Assignment 1: The Flatland Challenge

In the combined problems in this assignment (Part A and B), your job will be to schedule a set of trains through a railway network.

You will need to coordinate every train from its starting station to its destination as quickly as possible. As there can be many trains moving at the same time, you need to guarantee that each path is collision-free.

The assignment has three questions in increasing order of difficulty. The amount of points relative to each question is stated in the question heading. A total passing grade is 50%.

There are **two separate submission points** in this assignment.
Part A (question 1 only) will be due at an earlier date. Your job is to make a valid submission to the contest server by this due date (see the *FIT5222 Flatland Challenge Submission Instructions* for submission details). The marks you receive in this submission will be used for marking purposes for Q1. All points for Q1 can *only* be obtained by the Part A submission deadline.

The rest of the assignment, Part B, requires answering questions 2 and 3 as well as submitting a report detailing your approach and numerical results. Remember, each question in this assignment builds on top of the previous - it is important that you get started early and ask questions when you get stuck. Do not leave this assignment until the last minute!

Be sure to watch the introductory video on Moodle and to read the Introduction to Flatland documentation which we have prepared for you. Both are available from Moodle.

You must update your flatland code and installation before starting the assignment:

- **Under flatland folder**
- *git pull*
- *python setup.py install*
- **Now, run command** *python -m pip list* **, you should see flatland-rl version is updated to 2.2.4**
- **(Hint, use** *python3* **instead of** *python* **if** *python3* **points to the right one on your machine)**

Instructions to get code base (in assignment1_2025 branch of piglet repo) for assignment is in the Introduction to Flatland documentation.

Part A

Question 1 only, total 15 points, due Sunday, 24 August 2025, 11:55 PM (end of Week 4).

Question 1: Warm up (15 points)

You are given start and target locations, one at a time. Your job is to route each train independently from all the rest. In this question collisions are not possible and there is no time dimension.

For this question, you need to implement a successor function for the Flatland domain. You also need to choose an algorithm to help you find paths. You are free to use any of the techniques we have discussed in the lectures, that you have read about in the literature or can write your own new approach.

Your solution will be evaluated on 40 evaluation instances (only staff have these instances) with a 2 hours time limit. Each instance has 1 agent. We will calculate a score, p_{score}, based on your sum of individual costs (SIC); for agents who did not arrive at their goal location, their cost will be $T_{max} = 8 * (width + height)$ of the map):

$$p_{score} = (completed\_agents\_SIC + T_{max} * failed\_agents) / total\_agents$$

We'll compare your solution to an optimal solution implemented by teaching team and calculate your score using following method:

$$f_{score} = p_{staff} / p_{score}$$

where p_{staff} is the p score of staff implementation. Generally, a good solution has small p score and large f score.

Your final score will be $f_{score} \times 15$(since there are 15 points available for this question).

Marks for this question are available *only* by submission in Part A (i.e., no marks can be retroactively obtained when submitting Part B).

Submission

Part A submission to the contest server is due on Sunday, 24 August 2025, 11:55 PM (end of Week 4).

- Follow the ***FIT5222 Flatland Challenge Submission Instructions*** to submit your code to the contest server.

Part B

Questions 2, 3, and the report for a total of 135 points, due Friday, 12 September 2025, 11:55 PM (end of Week 7).

Question 2: Basic scheduling (25 points)

You are given start and target locations, one at a time, as well as a set of existing paths for trains that are already moving. Your job is to route each train individually while avoiding collisions with all the rest. You are free to use any of the techniques we have discussed in the lectures, that you have read about in the literature or can write your own new approach.

For this question, you need to modify your successor function to account for time. In addition, there might be the situation that the search algorithm failed to find a feasible solution as dynamic obstacles block all possible paths. Just return an empty list in this case.

Furthermore, each action and each location for every computed plan need to be collision-free.

Your solution will be still evaluated on 56 instances with a 2 hours time limit and your score in this question will again be computed as the sum of individual path costs (SIC) and compared to the best solution from students (and staff)!

Your f_{score} will be computed in the same way as for Question 1. But there are some differences:

- Here we compute f_{score} for each instance.
- Each instance contains multiple agents.
- Your final points will be $\sum_{i}^{n} f_{score}^{i} \div 56 \times 25$, where i is instance id.

There are up to 25 points available for this question.

Question 3: Challenge (60 points)

You are given sets of start and target locations at the same time. Your job is to route all the trains simultaneously in a way that is collision-free. **But, each agent has an expected arrival time, late arrival will result in a penalty.**

You are free to use any of the techniques we have discussed in the lectures, that you have read about in the literature or can write your own new approach.

Now, as all agents are under your control, you need to make all agents reach their goal locations.

In this question agents may run into malfunctions during execution. The evaluator will call the replan function when a new malfunction occurs. Implement the replan function to properly handle malfunction. Refer to the “Introduction to Flatland” document for details about malfunction and replan function.

