

프로그래밍 연습

# Project #1-1

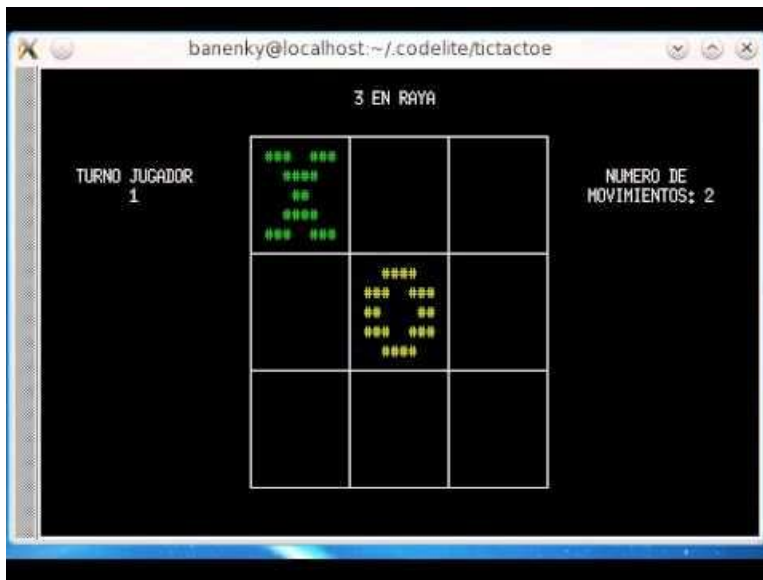
“INT\_MAX” Game Programming  
with NCURSES Library

## Exercise Submission

- 기존에 출제된 모든 Exercise 들은 12월 9일 (화) 까지 제출 완료한다.
- 코드를 그대로 보고 따라서 치는 종류의 Exercise는 Presentation Error 혹은 0 점이어도 제출만 했으면 무방하다. 단, 엉뚱한 파일, 혹은 빈 파일을 제출한 경우 최종 성적에 큰 감점 요소가 된다.
- 점수가 매겨지는 Exercise의 경우는 무조건 100점이 나오지 않으면 제출로 인정하지 않는다.
- Exercise는 Assignment와 달리 Test case 관련 질문이나 문제의 해법과 관련된 질문이 가능하고, 더욱 자세히 답변을 받을 수 있다. 막히는 부분이 있을 경우 반드시 적극적인 질문을 통해 해결하도록 한다.
- 12월 9일까지 Exercise를 모두 제출하지 않을 경우 F를 부여하거나, 최종 성적에 큰 감점이 가해지므로 최대한 빠른 제출을 권장한다.

## NCURSES library

- CUI (Command-line User Interface) 환경에서 GUI (Graphical User Interface) 와 같은 응용 소프트웨어를 개발하는 library이다.
- <http://ko.wikipedia.org/wiki/Ncurses>
- <http://en.wikipedia.org/wiki/Ncurses>



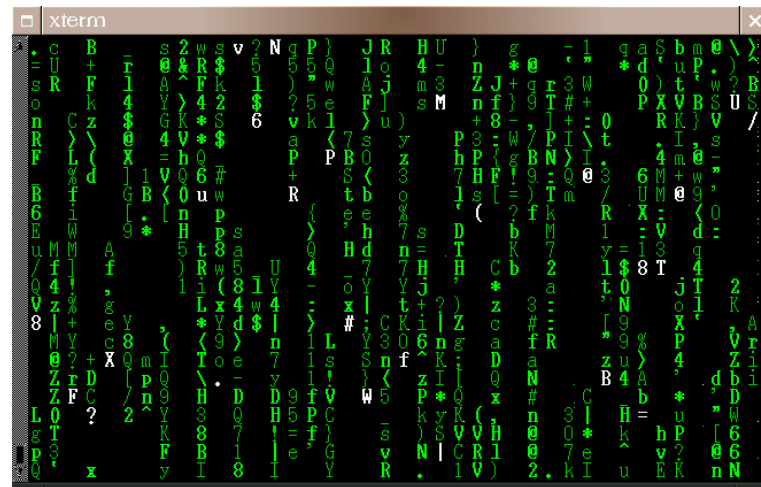
## NCURSES library

- Install NCURSES library
  - 실습서버에는 이미 설치되어 있으므로 따로 설치 과정을 거칠 필요가 없다.
  - 개인서버를 사용하는 경우, 터미널에서 아래와 같이 입력하여 설치한다.

```
sudo apt-get install build-essential
```

```
sudo apt-get install libncurses5-dev
```

```
sudo apt-get install libncursesw5-dev
```

A screenshot of a terminal window titled 'xterm'. The terminal displays a large, complex ASCII art graphic. The graphic is composed of various characters, including letters, numbers, and symbols, arranged in a way that creates a sense of depth and structure. The characters are primarily white and green on a black background. The graphic appears to be a stylized representation of a building or a complex structure, with many lines and curves. The terminal window has a standard Linux-style title bar and a scrollback buffer.

