

**ABSTRACTION & PARADIGMS**  
**FOR PROGRAMMING**  
**(CS 154)**

**Project Report**  
**On**  
**Connect-4**

14 March, 2011

By-

**Aman Mangal(100050015)**

**B.Tech.(1st Year)**

**Computer Science & Engineering Dept.**

**IIT Bombay**

Supervisor-

**Prof. Amitabha Sanyal**

**Head of the Dept.**

**Computer Science**

**IIT Bombay**

## **Problem Statement**

Connect-4 is a two player strategy game. The player has to get 4 checkers in a row horizontally, vertically or diagonally to win the game. The game continues until either of the players gets 4 in a row or the board is completely filled. If the board is filled & neither player has a 4 in a row, then the game is draw. So we have to-

- Generate an empty 7X6 board
- Prepare a user interface to play the game
- Check the presence of row of 4 checkers of same color as soon as user fills a checker in the board.
- Implement an AI player so that user can compete with an imaginary player.

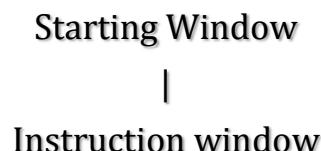
## **The Solution**

I've used 2D vectors (arrays) to represent my 7X6 board initialized with 0(means nothing). This board is defined globally so that it can be accessed by any of the intermediate function in the code.

The value for any cell could be 1 or -1as soon as it is filled by a yellow or red piece. The whole of the code can be roughly divided into three parts-

### **1. Graphics User Interface(GUI)**

I've used graphics/graphics library to make the GUI of the game. The navigation through the GUI is shown in the following table-





The variables used in the code are explained at suitable positions.

## **2. Checking the Presence of Row of 4**

The function Check-game-win checks the presence of row of four in the given board. It checks the presence of such a combination by checking the presence in every row, column and positively & negatively sloped diagonal.

As soon as it finds such a combination it changes the globally defined variable game-status and *comes out from the loop*.

There is another function called count-game-win. This checks the presence of those combinations of three, which can be converted into four later in the game.

### **3. Artificial Intelligence(AI)**

The general *min-max algorithm* is used to implement the Artificial Intelligence. I'm using the depth equal to 3, as it is neither time consuming nor too less. I've used *alpha-beta pruning* in a different fashion. There are three possible reasons for the move in order-

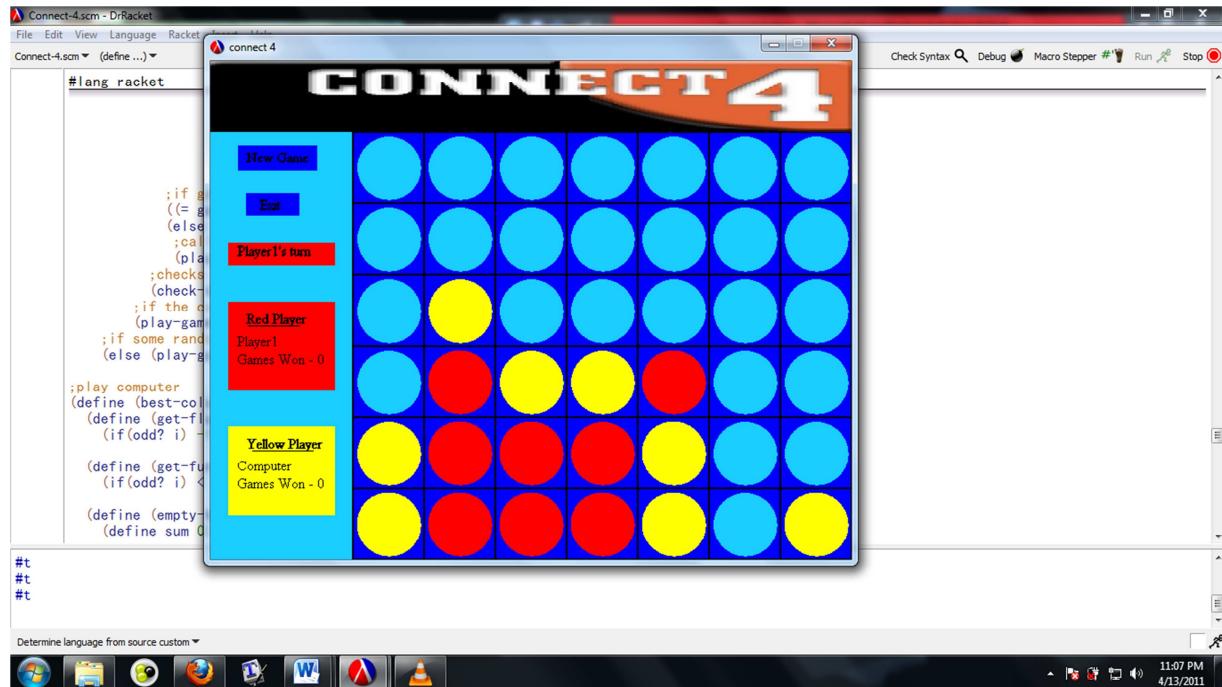
- It is a winning move. In this case the tree is cut & It doesn't go further. (the way I've implemented alpha beta) -> 1000 points
- If opponent wins just after the computer's turn. -> -1000 points
- Counting the number of threes present in a given board which can be converted into 4 later in the game. -> \*100 (-1 or 1) points

