ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KĨ THUẬT MÁY TÍNH

KIẾN TRÚC MÁY TÍNH MỞ RỘNG

Assignment

Connect Four

Giáo viên hướng dẫn: Phạm Quốc Cường
Sinh viên: Nhóm 10
Hồ Trọng Nhân (Nhóm trưởng) - 2111899.
Hoàng Đức Nguyên - 2110393 .
Nguyễn Hoài Trung - 2110631.
Giản Đình Thái - 2112278 .

THÀNH PHỐ HỒ CHÍ MINH, THÁNG 12 2022

Contents

1. Danh sách thành viên & Công việc

STT	Họ và tên	MSSV	Phần làm bài	Phần trăm công việc
1	Hồ Trọng Nhân	2111899	- Thiết kế các khối lệnh và giải thuật. - Làm báo cáo.	25%
2	Giản Đình Thái	2112278	- Lập trình khối Check - Kiểm thử lại chương trình.	25%
3	Hoàng Đức Nguyên	2110393	- Lập Trình khối BoardGame, DrawTokens - Kiểm thử lại chương trình.	25%
4	Nguyễn Hoài Trung	2110631	- Lập trình khối Input và các câu dẫn. - Làm báo cáo.	25%

2. Đề bài

Sử dụng MIPS để hiện thực một trò chơi "Connect Four" Yêu cầu:

- Thiết kế và hiện thực giao diện với Bitmap Display (bắt buộc)
- Viết một báo cáo 10 trang mô tả thiết kế, giải thuật thực hiện, và hướng dẫn chơi (không chép code vào báo cáo)
- GV sẽ sử dụng phần mềm MOSS của Stanford để bắt lỗi sao chép code. Nếu trùng quá 30% thì bài của nhóm sẽ không hợp lệ.
- Thời gian nộp bài: trước ngày thi cuối kỳ của môn học.

3. Hướng dẫn chơi

3.1. Bàn chơi và dụng cụ chơi

Trò chơi Connect 4 có nhiều phiên bản bàn chơi như: $8 \times 8, 5 \times 4, 8 \times 7, ...$ nhưng phổ biến nhất là phiên bản 7×6. Bảng chơi 7×6 ở ngoài đời thật có dạng như hình:

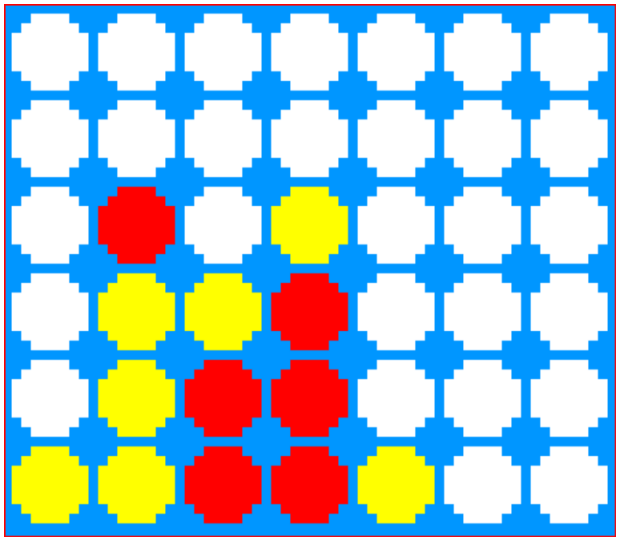
Ngoài ra, trò chơi còn có 21 quân cờ màu đỏ và 21 quân cờ màu vàng.

