RL Agent Playground – README

# Project Title

Reinforcement Learning Agent Playground with DQN and Random Agents

# Developed With

• Python  
• PyTorch  
• Gymnasium  
• Gradio  
• ImageIO / MoviePy

# Objective

This project demonstrates the power of Deep Q-Networks (DQN) in solving classic reinforcement learning environments like CartPole-v1 and MountainCar-v0. The interactive UI allows users to compare the DQN agent’s performance with a Random agent.

# Features

• Train a DQN agent for MountainCar-v0  
• Preload trained models for CartPole-v1 and MountainCar-v0  
• Compare DQN vs Random agent visually  
• Watch agent gameplay as a video  
• Interactive interface using Gradio

# Project Structure

RL\_Agent\_Playground/  
├── dqn\_mountaincar.pth # Trained model weights for MountainCar  
├── dqn\_cartpole.pth # Trained model weights for CartPole (optional)  
├── main.py # Full training + Gradio app code  
├── README.docx # This documentation  
├── requirements.txt # Python dependencies (optional)

# How to Run

1. Install Dependencies:

pip install gymnasium torch torchvision imageio moviepy gradio matplotlib pygame

2. Train MountainCar DQN (optional):

If dqn\_mountaincar.pth doesn’t exist, training is initiated in main.py:

train\_mountaincar\_dqn()

3. Launch Gradio Interface:

python main.py

# Gradio Interface Usage

• Game: Choose between CartPole and MountainCar  
• Agent Type: Select either DQN or Random  
• Episodes: Number of episodes to run  
• Output: A video of the agent playing the game and its total score

# Model Architecture (QNetwork)

self.fc1 = nn.Linear(state\_size, 128)  
self.fc2 = nn.Linear(128, 128)  
self.fc3 = nn.Linear(128, action\_size)  
Activation: ReLU  
Loss: MSELoss  
Optimizer: Adam

# Learning Outcomes

• Understand how DQNs work in classic control environments.  
• Build an interactive RL demo with Gradio.  
• Visualize and debug RL performance using video.

# Future Enhancements

• Add other environments like LunarLander or Acrobot  
• Support training via the Gradio UI  
• Add charts for episode scores and Q-values  
• Integrate leaderboard for best performance scores

# Credits

• Gymnasium (OpenAI)  
• PyTorch  
• Gradio  
• ImageIO  
• MoviePy