COMP2310

Assignment 1

Semester 2 2021

Tim James
u6947396

Table of Contents

1. Introduction

My solution utilizes a spherical formation and assigns a vehicle as a coordinator to effectively synchronize swarm behaviour and prioritize giving vehicles with the lowest charge access to a Globe. Provided a globe is located and there are no stray vehicles that never make contact with the rest of the swarm, my solution is able to keep all vehicles alive reliably in stages 2 and 3.

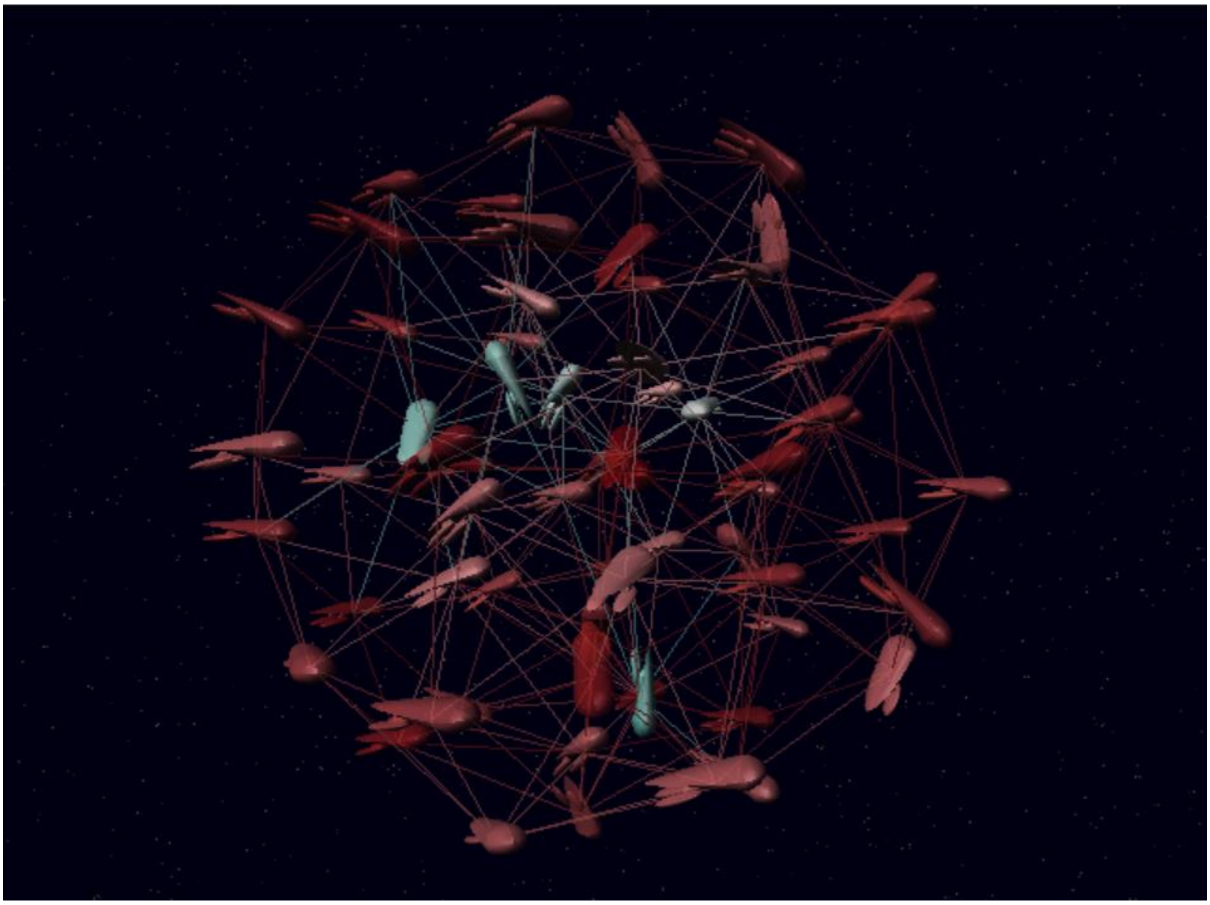

Figure 1: Sphere formation during stage 2

2. Design

2.1 Overview

Vehicles have different behaviour depending on an internal state.