## Hello World !!!

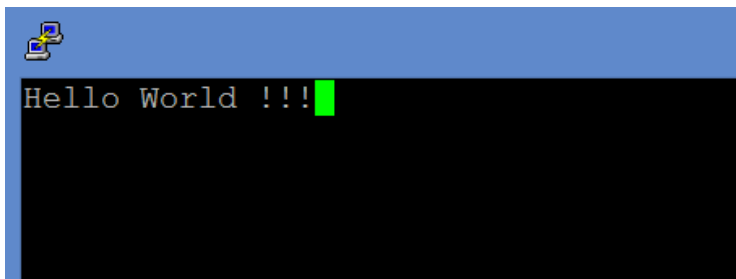
- 다음과 같은 코드를 실행한다.
  - 컴파일 시, 반드시 **-lncurses** 를 붙여준다. 이는 math.h 를 쓸 때처럼 컴파일 하는 과정에서 NCURSES library를 link 시켜 주겠다는 의미이다.

```
1 #include <ncurses.h>
2
3 int main()
4 {
5     initscr();
6     printf("Hello World !!!");
7     refresh();
8     getch();
9     endwin();
10
11     return 0;
12 }
13
```

```
dhshin@ubuntu:~/ncurses$ gcc hello.c -o hello -lncurses
dhshin@ubuntu:~/ncurses$
dhshin@ubuntu:~/ncurses$ ./hello
```

# Hello World !!!

## • 실행 결과



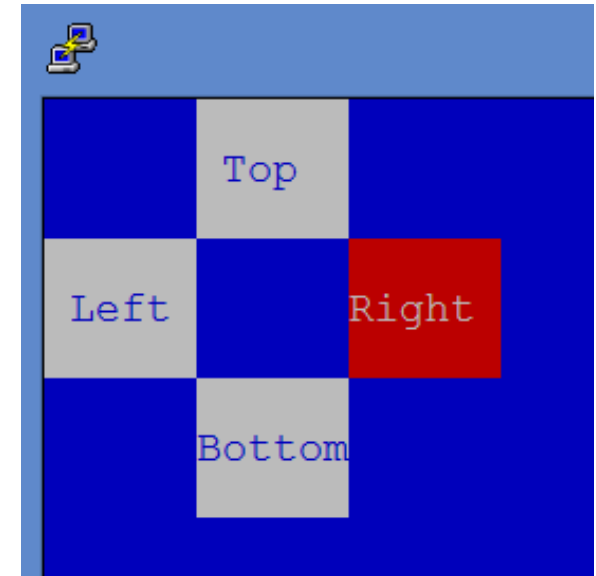
```
1 #include <ncurses.h>
2
3 int main()
4 {
5     initscr();
6     printw("Hello World !!!");
7     refresh();
8     getch();
9     endwin();
10
11     return 0;
12 }
13
```

- `initscr()` : `curses`모드로 터미널을 초기화한다. 프로그램의 시작부분에 항상 써줘야 한다.
- `printw()` : `stdscr` 이라는 `curses`의 가장 기본이 되는 윈도우에 글자를 출력한다. 사용법은 `printf` 함수와 같다.
- `refresh()` : `stdscr` 의 변경사항을 실제 `termina`에 띄워준다. `Ncurses`는 화면이 변경될 때마다 매번 바로 출력해주지 않고 그 정보를 모았다가 한번에 `refresh` 와 같은 함수를 통해 띄워주는 방식이다.
- `endwin()` : 프로그램 종료 직전 `curses` 모드를 끝내주는 역할을 한다. 이 코드를 생략하면 프로그램 종료 후 터미널이 이상하게 작동할 수 있다.

## WINDOW Example

- Moodle 에 게시된 window.c 예제를 다운받아서 직접 컴파일하여 실행시켜보고, 코드와 비교해보며 각 함수들의 역할 및 WINDOW 구조체의 개념을 익힌다.
- 코드를 마음대로 수정해보고, 그에 따른 실행 결과의 차이를 보는 것도 좋은 방법이다.
- 꼭 ncurses.h 만 include 해야 하는 것은 아니다. 가령 malloc이 필요하다면 stdlib.h 를 추가적으로 include 해도 전혀 상관없다.

```
1 #include <ncurses.h>
2
3 int main()
4 {
5     int key;
6     int window_line = 3;
7     int window_col = 6;
8     int window_selected = -1;
9     WINDOW *left, *top, *right, *bottom;
10
11     initscr();
12     noecho();
13     keypad(stdscr, TRUE);
14     curs_set(0);
15
16     start_color();
```



- 방향키를 누름에 따라 그에 해당하는 사각형이 빨간 색으로 강조된다.
- 실행종료는 'q' 를 입력한다.

