Sneaky Snake Battlesnake Contestant

Callbacks:

The server will POST to our server, and we will respond with one of these callbacks. All game moves should be done during /move.

/start

HTTP POST `/start` will be called when a new game is started. Three attributes are included, a unique game_id, the map height, and the map width. The response should include the snake's body color, its head_url, the snake's name, and its taunt, and should have HTTP status code `200 OK`.

Table 1 - `/start` Request Attributes

Attribute	Type
game_id	UUID
height	integer
width	integer

Table 2 - `/start` Response Attributes

Attribute	Type	
color	string	A valid css color. HSL, RGB, hex, or named color.
head_url	URL	*optional* – URL of an image to use as the head of your snake.
name	string	Your snake's name
taunt	string	*optional* – Message to display in the game client

Example request

Example response

```
{
      "color": "#FF0000",
      "head_url": "http://placecage.com/c/100/100",
      "name": "Cage Snake",
      "taunt": "OH GOD NOT THE BEES"
  }
```

/move

HTTP POST `/move` – This callback is requested any time it is the client's opportunity to move. This callback should be used for all game logic, and we only have 200ms to respond.

Table 3 - `/move` Request Attributes

Attribute	Type	
food	Array<Point>	Array of all food currently on the board
game_id	UUID	
height	integer	
width	integer	
turn	integer	The current turn
snakes	Array<Snake>	Array of all living snakes in the game
you	UUID	A reference to your snake's id, the snake object can be found in `snakes`.

Table 4 - `/move` Response Attributes

Attribute	Type	
move	string	"up" | "left" | "down" | "right"
taunt	string	*optional*

Example request

Example response

```
{
    "move": "up",
    "taunt": "gotta go fast"
}
```

OUTPUT:

Operation:

Features

What can Sneaky Snake do?

- Move up / down / left / right
- Avoid bigger snakes
    - Weigh cells around big snakes heads heavily
- Ignore smaller snakes
    - Weigh cells around smaller snakes heads lightly
- Potentially destroy smaller snakes
    - Reward paths that intersect smaller snakes heads path
- Dumb simulator of next potential move for opponent snakes
    - Possibly one or two space radius around snakes, looking for food or heads of smaller snakes
    - Based on the last move they made (ie a snake can't move backwards towards food)
    - Potentially pathfinding their future path
- Memory of old snake movements
    - Ex: if they move immediately towards food, make a target of destroying them or eating food first
- Head towards food if closest snake to food, otherwise evaluate if opposing snake is traveling towards or away, and if bigger (both including food and not)
- Remember our body length and position to avoid collisions
- Remember our health, weigh decisions based on how much is left
    - Avoid aggressive decisions if health is low
    - Potentially go into "open-area" health mechanism, where we circle one area waiting for food
- Weigh snake locations, snake potential moves, and food locations to find optimal path
    - Do in layers
        - Weigh Snake locations
        - Weigh Simulated snake locations
        - Weigh food items
- Create a priority queue of goals we'd like to go to (foods & smaller snake head locations)

- Simulate what the board would look like if we took our path. If there is no clear path to our tail, then we skip that goal
- Snake tail avoiding
- After 180ms if we don't have a decisions, go to simplified decision algorithm.
    - Instead, first do a simplified algorithm, then do Djikstras with food, then with snakes, etc. We refine the choice of movement with every next algorithm
- Timer interrupt to submit best next step at given time
- Using `python-igraph` for Djikstra's algorithm
- Person controlled interface
- Graphical output for pathfinding weights
- Output each best decision from each algorithm

Functions

getSnakes()

Creates a list of snake objects, with their lengths and full body locations.

getFoods()

Creates a list of food objects and returns a list with their distances from our snake's head.