

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | Multi-Agent Path Finding (MAPF) Individual Project | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Utsav Anantbhat 301446421 |  | | | |
|  | | | | 10/30/22—CMPT 417—Professor Hang Ma |  | | | |
|  | | |  | | |  | | |

**Introduction**

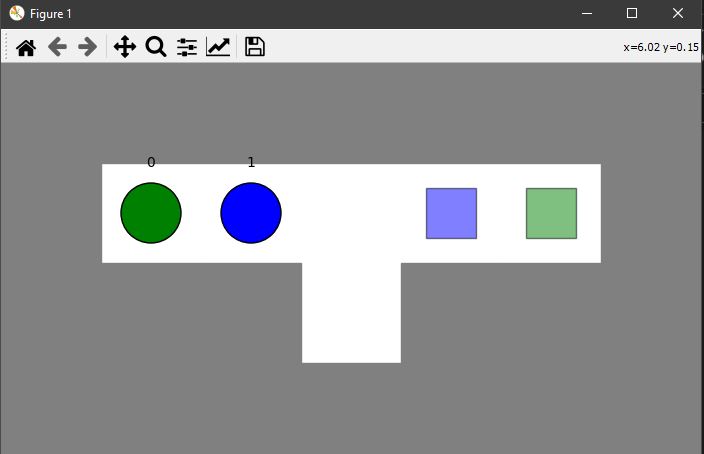
The purpose of this project is to demonstrate our understanding of Multi-Agent Path Finding (MAPF) by implementing a space-time A\* algorithm and some MAPF solvers, those being Conflict-Based Search (CBS) and CBS with disjoint splitting.

**Overview**

Multi-Agent Path Finding (MAPF) is defined as the problem of finding paths for multiple agents such that they reach their respective goal states without any collisions. MAPF has several applications from programs in automotives to warehouse management by robots (such as in Amazon warehouses).

