Drone Pathfinding + Efficiency and Speed Optimizations

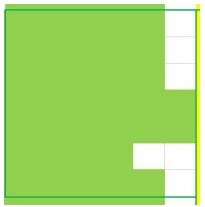
By Bridget Whitacre, Sylvia Krech, Sam Carpenter

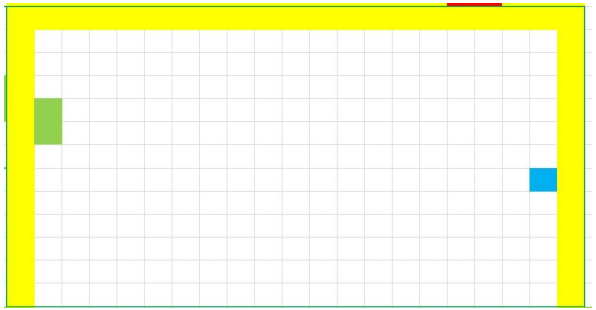
Introduction

- Core assumptions:
  - If we take the shortest possible path, it will be both the fastest, and most fuel efficient, in terms of spaces travelled.

Graph To List Conversion

- This python file converts the map to a list of boxes with coordinates.
- The map is divided up into 2 separate lists of boxes to help process the map.
- Result is a list of all of the coordinates contained inside the 2



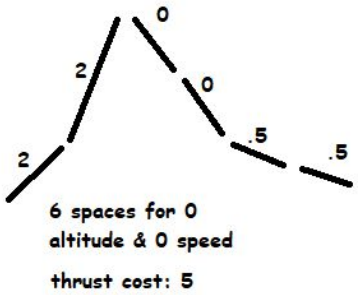


```
[[['B', 3], ['B', 4], ['B', 5], ['B', 6], ['B', 7], ['B', 8], ['B', 9], ['C', 3], ['C', 4], ['C', 5], ['C', 6], ['C', 7], ['C', 8], ['C', 9], ['D', 3], ['D', 4], ['D', 5], ['D', 6], ['D', 7], ['D', 8], ['D', 9], ['E', 3], ['E', 4], ['E', 5], ['E', 6], ['E', 7], ['E', 8], ['E', 9], ['F', 3], ['F', 4], ['F', 5], ['F', 6], ['F', 7], ['F', 8], ['F', 9], ['G', 3], ['G', 4], ['G', 5], ['G', 6], ['G', 7], ['G', 8], ['G', 9]], [['H', 3], ['H', 4], ['H', 5], ['H', 6], ['H', 7], ['H', 8], ['H', 9], ['H', 10], ['H', 11], ['H', 12], ['H', 13], ['H', 14], ['H', 15], ['I', 3], ['I', 4], ['I', 5], ['I', 6], ['I', 7], ['I', 8], ['I', 9], ['I', 10], ['I', 11], ['I', 12], ['I', 13], ['I', 14], ['I', 15], ['J', 3], ['J', 4], ['J', 5], ['J', 6], ['J', 7], ['J', 8], ['J', 9], ['J', 10], ['J', 11], ['J', 12], ['J', 13], ['J', 14], ['J', 15], ['K', 3], ['K', 4], ['K', 5], ['K', 6], ['K', 7], ['K', 8], ['K', 9], ['K', 10], ['K', 11], ['K', 12], ['K', 13], ['K', 14], ['K', 15], ['L', 3], ['L', 4], ['L', 5], ['L', 6], ['L', 7], ['L', 8], ['L', 9], ['L', 10], ['L', 11], ['L', 12], ['L', 13], ['L', 14], ['L', 15], ['M', 3], ['M', 4], ['M', 5], ['M', 6], ['M', 7], ['M', 8], ['M', 9], ['M', 10], ['M', 11], ['M', 12], ['M', 13], ['M', 14], ['M', 15], ['N', 3], ['N', 4], ['N', 5], ['N', 6], ['N', 7], ['N', 8], ['N', 9], ['N', 10], ['N', 11], ['N', 12], ['N', 13], ['N', 14], ['N', 15], ['O', 3], ['O', 4], ['O', 5], ['O', 6], ['O', 7], ['O', 8], ['O', 9], ['O', 10], ['O', 11], ['O', 12], ['O', 13], ['O', 14], ['O', 15], ['P', 3], ['P', 4], ['P', 5], ['P', 6], ['P', 7], ['P', 8], ['P', 9], ['P', 10], ['P', 11], ['P', 12], ['P', 13], ['P', 14], ['P', 15], ['Q', 3], ['Q', 4], ['Q', 5], ['Q', 6], ['Q', 7], ['Q', 8], ['Q', 9], ['Q', 10], ['Q', 11], ['Q', 12], ['Q', 13], ['Q', 14], ['Q', 15], ['R', 3], ['R', 4], ['R', 5], ['R', 6], ['R', 7], ['R', 8], ['R', 9], ['R', 10], ['R', 11], ['R', 12], ['R', 13], ['R', 14], ['R', 15], ['S', 3], ['S', 4], ['S', 5], ['S', 6], ['S', 7], ['S', 8], ['S', 9], ['S', 10], ['S', 11], ['S', 12], ['S', 13], ['S', 14], ['S', 15], ['T', 3], ['T', 4], ['T', 5], ['T', 6], ['T', 7], ['T', 8], ['T', 9], ['T', 10], ['T', 11], ['T', 12], ['T', 13], ['T', 14], ['T', 15], ['U', 3], ['U', 4], ['U', 5], ['U', 6], ['U', 7], ['U', 8], ['U', 9], ['U', 10], ['U', 11], ['U', 12], ['U', 13], ['U', 14], ['U', 15], ['V', 3], ['V', 4], ['V', 5], ['V', 6], ['V', 7], ['V', 8], ['V', 9], ['V', 10], ['V', 11], ['V', 12], ['V', 13], ['V', 14], ['V', 15], ['W', 3], ['W', 4], ['W', 5], ['W', 6], ['W', 7], ['W', 8], ['W', 9], ['W', 10], ['W', 11], ['W', 12], ['W', 13], ['W', 14], ['W', 15], ['X', 3], ['X', 4], ['X', 5], ['X', 6], ['X', 7], ['X', 8], ['X', 9], ['X', 10], ['X', 11], ['X', 12], ['X', 13], ['X', 14], ['X', 15], ['Y', 3], ['Y', 4], ['Y', 5], ['Y', 6], ['Y', 7], ['Y', 8], ['Y', 9], ['Y', 10], ['Y', 11], ['Y', 12], ['Y', 13], ['Y', 14], ['Y', 15], ['Z', 3], ['Z', 4], ['Z', 5], ['Z', 6], ['Z', 7], ['Z', 8], ['Z', 9], ['Z', 10], ['Z', 11], ['Z', 12], ['Z', 13], ['Z', 14], ['Z', 15], ['AA', 3], ['AA', 4], ['AA', 5], ['AA', 6], ['AA', 7], ['AA', 8], ['AA', 9], ['AA', 10], ['AA', 11], ['AA', 12], ['AA', 13], ['AA', 14], ['AA', 15], ['AB', 3], ['AB', 4], ['AB', 5], ['AB', 6], ['AB', 7], ['AB', 8], ['AB', 9], ['AB', 10], ['AB', 11], ['AB', 12], ['AB', 13], ['AB', 14], ['AB', 15]]]
```

