Assignment 1 report

Name: Wu Zefan Uid: u6474528 Date: 27 September 2019

Introduction

The swarm will wonder around with default behaviors in the three-dimensional space and the members of the swarm have to charge before they run out of the fuel and disappear. The goal is to make sure all of the members survive as long as possible. In this case, I use three attempts to achieve the goal from stage a to stage d.

Implementation (with many attempts)

Attempt 1:

Finding the energy globe by using the offered function, which will return an array of energy globes' information if one of the vehicles is close to one or more energy globes. In this attempt, the energy globe's position is declared as global variable and every vehicle can access it whenever they need it. Then the program enables them to charge when their current charge level is below at 30%. And let them fly away based on the coordinate of the energy globe plus (1.0, 1.0, 1.0) after charging, because they need to let the rest of vehicles have the space to charge.

- **Problem 1:** As shown in the *figure a*, the vehicles that need to be charged cannot find the correct coordinate of the energy globe because their message is the order coordinate of the globe and the globe is constantly moving.
- **Solution 1:** Decreasing the distance of flying away from the energy globe from (1.0, 1.0, 1.0) to (0.1, 0.1, 0.1) when the vehicles finish their charging phase will solve this problem, because it will make the vehicles floating around the globe in a near instance so that the globe's position can be updated in a swift way (*figure b*).

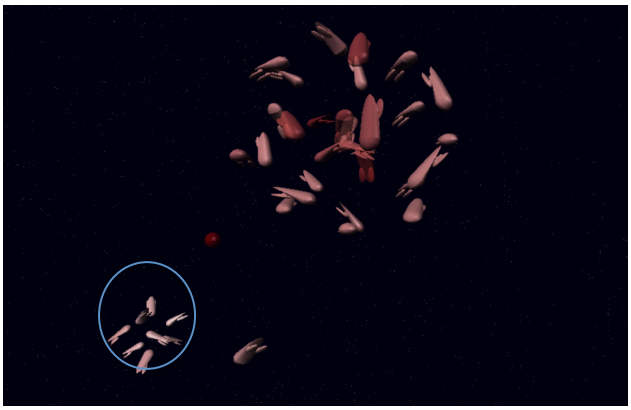

figure a

figure b

- **Performance of attempt 1 (single globe in orbit average of three tests):**

No\Time	1 min	2 min	3 min	4 min	5 min
64	64	64	64	64	64
128	128	127	127	126	126
192	192	191	188	187	184

It shows that this attempt cannot handle large size of vehicles because there are so many vehicles waiting for changing and disappearing on their way according to the performance table and the animation.

- **Problem raising after attempt 1:**There is no communication between the vehicles, using shared memory will stuck at stage a. Also, the large size of swarm will lead to traffic jam because the vehicles will rush to the energy globe together and some of them will starve to death.

Attempt 2 (solve questions left from attempt 1):

To step into stage b, the vehicles need to be fully distributed and only depend on the inner message passing and receiving. In this case, I let them follow the pattern: Receive the message -> Do something according to the message -> Send the message. In this way, every vehicle can receive the message and pass the message to others. To avoid them rushing together, I use the sphere equation $x = rsin(\alpha)cos(\beta), y = rsin(\alpha)cos(\beta), z = rcos(\alpha)$ to let them follow the sphere pattern (*figure c*) so that they will keep moving and let others have the chance to charge. Also, inner message includes the message whether the energy globe is available or not, which will reduce the simultaneous charging number because once they need to charge they will send the message to inform the nearby vehicles until they finish charging.

- **Problem 2:** This pattern is not stable and it will change from *figure c* to *figure d* because of the different moving directions of the energy globe. During the transition of two forms, some vehicles will die.

