An AI Implementation of the Classic Snake Game

Sydney Cholewinski & Eric Jordan II

Snake is a common childhood game in which the player maneuvers a line that grows in length, with the line itself being the main obstacle. The original concept of this game comes from a 1976 arcade game called Blockade which has now birthed many different versions on multiple platforms, some called snake and some called worm. This game's significance for real world applications has to do with edge detection. This is a crucial step in image processing in order to mark the point where the light intensity changed significantly which is widely used to detect gray-scale and color images in various fields such as medical image processing, machine vision system, and remote sensing. In short, edge detection helps in better isolating the objects in an image. Other applications include fingerprint matching, license plate detection, assembly line inspection, and surveillance systems. Algorithms and heuristics needed to implement this game should include a shortest path such as breadth first search or depth first search, a longest path, and hamilton cycle. The shortest and longest path algorithms will determine when the snake captures the fruit. The hamilton cycle is put in place to prevent our snake from hitting its own body that way it can escape after eating the fruit.

In the game, the snake moves inside a 2-dimensional map. At each time step, the snake must either move forward, turn left, or turn right. The game requires that the snake cannot stop moving. The game will randomly generate and place one piece of food on the game map whenever there is not already food on the map. When the snake moves through a piece of food, the snake eats the food and the length of the snake grows by one. The goal is to eat as many pieces of food as possible until the length of the snake fills the game board. The game ends if the

snake collides into itself or any of the walls. The game map is set to be 20 units tall and 20 units wide. The snake begins at four units long at the top left corner, heading right. The snake can eat up to 396 pieces of food until it fills up the map.

We set up a path solver which finds the shortest path and the longest path from the snake's beginning to certain other points on the game map. It does not directly decide the next moving direction of the snake but it is used in the final decision process. A Hamiltonian cycle is used on the game map first and then directs the snake to eat the food along the path of the cycle. The snake takes any possible shortcuts in order to reduce the average steps for the snake to win. If you want to build a Hamiltonian cycle on a four by four map, then you would have to assign the path index to each point on the map.

In conclusion, Snake is a very simple concept that can be used in a game and in real world applications. There are hundreds and hundreds of games based on the concept. Real world applications of the theme edge detection vary in many different fields such as the medical field, imaging practices, and more. It’s crazy to think that something as simple as this could also have qualities that inspire important fields of work. As time goes on, just as professionals seek to improve their efficiency with edge detection, there are developers who always seek to improve this concept in hope to make a better game.In the snake game, there are different algorithms that you can use to make this game come to life. Using a breadth first search wouldn’t actually be the shortest path to use. The algorithm starts at the root node and explores the neighboring nodes until the snake finds the fruit then terminates. On the contrary, Depth first search will actually hold neighboring nodes to explore. This means while BFS starts exploring with present nodes, DPS starts with the highest depth first which will check to see if the graph contains a cycle or not. For future work, a slightly better algorithm to use for a game like this would probably be

Djikstra or A*. Between the two, A* typically runs better than Dikstra since it is an informed search rather than an uninformed search.

References

Melissa DeLeon. 2000. A Study of Sufficient Conditions for Hamiltonian Cycles. (2000). Retrieved 2020

Urooj, Wajiha. “Snake Game.” Edureka!, 21 July 2020, www.edureka.co/blog/snake-game-with-pygame.

Wikipedia contributors. “Snake (Video Game Genre).” Wikipedia, 23 Oct. 2020, en.wikipedia.org/wiki/Snake_(video_game_genre).

T. Poggio and H. Voorhees. 1988. A regularized solution to edge detection. Journal of Complexity 4, 2 (1988), 106–188. DOI:http://dx.doi.org/https://doi.org/10.1016/0885-064X(88)90024-6.