

Introduction Main Assignment

15-09-2022

Main assignment

- Multi-agent planning and coordination
- Agents need to move between their start and goal locations
- Inefficient planning can limit capacity

Example: Warehouses & Airport Surface Movements



amazon

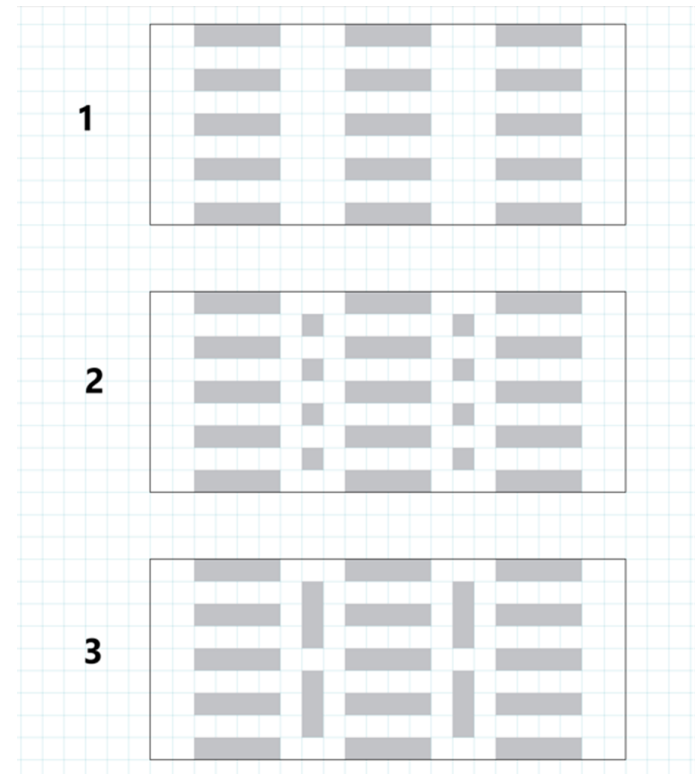


Example: Automated Baggage Handling



Assignment: three layouts with increasing number of obstacles

- Grey cell areas represent obstacles
- Agents can move on white cells
- Path planning and coordination needed to avoid obstacles and collisions.



Goal: Compare 3 planning methods

1. Prioritized planning with A*
 2. Conflict-based search (CBS)
 3. Distributed planning
- Any code provided may be used
 - Being creative is encouraged

