HW3 - Term Project Report

<curses 라이브러리를 이용한 2048 퍼즐 게임 구현>

과목: ELEC0462-002 시스템 프로그래밍 | 학부 : 컴퓨터학부 | 학번 : 2020114968 | 이름 : 유우석

1. 주제 소개

- 이 프로젝트는 Curses 라이브러리를 이용하여 만든 2048 퍼즐 게임입니다.
- 2048 게임은 4x4 격자 위에서 블록을 움직이는 방식으로 진행되며, 플레이어의 목표는 같은 숫자의 블록을 합쳐 최대한 큰 숫자를 만들어 나가는 것입니다.

다음은 2048 퍼즐 게임의 규칙입니다.
1) 게임 시작 : 빈 4x4 격자에, 2 블록 2개와 4 블록 1개가 랜덤한 위치에 생성됩니다.
2) 움직임(up, down, left, right) : 격자 위의 모든 블록을 그 방향으로 움직일 수 있습니다. 만약 해당 방향에 블록이 움직일 공간이 있거나 같은 숫자의 블록이 있다면, 블록은 그 방향으로 움직입니다. 움직였을 때 같은 숫자의 블록이 만나게 된다면, 두 블록은 합쳐지며 숫자는 원래 숫자의 2배가 됩니다.
3) 새 블록 생성 : 한 번의 움직임 이후, 격자의 빈 공간 중 랜덤한 위치에 새로운 블록이 생성됩니다. 생성되는 블록은 2 또는 4입니다.
4) 게임 오버 : 더 이상 어느 방향으로도 움직일 수 없을 때, 즉 모든 격자가 블록으로 차 있고 어떤 블록도 합쳐질 수 없을 때 게임은 종료됩니다.

- 터미널 상에서 작동하는 게임이고, 필요로 하는 터미널의 최소 크기가 정의되어 있습니다. 터미널 크기를 맞게 설정하지 않으면 실행이 되지 않습니다.
- 세 개의 페이지(시작 페이지, 2048 게임 페이지, 게임 종료 페이지)로 구성되어 있습니다.

2. 기능 및 설계 & 프로그램 실행 및 수행 결과

- 사용한 라이브러리 및 함수

1) curses.h

getmaxyx(stdscr, HEIGHT, WIDTH) : 터미널의 현재 크기를 받아온다.
initscr() : curses 라이브러리를 초기화하고, 터미널 스크린을 준비한다.
noecho() : 사용자가 입력한 키의 출력을 터미널에 보이지 않게 한다.
endwin() : curses 모드를 종료하고, 터미널을 원래 상태로 복구한다.
clear() : 터미널 스크린을 모두 지운다.
refresh() : 변경 사항을 터미널에 바로 반영한다.
move(row, col) : 지정한 row와 col 위치로 커서를 이동시킨다.
addstr(string) : 지정한 string을 현재 커서 위치에 출력한다.

2) unistd.h

fork() : 현재 프로세승의 복제본인 자식 프로세스를 생성한다.
execv(argv[0], argv) : 현재 프로세스를 새로운 프로세스 이미지로 대체한다.

3) signal.h

signal(signum, handler) : 특정 시그널에 대한 처리 함수(핸들러)를 설정한다

4) sys/wait.h

wait(NULL) : 자식 프로세스가 종료될 때까지 부모 프로세스가 대기하도록 한다.

5) string.h

strcpy(s1, s2) : s2 문자열을 s1으로 복사한다.
strcat(s1, s2) : s1 문자열의 끝에 s2 문자열을 붙인다.

6) stdlib.h

srand(seed) : 난수 생성기를 초기화한다.
rand() : 0부터 RAND_MAX 사이의 난수를 생성한다.

- 사용자 정의 함수

1) mainlayout.c

void setMainLayout(void) : 시작 화면을 설정하는 함수이다.

2) 2048.c

int board[4][4]; : 4x4 메인 격자이다.
short int check[4]; : 0 1 2 3 - 상 좌 하 우 방향 움직임이 가능하다면 0, 불가능하다면 1을 담는 배열이다.
int my_score; : 2048 퍼즐 게임에서의 현재 점수를 담는 변수이다.

int up(void)
: 윗 방향 움직임을 구현한다. 움직임 도중 합쳐진 모든 블록에 대하여 점수를 계산해서 반환해준다.

int down(void)
: 아랫 방향 움직임을 구현한다. 움직임 도중 합쳐진 모든 블록에 대하여 점수를 계산해서 반환해준다.

int left(void)
: 왼쪽 방향 움직임을 구현한다. 움직임 도중 합쳐진 모든 블록에 대하여 점수를 계산해서 반환해준다.

int right(void)
: 오른쪽 방향 움직임을 구현한다. 움직임 도중 합쳐진 모든 블록에 대하여 점수를 계산해서 반환해준다.

int generate(int init)
: init이 1인 경우 초기 4x4 격자의 랜덤한 위치에 3개의 블록을 생성하고, 0인 경우 움직임을 통해 격자의 내용이 변했을 경우 격자의 빈공간 중 랜덤한 위치에 1개의 블록을 생성한다.

int moved(int a[4][4]);
: 전역변수 board 격자와 현재 a 격자의 모든 원소들을 비교하여 움직임이 발생했는지 판단하는 함수이다.

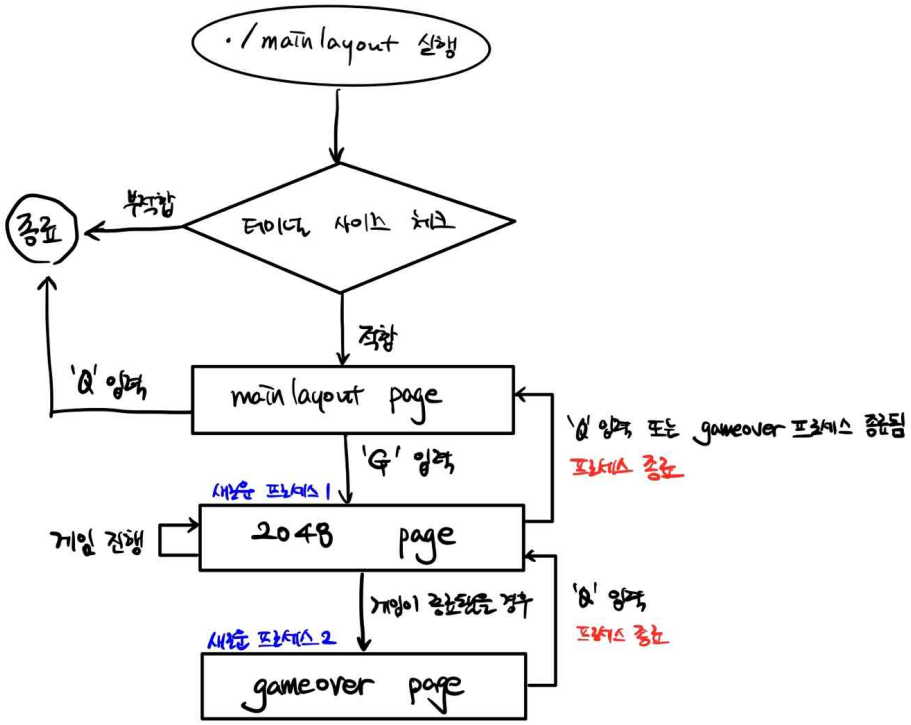
void initializeCheck(void); : 전역 번수 check 배열의 모든 값을 0으로 초기화하는 함수이다.

void defaultPage(void); : 2048 게임을 실행하는 메인 화면을 설정하는 함수이다.

3) gameover.c

void setGameoverLayout(void) : 게임 종료 화면을 설정하는 함수이다.

- 2048 게임 페이지 구조(간략한 순서도)

