|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 12주차 과제 | | | | | |
| 학년 | 2 | 학번 | 2017707007 | 이름 | 강민준 |

1. 과제 설명

D길이를 가진 함정을 탈출하면 승리하는 게임이다.

함정에는 상자가 던져지는데, 이 상자안에는 체력을 보충할 수 있는 음식이 들어 있다.

상자를 쌓아서 탈출할 수 있는 발판을 만들거나, 음식을 먹어 체력을 보충할 수 있다.

단, 두 행동을 동시에 할 수 없으며 둘 중 하나를 선택해야한다.

HP=10, 시각=0 으로 시작하고 HP는 시간이 지날 때 마다 1씩 감소한다. HP가 0이되는 순간 탈출은 실패하고, 함정의 깊이 D를 넘으면 탈출에 성공한다.

1. 과제 이론

#include<stdio.h>

#include<stdlib.h>

typedef struct \_man {

unsigned int height;

unsigned int hp;

unsigned int escape\_time;

}MAN;

typedef struct \_box {

unsigned int time;

unsigned int height;

unsigned int hp;

}BOX;

typedef struct \_trap {

unsigned int trap\_height;

unsigned int box\_num;

}TRAP;

void print\_man(MAN); //출력전용

void input\_box(int, BOX \*); //입력전용

void sort\_box(int, BOX \*); //Random하게 들어온 Box 정보 정렬

int main()

{

char flag;

MAN man = { 0,10, 0}; //height, hp

TRAP trap;

BOX \*box\_p;

printf("\*\*\*\* trap \*\*\*\*\n");

printf("trap depth : ");

scanf("%d", &trap.trap\_height);

printf("box num : ");

scanf("%d", &trap.box\_num);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

box\_p = (BOX\*)malloc(trap.box\_num \* sizeof(BOX));

input\_box(trap.box\_num, box\_p);

sort\_box(trap.box\_num, box\_p);

while (flag = getchar() != '\n' && flag != EOF);

int hptime = 0;

while (1) {

print\_man(man);

if (10 - man.hp == box\_p[hptime].time) {

printf("Do You eat food? [PRESS Y/N] : ");

scanf("%c", &flag);

if (flag == 'Y' || flag == 'y')

{

man.hp += box\_p[hptime].hp;

}

else {

man.height += box\_p[hptime].height;

}

while (flag = getchar() != '\n' && flag != EOF);

hptime++;

}

man.hp--;

man.escape\_time++;

if (man.height >= trap.trap\_height) { // man이 함정을 뛰어넘는순간.

system("cls");

man.hp++;

print\_man(man);

printf("YOU WIN!\n");

return 1;

system("exit");

free(box\_p);

}

if (!man.hp) { //gp가 0되는순간.

system("cls");

print\_man(man);

printf("YOU DIE!\n");

return 1;

system("exit");

free(box\_p);

}

}

}

void sort\_box(int box\_range, BOX \* box\_p)

{

BOX temp;

for (int i = 0; i < box\_range; i++) { //BOX값이 랜덤으로 입력되는데, time순으로 순차적으로 변경해놓은것.

for (int j = 0; j < box\_range - i; j++) {

if (box\_p[j].time > box\_p[j + 1].time) {

temp = box\_p[j];

box\_p[j] = box\_p[j + 1];

box\_p[j + 1] = temp;

}

}

}

}

void input\_box(int box\_range, BOX\* box\_p) { //input 전용함수

for (int i = 0; i < box\_range; i++) {

printf("\*\*\*\* Box %d \*\*\*\*\n", i + 1);

printf("time : ");

scanf("%d", &box\_p[i].time);

printf("height : ");

scanf("%d", &box\_p[i].height);

printf("hp : ");

scanf("%d", &box\_p[i].hp);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n");

}

}

void print\_man(MAN man) { //단순 출력전용함수

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("escape time : %d\n",man.escape\_time);

printf("HP : %d\n", man.hp);

printf("Height : %d\n", man.height);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

사람의 정보를 갖는 구조체(MAN)과, 박스의 정보를 가지는(BOX)구조체, 함정의 정보를 가지는 (TRAP)구조체를 선언하고 활용하였다.

초기상태의 사람의 hp는 10에, 사람의 높이는 0으로 설정하고, 함정의 높이를 입력받고, 박스의 개수 및 정보를 입력하고 게임을 진행한다. 이때 박스 내의 음식을 먹을 지, 혹은 쌓을 지 판단한다.

