Projects: Soft goals & resources in @Work

@Work domain:

- **Workstations**: s1, s2,…
- **Objects:** o1, o2,...
- **Initial state:** which objects are at which workstations.
- **Goal statement:** which objects should be at which workstations.
- **Assumptions:**
  - Fully observable domain (sort of)
  - Complete information (sort of)
  - No dynamic obstacles
  - Distances between workstations are not specified in the task specification, but known in advance: D(s1,s2) = 5

Projects: Soft goals & resources in @Work

Given:

- Three planning problems:
  - **P1**=‘BTT<initialsituation(<s1,(o1)><s2,(o2)><s3,(o3)>);goalsituation(<s3,line(o1,o2,o3)>)>’
  - **P2**=‘BTT<initialsituation(<s3,(o1,o2,o2,o3,o4)>);goalsituation(<s1,line(o1,o2,o2,o3,o4)>)>’
  - **P3**=‘BTT<initialsituation(<s5,(o5,o4,o1)><s2,(o7,o3,o5)><s3,(o2,o6,o8)>);goalsituation(<s1,line(o6,o4,M20)><s4,line(o8,o5,o5)><s6,line(o1,o7,o2)>)>’
- Distance Matrix (next slide)
- Reward vector: ‘R(<o1,10>,<o2,5><o3,1><o4,1><o5,1><o6,2><o7,2><o8,1>)’

Projects: Soft goals & resources in @Work

	s1	s2	s3	s4	s5	s6
s1	x	1	2	5	4	5
s2	1	x	1	6	4	5
s3	2	1	x	4	4	3
s4	5	6	4	x	2	1
s5	4	4	4	2	x	2
s6	5	5	3	1	2	x

Distance Matrix

Projects: Soft goals & resources in @Work

	Description	Points
1	**Choose** what you judge to be the “right” planner for this particular domain based on the IPC results.	5
2	Get it to **run.**	5
3	**Model the domain** for the basic fetch & carry	10
4	**Generate a plan** for each of the three planning problems	15
5	**Assign costs** to the distances and find the optimum plans (least cost).	15
6	Extend this to enable the robot to **carry only 3 objects** at any given time.	15
7	**Assign rewards** to the pieces and find the optimum plan (maximum reward).	15

*We want to test **correctness**, **scalability**, ability to **minimize cost**, ability to **maximize utility**, ability to **handle resources**.*

Projects: Soft goals & resources in @Work

Submit a .zip file containing:

- A **report** (.pdf):
  - Stating why you chose a particular planner.
  - Challenges you had installing and running it (include installation instructions & simple script to run each test).
  - An analysis of your results (number of plans generated, time it took to generate them, any difficulties, quality of the plans, anything else you think is of interest).
- A **folder/problem/setting** (i.e. 12 folders in total) which includes the planning problem, the model of your domain and the generated plans.
- A 10-minute **presentation** summarizing your project.

Projects: Soft goals & resources in @Work

Helpful Resources:

- Results of 2014 IPC:
  https://helios.hud.ac.uk/scommv/IPC-14/resDoc.html
- Very helpful in understanding the various tracks, especially: slide #9):
  https://helios.hud.ac.uk/scommv/IPC-14/repository/slides.pdf
- Shortest path:
  http://en.wikipedia.org/wiki/Shortest_path_problem

Projects: Making Tea @Home

@Home domain:

- **Assumptions:**
  - Partially observable domain
  - Incomplete information

Given:

- 1 plan
- **Test 1**: S0 includes a clean cup on the counter top, the tea bag next to it, the kettle is on the kettlebase and it is empty.
- **Test 2**: S0 as in Test 1, but the only clean cup is on shelf2 in cupboard4, and there are 75 other cups in various places, only 30 of which are known to be dirty. There is no information on whether the others are clean or not. The plan must use a clean cup.
- **Test 3**: S0 is as in Test 1 but with 2 clean cups on a shelf in a cupboard, and the kettle is full. The task is to make two cups of tea.