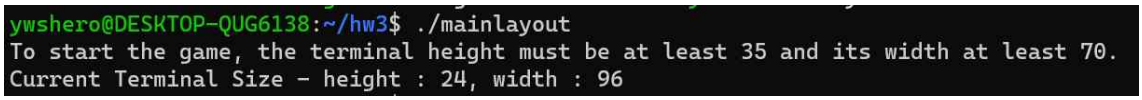


- 프로그램의 실행 과정 **(게임 이용 메뉴얼)**

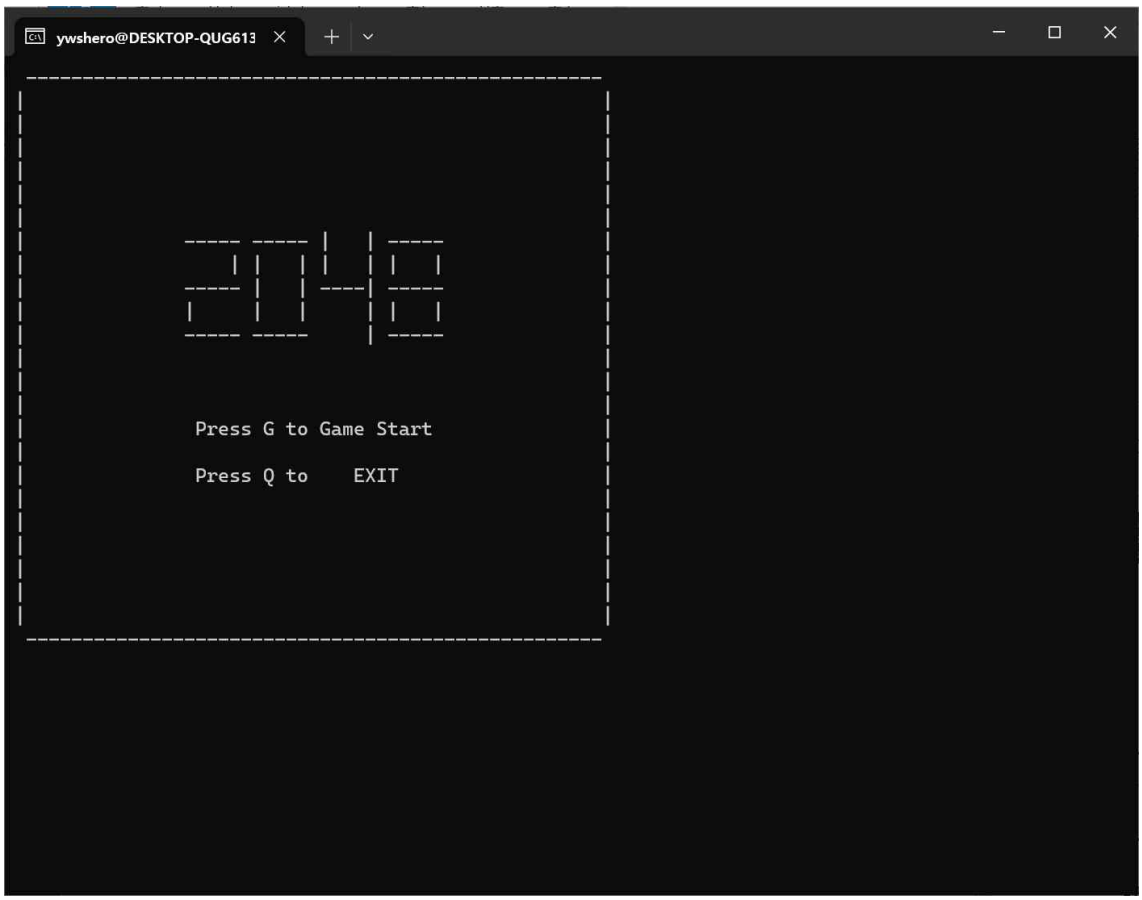
****모든 페이지에서 Ctrl + C 와 Ctrl + Backspace 신호는 무시됩니다.****

(1) 실행 파일 mainlayout을 실행합니다. (command : ./mainlayout)

- 만약, 현재 터미널 창의 크기가 기준보다 작은 경우, 실행이 되지 않고 터미널 크기를 늘리라는 메시지가 출력됩니다.



- 터미널 창의 크기가 적당할 경우 실행됩니다.

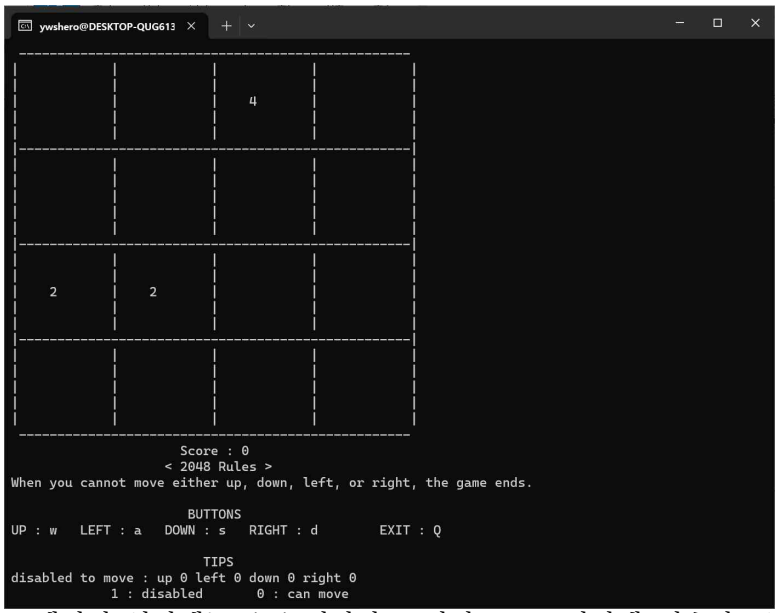


(2) 메인 페이지입니다.

Q를 입력하면 종료되고, G를 입력하면 fork() 함수로 자식 프로세스를 생성하고, 부모 프로세스는 자식 프로세스의 종료를 기다립니다. (입력을 받는 while문 안에서 다른 입력은 무시합니다.)

만들어진 자식 프로세스는 execv 함수를 이용하여 2048 페이지를 실행합니다.

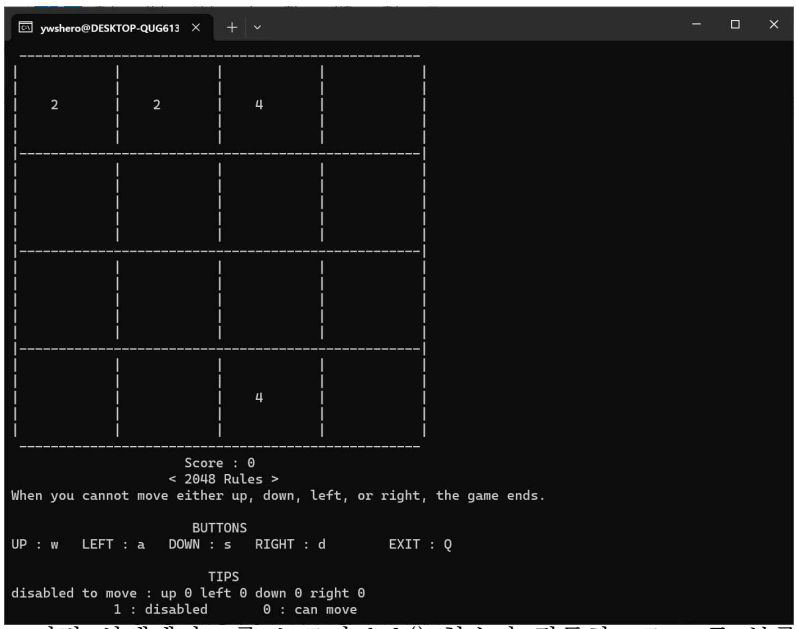
(3) 2048 페이지 초기 화면입니다.



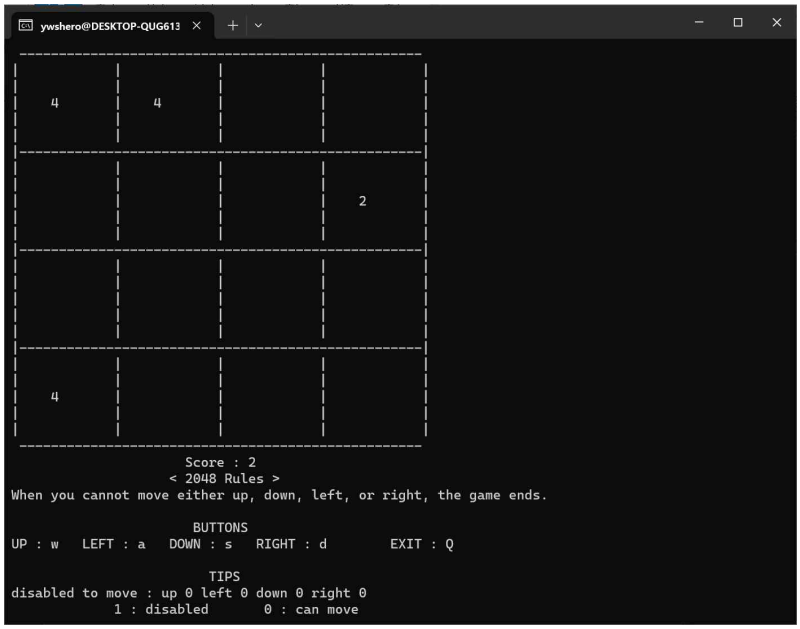
- 페이지 상단에는 4x4 격자가 표시되고, 그 하단엔 점수와 규칙, 버튼 설명, 팁이 적혀있습니다.
- w, a, s, d를 입력할 경우, w는 up(), a는 left(), s는 down(), d는 right() 함수를 실행시킵니다.

  예를 들어 w를 입력했다고 가정해봅시다. up() 함수가 작동할 것이고, 원소들이 상단으로 밀착될 것입니다. 하지만 합쳐진 블록은 없으므로 score는 오르지 않을 것입니다.

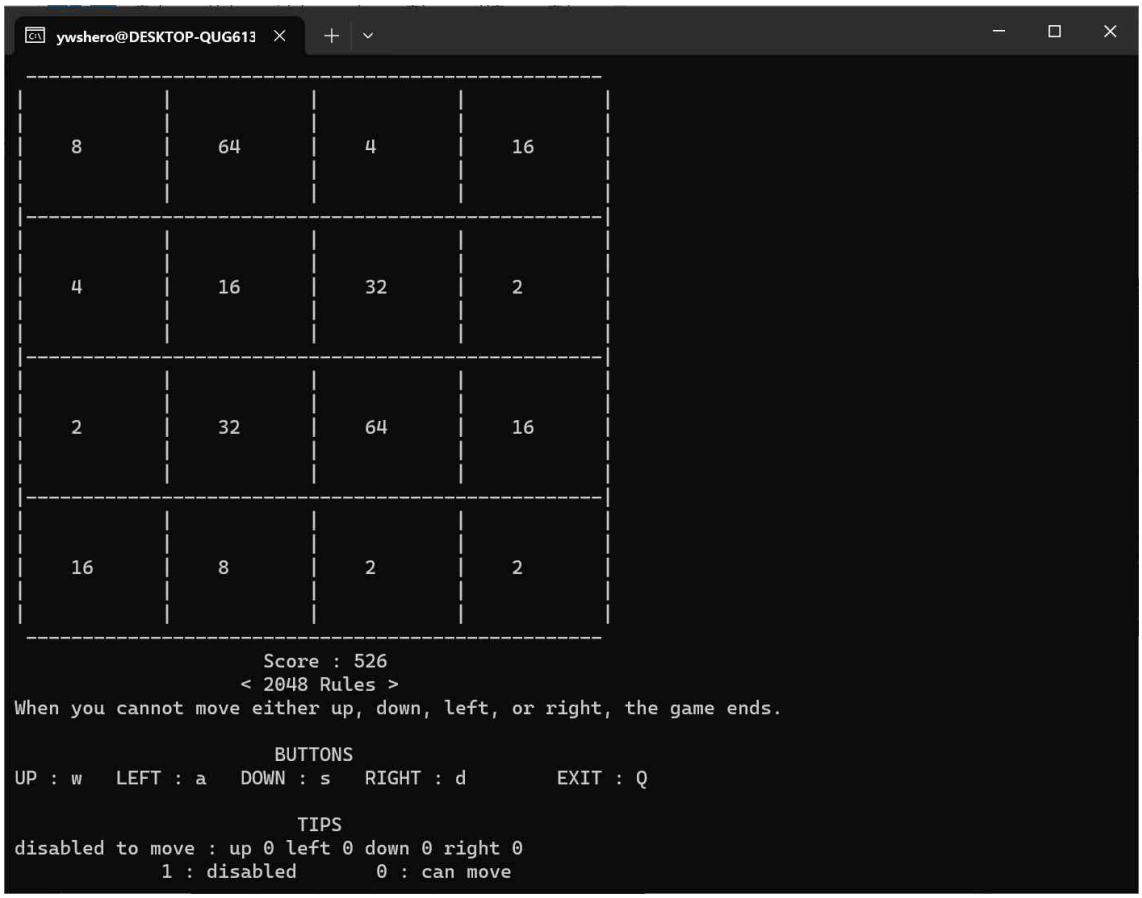
  움직임이 발생했으므로 generate 함수가 작동합니다. 따라서 w를 입력한 후에는 화면이 다음과 같이 변하게 됩니다.



- 이전 상태에서 a를 누르면 left() 함수가 작동하므로 모든 블록이 좌측으로 밀착되고, 같은 블록인 2 2 는 합쳐지게 됩니다. 따라서 score는 2 증가합니다.

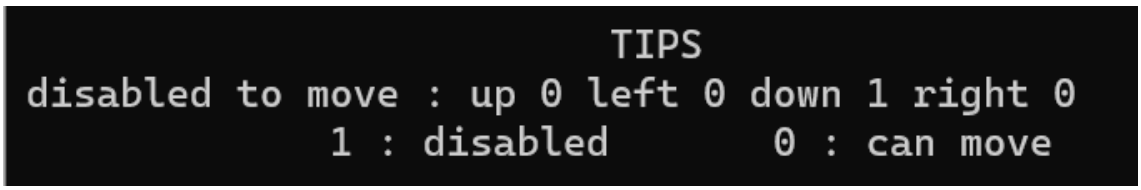


- 다음 사진은 w a s d 버튼을 여러번 눌러 게임을 진행한 상태입니다.

