1 Snake Game Mechanics

The Snake game is implemented using the Q-Learning reinforcement learning algorithm. The agent (the snake) learns to navigate the grid to eat food while avoiding collisions with the walls and its own tail.

1.1 State Representation

The state of the environment is defined by 8 boolean variables relative to the snake's head:

- **Danger Detection**: Is there an obstacle (wall or body segment) immediately Up, Down, Left, or Right? (4 variables)
- **Food Direction**: Is the food located Up, Down, Left, or Right relative to the head? (4 variables)

1.2 Actions and Rewards

The agent can take four discrete actions: Move Up, Down, Left, or Right. The learning process is driven by a reward system:

- **Positive Rewards**: +50 for eating food, +1 for moving towards the food.
- **Negative Rewards**: -100 for a collision (game over), -1.5 for moving away from food, -0.1 per step (to encourage efficiency).

The Q-value is updated using the standard Bellman equation:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s', a'))$$

where $\alpha = 0.1$ is the learning rate and $\gamma = 0.9$ is the discount factor.

2 Results Analysis

Figure 1 illustrates the agent's performance over time. The X-axis represents the generation (episode number). The Blue line indicates the game Score, while the Red line shows the Total Reward accumulated per episode.

Observations:

- **Initial Phase**: In the early generations, the agent acts mostly randomly due to the exploration rate ($\epsilon = 0.1$) and an uninitialized Q-table. This results in frequent collisions, low scores, and often negative rewards.
- **Learning Phase**: As the number of generations increases, the agent learns the optimal policy. We expect to see an upward trend in the Reward curve, indicating the snake is surviving longer and finding food more efficiently.

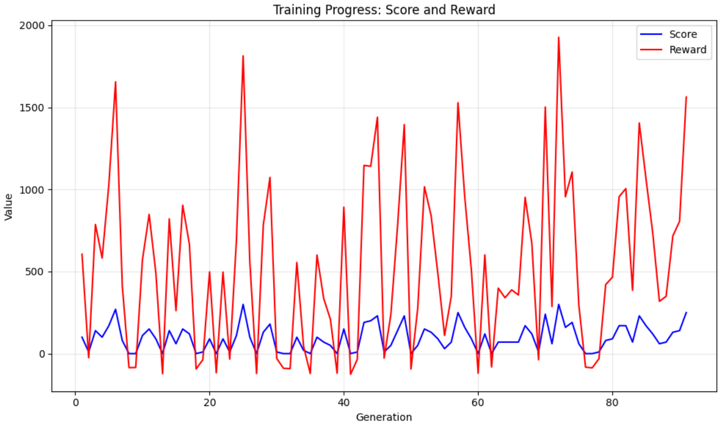
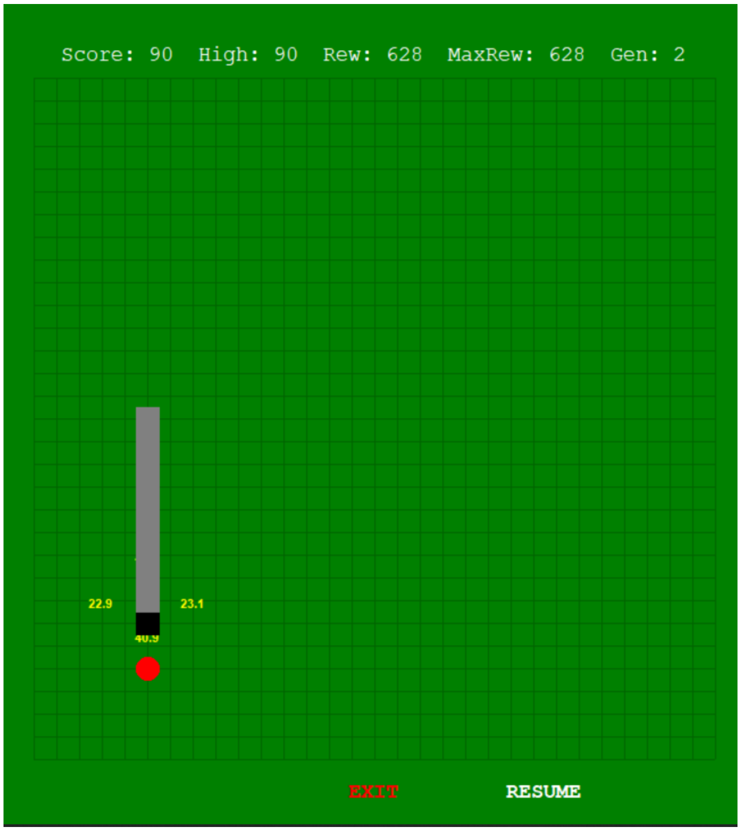


Figure 1: Training Progress: Score and Reward over Generations

- **Score Correlation**: The Score follows a similar trend to the Reward, as eating food provides the largest positive reinforcement.
- **Fluctuations**: The graph shows volatility. This is expected due to the random nature of food placement and the continued exploration (10% chance of random action) which prevents the agent from getting stuck in local optima but occasionally leads to suboptimal moves.



Score: 90 High: 90 Rew: 628 MaxRew: 628 Gen: 2

Figure 2: Screenshot of the Snake Game Environment