

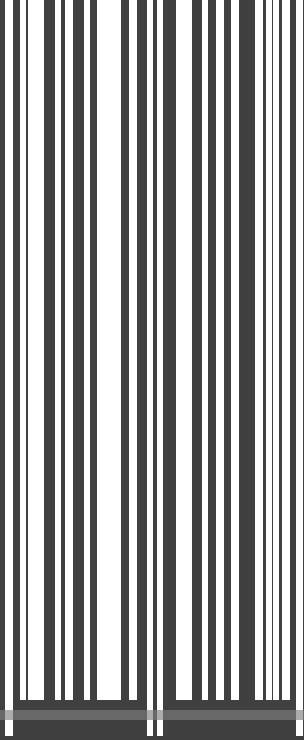
「줄다리기 게임(8_10_1)



강의: 게임 프로그래밍



2020975005
소프트웨어학과 강현준



CONTENTS

- ————— 기존 코드
- ————— 추가 기능 및 수정

1

Origin



기존 코드

기존 코드

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



라이브러리 선언

```
double intro_game(char team_name[][7]);
void gotoxy(int x, int y);
void display_line(int rand);
void display_start(char t_name[][7], int s_w[], int s_l[]);
void make_decision(int r_s, int s_w[], int s_l[], char t_name[][7]);
void game_control(int *r_s, clock_t st, double *ps);
```



함수, 전역변수 선언

기존 코드 : main

```
int main(void)
{
    int score_win[2]={0}, score_loose[2]={0}, r_start;
    char team_name[2][7];
    double pst, game_time;
    clock_t start;

    srand(time(NULL));
    game_time=intro_game(team_name);
    do
    {
        display_start(team_name, score_win, score_loose);
        r_start=20;
        start=clock();
        pst=0;
        do
        {
            game_control(&r_start, start, &pst);
        }while(pst<game_time && ((8<r_start) && (r_start<32)));
        make_decision(r_start, score_win, score_loose, team_name);
    }while((score_win[0]<2) && (score_win[1]<2));
    gotoxy(1, 14);
    printf("게임을 종료합니다. \n");
    return 0;
}
```

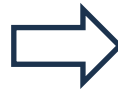
← 변수 선언

← 팀 이름, 제한시간 입력 받기
초기 줄 위치 등 게임설정

기존 코드 : intro_game

```
double intro_game(char team_name[][7])
{
    double game_time;
    int i;
    printf("줄다리기 게임 \n\n");
    printf("3번 싸워서 2번 먼저 이긴 팀이 승자입니다.\n\n");
    printf("참가자는 두 팀입니다. \n");
    for(i=0;i<2;i++)
    {
        printf("%d번 참가팀의 이름을 입력하고 Enter>", i+1);
        scanf("%s", &team_name[i]);
    }

    printf("\n");
    printf("줄다리기 게임의 제한 시간(초 단위)을 입력하고 Enter>");
    scanf("%lf", &game_time);
}
```



줄다리기 게임

3번 싸워서 2번 먼저 이긴 팀이 승자입니다.

참가자는 두 팀입니다.

1번 참가팀의 이름을 입력하고 Enter>d

2번 참가팀의 이름을 입력하고 Enter>c

줄다리기 게임의 제한 시간(초 단위)을 입력하고 Enter>10

기존 코드 : intro_game

```
system("cls");

gotoxy(1,1);
printf("제한시간 : %5.1f 초", game_time);
gotoxy(30, 1);
printf("경과시간 : %4.1f초",0);

gotoxy(38, 6);
printf("기준점");
gotoxy(40, 7);
printf("▼");
gotoxy(40, 9);
printf("▲");
return game_time;
```



```
제한시간 : 10.0 초      경과시간 : 0.0초

d                                기준점
0승, 0패                      ▼
                                □□□□◆□□□□■□□□□◆□□□□
                                ▲
```

기존 코드 : gotoxy, display_line

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



콘솔 화면에서 커서를 지정
한 위치로 이동시킴

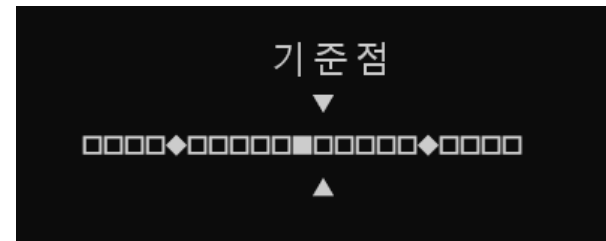
```
void display_line(int rnd)
{
    int i, line[21]={0};
    line[10]=1;
    line[4]=2;
    line[16]=2;
    gotoxy(1, 8);
    for(i=0;i<78;i++)
        printf(" ");

    gotoxy(rnd, 8);

    for(i=0;i<21;i++)
        if (line[i]==0)
            printf("□ ");
        else if (line[i]==1)
            printf("■ ");
        else
            printf("◆ ");
}
```



