VIETNAM NATIONAL UNIVERSITY - HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER ARCHITECTURE - CO2007

ASSIGNMENT

FOUR IN A ROW - A NEW VERSION

HO CHI MINH CITY, FEBRUARY 2023

ASSIGNMENT SPECIFICATIONS

1 Outcomes

After finishing this assignment, students can proficiently use:

- MARS MIPS simulator.
- Arithmetic & data transfer instructions.
- Conditional branch and unconditional jump instructions.
- Procedures.

2 Introduction

Four in a Row is the classic two player game where you take turns to place a counter in an upright grid and try and beat your opponent to place 4 counters in a row.

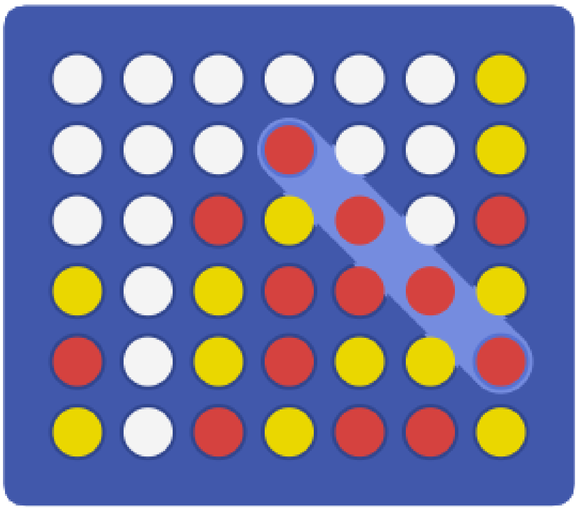

Figure 1: Four in a Row. [1]

The game is played with a **seven-column** and **six-row** grid, which is arranged upright. The players choose their name, and a game piece (X or O) is assigned to them randomly. Then they will place a piece in column. 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 (an example is shown in Figure 1). 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.

3 Requirements

Design and write MIPS assembly language for implementing a text-based Four in a Row game for two players as follows:

- First, let the players choose their name. After that, randomly assign the piece for them.
- Then, let the game begin. Four in a Row rules are based on the description at section 2.
- Finally, the output of the game is:
  1. If we have a winner, the program will show the name of the winner according to the number of the piece of this player on the board.
  2. On the other hand (TIE GAME), the program just show the result information with the final board.

Moreover, there are some addition requirements of the game, such as:

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.

4 Submission

- Students are requested to submit **the MIPS program(s)/source code (.asm files)** and **the Assignment report** to BK E-learning system (BKEL) no later than the last lab session of your group. Assignment must be done individually.
- Students have to demonstrate program(s) on MARS MIPS during the last lab session. Students who do not show up during the demonstration time will get 0 for the assignment.
- The report should not contain code. Instead, students should present the algorithms as well as the idea in your implementation.

5 Plagiarism

Similarity less than 30% in MIPS code is allowed. In other words, you will get 0 for assignment if your answers are similar to another student's more than 30%. We will use the MOSS system to check the similarity.

6 Rubric for evaluation

6.1 Friendly interface - 2 points

- Students can design and implement an amicable user interface so that players can play easily without any confusion (2 points).
- Students can design and implement a friendly user interface; however, players face some difficulty hen playing the game (1.5 points).
- Students can design and implement a user interface, but it is not friendly, or players need to do several steps for one move (1 point).
- Student can design and implement a user interface, but it fails to allow playing (0.5 points).

6.2 Application implementation - 6 points

- Students can implement an excellent application without any errors found (5.0 - 6.0 points).
- Students can implement a good application with some minor errors, but players do not need to restart the application to continue (3.0 - 5.0 points).

- Students can implement the application with some errors that prevent players from playing the game (1.0 - 3.0 point).
- Students cannot implement the application so that players can play/run (0 - 1.0 points).

6.3 Report - 2 points

- Students write such an excellent report that others can understand without any difficulty (2 points);
- Students write a good report but quite simple or lack of information to understand (1.5 points);
- Students write a report with a lot of code embedded without any explanation (1.0 points);
- Students write a simple report with most of the code attached (0.5 points)

References

[1] Four in a Row, AI Gaming, https://help.aigaming.com/game-help/four-in-a-row.

[2] Connect Four, Wikipedia, https://en.wikipedia.org/wiki/Connect_Four.

——————END——————