Project 2: Due 10/14 by 11:59 PM

# **Introduction:**

For this part of the project we will be looking at pathfinding. You will be updating the environment created in the last project by adding additional pathfinding code in order to cause the program to demonstrate some of the topics that we have covered within the class.

# **Requirements:**

1. For this project you will be dealing with multiple units. Just as in the last project, the units can be cubes or other types of geometric figures. There is no need to use models or anything else visually dramatic however if you wish to do so that’s really up to you.
2. You be implementing the basic A\*algorithm. Use the A\*algorithm with a single unit in order to demonstrate that the algorithm is working correctly. The unit that is running the A\*algorithm should be colored differently from the other units in the demonstration.
3. You will also be implementing a flocking algorithm. This algorithm will work on a group of units, those units should be colored differently from the unit that’s running the A\*algorithm. Allow the use of keyboard commands in order to change the parameters for the flocking algorithm. This could be done similarly to the way the authors have implemented their demonstration code.
4. Try testing your flocking algorithm out going through the bottlenecks that been built into the environment in project one and see what the effectiveness of the algorithm is. If there are issues with the flocking algorithm dealing with the bottlenecks see if you can come up with a way of correcting them.

# **Deliverables:**

There will be two deliverables for this project:

1. The completed project will be delivered as a single zip file.
2. Documentation will be included with the assignment as a separate Word format following the grading rubric that’s given down below for the document. This document will describe how to start the environment, how to run the A\*demonstration, how to run the flocking demonstration, and how to alter the flocking demonstration by changing parameters using keyboard controls. In the event there were issues with the bottlenecks in the environment, your document should explain what the issues were and how you resolve them.

# **Grading Rubric:**

Team # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Names: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| Criteria | Max Pts. | Earned Pts. |
| Multiple units are present in the demo | 10 |  |
| One character uses the A\* algorithm to determine a path. (Provide instructions on how to set the character’s destination.) | 25 |  |
| Use flocking with a group of characters. Use keyboard commands to change the parameters for the flocking. Your code should have a way to demonstrate the flocking – either keyboard controls or a fixed path they follow | 30 |  |
| Your flocking group should go through the bottleneck built into the environment. If it fails, try to correct the issue. | 15 |  |
| Documentation:   * Describe how to start the environment * Describe how to run the A\*demonstration * Describe how to run the flocking demonstration * Describe how to alter the flocking demonstration by changing parameters using keyboard controls. * If there were issues with the bottlenecks explain what the issues were and how you resolved them. | 4  4  4  4  4 |  |
| Total: | 100 |  |

Comments: