VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY

UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Lab REPORT

Computer Architecture CO2008

Assignment

Four in a Row

Instructor: Băng Ngọc Bảo Tâm

Student: Nguyễn Hoàng Thuận – 2052729 – CC02

HO CHI MINH CITY, 2023

Introduction

Four in a Row is the classic two player game where you take turns to place a counter in an upright grid and win the game by placing 4 counters in a row.

Each player alternately takes a turn placing a piece in any column that is not already full. The piece fall straight down, occupying the lowest available spot within the column or be stopped by another piece. The aim is to be the first of the two players to connect four pieces of the same colour vertically, horizontally or diagonally. If each cell of the grid is filled and no player has already connected four pieces, the game ends in a draw, so no player wins.

1. In the first move of each player, they must drop the piece in the center column of the board.

2. In the middle of the game (after their first move), each player has 3 times to undo their move (before the opponent's turn).

3. Each player also has one times to block the next opponent's move. However, if the opponent has a chance to win (already had three pieces of the same colour vertically, horizontally or diagonally), player can not use this function.

4. And, instead of dropping a piece, each player has one times to remove one arbitrary piece of the opponent. It means that if a player chooses to drop a piece, player can not remove the opponent's piece and vice versa. In case of removing a piece, if there are any pieces above this piece, they will fall down.

5. In addition, students have to handle the exception of placing a piece at an inappropriate column (out board or column that has full pieces) by restarting the move. And it also counts as a violation.

6. If any players try to violate all of the above conditions over 3 times. This player will lose the game.

7. In each turn, the program has to show: the number of remaining violation, undo, and the player's name according to this turn.

Main idea and explanation

In this assignment, I didn't do the requirement number 2, 3 and 4. It means I can't do feature Undo, Block and Remove. So I just report what I can do.

1. Creating the board

Explain the idea :

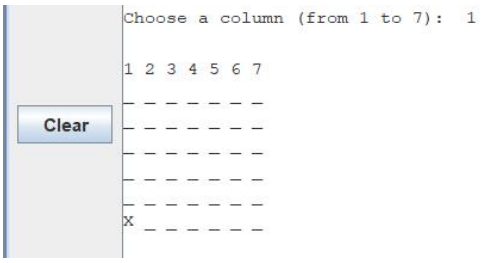
First of all, I declare an array with 42 characters (board 6x7) with the value '_'. The address distance of them are 4 bit, so I have an array with address respectively 0, 4, 8, 12,…, 164.

Therefore, the address of each character on the board should be:

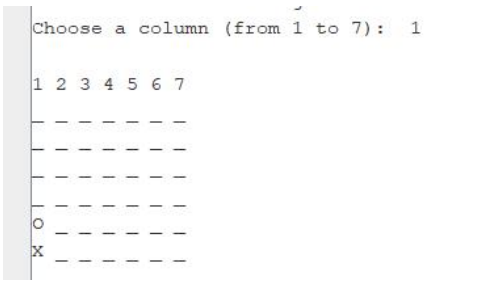
0	4	8	12	16	20	24
28	32	36	40	44	48	52
56	60	64	68	72	76	80
84	88	92	96	100	104	108
112	116	120	124	128	132	136
140	144	148	152	156	160	164

Now, the program require players to choose which column they want to place the counter. So there are 7 cases (from 1 to 7) and in each column the address of 2 blocks offset from each other by 28.

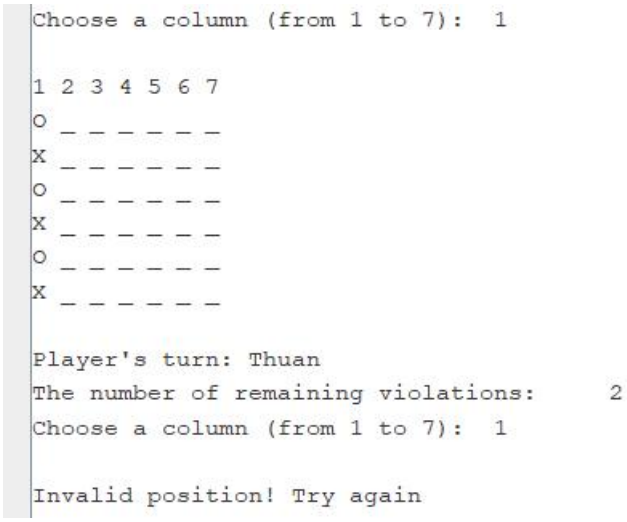
When the player choose a column, the counter will be placed in the biggest address block in that column.



If value of the block not equal '_', the address will be reduced by 28 and the counter in that address.



And it keep going till that column fulled, the program will request the player to choose another column.



2. Violations:

Also, the program provide each player 3 times to violate. Here are some rules:

First of all, in the first turn of each player, they must choose the center column to place counters.

Secondly, placing a piece at an inappropriate column (out board or column that has full pieces)

If they use all the number of violations, the players lose the game.

3. Check winning conditions:

In order to check the winning conditions. We must know how to win in this game. The player who succeeds in placing four of their marks in a horizontal, vertical, or diagonal row is the winner.

So overall, there are 3 styles of winning:
- 4 same pieces in a row
- 4 same pieces in a column
- 4 same pieces in a cross

With row case:

I will check all the cases where there are 4 identical marks in a row in a line.

First, I will check first 4 block in the first row.

0	4	8	12	16	20	24
28	32	36	40	44	48	52
56	60	64	68	72	76	80
84	88	92	96	100	104	108
112	116	120	124	128	132	136
140	144	148	152	156	160	164

Then, I will increase the address by 4 to check next 4 block until it reach the end of the row.

0	4	8	12	16	20	24
28	32	36	40	44	48	52
56	60	64	68	72	76	80
84	88	92	96	100	104	108
112	116	120	124	128	132	136
140	144	148	152	156	160	164

Similar for other rows

With column case:

I will check all the cases where there are 4 identical marks in a row in a column.

First, I will check first 4 block in the first column.

0	4	8	12	16	20	24
28	32	36	40	44	48	52
56	60	64	68	72	76	80
84	88	92	96	100	104	108
112	116	120	124	128	132	136
140	144	148	152	156	160	164

Then, I will increase the address by 28 to check next 4 block until it reach the end of the column.

0	4	8	12	16	20	24
28	32	36	40	44	48	52
56	60	64	68	72	76	80
84	88	92	96	100	104	108
112	116	120	124	128	132	136
140	144	148	152	156	160	164

Similar for other columns

With cross case:

I will check all the cases where there are 4 identical marks in a row in a cross.

First, I will check first 4 block in the first cross.

0	4	8	12	16	20	24
28	32	36	40	44	48	52
56	60	64	68	72	76	80
84	88	92	96	100	104	108
112	116	120	124	128	132	136
140	144	148	152	156	160	164

Then, I will increase the address by 24 to check next 4 block until it reach the end of the cross.

0	4	8	12	16	20	24
28	32	36	40	44	48	52
56	60	64	68	72	76	80
84	88	92	96	100	104	108
112	116	120	124	128	132	136
140	144	148	152	156	160	164

With each cross, the number of cases are also different. However the idea is similar for other crosses.

Conclusion

Through this assignment, I have learned how to implement a simple Four in a Row game in assembly language MIPS. I have learned more about using different MIPS instructions and logic of the game.

Thanks for reading