

# PP Final Project

2<sup>nd</sup> Week

# Notice

Regarding previous week submissions

# Notice

1. When drawing boxes please use **addch** or **mvaddch**, **not printw**.
2. Using ACS\_VLINE and ACS\_HLINE draws the boxes more smoothly than when using '|' and '—'. (They are different sets of characters.)
3. **When exiting the program, `endwin()` needs to be called!!**  
(If you **return**; prematurely in **main()** without calling **endwin()**, then console gets buggy. This makes it difficult for us to grade.)

# DEMO

For this week

# No skeleton

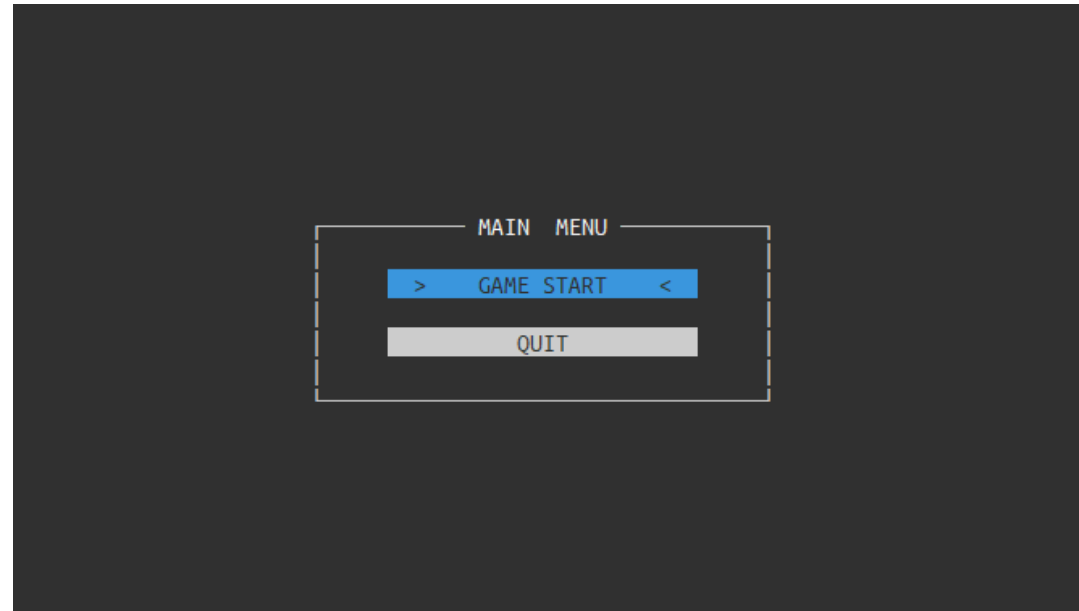
- No extra skeleton will be provided.
- Build on from your previous code.
- However, some tips for implementation will be provided!

# Tasks for 2<sup>nd</sup> week

1. Implement a 'Main Menu', that has 'GAME START' and 'QUIT' options.
2. In the game screen, display the current highlighted card. It should be possible to move the highlight around with the arrow keys.