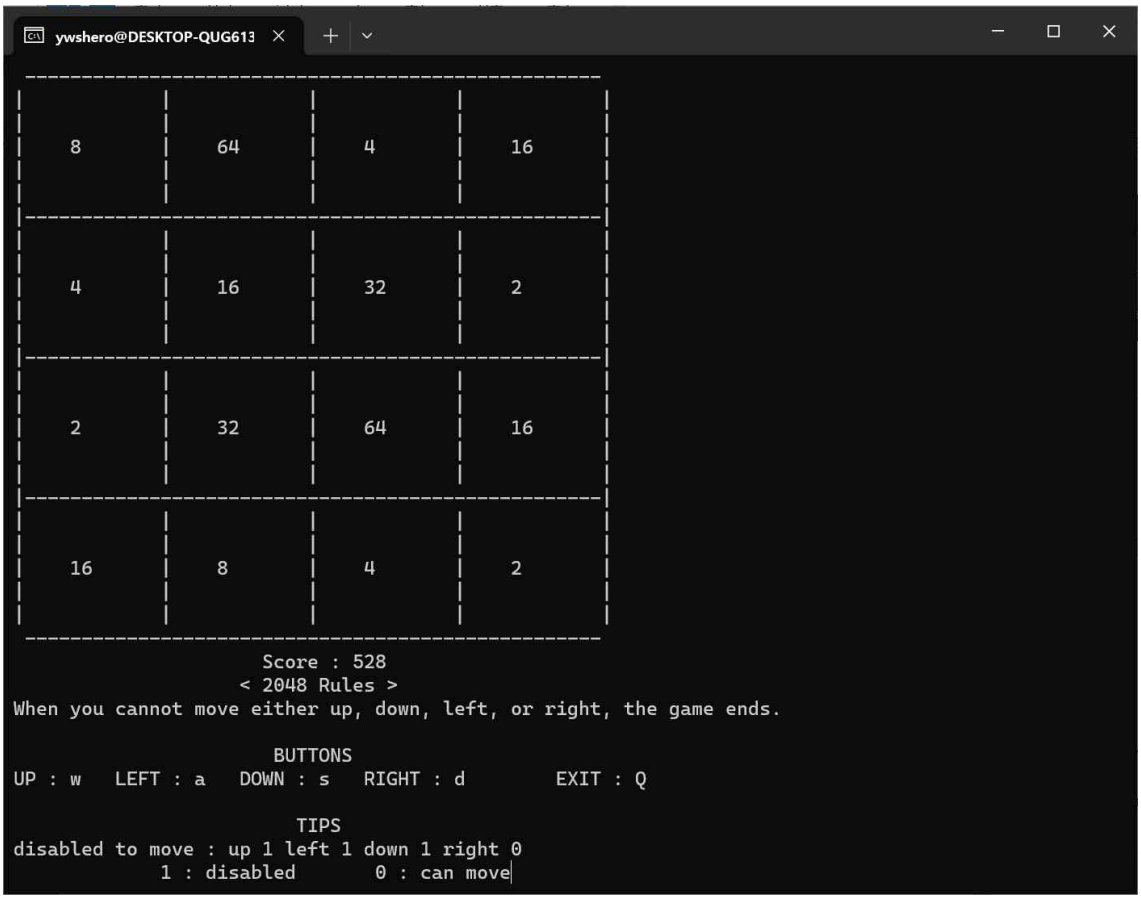


- 이 상태에서, s를 눌러 down() 함수를 실행해도 격자의 내용은 변하지 않습니다. 격자의 내용이 변하지 않았기에 generate 함수는 실행되지 않을 것이며, 아래로 움직일 수 없다는

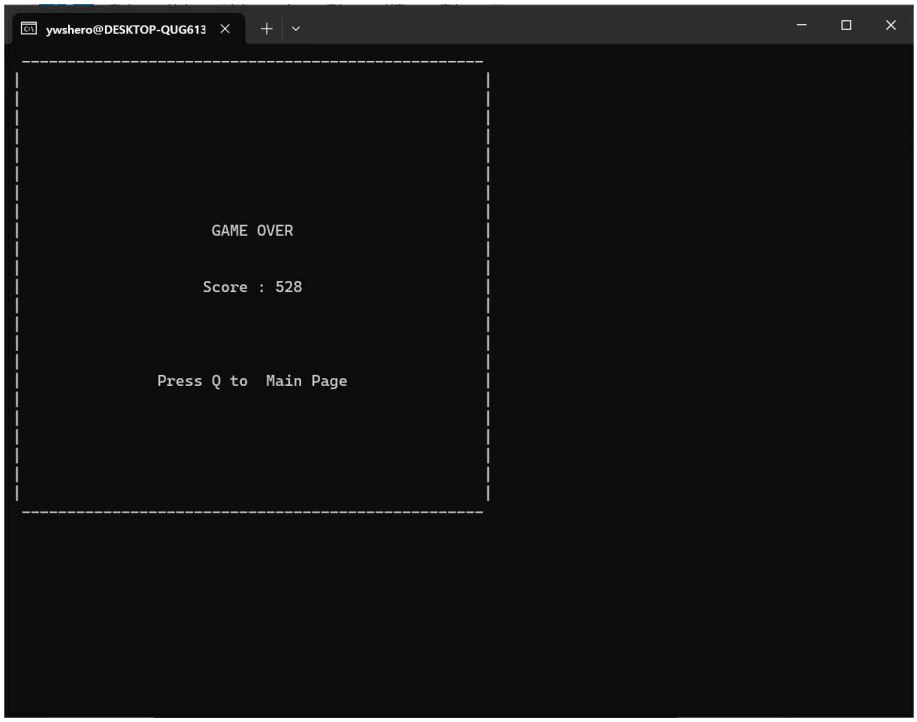
뜻으로 TIPS 부분의 down값도 1로 변하게 됩니다.



