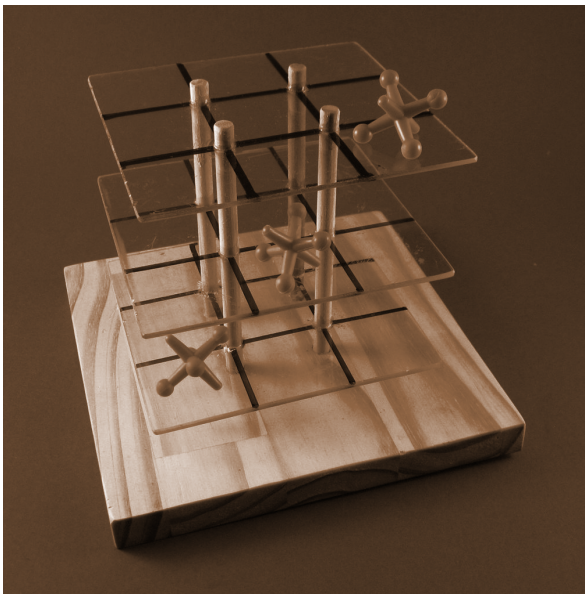
Documentation of
3-D Magic Cube and 3-D Tic-Tac-Toe



1. Algorithm for 3-D Magic Cube -

A magic cube of order n is a cubical array,

$$M_n=[m_n(i,j,k);\ 1\leq i, j, k \geq n]$$

containing n natural numbers from 1 to n^3 such that the sums of the numbers along each row and also along each of its four great diagonals are the same, i.e. $\mathbf{n(n^3+1)/2}$.

- Formula used to construct a 3-D magic cube:

If $\mathbf{n \equiv 1 (mod) 2}$ then,

$$\mathbf{m_n\ (i,j,k) = a_{i,j,k}\, n^2 + b_{i,j,k}\, n + c_{i,j,k} + 1}$$

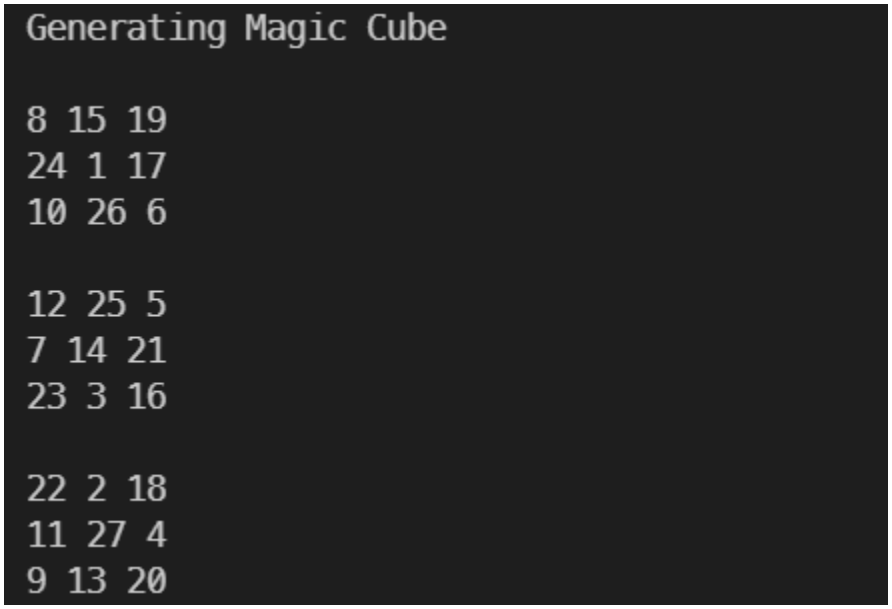
$$\text{where } a_{i,j,k} = (i - j + k - 1) \pmod{n},$$
$$b_{i,j,k} = (i - j - k) \pmod{n},$$
$$c_{i,j,k} = (i + j + k - 2) \pmod{n}.$$

In our case as **n=3** so,

$$3 \equiv 1 \pmod{2},$$
$$(3 - 1) \% 2 = 0$$

Hence we used this algorithm to construct our 3-D Magic Cube.

