

State Diagram

SOEN 6011 Tic-Tac-Toe Project

XOGEEKS

Assignment 5

Team - 4

Amir Hakim	24050711
Beerpreet Singh Guliani	27644930
Keerthana Gudavalli	27588569
Neha Sharma	27733240
Ramanjeet Gill	27732902
Ramanjit Dhillon	40010613
Shidokht Hejazi Sepehr	40002808
Sushil Patil	27148917

Submitted To: Prof Nicolangelo Piccirilli

Contents

1. Tic Tac Toe State Diagram.....	2
2. Heuristic State Diagram.....	3
2.1 Diagram.....	3
2.2 Description.....	4
2.3 Example.....	4

1. Tic Tac Toe State Diagram

The following figure displays the state diagram of the tic tac toe game. The heuristic will be run when it's the computer's turn.

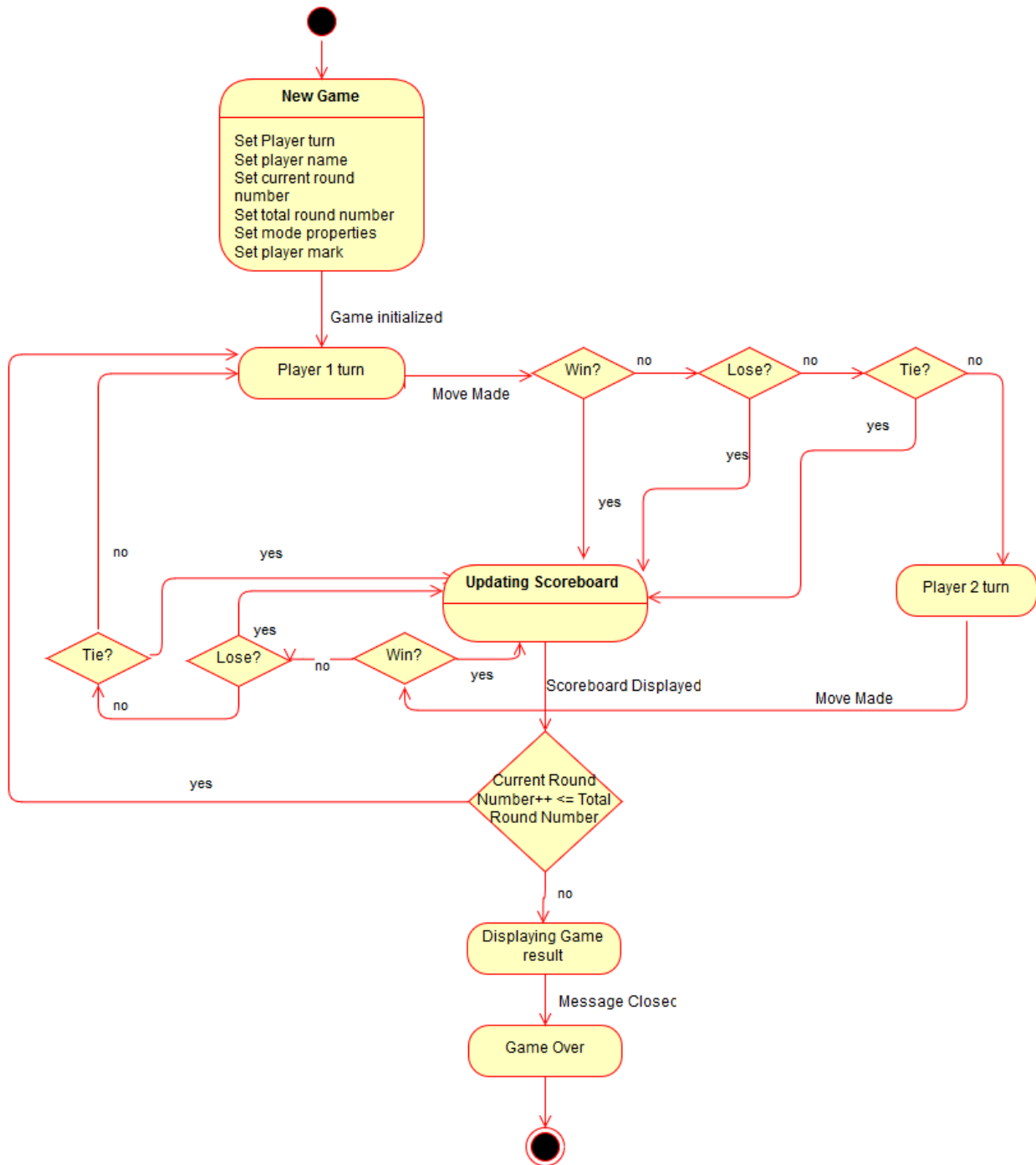


Figure 1: Game state diagram

2. Heuristic State Diagram

2.1 Diagram

The following figure displays the state diagram of the advanced level heuristic.

