

COMP90054 AI Planning for Autonomy - Project 2 - PDDL modelling

You must read fully and carefully the assignment specification and instructions detailed in this file. You are NOT to modify this file in any way.

- **Course:** [COMP90054 AI Planning for Autonomy](#) @ Semester 2, 2020
- **Instructor:** Dr. Nir Lipovetzky and Prof. Adrian Pearce
- **Deadline:** Sunday 13th September, 2020 @ 11:59pm (end of Week 6)
- **Course Weight:** 5%
- **Assignment type::** Individual
- **ILOs covered:** 1, 2, and 3
- **Submission method:** via git tagging (see Submission Instructions below for instructions)

The **aim of this project** is to improve your understanding of PDDL modelling.

Task

Your task is to read the description of the the domain below and encode the three planning problem instances specified filling the template files provided in this repository in the folder [pddl_template](#).

The Domain - Real Player Game Scenario

Help the hero to get out of dungeon!

A hero woke up in a dungeon full of monsters and traps (perhaps the party last night went wrong...) and needs your help to get out. Here are basic facts for the dungeon domain:

- The dungeon contains rooms that are **connected** by corridors (dungeon can thus be represented by undirected graph)
- each room can be **empty**, or can have a **monster** in it, or can have a **trap** in it, or can have a **sword** in it
- one of the empty rooms is the **goal**: it has an exit, so the hero can escape

The hero is lucky since she/he has full knowledge about the dungeon. But not that lucky – just after the hero leaves each room s/he visits, the room is **destroyed** and cannot be visited again.

The hero can perform the following actions – but only if s/he is alive!

- The hero can **move** to an adjacent room (connected by a corridor) that has not been destroyed (i.e., the hero has not already visited the room)
- **Pickup** the sword if present in the room the hero is currently in and the hero is empty handed
- **Destroy** the sword that the hero currently holds. However, this can have unpleasant effects if done in a room with a trap or a monster: it triggers the trap or invites the monster to eat the hero.
- **Disarm a trap** – if there is a trap in the room the hero is in and the hero is empty-handed (does not hold a sword), then the hero can disarm it

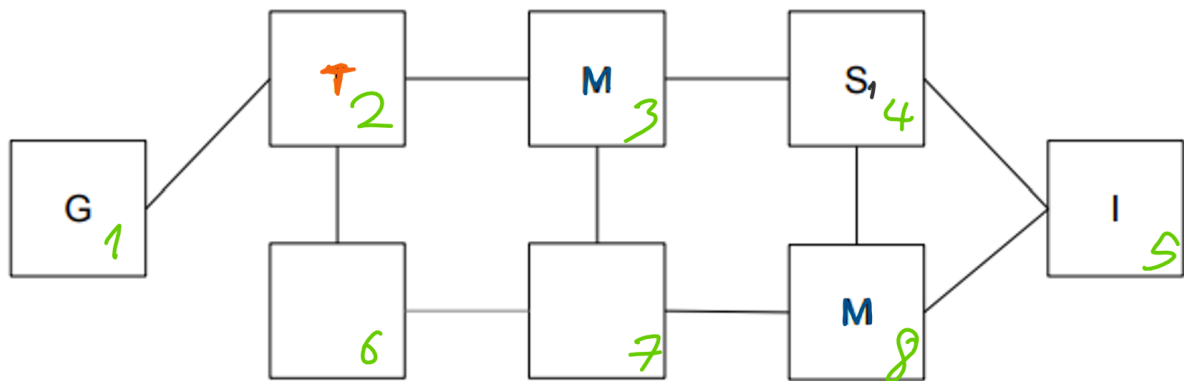
However, there are some (dangerous) constraints the hero has to consider:

- If the hero enters a room with a monster in it, s/he has to carry a sword (so the monster is afraid of him/her), otherwise the monster kills him/her. Notice that the hero is pacifist, so s/he cannot kill the monster.
- If the hero destroys the sword in a room with a monster in it, the monster kills him/her as well.
- The only action the hero can safely perform in a room with a trap in it is the **disarm a trap** action. Any other action (even moving away) triggers the trap which kills the hero.

