

The team project of System Programming

- Snake Game -

2013019033	수학과	정인준
2012105108	컴퓨터학부	황연우
2013064009	미술학과	김현지
2012105043	컴퓨터학부	박철우

The table of contents

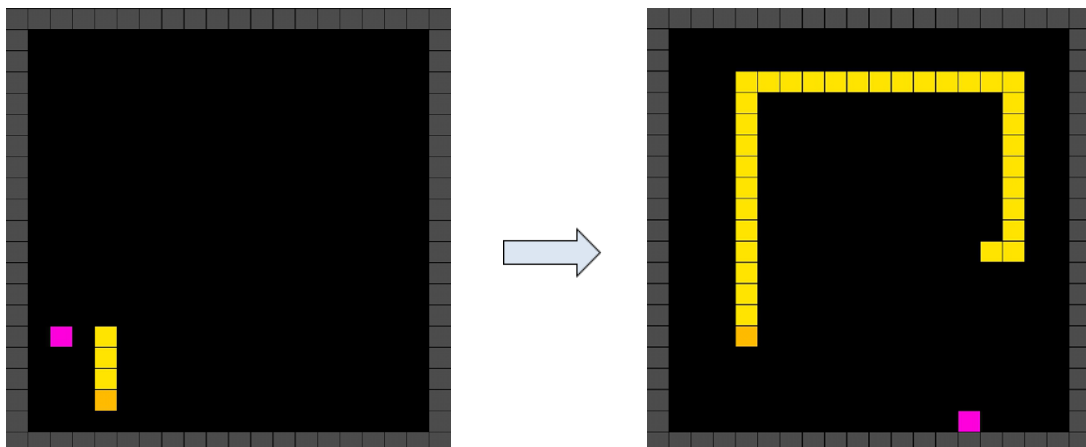
■ 주제 소개	-----	page 3
■ 코드 설명		
◆ moving_test.c	-----	page 4
◆ moving_add_snake.c	-----	page 5
◆ snake_ver1.c	-----	page 7
◆ snake_ver2.c	-----	page 9
◆ snake_ver3.c	-----	page 10
■ 참고 자료	-----	page 10

주제 소개

저희는 리눅스 환경에서 curses.h를 활용하여 Snake Game(뱀게임)을 제작하였습니다. Snake Game이란, 사각형의 격자들로 구성된 field 내에서 일부의 격자들로 이루어진 Snake가 방향키를 통해 이동하며 item을 획득 시, Snake의 길이가 길어지는 고전 게임입니다.

이 게임은 가능한 오랜 시간동안 많은 item을 획득하는 것이 목적입니다. 아래와 같은 행위가 발생할 시, 게임은 종료가 됩니다.

1. 경계(edge)에 Snake의 head가 닿을 경우
2. Snake의 head가 자신의 body에 닿을 경우



저희가 개발한 Snake Game은 기존의 방식과 다른 형태로 게임이 진행됩니다.

1. 시간의 흐름에 따라 score가 상승
2. score가 100 증가할 때마다, snake의 길이가 1 증가
3. item을 1개 획득 시, score가 50 증가
4. snake의 head가 경계(edge), 자신의 body, 그리고 장애물에 닿을 시, 종료

코드 설명

1. moving_test.c (add moving a block)

moving_test.c는 Ch 7. bounce_async.c의 on_alarm() 및 on_input()을 활용하여 실시간으로 이동하는 block과 입력된 값에 따라 block의 방향을 상, 하, 좌, 우로 전환하는 것을 구현하는 것이 목적입니다.

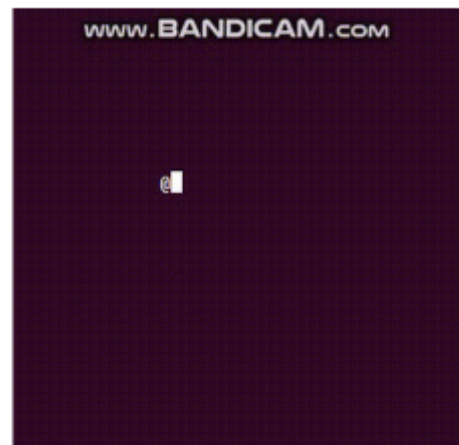
- moving_test.c (add moving a block)

```
/* move the snake in real time */
void on_alarm(int signum){
    updateWindow();
    makeObstacle();
    makeItem();
    changeDir(dir);
}
```

실시간 타이머로 인한 이동

```
/* change the direction of snake */
void on_input(int signum){
    int c = getchar(); /* grep the char */
    if (c == 'Q' || c == EOF || c == 'q' ) /* end */
        done = 1;
    else if ( c == 'W' || c == 'w' )
        dir = UP;
    else if ( c == 'A' || c == 'a' )
        dir = LEFT;
    else if ( c == 'S' || c == 's' )
        dir = DOWN;
    else if ( c == 'D' || c == 'd' )
        dir = RIGHT;
}
```

