The crowd simulation program was originally intended to be used with Dijkstra’s algorithm to find paths around obstacles simulated by cells within the grid that is created for the agents simulating people to walk on. I found it very interesting how the models were loaded from many different filetypes, as it shows how many moving parts go into modeling 3d agents. Running the crowdSim program on my Mac machine was very easy due to the makefile that can be used by unix systems. Once the correct GLUT libraries were included in the .cpp files, the makefile took care of the compilation commands within my hyper terminal and compiled the program without issue. The crowd that could be created was fixed at first with a for loop that was located within the Simulator class. Depending on how many iterations you wanted, you could change the size of the crowd by hard coding it in the for loop. Also, depending on the paths you set from the cal3d file, you can also add other models, but I stuck with the supplied ones for this project.

A while loop is created to handle the incoming file and read in the name of the model wanted that is used to reference a path located in the agent.cpp agent class. A function to get and set the path and the path name of the model that the user wants is set by the full path name and the data path of the agent. These values can be set from an input file as well as the speed with a set max velocity function and size of the model with the predefined setSize function that the user can change by inputting the changes into the input.txt file. I couldn’t fully implement the sizing portion of the input however, as when the size for some reason, as when I attempted to scale the models they disappeared.

I attempted with two different iterations of the crowdSim code to implement the controlling agent milestone. Unfortunately, when I added the code that we wrote together in class, the program had an error that looked like this:

crowdSim(10340,0x10c8515c0) malloc: \*\*\* error for object 0x404764f263f2c9e5: pointer being freed was not allocated

crowdSim(10340,0x10c8515c0) malloc: \*\*\* set a breakpoint in malloc\_error\_break to debug

[1] 10340 abort ./crowdSim

After doing a lot of research on the error, I found that it could be a number of things but is most often the issue with a memory leak or a misused and undeleted pointer. I was unable to properly implement the follow code because of this issue.

Integrating the flocking code with the crowdSim program was done by adding the force variables and separation components and cohesion components that were necessary for the flocking program to simulate properly, as well as the “if seen” algorithms that were used in the flocking program are added to the Agent.cpp agent class. This allows the already present agents to follow the separation components and by reducing the pathComponent value to 0, the agents are free to follow the flocking behaviors coded into the control statements.

To add some fun to the program, I added the menu from the flocking project and the x and y environment variables to the mouse button click handling function to add more agents to the environment. Adding these lines to the mousebutton function allowed this:

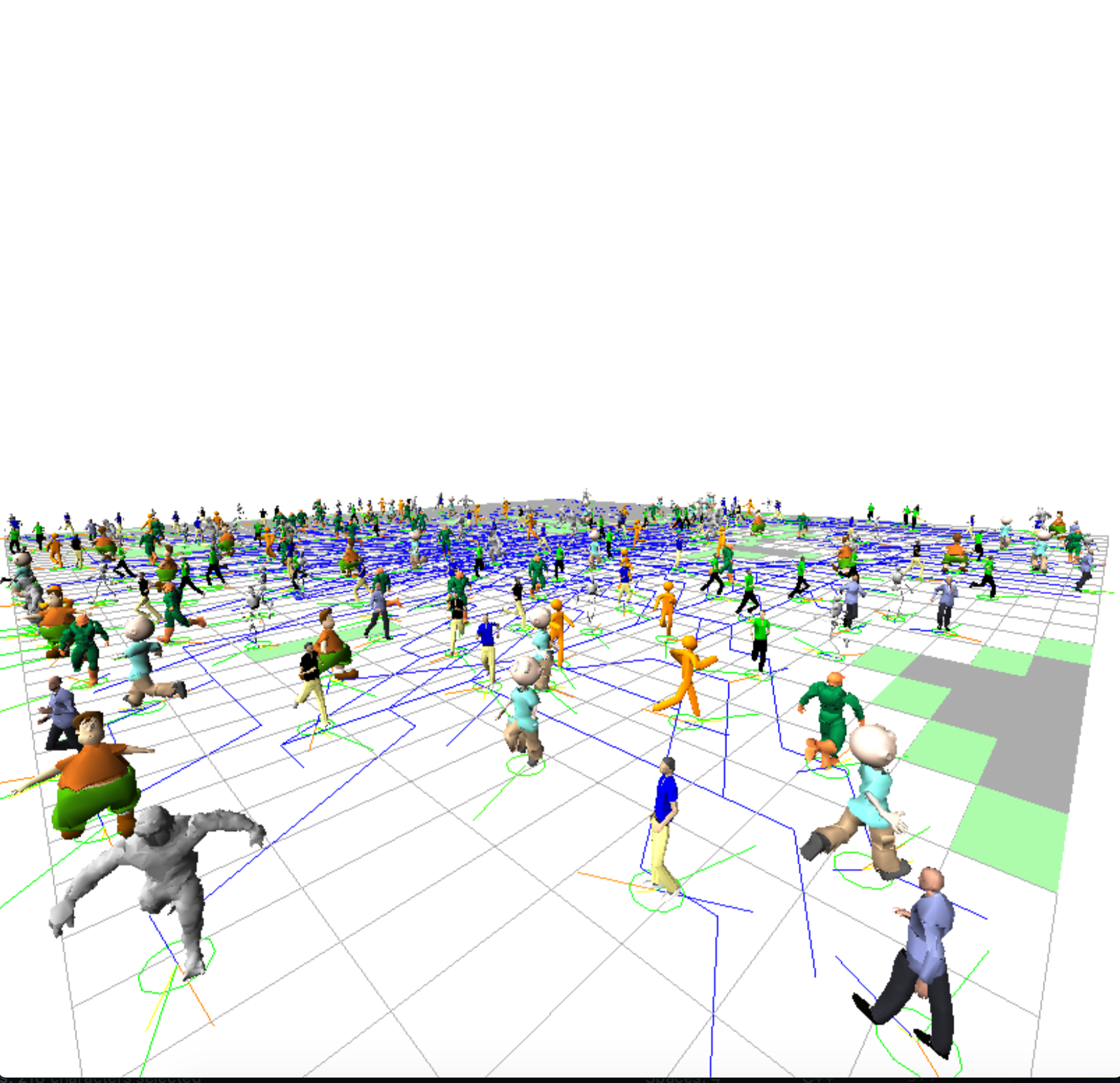
double x\_env = x - windowW / 2.0;

