games ai notes for report

For my snake I started from individual causes of death and expanded its intelligence from there. For example, I first had to stop it from running into the wall, so I made it turn when hitting a wall. Then I had to stop it from going in circles and send it to food.

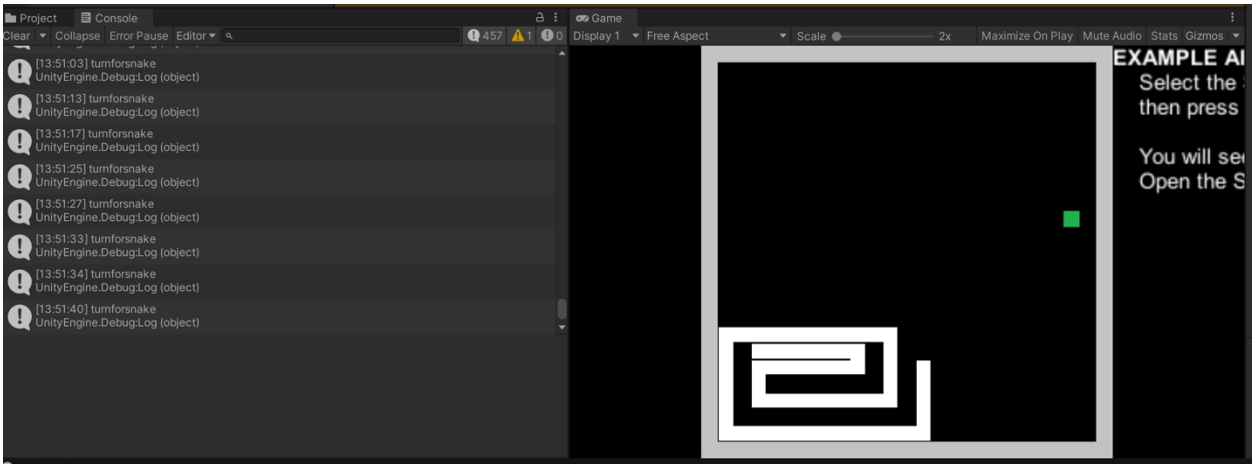
Initially I tried to do move towards food after triggering movement sequence of going to wall. I wanted it to alternate from getting food and spreading its body out by going alongside the wall, however I quickly realized it works on frame-by-frame basis so this didn't work. "moveTowardsFood" would happen for one frame and then then program perceived that action as being complete.

I then rethought my method. I made it reach for food if it was on the same axis as this would still be true after another frame. Then I had to start try to prevent it from hitting itself. I made the snake find which area has more room every time it hits a wall or itself and go towards this direction.

Now the main way he was dying was he wasn't able to escape when he curled up, and food wasn't available. I made it so as he curled there way an escape route available by making the snake hit sensitivity 1. This caused a spiral that had a potential exit that opened up when the body lengthened. However, I realized as the snake was turning in it would work well because it could move where there was more room which would naturally lead to the center. However, when it got to the end there was equal distance to either side of the snake. One side however, led to an exit and one led to it trapping itself.

To overcome this problem, I realized I had to find some way of changing the snake turning mechanism when the food was unavailable, so his behavior changed to 2 things:

1. turn the opposite way to what he turned last time, so he turns towards escape. When he gets to this position there is even space left or right
2. turn when snake comes into direct contact with itself as opposed to i square before it hits itself.



I then thought another way to prevent the snake from getting trapped was to spread the snake out by wanted to make the snake go in diagonal motion is food was diagonal from snake (Comparing Search Algorithms on Snake - AI Final Project, 2021).

problems: if snake is facing a different direction diagonal motion would be different instruction.

I solved diagonal problem by creating counter that alternated between 0 and 1 which would trigger either a North, or east if food was up diagonal.

I realized however that it got the best score when this was only implemented in one direction as this seemed to allow more opportunity for the spiral unwinding tactic.