Pathfinding Basic Plan

- Goto a corner
- Start doing “scanlines”
  - If the space below isn’t large enough, begin doing “square waves”
- If going to run into the start early, don’t
  - Then the new “wall” for the scanlines to run up against is the cord as the start pos.
- After that “scanwave” the way back up.
  - Once out of tiles, fly back to start as efficiently as possible.

Speed and Fuel Efficiency Plan



- To move as quickly as possible we:
  - Max accel, max accel, and then accel
  - Then cruise the remaining distance
  - Travels at maximum speed of 200
  - This keeps us at 200 ft, the exact height to be able to ignore the 200ft requirement.
  - Given that we only touch each tile once, and cruises only burn 1 fuel point, there is no way we are going over 500 fuel points.

- To move as efficiently as possible we:
  - Start by using Steep Climb -> Glide -> Glide to lose 10 speed while only costing 2 thrust units per 3 spaces (150% efficiency!)
  - Do this down to 70 (would do down to 60 but this somehow is slightly less efficient, at least in this scenario)
  - Continue forever through the pattern Climb -> Steep Climb -> Glide -> Glide -> Glide Cruise -> Glide Cruise
  - Travels 6 spaces for 5 thrust, with no loss of altitude or velocity (120% efficiency compared to cruising!)

Results

Optimizing for Fuel Efficiency: ~778.5 seconds, 239.0 fuel units remaining (261.0 used)

75.0	774.4502678264278	239.5	GLIDE-CRUISE	NW	AA10	200.0	70.0	778.4931249692848	239.0

Optimizing for Speed: ~316 seconds, 182.5 fuel units remaining (317.5 used)

200.0	314.53968253968253	183.5	CRUISE	NW	AA10	200.0	200.0	315.95468253968255	182.5

How To Run This Program

- Run ExportFormatted.py from the command line
- Solution will be located in solution.csv
- Edit the variable “maximizefuelefficiency” inside ExportFormatted.py to change priority to either minimize fuel or time spent
- To alter graph change the definition on line 103

```
maximizefuelefficiency = True
```

```
103     p.setGraph([[['B','G'],[3,9]],[['H','AB'],[3,15]]])
```

Questions?