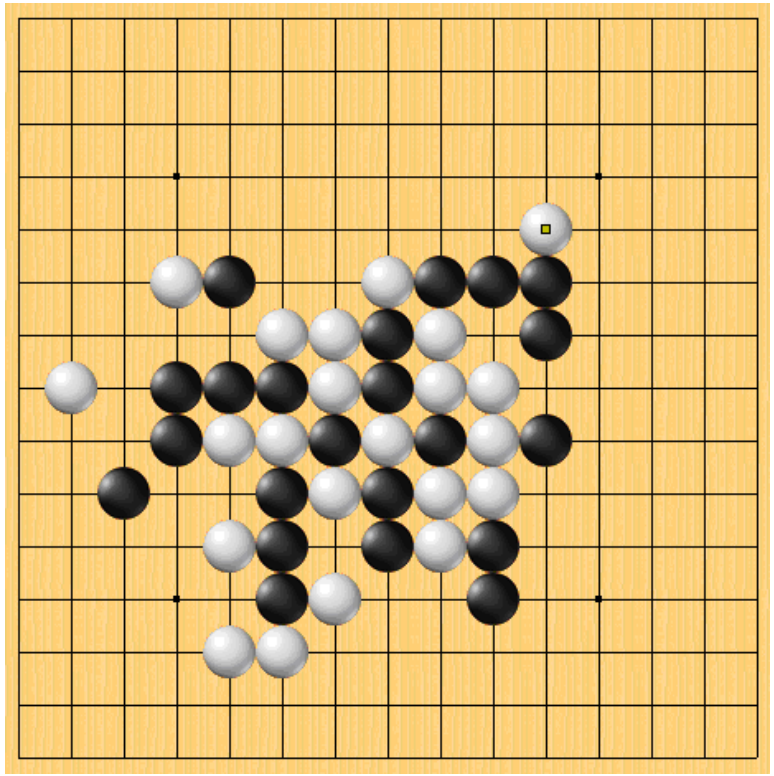
Data Structure

HW1

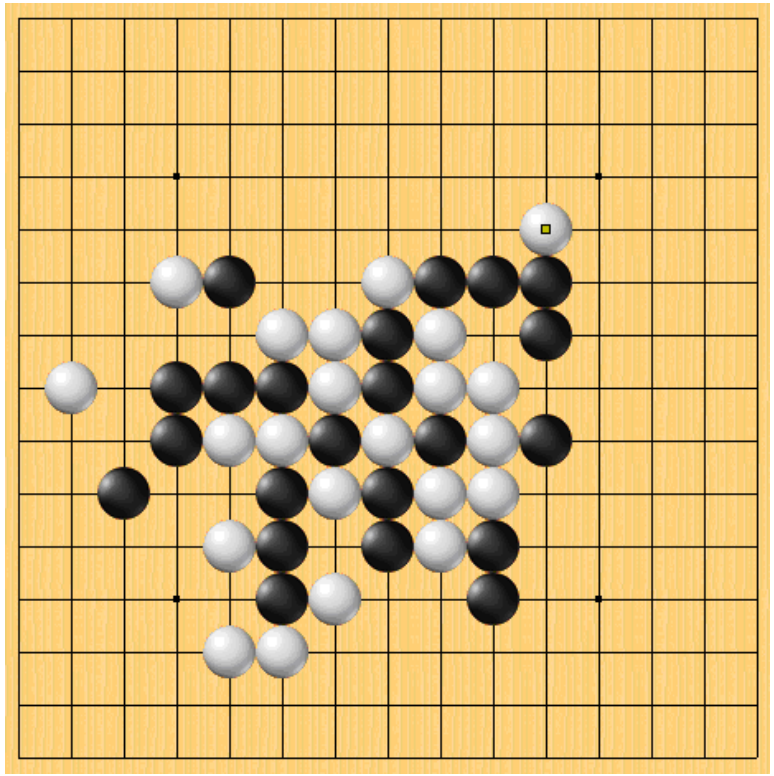
과제1 – 오목

- 오목은 상대방보다 먼저 대각선, 가로, 세로 방향 중 하나로 5개의 연속된 돌을 놓으면 승리하는 게임이다
- 오목은 사용자가 입력한 크기의 오목판 위에서 진행되며, 각 사용자가 돌을 번갈아 둔다.
- 본 과제1는 UNIX 기반 library 인 'ncurses.h' 를 사용하여 구현한다.



