Ander Azzopardi SCFG Assignment

Task 6

Include a pdf in your assets folder with a short paragraph explaining the difference between Vector3.distance, distance-based code and A* pathfinding.

Vector3.distance takes 2 position variables and returns the distance between them. This is the slowest out of the 2. It also will never return a negative value as it doesn't subtract the variables.

Astar works by choosing the smallest path tree from the start node to the target node

Task 7

My game is a packman inspired game. Instead of the traditional map layout I have opted to use a random tile generator which ensures my game to be random. This means that every playthrough will have a different map combination. The enemies work with astar pathfinding which is an excellent pathfinder that will chase the player. I have implemented a high score system that works and when the game is closed the high score is saved. There is a menu scene which involves 3 levels of difficulty which increases the enemy's speed drastically. I have also included a tutorial which will help new players learn how to control their character and some basic survival rules.

As this is a new product bugs are to be expected and work will be done to fix issues and bugs related to code and the astar system.

Finally, some improvements planned is actual animations for our player and enemy objects. Build nicer assets for all our game objects instead of primitive shapes. Sound will be added to the game so there will be a soundtrack and also game sounds such as death sound and collecting point. A win scene will be added, and finally bigger and better map variations will be implemented in the project.

Task 8

A* pathfinding is a computer algorithm which is mainly and widely used in pathfinding and for traversing graphs. This algorithm efficiently plots a walkable path between multiple nodes, or points, on the graph.

When a map has many obstacles the pathfinding from a point to another (A to B) can prove quite difficult. For example, if you have a robot and it doesn't have much direction it will continue forward until it hits an obstacle.

However, the A* pathfinder introduces a heuristic into a regular graph-searching algorithm which plans ahead for each step as it takes the best decisions in order to reach the other point. If in this case the robot had A* it would find an efficient path to the end.

A* is a modification of Dijkstra's Algorithm that is optimized for a single destination. Dijkstra's Algorithm can find paths to all locations. It prioritizes paths that seem to be leading closer to a goal.

A* "assigns a weight to each open node equal to the weight of the edge to that node plus the approximate distance between that node and the finish. This approximate distance is found by the heuristic and represents a minimum possible distance between that node and the end. This allows it to eliminate longer paths once an initial path is found. If there is a path of length x between the start and finish, and the minimum distance between a node and the finish is greater than x, that node need not be examined". (Pathfinding - Wikipedia, 2022)

The heuristic used is a user defined function, so it estimates the distance from a given point to the destination. Instead of pushing the distance to the current node into a priority queue, you push in distance to the current node + estimated distance to the destination from the current node. This gives a higher preference to directions in which estimated distance is less.

You are always guaranteed to find the shortest path if you choose the heuristic function such that the estimated distance is always less than the actual distance. If the estimated distance is set to 0 the algorithm is the same as Dijkstra's.

An important Aspect of A* is f = g + h. The f, g, and h variables are in our Node class and get calculated every time we create a new node.

What do these variables mean?

F is the total cost of the node.

G is the distance between the current node and the start node.

H is the heuristic — estimated distance from the current node to the end node.

As seen in the website https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2 these following graphs will give a better understanding of how astar works.

7	6	5	6	7	8	9	10	11		19	20	21	22
6	5	4	5	6	7	8	9	10		18	19	20	21
5	4	3	4	5	6	7	8	9		17	18	19	20
4	3	2	3	4	5	6	7	8		16	17	18	19
3	2	1	2	3	4	5	6	7		15	16	17	18
2	1	0	1	2	3	4	5	6		14	15	16	17
3	2	1	2	3	4	5	6	7		13	14	15	16
4	3	2	3	4	5	6	7	8		12	13	14	15
5	4	3	4	5	6	7	8	9	10	11	12	13	14
6	5	4	5	6	7	8	9	10	11	12	13	14	15

Node 0 is that starting position and is marked in green. Node 19 is the end position and is marked in blue. The current node is at the red square which is Node 4.

G is the distance between the current node and the start so node 4 is 4 spaces away from node 0. This means that G is 1 more than the parent node (node 3) so for the current node (4) currentNode.g = 4.

H is the heuristic as mentioned before. So if to reach the end node (19) we have to go over 7 spaces and up 3 spaces.

If pythagoreas's theorem is applied which is $a^2 + b^2 = c^2$ we'll see that currentNode.h = $7^2 + 3$ or currentNode.h = 58.

The square root doesn't need to be applied and it can be skipped for each calculation on every node and still get the same output.