3.2. Luật chơi

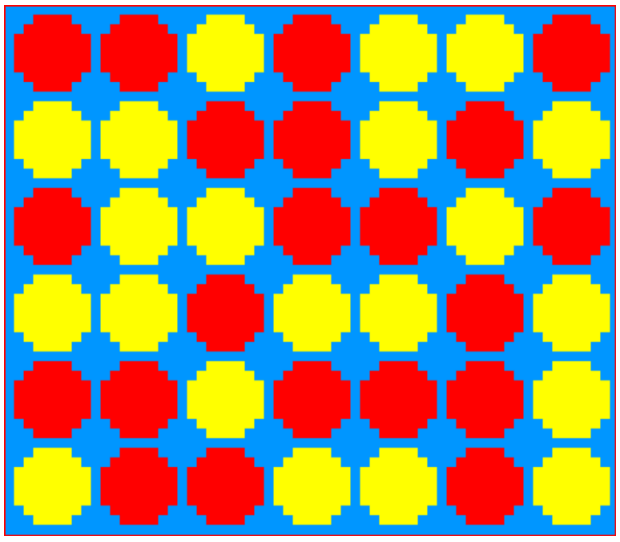
Trò chơi bao gồm 2 người chơi, mỗi người chơi chọn màu và lấy hết tất cả 21 quân cờ có màu đó và chọn người chơi đi trước.

Trò chơi được diễn ra luân phiên theo lượt giữa 2 người. Trong lượt của mình, bạn thả một quân cờ của mình vào một trong 7 cột của bàn cờ được dựng thẳng đứng, quân cờ đó rơi tự do cho đến khi chạm đáy hoặc nằm phía trên một quân cờ khác. Sau đó đến lượt người tiếp theo làm tương tự.

Người chiến thắng được xác định nếu sau một lượt bất kì người chơi nào sắp xếp 4 quân cờ màu của mình theo một hàng liên tiếp ngang dọc hay chéo sẽ là người chiến thắng.
Ví dụ 1 *Ví dụ cho trường hợp người chơi cầm quân cờ vàng thắng:*



Trò chơi sẽ được kết thúc khi có người chiến thắng hoặc nếu cả hai người chơi đều đã sử dụng hết 21 quần cờ, tức bảng được lấp đầy mà chưa tìm ra được người chiến thắng, trò chơi sẽ kết thúc với kết quả hòa.
Ví dụ 2 *Ví dụ cho trường hợp* 2 *người chơi hòa:*

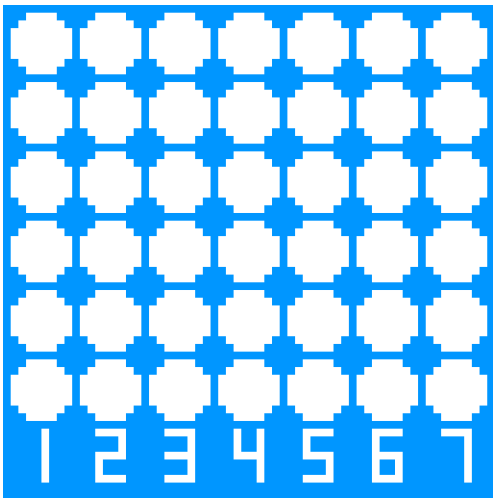


4. Mô tả thiết kế, giải thuật thực hiện:

4.1. Giao diện trò chơi:

Trong thực tế, game board của trò chơi Connect Four được thiết kế dạng hình chữ nhật với 6 hàng và 7 cột với những ô tròn nhỏ. Với việc hiện thực bằng Bitmap Display, các cấu hình của giao diện như sau:

1. Unit width in pixels: 8
2. Unit height in pixels: 8
3. Display width in pixels: 512
4. Display height in pixels: 512
5. Base Address for Display: 0x10040000 (heap)



Hình ảnh Game Board trên giao diện bằng Bitmap Display.

Lời dẫn khi bắt đầu trò chơi:

"Welcome to the Connect 4."

Lời dẫn thông báo đến lượt chơi:

"Player 1's turn: "

Người chơi sẽ tiến hành trò chơi bằng cách nhập input vào cửa sổ Run I/O.
Thông báo khi có người chiến thắng:

"Player 1 wins!!!"

Thông báo khi có kết quả hòa:

"Tie!!!"