`Searching` (initial):

- **Message sending**:
    - Search for the nearest Globe, if one is found set `Target_Globe_Pos` and they become a `Coordinator` and `Broadcast_Globe_Pos`.
- **Message receiving**:
    - Accept `Broadcast_Globe_Pos` messages and attach themselves to the `Coordinator` which sent the message. State becomes `Waiting_For_Turn`.
- **Movement**:
    - Default swarm behaviour.

`Coordinator`:

- **Message sending**:
    - Search for the nearest Globe and update local `Target_Globe_Pos`, `Broadcast_Globe_Pos` to vehicles `Waiting_For_Turn` on `Coordinator`.
    - Find the vehicle in local `Waiting_Vehicles` map (key - vehicle ID, value - charge) with the lowest charge and send that vehicle a `Release` message targeted at that vehicle's ID. Enforce a delay between releases (implemented as a number of iterations of `Wait_For_Next_Physics_Update` loop) such that not too many vehicles are released at once and collide.
    - *(Note: unused in final solution)* If charge is too low, send `Transfer_Coordinator` to vehicle with highest charge in `Waiting_Vehicles` map. State `Approaching_Globe` Globe and restore charge.
- **Message receiving**:
    - `Notify_Of_Charge` messages if target, add/update `Waiting_Vehicles` value corresponding to sender ID.
    - (*Note: unused in final solution*) `Transfer_Coordinator`, accept if target. This is a case of multiple `Coordinators`, so combine `Waiting_Vehicles` map from message with local and all vehicles `Waiting_For_Turn` on the previous `Coordinator` now wait on this existing `Coordinator`.
- **Movement**:
    - Maintain position close to `Target_Globe_Pos` in the centre of the overall swarm formation.

`Waiting_For_Turn`:

- **Message sending**:
    - Send `Notify_Of_Charge` periodically to `Coordinator` target.
- **Message receiving**:
    - `Broadcast_Globe_Pos` messages if send by `Coordinator`. Updates local `Target_Globe_Pos` and forward to other vehicles if `Forward_Count` < `Max_Forwards`.
    - Accept `Release` message, if target then change state to `Approaching_Globe`. If not the target, forward to other vehicles if `Forward_Count` < `Max_Forwards`.
    - *(Note: unused in final solution)* `Transfer_Coordinator`, if target then take `Waiting_Vehicles` map from message and store locally, and change state to be

Coordinator. All vehicles Waiting_For_Turn on the previous Coordinator now wait on this new Coordinator.

- **Movement**:
    - ○ Maintain set distance away from Globe (throttle 0 while at set distance), forming a sphere around the Globe with the swarm.

Approaching_Globe:

- **Message receiving**:
    - ○ Receive and forward messages similar to Waiting_For_Turn.
- **Movement**:
    - ○ Set max throttle and set destination to Target_Globe_Pos.
    - ○ Once Globe is reached and charge restored, return to Waiting_For_Turn on Coordinator.

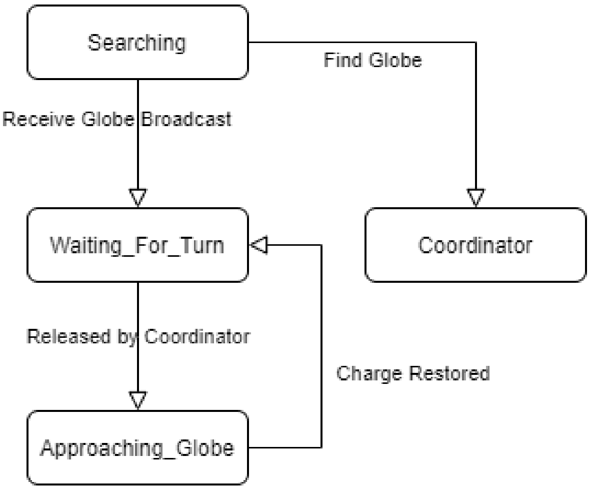

Figure 2: Simplified state transitions

2.2 Process

My initial design for stage 2 was to simply have vehicles broadcast the Globe position and their charge and then approach if they have the lowest charge from those received. However, this lack of synchronization lead to chaos with vehicles colliding and getting stuck, and others straying too far from the Globe. Without a central point to coordinate behaviour, vehicles would operate with incomplete information.

