Hide 'n' Eat

- Siddharth Verma, Sanghamitra Ahirrao, Swetha Mahadevan

Key Goals:

-Implement a simulation where a 'Snake' is trying to reach its goal, i.e., an 'Apple' while trying to save its tail from a 'Ghost'.
-Use different motion planning algorithms for the Snake and Ghost to demonstrate a comparison of the algorithms.

Problem Overview:

The Snake plans its path using A* and the ghost uses an Isotropic Power Law model. The snake follows a grid based environment and can only move Up, Left, Down or Right. The ghost has no restrictions on its direction of movement except avoiding obstacles. It is capped to a maximum speed. The environment has static obstacles which should be avoided by both, the Snake and the Ghost. The length of the snake increases every time it has eaten an Apple and reduces every time the ghost catches its tail. The apple is generated randomly at accessible locations.

Initially, we intended to use RRT* for path planning of the snake. However, the RRT* algorithm was not quick enough for planning paths with constantly changing environments. Hence we decided to use A*.

Planning Algorithm Description:

Main Program(main.py):
1. The grid is drawn with obstacles using a file, obstacle.py
2. Starting locations of the Snake, Ghost and the Apple are set.
3. The coordinates of obstacles and the snake's body are updated in the grid and set as inaccessible space.
4. The grid and the location of the ghost is sent to the A* program(astar.py) for path planning.
5. The coordinates of the tail of the snake are sent to the ghost's program(powerlaw.py).
6. Power law model returns the updated coordinates of the Ghost.
7. The locations of the ghost and the snake are updated.
8. Return to Step 3.

A-Star:
1. The snake builds a heuristic map with the location of the Apple as its goal.
2. The heuristic is then updated with the location of the ghost. The heuristic values around the ghost are increased in order to make the snake avoid the ghost.
3. A path is planned while considering the obstacles and the snake's own body as inaccessible space.
4. The coordinates of the path are sent to the main program.

Isotropic Power Law Model:
1. The location of the tail of the snake is set as the goal for the ghost.
2. The ghost checks for obstacles within its sensing radius.
3. The obstacles are simply defined as stationary agents. They dampen the force experienced on them

by the ghost, and reflect it back to the ghost.
4. Force experienced by the ghost is capped to a maximum value and its position is updated.
5. The updated coordinates are returned to the main program.

Graphics:

The environment and objects are drawn using Tkinter. The grid size is 40x40 cells where each cell consists of 30x30 pixels. The data for computation is in the form of a smaller grid and is scaled up to pixel values for visual representation. Images are drawn to represent the apple and the ghost. Sounds are also provided when the Snake eats and apple and when the ghost eats the snake's tail.