## Extended Characters

- Putty 의 기본 설정으로는

아래 표의 NCURSES extended characters 를 제대로 출력할 수 없다.

NCURSES EXTENDED CHARACTERS		
Upper left corner	[	ACS_ULCORNER
Lower left corner	[	ACS_LLCORNER
Upper right corner	]	ACS_URCORNER
Lower right corner	]	ACS_LRCORNER
Tee pointing right	├	ACS_LTEE
Tee pointing left	┤	ACS_RTEE
Tee pointing up	┬	ACS_BTEE
Tee pointing down	┴	ACS_TTEE
Horizontal line	—	ACS_HLINE
Vertical line		ACS_VLINE
Large Plus or cross over	⊕	ACS_PLUS
Scan Line 1	—	ACS_S1
Scan Line 3	—	ACS_S3
Scan Line 7	—	ACS_S7
Scan Line 9	—	ACS_S9
Diamond	◆	ACS_DIAMOND
Checker board (stipple)	⌘	ACS_CKBOARD
Degree Symbol	°	ACS_DEGREE
Plus/Minus Symbol	±	ACS_PLMINUS
Bullet	·	ACS_BULLET
Arrow Pointing Left	<	ACS_LARROW
Arrow Pointing Right	>	ACS_RARROW
Arrow Pointing Down	v	ACS_DARROW
Arrow Pointing Up	^	ACS_UARROW
Board of squares	#	ACS_BOARD
Lantern Symbol	☞	ACS_LANTERN
Solid Square Block	#	ACS_BLOCK
Less/Equal sign	≤	ACS_LEQUAL
Greater/Equal sign	≥	ACS_GEQUAL
Pi	π	ACS_PI
Not equal	≠	ACS_NEQUAL
UK pound sign	£	ACS_STERLING

- 참고

(<http://melvilletheatre.com/articles/ncurses-extended-characters/index.html>)



## Extended Characters

- 제대로 출력하려면 서버에 접속해서 다음을 입력한다.

```
prg2@prg2:~$  
prg2@prg2:~$ tmux
```



```
prg2@prg2:~$ export TERM=screen-256color  
prg2@prg2:~$
```

- 단, 이와 같이 입력한 경우, shell의 동작이나 vim 화면이 깨지는 등의 현상이 일어날 수 있으므로, 프로그램을 실행하는 용으로 putty 창을 하나 더 켜고 위와 같이 입력한 후, 결과 확인 용으로만 쓰고, 작업은 기본 putty에서 하는 것이 좋다.

## Tips for learning NCURSES

- 아래의 웹사이트에 NCURSES 에 대한 설명 및 예제가 자세히 적혀있다.

<https://wiki.kldp.org/wiki.php/NCURSES-Programming-HOWTO>

: 어떤 모르는 라이브러리를 처음 익힐 때에는 위와 같은 설명이 적힌 페이지를 처음부터 끝까지 정독한다는 느낌보다는 사전을 찾아보는 것처럼 필요한 부분이나 필요한 함수를 위주로 읽고, 주로 직접 코딩을 통해 익히는 것이 제일 효과적이다. 물론 정독해서 나쁠 것은 없다. 하지만 위의 페이지만 전부 꼼꼼히 읽는다고 프로젝트를 완수할 수 있는 것은 아니니 프로젝트 수행 중 그때 그때 필요한 함수를 찾아보는 것을 위주로 최대한 여러 자료를 참고하여 학습하도록 한다. 라이브러리의 모든 내용을 다 숙지하고 나서 프로그램 작성을 시작하려는 것은 비효율적이며 시간소모가 크다.

## Tips for learning NCURSES

- 예제 코드에 있는 함수들만으로는 프로젝트의 요구사항을 모두 충족시키기 힘들다. 필요한 함수를 스스로 찾아본다.
- 모르는 함수는 Google에 검색하여 그 기능과 사용법을 찾아보거나, Linux man 페이지에서 검색한다. (<http://linux.die.net/man/>)
- 가령 WINDOW에 특정 색을 칠하는 작업을 하고 싶은데, 방법을 모를 경우, "ncurses window color" 등과 같이 Google에 검색해 볼 수 있다. 주로 stackoverflow.com 이라는 웹사이트에 질문과 답변이 많기 때문에 많이 검색되며 유용하게 참조해볼 수 있다.

