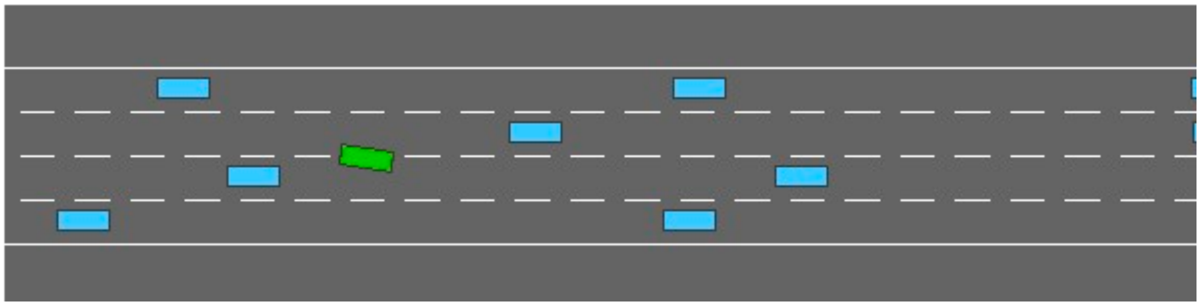
IDC – Reinforcement Learning - Final Project - 2022

The main goal of this final project is to summarize the main topics that we have discussed in the course using some practice and theory, **and especially the second part of course (Deep RL)**

Project Definition:
In this project you will solve several variations of the Highway env.
The Environment: https://github.com/eleurent/highway-env
Additional documentation: https://highway-env.readthedocs.io



Highway is a collection of environments for autonomous driving and tactical decision-making tasks.

Our goal is to help the user car overtake the bot cars on its own in a roadway environment. We train the user car with the help of deep reinforcement learning, the reward function will penalize the user car every time it slows down, every time it crashes into bot car and if there are any bot cars in front of it.

In the following environments we will use the raw pixels as our state space, therefore, it will allow to train CNN Neural Networks.

You will find demonstration of how to use/render the game (random actions) in the template notebook :

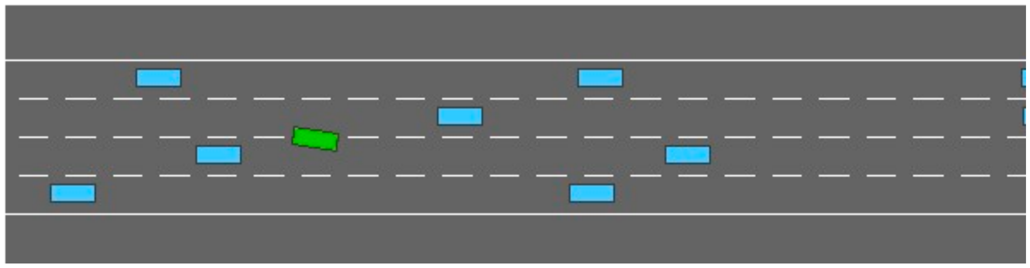
https://colab.research.google.com/drive/1SB4Rlbcw6TnSQOXpS1nITgM7lrfOD974?usp=sharing

After Entering the notebook click on the "File" tab and then on "Save a copy into drive".
You can then work on the project within your personal drive environment.

Read the env. documentation carefully and make sure you understand what is the environment, the observation-space, the action-space, and how are the rewards defined. Also, make sure you understand what's the difference between the versions.

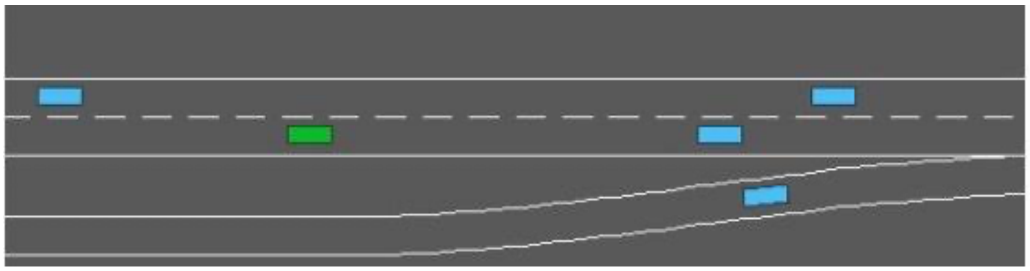
A brief explanation of the different environments you will encounter in this project:

Highway-Env:



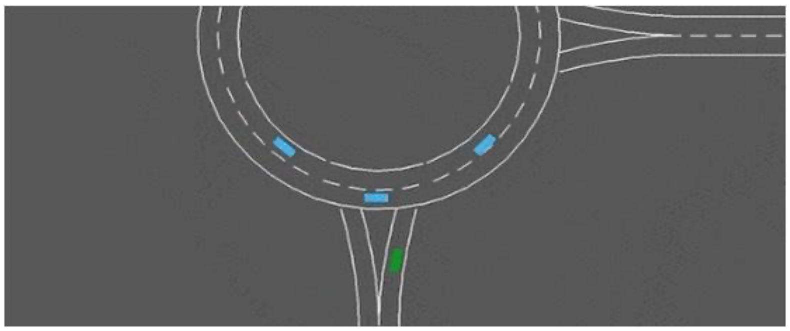
In this environment, the ego-vehicle is driving on a multilane highway populated with other vehicles. The agent's objective is to reach a high speed while avoiding collisions with neighboring vehicles. Driving on the right side of the road is also rewarded.

Merge-Env:



In this environment, the ego-vehicle starts on a main highway but soon approaches a road junction with incoming vehicles on the access ramp. The agent's objective is now to maintain a high speed while making room for the vehicles so that they can safely merge in the traffic.

Roundabout-Env:



In this environment, the ego-vehicle is approaching a roundabout with flowing traffic. It will follow its planned route automatically, but has to handle lane changes and longitudinal control to pass the roundabout as fast as possible while avoiding collisions.

<u>Your goals in this project are</u>

Solve the following environments:

a. EX1 - Highway-Env - Easy - 3 lanes, Easy mode (bot cars), Observations: 4 X 128 X 128 image, with **stochastic transition - where the output action will run correctly at 85% of the cases and in the remain 15% it will act like other action. You will need to implement it. (An example of implementation. You can implement it in other ways.**

   [Click here to see an example in the code of a stochastic environment]

   Here you are asked also to encourage exploration of the agent using method similar to https://pathak22.github.io/noreward-rl/resources/icml17.pdf (Learnt at last lecture)

b. EX2 - Highway-Env - Medium - 6 lanes, Aggressive mode (bot cars), more bots, more density (between bot cars), Observations: 4 X 128 X 128 image.

c. <u>EX3</u> : Super Highway Agent - In this task you will need to train an "Super" agent who knows how to handle any of the following three environments in their different modes (example of how to change configuration can be found here: https://highway-env.readthedocs.io).
   Do not change the observation type (must be gray-level), do not change the input image size).
   These are the three environments that the "Super Agent" needs to know to solve:

   a. Highway-Env
   b. Merge-Env
   c. Roundabout-Env

   You will find a running example of each of the above environments in the Template notebook.

The main goal for all the environments is to “solve” as fast as you can - **this is a competition part.**

Guidelines

1. You are not allowed to use an existing RL lib. Code the algorithm yourself. Remember: The course staff want to see that you are talking the “RL language”. Note that you just know to download an existing lib and run it.
2. You can of course use existing DL platforms such as Keras, Tensorflow, Pytorch.
3. Use sample-based methods
4. At the first stage, only methods and algorithms that we have learnt in the course. In your final report, compare different approaches, different hyper-params, etc. You can use a method that encourages exploration

5. After working and analyzing with the learnt method, you can use advanced method.

6. Use all the knowledge that you got in your course (and more) studies and more papers to solve the challenge.

7. Submit a final report.

8. You need to supply your code as google colab notebooks. Such that the course team can run it.

Note: Even if you did not solve an environment - Supply graphs describing the average rewards on 100 episodes (after finishing the training phase). Note: stop an episode after 500 iterations (if it was not finished)

For each environment, you will save the trained weights and submit them to the submission box (More details on the form of submission below).
In the notebook you will build a function that loads the weights I mentioned above using the Google Colab **files.upload ()** feature, run the environment, print the number of steps the agent took to solve the environment and show a full video of the solution.
You can read about the files feature at the following link:
https://colab.research.google.com/notebooks/io.ipynb

Due Date
Final Project submission is due **July 05th at 11:59pm**.