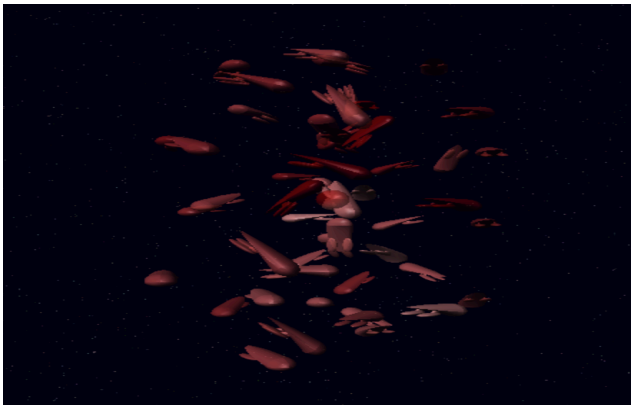

figure c

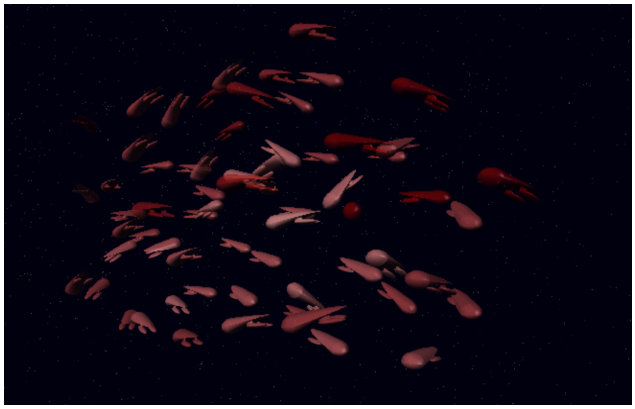

figure d

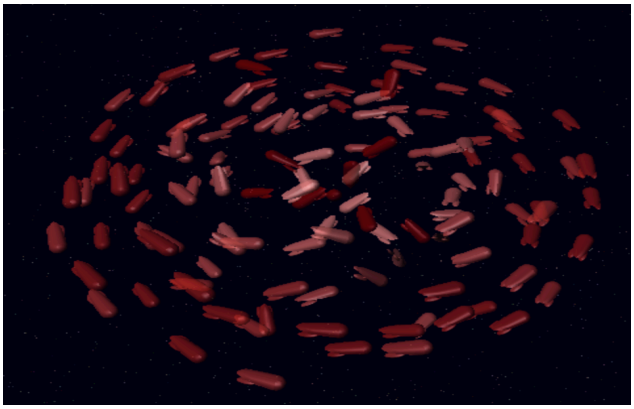

figure e

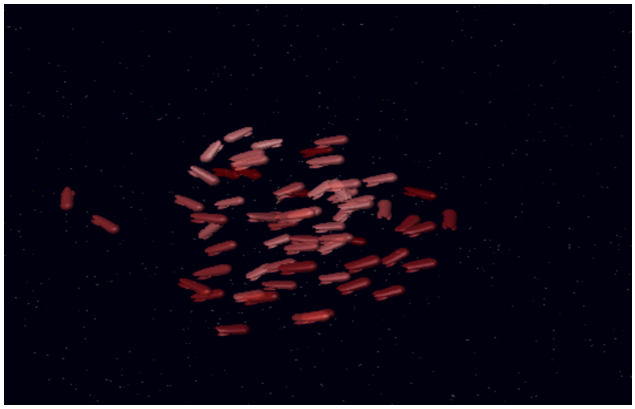

figure f

- **Solution 2:** I change the pattern from sphere to a simple circle using circle's parametric form $x = a + rcost(t), y = b + rsin(t)$ to make sure that the pattern vehicles followed is stable. And I use the current charge as the parameter for the circle's radius so that the lower charge level will be the inside. Also, I create 16 groups for the vehicles with different initial angle so that vehicles will form a perfect circle (*figure e*) instead of a whole group's motion likes a circle (*figure f*). To do that will make sure the vehicles can be charged in an even way and less traffic jam.
- **Performance of attempt 2 (single globe in orbit average of three tests):**

No\Time	1 min	2 min	3 min	4 min	5 min
64	64	64	64	64	64
128	126	125	124	124	124
192	152	145	142	139	135

The animation and the result of performance show that this attempt is suitable for small size of the swarm. There are still many vehicles disappearing during the first round of charging if the size of the swarm is large.

- **Problem raising after attempt 2:** This pattern is not suitable for large size of the swarm because the vehicles on the circle can only explore the energy globe in a two-dimensional space instead of three dimensions, which will cause some of the vehicles are squeezed out of the pattern then they will lose the communication with other vehicles and die. Also, the vehicles will lose the information of the energy globe if it shifts its position on the y-axis.

Attempt 3 (final attempt):

It seems that I need to change the pattern to make sure that the vehicles are spread as wide as possible so that their explore area can cover every possible directions. Here, I use the

design (*figure g* & *h*) to achieve that goal. The outer four circles are for those vehicles whose current charge levels are larger than 70% and the inner circle is for those current charge levels below 70%. Also, there is an emergency case for those current charge levels below 30%, which means that they can direct set the destination to the position of the energy globe and inform other vehicles instead of waiting until the energy globe is available. By the way, the four outer circles are the same clockwise directions to make the whole motion smoother and without collision.