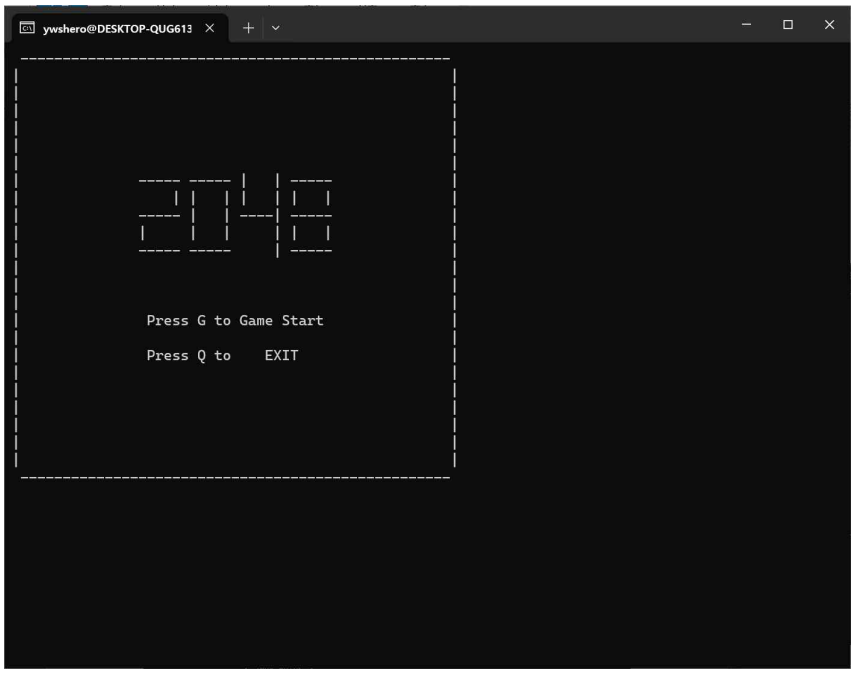
- 게임을 더 진행해서 다음과 같은 상황이 됐습니다.



- TIPS를 보면, up left down 방향으로 모두 움직일 수 없다는 것을 알 수 있습니다. 움직일 수 있는 방향은 right만 남았습니다만, 오른쪽으로 움직여도 판의 내용은 바뀌지 않을 것입니다. 따라서, d를 입력하면, 오른쪽으로도 이동할 수 없다는 뜻으로 TIPS의 모든 값들이 1로 바뀌게 됩니다.
  TIPS에 적혀있는, check 배열의 모든 원소가 1일 경우 게임이 종료되었다고 판단합니다. 따라서 이 상황에서 d를 입력하게 되면 fork() 함수로 자식 프로세스를 생성하고, 부모 프로세스는 자식 프로세스의 종료를 기다립니다. 자식 프로세스는 execv 함수로 gameover 페이지를 실행합니다.



- 게임 종료 페이지에서는 이때까지 게임을 진행하면서 쌓아왔던 점수를 표시해줍니다. Q를 입력하면 gameover 프로세스를 종료하게 됩니다.
  gameover 프로세스가 종료된다면, 이 프로세스의 부모 프로세스인 2048 프로세스로 돌아가게 되지만, 이미 게임이 종료되었다고 이전에 판단했으므로 2048 프로세스 또한 종료되게 됩니다.
  따라서, 2048 프로세스의 종료를 기다리고 있던 부모 프로세스 - mainlayout 페이지로 돌아가게 됩니다.



- 게임을 한 번 더 진행하고자 하면 ‘G’를 입력해 진행하면 되고 게임을 종료하고자 하면 ‘Q’를 입력하면 종료됩니다.

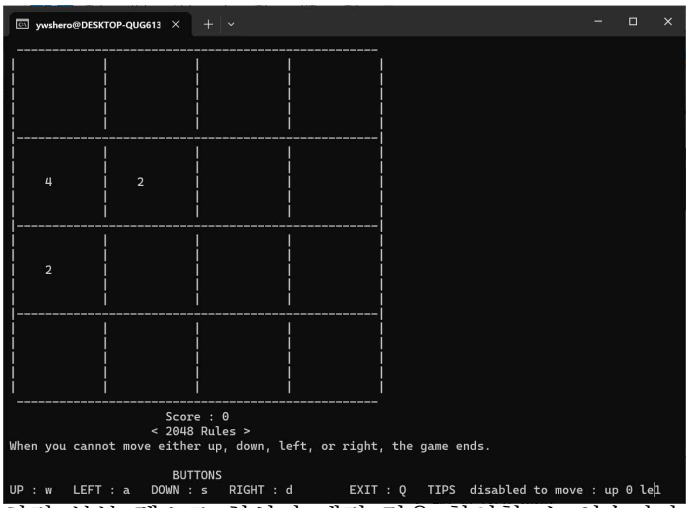
```
ywshero@DESKTOP-QUG6138:~/hw3$ ./mainlayout
To start the game, the terminal height must be at least 35 and its width at least 70.
Current Terminal Size - height : 32, width : 96
ywshero@DESKTOP-QUG6138:~/hw3$ ./mainlayout
ywshero@DESKTOP-QUG6138:~/hw3$
```

일련의 프로그램 실행 과정을, 수행 결과와 함께 서술하였습니다.