**Task 1: Implementing Space-Time A\***

**1.1: Searching in the Space-Time Domain**



1: Start State

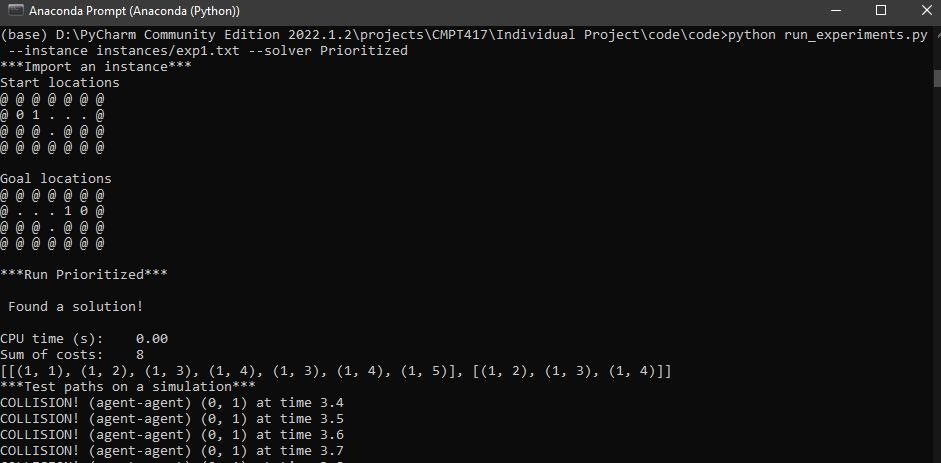


2: 0 collides with 1 in attempt to get to its goal

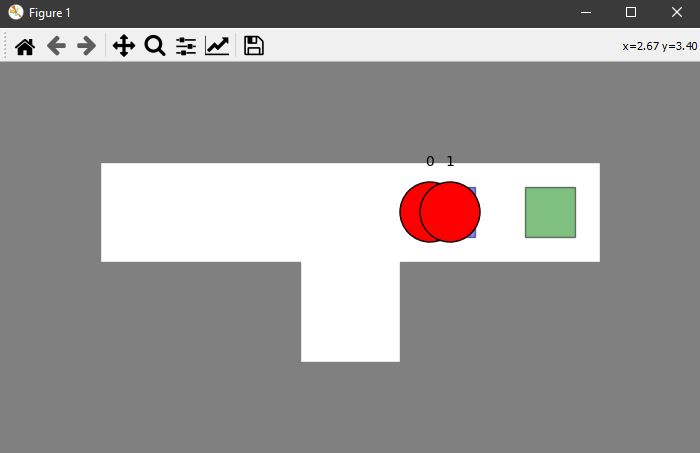


3: Both agents reach their goals

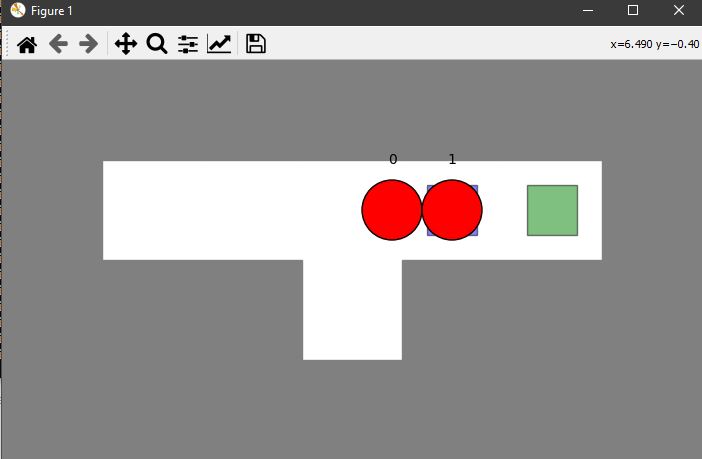
**1.2: Handling Vertex Constraints**



1: Terminal Output



2: Collision



