



Department of Computer Science and Engineering,
Khulna University of Engineering & Technology, Khulna.

CSE 4110 : Artificial Intelligence Laboratory

Report on Partitioning Game: An AI based Game.

Presented by

MD Tahmid Hasan Fuad
Roll: 1707114

MD Likhon Sarker
Roll: 1607069

MD Samin Yeasar
Roll: 1507099

Submission Date: July 3, 2022

Objectives

The objectives of this project is to

- Build a two player game where the first player is human and the second one is a computer.
- Implement a heuristic search algorithm to find the optimal path for the computer.
- Implement the optimized version of heuristic search algorithm.
- Implement the game into a decent user interface (UI).
- Use Kivy framework for the UI.

Introduction

Partitioning Game is based on a game theory problem. In this problem, the user first has to input the number of blocks. Then, the game will go on by dividing the number blocks into two unequal parts. If one player cannot make an unequal partition, that player will be lost. In this game, we use python as both backend and frontend language. We design the user interface using python's Kivy framework. We implement the backend code using dynamic programming and the optimized minimax algorithm.

Game Description

The sequence of the game:

- At first the player has to input the no of blocks.
- Then, he will divide the blocks into two unequal parts and give the pair of numbers as input into the text box.
- After submitting that, the computer will begin calculating the optimal path from those previous players' partitions using the search algorithm.
- Then, the computer will generate an output.
- This play will go on until one player loses the game.

In the figure 1, 2 & 3 the sequence of the game is shown.

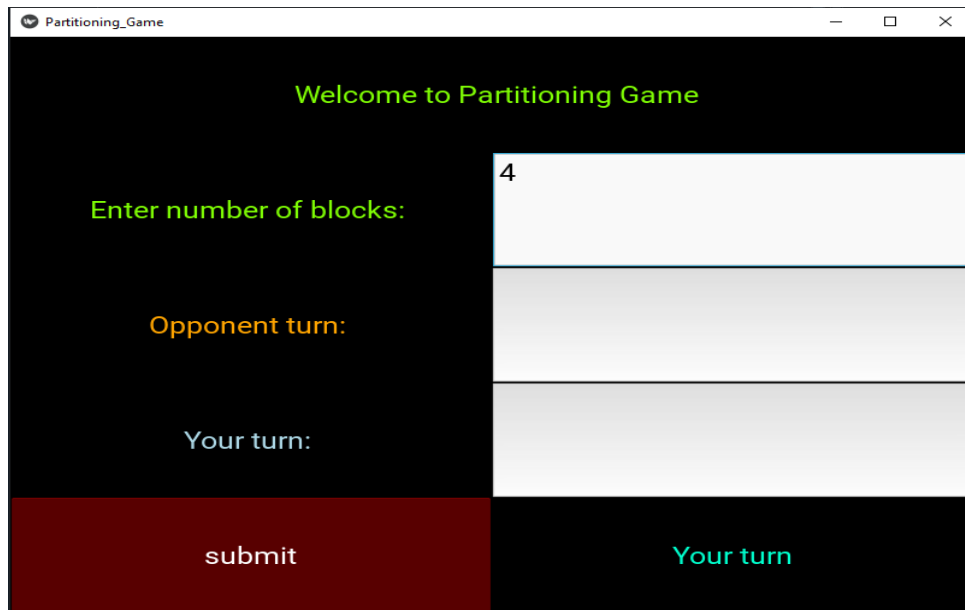


Figure 1: Player 1 Submits Number of Blocks.

Algorithm

- **Step 1:** For each number of blocks, partition it into all possible parts.
- **Step 2:** Run a loop from 3 to 1000 and divide each number into all possible two parts.
- **Step 3:** $DP[block]$ will store 1 if there exists a way to partition the block that can lead to winning position.
- **Step 4:** $DP[block]$ will store 0 if there does not exist a way to partition the block

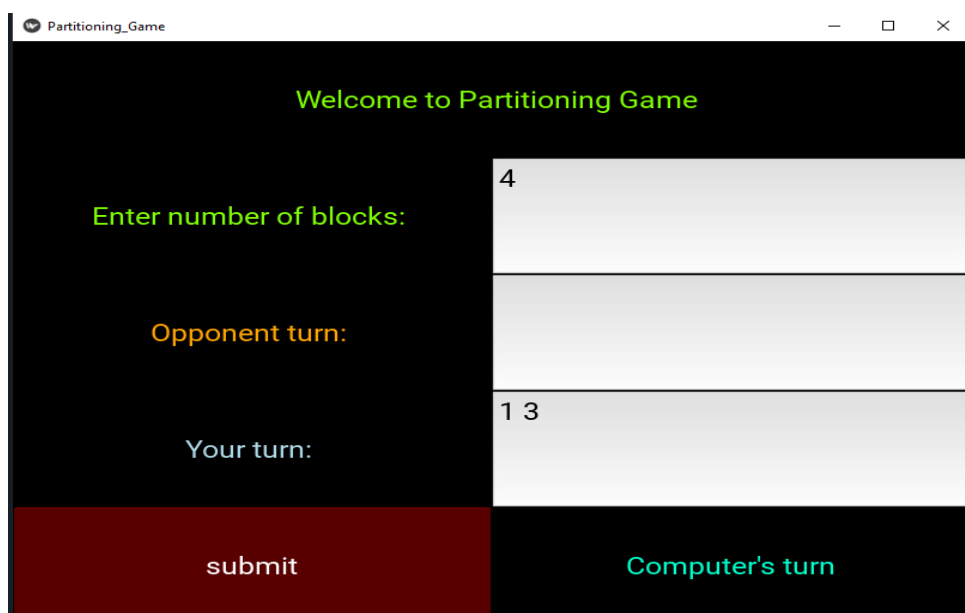


Figure 2: Player 1 Submits Partitions.



