Elliot Bidwell

CS 2300 – 001

Programming Assignment #2

Pseudocode:

Main function:

* Call the input taking function, then the board generator
* Create a list to hold a record of all previous lines
* Repeatedly call the turn taking function in a for loop and iterate a number of times equal to the 1st dimension of the input array
* Call the score calculation function and print its return value, as well as the game board array to a file

Game board generator function:

* Take the first element of the array returned by the file input function
* Create and a 2D array with the dimensions specified by the argument and populate it with -’s
* return the array

Score calculating function:

* Take in the array returned by the board generator function
* Initialize scores at 0
* Nested for loop through the array and search for instances of ‘x’, iterate player 1 score by 1 with each instance
* Repeat process with player 2’s score
* Return a tuple of the two scores

File input function:

* Create an input file object
* Use a nested for loop to iterate through each element of each line and populate a 2D array
* Return the array

Turn taking function:

* Take 4 integers as arguments, 1 for each component of the 2 points, as well as list containing all the previous lines played, and an integer k representing how many lines to compare to the new one
* Calculate a vector by subtracting the second point components from the first
* Store the points and the vector in an array to be referenced later
* Loop through the list of previous lines backwards k times, each time comparing the new line to test for shared points or perpendicularity
* Update the list to include the new line
* If the line passes the comparison tests:
  + Calculate the foot of every point on the board with respect to the line and place each one in a list
  + Calculate the distance between every point and their respective feet and store these in another list in the same order
  + Bubble sort the list of distances while simultaneously putting the list of points in the same order in order to maintain their correspondence
  + Choose the least distant n points and color them, where n is the length/width of the game board
* Return the list of previous lines

Final Thoughts:

Like with many of the programs I write, this one ended up being vastly different from what I had planned through my pseudocode. While lists in python do have many capabilities, and I did end up using many of them, they don’t work well when it comes to representing the parametric equation of a line. Because of this I decided to create a line class which would store every important element of a line as well as perform operations on itself. It also became apparent that I should create classes for the line record, and the game board. These two classes exist for the same reason as the line class: the use of lists is necessary but still not enough. I learned all this very quickly, as I taught myself python specifically for this project and noticed that for the first few hours, I was treating it like I was writing in C, which is a terrible language for working with matrices in my opinion. After some practice and many online tutorials and documentation, I got the program to work. I have included a list of resources that I used at the end of this document. There was also one problem that had to do with the assignment description. I was unclear as to whether I needed to compare each line to the previous k lines for perpendicularity or shared points or both. My reading of the assignment description document gave me the impression that I needed to compare with the last k lines for shared points, while I needed to compare with all previous lines for perpendicularity. However, the video posted mentioned that I needed to compare the last k lines for perpendicularity. The problem could very well have been my own misinterpretation, and my program is easily adaptable for any of these cases, so it really was a very small issue.

Resources:

<https://youtu.be/_uQrJ0TkZlc>

<https://youtu.be/kqtD5dpn9C8>

<https://www.geeksforgeeks.org/bresenhams-line-generation-algorithm/><https://docs.python.org/3/contents.html><https://stackoverflow.com/questions/6583573/how-to-read-numbers-from-file-in-python><https://www.w3schools.com/python/default.asp>