3: Vertex Condition – 0 waits at timestep 4 before going to its goal

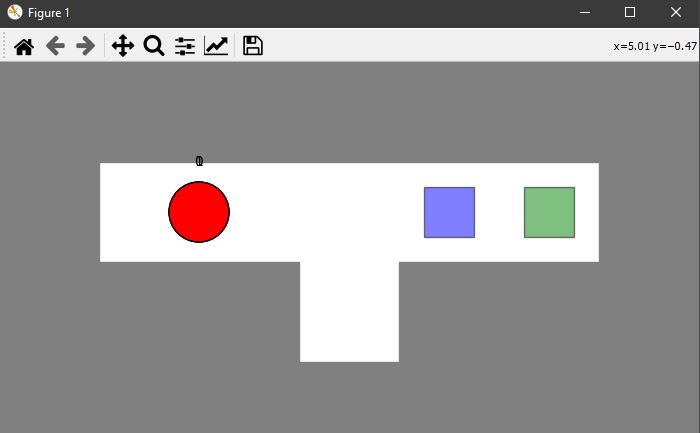


4: 0 goes to goal after the wait at timestep 4

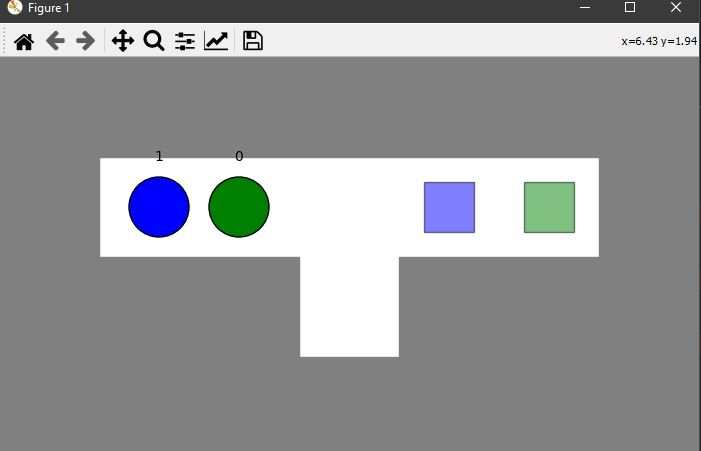
**1.3: Adding Edge Constraints**



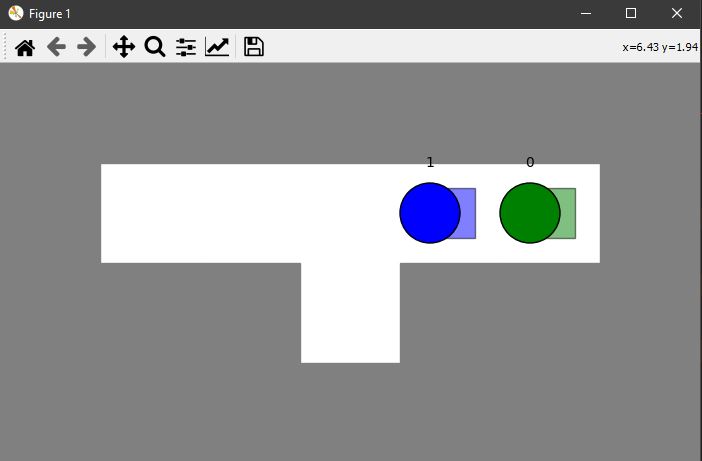
1: Terminal Output



2: Collision at timestep 1; agent 1 can’t go from vertex (1,2) to (1,3) at timestep 1