해당 과제1의 목적

- 1학년 과정에서 학습한 c/c++ 언어의 변수, 반복문, 조건문, 파일 입출력 등의 문법적 요소들을 복습하고 자료구조 프로그램을 준비한다.
- 함수를 사용하여 프로그램을 모듈화 하는 법 사용.
- 제공된 코드를 이해하여 응용.
- 제공된 library 에 대한 documentary 를 찾아보고 이해하며 프로그램에 적용.



과제1 – 오목 (과정 설명)

- 과정 1.
  사용자로부터 오목판의 크기를 입력 받고 해당 크기의 오목판을 초기화 하고 게임을 시작한다.
- 과정 2.

사용자가 누른 입력 키 방향으로 커서(cursor) 를 이동시킨다. 원하는 위치로 이동한 후, Enter 또는 Space 를 입력하여 돌을 둔다.

- 과정 3.

원하는 위치에 돌을 두었으면, 게임이 끝이 났는지 확인한다. 만약 끝이 났으면 어떤 사용자가 승리했는지 출력 후 게임을 마친다.

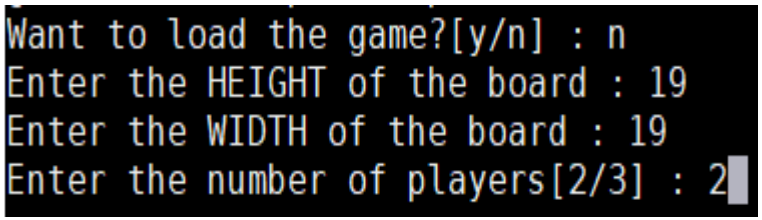
- 추가 과정 a1

게임 도중, 게임을 저장할 수 있다. 숫자 '1' 을 입력하면 게임을 저장하고 마친다. 저장한 게임을 다시 불러 올 수도 있다. 게임 시작 시, 저장했던 이름을 입력하면 해당 게임을 불러와서 저장했던 그대로 재개한다.

- 추가 과정 a2

3 인 4 목 게임을 구현한다. 해당 게임은 3 인이 게임에 참가하며, 4 개의 연속된 돌을 둔 사용자가 승리하는 게임이다.

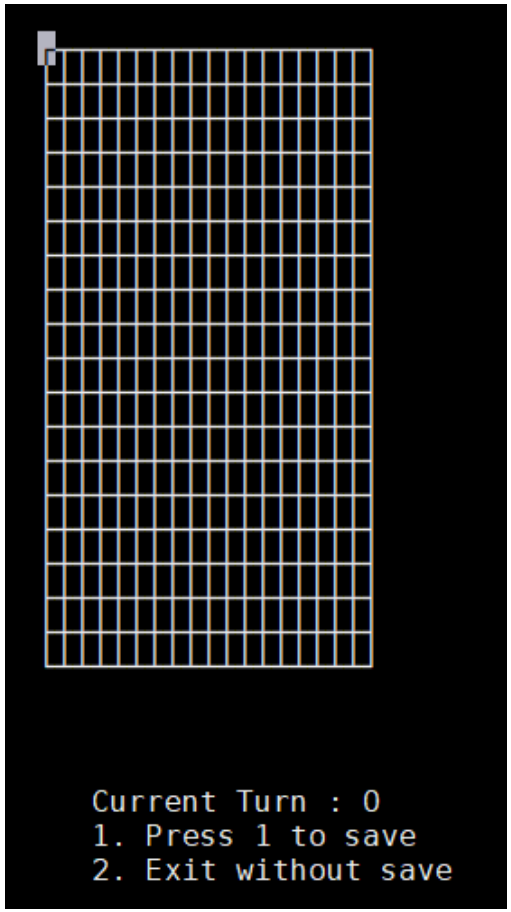
과제1 – 오목 (실행화면)



- 오목 게임을 시작하면 왼쪽 그림과 같은 입력 화면이 나타난다. 오목판의 크기를 입력 받는다. (세로, 가로 순)