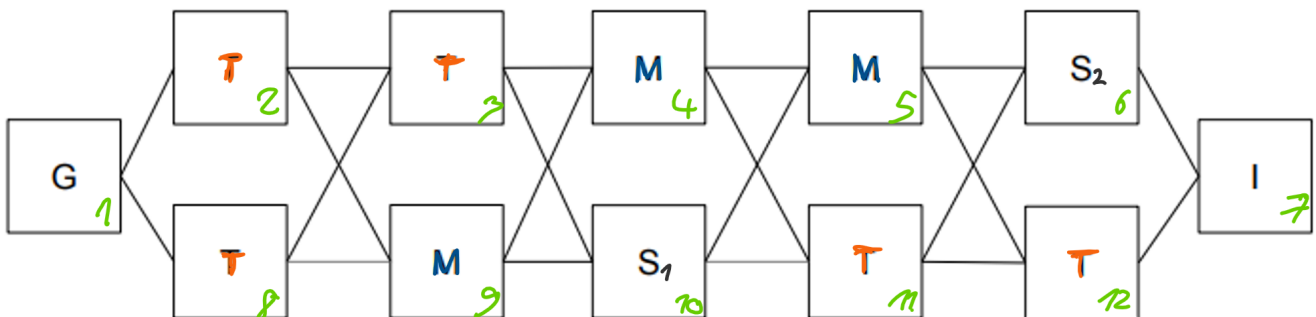
Important: We recommend reading the description carefully several times to make sure you fully understand it before starting coding in PDDL.

Problem 1

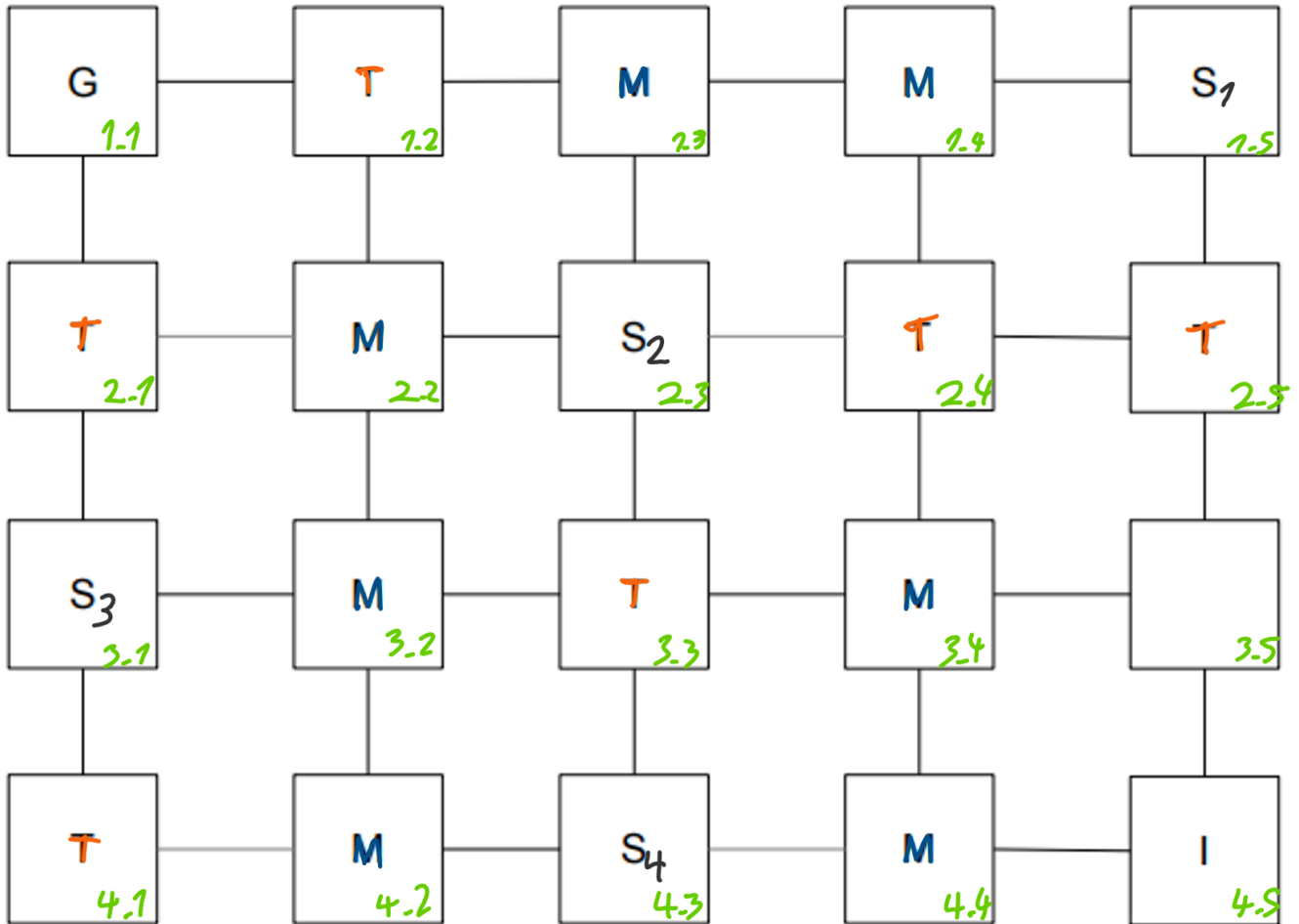
The problems are specified such that cells stand for rooms and edges between them represent corridors. **I** is the hero's initial position, **G** is hero's desired goal position, **\$S_i\$** indicates a sword, **M** is a monster, and **T** stands for trap. Good luck!