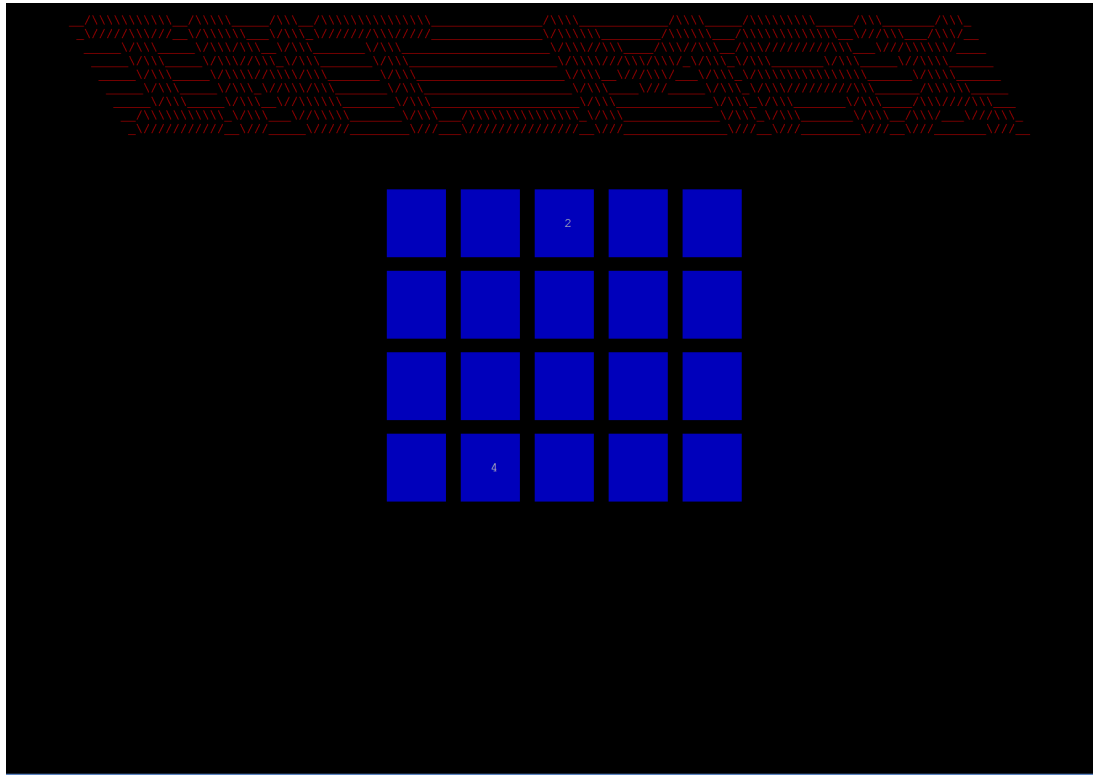
## Keep in mind...

- 본 프로젝트는 C언어의 기본 문법을 다 숙지한 상태에서 **필요한 library를 스스로 익히고 쓸 수 있는 능력**을 배양하는 것이 목표이다.
- 앞으로의 컴퓨터공학부 수업이나 실무에서 특정 library 를 세세하게 가르쳐주는 경우는 거의 없다. 그러나 기본 문법을 아는 상태라면 스스로 찾아보고 배울 수 있는 자료는 많다.
- 따라서 NCURSES library의 세세한 내용에 대해서는 최대한 웹 상의 예제들 및 documentation을 참조하여 스스로 공부하고 직접 코드를 작성해보며 익히도록 한다.
- 웹에 있는 코드 등은 필요하면 그대로 가져다 써도 무방하다.
- **단, 기존의 NCURSES를 이용한 2048게임의 코드를 가져와 쓰는 것은 제한한다.**  
**(웹 상에 공개된 코드를 포함하여, 엄격한 카피 체크를 진행할 예정)**
- ETL 질문/답변 게시판에 NCURSES 라이브러리와 관련된 질문은 받지 않음.  
(조교들도 NCURSES 라이브러리를 정통하고 있는 것은 아님.)