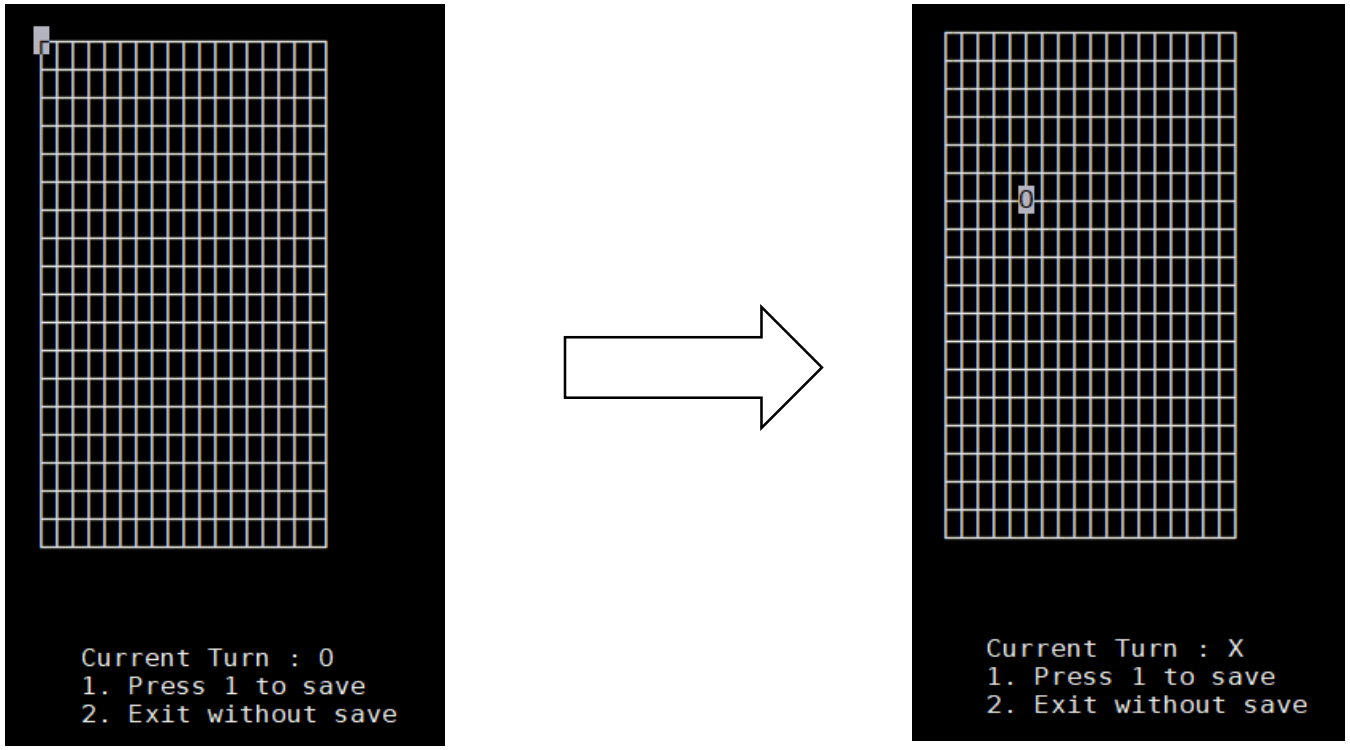
- 추가 기능 구현 시, 아래와 같은 입력 문구를 추가한다.
  (a1) 저장된 게임을 불러올 것인지,
  (a2) 사용자의 수는 몇 명으로 할 것인지를 입력 받는다.

과제1 – 오목 (실행 화면)



