Agent for Snake Game

Rodrigo Abreu, 113626
Eduardo Lopes, 103070
João Neto, 113482

Architecture

Files:

- Tree Search
- Snake Domain
- Agent

Tree Search: We used a tree search strategy to calculate the paths. The variant was A* for when the agent has traverse, because it guarantees that the path found is optimal, and Greedy for when the agent does not have traverse, because it calculates a path faster. This takes into account the entire body of the snake and not just the head.

Snake Domain: Domain of the snake, calculates the coordinates to avoid depending on the value of the traverse, if it isn't active, avoids walls and edges of the map, otherwise ignores it. The body is dynamically calculated to avoid it efficiently.

Agent: It applies the logic of the algorithm and calls Tree Search for the Snake Domain to calculate paths, when necessary, for example, when calculating the path to a checkpoint or fruit.

Algorithm

Search for food: We generate checkpoints on the map (coords that the snake should visit), these are generated differently based on traverse value:

- Traverse: The checkpoints are generated so the snake runs the map row by row separated by its sight size.
- No traverse: The checkpoints are generated so the snake runs the map row by row separated by its sight size in “S” shape.

Both checkpoint generations are dynamic, based on the position the snake is on.

Sees food: When the snake sees food, if only one, goes to that its coords and eats it. If it is more than one stores both foods in a cache and goes to the nearest one and then the other one. If going to a food sees another it adds the new one to the cache and recalculates the closest one.

Handling super food: When the agent does not have the traverse or the sight is small, the agent goes on super food direction to eat it, when it appears on its sight. Otherwise, the snake ignores it (doesn’t avoid it).

Multiplayer mode: Our strategy is to avoid the other snake if it appears on sight.

Avoid trapping itself: Avoids some dangerous situations, but it is still the agent biggest flaw.

Changes 2nd delivery

- Now we pass the full snake body to the domain, instead of only the snake head position.
- Calculate the snake body positions dynamically in each step.

Agent Benchmark

1st delivery:

- Average: 82.7 (10 runs, including 2 multiplayer runs winned)
- Highest Score 186 (single player, not related to the average)

2nd delivery:

Test (10 runs, single player):

- Average: 80,3/1964,4
- Max: 137/3000
- Min: 33/806
- Median: 76

We obtain more consistency on the 2nd delivery (the multiplayer improved a lot the results of the 1st delivery).

Conclusion

Single player:
The key points to have success in this mode are:

- An efficient fruit search algorithm.
- Maintaining the sight big the longest time possible
- Get Traverse (less chance of getting trapped and shorter paths)
- Avoid dangerous spots to getting trapped. (ex: dead end)

Multiplayer: The same as the single player but playing defensively, avoiding the enemy snake (if successful, the enemy snake eventually will hit my agent body).

Calculating paths: An Adequate path calculation time limit, will balance the efficiency of the path search and the time consumed to make a decision, in time and correct.