Brief Summary of DQN

Implement DQNAgent class:

- Initialize agent with ID, state/action dims, seed; set hypers (`gamma=0.99, epsilon=1.0→0.01@0.995/episode, LR=1e-4, batch=128, buffer=50k, target update=500 steps`); build linear net `[2→128→128→64→15]`; use Adam, deque memory.

- Store transitions in `remember()`.
- Select action epsilon-greedy in `select_action()`.
- Train in `replay()`: sample batch, compute Double DQN targets via Bellman $y = r + \gamma(1 - d) \max_{a'} Q_{target}(s', a')$, Huber loss, clip grads@1.0, Adam update; hard-copy target every 500 steps.

- Decay epsilon in `update_epsilon()`; save/load checkpoints.

Conversation: Reviewed dqn.py vs. papers (Mnih 2015, van Hasselt 2016, etc.); matched theory/impl (Double DQN, replay, etc.); tabulated hypers/sources (mostly Kastius 2021); elaborated equations (Bellman, Huber), hypers, updates in bullet/math format.