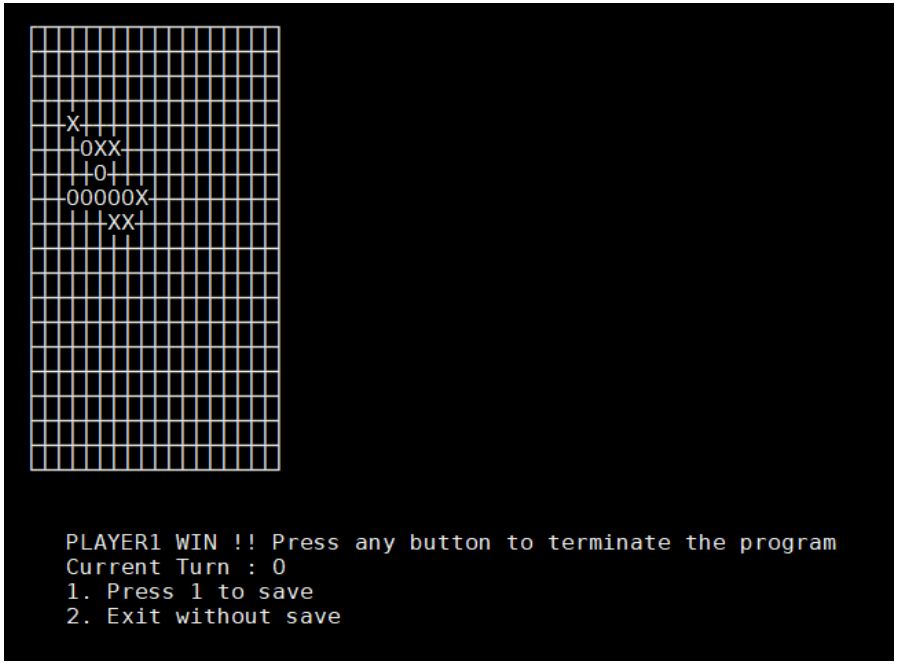
- 게임 시작 시, 다음과 같이 화면을 입력 받은 오목판 크기로 초기화 하고, cursor 는 왼쪽 그림과 같이 바둑판의 왼쪽 위 모서리에 위치한다.
- 처음 두는 사용자는 'O' 로 표시하고, 두 번째는 'X' 로 표시하며, 세 번째 사용자가 존재하면 'Y' 로 표시한다.
- 키보드의 arrow key (상,하,좌,우) 를 이용하여 cursor 를 이동시킨다.

과제1 – 오목 (실행화면)



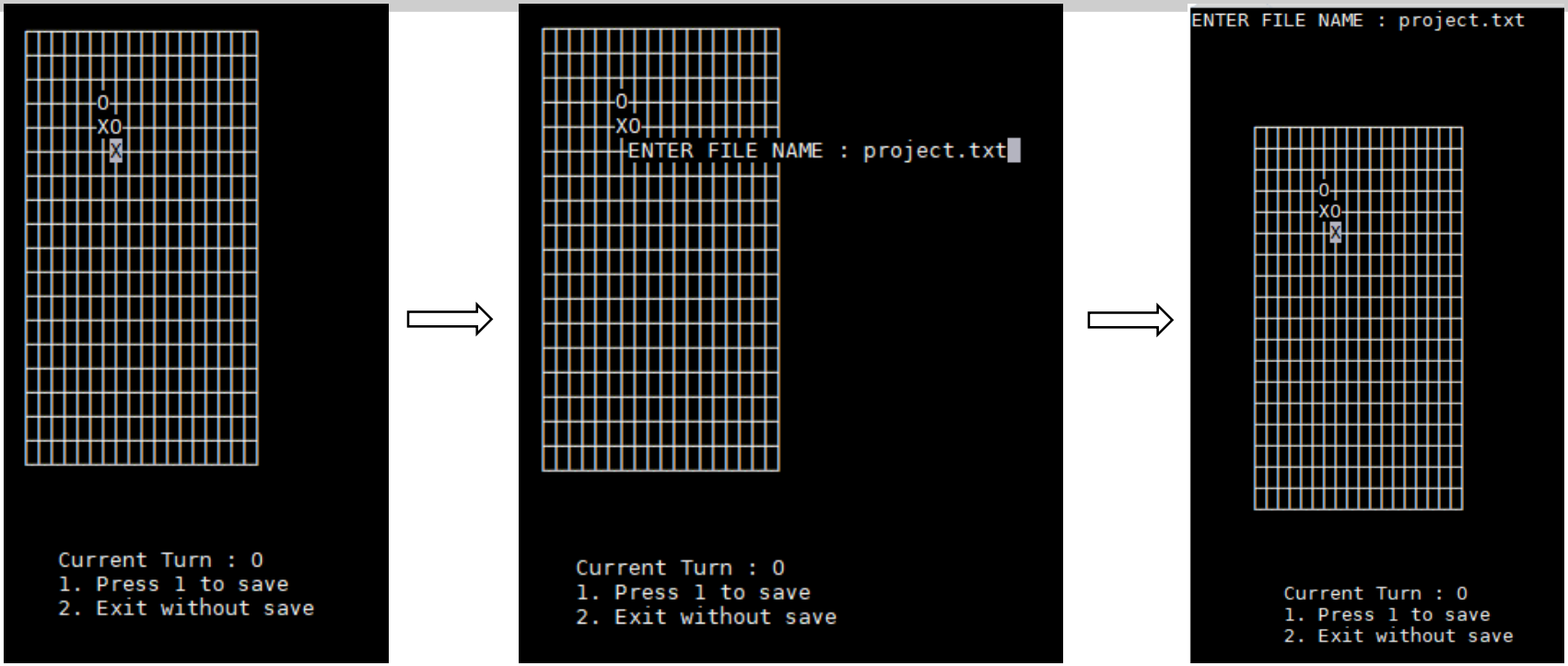
- 위 그림은 커서를 이동시켜서, space bar 또는 enter key 를 눌러 'O' 사용자가 본인의 차례를 마쳤을 때의 그림이다.
- 커서를 이동시킬 때, 현재 차례의 돌이 표시될 필요는 없다.
- Current Turn 은 다음사용자로 변경된다.

