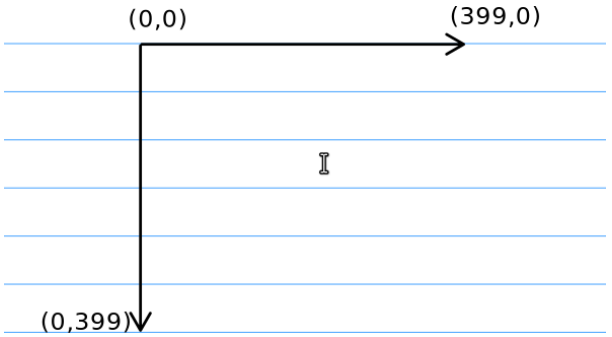
Generic Requirements for State Space Search Algorithm

Requirements

1. Method for state representation
2. Method for generating new states
3. Method for evaluating new states i.e. Cost functions and heuristics

State Representation



- Define a way to represent your current state.
- In our skeleton code, the state object contains two attributes:

  1. Food coordinates

  2. List of SnakeBodyAttr

- The SnakeBodyAttr defines where the each line composing the snake's body is located and how it will move at every iteration.

  X_1,Y_1 starting point of the line (For the first line this points to the head of the snake)

  X_2,Y_2 ending point of the line (For the last line this points to the tail of the snake)

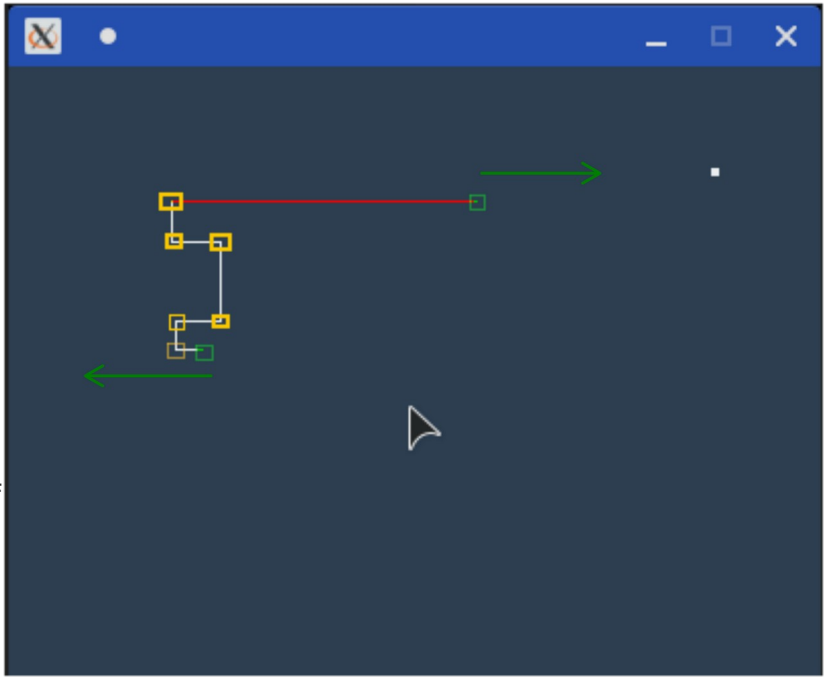
More on next slide

  X-incr,Y-incr represents what will happen to the respective point at every iteration.

- At any given instance only the starting point of the first line (head) and the end point of the last line (tail) will have both non-zero incr values. All other intermediate lines will be stationery.

Generating New States

- The motion is defined by X and Y increment values. All points other than the head and the tail are stationery.
- For all Yellow points, the X and Y increment values are **0**.
- For all Green points, the X and Y increment values are based on the table below.
- The last line will get deleted once its $X_1,Y_1 == X_2,Y_2$. Then the last point of the previous line becomes the tail of the snake.
- You can generate new states by adding [-1,0], [1,0], [0,-1], [0,1] to the head coordinates for making Left, Right, Up, Down operation respectively.



```
#   x1_incr/x2_incr     |      y1_incr/y2_incr     |       Direction
#            0          |           1              |       Down
#            0          |          -1              |       Up
#            1          |           0              |       Right
#           -1          |           0              |       Left
#            0          |           0              |       Stationary
```

Evaluating New States

- You need to define a set of heuristics and a cost function to evaluate each state.
- The primary cost function should be Manhattan distance as it fits our movement constraints perfectly. It will also improve performance of the game as it discourages unnecessary turns.
- Additional heuristics may include euclidean distance, collision detection or even a mathematical model for defining the probability of the snake getting stuck in its own loop.