Drone Sim Extension: Singleton Data Collection

By Andrew Carlson (carl6090) and Josh Subhan (subha013)

What our extension is:

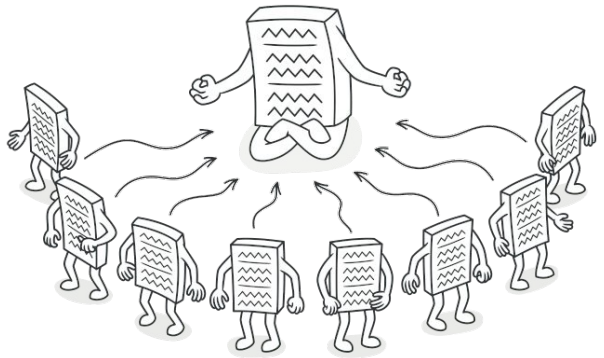
- Singleton design pattern based around collected and analyzing data from the drone sim
  - Data collected includes package pickup and dropoff times, distance packages travel, routing strategies used by each package, and the downtime for drones to pick up newly placed packages.
  - This data is then analyzed in order to determine the average speed, distance, and time of each different strategy used by the drone in delivering the different packages.
- The data is then outputted to a CSV file upon completion of the sim, or whenever the user presses the “Produce CSV” button to allow users to observe this data.

Why is this important:

- By collecting and analyzing this data, we are able to clearly measure the difference in performance between pathing strategies, and what changes may need to be made to the sim in order to refine package delivery.
    - For example, dfs search takes significantly longer than other search strategies used in the sim.
- Data collection also allows us to more easily determine faults in our system, as running numerical analysis on the data may allow us to detect issues with the sim that we may otherwise not catch.

What Design Pattern we used:

- For data collected, we decided to implement the **Singleton** design pattern
  - This is because we want all the data collected in a single point, but allow for global access to this data for this collection.
  - We also don’t want users manipulating the data, as this would render it useless. Singleton’s allow us to guarantee that won't happen, as only one instance of the singleton method can exist at a time
- By using singleton, we guarantee that our data is being collected in only one place, and is not being interfered with or overwritten.



Demonstration:

Deliveries:

a1: (522.0, 254.6) ---> (604.0, 255.6)

a2: (515.0, 261.0) ---> (399.0, 202.0)

a3: (677.0, 265.0) ---> (669.0, 169.0)

a4: (246.0, 302.0) ---> (486.0, 266.0)

b1: (657.0, 257.0) ---> (727.0, 257.0)

b2: (787.0, 177.0) ---> (734.0, 256.0)

b3: (243.0, 300.0) ---> (390.0, 236.0)

b4: (648.0, 389.0) ---> (747.0, 389.0)

d1: (418.0, 113.0) ---> (403.0, 266.0)

d2: (655.0, 190.0) ---> (654.0, 261.0)

d3: (610.0, 63.0) ---> (676.0, 65.0)

d4: (650.0, 257.0) ---> (564.0, 257.0)

q1: (158.0, 310.0) ---> (428.0, 257.0)

q2: (666.0, 178.0) ---> (746.0, 365.0)

q3: (837.0, 199.0) ---> (817.0, 447.0)

q4: (941.0, 292.0) ---> (409.0, 343.0)

analysis.csv

```
Raw Data
Packages, Pickup, Dropoff, Time, Strategy, Downtime, Distance
package 0, 143.436877 . 254.665000 . -103.175535, 390.891086 . 254.665000 . -100.301503, 36.061077, astar, 0.000000, 1081.832302,
package 1, 122.312714 . 254.665000 . -84.614088, -227.744459 . 254.665000 . -254.182023, 333.523455, astar, 8.967871, 10005.703651,
package 2, 611.185664 . 254.665000 . -73.117957, 587.043790 . 254.665000 . -349.025105, 266.656458, astar, 28.608234, 7999.693742,
package 3, -689.457800 . 254.665000 . 33.221256, 34.798421 . 254.665000 . -70.243924, 501.489743, astar, 44.416821, 15044.692296,
package 4, 550.830979 . 254.665000 . -96.110219, 762.072377 . 254.665000 . -96.110219, 97.207287, bfs, 17.222681, 2916.218604,
package 5, 943.136432 . 254.665000 . -326.032842, 783.196516 . 254.665000 . -98.984252, 360.091923, bfs, 9.755261, 10802.757693,
package 6, -698.511003 . 254.665000 . 27.473191, -254.904067 . 254.665000 . -156.464908, 449.105024, bfs, 49.569801, 13473.150732,
package 7, 523.671370 . 254.665000 . 283.262109, 822.427062 . 254.665000 . 283.262109, 244.243416, bfs, 29.805659, 7327.302472,
package 8, -170.407508 . 254.665000 . -509.970941, -215.673522 . 254.665000 . -70.243924, 17122.101707, dfs, 42.360085, 513663.051212,
package 9, 544.795510 . 254.665000 . -288.670416, 541.777776 . 254.665000 . -84.614088, 12063.971294, dfs, 26.373878, 361919.138812,
package 10, 408.997469 . 254.665000 . -653.672580, 608.167930 . 254.665000 . -647.924515, 4572.146475, dfs, 19.478141, 137164.394245,
package 11, 529.706839 . 254.665000 . -96.110219, 270.181693 . 254.665000 . -96.110219, 378.225215, dfs, 18.578816, 11346.756450,
package 12, -955.018415 . 254.665000 . 56.213518, -140.230166 . 254.665000 . -96.110219, 1960.532077, dijkstra, 41.154422, 58815.962318,
package 13, 577.990587 . 254.665000 . -323.158810, 819.409327 . 254.665000 . 214.285322, 633.022070, dijkstra, 25.108478, 18990.662091,
package 14, 1094.023145 . 254.665000 . -262.804121, 1033.668460 . 254.665000 . 449.956010, 1088.022501, dijkstra, 18.349299, 32640.675037,
package 15, 1407.867507 . 254.665000 . 4.480928, -197.567117 . 254.665000 . 151.056600, 1757.422309, dijkstra, 19.392810, 52722.669267,


Analyzed Data
Strategies, Average Speed, Average Distance, Average Time
Astar, 30.000000, 8532.980498, 284.432683
BFS,30.000000, 8629.857375, 287.661913
DFS,30.000000, 256023.335180, 8534.111173
Dijkstra,30.000000, 40792.492178, 1359.749739
Beeline,30.000000, 485.210652, 16.173688

```

Conclusion

- Singletons are very useful for data collection due to their global accessibility and security of stored data
- Data collection is a valuable addition to this sim as it allows for further analysis of the sims performance
- Further improvements that could be applied to this include gathering more data about average package location, most popular strategies used, and performance of specific drones