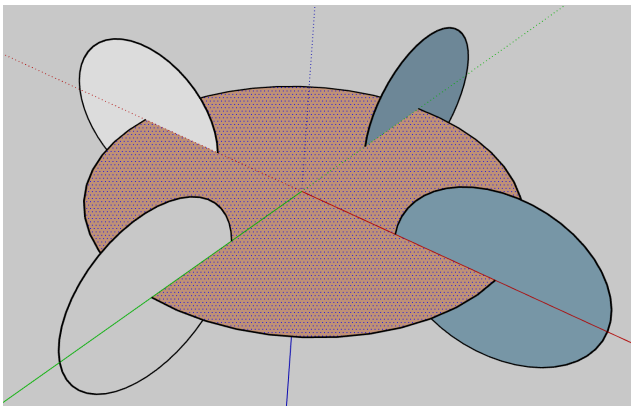

figure g

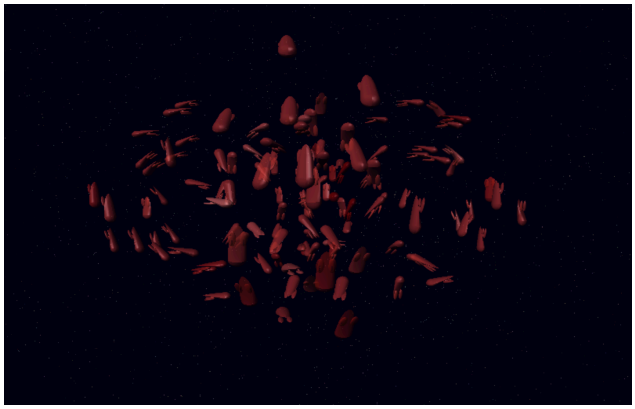

figure h

- **Problem 3:** They still get the old message when I try to run the test of the random energy globes in orbits since I notice that some of the vehicles still waiting in a spot without any energy globes. In a random energy globes circumstance, it not only requires quick reaction rate but also the correct message.
- **Solution 3:** The reason behind this is that the messages are not filtered and some of the old messages will be kept forever inside of the system. In such a situation, I have to implement timestamps for the messages so that the vehicles will check the message's time first then take action based on it. Also, they will upload the local message if the receiving message's time is less than local time to make sure every one has the ability to filter the message.
- **Problem 4:** how does the swarm shrink the size into the target number?
- **Solution 4:** For shrinking the size, I use the set to store the vehicles' number. The vehicles will try to insert their vehicles' number into the set when the length of the set is less than the target number. Also, the vehicles have to agree with that the set's content is the same so that they will die until reaching the target number. In order to achieve that, I take advantages of the timestamps so that once it reaches the target number it will become the newest message and spread to other vehicles. Then every vehicle will check

whether their vehicle number in the set. If it is in the set follow the pattern mentioned before, otherwise, waiting until death.

- **Performance of the final attempt (average of three tests):**

Single_Globe_In_Orbit:

No\Time	1 min	2 min	3 min	4 min	5 min
64	64	64	64	64	64
128	128	124	124	124	124
192	192	190	190	190	190

Random_Globes_In_Orbits：

No\Time	1 min	2 min	3 min	4 min	5 min
64	62	62	62	62	62
128	128	127	126	126	126
192	191	189	189	189	189

Single_Globe_In_Orbit (shrinking size to target number):

From\To	42	84	126
64	59.55 s	\	\
128	1 min 10 s	1 min 53 s	11 min 37 s
192	1 min 28 s (1 more dead)	2 min 24 s (2 more dead)	4 min 28 s (3 more dead)

Random_Globes_In_Orbits (shrinking size to target number):

From \ To	42	84	126
64	1 min 36 s	\	\
128	3 min 42 s	1 min 51 s	9 min 27 s
192	2 min 08 s	2 min 23 s	3 min 28 s

The result and animation shows that this model can handle different size of the swarm. However, the more vehicles they are, the slower motion and it will influence the performance. Also, when the swarm shrinks its size to a specific number, it is slow when the target number is close to the initial number of the swarm because the set counts them from 0 to the target number and the verification of the set depends on the message dissemination velocity.

- **Concerns rising after final attempt:**

(1) I try to make my design suitable for all stages. But for stage d, I have to use set to check the survival vehicles' number. However, when I run stage b or c this part of codes seems like a burden and will make them loss the pattern since some of the

vehicles will die out of control from time to time even though I let them follow the pattern. The reason I think is that it will take time to loop through all the messages in the buffer and they will lose control during this period. And this will decrease the performance rapidly. So I have no choice to test the different stages with two different versions even only a small part of the codes is different.

(2) Unfortunately, set is a dynamic data structure, which means all the vehicles will get access to the memory address of the set. I don not have enough time to implement in the form of array after I realized that, which can be regarded as a concern and be improved in the future.

(3) The performance is relative to many factors such as the FPS, the radius of the pattern, and the threshold when to charge and so on. A better way to deal with these constraints is to use numeric tests or even AI training system to get the best results.

Reflection

Through these attempts, I realize that the important things are the accuracy of the message and preventing traffic jams. For the accuracy part I use the timestamps to make sure each vehicle receiving the newest message. As for dealing with traffic jams, I use different charge levels with different behaviors and also use inner message (a flag) to restrict the charging number.

Conclusion

The way to stage d, I use three different attempts and each attempt is based on previous one to solve some specific problems. By implementing a certain pattern for the swarm, I manage to make them live as long as possible in the three-dimensional space, but there are still some concerns needed to be solved.