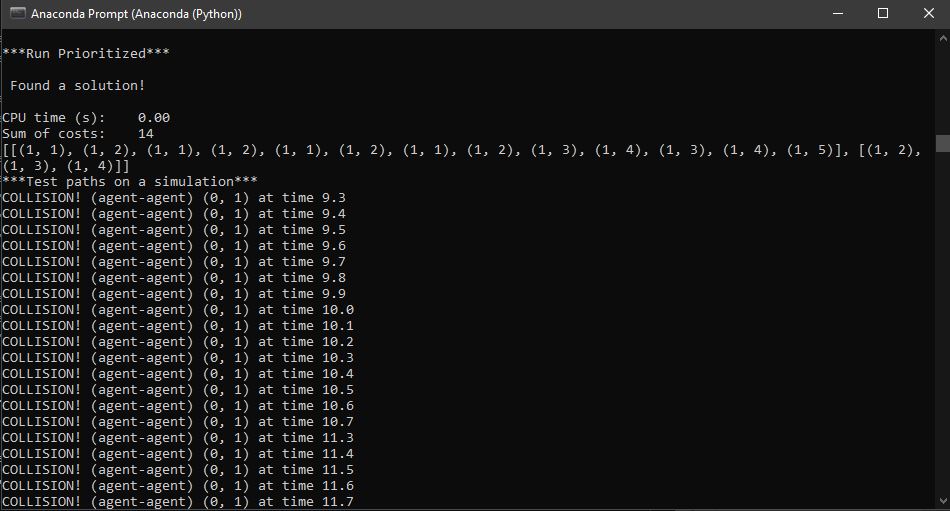


3: Timestep 2 – 1 and 0 swap places

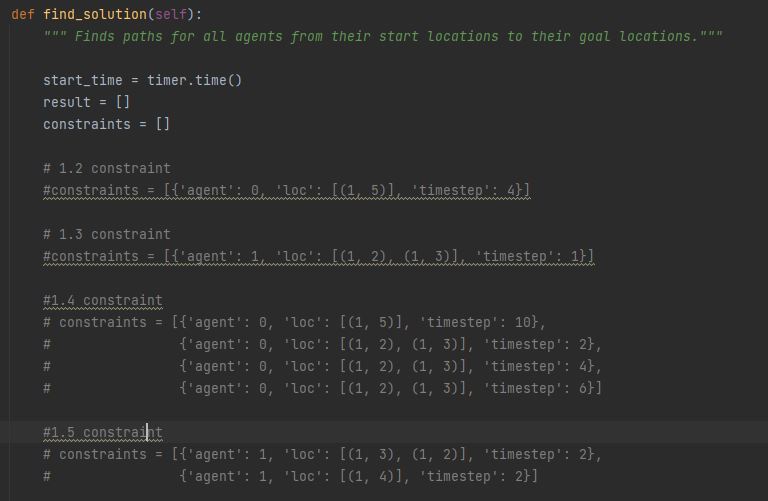


4: Both reach their goal state

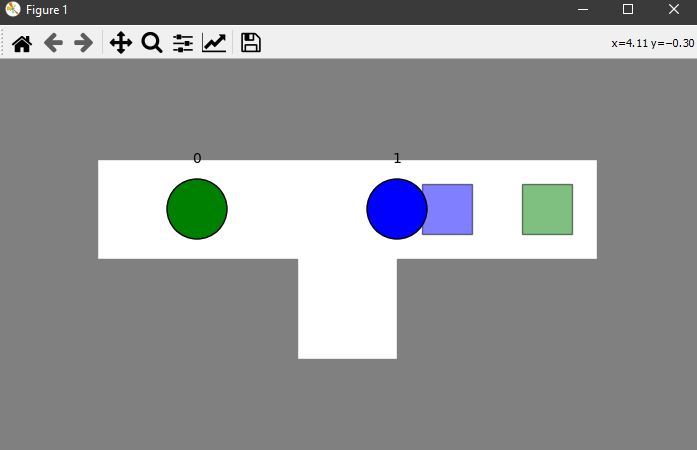
**1.4: Handling Goal Constraint**



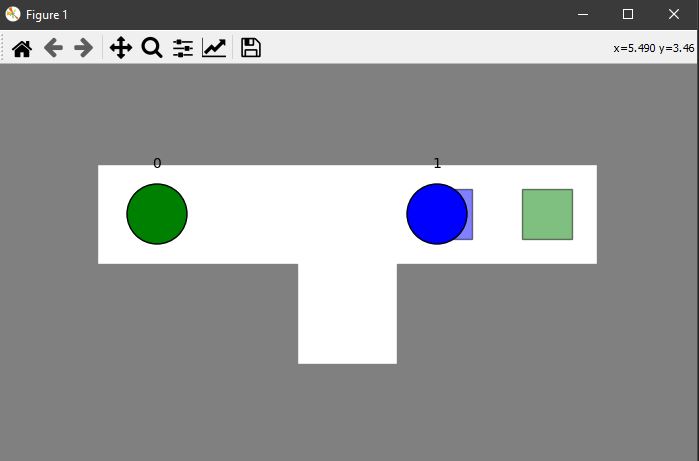
1: Terminal output

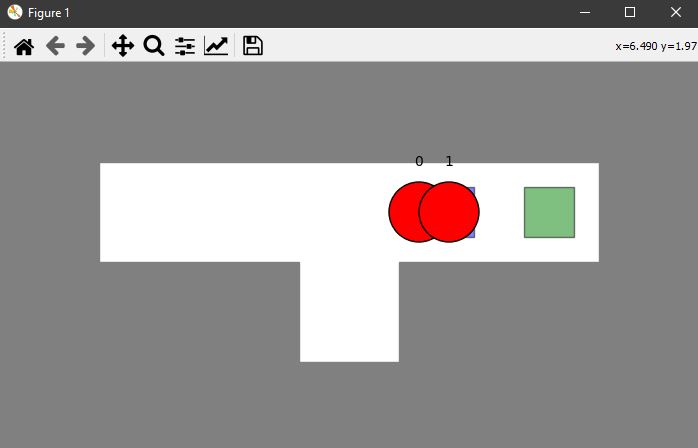


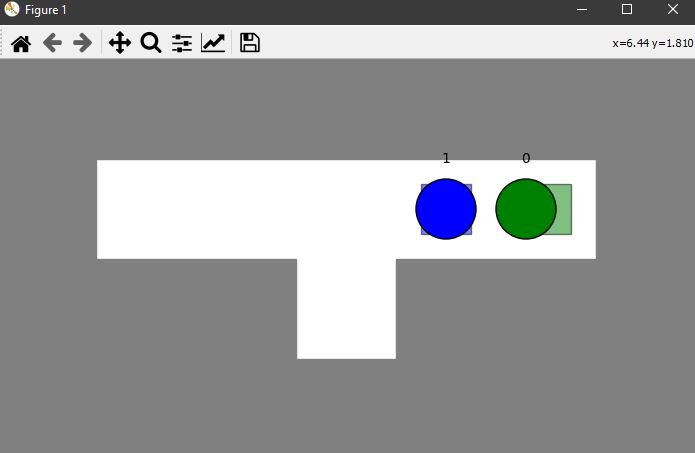
2: Code – 1.4 constraint ensures agent 0 does not achieve goal state before timestep of 10



