**Developing Reinforcement Learning Agents for Super Mario with Optional Imitation Learning**

**Overall Objective: The goal of this project is to develop two distinct reinforcement learning (RL) agents for playing Super Mario Bros using OpenAI Gym's Super Mario environment. These agents will focus on two specific tasks:**

1. **Speed Run Agent: This agent will aim to complete levels as quickly as possible.**
2. **Coin Collecting Agent: This agent will aim to maximize the number of coins collected while playing.**

**We will also explore using Imitation Learning (IL) as a potential enhancement to these RL models, though it may be optional depending on progress.**

**Detailed Goals:**

1. **Speed Run Agent:**
   * **Objective: Train an RL agent to complete each stage of Super Mario as fast as possible.**
   * **Key Performance Metric: Time to completion of a level, minimizing deaths.**
   * **Reward Structure: The agent will be rewarded for moving to the right (x\_pos) and penalized for time spent and for dying.**
2. **Coin Collecting Agent:**
   * **Objective: Train an RL agent to maximize coin collection while minimizing deaths.**
   * **Key Performance Metric: Total coins collected per level, ensuring survival.**
   * **Reward Structure: The agent will receive rewards for collecting coins and be penalized for taking too much time or dying.**
3. **Imitation Learning (Optional):**
   * **Objective: Enhance the RL agents by using human gameplay data to bootstrap the learning process. This will involve recording human gameplay and using behavior cloning to teach agents to imitate human strategies.**
   * **Challenges: We need to ensure that human gameplay data is properly captured, and the imitation learning models should generalize well from human data to reinforcement learning.**

**Things to Explore/Find Out:**

1. **RL Algorithm Optimization:**
   * **Explore Q-learning and Double DQN for both agents, determining which variant gives the best results for speed runs and coin collection.**
   * **Investigate optimal reward shaping: balancing between rewarding progress (x\_pos), penalizing time, and penalizing deaths.**
   * **Fine-tune hyperparameters (e.g., learning rates, epsilon decay, replay buffer size) to improve performance.**
2. **Stage-Specific Agent Training:**
   * **Explore the idea of training one model per stage for both speed running and coin collection. We need to understand if stage-specific agents can generalize better than a single model across all levels.**
3. **Reward Structures:**
   * **We need to experiment with various reward structures to ensure that agents focus on the right objectives (speed vs. coins), without conflicting incentives.**
4. **Handling Exploration vs. Exploitation:**
   * **Finding the right balance between exploration (trying new actions) and exploitation (maximizing rewards from known actions) is key to the success of both agents. We need to explore how epsilon-greedy policies impact both agents.**
5. **Imitation Learning (IL):**
   * **If used, how well can the agents learn from human gameplay data?**
   * **Will imitation learning make the RL agents more efficient during training, or will they struggle to generalize beyond human strategies?**
   * **Explore whether IL is worth keeping, or if we should drop it to focus purely on RL techniques.**

**Things We Are Unsure About:**

1. **Imitation Learning Feasibility:**
   * **We are unsure if the recorded human gameplay data will be diverse enough to provide a significant boost to agent performance.**
   * **The quality of the data, and the ability to effectively train imitation models that generalize well, is still uncertain. There’s a possibility that the IL models could become overfitted to human behavior.**
2. **Stage-Specific Agents vs. General Agent:**
   * **We don’t know yet if training a single general agent across all stages is feasible or if the agents will perform better when trained specifically for individual stages.**
3. **Reward Function Design:**
   * **Getting the right reward structure for speed and coin collection will require experimentation. Reward functions that are too simple may result in suboptimal performance, while overly complex reward functions may slow down learning.**
4. **Time Constraints:**
   * **Given the limited time (6 weeks), we are unsure if we can effectively implement both RL and IL while producing robust models. There’s a chance we might drop the IL part if it proves to be too time-consuming or less impactful than expected.**

**Possible Changes to the Goal:**

1. **Dropping Imitation Learning:**
   * **If Imitation Learning proves to be too complex or doesn’t provide enough benefit within the timeframe, we may drop this aspect and focus entirely on reinforcement learning for both the speed run and coin collection agents.**
2. **Focusing on One Agent Type:**
   * **Depending on progress, we may focus on just one agent type (e.g., speed run or coin collection) and deliver a more polished and optimized model rather than spreading resources thinly across both.**
   * **Training two reinforcement learning agents for Super Mario, one focused on speed runs and the other on collecting coins, can be achieved with minimal additional effort once the first agent is trained. Both agents will share the same environment and model structure, so the main distinction lies in their reward functions. The speed run agent will be rewarded for completing levels quickly, while the coin collection agent will be rewarded for gathering as many coins as possible. By simply adjusting the reward criteria, we can efficiently train both agents without needing to significantly alter the underlying implementation.**