[1] (!goto kettle1 ForGrasping)
[2] (!access kettle1)
[3] (!open kettle1)
[4] (!grasp kettle1 ForTransport)
[5] (!goto kitchenSink ForFilling)
[6] (!position kettle1 ForFilling)
[7] (!opentap coldtap ForFilling)
[8] (!closetap coldtap)
[9] (!grasp kettle1 ForTransport)
[10] (!goto kettleBase ForReplacing)
[11] (!access kettleBase1)
[12] (!replace kettle1 kettleBase)
[13] (!close kettle1)
[14] (!boilWaterInKettle kettle1)
[15] (!goto teacup2 ForGrasping)
[16] (!access teacup2)
[17] (!grasp teacup2 ForTransport)
[18] (!placeNextTo teacup2 kettle1)
[19] (!goto peppermintTeabag ForGrasping)
[20] (!access peppermintTeabag)
[21] (!grasp peppermintTeabag ForMakingTea)
[22] (!placeIn peppermintTeabag teacup2)

Projects: Making Tea @Home

	Description	Points
1	**Choose** an HTN planner and get it to run.	5
2	**Model the domain** (operators, methods, decomposition) that results in this plan.	20
3	**Create the basic planning problem** to test your domain model (Test 1).	5
4	**Generate plans** for this basic planning problem and refactor your domain if needed.	5
5	**Extend** your domain to handle test 2 and generate plans for it.	10
6	**Extend** your domain to handle test 3 and generate plans for it.	10
7	Modify the problem and the planner to provide explanations of planning failures (failed preconditions, for which methods or operators, etc.).	25

*We want to test **correctness**, **scalability**, ability to **handle various situations**.*

Projects: Making Tea @Home

Submit a .zip file containing:

- A **report** (.pdf):
    - Stating why you chose a particular planner.
    - Challenges you had installing and running it (include installation instructions & simple script to run each test)
    - An analysis of your results (number of plans generated, time it took to generate them, any difficulties, quality of the plans, anything else you think is of interest).
- A **folder/problem** (i.e. 3 folders in total) which includes the planning problem, the model of your domain and the generated plans.
- A 10-minute **presentation** summarizing your project.

Projects: Making Tea @Home

Helpful Resources:

- “HTN planning: Overview, comparison, and beyond” by Ilche Georgievski, Marco Aiello <http://www.sciencedirect.com/science/article/pii/S0004370215000247>
- Very helpful notes on authoring domains for HTN planners: <http://www.ida.liu.se/~TDDD48/labs/2012/lab4.en.shtml>

Projects: Continual Planning @Home

@Home Continual Planning domain:

- **Assumptions:**
  - Partially observable domain (RoboCup @Home Lab C-069)
  - Incomplete information
  - You can provide a mockup function which provides observations (random)

Given:

- A fixed number of cleaning sub tasks are supported. i.e.: removing clutter from kitchen table, washing counter, clearing coffee table.
- All objects observed are known, and their correct locations are known.
- Robot Jenny is limited by only having one arm.
- Unlike other projects, this requires only one planning problem, as the random generators will change the results. But the output of the random generators must be logged, essentially recording the planning problem which wasn't known at the start.

Projects: Continual Planning @Home

	Description	Points
1	**Choose** a planner and get it to run.	10
2	**Model the domain** (operators, methods, decomposition).	20
3	**Create the basic planning problem and required mockup components** to test your domain model.	10
4	**Generate plans** for this planning problem and refactor your domain if needed.	5
5	**Extend** your domain to handle unknown objects, and ask for user input where it belongs.	5
6	**Extend** your domain to enable it to minimize the length of the plans (e.g. use a container to collect multiple objects for drop off at the same locations).	5
7	**Specify** monitoring actions and recovery mechanisms for each operator in the domain and specify what recovery actions should be used in case of failure.	25

*We want to test **correctness**, **scalability**, ability to **handle various situations**.*

Projects: Continual Planning @Home

Submit a .zip file containing:

- A **report** (.pdf):
    - Stating why you chose a particular planner.
    - Challenges you had installing and running it (include installation instructions & simple script to run each test)
    - An analysis of your results (number of plans generated, time it took to generate them, any difficulties, quality of the plans, anything else you think is of interest).
- A **folder/problem** (i.e. 3 folders in total) which includes the planning problem, the model of your domain and the generated plans.
- A 10-minute **presentation** summarising your project.

Projects: Continual Planning @Home

Helpful Resources:

- http://www.ifs.tuwien.ac.at/~silvia/pub/publications/mik_ecai2000.pdf
- https://www.aaai.org/ocs/index.php/WS/AAAIW11/paper/viewFile/3981/4240
- http://gki.informatik.uni-freiburg.de/papers/brenner-nebel06.pdf
- http://www.inf.ed.ac.uk/teaching/courses/plan/slides/Temporal-Planning-Slides.pdf