과제1 – 오목 (종료 화면)



- 위 그림과 같이 한 사용자의 5 개의 돌이 연속되어 위치되어 있으면 해당 사용자가 승리하였다고 표시한다.
- 이후 아무 입력을 받으면 프로그램을 종료한다.

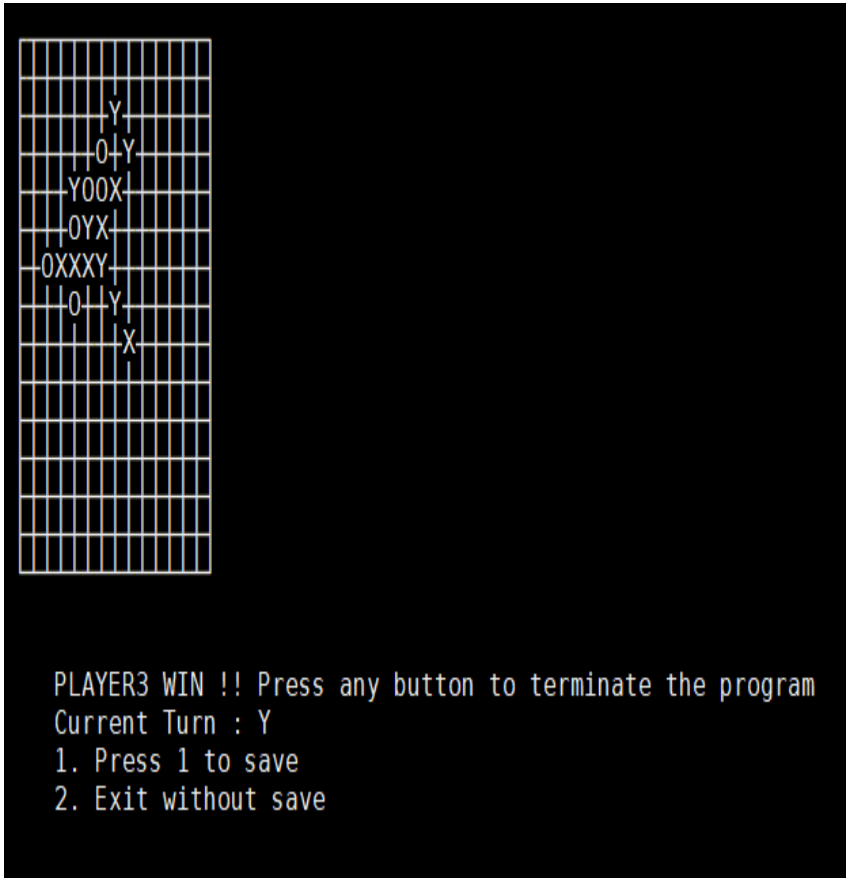
과제1 – 오목 추가구현 (a1)



