David Kopp

david.kopp.330@my.csun.edu

5/15/2016

Comp 565

Project 2

Project 2: Navigation and Pack Behavior with AGMGSKv7

**Disclaimer:** I was not able to complete all of the necessary parts for this project. Due to my personal matters, four of them to be exact, I was not able to complete three parts of the project. The issues are below:

1. **A-Star Path finding Algorithm:** I have never used this algorithm before so I decided togo off the pseudo code off the lecture notes.I have the algorithm coded, but it does not seem to correctly display the right path. I was trouble shooting it for a long time, but because of time I had to give in. A path is generated, but the open and closed nodes are not correct. The path looked to be correct. My next steps for this issue would be to break down the problem into a smaller one in a premade small graph simulation. Then I can figure out what is the problem with my implementation.
2. **NPAgent Pathing:** Since this works very tightly with the A-Star algorithm this was a double whammy for me. I am was able to make the NPAgent scan for treasures and call the astar algorithm to get a path, which I assigned to the agent as a new path that loops. But, I was not able to get a clear picture in my head at the time to fix the project 1 conditions to make it so that the agent will stay on course and go to the treasure. My next steps in an attempt to fix these issues are to restructure my if-conditions to handle the Agent’s behavior. For example: Is the agent on a treasure hunt, is the agent on their normal loop – create a-star path, and then check status of the treasures and stuff in the mean time.
3. **Pack Behavior:** I was able to get the dogs to follow me around and using the info in the lectures I believe everything is correct. The only issue is that when I stop moving they start to pack up in front of me in a limbo line. I did not have enough time, but what I wanted to do is have them look way every now and then and take a step. I was hoping that would help with them getting stuck together. It is also possible that my separation force is being calculated incorrectly. I will be looking into this over summer, because the subject matter is very interesting to me.

**Description of the Navigational Graph:**

The navigational graph took the major chuck of my time and now after completing it I have no idea why. What I did was I started with stamping out all the walls to make with a NavNodeEnum called COLLISIONPOINT. After I stamped out all the wall and temple pieces I started the stamping on the graph. I choose to start my stamping at vertex 3,3. My stamps are made 23x23 vertices in size. I used rectangles of the stamp and one enclosed around each collidable object and see if they contained each other. If there was a collision I would call FineStamp which proceeded to stamp vertex by vertex the entire 23x23 section. Otherwise I would stamp the 23x23 stamp with five nodes. (Which looks like a five on throwing dice). I did have to add two manual finestamps because the left edge of the walls where between stamps. I was worried about adjacency issues later. The adjacency list was created by looking at each node against another node in the NavGraph and looking the offset whether it was a normal stamp approx. 3550 or fine stamp approx. 160. If the two nodes where within distance of the offset, then an adjacency edge would be created.

The methods that where modified or created for the above description:

Entire NavGraph Class

Stage Class Methods:

createGraph, normStamp, fineStamp, distance

NavNode Class Methods:

NavNode, insertAdjacentNode

**Description of Dog Packing Behavior:**

This is not working correctly as explained above. What works is that the dogs are looked at one at a time and see if they are exploring or they are packing passed on the roll against the packing probability. If they are packing then I calculate the cohesion, alignment, and separation forces and apply it to the dog. The separation force takes into account not only the leader but also by the other dogs. The alignment is from the leader’s forward vector, and the cohesion is calculated by distance from leader.

The methods that where modified or created for the above description:

Pack Class methods:

Update, calcAlignment, calcCohesion, calcSeperation

Inspector Class methods:

Draw, setFlockingboard

**Description of Path Finding and NPAgent Traversal:**

The description explains all about the issues and my thoughts on these two problems. I would like to list the methods that are used in all these.

The methods that where modified or created for the above description:

NavGraph Class methods:

nearestGraphPoint, aStarAlgorithm, Draw

NPAgent Class methods:

Update – The commented out code is from Project 1.

**New Input for AGMGSKv7**

Press “N” to Enable Treasure Mode for the NPAgent. (Player will always be in treasure mode)

Press “L“ to Enable all MovableObject3D to follow the terrain. (Lerp)

Press “S” to Show the Treasure Score Board. (Works as a toggle)

Press “D” to show the pane with the dog probability of packing. (toggles)

Press “P” to increase the probability of packing 0, .33, .66, .99. (rolls over to 0)

**Treasure Locations:**

Treasure1 = (442, 466)

Treasure2 = (418, 442)

Treasure3 = (420, 481)

Treasure4 = (474, 474)

Treasure5 = (463, 417)