COMP 469
Group Project Presentation

Presented by VENOM

Group members of VENOM

Devaansh Mann

Abigail Rosas

Vanessa Juarez

Snake Game with reinforced learning

This game serves as the source for exploration and implementation of Q-Algorithm and the use of data tracking to track the progress our snake, Venom.

The Classic Snake Game

The Snake Game is a classic arcade game where the player maneuvers a line that grows in length, with the food being a primary target. The snake consumes the food and grows in length. The snake has to be in a specific grid in order to protect itself from dying by colliding with the grid walls. The snake also has to avoid hitting itself to continue playing the game.

Objective: The goal is to eat food that appears on the screen, causing the snake to grow longer. The game ends if the snake collides with the walls or itself.

Key Concepts used and implemented: Reinforcement learning, Q-Algorithm, Data Tracking

Code Snippet

```python
import pygame
import random
import numpy as np
import matplotlib.pyplot as plt

pygame.init()
pygame.font.init()  # Initialize the font module

class AISnake:

    # Colors
    yellow = pygame.Color(255, 255, 0)
    black = pygame.Color(0, 0, 0)
    white = pygame.Color(255, 255, 255)
    red = pygame.Color(255, 0, 0)

    # Window Displays
    winWidth = 720
    winHeight = 480
    winDisplay = pygame.display.set_mode((winWidth, winHeight))
    pygame.display.set_caption("COMP 469: Venom - Snake Game")

    frameSpeed = pygame.time.Clock()

    def __init__(self):
        self.reset_game()
        self.score_list = []
```

```python
def track_scores(self):
    plt.plot(range(len(self.score_list)), self.score_list)
    plt.xlabel('Games')
    plt.ylabel('Scores')
    plt.title('Snake Game Scores')
    plt.grid(True)
    plt.show()
```

```python
def scoreCount(self):
    scoreFont = pygame.font.SysFont("arial", 25)
    scoreDisplay = scoreFont.render("Score : " + str(self.score), True, self.white)
    self.winDisplay.blit(scoreDisplay, [0, 0])

def is_collision(self, point=None):
    if point is None:
        point = self.snakePos
    if point[0] > self.winWidth - 10 or point[0] < 0 or point[1] > self.winHeight - 10 or point[1] < 0:
        return True
    for block in self.snakeSize[1:]:
        if block[0] == point[0] and block[1] == point[1]:
            return True
    return False

def play_step(self, action):
    self.direction = self.change_direction(action)

    self.snakePos[0] += self.snakeSpeed if self.direction == "RIGHT" else -self.snakeSpeed if self.direction == "LEFT" else 0
    self.snakePos[1] += self.snakeSpeed if self.direction == "DOWN" else -self.snakeSpeed if self.direction == "UP" else 0

    self.snakeSize.insert(0, list(self.snakePos))
    if self.snakePos[0] == self.foodPos[0] and self.snakePos[1] == self.foodPos[1]:
        self.score += 1
        reward = 10
        self.foodPresent = False
    else:
        self.snakeSize.pop()
        reward = 0

    if not self.foodPresent:
        self.foodPos = self.random_food_position()
    self.foodPresent = True

    game_over = False
    if self.is_collision():
        reward = -10
        game_over = True
```

Reinforcement Learning

Definition: A type of machine learning where an agent learns to make decisions by performing actions in an environment to maximize cumulative reward.

Components:

- Agent: The snake.
- Environment: The game grid.
- Actions: Move up, down, left, right.
- Rewards: Positive reward for eating food, negative reward for hitting walls or itself.

Application to Snake Game

The snake learns optimal strategies to maximize the score by exploring different movement patterns.

Q-Learning Algorithm

Q-Learning: A model-free reinforcement learning algorithm to learn the value of an action in a particular state.

Key Components:

- Q-Table: Stores the Q-values (expected future rewards) for each state-action pair.
- Learning Rate (αα): Determines how much new information overrides the old.
- Discount Factor (γγ): Measures the importance of future rewards.

Implementation in Snake Game

- The snake uses the Q-table to decide the best action based on the current state.

Data Tracking

Data Tracking in the Game

Importance: Allows monitoring of the agent’s performance and aids in debugging and improving the algorithm.

Types of Data Collected:

- Scores per game.
- Learning rate and reward progression.
- The program graphs the data in a graph which shows number of games on the x-axis, and number of points scored on the y-axis.

Results and Analysis

Performance Metrics

- **Metrics:**
  - Score: Average score over episodes.
  - Steps: Average number of steps before game over.
  - Convergence: Measure how quickly the Q-values stabilize.

- **Graphs and Charts:**
  - Plot average score per episode.
  - Plot average steps per episode.

Analysis:

- Evaluate the learning progress.
- Identify patterns or anomalies in the movement.

Challenges and Solutions

Challenges Faced

- Exploration vs. Exploitation: Balancing exploration of new strategies and exploiting known strategies.
- State Space Complexity: Large state space making it difficult to converge.
- Reward Design: Designing a reward system that effectively guides the learning process.

Solutions

- Exploration Techniques: Use epsilon-greedy strategy with decay.
- State Representation: Simplify the state space or use function approximation.
- Reward Shaping: Carefully design rewards to incentivize desired behaviors.

Future Work

Potential Improvements

- Algorithm Enhancements: Explore more advanced algorithms like Deep Q-Learning.
- Feature Engineering: Improve state representation for better learning.
- Multiplayer Mode: Extend the game to support multiple snakes.

Additional Features

- Visualization Tools: Enhanced visualizations for better understanding of the learning process.
- User Interface Improvements: Improve the game’s UI for better player experience.

Now, we would like to present you all, the actual game which we worked on as a team over the summer for the class.

I will now ask my teammates, Abigail and Vanessa, to present the game to you all.

Thank you