Finally mini-max decides the move.

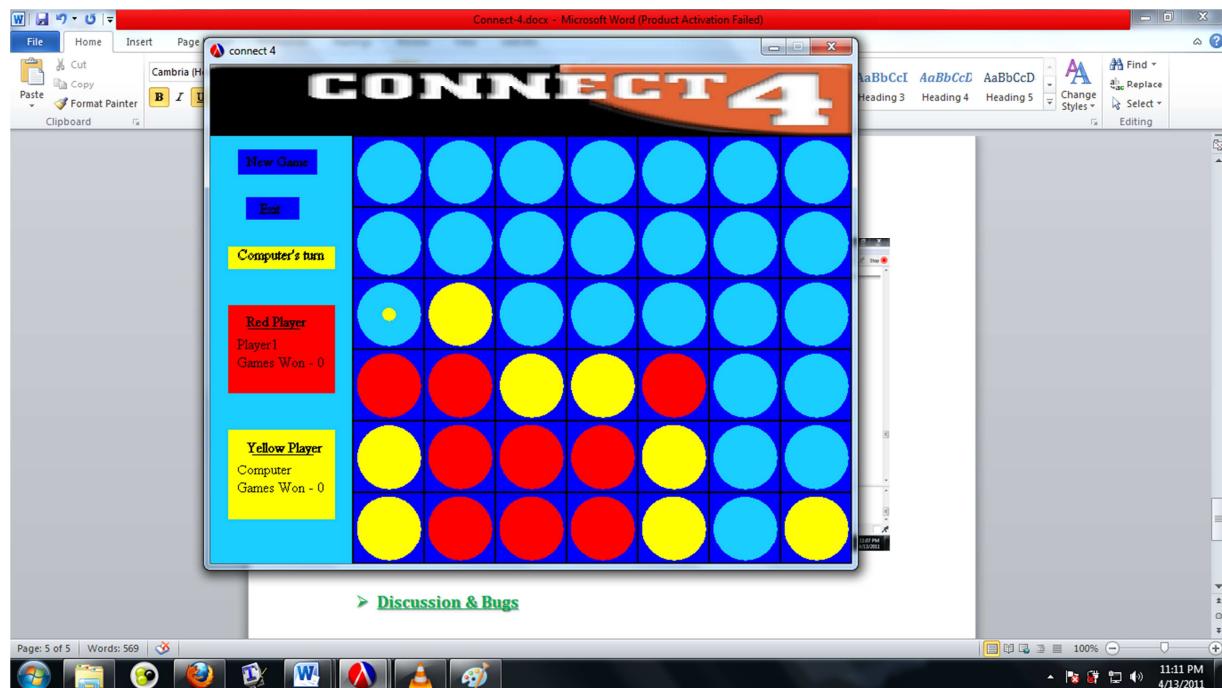
#### **➤ Sample Input-Output**

Inputs are taken by mouse. As soon as player clicks anywhere in the column, corresponding color piece is dropped in that column. The yellow color is assigned to computer if single player is chose. Some of inputs & outputs are shown in following figures-