Figure 3: Computer Gives Its' Moves and Player 1 Loses.

that can lead to winning position.

- **Step 5:** If there is a way to partition a block into two parts (X,Y) and both $DP[X]$ and $DP[Y]$ store 0, then this is an optimal partition, assign $DP[block]=1$ and store these two parts(X,Y).
- **Step 6:** If there doesn't exist a partition that can lead to winning position, assign $DP[block]=0$.
- **Step 7:** Now the game begins.
- **Step 8:** User will partition the block into two parts(A,B).

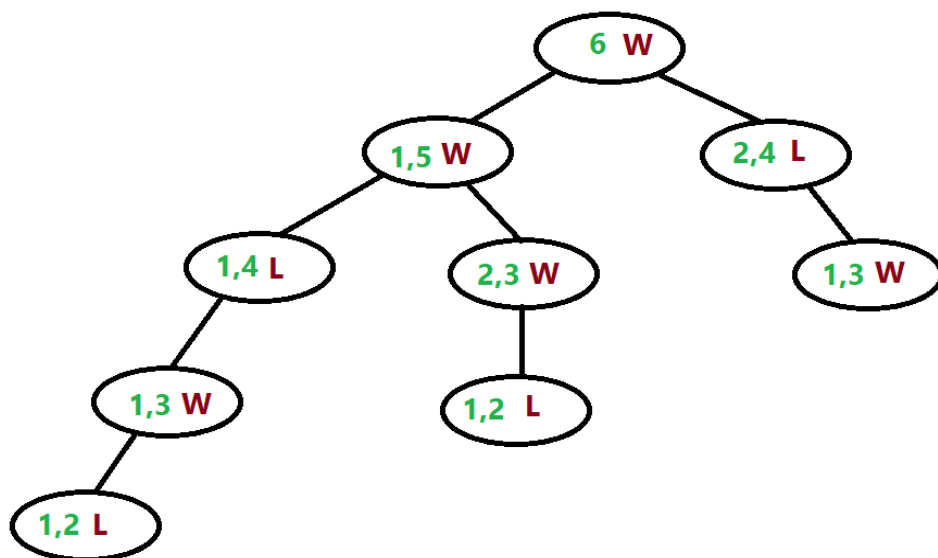


Figure 4: Tree When Input Block Number Is 6.

- **Step 9:** Computer will check both the DP[A] and DP[B]. If DP[A] stores 1, computer will choose part A and discard part B. Otherwise, computer will choose part B and discard part A.
- **Step 10:** The player who will get the block size 1 or 2, will lose the game as there is no valid move from this position.

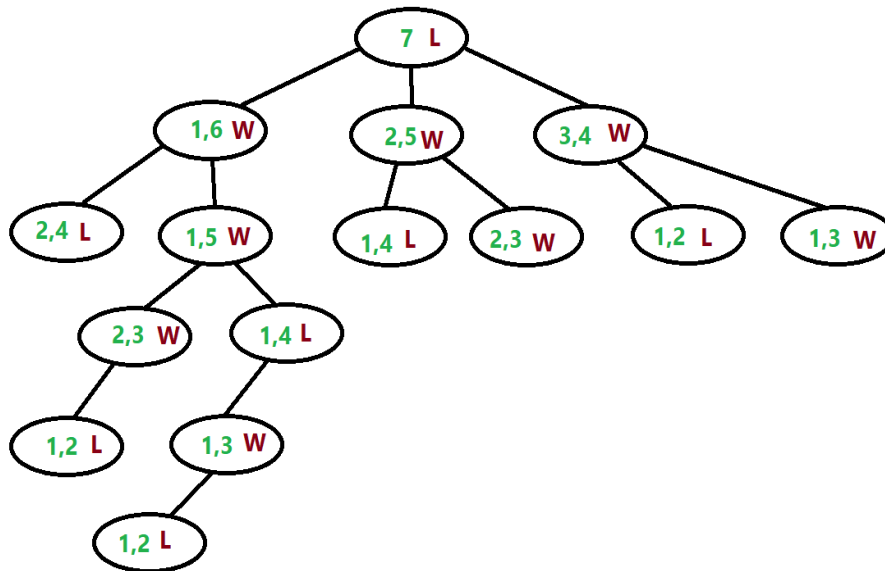


Figure 5: Tree When Input Block Number Is 7.

Discussion

Through this project, a two-player game named 'Partitioning Game' has been implemented with a python program as the backend and Kivy framework as the frontend. Here, one player would be human, and the other would be a computer or an AI. So, an AI program has been designed with the help of a dynamic programming-based search algorithm in this game. The human player can choose the number of blocks at first. It is not easy to win over the AI as AI always gives the optimal move. The game and the outcome have been checked several times through simulation.

Conclusion

We can state that the Partitioning Game is based on game theory problems. Here, a human competes with an AI to win the game. The performance of the AI ensures the average difficulty level. It is a tricky brainstorming game because a player must need thinking about giving optimal partition numbers.

Reference

- Artificial Intelligence: A Modern Approach
by Stuart Russel, Peter Norvig
- <https://kivy.org/>