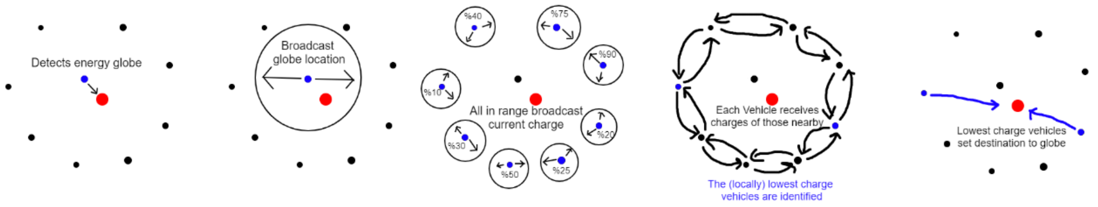
Detects energy globe
Broadcast globe location
%40
%75
%90
%10
All in range broadcast current charge
%20
%30
%50
%25
Each Vehicle receives charges of those nearby
The (locally) lowest charge vehicles are identified
Lowest charge vehicles set destination to globe

Figure 3: Early design concept

For this reason, I assigned different 'roles' or states to vehicles. The first vehicle to find a Globe would become a central Coordinator with other vehicles waiting on a Coordinator for their turn to approach the Globe. See the section 2.1 for details on how vehicles with different states interact.

With the swarm being allowed to roam free, ensuring that vehicles could receive messages from the Coordinator and travel to the Globe in a predictable way was problematic. For this reason, I ensured the vehicles maintained a strict formation. The Coordinator would move within close proximity of the Globe and perform the initial handshakes with other vehicles there. Once a vehicle was in the waiting state, it would attempt to maintain a certain distance away from the Globe. This was achieved by setting Throttle to 0 when the distance to the Globe was below a threshold, but setting throttle and destination to the Globe otherwise. The Coordinator would periodically check the position of the Globe and broadcast the updated position to the formation. This served to keep all vehicles close to the Globe as it moved around, so that vehicles did not need to make a substantial journey when released and the swarm would not spread out. Naturally, the sphere formation helped to keep the Coordinator within sensing range of as many vehicles as possible. Though if the sphere were too small with a large number of vehicles, there would be frequent collisions, so it was not possible to have all vehicles within sensing distance of the Coordinator. This had the added benefit of using less charge, as vehicles would only move when needed rather than swarming without a purpose.

While my initial design anticipated that there would be multiple Coordinators and the Coordinator state would need to be transferred when Coordinators themselves run low, I did not need to implement this with the new sphere formation approach. With the Coordinator being in the centre and maintaining a close distance from the Globe, it will constantly be replenishing its own charge. I initially thought that this would cause collisions with released vehicles, so the Coordinator would need to temporarily move out of the way. However, by setting throttle to max vehicles are able to make contact with the Globe fast enough, provided there are no other vehicles released simultaneously.

In the case of multiple Globes (stage 3), it is possible that several Vehicles will find Globes and become Coordinators initially. However, my approach is compatible with this, as messages contain a target. So, any waiting vehicles will be associated with a specific Coordinator and only respond to messages from that Coordinator. Multiple groups following different Globes can coexist effectively, or a single group that ignores other Globes, as when vehicles search for Globes they use the nearest.

One major issue with my initial implementation was that the Coordinator would attempt to release vehicles from waiting, but these messages would not necessarily be received by the target vehicle. This would occur if the Coordinator was not within sensing distance of that particular vehicle. To resolve this, I attempted several potential solutions.

First, I tried having vehicles that receive a release message that are not the target forward the message on to more vehicles. This would propagate messages across the swarm. So, if target A and vehicle B were within sensor range, and so were vehicle B and the Coordinator, then vehicle B could forward the Coordinator's message to A. Though this posed its own set of challenges. The forwarded message needed to have a limit on how many times it could be forwarded, otherwise it would simply continue being sent across the swarm and take resources long after the target had received the message. This could also fill up vehicles' mailboxes (messages on `Messages_Waiting`) with potentially irrelevant messages, and though mailbox size could be limited this meant that more urgent messages may be lost. Additionally, if the target received a release message, then travelled to the Globe and returned to waiting, then if the message was still being propagated the original target may be released more than once.

