Artificial Intelligence

Reinforcement Learning Agent for the Snake Game

Our Team

Faryal Bahawi

Jatin Nabhoya

Course

Artificial Intelligence

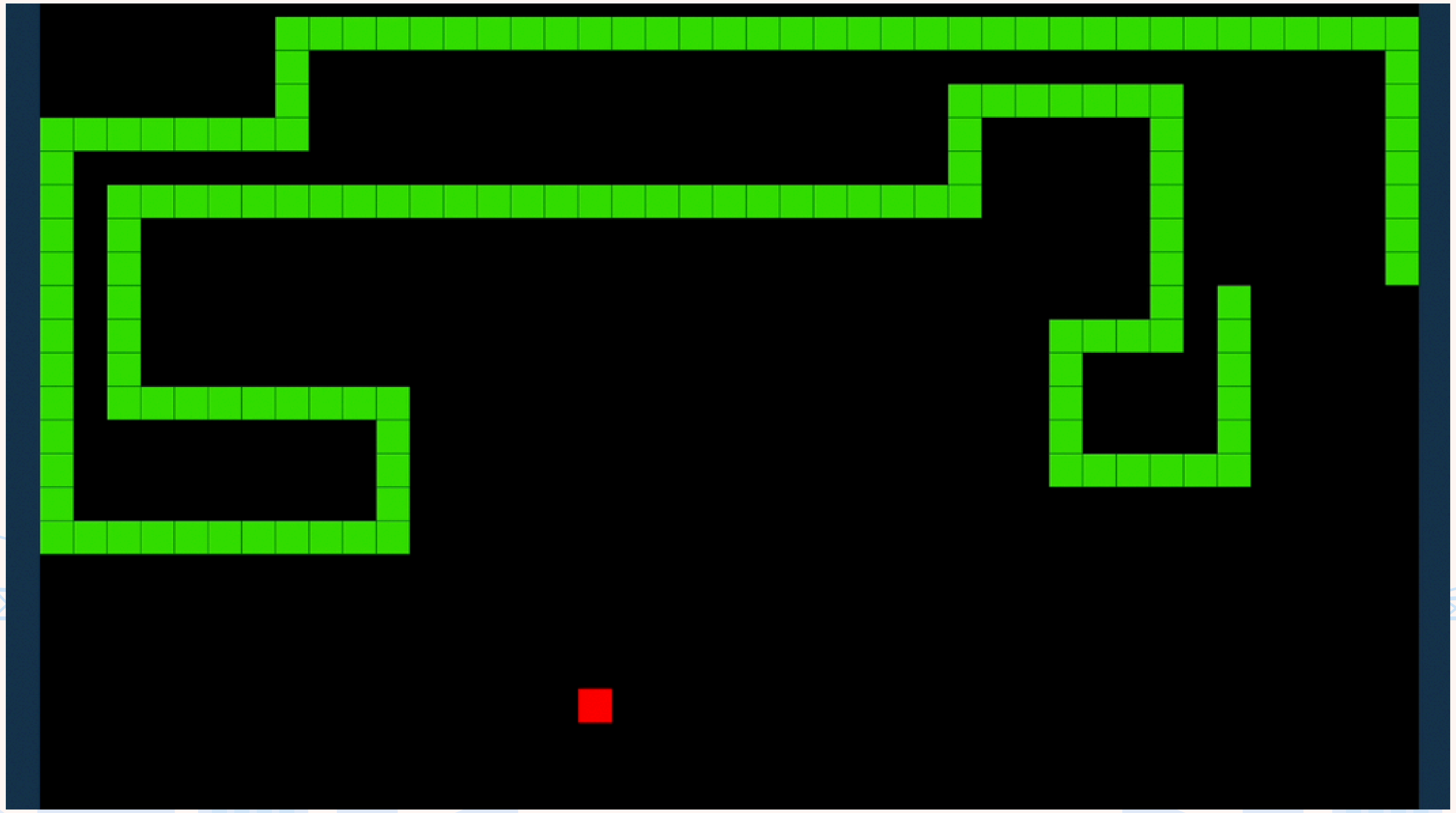
Instructor

Dr. Vahid Behzadan

Dec 2025

Project Overview

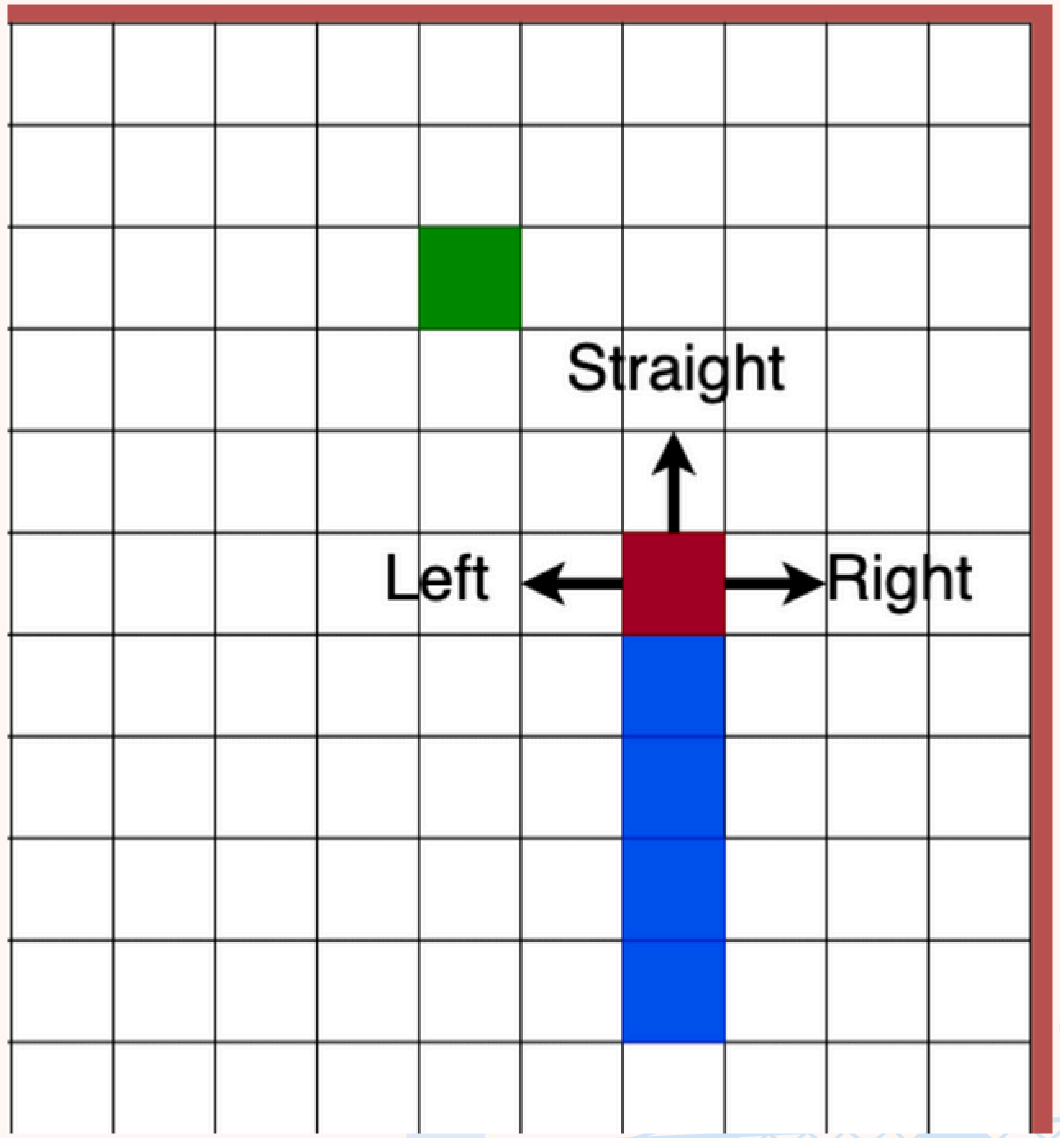
- Built a custom 10×10 Snake game environment.
- Implemented Q-learning agent from scratch.
- Created baseline random policy for comparison.
- Added a graphical Pygame game with Human & AI modes.
- Integrated sound effects (eat + game over).
- Goal: understand RL behavior and visualize learned strategy.



Environment & Rewards

State Representation:
(snake_x, snake_y, food_x, food_y)
Actions: Up, Down, Left, Right
Reward Function:

- +1 → eat food
- −1 → hit wall
- +0.1 → move closer to food
- −0.1 → move farther from food
- 0 → normal move
- Episode ends on wall collision



Q-Learning Method

Update Rule:

$$Q(s,a) = Q(s,a) + \alpha\ (\text{reward} + \gamma\ \max(Q(s')) - Q(s,a))$$

Hyperparameters:

- Learning rate $\alpha = 0.1$
- Discount $\gamma = 0.9$
- Exploration $\varepsilon = 0.1$
- Episodes = 2000

Q-table stored using:

- defaultdict(float)

Results: Random Baseline vs RL Agent

Random Baseline (200 episodes)

- Reward: −0.83
- Foods eaten: 0.17
- Survival: 30 steps

Q-learning Agent (last ~100 episodes)

- Reward: +0.82
- Foods eaten: 1.27
- Survival: 12 steps
- Trained using 5000 episodes (Q-table generated)
- Final Q-table size ~ thousands of learned state-action pairs

Interpretation:

RL agent successfully moves toward food instead of wandering.
Baseline behaves randomly and almost never finds food.

Pygame Visual Game

Features Added

- Human control (WASD / Arrow keys)
- AI-controlled gameplay using saved Q-table
- Wall-collision logic
- Smooth grid rendering
- Sound effects:
    - eat_drink.wav
    - game_over.wav

Game Modes

- SPACE → Start
- H → Human mode
- A → AI mode
- ESC → Quit

Conclusion

- The agent learned a clear strategy: move toward food.
- Q-learning outperformed the random policy significantly.
- Custom environment helped understand RL fundamentals.
- The trained Q-table captures the agent's learned policy and can be extended into future versions of the graphical game.
- Pygame visualization made learning behavior obvious and interactive.
- Project demonstrates successful integration of AI + game development + RL theory.

Thank You