**Project Title : “Nim-game”**

**Creator Details :**

Muhammad Zeeshan Sadiq

[zeeshansadiq356@gmail.com](mailto:zeeshansadiq356@gmail.com)

**Introduction:**

The Red-Blue Nim Game is a mathematical game of strategy that involves two players taking turns removing objects from distinct heaps or piles. The game consists of two piles, one red and one blue, and players alternate turns removing objects from one of the piles.

There are two main versions of the Red-Blue Nim Game: Standard and Misère.

**Standard Version:** In the standard version of the game, the objective is to be the last player to remove an object from the piles. The game ends when all objects have been removed, and the player who made the last move wins.

**Misère Version:** In the Misère version of the game, the objective is to force your opponent to take the last object. This means that the game ends when only one object is left, and the player who did not take the last object wins.

Implementing the Red-Blue Nim Game in Python can help achieve several objectives and goals:

1. **Game Development:** By implementing the game in Python, you can develop a fully functional game that can be played against a human opponent or an AI agent.
2. **Strategic Analysis:** Implementing the game in Python allows you to analyze and compare different strategies for playing the game, which can lead to a deeper understanding of the game's mechanics and optimal play.
3. **AI Development:** By implementing the game in Python, you can develop AI agents that can play the game against human opponents, which can help improve the AI's decision-making and strategic thinking capabilities.
4. Also you can get a deep knowledge of Python programming and AI by using implementing Minmax algorithm, alpha-beta pruning method and many more things like that.

**Scoring Conditions of the game :**

In both these versions (standard and misere) of the game, the scoring conditions at the end of the game are same , scores are generated according to the no of marbles left at each player’s side. For 1 red-marble left, there will be 2 points and for 1 blue-marble left there will be 3 points. Final score will be the sum of the number of these two marbles. If it’s a tie between human and computer player then there will be zero score for the players.

**Implementation of Minmax Algorithm & Alpha-beta Prunning:**

To implement the MinMax algorithm with Alpha Beta Pruning in the given code, we need to make the following changes:

* Create a function to evaluate the game state. This function will return a score based on the current state of the game.
* Create a function to perform the MinMax algorithm with Alpha Beta Pruning. This function will recursively explore the game tree and return the best move.
* Modify the play\_game method to use the MinMax algorithm with Alpha Beta Pruning to determine the computer's move.

**Game over conditions:**

The game ends under the following conditions:

* Standard Version: The game ends when one of the piles is empty. The winner is the player who makes the last move, i.e., the player who removes the last stone from either pile. If both piles are empty at the same time, the game is a tie.
* Misere Version: The game ends when one of the piles is empty. The winner is the player who does not make the last move, i.e., the player who does not remove the last stone from either pile.
* In both versions, the game ends when one of the piles is empty, but the winner is determined differently.

**Github Repository link for the project :** <https://github.com/DavidSunderland7/DEP.git>