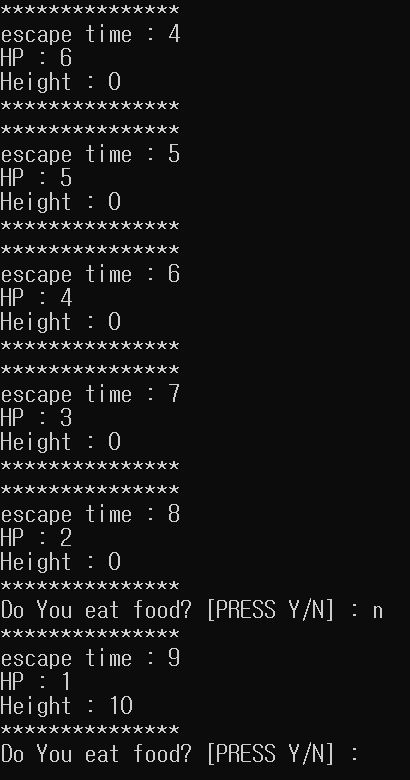
후 문제의 조건대로 출력한다.

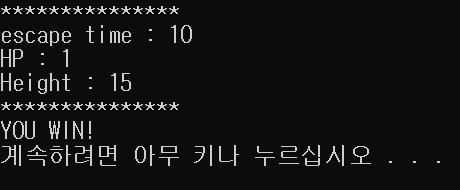
1. 주요 소스 설명

|  |
| --- |
| #1번 주요소스  box\_p = (BOX\*)malloc(trap.box\_num \* sizeof(BOX));  input\_box(trap.box\_num, box\_p);  sort\_box(trap.box\_num, box\_p);    while (flag = getchar() != '\n' && flag != EOF);  int hptime = 0;  while (1) {    print\_man(man);    if (10 - man.hp == box\_p[hptime].time) { // box를 얻을 때,    printf("Do You eat food? [PRESS Y/N] : ");  scanf("%c", &flag);  if (flag == 'Y' || flag == 'y')  {  man.hp += box\_p[hptime].hp;  }  else {  man.height += box\_p[hptime].height;  }  while (flag = getchar() != '\n' && flag != EOF);  hptime++;  }  man.hp--;  man.escape\_time++;    if (man.height >= trap.trap\_height) { // man이 함정을 뛰어넘는순간.  system("cls");  man.hp++;  print\_man(man);  printf("YOU WIN!\n");  return 1;  system("exit");  free(box\_p);  }  if (!man.hp) { //hp가 0되는순간. 종료  system("cls");  print\_man(man);  printf("YOU DIE!\n");  return 1;  system("exit");  free(box\_p);  }  }  }  설명은 주석과 같다. |
| #2번 주요소스  void input\_box(int box\_range, BOX\* box\_p) { //input 전용함수  for (int i = 0; i < box\_range; i++) {  printf("\*\*\*\* Box %d \*\*\*\*\n", i + 1);  printf("time : ");  scanf("%d", &box\_p[i].time);  printf("height : ");  scanf("%d", &box\_p[i].height);  printf("hp : ");  scanf("%d", &box\_p[i].hp);  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n");  }  } |
| #3번 주요소스  void print\_man(MAN man) { //단순 출력전용함수  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  printf("escape time : %d\n",man.escape\_time);  printf("HP : %d\n", man.hp);  printf("Height : %d\n", man.height);  printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  } |

1. 실행화면







System(“cls”) 명령어를 사용하여 결과값을 출력할 때는 화면이 지워짐.

둘다 N를 입력한 상황

1. 고찰

과제를 하며, time을 어떻게 흐르게 해야하는지 고민함. 실제 time.h에 포함된 함수를 사용할지, 아두이노에서 사용되는 delay함수와 같은 것을 찾아서 사용할지를 고민했지만 그렇지 않고, while(1)으로 무한루프로 구성하여, hp를 1씩 감소하면서 time으로 만들었음.

따라서 박스가 던져질 때만 멈추고 내용물을 먹을지 물어보는 구조를 띄고있음.

크게 어렵지 않은 게임이여서 소스코드를 짜는데 오랜시간이 걸리지 않아 성공적인 작업이었다 판단.