Snapshot of our Magic Cube-



2. Methods and Data Types used in 3D Tic-Tac-Toe -

- ***struct Point*** - a structure which holds the x,y,z coordinates of a point in cube

- ***Point valueToCoordinate(int)*** - Returns the coordinates of the corresponding number on the magic cube.

- ***int coordinateToValue(int, int, int)*** - Accepts the coordinates and returns its corresponding magic number using the algorithm formula.

- ***bool moveByUser()*** - Accepts the coordinates from the user and checks whether that space is already filled or empty. If empty, then insert the user's move there and update the user's move list.

- ***bool spaceLeft()*** - checks whether the particular point inserted by the user is empty or not.

- ***void showMoves()*** - it will update the user's and computer's moves list every time and display it.

- ***double distance(Point, Point)*** - it will return distance between two given points, used for checking collinearity.

- ***bool isCollinear(Point, Point, Point)*** - it will return true if 3 given points are collinear else false.

- ***Point make_2()*** - this method will return the most suited point for the computer only when it is impossible for him to win and impossible for him to block the user's winning move.

- ***Point possibleWin()*** - this function is used by the computer to make its next move. The conditions which the computer checks are -

    1. Make a winning move if possible.
    2. If necessary, prevent the opponent from making a winning move.
    3. Otherwise, choose an empty square at random.

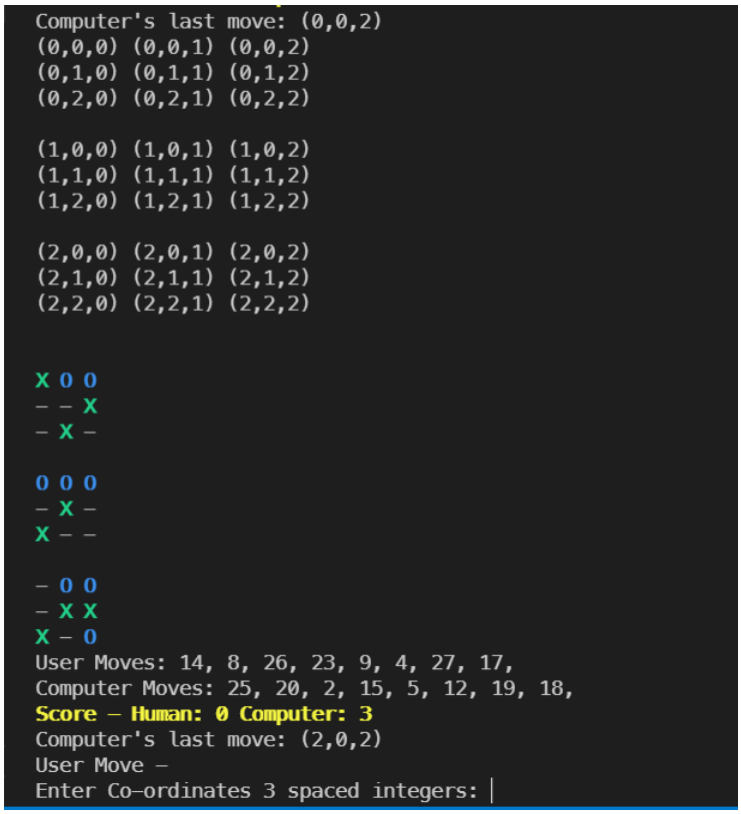
- ***bool moveByAI()*** - this method will get the most convenient point from the Possiblewin() method and update its value in gameBoard and usedSpaces.

- ***void scoreBoard()*** - this method will keep record of the winning of user and computer by checking collinearity and magic sum condition and accordingly update the score of both user and computer.

- ***void printCube()*** - this function displays the magic Cube.
- ***void drawGameBoard()*** - this function renders the game board.

- ***void showMoves()*** - this function shows all the moves made by the user and the computer.

Snapshot of our 3-D TicTacToe-



3. References -

3.1 http://math.ku.sk/~trenkler/05-MagicCube.pdf

4. Contributors -

Aman Yadav	Akansh Mittal	Aryaman Singh Rana