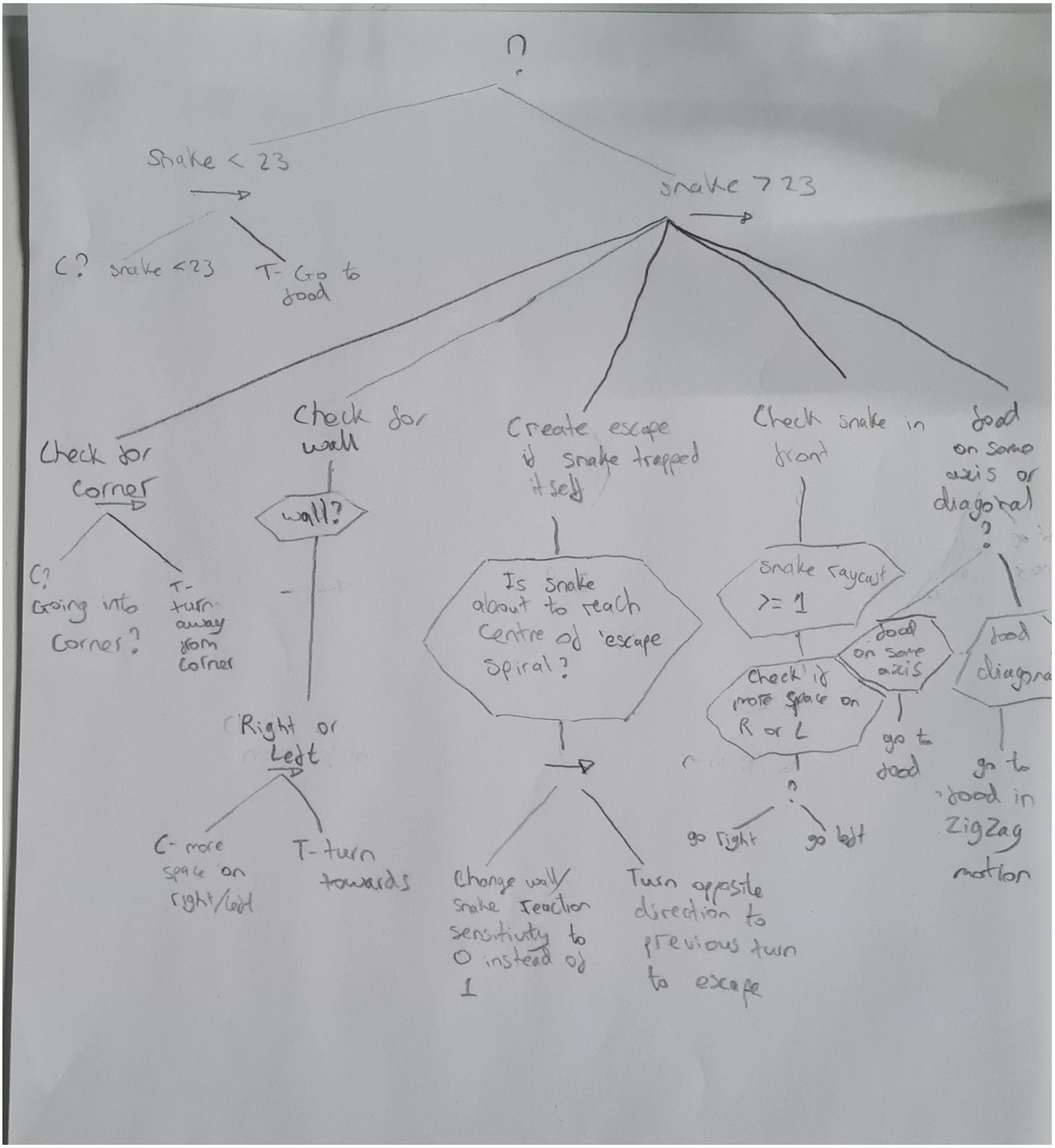
I then found if the snake was on the wall and saw food it would often get trapped in a dead end with the snake body on one side and the wall on the other. I made a function which took into account the direction it was going and whether there was a snake to the upper right of the snake and a wall on the other side. If this was the case the snake would turn away from the dead end, if that dead end was on a wall.

I realized also that the snake was moving towards itself and choosing this instead of a space with an escape route, if the distance towards its body was less than the distance towards the wall. To fix this I made the snake prioritize the wall ray cast over the snake ray cast.

One more branch I made but didn't implement in the end in the behaviour tree, is a function which changes its direction before the snake hits a wall if a significantly larger space opens up. I did this by storing a previous raycast value and a current raycast value, and then comparing the two results. If the current value has over a certain threshold of more available space, then it will go towards it. I took this out as it helped slightly with the score on the challenge but it had more of an impact in decreasing the score of the snake when it wasn't in challenge mode.

One final thing I added which isn't included in my behaviour tree diagram below, was a group of functions that make the snake move towards food if it is stuck in a loop. This was made soley for the challenge part of the assignment as this was constantly making my snake get stuck. I did this by making a threshold of duplicate co ordinates for the snake head to prevent repeat patterns.

Interestingly this did seem to have an impact on my non challenge score as it seemed to lower it a bit. I realised this was because my tuple that stored the coordinates wasn't able to be set to false. I'm not sure why this was. This meant my score average on the normal challenge was significantly decreased. To see it performed at the normal level the functions:
bool duplicates(),void clearDuplicates(), and void addValuesToCoordList() would need to be commented out. As well as the last sequence node of the behavior tree which calls these functions.



Youtube.com. 2021. *Comparing Search Algorithms on Snake - AI Final Project*. [online] Available at: <https://www.youtube.com/watch?v=hrMI7ppUGyU> [Accessed 7 December 2021].

Hikaru Clark, T., 2021. *Training a Snake Game AI: A Literature Review*. [online] Medium. Available at: <https://towardsdatascience.com/training-a-snake-game-ai-a-literature-review-1cdddcd1862f> [Accessed 7 December 2021].

James, M., 2021. *The Perfect Snake*. [online] I-programmer.info. Available at: <https://www.i-programmer.info/news/144-graphics-and-games/5754-the-perfect-snake.html> [Accessed 7 December 2021].

Note* My challenge score increased after my final implementation – I will email the results as my testing was too slow on my computer to show this.