# Project #1-1

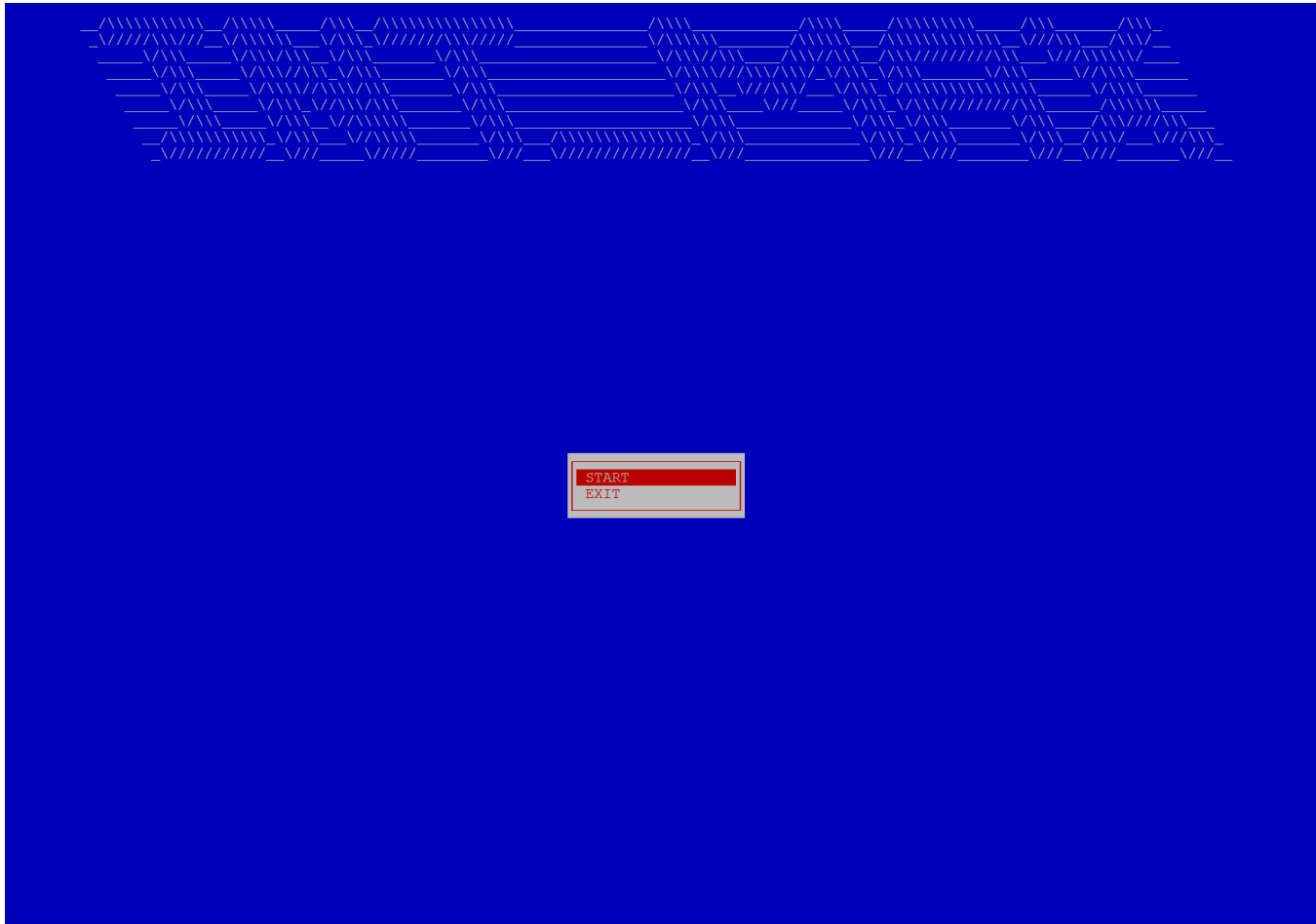
## Project Description

- NCURSES를 이용하여 2048 게임의 변형인 "INT\_MAX" 게임을 만든다.
- 기존 2048 게임과 대부분 동일하지만 GRID를 4 x 5로 생성한다.



## Project Description

- 실행 초기 화면 – Main menu



## Project Description

- 실행 초기 화면 – Main menu

1. 중앙 정렬된 메뉴를 출력한다.

- 아주 정밀하게 중앙을 맞추는 필요는 없으나, 대강은 맞아야 하고, 어떤 크기의 터미널로 켜든 모두 중앙에 있도록 작성한다.

2. 메뉴의 항목은 START, EXIT 두 항목으로 한다.

- 현재 어떤 메뉴가 선택되어 있는지 표시 (highlighting) 해 줄 수 있어야 한다.
- Highlighting 방식은 자율적으로 정한다. 단, 알아볼 수 있어야 한다.
- START가 선택된 상황에서 Enter를 입력 시 화면이 바뀌며 게임 화면으로 넘어간다.
- EXIT이 선택된 상황에서 Enter를 입력 시 프로그램이 종료된다.
- 메뉴는 항상 START 혹은 EXIT, 둘 중 하나가 선택되어 있는(highlight 되어 있는) 상황이어야 한다.
- Up, Down 방향키를 이용해 START 와 EXIT의 선택을 전환한다.