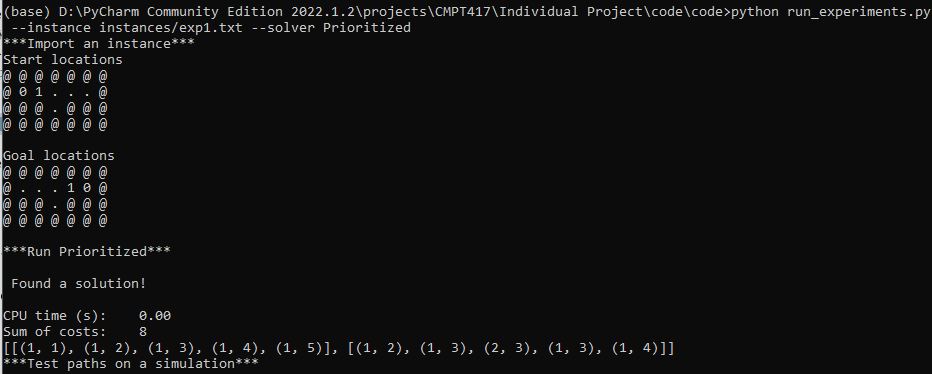
3: Agent 0 goes back and forth between (1,2) and (1,3)



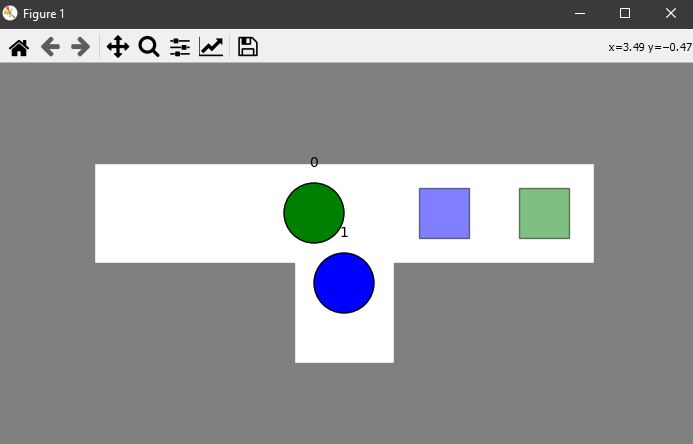




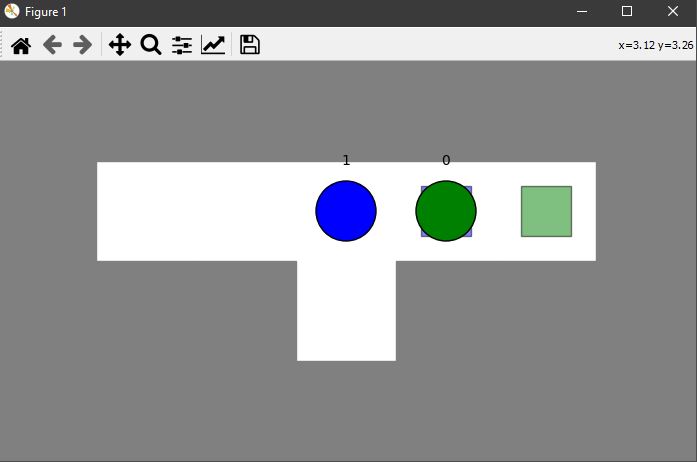
**1.5: Designing Constraints**

****

1: Terminal Output

****

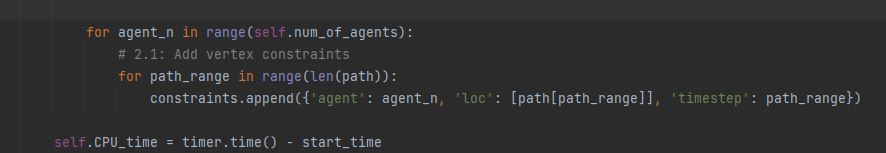
2: No collisions – at (1,3), agent 1 goes to (2,3), allowing agent 0 to pass without any collisions