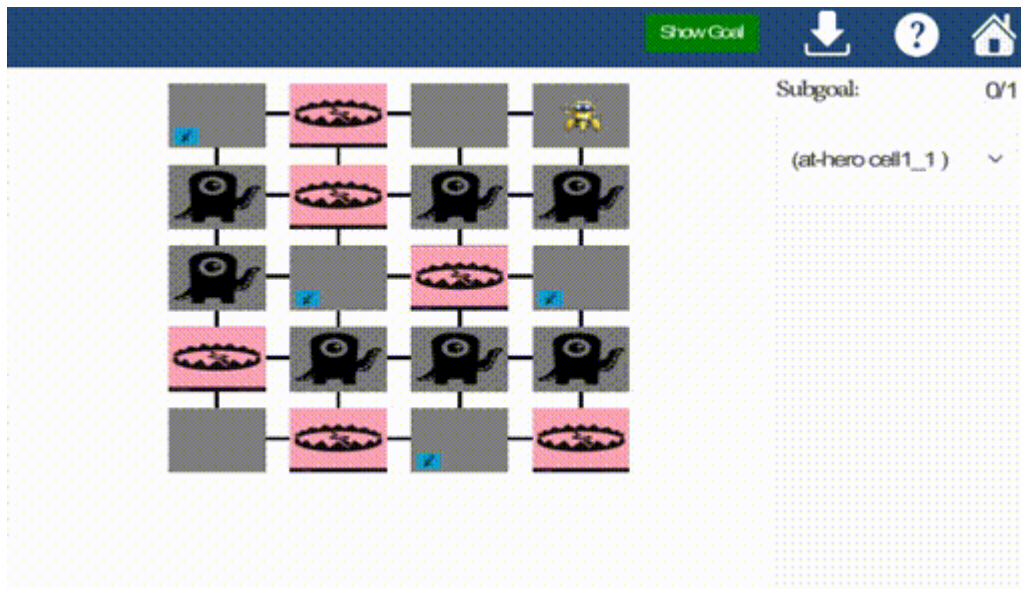
Problem 2



Problem 3



Planimated Example Solution



Marking Criteria

This assignment has a weight of 5% over the final grade. The criteria for marking will be:

- 3 points for encoding the domain correctly. Each correct action counts 0.5 points. If a precondition or effect is missing from an action, the action will earn 0 points.

- 1 point for encoding the 3 problems correctly
- 1 point for generating correct solutions as per the specification of the problem