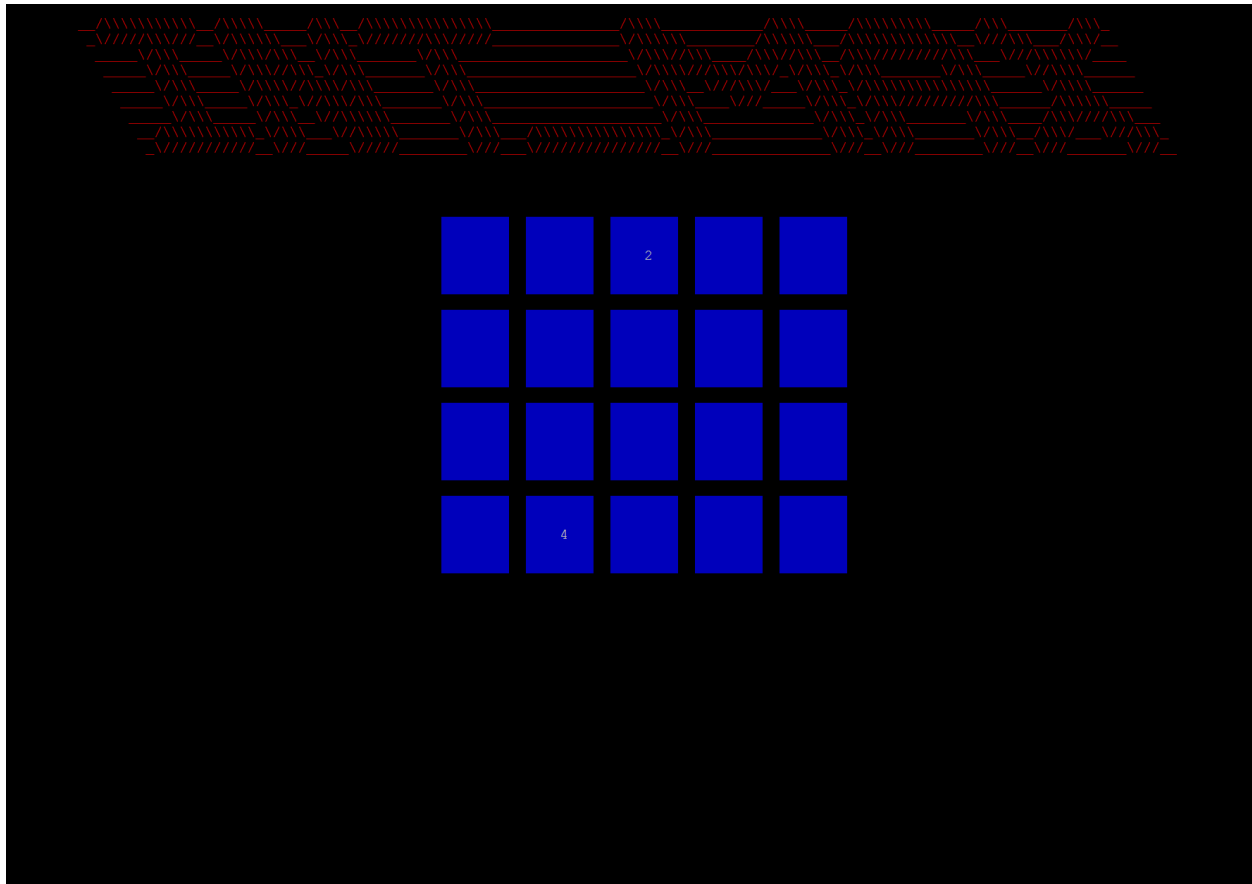


## Project Description

- 실행 초기 화면 – Main menu
- 3. 터미널의 Size를 드래그 등으로 바꿨을 때, 그에 맞춰 프로그램이 resize 될 필요는 없다.
  - 즉, 처음 프로그램을 켤 때의 터미널 size가 유지된다고 가정하고 코드를 작성해도 좋다.
- 4. 화면 상단에 "INT\_MAX"라는 글자의 ASCII Art 출력한다. 아래 웹 페이지를 이용한다.
  - [http://patorjk.com/software/taag/#p=display&f=Small%20Slant&t=INT\\_MAX](http://patorjk.com/software/taag/#p=display&f=Small%20Slant&t=INT_MAX)
  - 디자인은 마음에 드는 것을 선택하고, 나오는 결과를 코드 내에 복사, 붙여 넣기 하고 알맞은 출력 함수를 이용하여 출력한다.
- 5. 글자 색, 윈도우 배경 색 등등 특별히 제한하지 않은 것들은 모두 자유롭게 선택하여 구현한다.

