Group Project

Upload Your Group Information

- https://forms.gle/F57obMDWrQDpNkdu6

Part 1: Install and Run

- https://github.com/ccslab1/Group-Project

- **Install Python 3 and Gymnasium**
  Follow the setup guide available on *eLearn* to install Python 3 and the Gymnasium environment on your computer.
- **Verify Your Installation**
  Run **Part 1** of the project to confirm that your installation is working correctly.
- **Development Environment**
  You may use **any IDE** of your choice to develop your project

Part 2: Frozen Lake

Run the Frozen Lake:

- Goal:
  - Revise the sample code to achieve a **consistent success rate > 0.70** on **without changing**: `num_episodes` and `max_steps_per_episode`
  - You may **only tune**: ‘min_exploration_rate (currently is 0)’ and ‘epsilon_decay_rate’
- You should demonstrate the agent’s performance.
  - 1. **Train** with your tuned exploration settings (no change to episodes/steps).
  - 2. **Evaluate** success rate over a **fixed evaluation run** (e.g., 500–1000 test episodes at ε≈0).
  - 3. **Report**:
  - - Final **success rate** (wins/episodes)
  - - (Optional) a short **moving-average curve** over training episodes
  - **Success definition:** An episode counts as success if it reaches the goal (Gymnasium returns reward `1.0` at termination).
- Keys:
  - env = gym.make("FrozenLake-v1", render_mode="ansi")
  - print(env.render())

Part 3: Choose Your Own Adventure

- Goal:
  - Design and implement a project that demonstrates OOP principles (such as abstraction, inheritance, and polymorphism) while extending your understanding and practical experience with programming
- Suggestions:
  - Use a Classic Control Environment: e.g. a classic control problem available in Gymnasium (e.g., CartPole, MountainCar, or Acrobot).
    - Implement classes for the agent, environment interaction, and training loop.
    - Emphasize modularity and code reusability.
  - Create a Custom Environment: Design and implement your own simple environment to explore or visualize concepts.
    - Define a custom Env class following the Gymnasium interface (reset, step, render).
    - Test how your agent adapts and learns from your designed state and reward structure.

Project Evaluation Policy

Oral Demo

- Part 1 (1%)
- Part 2 (19%)
- Part 3 (80%)
- Please upload all your source code to your public GitHub repository