Project 2-1

Prof. Junbeom Hur
TA. Hyunsoo Kwon
Korea University

Project 2-1 Snake Byte

- Write a program “Snake Byte”.
- Project guideline is as follows.

Project 2-1 Snake Byte

- Firstly, you should input the number of foods for snake, which is presented by a heart (♥) as you can see in Figure 2.

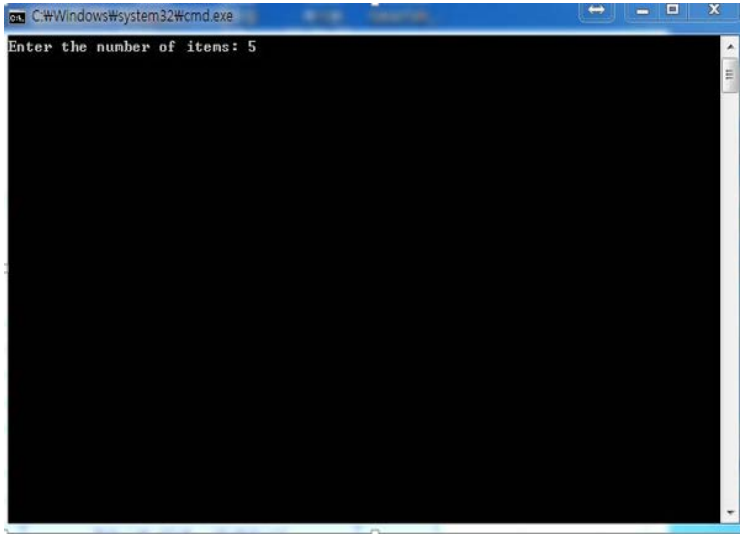


Figure 1

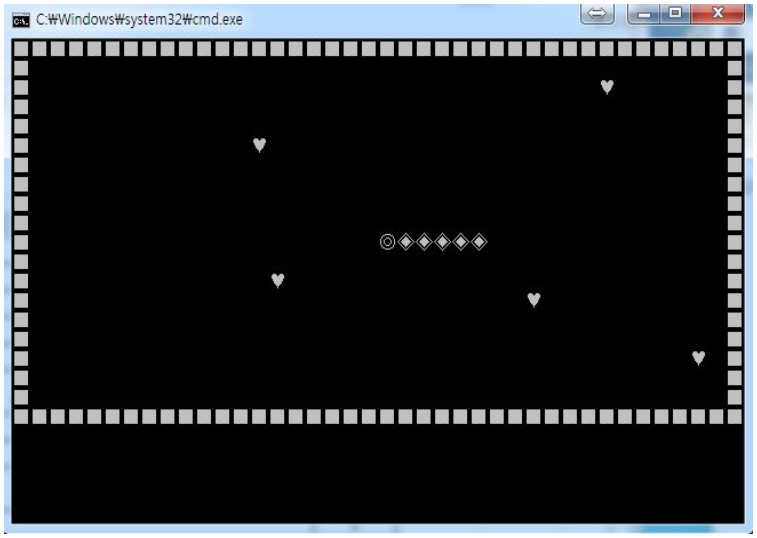


Figure 2

Project 2-1 Snake Byte

- The size of Map(array) is 20 × 40.
- Map boundary is represented by a square (■) and it acts as a barrier.

Figure 3

Project 2-1 Snake Byte

- The initial length of snake is 6 (1 head and 5 tails) and the starting point is randomly chosen.
- Initial direction of snake to move is also random.
- Note. Snake must be located inside the map.

Project 2-1 Snake Byte

- When you type keyboard arrow keys, snake moves to typed direction. Before you type a new direction, snake keeps moving continuously.
- The locations of food are determined randomly except initial location of snake.

Project 2-1 Snake Byte

- Rules of the game.

- You win
  - When the snake eat all of the food in the map.

- You lose
  - when the snake's head bumps into it's tail

Project 2-1 Snake Byte

- A part of project code is provided.
- You should fill in the rest of the program code.
- See details in the next slide.

Project 2-1 Snake Byte

```c
#include<stdio.h>

void main( ) {
    int ekey = 0;

    init( );                    // 변수값 초기화 함수

    while (1) {

        if (_kbhit( )) {        // 키보드 입력이 있으면 1, 없으면 0을 리턴
            ekey = _getch( );   // 입력받은 키 값을 받음
            _flushall( );


        }
        update( );              // 변수 업데이트
        draw( );                // 화면에 출력

        Sleep(50);
    }
    release( );
}
```

Figure 4

- In main function, init(), update(), draw(), release() functions are called.
- _kbhit() function waits the input and if the input exists, return 1, 0 otherwise (In <conio.h>).

Table 1

↑	←	→	↓
72	75	77	80

- _getch() function returns the keyboard input. Table 1 shows the basic keyboard input value.