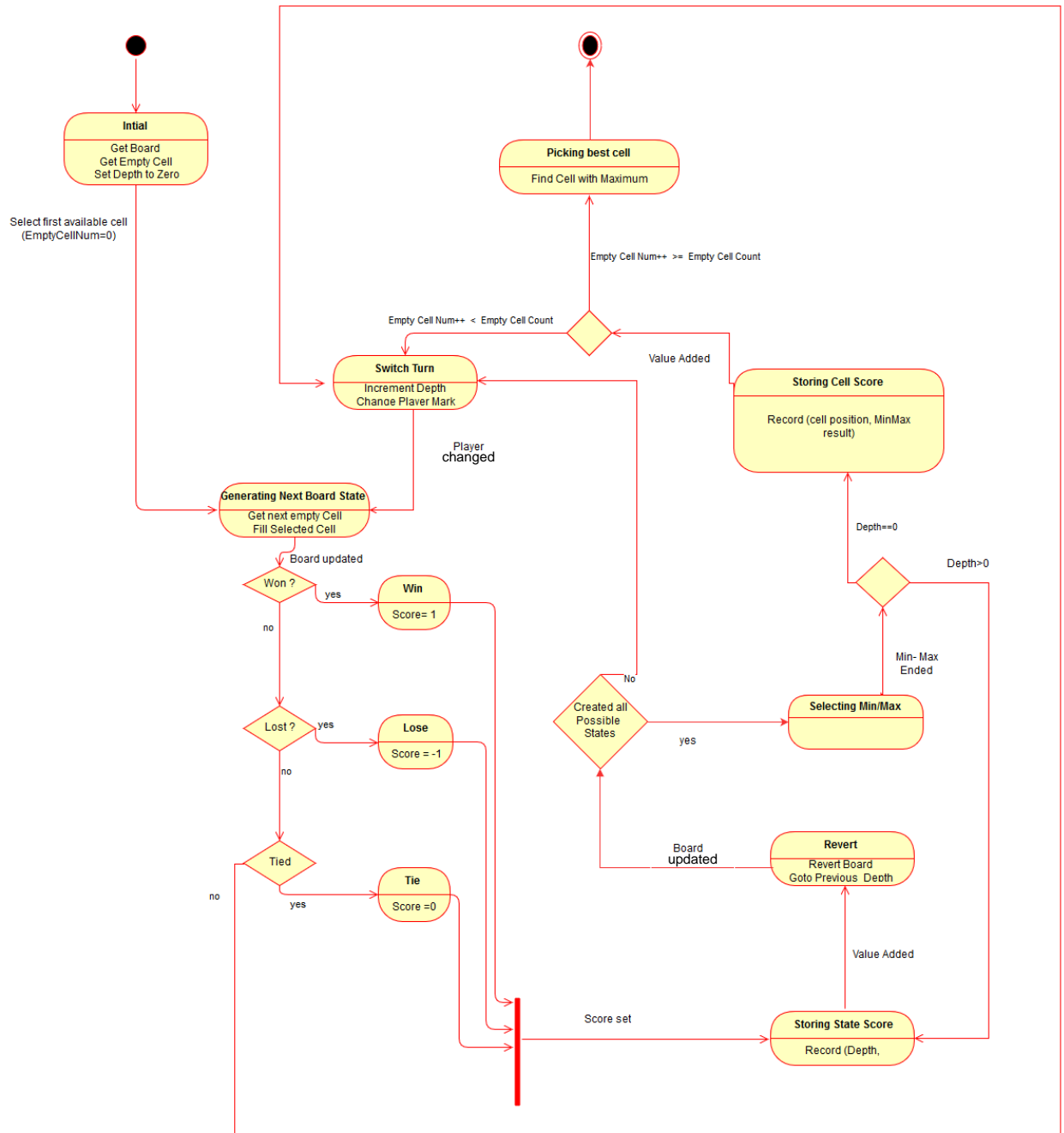


Figure 2: Heuristic state diagram

2.2 Description

The iterative heuristic starts with getting the board and finding the empty cells. Each empty cell is an option for placing the computer mark. Each one will be evaluated by creating all possible game states and performing the minimax algorithm. A tree with possible game states is simulated for this evaluation.

