OBJECT ORIENTED PROGRAMMING USING C++

Asaduddin Bilal Mohammed

Introduction:

Brainvita, also called Peg Solitaire, is a single person board game. It involves moving marbles from one position to another position on the board according to rules. The objective is to have as few marbles as possible at the end of the game.

The rules of the games are as follows :

1. There are a set of marbles placed on the board with only the centre portion left vacant. All you have to do is move one marble over the other either horizontally or vertically (diagonal moves are not allowed). As soon as you do this, the marble over which you took the other marble disappears.
2. Continue moving one marble over the till you have no other option. The aim of the game to leave only ONE marble on the board.

Approach:

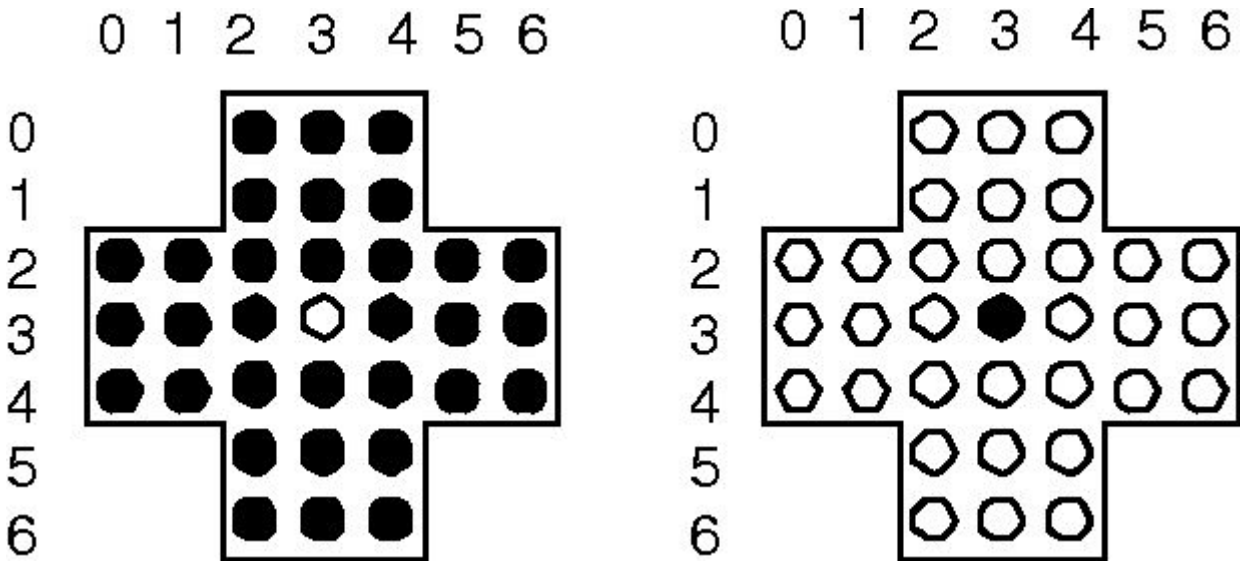
I started the code by creating a board using 2 dimensional arrays and pointers.

After creating the board, I created a function for accepting the username and storing the result in a text file.
Ascertaining the validity of the move was an uphill battle but I was successful in determining the valid moves by the usage of multiple loops and conditional statements.
Upon each successful move, the logic behind the board creation is used recursively and the board is printed with the appropriate changes.
After each move, a function is executed to determine whether the user has reached the goal.
Usage of a class and defining the methods in a class eased my job of creating the game.

Model:

The following is a brief explanation of my model.

Board:
The board is a two dimensional array of integers, which consists of values 0 and 1, where 1 represents a marble and 0 represents empty space.



Moves:

The moves of marbles are based on computations which check the validity by permitting only horizontal or vertical moves. Diagonal moves are invalid, empty spaces cannot be accessed. A marble can only be moved to a new position if it jumps over another marble. A marble cannot jump over an empty space. All these conditions had to be calculated for validating a move.

Game Instructions:

Since the board is represented in the form of a 7x7 matrix, to play the game, the user has to input the row and column number of a marble.

Limitations:

My game implements only one type of board for this game, there are multiple types of boards that are used to play the game. The option to perform a reversal is not available to the user.

Compilation Instructions:
I have compiled the code in DevC++ compiler.