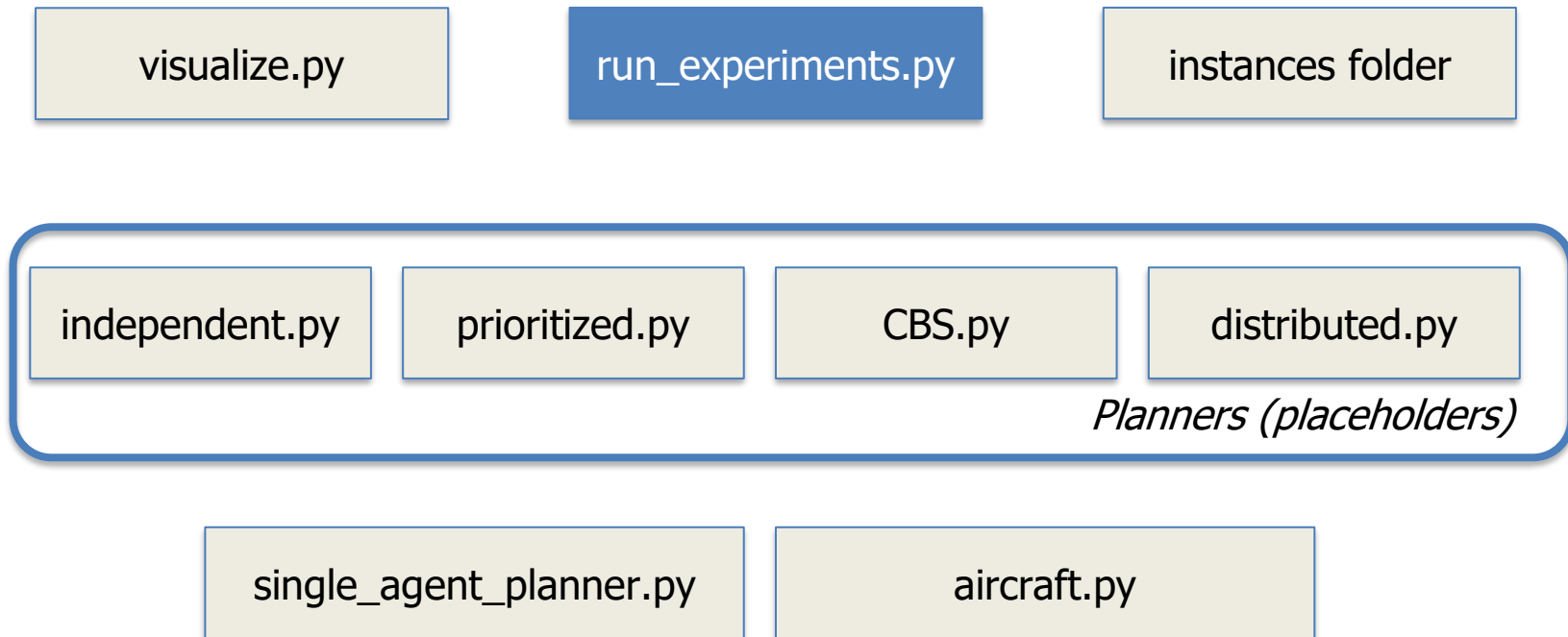
Deadline: 20th November

Deliverables

- Report
- Code

- 2 persons per group
- All group members submit individually
- Report & code should be the same for all group members

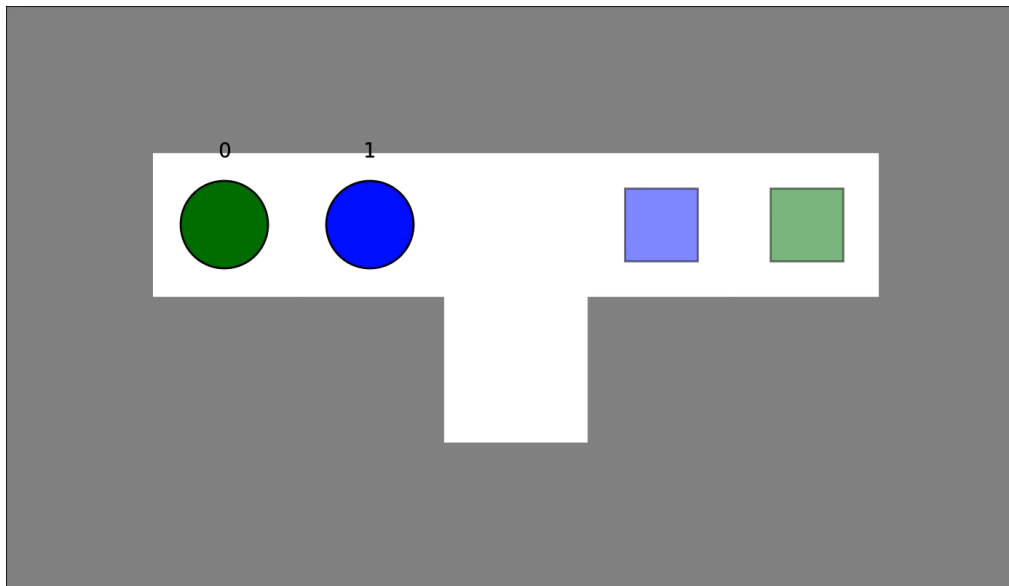
Base Model Structure



Run only "run_experiments.py"!


Run_experiments.py

- Main file to run simulation
- Needs command line options to execute
- Visualizes the planned paths *after* planning



Independent planner
should run out of the
box!