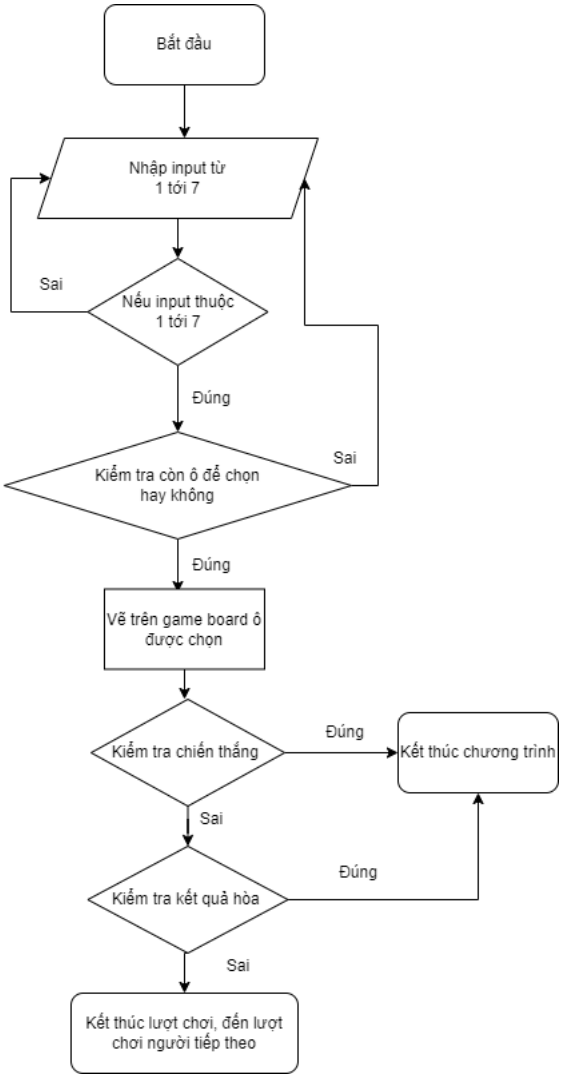
4.2. Mô tả khái quát giải thuật:

Ta sẽ dùng một mảng 42 phần tử (chỉ số từ 0 tới 41) để chỉ trạng thái của các ô trong game board theo thứ tự chỉ số các phần tử tương ứng với các ô như sau:

35	36	37	38	39	40	41
28	29	30	31	32	33	34
21	22	23	24	25	26	27
14	15	16	17	18	19	20
7	8	9	10	11	12	13
0	1	2	3	4	5	6

Với mỗi phần tử trong mảng mang giá trị ban đầu là 0 (tức đang ở trạng thái trống), khi mang giá trị là 1 thì ô đã được chọn bởi người chơi số 1, và mang giá trị là 2 khi ô đã được người chơi số 2 chọn.
Một thanh ghi (thanh ghi $s0) sẽ được dùng để đại diện cho lượt chơi của người 1 hoặc người 2. Thanh ghi này mang giá trị 1 khi đang ở lượt chơi của người 1 và mang giá trị là 2 khi đang ở lượt chơi của người 2. Sau mỗi lượt chơi, thanh ghi này sẽ thay đổi tuần tự để đến lượt chơi người tiếp theo. Với mỗi lượt chơi, phần tử có chỉ số chỉ đến ô được chọn trong mỗi lượt chơi cũng sẽ lưu trữ giá trị bằng giá trị trong thanh ghi này (1 hoặc 2, đại diện cho người chơi).
Đối với mỗi lượt chơi (hàm Input) sẽ lần lượt có những thao tác quan trọng sau:

1. Nhập input từ 1 tới 7 tương ứng với việc chọn các cột thứ nhất tới cột thứ bảy trong game board để thả các token như trong thực tế.
2. Kiểm tra input này, nếu cột không còn ô trống, tiến hành nhập lại input, nếu còn ô trống, xác định ô sẽ được chọn ở cột nào và hàng nào (hàm CheckInput).
3. Vẽ trên game board (Bitmap Display) token ở ô vừa được chọn vừa được xác định ở bước trên (hàm DrawToken).
4. Kiểm tra liệu có kết quả chiến thắng hay không (hàm CheckWin).
5. Kiểm tra liệu ván chơi có kết quả hòa hay không (hàm CheckTie).
6. Nếu không, tiếp tục lượt chơi của người tiếp theo.