- 왼쪽 그림과 같이 게임 도중, 저장하고 마칠 수 있다.
- 가운데 그림과 같이 '1' key를 누르고, 저장할 이름을 입력하면 종료된다.
- 오른쪽 그림과 같이 저장한 파일 이름을 입력하여 게임을 저장한 그대로 재개할 수 있다. (커서의 위치까지 저장할 당시 그대로 한다)

과제1 – 오목 추가구현 (a2)

- 오른쪽 그림과 같이, 3인이 게임을 진행할 경우 5목이 아닌 4목으로 한다.
- 순서는 player1 부터 3까지 순서대로 진행하고, 연속으로 4 개의 돌을 먼저 둔 사용자가 승리한다.



과제1 – 오목 (구현조건)

- **본 과제1는 아래와 같은 조건 하에서 작성한다.**
  - 조건
    - cspro 서버 상에서 수행하도록 한다.
    - 주어진 “user_omok.c”를 기반으로 (skeleton code) 프로그램을 작성.
      (주어진 **기존 코드 변경 불가**)
    - 사용 가능한 문법은 ncurses library 와 기존 1학년 C/C++ 에서 배운 내용으로만 한정한다. **큐, 스택, 트리, 그래프 등의 자료구조를 구현하거나 이미 구현된 기존 자료구조를 호출해서 사용할 수 없다**.
      (사용할 경우 0점 처리)
    - 부록에 정의한 함수는 기능에 맞게 **반드시** 구현해야 한다.
    - 컴파일 : gcc user_omok.c -lncurses (cspro 서버)로 컴파일 되어야 한다.

  (C++를 사용한 경우에는 제출시 본인의 컴파일 명령어를 같이 제출할 것.)

과제1 – 오목 (평가)

- 시험 및 평가 방식:작성한 프로그램이 다음 조건들을 만족하며 안정되게 작동하는지 확인한다. (다음은 평가 예)
  - 커서가 오목판을 벗어나지 않는다. 커서의 이동에 대해서 안정적으로 작동한다.
  - 승리 조건을 오류없이 만족한다.
  - 프로그램 종료 후, terminal 로 정상적으로 원상 복귀된다.
  - 입력 prompt 로 입력을 받을 경우, keyboard 입력 문구가 보여야하며, 입력 prompt 로 입력을 요구 받은 경우가 아닌 경우, keyboard 입력은 화면에 보이지 않는다.
  - 메뉴에 표시된 keyboard 입력을 받으면 이에 상응하는 action 이 실행된다. (저장:1, 저장하지 않고 종료:2)
  - 각 입력에 대해서 정상적으로 동작한다. (Enter, Space, 방향키 등등)
  - 추가 기능 구현 시, 설명한 기능들이 모두 정상적으로 작동한다.
  - 제시한 추가 기능 이외의 추가 기능을 구현할 경우, 추가 점수를 더 받을 수 있다.
  - 부록에 첨부한 함수들을 구현한다.
  - 전역 변수의 사용은 오목판의 세로, 가로 길이를 담는 변수만 가능하다.

부록

user_omok.c

- int main()
  - 게임 시작 시 실행되며, 오목판의 크기를 입력 받고, 추가 기능 구현 시 필요한 입력을 받는다.
  - W새로운 윈도우를 불러올 뿐만 아니라, 터미널을 원상 복귀 시키기 위한 함수를 호출하여 종료시킨다.
- int gameStart(WINDOW *win)
  - 오목판을 초기화 시키는 함수를 호출한다.
  - 게임 진행에 요구되는 사항들(오목판 초기화, 오목판 그리기, 메뉴 그리기, 입력 받기, 입력에 대한 적절한 행동하기)을 순서대로 실행한다.
- int paintBoard(int **board, WINDOW *win, int row, int col)
  - 오목판과 올려진 돌을 화면에 출력시키는 역할을 한다.