****

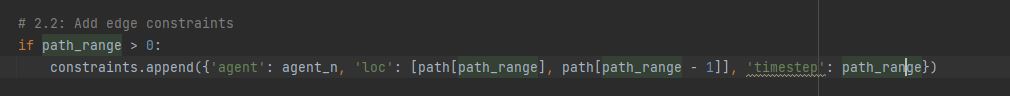
3: Both agents eventually reach goal state without collisions

**Task 2: Implementing Prioritized Planning**

**2.1: Adding Vertex Constraints**

****

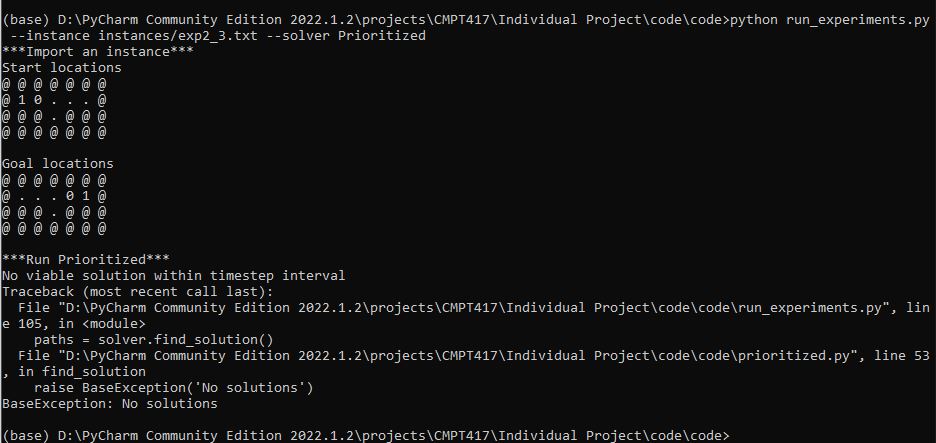
**2.2: Adding Edge Constraints**

****

**2.3: Adding Additional Constraints**

****

**2.4: Addressing Failures**

****

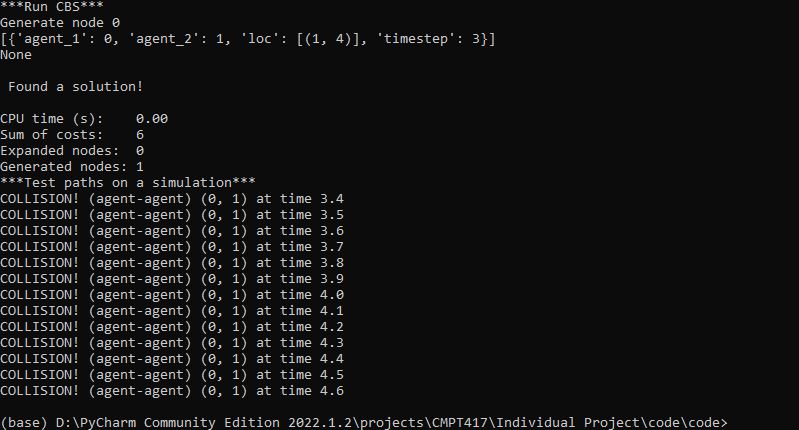
No solutions for exp2\_3

**2.5: Showing that Prioritized Planning is Incomplete and Suboptimal**

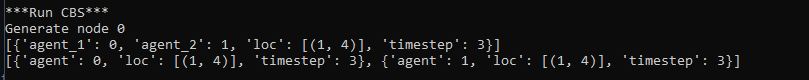
As we can see, there are no solutions for exp2\_3. There is a solution but not within 10 timesteps. Furthermore, other cases/experiments did provide a solution but contained collisions so it wasn’t viable.