Minh họa lượt đi của người chơi bằng lưu đồ.

4.3. Mô tả chi tiết các thao tác thực hiện trong giải thuật

Ngay khi bước vào trò chơi, sẽ có câu dẫn để giới thiệu ván chơi bắt đầu.
Bước tiếp theo, vẽ game board ban đầu trên Bitmap Display và gán thanh ghi $s0 giá trị ban đầu là 1 (tức lượt chơi đầu tiên là của người 1).

4.3.1. Hàm GameBoard:

Vẽ game board như đã mô tả.

4.3.2. Hàm Input:

1. Hiện câu dẫn thông báo lượt chơi của người 1 hoặc 2 (tùy theo giá trị đang được lưu trữ trong thanh ghi $s0).

2. Thông báo câu dẫn nhập input, yêu cầu người chơi nhập input bao gồm số có giá trị từ 1 tới 7, tiến hành nhập lại nếu ngoài 1 tới 7. Input này sẽ đươc lưu trữ trong thanh ghi $s4.
3. Đi đến hàm CheckInput để kiểm tra liệu còn ô trống để trong game board để chọn hay không, nếu không nhập lại input. Ngược lại nếu có, tiến hành cập nhật thanh ghi $s4 chứa chỉ số cột chứa ô được chọn trong game board và $s7 chứa chỉ số hàng chứa ô được chọn trong game board (các chỉ số này đều được tính từ 0).
   Ví dụ, ô được chọn nằm ở hàng đầu tiên (tính từ dưới lên) và cột thứ hai thì $s4 mang giá trị 1 và $s7 mang giá trị 0.
4. Lưu trữ giá trị đang có trong $s0 (1 hoặc 2) vào phần tử mang chỉ số đại diện cho ô được chọn trong game board (phần tử mang chỉ số $s4+$s7x7) sau khi thực hiện hàm CheckInput.
5. Với $s4 đại diện cho chỉ số chỉ số cột chứa ô được chọn trong game board và $s7 chứa chỉ số hàng chứa ô được chọn trong game board, ta tiến hành vẽ token ứng với ô được chọn trong game board ở hàm DrawToken.
6. Đi đến hàm CheckWin kiểm tra liệu có chiến thắng hay không.
7. Đi đến hàm CheckTie kiểm tra liệu có kết quả hòa hay không.
8. Cập nhật thanh ghi $s0 để chuẩn bị lượt chơi mới. Nếu thanh ghi $s0 đang mang giá trị là 1, tiến hành tăng thành 2. Ngược lại, nếu đang mang giá trị là 2, tiến hành giảm thành 1.
9. Lặp lại hàm Input với $s0 vừa được cập nhật (j Input).

4.3.3. Hàm CheckInput:

Với mảng 42 phần tử đại diện cho trạng thái từng ô cùng với input từ 1 tới 7 hàm CheckInput ta sẽ kiểm tra liệu còn ô trống được chọn hay không.
Để việc kiểm tra dễ dàng, ta sẽ sử dụng một mảng 7 phần tử để lưu số ô đã được chọn trong các cột đó. Do đó, ban đầu mảng này sẽ được khai báo gồm 7 phần tử mang giá trị 0 (mảng này được đặt tên int_col).
Việc kiểm tra sẽ gồm các bước như sau:

1. Giảm giá trị đang chứa trong thanh ghi $s4 (giá trị của input) xuống 1 đơn vị để phù hợp với việc chỉ số cột của ô được chọn tính từ 0 như đã quy ước ban đầu.
2. Kiểm tra giá trị của phần tử có chỉ số được đại diện bởi giá trị đang lưu trữ trong $s4 trong mảng int_col, nếu giá trị này lớn hơn hoặc bằng 6 (tức không còn ô trong cột có chỉ số $s4 trống), tiến hành quay lại hàm Input nhập lại input (j Input), ngược lại gán giá trị này vào trong $s7.
3. Tăng giá trị phần tử vừa được truy xuất trong mảng int_col lên 1 đơn vị nhằm thể hiện thêm môt ô ở cột này vừa được chọn.

4.3.4. Hàm DrawToken:

Với giá trị $s4 và $s7 đại diện cho chỉ số cột và hàng chứa ô vừa được chọn, ta sẽ tiến hành vẽ các token với màu tương ứng với người chơi trên game board.
Các token được vẽ theo màu vàng tương ứng người chơi 1 (khi $s0=1) và màu đỏ tương ứng người chơi 2 (khi $s0=2).

Thanh ghi $t9 sẽ chứa mã HEX của màu sẽ được vẽ cho token.

Hình ảnh token được vẽ trên giao diện bằng Bitmap Display.

4.3.5. Hàm CheckWin:

Ta sẽ tiến hành kiểm tra xem liệu có 4 ô liên tiếp nào đã được chọn bởi người chơi đang thực hiện lượt chơi hiện tại hay chưa.
Đồng thời, ta sẽ kiểm tra liệu tất cả các ô trong game board đã được chọn hết hay chưa. Tín hiệu mảng đã đầy sẽ được lưu trữ trong thanh ghi $s6. Thanh ghi này ban đầu được khơi tạo bởi giá trị 1 (tức đang được giả sử đã đầy), sau mỗi lần lặp qua các ô, nếu ô còn trống $s6 sẽ lập tức được đổi thành giá trị 0 (tức vẫn còn ô trống).
Ta sẽ lặp qua lần lượt các ô trong board game để kiểm tra lần lượt 4 trường hợp, do đó ta cần có thanh ghi $t0 ban đầu có giá trị bằng 0 để làm biến đếm (nếu $t0 lớn hơn 41, tức đã lặp xong, quay lại vị trí trước khi gọi hàm, jr $ra).
Bốn trường hợp sẽ được kiểm tra như sau:

1. Trường hợp 1: Kiểm tra đường chéo lên bên phải.
   Nếu giá trị trong $t0 chia 7 có số dư nhỏ hơn 4:
   (a) Ta tiến hành kiểm tra lần lượt các phần tử trong mảng int_arr có chỉ số lần lượt là $t0, $t0+8, $t0+16, $t0+24.
   (b) Nếu một trong 4 giá trị của các phần tử trên khác giá trị đang được lưu trữ trong thành ghi $s0, rẽ nhánh tới kiểm tra trường hợp tiếp theo.
   (c) Ngược lại nếu cả 4 phần tử đều bằng giá trị được lưu trữ trong $s0. In câu dẫn người chiến thắng, kết thúc chương trình (j Exit).

   Ngược lại, tức giá trị trong $t0 chia 7 có số dự lớn hơn hoặc bằng 4, rẽ nhánh tới kiểm tra trường hợp tiếp theo.

35	36	37	38	39	40	41
28	29	30	31	32	33	34
21	22	23	24	25	26	27
14	15	16	17	18	19	20
7	8	9	10	11	12	13
0	1	2	3	4	5	6

2. Trường hợp 2: Kiểm tra đường chéo lên bên trái.
   Nếu giá trị trong $t0 chia 7 có số dư lớn hơn 2:
   (a) Ta tiến hành kiểm tra lần lượt các phần tử trong mảng int_arr có chỉ số lần lượt là $t0, $t0+6, $t0+12, $t0+18.
   (b) Nếu một trong 4 giá trị của các phần tử trên khác giá trị đang được lưu trữ trong thành ghi $s0, rẽ nhánh tới kiểm tra trường hợp tiếp theo.