Your solution will be still evaluated on 56 instances with different difficulty levels in 2 hours and your score in this question will again be computed as the sum of individual path costs (SIC) and compared to the best solution from students (and staff)!

Your f_{score} computation is similar to Question 1. But there are some differences:

- $p_{score} = (completed\_agents\_SIC + penalty + T_{max} * failed\_agents)/total\_agents$
- The $penalty$ is $2 * total\_delayed\_timesteps$.
- Each instance contains multiple agents.
- Each instance’s f_{score} will refer to a baseline implementation p_{base} and an advanced implementation p_{adv} (or best student solution which ever is better):

  $f_{score} = min(0.5 * \frac{p_{base}}{p_{score}}, 0.5) + max(0.5 * \frac{p_{base} - p_{score}}{p_{base} - p_{adv}}, 0)$
- $\sum_{i}^{n} f_{score}^{i} \div 56 \times 60$, where i is instance id.

There are up to 60 points available for this question.

Report (50 points)

You need to create a report that describes your approach to each of the questions. This includes a textual description of your approaches, why you adopted that particular approach and a thorough discussion along with any supplementary material required (such as pseudo code, images, graphs, tables…).

Bonus Points

We will issue bonus points for students implementing multiple approaches for the same problem, and/or for implementations of algorithms from the scientific literature that are not discussed during tutorials/lectures (e.g., from one of the recommended papers or elsewhere). The size of the bonus depends on how ambitious the implementation is, its effectiveness and the quality of the writeup.

BONUS POINTS CAN MAKE THE DIFFERENCE BETWEEN 'D' and 'HD'

Grades

The total points for the assignment is 150. Part A is worth 15 marks, and Part B is worth 135 marks. A passing grade is 50% of the total points, which is 75 points.

Submission

The assignment is due on **Friday, 12 September 2025, 11:55 PM (end of Week 7)**. You must submit your work in the following two places.

1. Submit your code to the Contest Server:

- Follow the ***FIT5222 Flatland Challenge Submission Instructions*** to submit your code to the contest server.
- Your code will be auto marked on the contest server. You **cannot** receive marks for your code submission if you do not submit to the contest server.
- You must submit your code using your Monash credentials. You **cannot** receive marks for code submitted via a personal account.

2. Submit your code and report to Moodle:

1. Your implementation source code, in a single directory called "src" (you can copy everything in the piglet folder to "src"). Zip the codes directory with file name last_name_student_id_flatland.zip. (For example, Chen_123456_flatland.zip)

2. A scientific report describing your approaches as a single pdf file. Name the pdf as last_name_student_id_report.pdf. (For example, Chen_123456_report.pdf)

3. Submit the two files separately to the Moodle assignment submission page. (**Do not zip the pdf report, and make sure your PDF has selectable text.** Moodle needs a searchable PDF to generate TurnItIn report). Failure to submit a correctly formatted report may incur a penalty.

Generative AI

There are no restrictions on use of generative AI for this assessment task, but students should accomplish their assessments individually. Students **MUST** acknowledge any use of generative AI with an AI statement in their report.

Report Marking Rubric

Criteria	N 0%-49%	P 50%-59%	C 60%-69%	D 70%-79%	HD 80%-100%
Description of your approach (35 points)	Incomplete or insufficient description of the approach and/or pseudo-code	High-level description of the approach and pseudo-code	+ Discussion and algorithmic analysis. E.g., time, space, completeness, optimality.	+ Reflections: advantages and disadvantages of your approach(es)	+ Numerical experiments, analysing the efficiency of your implementations (e.g. on standard benchmarks and vs. appropriate reference algorithms)
Communication skills (15 points)	Hard to follow with no clear narrative. Inadequate or no separation of discussion text into coherent sections. Writing is not accurate or articulate. Inadequate supporting materials. Inadequate or missing referencing.	The writing has a tenuously logical narrative. Some attempt at the expected structural elements (e.g. Intro, conclusion). Writing is not accurate or articulate most of the time. The document has few supporting materials (tables, images, pseudo-code) . The student has attempted to undertake citing and referencing with frequent errors.	The text has a clear logical narrative and expected structural elements (e.g. intro, conclusion). Writing is not accurate or articulate most of the time. There are some supporting materials (tables, images, pseudo-code) but not well integrated with the rest of the text. The student follows the requirements for citing and referencing, with some notable errors.	The writing is well composed and has a clear and logical narrative and is well structured. Writing is generally accurate and articulate. The document has appropriate supporting materials that are well integrated with the rest of the text. The student follows the requirements for citing and referencing, with some minor errors.	The writing is very well composed and has a very clear and logically-formed narrative as a whole. Writing is accurate and articulate. The document is expertly structured in the style of a scientific report, including appropriate supporting materials that clearly improve the quality of associated discussion. The student follows the requirements for citing and referencing.