Final Projects

Please pay attention to the following tips:

- The project deadline is 7th of Bahman
- Except for the Chess project, which is done individually, other projects can be done in the form of two-membered groups (Maximum)
- All team members should be aware of the details of the project
- Your project includes an in-person presentation, which contains 30% of your overall score.
- Your implementation, results and your project's functionality include 30% of your total score.
- Also, send a document about your implementation and your approach. This documentation includes 40% of your total score.
- For Reinforcement Learning projects, be aware that you may need a few days for the agent training phase, so plan ahead to avoid problems.
- Enter your selected projects in https://forms.gle/xWnZDQzL49mK3zzy7 form.
- Feel free to ask your questions from the teaching assistants.
- No need to mention but in case we find anything suspicious with the project codes -- containing any similarity to the codes available on the internet -- your score will be "Zero".

1. Implementing intelligent agents for board-games:
    - Python-chess
        - This is a pure Python chess library with move generation, move validation and support for common formats.
        - For more information visit: https://pypi.org/project/python-chess/
        - The base logic of your algorithm for solving it must be **MinMax**
        - You can only use the **Standard** mode.
        - Download "chess.zip" which is available on the course's website

Gym Environment:

Projects in this section include implementing Reinforcement Learning agents using Gym for different problems. Gym is a tool for Implementing Reinforcement Learning agents. For more information, and to install it, please visit: https://gym.openai.com/docs/

2. Boxing: In this project, you are going to implement an agent using RL methods for Boxing game on Atari. For more information about game logic, visit: https://en.wikipedia.org/wiki/Boxing_(Atari_2600)
You should choose one of the following projects:

- ❏ https://gym.openai.com/envs/Boxing-v0/: In this environment, the observation is an RGB image of the screen, which is an array of shape (210, 160, 3) Each action is repeatedly performed for a duration of *k* frames, where *k* is uniformly sampled from {2,3,4}.
- ❏ https://gym.openai.com/envs/Boxing-ram-v0/: In this environment, the observation is the RAM of the Atari machine, consisting of (only!) 128 bytes. Each action is repeatedly performed for a duration of *k* frames, where *k* is uniformly sampled from {2,3,4}.

3. Riverraid: In this project, you are going to implement an agent using RL methods for Riverraid game on Atari. For more information about game logic, visit: https://en.wikipedia.org/wiki/River_Raid
You should choose one of the following projects:

- ❏ https://gym.openai.com/envs/Riverraid-v0/: In this environment, the observation is an RGB image of the screen, which is an array of shape (210, 160, 3) Each action is repeatedly performed for a duration of *k* frames, where *k* is uniformly sampled from {2,3,4}.
- ❏ https://gym.openai.com/envs/Riverraid-ram-v0/: In this environment, the observation is the RAM of the Atari machine, consisting of (only!) 128 bytes. Each action is repeatedly performed for a duration of *k* frames, where *k* is uniformly sampled from {2,3,4}.

4. Tennis: In this project, you are going to implement an agent using RL methods for a Tennis game on Atari.

You should choose one of the following projects:

- ❏ https://gym.openai.com/envs/Tennis-v0/: In this environment, the observation is an RGB image of the screen, which is an array of shape (210, 160, 3) Each action is repeatedly performed for a duration of *k* frames, where *k* is uniformly sampled from {2,3,4}.
- ❏ https://gym.openai.com/envs/Tennis-ram-v0/: In this environment, the observation is the RAM of the Atari machine, consisting of (only!) 128 bytes. Each action is repeatedly performed for a duration of *k* frames, where *k* is uniformly sampled from{2,3,4}.

5. CarRacing: In this project, you are going to implement an agent using RL methods for Car Racing.
This is the OpenAI link: https://gym.openai.com/envs/CarRacing-v0/

6. BipedalWalker: In this project, you are going to implement an agent using RL methods for BipedalWalker.
This is the OpenAI link:https://gym.openai.com/envs/BipedalWalkerHardcore-v2/