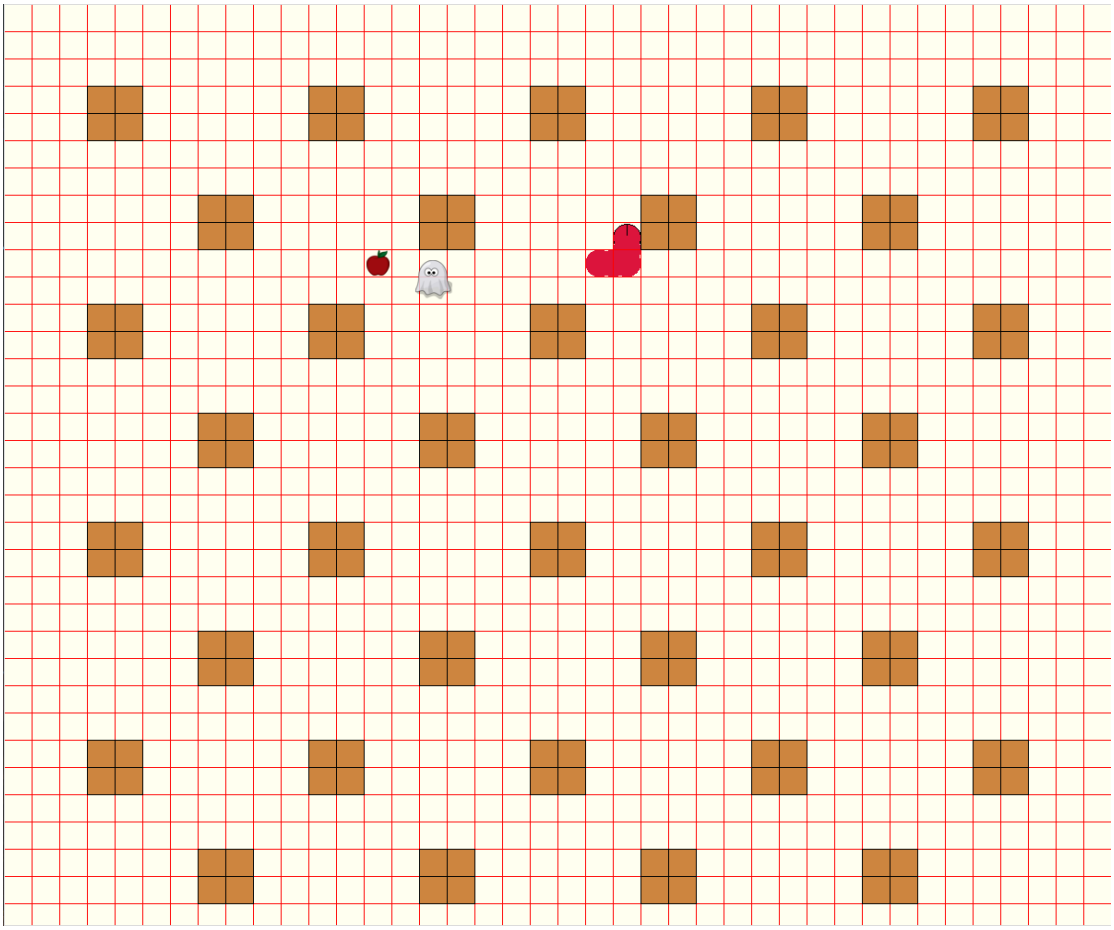

Figure 1. Grid with Obstacles(Orange), Snake (Red), Ghost(White), Apple (Dark Red)

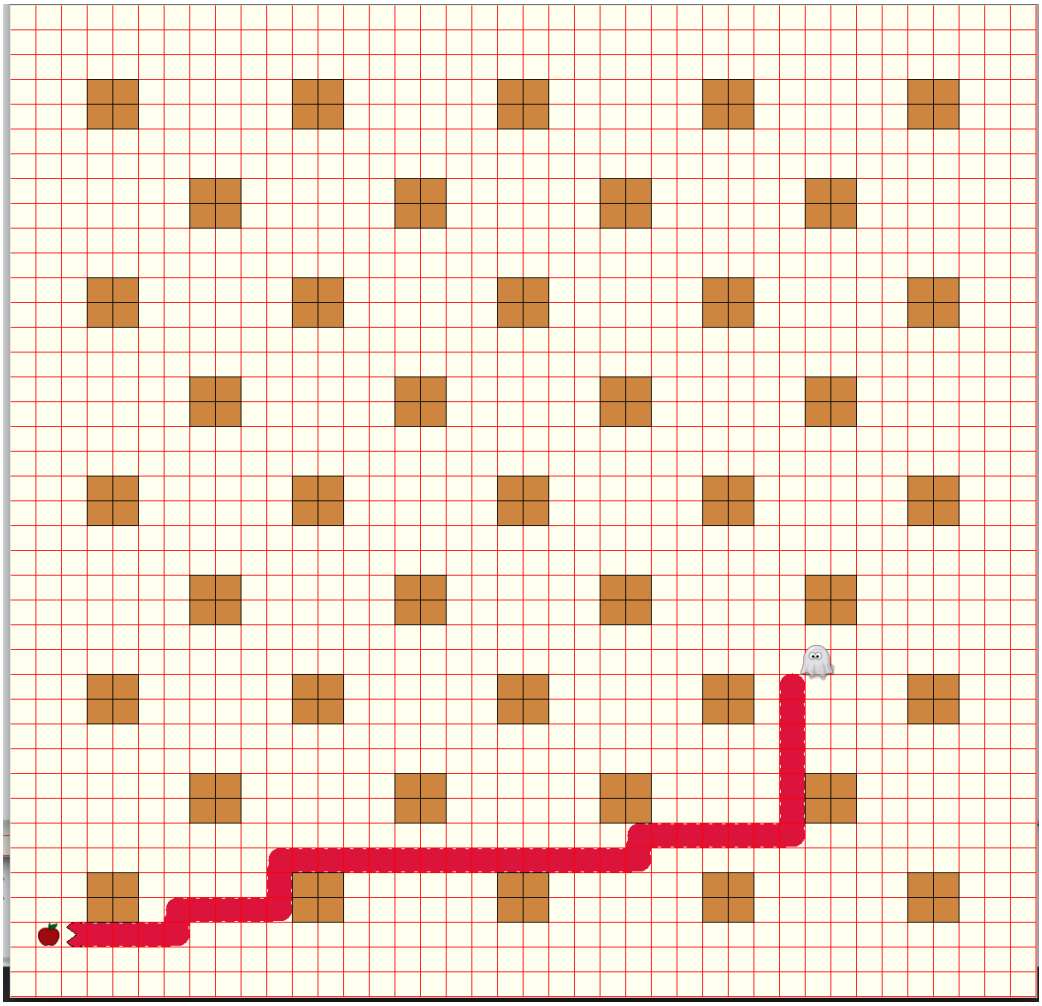

Figure 2. Image of Snake about to catch the tail, and Snake about to eat the Apple