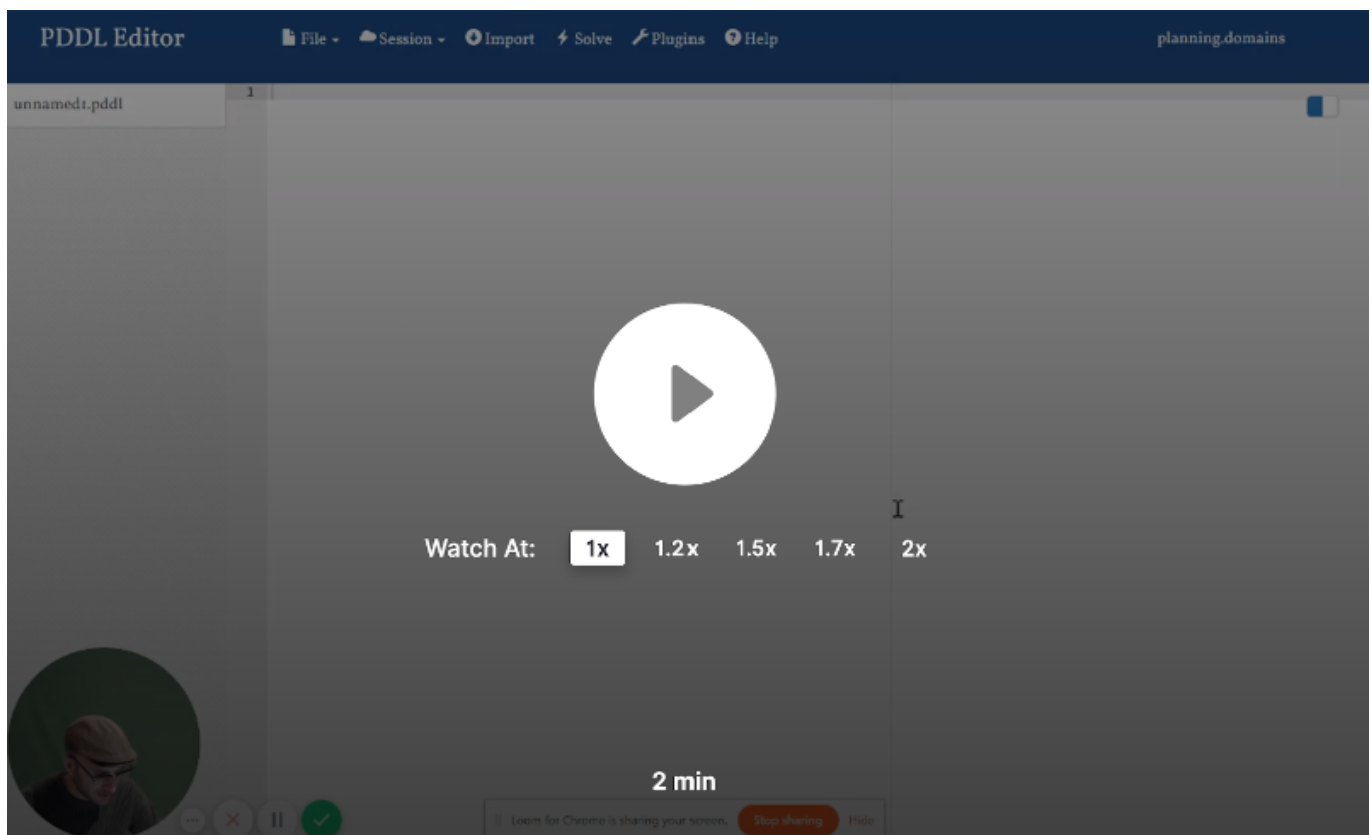
Your marking will be based on the domain and problem files that you need to complete in the folder [pddl_template](#).

Important: Make sure to tag your commit with your updated pddl files inside that folder.