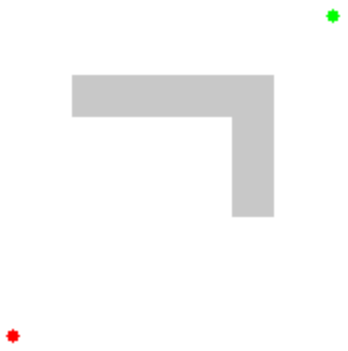
F is the total cost of the node. So add up h and g to get the total cost of our node.

currentNode.f = currentNode.g + currentNode.h.

Or currentNode.f = 4+ 58.

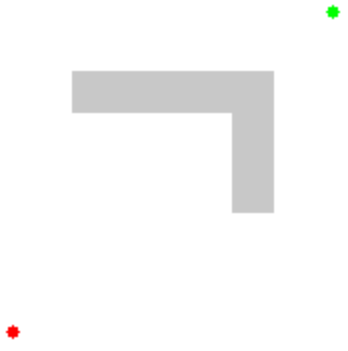
Or currentNode.f = 62.

With the f value the a* algorithm can rather than running through every node, pick the best one that has the best chance of getting to the end.



Dijkstra's Algorithm

Dijkstra's algorithm just keeps searching as it has no idea which node is the best so it has to run a calculation for every node



A* Algorithm

A* algorithm prioritizes the node with the lowest f and the best chance of reaching the end once it clears the obstacle

A* Algorithm pseudocode

The goal node is denoted by node_goal and the source node is denoted by node_start

We maintain two lists: **OPEN** and **CLOSE**:

OPEN consists on nodes that have been visited but not expanded (meaning that sucessors have not been explored yet). This is the list of pending tasks.

CLOSE consists on nodes that have been visited *and* expanded (sucessors have been explored already and included in the open list, if this was the case).

```
1  Put node_start in the OPEN list with f(node_start) = h(node_start) (initialization)
2  while the OPEN list is not empty {
3    Take from the open list the node node_current with the lowest
4         f(node_current) = g(node_current) + h(node_current)
5    if node_current is node_goal we have found the solution; break
6    Generate each state node_successor that come after node_current
7    for each node_successor of node_current {
8      Set successor_current_cost = g(node_current) + w(node_current, node_successor)
9      if node_successor is in the OPEN list {
10       if g(node_successor) ≤ successor_current_cost continue (to line 20)
11     } else if node_successor is in the CLOSED list {
12       if g(node_successor) ≤ successor_current_cost continue (to line 20)
13       Move node_successor from the CLOSED list to the OPEN list
14     } else {
15       Add node_successor to the OPEN list
16       Set h(node_successor) to be the heuristic distance to node_goal
17     }
18     Set g(node_successor) = successor_current_cost
19     Set the parent of node_successor to node_current
20   }
21   Add node_current to the CLOSED list
22 }
23 if(node_current != node_goal) exit with error (the OPEN list is empty)
```

Astar pseudocode

Quora. 2022. How does a star algorithm work?. [online] Available at: <https://www.quora.com/How-does-a-star-algorithm-work> [Accessed January 22, 2022].

Brilliant.org. 2022. A* Search | Brilliant Math & Science Wiki. [online] Available at: <https://brilliant.org/wiki/a-star-search/> [Accessed January 22, 2022].

Patel, A. Introduction to A*. Retrieved January 22, 2022, from http://theory.stanford.edu/~amitp/GameProgramming/concave1.png

Patel, A. Introduction to A*. Retrieved January 22, 2022, from http://theory.stanford.edu/~amitp/GameProgramming/concave2.png

En.wikipedia.org. 2022. *Pathfinding - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Pathfinding> [Accessed 22 January 2022].

2022. [online] Available at: <https://www.redblobgames.com/pathfinding/a-star/introduction.html> [Accessed 22 January 2022].

Task 9.

Instruction Manual

First download the zipped file online or from a link you have received. Then extract the WinRAR or zipped file which contains all the game files to a new folder on your desktop and call it Pacman Reborn to keep your game files neat and tidy. Go into the pacman reborn folder and search for a .exe file extension. Double click the pacman.exe file and wait for the unity to load. Select which settings best suits your personal computer, what display you want the game to run under and the quality that it should be. Once the settings are to your liking click play and enjoy