**Summary of the Current Focus:**

* **Build two distinct RL agents: one for speed running, one for coin collecting.**
* **Explore the utility of Imitation Learning by collecting human gameplay data to potentially jumpstart training.**
* **Experiment with different reward structures to balance the focus between progress, time, and survival.**
* **Iterate quickly to meet key milestones, with flexibility to drop less impactful aspects like IL if needed to ensure high-quality RL agents by the final deadline.**

**Week-by-Week Plan:**

**Week 10 (Oct 29 - Nov 5): Project Proposal and Setup**

**Tasks:**

* **Finalize Project Proposal**:
  + Define objectives, scope, and methodology for the two RL agents.
  + Include the role of imitation learning for the speed run and coin-collecting agents.
* **Set up the Environment**:
  + Install **gym-super-mario-bros** and the required dependencies.
  + Familiarize yourself with SuperMarioBros-v0 environment details such as x\_pos, coins, life, and time to determine appropriate reward structures for speed and coin agents.
* **Simulator Setup for IL**:
  + Build a basic **simulator** that allows human gameplay to record states, actions, and rewards (use a simple GUI to record the gameplay data).

**Deliverable**: Project Proposal, Environment Setup, and Simulator framework.

**Week 11 (Nov 6 - Nov 12): Initial Agent Development & Recording Gameplay**

**Tasks:**

* **Speed Run Agent (Initial Version)**:
  + Develop the first version of the **speed run agent** focusing on maximizing x\_pos and minimizing time left. Use a simple reward function:
    - Reward for moving right: +1 \* delta\_x
    - Penalty for loss of life: -15
    - Penalty for time spent: -1 for each tick.
* **Coin Collection Agent (Initial Version)**:
  + Develop the first version of the **coin-collecting agent** with a reward function focused on:
    - Reward for collecting coins: +1 \* coin\_delta
    - Penalize time with a smaller weight.
* **Recording Human Gameplay**:
  + Have multiple team members or volunteers play the game via the simulator and **record their gameplay** (states, actions, rewards) to start building the dataset.

**Deliverable**: Initial RL agent models (speed and coin), initial recorded gameplay data.

**Week 12 (Nov 13 - Nov 19): Fine-Tuning Agents & Imitation Learning (IL) Preparation**

**Tasks:**

* **Fine-tune the Reward Functions**:
  + **Speed Agent**: Adjust the weight of x\_pos gain vs. time left to incentivize fast completion without dying.
  + **Coin Agent**: Modify reward based on maximizing coins collected without death and without getting stuck.
* **Begin Imitation Learning Preparation**:
  + Start processing the **recorded human gameplay data** into a structured format (state-action pairs).
  + Implement a basic behavior cloning algorithm to initialize the IL agent for both speed runs and coin collecting.

**Deliverable**: Improved RL models, processed gameplay data ready for IL.

**Week 13 (Nov 20 - Nov 26): Imitation Learning (IL) and Further Agent Improvement**

**Tasks:**

* **Implement Imitation Learning**:
  + Train a model using **behavior cloning** from the human data.
  + Use the trained model to initialize the policy for both the speed run and coin agents.
  + Fine-tune the agents using **reinforcement learning** to refine policies from the human baseline.
* **Agent Testing**:
  + Run multiple simulations of the agents on specific stages.
  + Log the **performance metrics** (e.g., time to completion, number of coins collected, survival rate).

**Deliverable**: Imitation learning models, fine-tuned agents, initial performance reports.

**Week 14 (Nov 27 - Dec 3): Evaluation & Final Adjustments**

**Tasks:**

* **Evaluate Performance**:
  + Compare the **speed agent**’s performance across different stages: evaluate how fast it completes and the average number of deaths.
  + Compare the **coin agent** across stages: total coins collected and time taken.
* **Run Comparative Experiments**:
  + Compare RL agents trained from scratch vs. those enhanced via imitation learning.
* **Fix Bugs and Improve Generalization**:
  + Test agents on unseen levels and further adjust hyperparameters.

**Deliverable**: Refined models and final evaluation reports.

**Week 15 (Dec 4 - Dec 9): Final Report, Code, and Model Submission**

**Tasks:**

* **Final Training and Model Saving**:
  + Run final training iterations to solidify the model performance.
  + Save trained models for both RL agents and IL-enhanced agents.
* **Prepare Final Report**:
  + Document the entire process: problem setup, methodology, RL agent design, imitation learning, results, and future work.
  + Prepare all the code and datasets for submission.

**Deliverable**: Final project report, trained models, and code.

**Week 16 (Dec 10): Presentation and Demo**

**Tasks:**

* **Prepare for the Final Presentation**:
  + Summarize results, challenges, and the overall methodology in a 10-minute presentation.
  + If possible, **demo** the agents in real-time or through a recorded video showcasing the agents on different Mario stages.

**Deliverable**: Final presentation and demo.