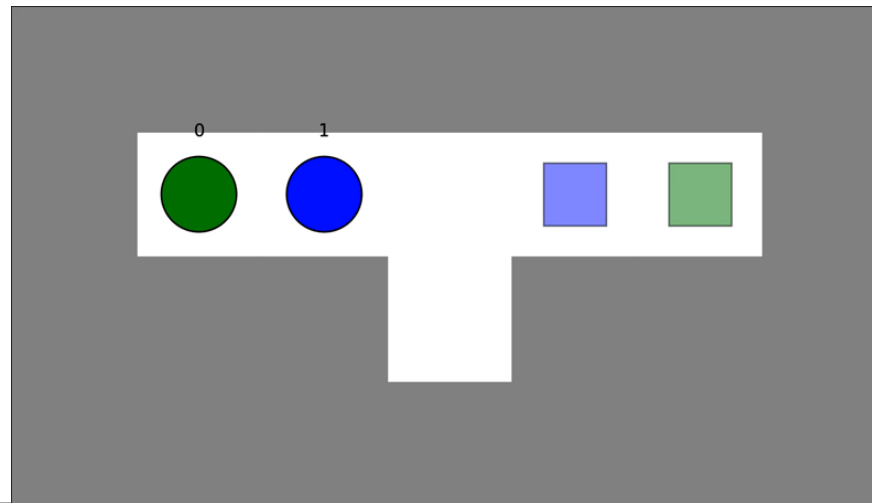
instances folder

 *exp0.txt - Notepad

File Edit Format View Help

```
4 7          --> grid size is 4 x 7
@ @ @ @ @ @ @ --> first row only has obstacles (@)
@ . . . . . @ --> second row has 5 cells to move on (.)
@ @ @ . @ @ @
@ @ @ @ @ @ @
2            --> 2 agents in this mapf instance
1 1 1 5      --> agent 0 start coords (x=1, y=1) and goal coords (x=1, y=5)
1 2 1 4      --> agent 1 start coords and goal coords
```

Remember:
Python starts
counting from 0!



aircraft.py

- **Aircraft class**

```
class AircraftDistributed(object):  
    """Aircraft object to be used in the distributed planner."""  
  
    def __init__(self, my_map, start, goal, heuristics, agent_id):  
        """  
        my_map    - list of lists specifying obstacle positions  
        starts     - (x1, y1) start location  
        goals      - (x1, y1) goal location  
        heuristics - heuristic to goal location  
        """  
  
        self.my_map = my_map  
        self.start = start  
        self.goal = goal  
        self.id = agent_id  
        self.heuristics = heuristics
```

- Suggested to use this class in the implementation of individual planning (next to DistributedPlanningSolver class).
- You can add class *attributes* and *methods* yourself (more info: <https://docs.python.org/3.9/tutorial/classes.html>).

single_agent_planner.py

- Supporting functions that can be used in planners. For example:
 - Basic version of A*
 - Heuristic calculation
- **Functions shall be adapted to be used in planners that you will apply/design in the assignment.**

Exercises and advised planning

| Exercise | Suggested deadline |
|---|--------------------|
| Explore baseline code | Today |
| 1 – Identifying performance indicators | 22 Sept |
| 2a – Prioritized planning with A* + evaluation | 29 Sept |
| 2b – Prioritized planning with CBS + evaluation | 6 Oct |
| 3 – Individual planning of moving agents | 13 Oct |
| 4 – Coordination between agents | 20 Oct |
| 5 – Evaluation of distributed planning | 27 Oct |
| 6 – Comparison of planning methods | 3 Nov |
| Reporting | 20 Nov |

Group

- form groups of two yourself
- see also Brightspace forum to find a partner

Next steps

- **You are advised to use an IDE such as Spyder, Pycharm, etc.**
- **Perform tasks of the tutorial to get base model running**
- **Explore the model and start assignment as soon as possible**
- **Questions?**
 - Google first! Online resources such as Stack Overflow might already answer your question
 - Practicum
 - Open office hours
 - MS Teams channel (helping each other is encouraged)

Next steps: additional material

- lecture slides
- tutorial slides
- overview + introduction to assignment
- code base

→ see Brightspace, Lab Assignments, Related material

Reporting Python issues

1. Try to figure out the root cause of your problem. Suggested steps:

- Use print statements or debugger option to check your code step by step
- Google error types (use for example stack overflow)

2. Check if your problem was discussed before in a MS Teams channel, you might find an answer there.

3. Share relevant information in your question and pick the relevant teams channel, e.g:

- Type of error + traceback, file you are working in & context, steps you have taken to resolve it, expected behavior vs. observed behavior, code that you think causes the failure & why, other information that you think might be relevant.

So take the time to formulate your question! A well formulated question has the highest chance of receiving a helpful answer quickly.

MS Teams group

- unresolved Python issues
- helping each other

→ link:

<https://teams.microsoft.com/l/channel/19%3aD3z6sYf6uDwDYIp74BPDmQuscb5-UQzFi69equNgMWg1%40thread.tacv2/General?groupId=e0c8b787-018f-4960-ab93-5a0f2b47229a&tenantId=096e524d-6929-4030-8cd3-8ab42de0887b>

Help with Anaconda

- Anaconda Cheat Sheet:
https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf
- our advice is (use the Anaconda prompt to do the following):
 - create a new environment: `conda create --<name> python=3.9`
 - activate newly created environment
 - install packages: `conda install spyder numpy matplotlib`
 - open Spyder (<name>)
 - happy coding! 😊