3 responses

Accepting responses

Summary | Question | Individual

Did you like this game

3 responses

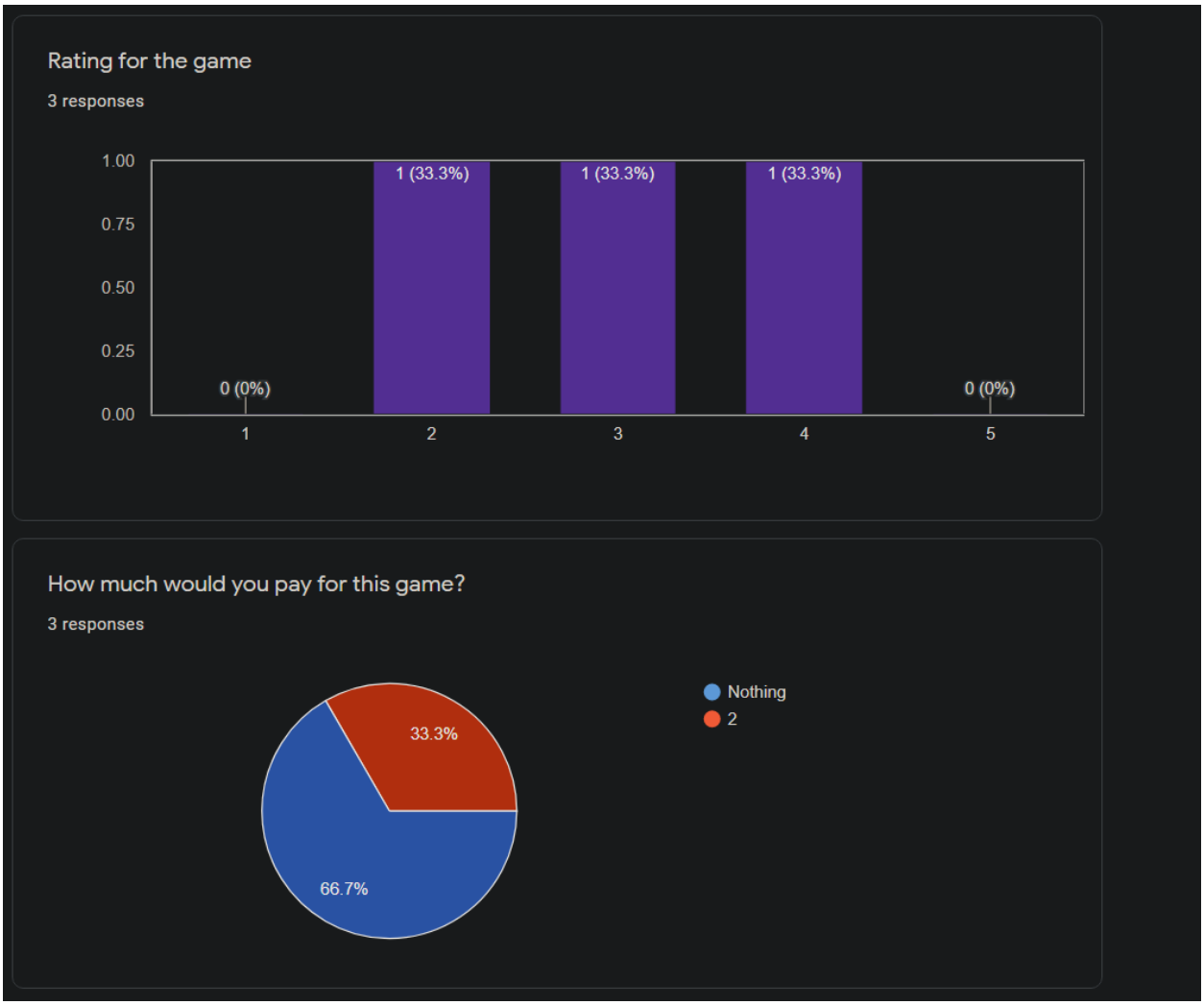
Explain what you liked or disliked from the game

3 responses

I liked the enemies and how the player moves. Art was simplistic and the tutorial was helpful. I disliked that you couldnt get more points and difficulty doesnt get harder

The tutorial was helpful and the menu was nice. Enemy sometimes goes through the obstacle or become one

I liked the game but it wasnt very difficult



3. Write a 400 – 500 word report regarding the game developed and the experience of players. It is recommended that you reference the feedback you received from the questionnaires.

As mentioned before the game still needs a lot of work. The base of the game is done but as many people pointed out theres a lot of issues and content to be dealt with and added. Sometimes the enemies collide with each other or pass-through obstacles, but this is because of the pathfinding system. The difficulty levels don't work very well. Currently the score only goes up to 10 and there is no win screen. These are all planned to be fixed shortly.