# Specific requirements

## 1. Main Menu



# Specific requirements

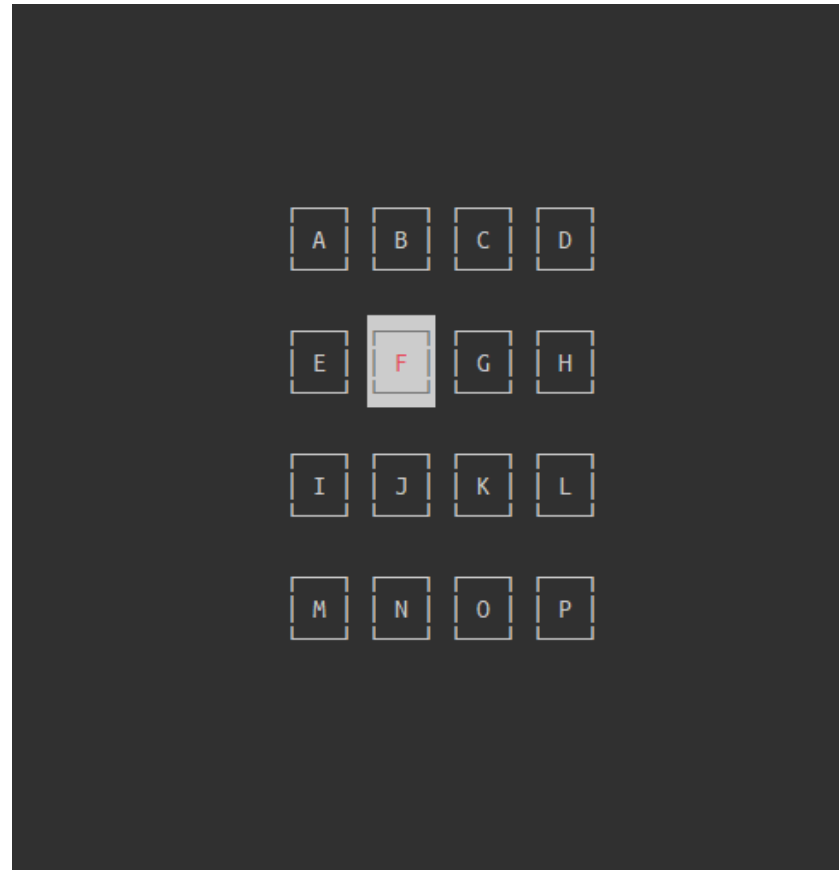
## 1. Main Menu

- 'GAME START' and 'QUIT' options are necessary. **(5 pts)**
- Currently selected option should be highlighted, and it should be possible to select different options using the **<arrow keys>** and **<spacebar>**. **(5 pts)**
- When 'QUIT' is selected, quit the game (exit the program). The program should not exit in any other circumstance (other than when QUIT is entered from the Main Menu). **(5 pts)**
- When 'GAME START' is selected, go to in-game screen. **(5 pts)**



# Specific requirements

## 2. In-game screen



# Specific requirements

## 2. In-game screen

- Display a grid of 4x4 cards, each with different alphabets written. **(Previous Week)**
- A single card should be highlighted to indicate it is being selected. **(10 pts)**
- It should be possible to move the highlight around with **<arrow keys>**. **(10 pts)**  
(Highlight should not disappear at any point. No more than 1 card should be highlighted at the same time.)
- Pressing 'q' at any point in the 'in-game' screen should return the screen back to the Main Menu. (Also, make sure that selecting GAME START once again in the Main Menu, enters back to the 'in-game' screen again.) **(10 pts)**

# Grading policy

- Other specific implementation and designs are all up to you.
  - Ex) Color of highlight, box size, word-phrasing, etc.
- However, **bugs** that go against common-sense will be deducted.
  - Ex) More than 1 option/card is highlighted at the same time.
  - Ex) Some cards don't highlight correctly. / Game crashes when trying to move the highlight past the right-end or bottom-end, etc.
  - Ex) Screen goes blank for no reason. / Game just randomly quits itself.
  - etc.
- Also, make sure that **endwin()** is called before exiting **main()** function.  
(Points will be deducted if this requirement is not met.)

# Grading Environment

- Lab computer & Martini server  
(Code should compile correctly and run in at least one of these environments.)
- Compile command:  
**`gcc <yourfile.c> -o game -O2 -lm -lncurses`**

# Submission

- Submit by E-mail [pp20182ta@gmail.com](mailto:pp20182ta@gmail.com)
- Email title format: [project] week# <student id> <name>  
example) [project] week2 2018-12345 홍길동
- **Submit 1 file:** 20xx-xxxxx.c (Use your student ID as filename)
  - If your implementation is intentionally different from the given specifications (for a good reason), or if additional features are included etc., please specify it for us in the email.
  - If your implementations are same as given specifications, there's no need to write it.