Submission:
Final submission
Detailed written Report, graphs, Source code links (to google colab) , relevant images and README.
The report should be in the style of a conference paper, including introduction, motivation, related work, etc.
Moreover, For each part of the project show experiments, if you failed in a particular experiment, explain why you think you failed and how you improved from there.
Please do not skimp on explanations and use of visual means to show your results (plots, graphs. etc. ...). Beyond success in the practical part, more important is the way you write your report. Please treat the report as the heart of this project.
You will write the report by using an online collaborative LaTeX editor called overleaf (link below).

An example: https://www.overleaf.com/read/kqxwymfqrmxv

All writing should be your own -- all quotes must be clearly attributed.

The report will be submitted to the submission box. The file name of the report will contained your ID's as following: **report_ID1_ID2.pdf**

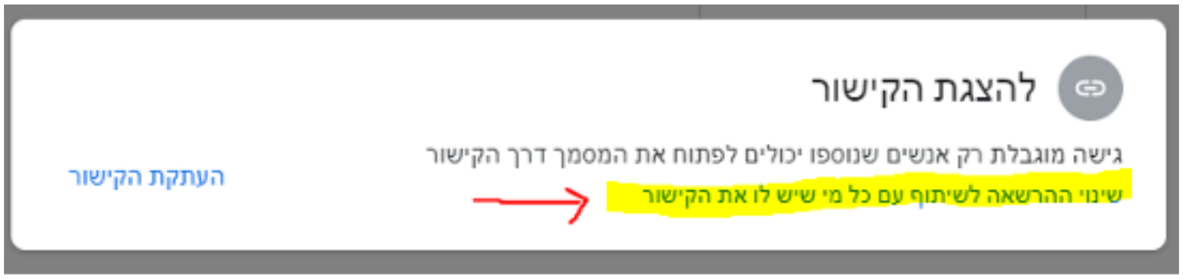
Be very clear about what code you've used from other sources, if any. Clear citations are essential. Failure to credit ideas and code from external sources is cheating.
Make sure you evaluate both the good and bad points of your approach.
Even if you didn't accomplish your goal, evaluate what you did.
Do not forget to include the project title, your name and ID in this file.
Max number of pages: 10 (but you don't have to use them all).

The notebook will be presented in an orderly and clean manner, it will contain separate code cells and text cells that explain the actions performed.
** **Very important** ** - when submitting, the notebooks will contain all the outputs relevant to the training results.

To the submission box you will submit the trained weights you saved while training your model.
You will store the weights in three different folders according to the number of exercises in the project. The folders names will be **ex1_w**, **ex2_w**, **ex3_w**.
You will shrink the folders into a zip file that you will submit in the submission box, the name of the file will be **weights.zip**.

In addition, you will submit an **explainer.README** file that contains instructions and explanations on how to operate the function that performs your weight loading and other relevant details that need to be known to those who want to use your notebook.

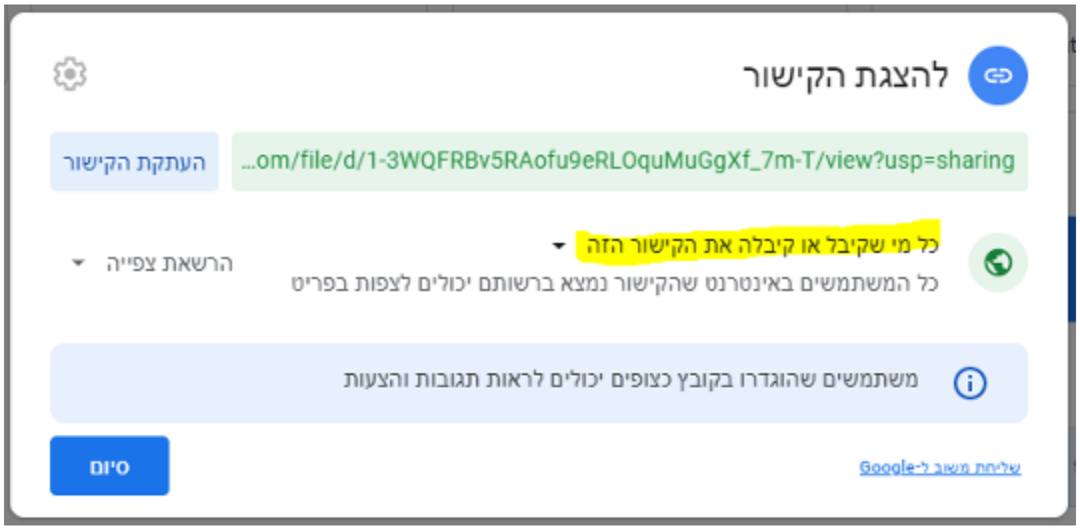
You will share the notebook from your "Google Drive" account, it can be shared with anyone who holds the link as follows:



להצגת הקישור

גישה מוגבלת רק אנשים שנוספו יכולים לפתוח את המסמך דרך הקישור

שינוי ההרשאה לשיתוף עם כל מי שיש לו את הקישור



להצגת הקישור

...om/file/d/1-3WQFRBv5RAofu9eRLOquMuGgXf_7m-T/view?usp=sharing

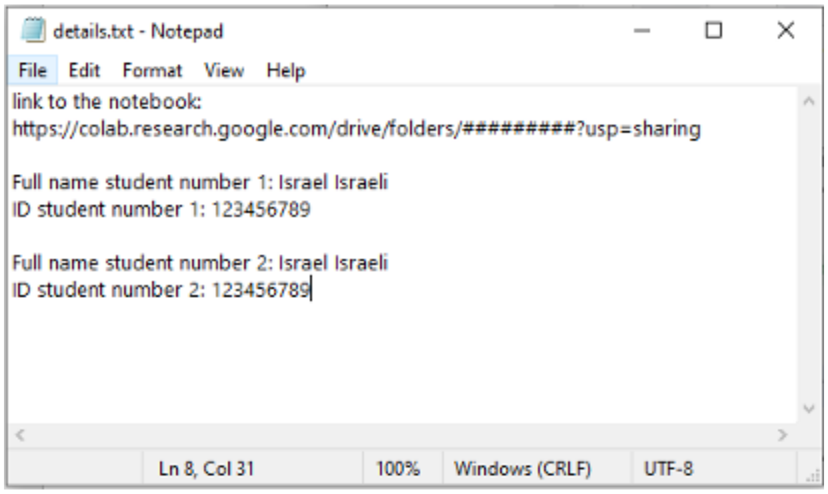
כל מי שקיבל או קיבלה את הקישור הזה

כל המשתמשים באינטרנט שהקישור נמצא ברשותם יכולים לצפות בפריט

הרשאת צפייה

משתמשים שהוגדרו בקובץ כצופים יכולים לראות תגובות והצעות

Submission will be done in pairs when only one of the partners submits the assignment to the submission box.
Enter your details in the text file named **submit.txt**, the address of your notebook, the names and ID of the two partners, as follows:



```
link to the notebook:
https://colab.research.google.com/drive/folders/#########?usp=sharing

Full name student number 1: Israel Israeli
ID student number 1: 123456789

Full name student number 2: Israel Israeli
ID student number 2: 123456789
```

To sum up, in the submission box, submit the following files:

1. submit.txt
2. report_ID1_ID2.pdf
3. weights.zip
4. explainer.README

Team size
The project will be performed in groups of 2 students.

Academic Integrity
Team/student may not copy code from other team/students. Copying answers or code from other students for a project is a violation of the university's honor code and will be treated as such. All suspicious activity will be reported to the Department Head and the university authorities.

Giving code to another student is also considered a violation. Students are responsible for protecting their work from copying.

If you build some of your code on existing work and utilize existing code (your own or code found on the web), you must give proper attribution to all existing work that you build on and make clear what your new contribution is. Any unattributed or uncited work that you use will be considered a breach of academic honesty and dealt with according to the course policy in the syllabus.

In any matter related to the project, even administrative, you can contact Aviv German at:
aviv.german@post.idc.ac.il