First, computer generates the next board state by placing the current player's mark (computer in the beginning) on the first available cell in "Generating next board state". Then it will check if this move results in a win, tie, or lose scenario. If the game is not in a final state, current player's mark will be switched and the next tree depth will be created in "Switch turn" state. If a final game state has occurred, the leaf of the tree has been reached. The corresponding score value will be temporary stored in "Storing state score" state, since the minimax algorithm needs to be performed later. After storing values, the computer reverts the board and tree depth to the previous state. All the possible states for each depth have to be created before performing the minimax for evaluation. "Generating next board state" keeps track of which cells have been filled in each depth to create all possible states. The "Created all possible states" condition checks to see whether all the children for a certain node have created or not. If all the children states have been created at that depth, "Selecting min/max" state is activated in which minimax is performed on the children. At this state if the current player is the opponent i.e. depth is an odd number, the minimum value of the children score is chosen and assigned to the depth. Otherwise the maximum value is chosen. This process is repeated for each depth until we reach the root (depth zero) which is the state of the game where the computer needs to make a move. At the root, the minimax value calculated is going to be stored and associated to the empty cell, showing its value. Once all the empty cells have been evaluated, the computer chooses the cell with the maximum value to make its move.

2.3 Example

The figure below shows an example of how the heuristic works with computer being the X player.

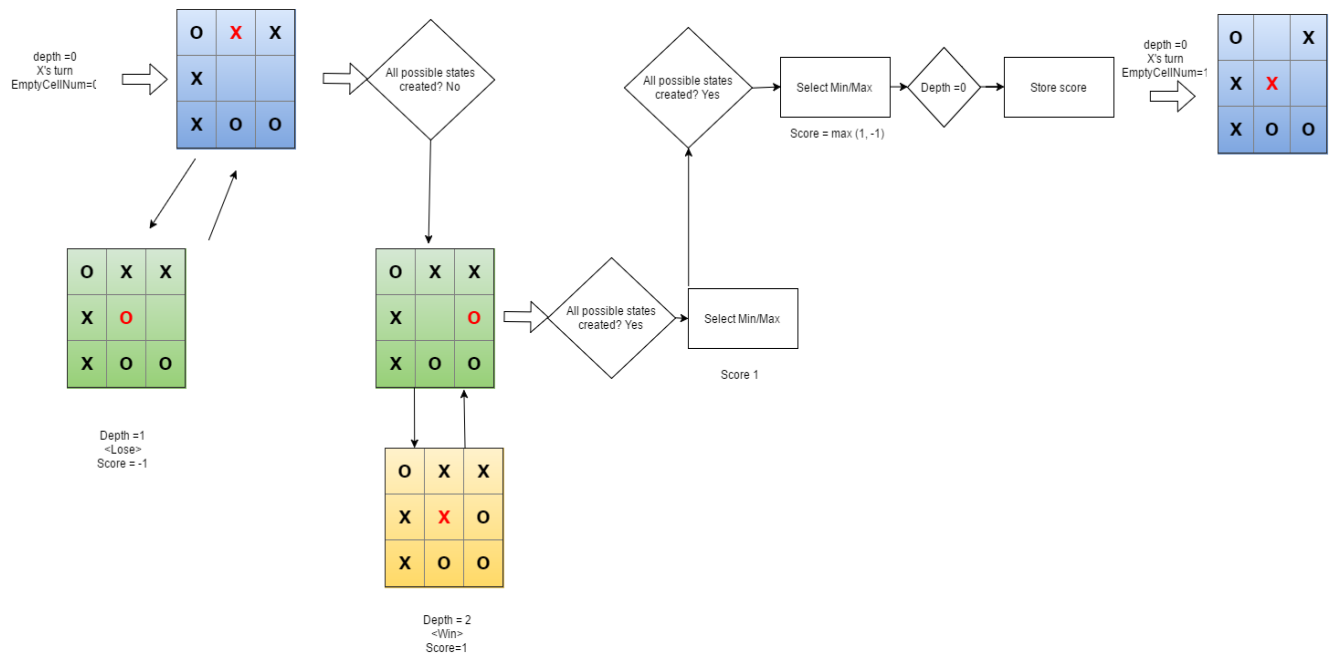


Figure 3: example