(c) Ngược lại nếu cả 4 phần tử đều bằng giá trị được lưu trữ trong $s0. In câu dẫn người chiến thắng, kết thúc chương trình (j Exit).

Ngược lại, tức giá trị trong $t0 chia 7 có số dư nhỏ hơn hoặc bằng 2, rẽ nhánh tới kiểm tra trường hợp tiếp theo.

35	36	37	38	39	40	41
28	29	30	31	32	33	34
21	22	23	24	25	26	27
14	15	16	17	18	19	20
7	8	9	10	11	12	13
0	1	2	3	4	5	6

3. Trường hợp 3: Kiểm tra đường ngang.
Nếu giá trị trong $t0 chia 7 có số dư nhỏ hơn 4:

   (a) Ta tiến hành kiểm tra lần lượt các phần tử trong mảng int_arr có chỉ số lần lượt là $t0, $t0+1, $t0+2, $t0+3.

   (b) Nếu một trong 4 giá trị của các phần tử trên khác giá trị đang được lưu trữ trong thành ghi $s0, rẽ nhánh tới kiểm tra trường hợp tiếp theo.

   (c) Ngược lại nếu cả 4 phần tử đều bằng giá trị được lưu trữ trong $s0. In câu dẫn người chiến thắng, kết thúc chương trình (j Exit).

   Ngược lại, tức giá trị trong $t0 chia 7 có số dự lớn hơn hoặc bằng 4, rẽ nhánh tới kiểm tra trường hợp tiếp theo.

35	36	37	38	39	40	41
28	29	30	31	32	33	34
21	22	23	24	25	26	27
14	15	16	17	18	19	20
7	8	9	10	11	12	13
0	1	2	3	4	5	6

4. Trường hợp 4: Kiểm tra đường dọc.
Nếu giá trị trong $t0 cộng với 21 nhỏ hơn 41:

   (a) Ta tiến hành kiểm tra lần lượt các phần tử trong mảng int_arr có chỉ số lần lượt là $t0, $t0+7, $t0+14, $t0+21.

   (b) Nếu một trong 4 giá trị của các phần tử trên khác giá trị đang được lưu trữ trong thành ghi $s0, rẽ nhánh tới kiểm tra trường hợp tiếp theo.

   (c) Ngược lại nếu cả 4 phần tử đều bằng giá trị được lưu trữ trong $s0. In câu dẫn người chiến thắng, kết thúc chương trình (j Exit).

   Ngược lại, tức giá trị trong $t0 cộng 21 lớn hơn 41, tăng biến đếm trong thanh ghi$t0 lên 1, tiến hành kiểm tra ô tiếp theo.

35	36	37	38	39	40	41
28	29	30	31	32	33	34
21	22	23	24	25	26	27
14	15	16	17	18	19	20
7	8	9	10	11	12	13
0	1	2	3	4	5	6

4.3.6. Hàm CheckTie:

Hàm sẽ kiểm tra kết quả hòa có xảy ra hay không.
Nếu thanh ghi $s6 có giá trị là 0, tức vẫn còn ô chưa được chọn, quay lại hàm Input để tiếp tục lượt chơi mới (jr $ra).
Ngược lại, in câu dẫn kết quả hòa ra màn hình, kết thúc trò chơi (j Exit).

5. Kết luận

Thông qua bài tập mở rộng này, chúng em đã có cơ hội để cải thiện kỹ năng lập trình MIPS cũng như cách để lập trình một trò chơi ở mức cơ bản. Ngoài ra, chúng em cũng cải thiện được các kỹ năng như làm việc nhóm, trình bày báo cáo khoa học, phân chi công việc và quản lý thời gian hợp lý. Đây là những kỹ năng quan trọng sẽ giúp ích chúng em rất nhiều trong các môn học sau này.