

Block Stack 개선

예제 8_6_1

학번 : 2021864039

이름 : 김태헌

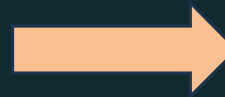
목차

- 프로그램 개선점
- 코드 설명
- 실행 화면
- 추가 개선점



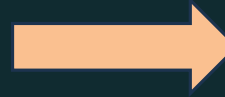
프로그램 변경점

아무키나 누르면 실행



난이도 선택 후 실행

기본 좌우 이동 속도(sleep(50))



난이도는 1 - 5 단계

난이도별 블록의 좌우 이동 속도 (sleep의 값)

1단계 80 | 2단계 60 | 3단계 40 | 4단계 20 | 5단계 10

main

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <windows.h>

#define box_length 15 // 게임의 영역(좌우 길이)
#define box_height 15 // 바닥의 높이(상하 길이)

void intro_game(void);
void game_control(int sleep_value);
void gotoxy(int x, int y);
int left_right_move(int sleep_value);
void move_down(int x);
void draw_rectangle(int c, int r);
int max_block(void);

int block_stack[box_length * 2 + 1] = {0}; // 해당 위치의 값을
0으로 초기화
```

```
int main(void)
{
    int sleep_value;
    intro_game();

    do
    {
        scanf("%d", &sleep_value);
        if(sleep_value < 1 || sleep_value > 5)
        {
            printf("잘못된 난이도 설정입니다. 1부터 5 사이의 값을
            입력하세요.\n");
        }
    }while(sleep_value < 1 || sleep_value > 5);

    game_control(sleep_value);
    gotoxy(1, box_height + 3);
    printf("게임이 종료되었습니다.\n");
    return 0;
}
```

변경점

1. 함수 호출에서 game_control과 left_right_move에서 int sleep_value 값을 받음
2. Main에서 난이도를 입력 받음 1-5 사이의 숫자를 입력 받고 그 이외의 숫자를 입력하면 다시 print문을 출력하고 입력받음

Intro_game

변경점

1. 난이도 선택을 안내하는 문구 추가

블록 쌓기

블록이 좌우로 움직일 때 스페이스키를 누르면
블록이 떨어져 바닥에 쌓입니다.

난이도 선택:

1단계 매우쉬움 | 2단계 쉬움 | 3단계 보통 | 4단계 어려움 | 5단계 매우 어려움
난이도 설정 (1-5): _

```
void intro_game(void)
{
    system("cls");
    printf("블록 쌓기\n\n");
    printf("블록이 좌우로 움직일 때 스페이스키를 누르면\n");
    printf("블록이 떨어져 바닥에 쌓입니다.\n\n");
    printf("난이도 선택.\n");
    printf("1단계 매우쉬움 | 2단계 쉬움 | 3단계 보통 | 4단계 어려움 | 5단계 매우 어려움\n");
    printf("난이도 설정 (1-5): ");

    getch();
}
```

game_control

gotoxy

```
void game_control(int sleep_value)
{
    int x, count = 0;
    system("cls");
    draw_rectangle(box_length, box_height);
    gotoxy(box_length * 2 + 5, 3);
    printf("블록의 개수: %2d", box_height);
    gotoxy(1, box_height + 3);
    printf("스페이스키를 누르면 블록이  
떨어지고\n");
    printf("바닥에 쌓입니다.\n");
    while (count < box_height)
    {
        gotoxy(box_length * 2 + 5, 4);
        printf("시도한 횟수: %2d", count + 1);
        gotoxy(box_length * 2 + 5, 5);
        printf("쌓인 블록수: %2d",
max_block());
        x = left_right_move(sleep_value);
        move_down(x);
        count++;
        getch();
    }
}
```

```
void gotoxy(int x, int y)
{
    COORD Pos = {x - 1, y - 1};

    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),
Pos);
}
```

left_right_move

변경점

Sleep의 값을 결정하는 Switch 문 추가

```
int left_right_move(int sleep_value)
{
    int x = 3, y = 2, temp = 2;
    int sleep_duration;
    switch(sleep_value) {
        case 1:
            sleep_duration = 80;
            break;
        case 2:
            sleep_duration = 60;
            break;
        case 3:
            sleep_duration = 40;
            break;
        case 4:
            sleep_duration = 20;
            break;
        case 5:
            sleep_duration = 10;
            break;
    }
```

```
do
{
    x += temp;
    if (x > (box_length * 2)) // x방향 최대값
        설정
        temp = -2;
    if (x < 3)
    {
        x = 3;
        temp = 2;
    }

    gotoxy(x, y);
    printf("□");
    Sleep(sleep_duration);
    // 블록이 좌우로 움직이는 속도를 조절
    gotoxy(x, y);
    printf(" ");

} while (!kbhit());
block_stack[x] += 1;
return x;
}
```

move_down

Max_block

```
void move_down(int x)
{
    int y;
    for(y=2;y<box_height+2-
block_stack[x];y+=1)
    {
        gotoxy(x, y);
        printf("□");
        Sleep(20);
        gotoxy(x, y);
        printf(" ");
        Sleep(10);
    }
    gotoxy(x,box_height+2-block_stack[x]);
    printf("□");
}
```

```
int max_block(void)
{
    int i, max = 0;
    for (i = 1; i < box_height * 2 + 1; i++)
    {
        if (max <= block_stack[i])
            max = block_stack[i];
    }
    return max;
}
```



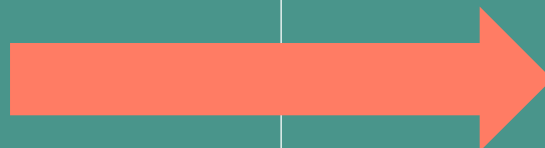
draw_rectangle

```
void draw_rectangle(int c, int r)
{
    int i, j;
    unsigned char a=0xa6;
    unsigned char b[7];
    for(i=1;i<7;i++)
        b[i]=0xa0+i;

    printf("%c%c",a, b[3]);
    for(i=0;i<c*2+1;i++)
        printf("%c%c", a, b[1]);
    printf("%c%c", a, b[4]);
    printf("\n");
}
```

```
for(i=0;i<r;i++)
{
    printf("%c%c", a, b[2]);
    for(j=0;j<c*2+1;j++)
        printf(" ");
    printf("%c%c",a, b[2]);
    printf("\n");
}
printf("%c%c", a, b[6]);
for(i=0;i<c*2+1;i++)
    printf("%c%c", a, b[1]);
printf("%c%c", a, b[5]);
printf("\n");
```

변경 전



변경 후

블록 쌓기

블록이 좌우로 움직일때 스페이스키를 누르면
블록이 떨어져 바닥에 쌓입니다.

아무키나 누르면 게임을 시작합니다.



블록 쌓기

블록이 좌우로 움직일 때 스페이스키를 누르면
블록이 떨어져 바닥에 쌓입니다.

난이도 선택

1단계 매우쉬움 | 2단계 쉬움 | 3단계 보통 | 4단계 어려움 | 5단계 매우 어려움
난이도 설정 (1-5):



출처

<https://jaimemin.tistory.com/191>

