**ECEN 689: RL: Reinforcement Learning**
**Assignment 2**

---

In this homework you will implement Value Iteration, Policy Iteration and Q Learning on the frozen lake environment from openAI Gym (Link).

# Instructions

Skip 1 and 2 if you have gym already installed.

1. Ensure you are using Python 3.5+ (for Gym) and have the following libraries/dependencies: time, seaborn, matplotlib.pyplot, numpy, math, random
   (a) People working with Windows can find solace using Anaconda (Link).
   (b) People working with Mac OS or any Linux OS have an inbuilt python at their disposal. So, just follow the installation procedure given in OpenAI website.
2. Refer (Link) for installing gym.
3. This homework comes with a helper file, it consists of a few base codes that will be useful.
4. **Value Iteration**
   - Implement value iteration on the frozen lake environment. For this, you will have to make use of the transition kernel.
   - Plot $||V_k - V_{k-1}||_2$. Where $V_k$ is the value during the $k^{th}$. This shows the convergence of your algorithm.
   - Compute the optimal policy.
   - Use the *fancy_visual* function, which is available in the helper file to plot the heat maps of the optimal policy and value function. **Note that this function also prints the policy and value function.**
   - Compute and print the optimal Q function.
5. **Policy Iteration**
   - Implement Policy iteration, plot $||V_{\pi_k} - V_{\pi_{k-1}}||_2$ and heat map. Also print the optimal $Q$ function.
6. **Tabular Q Learning**
   - Implement tabular Q learning (not Deep Q Learning) on the frozen lake environment. Here you will not use the knowledge of the transition probability function. The helper file has some basic examples on sampling the environment.

- Plot $G_k$, where $G_k$ is the cumulative reward obtained in episode $k$.
- Plot $||Q_k - Q^*||_2$ where $k$ is the episode number and $Q^*$ is the optimal Q function obtained from 4 or 5.
- Plot both the plots again using a sliding window averaging.
- Compute the final policy and plot the heat maps.
- *Hint: Try various functions to decay exploration and learning rate. This will help you converge in lesser number of episodes.*

7. Create three separate IPython Notebooks (.ipynb) these files show contain all your plots and outputs. Name the files as follows,
   - VI.ipynb for value iteration (Should consists of 2 plots and the optimal Q function, Value Function and Policy)
   - PI.ipynb for policy iteration (Should consists of 2 plots and the optimal Q function, Value Function and Policy)
   - QL.ipynb for Q learning (Should consists of 5 plots and the final Q function, Value Function and Policy)
8. While it is a good practice to print out intermediate steps to debug your code, please refrain from doing so in the final submission.
9. You may choose to make your code modular, in that case please submit all relevant files.
10. Your submission should consists of the following files,
    - All the files in 7
    - HTML versions of all the files in 7
    - All relevant files if 9 applies to you
11. Zip all files in 10 and name it **LastName_FirstName.zip** and submit it on ecampus.