressed == true){
- When the mouse is pressed the cell moves randomly between three different time intervals of Perlin Noise:
  int r = random(0,3);
  if (r ==0){
  noisestep(.05, .05);
  if (r ==1) { etc…

- The Cancer cell is displayed using a display() function which creates a small red ellipse and a large white ellipse at the x and y location of the location vector .
- In the setup() function, I created a window with a width and a height of 1000, initialized my cancer cell and set the background color to black.
- In the draw() function I called the step() and display() functions from the Cancer class.

Results:

I have successfully written and implemented a Cancer class that moves using various speeds of Perlin noise as well as my own algorithm. The Cancer object moves using Perlin noise 60% of the time and moves with my own algorithm 40% of the time. When the mouse is pressed, the object moves at three various step sizes of Perlin noise. The object is displayed in a window as a red circle inside of a white circle. In the Figure 1 I show the differences of movement when the mouse is pressed vs. when it is not. The image on the left is when the mouse is not pressed and the image on the right is when the mouse is pressed

Figure 1: As you can see, when the mouse is pressed, the Cancer object moves at a fairly consistent speed, but when the mouse is pressed, the speed becomes more erratic.

Conclusion: I have created a Cancer class that moves using various techniques including Perlin Noise.

Credit/Acknowledgements: I used resources and code examples from the Processing tutorial on Objects and book: Nature of Code, Introduction and Chapter 1, in constructing this program/sketch.

Citations:

Shiffman, Daniel, Shannon Fry, and Zannah Marsh. The Nature of Code. D. Shiffman, 2012.

Objects, Processing Tutorial, Daniel Shiffman. https://processing.org/tutorials/objects/