## Project Description

- 게임 화면



## Project Description

### • 게임 화면

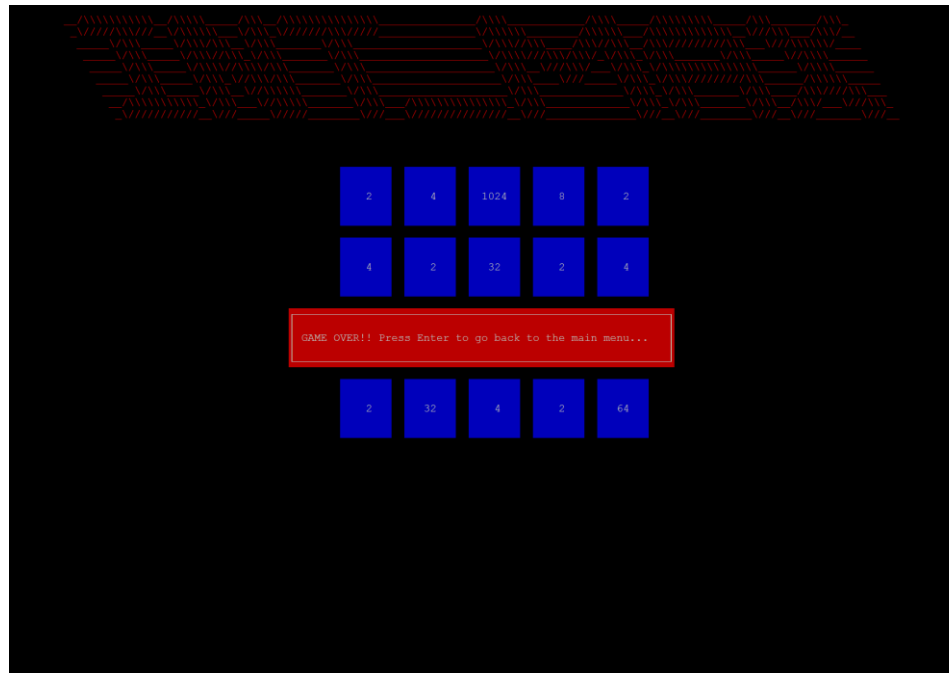
1. 실행 초기 화면과 마찬가지로 상단에 ASCII Art를 출력한다.
  - 실행 초기 화면의 ASCII Art와 디자인 및 크기가 달라져도 상관없다.
2. 게임 판은 **행 4, 열 5** 로 만든다.
3. 게임 판은 중앙 정렬되어야 한다.
  - 역시 정확하게 중앙일 필요는 없지만, 이상하지 않을 정도로는 맞춰주어야 한다.
4. 2048과 게임과 마찬가지로 방향키 입력을 받는 방식으로 게임이 진행되어야 하며, 오류 없이 진행되어야 한다.

## Project Description

### • 게임 화면

5. 더 이상 움직일 것이 없어 Game over가 될 경우, Game over 임을 알리는 조그마한 창이 뜨도록 한다.

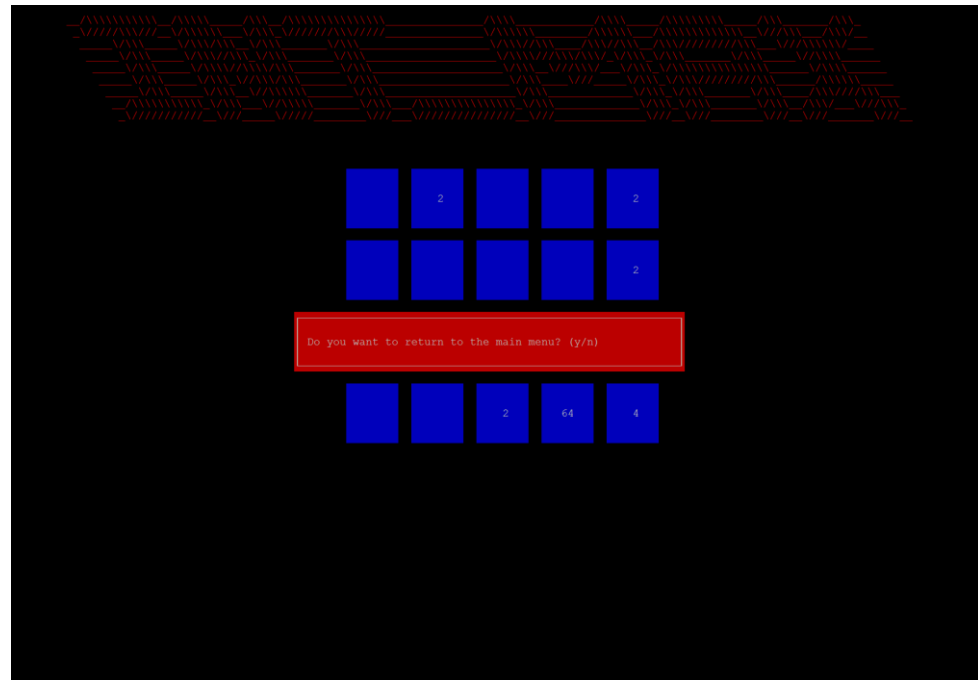
- Game over 창이 뜬 상태에서 Enter를 누르면 다시 실행 초기 화면으로 돌아간다. (그 외의 입력은 무시)
- 창에 뜨는 안내 메시지 및 색상, 디자인은 자유이지만, Game over 창임은 명확히 알 수 있도록 작성한다.



