• **Game Overview**

|  |  |  |
| --- | --- | --- |
| 1 | -1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | -1 |

Modified Add the Numbers is a fun logic based game in which you control a box with a numerical value on it. You can move **left**, **right**, **up**, **down** but whatever value is on the box that you replace effects your value. Some will be positive, and some will be negative. A positive will add to your value, but a negative will subtract.

The aim of the game is to get the highest score, givenmaximum number of movements and minimal score could be achieved. Good Luck and get adding!

***Board***

• Given 3x3 board initially initialize randomly with range [-1, +1].

• Position of player initially down left the board (Green tile in the pic) ***Rules of Play***

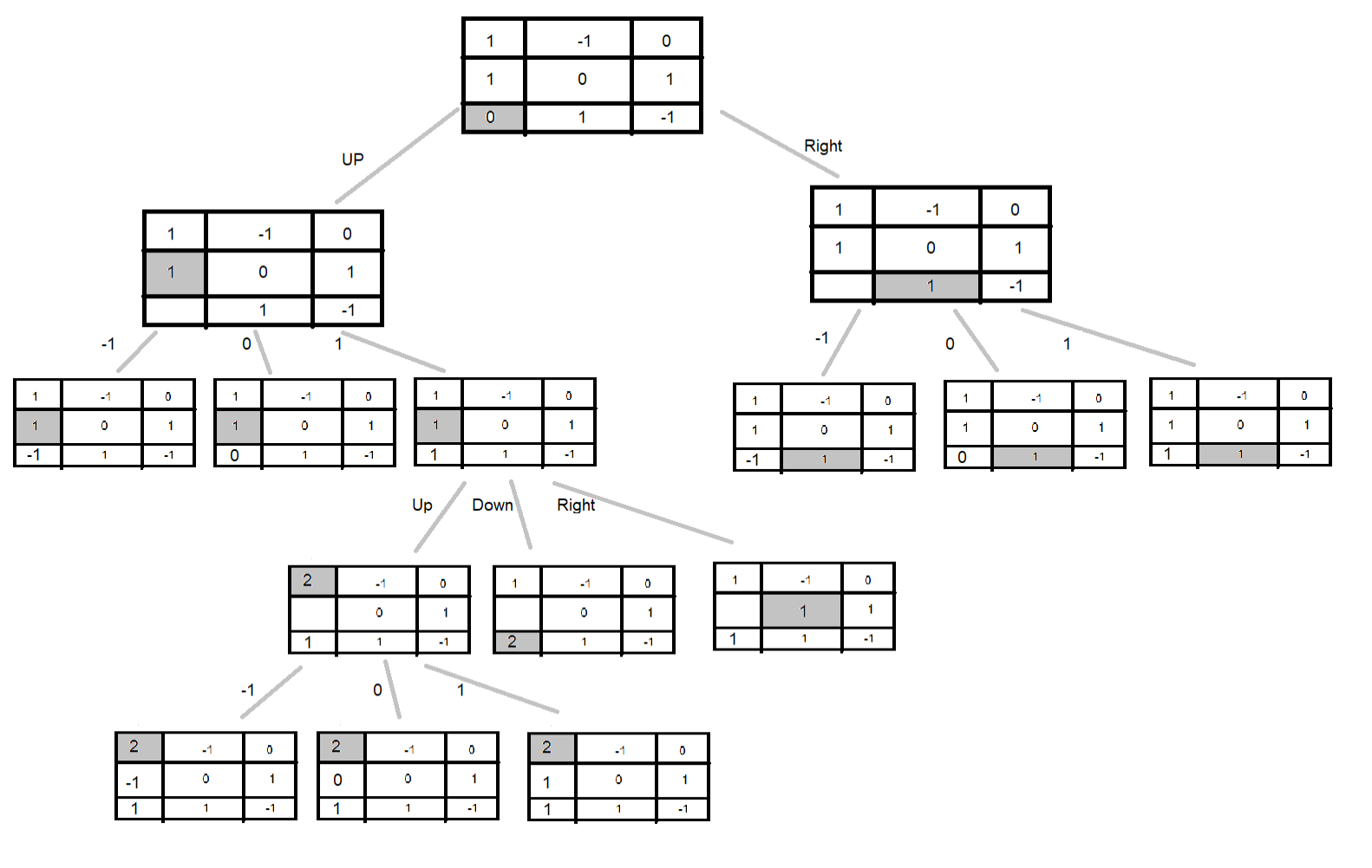
• The player can only move within the board’s boarder by the move up, down, left or right and he will represented as maximum (Max) player in alpha-beta algorithm.

• The computer will generate a random integer value range between [-1, 1] that will be represented as the minimum (Min) player in alpha-beta algorithm.

• If the player at any corner of board, he can’t move to another corner. Example: If the left most corner he can’t move to the right one if he pressed left button.

• Given the previous figure of the game, if the green tile moves to the right it’s value will be the addition of the old green tile value [0] and the right tilevalue [1] which willmeans that the value of the new green tile will be [1] and the value of the old green tile will be replaced by a random value from [-1,1] let’s say computer chose it to be -1 then the next state would look be as represented in the next figure:

|  |  |  |
| --- | --- | --- |
| 1 | -1 | 0 |
| 1 | 0 | 1 |
| -1 | 1 | -1 |



• **Project Components: a. Game Engine**

• The main process of this engine is initially take the user input and starts to update the game state using the implementation of the alpha-beta algorithm, to select the computer move, and show the board state and so on until the game ends and show the winning state if reached.

**b. Project Knowledge** • Game state.

• The board state; positions of the tiles as well as the possible moves in the board. • **Project Input:**

1- Minimal Level Goal, which choose the computer exceed this value. (for example 10)

2- Maximum number of moves, maximum movement to the computer to achieve the goal. (for example 20)

• **Project Output:**

1. For each step the current game state should be displayed, using textual representation or in GUI.

2. After the game is over, the final results are given including the input values, the current score, and if he wins or fail.