Bugs and Limitations:

Lags - Generally lags occur when the Apple is generated close to the location of the ghost. Because of the higher heuristics around the ghost, more number of nodes are expanded while planning the snake's path to the goal.
Glitches – The ghost is seen to move over the obstacles instead of avoiding them. (Figure 3).This occurs rarely.
Impossible paths – In cases where the path to the apple is completely blocked by the body of the snake (occurs when the snake is very long), the algorithm is unable to find a feasible path since the body of the snake is also part of inaccessible space. A workaround for this could be to simply keep moving in a random direction till the path clears out.

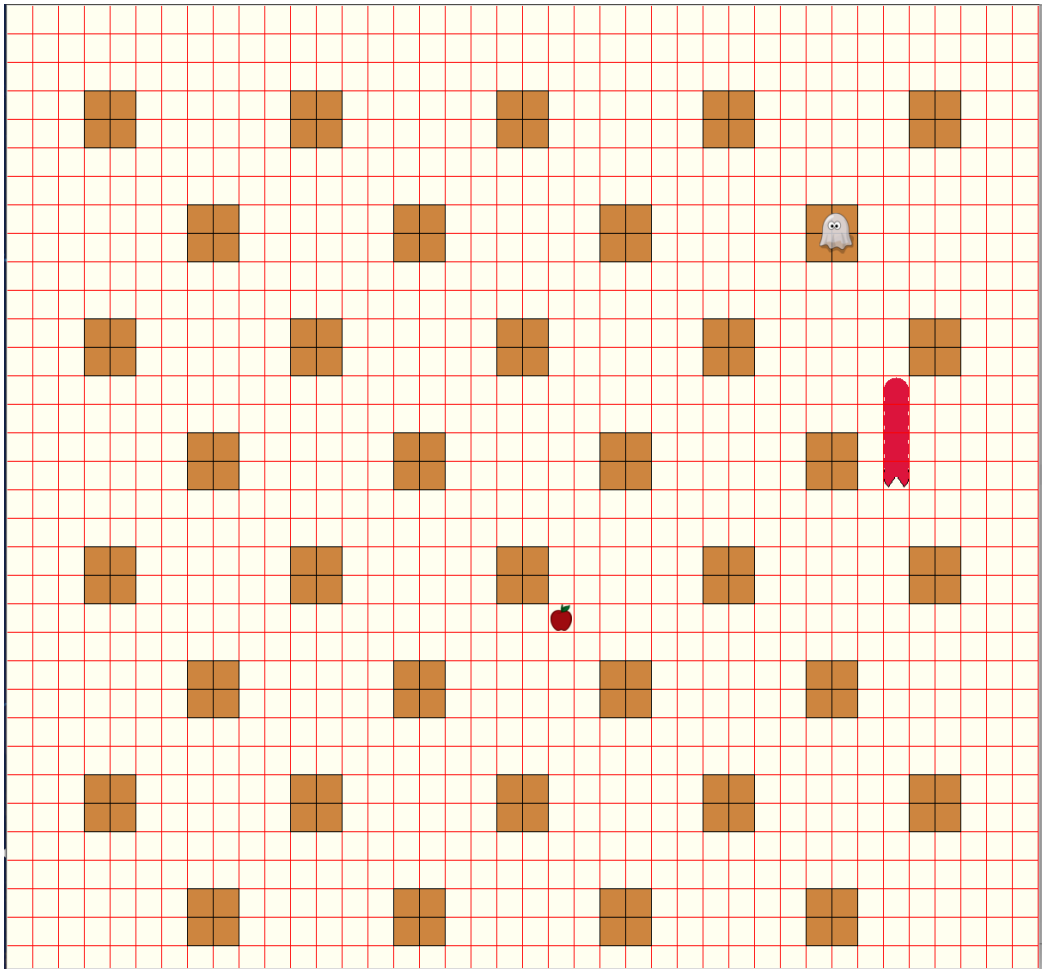

Figure 3. Ghost moving over an obstacle

Conclusions:

We were able to successfully program a simulation where two agents simultaneously plan paths to their dynamic goals using different path planning algorithms. The simulation of a snake and ghost also highlights the advantages ad disadvantages of their respective algorithms.