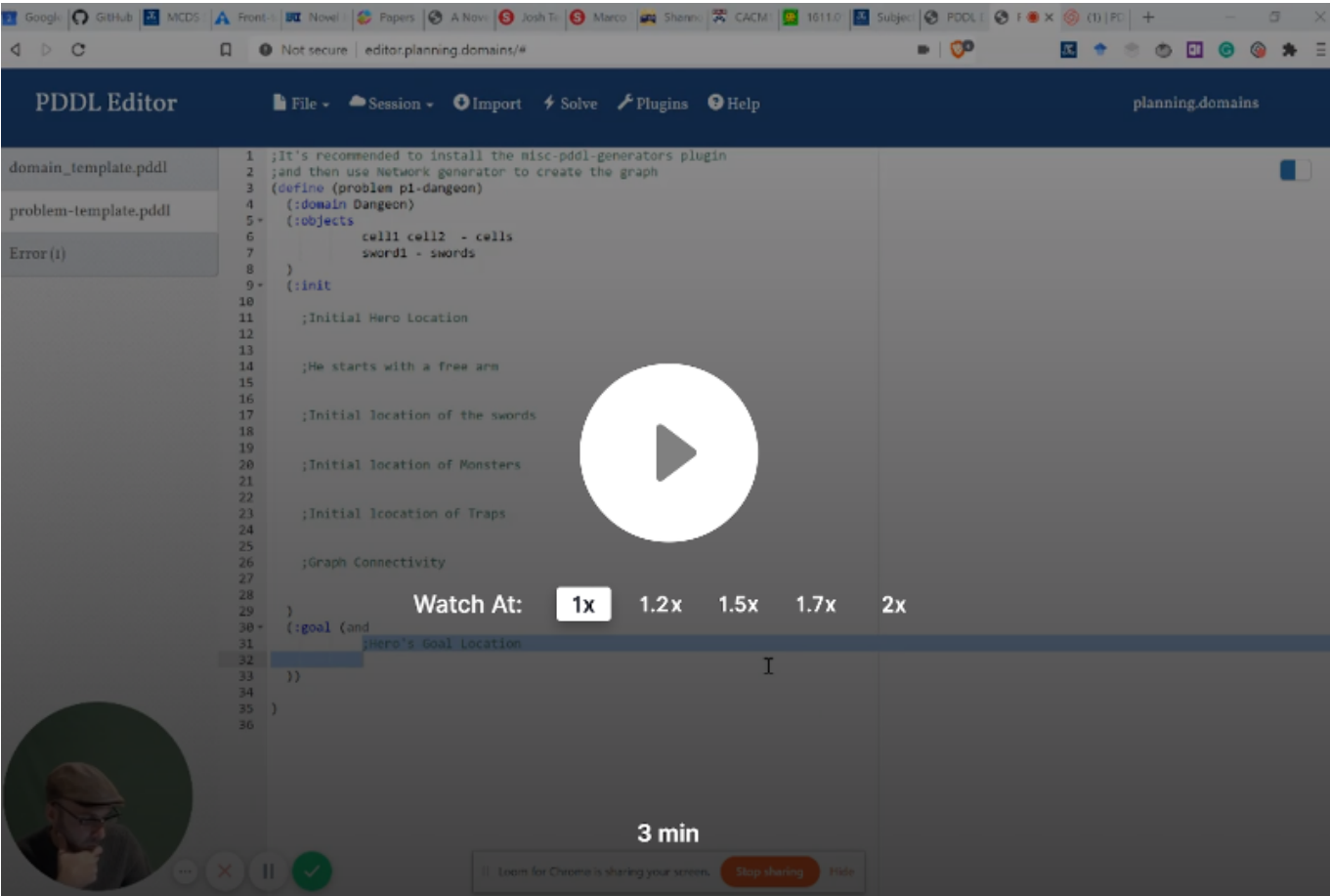
How to get started

Once you clone the files from this repository onto your local machine, watch the following videos on how to get started. Open the [pddl_template](#) folder and start modeling.