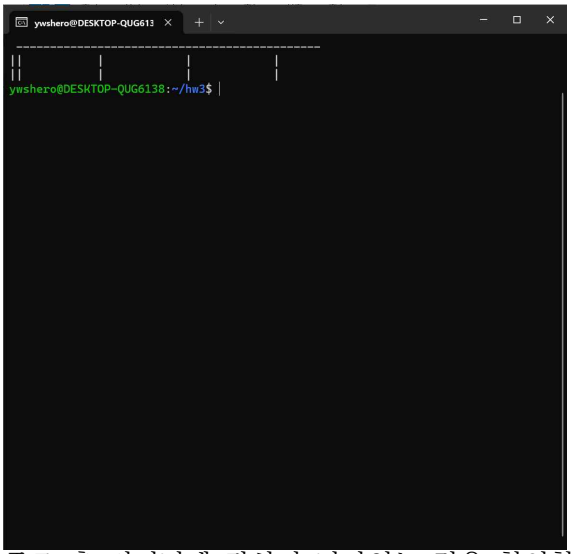
3. 이슈 사항 및 해결

<이슈 1>

터미널의 크기가 충분하지 않았을 때, 표시되는 내용의 형식이 깨지고, 종료했을 때 터미널에 잔상이 남는 오류가 발생했습니다.



하단 부분 텍스트 형식이 깨진 것을 확인할 수 있습니다.

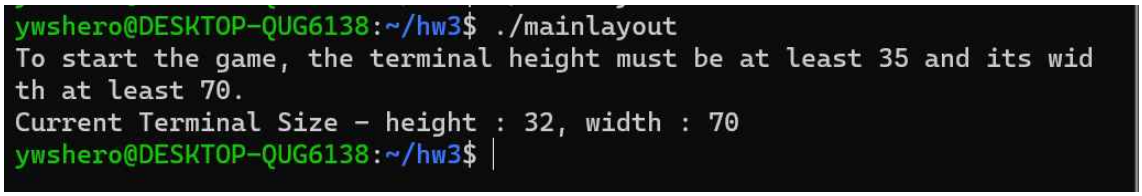


종료 후 터미널에 잔상이 남아있는 것을 확인할 수 있습니다.

<해결 1>

터미널의 크기가 기준보다 작을 경우, 게임이 실행되지 않고 터미널 크기를 수정하라는 메시지를 출력하도록 설계했습니다.

터미널의 현재 크기를 받아오는 curses.h의 getmaxyx(stdscr, HEIGHT, WIDTH) 함수를 이용하여, 높이 35, 너비 70이 되지 않을 경우 오류 메시지를 출력했습니다.



<이슈 2>

2048 게임의 움직임 함수 구현을 하는 과정에서 수많은 알고리즘 오류들이 발생했습니다.

예를 들어, [움직인 방향으로 수가 밀착되지 않고 한 칸씩 움직임], [한 번에 모든 수가 합쳐짐], [움직인 방향의 끝에 있는 숫자가 사라짐] 과 같은 오류들이 발생했습니다.

<해결 2>

많은 시행착오를 겪으며, 움직임 알고리즘을 여러 차례 수정함으로써 정상적으로 움직이지 않는 문제를 해결했습니다.

4. 메뉴얼 (사용 방법)

- make 명령어를 이용해 실행파일 생성하기
  make 또는 make all 명령어를 이용하여 모든 실행파일을 생성할 수 있습니다.

(Makefile 내용)

```
all : mainlayout 2048 gameover

mainlayout : mainlayout.c
        gcc -o mainlayout mainlayout.c -lcurses
2048 : 2048.c
        gcc -o 2048 2048.c -lcurses
gameover : gameover.c
        gcc -o gameover gameover.c -lcurses
clean :
        rm -f mainlayout 2048 gameover
```

- 실행파일이 모두 생성되었다면, mainlayout 실행 파일으로 게임을 실행합니다.
  (상세한 게임 설명은 목차 2에 자세히 서술하였습니다.)

- 매뉴얼

1. 게임 시작 : ./mainlayout 명령어로 게임 실행
2. 메인 화면 : G 키를 눌러 게임 시작 / Q 키를 눌러 게임 종료
3. 2048 게임 화면 :

w - up(), a - left(), s - down(), d - right() 방향으로 모든 블록을 원하는 방향으로 움직일 수 있음.

4. 종료 조건 : 격자에 있는 블록을 더 이상 어느 방향으로도 움직일 수 없을 때, 즉 모든 격자가 블록으로 차 있고 어떤 블록도 합쳐질 수 없을 때 게임 종료. 게임 종료 페이지 자동 실행
5. 게임 종료 : 게임 종료 페이지에서 Q 키를 눌러 메인 페이지로 돌아감.

이상으로 curses 라이브러리를 이용한 2048 퍼즐 게임 구현에 대한 보고서를 마치겠습니다.
긴 글 읽어주셔서 감사합니다.