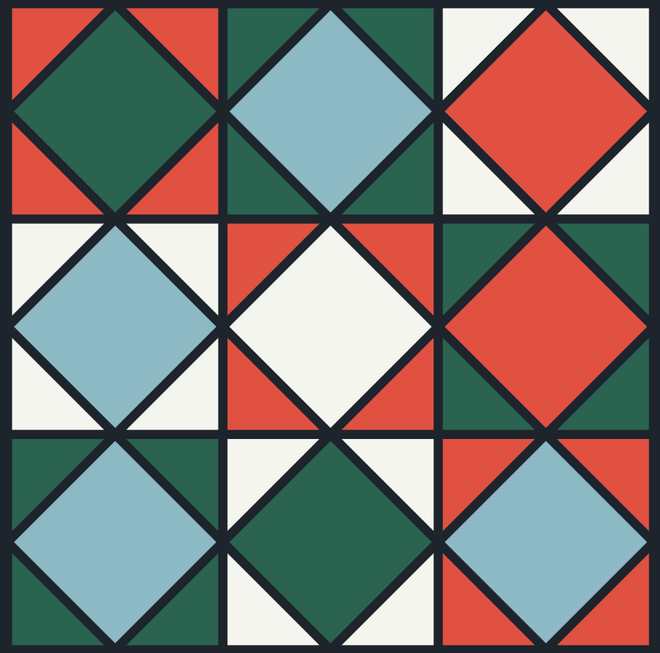
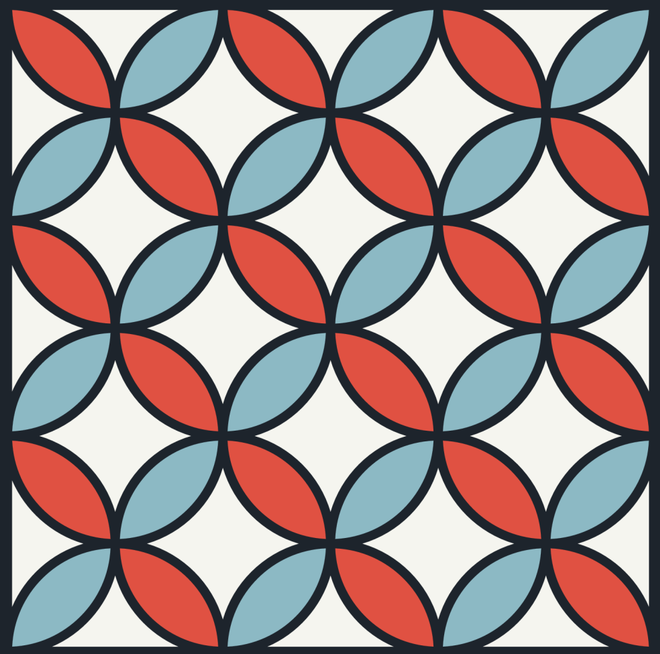
SNAKE GAME
&
RL





SNAKE GAME?

LET US REMIND YOU

Introduction

PROBLEM STATEMENT

The objective of this project is to fully emulate the classic Snake game and use various Reinforcement Learning (RL) algorithms to develop an optimal strategy for the agent (the snake) to maximize the score.

SCOPE OF THE PROJECT

Implement 3 to 5 RL approaches and evaluate their effectiveness. Focus on exploring different reward functions and comparing the performance of algorithms.

MOTIVATION

The Snake game, with its dynamic environment, offers a challenging problem for RL due to the constant movement, non-deterministic food placement, and the possibility of collisions, making it an ideal candidate for testing RL strategies.

OUR TEAM

IVAN

MAXIM

ROMAN

ENVIRONMENT, STATE-SPACE & ACTION-SPACE

GAME ENVIRONMENT

The Snake game involves a grid where the snake moves to eat food while avoiding collisions with the walls or itself. The environment will be simulated in Python using libraries like Pygame or a custom grid implementation.

STATE-SPACE

Each state represents the current position of the snake's body, the location of the food, and potentially additional features such as distance to walls, self-collisions, etc. You could represent the state as a tuple or a matrix.

ACTION-SPACE

The action space will consist of four possible actions: up, down, left, and right. The RL agent will learn to choose one of these actions based on the current state.

REWARD FUNCTION?

SEVERAL REWARD STRUCTURES CAN BE DEFINED FOR THE GAME

SIMPLE REWARD

+1 for eating food
-1 for colliding with the wall or itself

TIME-BASED REWARD

Small penalty for each move to encourage faster solutions (discourage long survival without eating).

DISTANCE-BASED REWARD

Positive reward for getting closer to the food
Negative reward for moving further from the food or heading toward a collision.

CANDIDATE ALGORITHMS

Q-LEARNING

SARSA

POLICY GRADIENT METHODS (REINFORCE)

ACTOR-CRITIC

DEEP Q-NETWORK (DQN)

TOOLS & ENVIRONMENT

- TOOLS:
  - PYGAME
  - PYTORCH
  - TENSORFLOW
- ENVIRONMENT:
  - EPISODIC
  - ENDS WHEN THE SNAKE HITS THE WALL OR ITSELF
  - AGENT TAKES ACTIONS, RECEIVES REWARDS
  - AGENT LEARNS THROUGH TRIAL AND ERROR TO MAXIMIZE ITS SCORE

CALENDAR

WEEK 6 - PROPOSAL PRESENTATION

WEEK 7 - MODEL-FREE PREDICTION-II

WEEK 8 - MODEL-FREE CONTROL

WEEK 9 - VALUE FUNCTION APPROXIMATION & POLICY GRADIENT METHODS

WEEK 10 - INTEGRATING LEARNING AND PLANNING

WEEK 11 - EXAM

WEEK 12 - STATE-OF-THE-ART RL ALGORITHMS

WEEK 13 - SUPPORT WEEK

WEEK 14 - FINAL PRESENTATIONS AND REPORT/CODE SUBMISSION

END