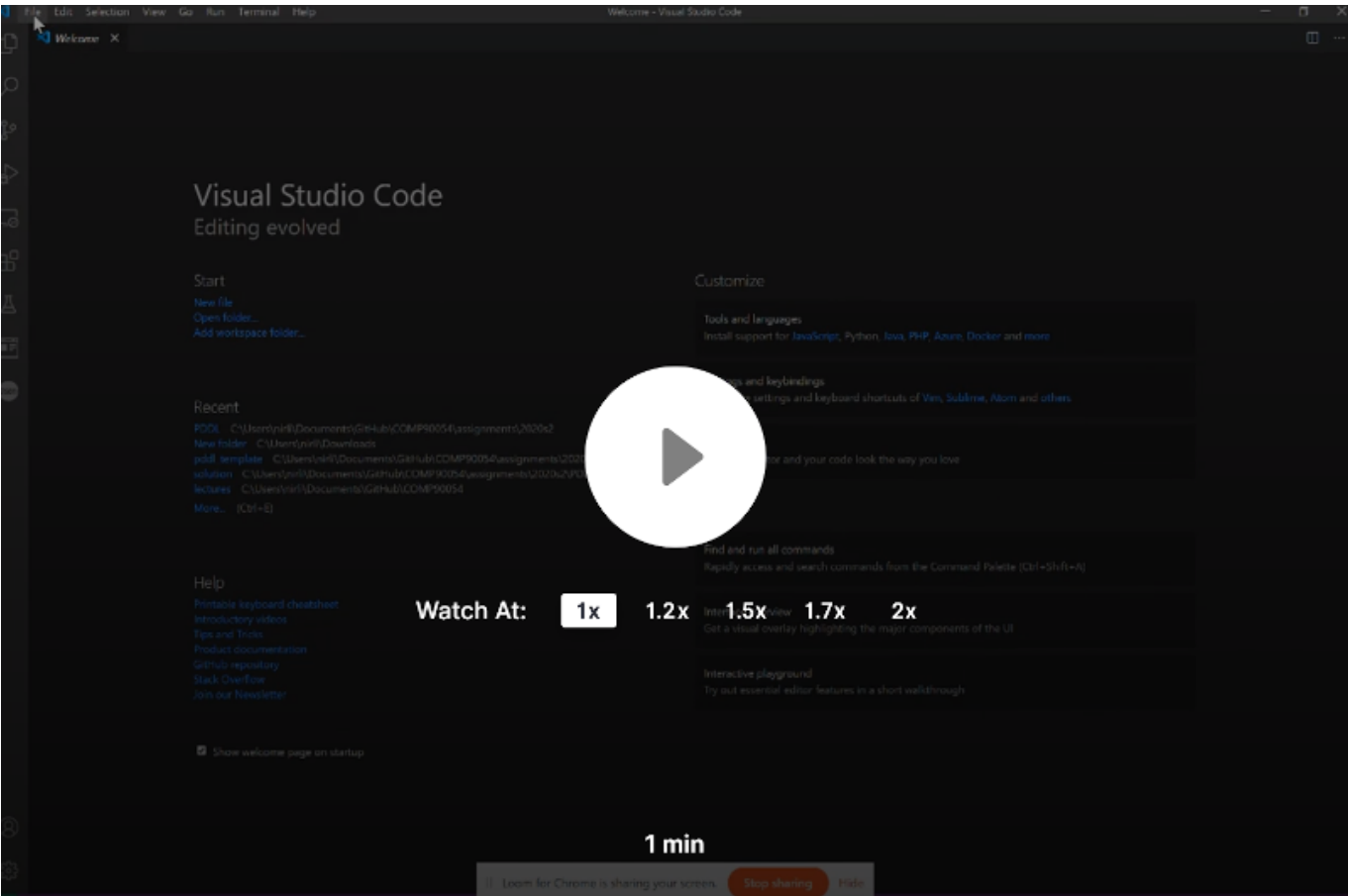
Upload to Editor.planning.domains, how to save a session and work online



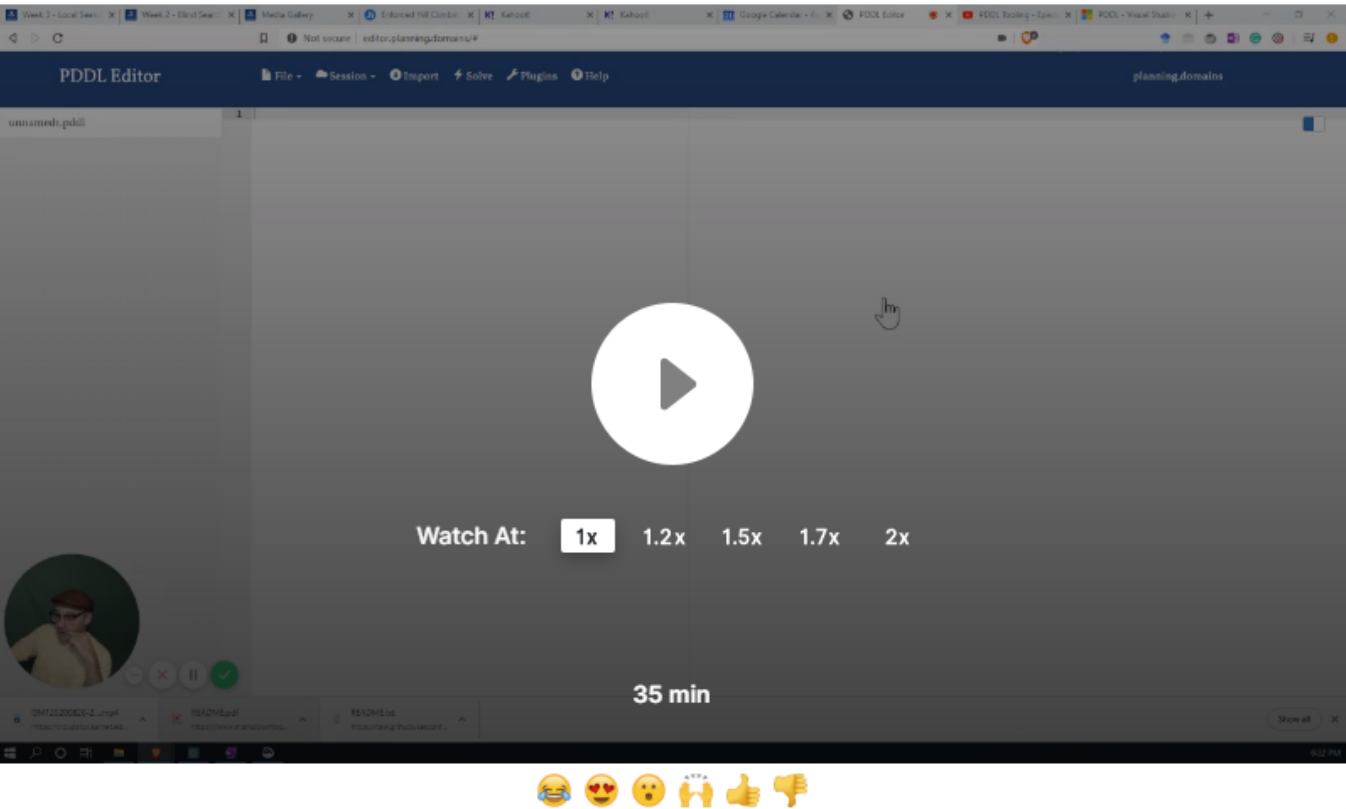
Download session offline with vscode and call the online planner



