Q Learning Agent for Pac-Man

Done By: Suha Ahmed

Q learning:

This coursework implements a tabular Q-Learning agent that learns state-action values through episodic interaction with the game. Training is performed in train() by running short simulated episodes against StarterGhosts, using an ε-greedy policy for exploration: with probability ε choose a random legal action, otherwise choose argmax_a Q(s,a).

Configuration:

- **Agent:** QLearningAgent

- **Training episodes executed in implementation:** 5

- **Learning rate (α):** 0.1

- **Discount factor (γ):** 0.9

- **Initial exploration rate (ε):** 0.3

- **Epsilon decay used:** epsilon = Math.max(0.05, epsilon * 0.9) applied after each episode

- **Ghost controller during training:** StarterGhosts (moving)

- **State representation:** tabular GameState abstraction (GameState.fromGame / StateGenerator)

- **Execution entry:** QLearningAgent.train() invoked from constructor; Executor.runExperiment(new QLearningAgent(), new StarterGhosts(), 50) for evaluation

Methods implemented:

train():

- Runs episodic Q-learning loop for 100 training episodes
- Uses ε-greedy action selection (random with probability ε, otherwise argmax_a Q(s,a))
- Simulates transitions using game.advanceGame(action, ghosts.getMove())
- Computes reward as r = score(s') - score(s)

- Updates Q(s,a) using the temporal difference update rule
- Decays exploration rate after each episode

getBestAction(GameState state):

- Returns argmax_a Q(s,a) for exploitation
- Fallback to MOVE.NEUTRAL if no Q-values exist

getMaxQ(GameState state):

- Helper method returning max_a Q(s,a) for TD target calculation

getMove(Game game, long timeDue):

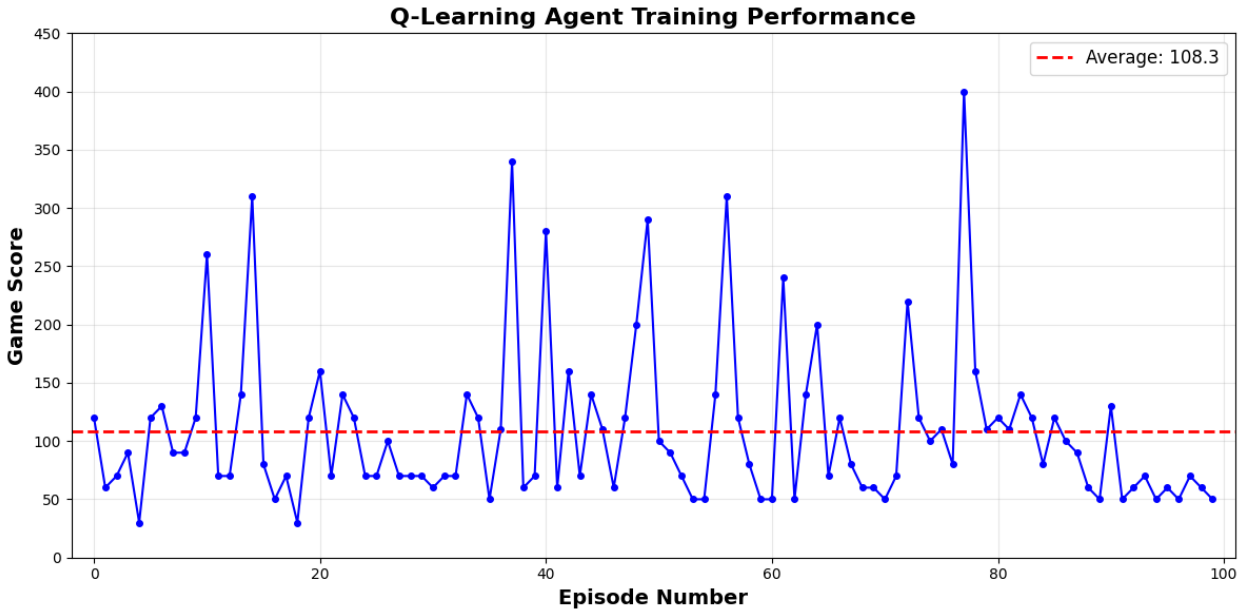
- Runtime action selection using learned Q-table
- Maps current game state to GameState and returns best action

Results:

Training Configuration

- **Number of training episodes:** 100
- **Opponent during training:** StarterGhosts (moving ghosts)
- **Learning rate (α):** 0.1
- **Discount factor (γ):** 0.9
- **Initial exploration rate (ε):** 0.3 (decayed to 0.05)

Training Performance Plot:



Q-Learning agent score progression over 100 training episodes against StarterGhosts. The red dashed line indicates the average training score of 108.3 points.

Training Analysis

The training performance shows high variability across episodes, with scores ranging from 30 to 400 points. Key observations:

- **Average training score:** 108.3 points
- **Best episode:** Episode 77 achieved 400 points
- **Worst episode:** Episodes 4, 18, 89, 91, 94, 96, 99 scored only 50 points
- **Trend:** No clear upward learning trend observed; scores remain highly variable even in later episodes (90-100)

The lack of convergence suggests that 100 episodes are insufficient for the agent to learn a stable, optimal policy against moving ghosts. The high variance indicates the agent continues to encounter novel or challenging situations it has not adequately learned to handle.

Evaluation:

Evaluation configuration:

- **Number of test games:** 50
- **Opponent:** StarterGhosts (moving ghosts)
- **Agent behavior:** Pure exploitation (ε = 0, using learned Q-table only)
- **Execution mode:** Non-visual batch mode
  using Executor.runExperiment()

Evaluation Scores:

Per-game scores (50 games):

70, 60, 60, 70, 50, 70, 50, 50, 60, 60,

50, 50, 70, 60, 70, 70, 70, 50, 70, 70,

50, 50, 50, 60, 70, 50, 50, 50, 70, 60,

50, 70, 50, 50, 50, 50, 70, 50, 60, 50,

50, 50, 70, 50, 50, 70, 70, 50, 70, 70

Statistical Summary:

- **Average score:** 58.8 points
- **Minimum score:** 50 points
- **Maximum score:** 70 points
- **Mode:** 50 points (25 games, 50%)

Win/Loss Statistics:

Based on game completion analysis:

- **Wins:** 0/50 (0%)
- **Losses:** 50/50 (100%)

A "win" is defined as Pac-Man clearing all pills without being caught. A "loss" occurs when Pac-Man is caught by a ghost. The consistently low scores (50-70 points) indicate Pac-Man was caught very early in all 50 games, resulting in a 0% win rate.

Score Distribution:

Score	Frequency	Percentage
70	16	32%
60	9	18%
50	25	50%

Observation:

The Q-Learning agent successfully implements the core algorithm but fails to achieve competitive performance against moving ghosts, scoring an average of 58.8 points with a 0% win rate over 50 evaluation games.