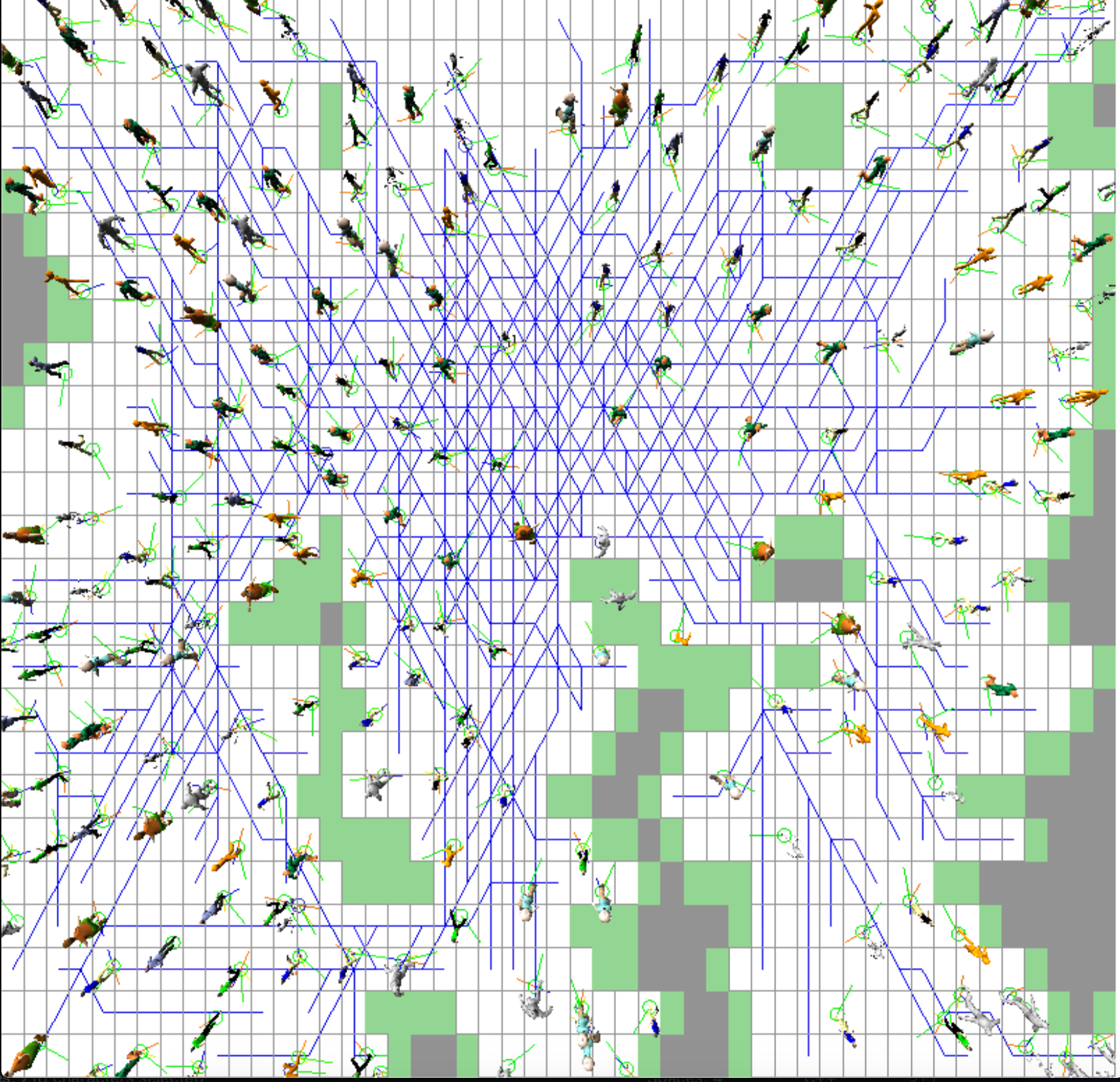
double y\_env = (windowH / 2.0 - y);

//Using this and passing in agents to add is selected from the menu

gSim.addAgent(agentsToAdd, x\_env, y\_env );

Screen Shots:





--Instructions for how to use the program:

Use the input.txt file and use this format to input user selected models upon runtime:

masterchief

1

12

Single spaced, with no space between other additional names. This corresponds to the name of the model, the speed of the models’ walking, and the size of the model. The names of available models are:

* dino
* fatboy
* figure
* masterchief
* suitman
* terminator
* cimpleman
* Coolkid

By right-clicking on the program while its running, a menu can be used to toggle the different menu options for the program. If you click on one of the “Add...” menu items, clicking on the surface of the crowd sim program will allow you to add more agents of random model types of the number you selected.