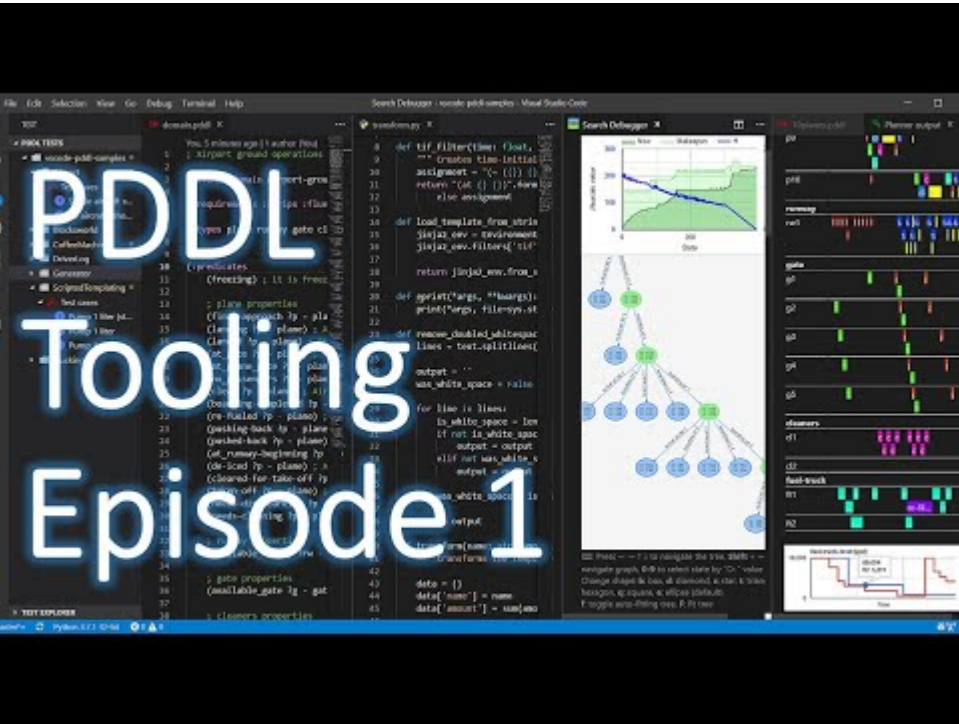
Work with VScode directly and the online planner



Live modeling session in PDDL:



VScode PDDL Tutorial:



Submission Instructions

Tag your submission to submit your assignment by completing the following three steps:

1. Complete the [STUDENT.md](#) file with your details of the submission.

2. Please ensure that your solution does not include personal information, like your student number or name.
3. Tag the commit version you want to be graded with tag **submission**.
 - The commit and tagging should be dated before the deadline.
 - Note that a tag is NOT a branch, so do not just create a branch called "submission" as that will not amount to tagging.
4. Fill the [Project 2 Certification Form](#).
 - Non-certified submissions will not be marked and will attract zero marks.