Another option is for the Coordinator to continue attempting to release the target vehicle until it receives an acknowledgement message in return, essentially a handshake. However, it was possible that the release message would be received but the acknowledgement itself could not, so the Coordinator would continue trying to release the target without realising it had already been successful. A limit to the number of times the Coordinator would try to contact a vehicle helped mitigate this somewhat, but ultimately this would slow down the Coordinator too far to be worthwhile.

Building on what I learnt from implementing the previous two potential solutions, I limited the number of times a message could be forwarded to a low value. I also made waiting vehicles periodically send the charge status to the Coordinator. If the Coordinator sent a release message that was never received, it would continue uninterrupted regardless, but would likely receive the updated charge from the missed vehicle after a short duration. This drastically improved the chance that low charge vehicles would be released, without impeding the Coordinator. The frequent charge status updates also had the added benefit of keeping the Coordinator updated, and making it was less likely to miss receiving charge information after the initial handshake. Note that the messaging forwarding method was also used for other messages, such as globe position broadcasts, which was instrumental in initially creating the initial sphere formation and catching stray vehicles.

Even with these improvements, there were still cases where the vehicle release messages were not received in time. To remedy this, waiting vehicles were given a 'last resort', where they would release themselves if charge got critically low. Naturally, this was a risky move that had potential for collision. Yet the risk of slowing down the swarm did not outweigh that of losing a vehicle. I experimented with having critically low charge vehicles to perform a handshake with the Coordinator first for better synchronisation, though this proved to take too long for such a critical moment.

2.3 Testing

Testing involved having vehicles print information about their actions to console, then taking observations for insight into how the solution could be improved. See the appendices 4.1 and 4.2 for example console output. Visual observations, and the duration of surviving vehicles was also utilized to assess effectiveness. Notable observations that were made and addressed are discussed in section 2.2. Here are some additional observations for potential issues:

- Even with message forwarding implemented, the Coordinator still attempts to send Release messages that are not received and acted upon by the target.
- It is possible (though highly rare) for two Coordinators to exist on the same Globe. In this case, the swarm will be split between these Coordinators. Vehicles will be synchronized as usual, though there is an increased chance of collisions and overall less stability.
- Sometimes a vehicle will be released multiple times in a row, even with high charge. Likely due to remaining forwarded messages still propagating through the swarm.
- After being released and travelling past the Globe, vehicles will temporarily have increased distance to the Globe. Likely due to max throttle.

3. Design Evaluation

My solution is effective, and typically able to keep all Globes alive up to stage 3, but takes the approach of never assuming that messages are received. For example, the Coordinator does not check to see if released vehicles actually reach their destination, but is flexible enough to continue operation regardless. This makes messaging faster overall, but additional countermeasures needed to be introduced which increases overall complexity. As for maintaining the code itself, operations are split into message sending, receiving and movement, with case statements for states delivering specific behaviour where needed.

The sphere formation behaviour is optimized for the default 64 vehicles, so scaling to a larger swarm would require a greater diameter due to collisions. Potentially, the Coordinator could estimate a diameter based on how many vehicles are waiting on it, though this would mean the swarm may not be in range of the swarm after a certain point. Additionally, the Coordinator may be overwhelmed with messages at large swarm sizes.

The solution is able to handle disappearing vehicles effectively, as the Coordinator will not wait for an acknowledgement after releasing vehicles. If the Coordinator itself is terminated without transferring its state, however, the swarm will be stuck waiting for release messages that will never be sent. A potential solution could be to reset vehicles to searching if they do not receive messages from their Coordinator after a delay.