입력 값에 따른 방향 전환



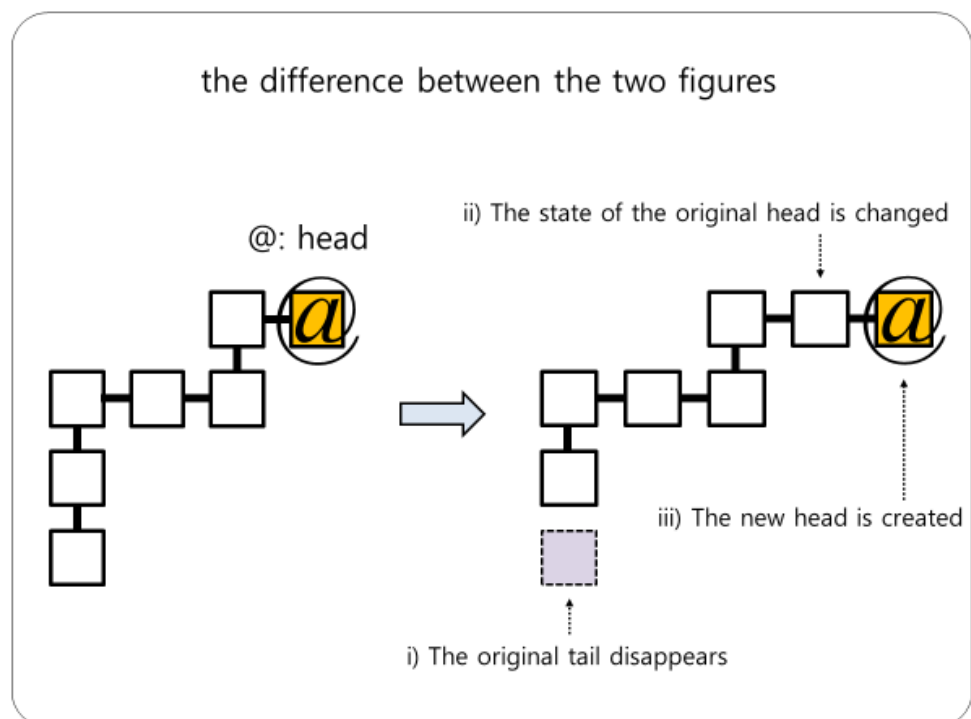
moving_test.c 구현

코드 설명

2. moving_add_snake.c (converting block into block)

moving_add_snake.c는 moving_test.c을 활용하여 이동하는 대상을 block이 아닌 snake로 구현하는 것이 목적입니다. 하나의 snake를 linked list로 표현하기 위하여 snake 구조체를 만들고, 한번의 이동을 하는 과정에서 어떠한 변화가 있는지 관찰한 결과 아래와 같은 3가지 특징을 발견하였습니다.

- i) 기존의 tail이 사라짐
- ii) 기존의 head가 body로 변경
- iii) 새로운 head가 추가



이와 같은 절차를 기반으로 moving_add_snake.c의 코드를 다음과 같이 개발하였습니다.

- moving_add_snake.c (converting block into snake)

```
/* the cell of snake */
typedef struct node {
    int row;
    int col;
    NodePtr next;
    NodePtr back;
} Node;

/* a snake */
typedef struct snake {
    NodePtr first;
    NodePtr last;
    int lenth;
} Snake;
```

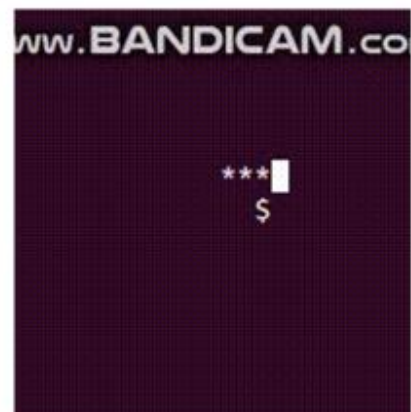
Linked list를 표현 하기 위한 snake 구조체

새로운 header 추가하는 insertHeader()

```
/* update the head of snake */
void insertHeader(int row, int col, SnakePtr s) {
    NodePtr newHeader, preHeader;
    newHeader = create(row, col);
    preHeader = s->first;
    preHeader->next = newHeader;
    newHeader->back = preHeader;
    s->first = newHeader;
}
```

기존 tail 삭제하는 deleteLast()

```
/* delete and change the tail of snake */
void deleteLast(){
    mvaddch(s->last->row, s->last->col, BLANK);
    s->last = s->last->next;
}
```



moving_add_snake.c 구현

코드 설명

3. snake_ver1.c (add item, unchanging, direction, end window)

snake_ver1.c는 field 내 random한 위치에 item을 생성하고, 방향전환이 불가능한 제한 조건과, 그리고 종료 화면을 추가하는 것이 목적입니다.

```
void makeItem(){
    if ( ( (int) score ) % 300 == 0 ){
        mvaddch(YPOS, XPOS, BLANK);
        while(1){
            srand((unsigned)time(NULL));
            XPOS = (rand() % 47) + 15;
            YPOS = (rand() % 21) + 8;
            if (!checkPass(YPOS, XPOS))
                break;
        }
        mvaddch(YPOS, XPOS, ITEM);
    }
}
```

Random 함수를 사용하여 random한 위치에 item 생성

```
/* change the position of the snake according to new direction or present direction */
void changeDir(int dir){
    /* changing constrained if the new direction is same of the counter present direction */
    switch(dir){
        case LEFT:
            if(s->first->row == s->first->back->row && ((s->first->col) - 1) == s->first->back->col)
                dir = predir;
            break;
    }
```

새롭게 입력된 direction이 적용될 수 없는 상황의 제한 조건 추가

```
void endWindow(){
    clear();
    setWindow();
    updateWindow();
    mvaddstr(17, 34, "Game over.");
    mvaddstr(19, 28, "Press any key to quit.");
    refresh();
    pause();
    done = 1;
}
```

게임 종료 화면 생성

코드 설명

5. snake_ver3.c (add comment)

snake_ver3.c는 주석과 코드를 정리하는 것이 목적입니다.

참고 자료

강의 자료

- Chap 6. Programming for Humans Terminal Control and Signals
- Chap 7. Event-Driven Programming Writing a Video Game

위키피디아

- Snake (video game genre)
: [https://en.wikipedia.org/wiki/Snake_\(video_game_genre\)](https://en.wikipedia.org/wiki/Snake_(video_game_genre))