From this repository, we will copy *only* the PDDL files: [domain.pddl](#), [problem1.pddl](#), [problem2.pddl](#), and [problem3.pddl](#). Please do not submit any other file with a different name as part of your solution, we will not be able to mark it. Breaking these instructions breaks our marking scripts, delays marks being returned, and more importantly, gives us a headache. Submissions not compatible with the instructions in this document will attract zero marks and do not warrant a re-submission. Staff will not fix your submission.

Please view the following to learn how to *Tag* your commit version you want to be graded:

How to create a Tag using the Command Line:

<https://www.loom.com/share/17ec72b954454bc89bbe1dbb0bd2874f>

Another way to create a Tag using the User Interface:

<https://www.loom.com/share/3cd39e97919e4b688d9841613aba6973>

Important information

Corrections: From time to time, students or staff find errors (e.g., typos, unclear instructions, etc.) in the assignment specification. In that case, corrected version of this file will be produced, announced, and distributed for you to commit and push into your repository. Because of that, you are NOT to modify this file in any way to avoid conflicts.

Late submissions & extensions: A penalty of 10% of the maximum mark per day will apply to late assignments up to a maximum of five days, and 100% penalty thereafter. Extensions will only be permitted in *exceptional* circumstances; refer to [this question](#) in the course FAQs.

About this repo: You must ALWAYS keep your fork **private** and **never share it** with anybody in or outside the course, except your teammates, *even after the course is completed*. You are not allowed to make another repository copy outside the provided GitHub Classroom without the written permission of the teaching staff.

Please do not distribute or post solutions to any of the projects.

Academic Dishonesty: This is an advanced course, so we expect full professionalism and ethical conduct. Plagiarism is a serious issue. Please **don't let us down and risk our trust**. The staff take academic misconduct very seriously. Sophisticated *plagiarism detection* software (e.g., [Codequiry](#), [Turinitin](#), etc.) will be used to check your code against other submissions in the class as well as resources available on the web for logical redundancy. These systems are really smart, so just do not risk it and keep professional. We trust you all to submit your own work only; please don't let us down. If you do, we will pursue the strongest consequences available to us according to the **University Academic Integrity policy**. For more information on this see file [Academic Integrity](#).

We are here to help!: We are here to help you! But we don't know you need help unless you tell us. We expect reasonable effort from you side, but if you get stuck or have doubts, please seek help. We will run labs to support these projects, so use them! While you have to be careful to not post spoilers in the forum, you can always ask general questions about the techniques that are required to solve the projects. If in doubt whether a question is appropriate, post a Private post to the instructors.

Silence Policy: A silence policy will take effect **48 hours** before this assignment is due. This means that no question about this assignment will be answered, whether it is asked on the newsgroup, by email, or in person. Use the last 48 hours to wrap up and finish your project quietly as well as possible if you have not done so already. Remember it is not mandatory to do all perfect, try to cover as much as possible. By having some silence we reduce anxiety, last minute mistakes, and unreasonable expectations on others.

Please remember to follow all the submission steps as per project specification.

COMP90054 Code of Honour

We expect every UoM student taking this course to adhere to it **Code of Honour** under which every learner-student should:

- Submit their own original work.
- Do not share answers with others.
- Report suspected violations.
- Engage in any other activities that will dishonestly improve their results or dishonestly improve or damage the results of others.

Unethical behaviour is extremely serious and consequences are painful for everyone. We expect enrolled students/learners to take full **ownership** of your work and **respect** the work of teachers and other students.

I hope you enjoy the project and learn from it, and if you still **have doubts about the project and/or this specification** do not hesitate asking in the [Piazza Course Discussion Forum](#) and we will try to address it as quickly as we can!

GOOD LUCK and HAPPY MODELLING!

Acknowledgements

This problem was originally proposed at *The Fifth International Competition on Knowledge Engineering for Planning and Scheduling* [ICKEPS 2016](#). Author: [Lukas Chrpá](#), Charles University in Prague.

If you want to practice modeling other domains, take a look at the [other problems](#) presented in the competition. All but the first one can be modeled using STRIPS and actions costs.