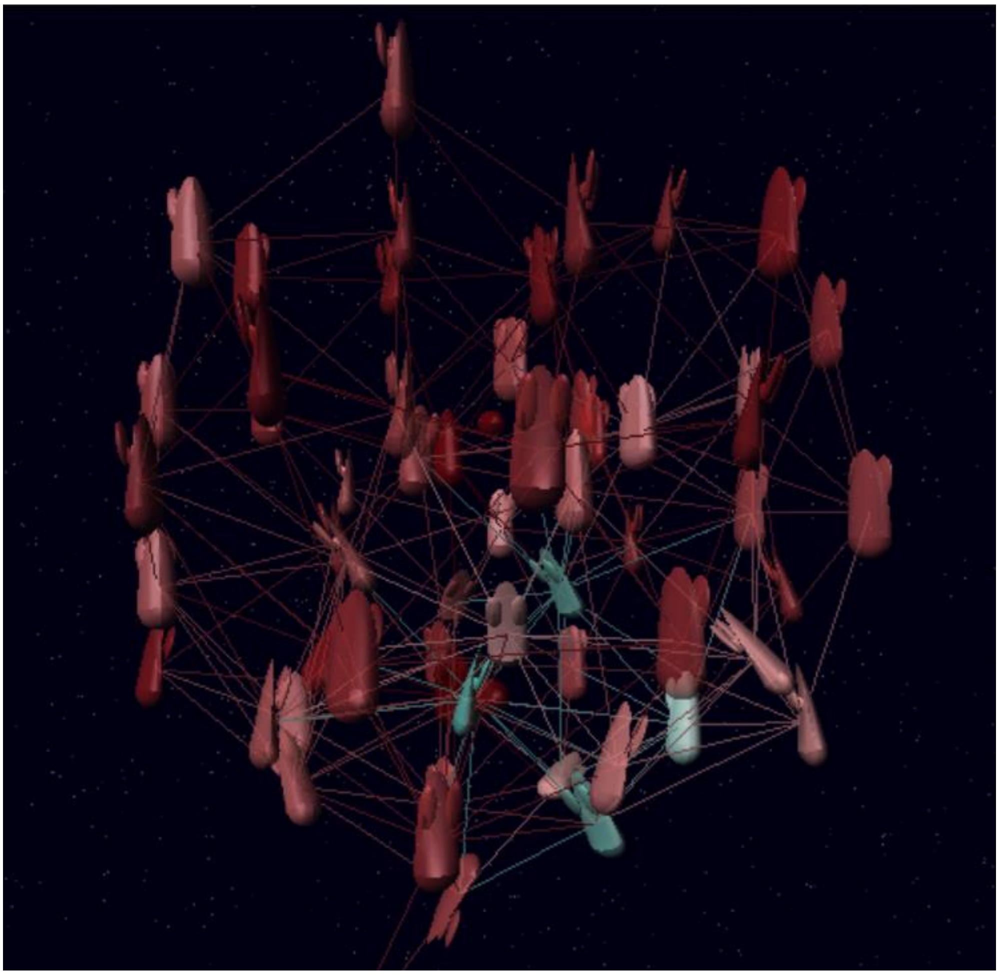

Figure 4: Swarm in stage 3 (multiple Globes)

4. Appendices

4.1 Sample Initial Output

```
 45 found Globe, became Coordinator
 45 Coordinator got charge info from  20
 45 Coordinator got charge info from  22
 45 Coordinator got charge info from  26
 45 Coordinator got charge info from  35
 45 Coordinator got charge info from  36
 45 Coordinator got charge info from  41
 45 Coordinator got charge info from  53
 45 Coordinator got charge info from  56
 45 Coordinator got charge info from  57
 45 Coordinator got charge info from  60
 45 Coordinator got charge info from  9
 45 Coordinator attempted to release  56
 45 Coordinator attempted to release  35
 45 Coordinator attempted to release  60
 45 Coordinator got charge info from  35
```

4.2 Sample Release Output

```
 51 Coordinator got charge info from  16
 51 Coordinator got charge info from  58
 51 Coordinator got charge info from  59
 51 Coordinator attempted to release  31
 62 finished travelling to Globe
 51 Coordinator got charge info from  62
 31 received release and travelling to Globe from Coordinator  51
 51 Coordinator attempted to release  58
 58 received release and travelling to Globe from Coordinator  51
 31 finished travelling to Globe
 51 Coordinator attempted to release  64
 64 received release and travelling to Globe from Coordinator  51
 51 Coordinator got charge info from  30
 51 Coordinator got charge info from  18
 51 Coordinator got charge info from  56
 51 Coordinator got charge info from  63
 58 finished travelling to Globe
 51 Coordinator got charge info from  58
```