Project 2-1 Snake Byte

```
void init() {
    sInit();
}

void update() {
    //move

    //collision
}


void draw() {
    int x = 30, y = 20;

    sClear();


    sPrint(x*2, y, "hello");


    sFlipping();
}

void release() {
    sRelease();
}
```

Figure 5

- In init() function, you need to initialize the variables.
- Change the value of variables during the game using the update() function.
- Print the current value of variables using the draw() function.
  (Detailed description of drawing process will be described in later slide)
- Memory deallocation process is implemented in release() function.
- We provide <screen.h, screen.c> including sInit(), sClear(), sPrint(), sFlipping(), and sRelease().

Project 2-1 Snake Byte

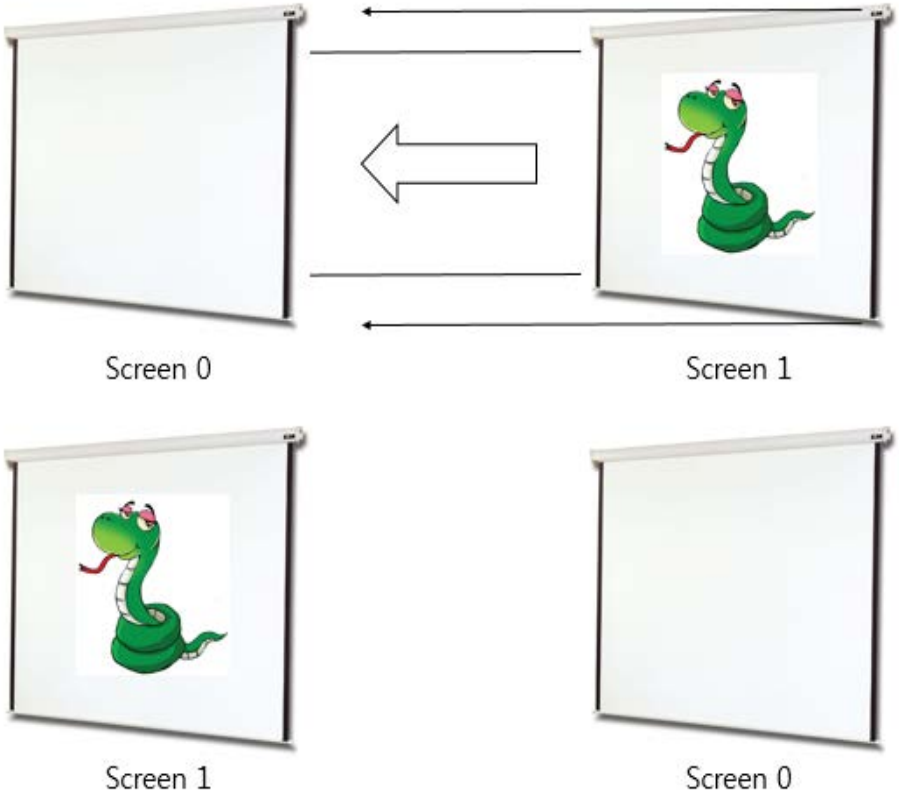
```
typedef struct _pos {

}pos;

typedef struct _object {

}object;

typedef struct _snake {

}snake;
```

Figure 6

- You should implement 3 structs described in Figure 6.
- _pos presents the location value
- _object presents the item.
- _snake presents head and tail of snake.

Project 2-1 Snake Byte

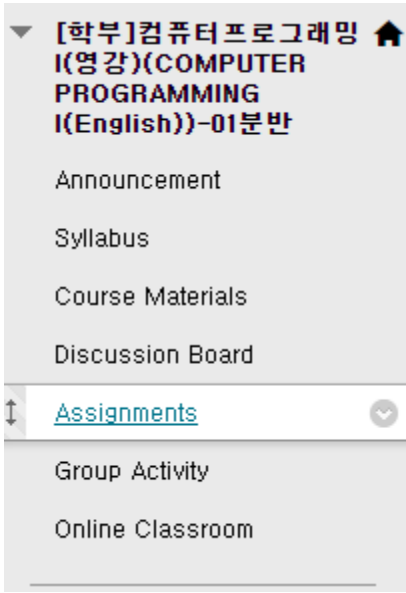
※ Understanding of the drawing process in the screen



- First you should create two Screen 0 and 1. Screen 0 will be displayed in the window and Screen 1 exist in the background.
- If you want to display updated screen, draw on the background Screen 1 and exchange it with the Screen 0.
- In draw() function, sClear() function is used to erase the background screen and sPrint() function is used to draw updated screen on the background screen.
- sFipping() function exchange the Screen 0 and 1.

Project 2-1 Snake Byte

- Submit a file “2016123456_proj2.zip”
  - Source File, Executable File



- BlackBoard(kulms.korea.ac.kr) → Assignments
- Due Date : 2016/05/31 23:59