CS 286 - Homework 1, Nerd-Herd Part A

Adolfo Roquero, Leticia Schettino, Kobi Greene

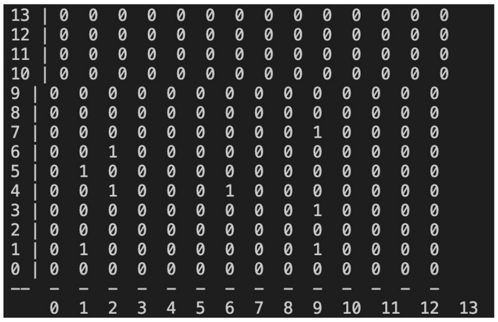
February 2022

1 Write up

aggregate

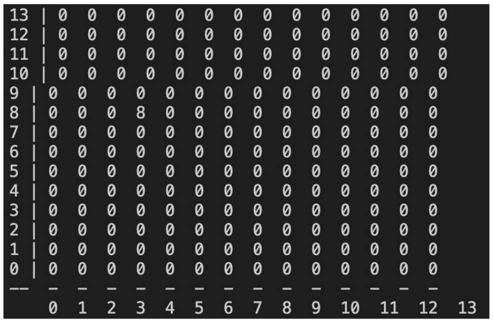
For aggregate, our robots move together to the location specified in the function call. We continue moving our robots until they are all at the same specified location. This is implemented by looping through all the robots that are not already in the location and calling the _move_towards_step function on them.

Here, is an example of running the function **aggregate(3,8)** in the context of the call to the flock function.



```
13 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
12 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
11 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
10 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
9 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
8 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
7 | 0  0  0  0  0  0  0  0  0  1  0  0  0  0
6 | 0  0  1  0  0  0  0  0  0  0  0  0  0  0
5 | 0  1  0  0  0  0  0  0  0  0  0  0  0  0
4 | 0  0  1  0  0  0  1  0  0  0  0  0  0  0
3 | 0  0  0  0  0  0  0  0  0  1  0  0  0  0
2 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
1 | 0  1  0  0  0  0  0  0  0  1  0  0  0  0
0 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
--  -  -  -  -  -  -  -  -  -  -  -  -  -  -
    0  1  2  3  4  5  6  7  8  9  10  11  12  13
```

Figure 1 – Initial grid with 8 robots



```
13 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
12 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
11 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
10 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
9 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
8 | 0  0  0  8  0  0  0  0  0  0  0  0  0  0
7 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
6 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
5 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
4 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
3 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
2 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
1 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
--  -  -  -  -  -  -  -  -  -  -  -  -  -  -
    0  1  2  3  4  5  6  7  8  9  10  11  12  13
```

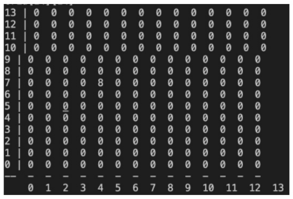
Figure 2 – Updated grid after call to aggregate,
All 8 robots at position (3,8)

safe_wander

For safe wander, we have our robots who are in the same flock move together to a random location in the matrix. If the flock flag is set to True, a target location is randomly generated for every flock and every robot in the flock moves towards the location.
If the flock flag is set to False, robots move independently (not as a flock) and

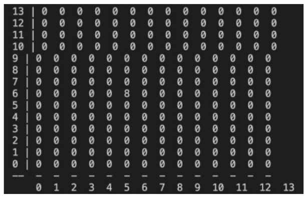
each robot moves towards a random location.

Here is an example of the execution of safe_wander in the context of the call to flock.
In particular, safe_wander is called with the flock flag set to True, after all of the robots have been aggregated into one location and have formed a single flock.



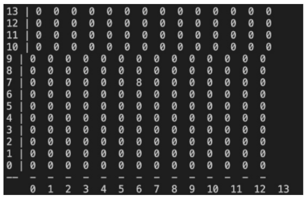
```
13 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
12 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
11 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
10 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
9 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
8 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
7 | 0  0  0  0  8  0  0  0  0  0  0  0  0  0
6 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
5 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
4 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
3 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
2 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
1 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
--  -  -  -  -  -  -  -  -  -  -  -  -  -  -
    0  1  2  3  4  5  6  7  8  9  10  11  12  13
```

Figure 3a – all robots at (4,7)
Safe wander 0



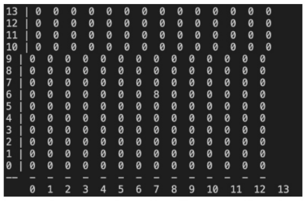
```
13 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
12 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
11 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
10 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
9 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
8 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
7 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
6 | 0  0  0  0  0  8  0  0  0  0  0  0  0  0
5 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
4 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
3 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
2 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
1 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
--  -  -  -  -  -  -  -  -  -  -  -  -  -  -
    0  1  2  3  4  5  6  7  8  9  10  11  12  13
```

