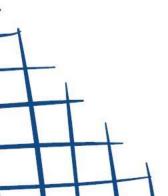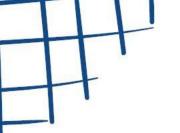# Python Programming

About Us

At Digital Empowerment, we envision a future where youth are equipped with the skills, knowledge, and mindset needed to thrive in a rapidly evolving world. Our mission is to bridge the digital divide, foster leadership development, and enhance academic growth, empowering young minds to realize their full potential.
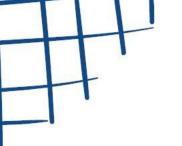
To achieve this, we offer comprehensive virtual internships across various domains, providing students with invaluable hands-on experience and practical skills essential for success. Furthermore, we are committed to assisting exceptional students in securing positions at prestigious companies, helping to launch their careers and build a brighter future for themselves and future generations.

Instructions

o Enhance your online presence by updating your LinkedIn profile.

o   For successful completion of the internship, complete one level (Level 1, 2, or 3) of the Web Development internship at your own pace.

o Keep your tasks organized in a separate GitHub repository named "DEP" and share the link in the task submission form. Utilize online resources like Google Search, tutorials, and videos to assist you in completing the tasks.

Introduction

Provide an overview of the Red-Blue Nim Game and explain the two game versions (Standard and Misère). Highlight the objectives and goals of implementing this game in Python.

Game Rules

2.1 Standard Version

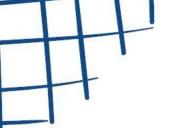Players lose if either pile is empty on their turn.

2.2 Misère Version

Players win if either pile is empty on their turn.

2.3 Scoring

Each red marble left: 2 points.
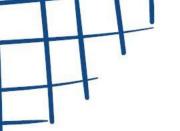
Each blue marble left: 3 points.

Command Line

Outline how to use the command-line to start the game. Provide an example of the invocation.

3.2 Parameters

Explain each parameter in detail:

•<num-red>: Number of red marbles.

•<num-blue>: Number of blue marbles.

•<version>: 'standard' (default) or 'misere'.

•<first-player>: 'computer' (default) or 'human'.

•<depth>: Search depth for AI (optional).
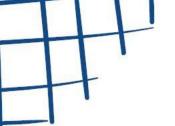
Game Flow

4.1 Turn Order
Describe how the game alternates between the human and computer player until the game ends.

4.2 Human Move
Explain how the program prompts the human player for their move and validates the input.

4.3 Computer Move
Describe how the program determines the computer's move using the MinMax algorithm with Alpha Beta Pruning.

5. MinMax Algorithm

5.1 Overview

Provide an overview of the MinMax algorithm and how it is used to optimize decision-making in the game.

5.2 Alpha Beta Pruning

Explain the concept of Alpha Beta Pruning and its role in improving the efficiency of the MinMax algorithm.
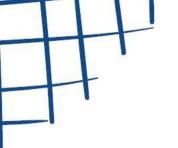
5.3 Move Ordering (Standard)

Describe the move ordering used in the standard version:

- o Pick 2 red marbles.
- o Pick 2 blue marbles.
- o Pick 1 red marble.
- o Pick 1 blue marble.

5.4 Move Ordering (Misère)

Describe the inverted move ordering used in the misère version:

- o Pick 1 blue marble.
- o Pick 1 red marble.
- o Pick 2 blue marbles.
- o Pick 2 red marbles.

Depth Limited Search (Extra Credit)
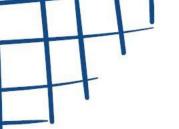
6.1 Purpose

Explain the purpose of implementing depth-limited search for faster decision-making.

6.2 Evaluation Function

Describe the heuristic evaluation function used to evaluate non-terminal game states.

6.3 Extra Credit Details

Provide details on the extra credit assignment and the requirements for describing the evaluation function reasoning.

7. End of Game

7.1 Game Over Conditions

Describe the conditions under which the game ends.

7.2 Scoring Calculation

Explain how the final score is calculated based on the remaining marbles.
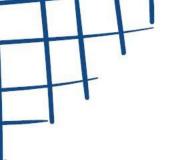
Implementation Details

8.1 Modules

Describe the different modules implemented in the game, such as:
Command-line parsing
Game mechanics
Human and computer moves
AI decision-making with MinMax and Alpha Beta Pruning

Demonstration

9.1 Walkthrough

Provide a walkthrough of a sample game, showcasing both human and computer interactions