**Task 3: Implement Conflict Based Search (CBS)**

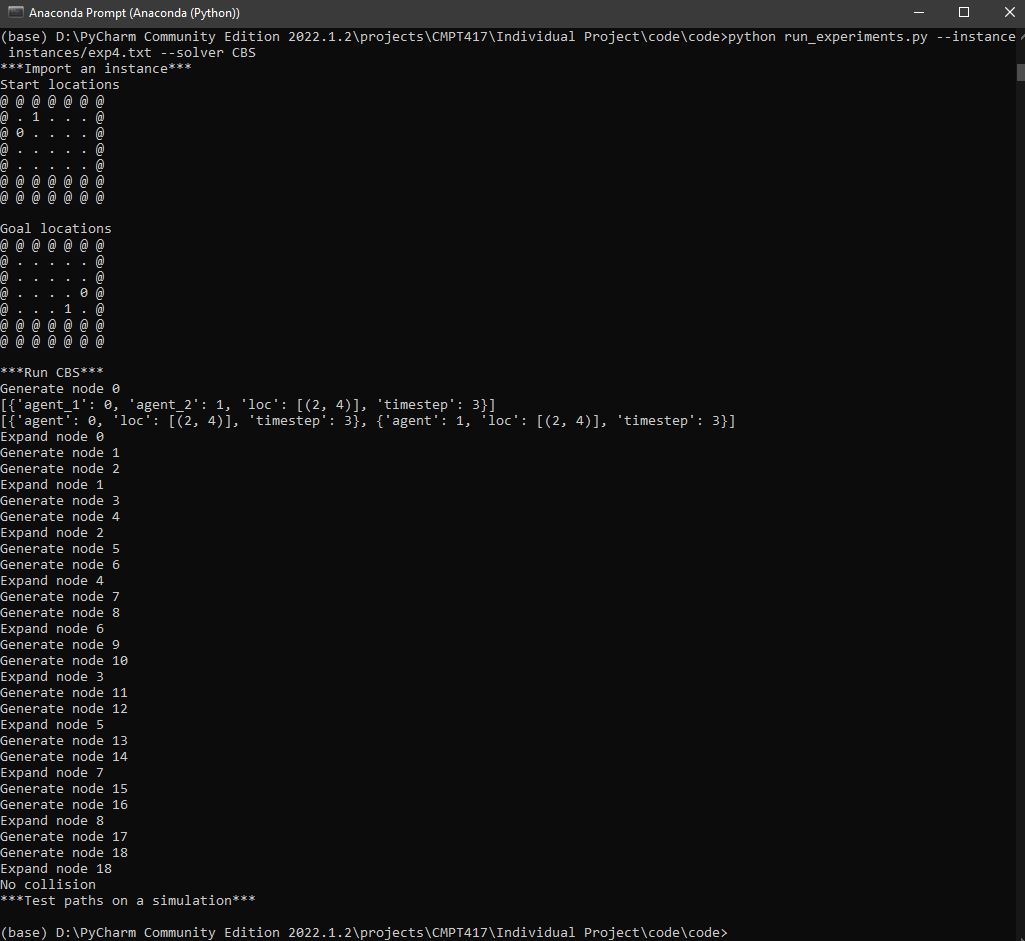
**3.1: Detecting Collisions**

****

**3.2: Converting Collisions to Constraints**

****

**3.3: Implementing High-Level Search**

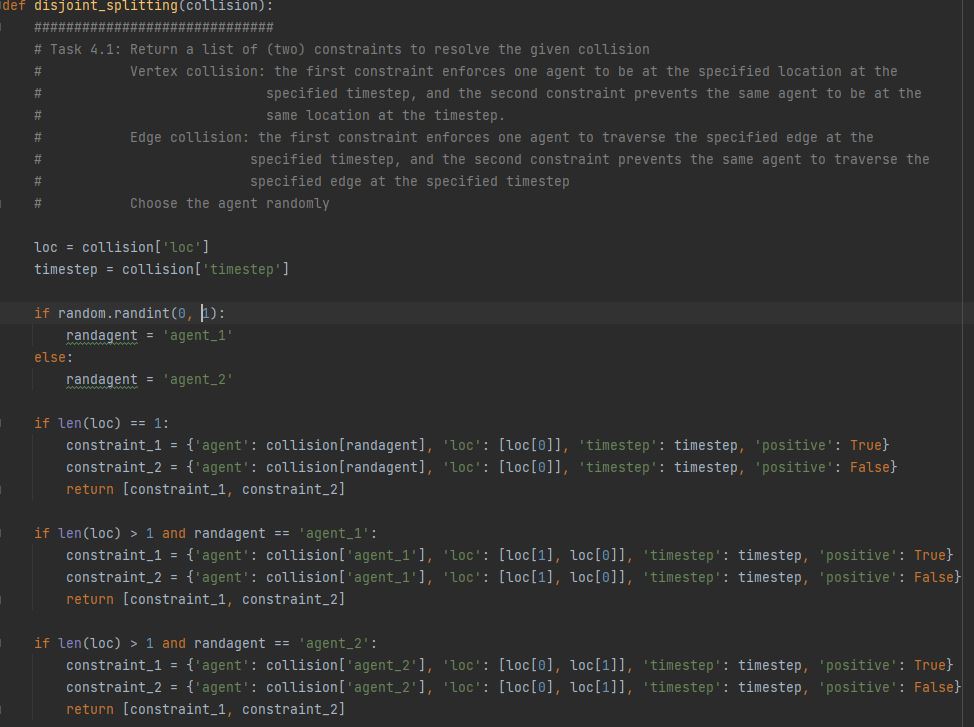
****

**Task 4: Implementing CBS with Disjoint Splitting**

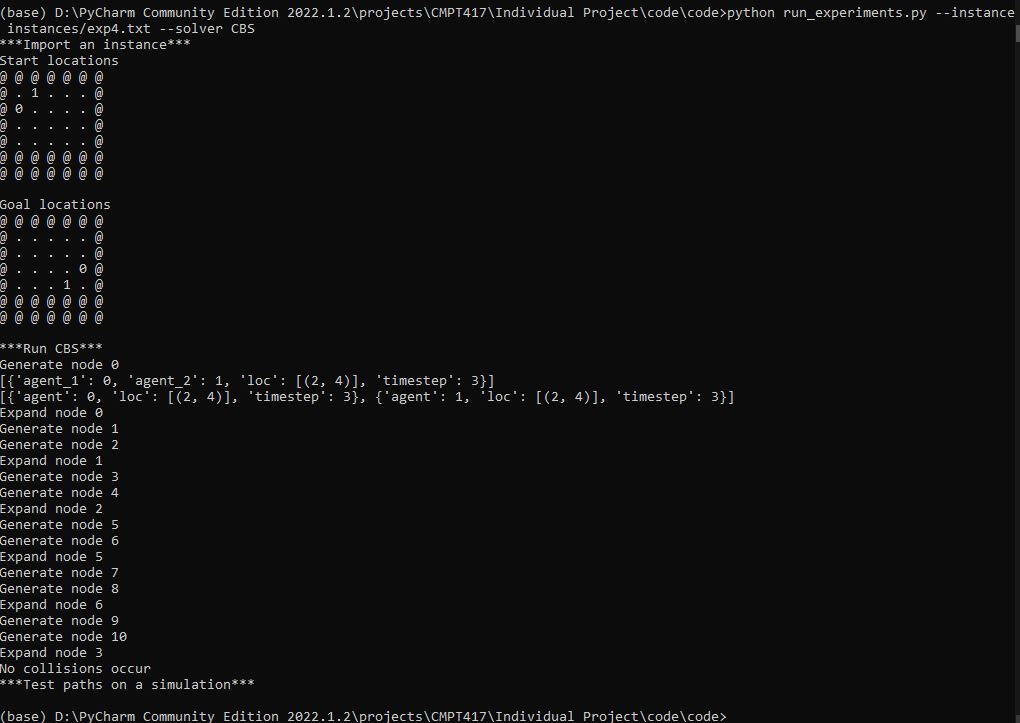
**4.1: Supporting Positive Constraints**

****

**4.2: Converting Collisions to Constraints**

****

**4.3: Adjusting High Level Search**

****

**Task 5: Benchmarking MAPF Solvers**

There were various benchmark instances but unfortunately, none of them worked upon running. The maps appear to be too big to run.