Figure 3b – all robots at (5,6)
Safe wander 1



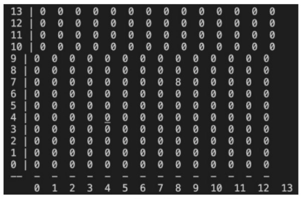
```
13 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
12 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
11 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
10 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
9 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
8 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
7 | 0  0  0  0  0  0  8  0  0  0  0  0  0  0
6 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
5 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
4 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
3 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
2 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
1 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
--  -  -  -  -  -  -  -  -  -  -  -  -  -  -
    0  1  2  3  4  5  6  7  8  9  10  11  12  13
```

Figure 3c – all robots at (6,7)
Safe wander 2



```
13 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
12 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
11 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
10 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
9 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
8 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
7 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
6 | 0  0  0  0  0  0  0  8  0  0  0  0  0  0
5 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
4 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
3 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
2 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
1 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
--  -  -  -  -  -  -  -  -  -  -  -  -  -  -
    0  1  2  3  4  5  6  7  8  9  10  11  12  13
```

Figure 3d – all robots at (7,6)
Safe wander 3

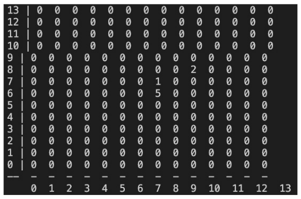


```
13 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
12 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
11 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
10 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
9 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
8 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
7 | 0  0  0  0  0  0  0  0  8  0  0  0  0  0
6 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
5 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
4 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
3 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
2 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
1 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
--  -  -  -  -  -  -  -  -  -  -  -  -  -  -
    0  1  2  3  4  5  6  7  8  9  10  11  12  13
```

Figure 3e – all robots at (8,7)
Safe wander 4

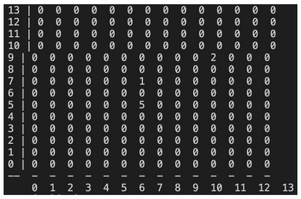
disperse

For disperse, we loop through each flock to get the proper centroid for each one. Then we use our centroids and have each robot move away in random direction for one step. Each robot will do 3 steps of moving away from the centroid.



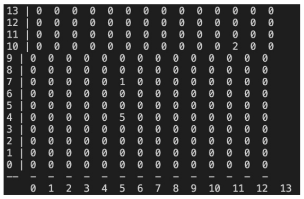
```
13 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
12 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
11 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
10 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
9 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
8 | 0  0  0  0  0  0  0  0  0  2  0  0  0  0
7 | 0  0  0  0  0  0  0  1  0  0  0  0  0  0
6 | 0  0  0  0  0  0  0  5  0  0  0  0  0  0
5 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
4 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
3 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
2 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
1 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
--  -  -  -  -  -  -  -  -  -  -  -  -  -  -
    0  1  2  3  4  5  6  7  8  9  10  11  12  13
```

Figure 4a – disperse from (8,7)
Disperse 0



```
13 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
12 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
11 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
10 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
9 | 0  0  0  0  0  0  0  0  0  0  2  0  0  0
8 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
7 | 0  0  0  0  0  0  1  0  0  0  0  0  0  0
6 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
5 | 0  0  0  0  0  0  5  0  0  0  0  0  0  0
4 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
3 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
2 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
1 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
--  -  -  -  -  -  -  -  -  -  -  -  -  -  -
    0  1  2  3  4  5  6  7  8  9  10  11  12  13
```

Figure 4b – disperse from (8,7)
Disperse 1



```
13 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
12 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
11 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
10 | 0  0  0  0  0  0  0  0  0  0  0  2  0  0
9 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
8 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
7 | 0  0  0  0  0  1  0  0  0  0  0  0  0  0
6 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
5 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
4 | 0  0  0  0  0  5  0  0  0  0  0  0  0  0
3 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
2 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
1 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0
--  -  -  -  -  -  -  -  -  -  -  -  -  -  -
    0  1  2  3  4  5  6  7  8  9  10  11  12  13
```

Figure 4c – disperse from (8,7)
Disperse 2

flock

For flock, we put all our previous methods together. First, we aggregate our robots to a given location. Then, we call safe_wander for however many steps specified in the call to flock. Finally, we disperse our robots. In between each function call we wait 3 seconds to properly observe how our robots operate.