## Project Description

### • 게임 화면

6. 게임 도중 'q'를 누르면 실행 초기 화면으로 돌아갈 것인지 묻는 창이 뜨도록 한다.
  - 해당 창이 뜬 상태에서 'y'를 누르면 실행 초기 화면으로, 'n'을 누르면 창을 없애고 게임을 재개할 수 있도록 한다. (그 외의 입력은 무시)
  - 창에 뜨는 안내 메시지 및 색상, 디자인은 자유이지만, 실행 초기 화면으로 돌아갈 것인지 묻는 창임은 명확히 알 수 있도록 작성한다.



## Project Description

### • 게임 화면

7. 게임 판의 색깔, 테두리, 디자인 등은 자유이지만 너무 작게 구현하지 않도록 한다.
  - 숫자가 게임의 한 칸을 벗어나는 등의 문제 또한 감점 요인이다.
8. 2048과 똑같이 동작하도록 작성하되, 다음을 유의하도록 한다.
  - 매 번 유저가 방향키를 누를 때마다 임의의 빈 칸 하나에 숫자 2가 채워진다.
  - 단, 유저가 방향키를 눌렀을 때 아무 일도 일어나지 않는 경우에는 새로운 숫자가 추가되지 않는다.
  - 게임의 시작 시에는 4 x 5 칸 중 임의의 2칸에 2 혹은 4의 숫자가 있도록 한다. 이 때, 2는 90%, 4는 10%의 확률로 등장하도록 작성한다.
  - 기존 2048과 달리 더 이상 움직일 수 없는 Game over의 상황이 올 때 까지 게임을 멈추지 않고 계속 할 수 있도록 한다.
  - 위의 게임의 룰을 정확히 구현하지 못하면 감점 대상이다.
  - 프로그램 실행 도중 어떠한 형태의 에러라도 파악되면 감점 대상이다.

## Project Description

- 난수(Random Number) 생성

- 프로젝트를 수행을 위해서는 임의의 수를 생성하는 과정이 필요하다.
- 프로그램 작성 시, 임의의 수, 즉 난수 생성은 rand 함수를 이용한다.
  - <http://stackoverflow.com/questions/822323/how-to-generate-a-random-number-in-c>
  - <http://www.jynote.net/64>

- 채점

- 오류가 발견될 때마다 만점에서 감점을 하는 방식으로 진행된다.

## Submission

- 작성한 c 소스파일과 makefile을 함께 [prgxx].zip 파일로 압축하여 moodle의 “Project 1-1” 항목에 upload한다. (파일명 포맷이 맞지 않을 경우 감점됨.)
  - ex) prg01.zip , prg1.zip
- 필요한 경우, 파일을 여러 개로 분할하여 프로그램을 작성 후 제출해도 된다. (헤더 파일을 만들었을 경우, 함께 제출)
- 채점은 다음과 같이 진행된다.

압축을 해제한 뒤, 해당 폴더에서

**make clean**

**make all**

**make run**

을 순서대로 입력하여 프로그램이 실행되면 채점을 시작.

만일 위와 같이 진행하여 프로그램이 제대로 실행되지 않으면

더 이상 채점을 진행하지 않고 감점과 함께 재제출을 요구하는 공지를 할 예정.



## Submission

- 제출 기한은 **12월 5일 (금) 23시 59분**까지이다.
- Delay 정책은 다음과 같다.
  - 12월 8일 (월) 23시 59분까지 제출 - 100점 만점에 75점부터 시작하여 오류 발견 시 감점.
  - 12월 11일 (목) 23시 59분까지 제출 - 50점부터 시작하여 오류 발견 시 감점.
  - 12월 14일 (일) 23시 59분까지 제출 - 25점부터 시작하여 오류 발견 시 감점.
  - 그 이후 - 0점