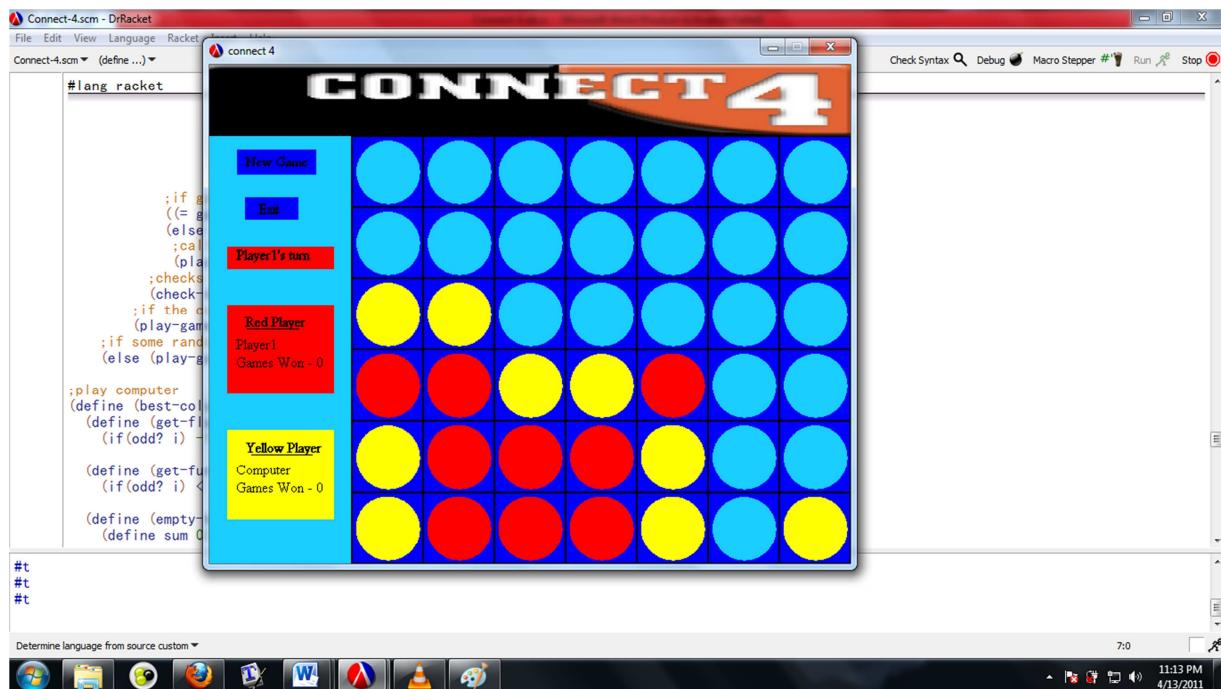
## Input-



If player clicks in the first column near the yellow point, a red piece is dropped in first column.



If game is single player, then a yellow color piece is dropped in the same column (first column), so that the player cannot win the game in his next move.



## ➤ Discussion & Bugs

Nothing is perfect in this world. Like every code, this code must have errors. I did a lot of experiments with the AI of game & found out following bugs-

- Rarely it gives error that the index is out of range in the function `2d-vector-ref`. The reason is very difficult to find out because it is very rare.
- Initialization of the game is very random which leads to an easy win of the opponent.

- The strategy which I've used mostly focuses on the thing that AI player tries to tie the game. It doesn't try to win the game. But even then it sometimes wins the game.
- If a click is made on the main window at the time some processing is going on or another window is opened, It records the move & use it later.

There could be some more in the code, but these are all which I could find out.

*There are so many things which could be added in the code to make it batter & better. In this project I've completed the entire basic thing which I've expected at the start of the project & written in the proposal. I've always focused on the thing which I thought initially not on the perfect algorithm.*

## **BIBLIOGRAPHY**

- Wikipedia- The free encyclopedia
- DrRacket help desk