# Please follow the submission format

- Please follow the designated submission format (in previous slide).
- From now on, **points will be deducted if you do not follow it.**
- Ex) Points will be deducted if:
  - email title does not follow the designated format.
  - filename of code is not your student id.
- Also, please just attach as **.c** file. **No need to zip it.**

# Tips

For this week's implementation

# Tip 1: scene

- Maintain a (global) variable that indicates which *scene* to show:
  - At any point in the program, it should either currently be in the 'Main Menu' scene or in the 'in-game' scene.
  - Decide which scene to enter into, based on the flag variable (*scene*).

\* **Flag variable:** variable where different values indicate designated states (or *flags*).  
e.g. 0 == exit, 1 == main-menu, 2 == game-scene, ...



# Tip2: highlight

- You can highlight the currently selected option/card, by turning "attributes" on and off.
- "Attributes" include **color**, **bold**, underline, etc. (See next slides)

# NCURSES Library - Display Attributes

<code>attron(ATTRIBUTES)</code>	Enable given attributes.
<code>attroff(ATTRIBUTES)</code>	Disable given attributes.
<code>A_BOLD</code>	Extra bright or bold
<code>A_UNDERLINE</code>	Underlining
<code>COLOR_PAIR(n)</code>	Set text and background color as input. (Your color pair)

## Usage Example

```
start_color();  
  
init_pair(1, COLOR_BLACK, COLOR_CYAN);  
attron(COLOR_PAIR(1));  
mvprintw(LINES/5, COLS/2, "Highlighted Text");  
  
attron(A_BOLD | A_UNDERLINE);  
mvprintw(LINES/5+2, COLS/2, "Bold+Underline");  
  
attroff(COLOR_PAIR(1) | A_BOLD | A_UNDERLINE);
```

You can search  
more attribute options on google.

- \* Use bitwise 'OR' to combine multiple attributes.
- \* Try copying the example code on the left, to see for yourself how attributes work in NCURSES.

# NCURSES Library - Display Attributes

<code>attron(ATTRIBUTES)</code>	Enable given attributes.
<code>attroff(ATTRIBUTES)</code>	Disable given attributes.
<code>A_BOLD</code>	Extra bright or bold
<code>A_UNDERLINE</code>	Underlining
<code>COLOR_PAIR(n)</code>	Set text and background color as input. (Your color pair)

## Usage Example

```
start_color();  
  
init_pair(1, COLOR_BLACK, COLOR_CYAN);  
attron(COLOR_PAIR(1));  
mvprintw(LINES/5, COLS/2, "Highlighted Text");  
  
attron(A_BOLD | A_UNDERLINE);  
mvprintw(LINES/5+2, COLS/2, "Bold+Underline");  
  
attroff(COLOR_PAIR(1) | A_BOLD | A_UNDERLINE);
```

You can search  
more attribute options on google.

Highlighted Text

**Bold+Underline**

# NCURSES Library - Colors

<code>init_color(COLOR, short r,           short g, short b)</code>	Set color with RGB value. RGB values are mapped in range (0 ~ 999)
<code>init_pair(PAIR, COLOR1,           COLOR2)</code>	Set color pair. COLOR1 for text, COLOR2 for background.

## Pre-Defined Colors

COLOR_BLACK	0
COLOR_RED	1
COLOR_GREEN	2
COLOR_YELLOW	3
COLOR_BLUE	4
COLOR_MAGENTA	5
COLOR_CYAN	6
COLOR_WHITE	7

## Usage Example

```
start_color();
```

```
init_color(8, 600, 749, 400);  
init_pair(1, COLOR_BLACK, COLOR_CYAN);  
init_pair(2, COLOR_RED, 8);
```



Red: 600  
Green: 749  
Blue: 400

## Tip2: highlight (cont.)

- Maintain a variable to indicate which option/card is being currently selected.
- When arrow keys are inputted, change the value of that variable (to indicate the newly selected option/card).
- Before printing the option/card that is currently selected, highlight it by *e.g.* changing the background color.