부록

user_omok.c (명시한 함수의 인자는 변경 가능함)

- int Action(WINDOW *win, int **board, int keyin, int *row, int *col, int *turn, int players)
  - 사용자의 입력(keyin)에 대해서 그에 상응하는 행동을 구현한다.
  - 커서 이동, 바둑돌 두기, 게임 저장(추가 구현시), 게임 종료를 반드시 포함한다.
  - 한 사용자가 이겨서 게임이 끝날 경우 1을 return 하고 이외엔 0을 return 한다.
- int checkWin(형식자유)
  - 게임이 끝났는지를 체크한다.
- void saveGame(int **board, int row, int col, int turn)
  - 게임을 입력 받은 파일에 저장한다.
- int** readSavedGame(int **board, int *row, int *col, int *turn)
  - 저장했던 게임의 파일 이름을 읽어들여서 return 한다.

부록

ncurses.h

- 유닉스 환경 상에서, 커서를 움직이거나, 색깔을 추가하는 등의 기능을 사용하고 싶을 때 이용하는 library.
- 제공하는 함수들과 자세한 사용방법은 http://www.cs.ukzn.ac.za/~hughm/os/notes/ncurses.html 를 참고한다.
- 본 과제1에선, ncurses 로 표시한 화면상(WINDOW *win)에 while 문을 사용하여 저장된 오목판을 계속적으로 출력하는 형식으로 진행된다.

부록 (ncurses.h)

- 프로그램 상에서 작동 구조 설명
  - ‘board’ 변수를 이용하여서 실시간으로 화면에 출력할 그림들을 저장한다. (오목판, 사용자가 위치시킨 돌 등)
  - 사용자는 새로운 WINDOW 변수를 선언하고 초기화 한다. 사용자가 keyboard 를 통해서 입력에 따라서 특정한 action 을 취한 뒤 board 를 다시 출력한다.
  - 이 때, cursor 를 이동하는 action 은 ncurses library 내장 함수를 사용한다.
  - 위 과정을 계속해서 반복한다.

```
void gameStart(WINDOW *win, int load, int players){
        int **board;
        int row = 0;
        int col = 0;
        int keyin;
        wmove(win, row, col);
        board = initBoard(board, &row, &col, &turn, load); // Initiating the board

        while(1){
                /*
                        This While loop constantly loops in order to
                        draw every frame of the WINDOW.
                */

                // TODO LIST
                  // PAINT THE BOARD
                  // PAINT MENU
                  // MOVE CURSOR TO THE LAST POINT
                  // GET KEYBOARD INPUT
                  // DO ACTION ACCORDING TO THAT INPUT
                  // update WINDOW
        }

        return;
}
```

제출 방법 및 형식

- 제출 마감: 3월 31일 자정, 사이버캠퍼스 과제 메뉴를 통해 파일명 양식에 맞게 제출. (gcc user_omok.c –lncurses (cspro 서버)로 컴파일 되어야 한다. C++를 사용한 경우에는 제출시 본인의 컴파일 명령어를 같이 제출할 것.)
- 제출 기한 이후의 메일은 미제출로 간주
- 과제 채점은 Microsoft Visual Studio 2022 기준 (cspro 계정과 실행환경이 완료 되지 않았을 경우에만, 추후 공지할 예정)
- Copy 검사 실시함. 1회 발견시 과제 0점, 2회 발견시 F학점.
- 제출 내용(이후 슬라이드 참고):
  - 1. 보고서
  - 2. 소스코드

제출 방법 및 형식

소스코드:

- 파일 이름: HW1_학번_문제번호.c(or .cpp)

  ex)HW1_20240000_1.c(or .cpp)

- 확장자는 **무조건** .c 혹은 .cpp 이어야 함.

이외의 파일(.txt 등)은 **절대** 받지 않음(**미제출로 간주**)

- 컴파일 에러가 발생할 경우 0점 처리
- 무한 루프 / 세그멘테이션 오류는 해당 testcase 0점 처리
- 입출력 양식이 틀릴 경우 감점

제출 방법 및 형식

- **보고서: 형식 자유**
- **파일 이름: HW1_학번_Document.pdf**
- **반드시 PDF 파일로 제출할 것; 이외의 파일(.docx, .hwp 등)은 절대 받지 않음(미제출로 간주)**
  - 전체 프로그램의 Flow chart 및 본인이 구현한 각 함수의 Flow chart를 포함하여, 구현 내용을 설명한다.
  - 본인 스스로의 프로그램 평가 내용을 포함하여 설명한다.