게임 진행 중 로프의 현재 상태를 화면
에 표시



기존 코드 : display_start

```
void display_start(char t_name[][7], int s_w[], int s_l[])
{
    int i, r_start=20;
    display_line(20);

    for(i=0;i<2;i++)
    {
        gotoxy(i*55+10, 6);
        printf("%s", t_name[i]);
        gotoxy(i*55+10, 7);
        printf("%d승, %d패", s_w[i], s_l[i]);
    }

    gotoxy(1,13);
    printf("아 무 키 나 누 르 면 경 기 를 시 작 합 니 다 . ");
    getch();
    gotoxy(1, 12);
    printf(" ");
    gotoxy(1,13);
    printf(" ");
}
```

각 라운드의 시작 시 팀 정보와 스코어를 화면에 표시하고, 게임 시작을 대기



```
제한시간 : 10.0 초      경과시간 : 0.0초

d                                기준점                                c
0승, 0패                  ▼                                         0승, 0패
                        □□□□◆□□□□□□□□□□◆□□□□
                        ▲
아무키나 누르면 경기를 시작합니다. |
```

기존 코드 : game_control

```
void game_control(int *r_s, clock_t st, double *ps)
{
    int rnd, i;
    clock_t end;

    rnd=rand()%100;
    if (rnd%2)
        rnd=rnd%4;
    else
        rnd=-rnd%4;
    if (rnd<0)
    {
        for(i=*r_s;i>(*r_s+rnd);i--) //좌측으로 좌측방향으로 이동
        {
            display_line(i);
            Sleep(100);
        }
    }
    else
    {
        for(i=*r_s;i<(*r_s+rnd);i++) //우측으로 우측방향으로 이동
        {
            display_line(i);
            Sleep(100);
        }
    }
    if (rnd<0)
        *r_s=i+1;
    else
        *r_s=i-1;

    end=clock();
    *ps=(double)(end-st)/CLK_TCK;
    gotoxy(30, 1);
    printf("경과 시간 : %4.1f초", *ps);
}
```



랜덤으로 줄 위치 이동, 경과시간 출력



2

Change



추가 기능 및 수정

●● 추가 기능 : 프롬프트 제목 변경

```
void set_console_size() {  
    system("mode con: cols=100 lines=30");  
    SetConsoleTitle("Tug Of War"); // 콘솔 창 제목 설정  
}
```



추가 기능 : 시작화면

```
void display_title() {
    system("cls");
    set_text_color(10);

    gotoxy(20, 5);
    printf("_____");
    gotoxy(20, 6);
    printf("|_ _| | _ | / _ | | | |");
    gotoxy(20, 7);
    printf("| | | _ _ _ _ | | | | | | | | _ _ _");
    gotoxy(20, 8);
    printf(" | | | | | / _ | | | | | | | | | / _ | | |");
    gotoxy(20, 9);
    printf(" | | | | | | | | | | | | | | | | | | | |");
    gotoxy(20, 10);
    printf(" \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \");
    gotoxy(20, 11);
    printf("      _/ |");
    gotoxy(20, 12);
    printf("      |_/");

    set_text_color(14);
    gotoxy(35, 14);
    printf("Welcome to the Tug of War Game!");
    gotoxy(30, 16);
    printf("1. 랭킹 보기 or 아무 키나 눌러서 시작하기");
    set_text_color(7);

    Sleep(500);
    gotoxy(40, 18);
    set_text_color(12);
    printf("<<< Are you ready? >>>");

    set_text_color(7);
}
```



추가 기능 : 랭킹

```
void show_rankings() {
    system("cls");
    Rank rankings[MAX_RANK];
    int count = load_rankings(rankings);

    gotoxy(40, 5);
    set_text_color(14);
    printf("=== 랭킹 ===");
    set_text_color(7);

    for (int i = 0; i < count; i++) {
        gotoxy(35, 7 + i);
        printf("%d위: %s - 난이도: %d, 남은 시간: %.1f초", i + 1, rankings[i].name, rankings[i].difficulty, rankings[i].remaining_time);
    }

    if (count == 0) {
        gotoxy(40, 9);
        printf("아직 랭킹이 없습니다.");
    }
}

void update_rankings(char team_name[], int difficulty, double remaining_time) {
    Rank rankings[MAX_RANK + 1]; // 최대 랭킹 수보다 1 크게!
    int count = load_rankings(rankings);

    // 새로운 기록 추가
    strcpy(rankings[count].name, team_name);
    rankings[count].difficulty = difficulty;
    rankings[count].remaining_time = remaining_time;
    count++;

    // 랭킹 정렬
    qsort(rankings, count, sizeof(Rank), compare_rank);

    // 최대 랭킹 수 유지
    if (count > MAX_RANK) {
        count = MAX_RANK;
    }

    // 랭킹 저장
    save_rankings(rankings, count);
}

int compare_rank(const void *a, const void *b) {
    Rank *rankA = (Rank *)a;
    Rank *rankB = (Rank *)b;

    // 난이도 내림차순 비교
    if (rankA->difficulty != rankB->difficulty) {
        return rankB->difficulty - rankA->difficulty;
    }
    else {
        // 남은 시간 내림차순 비교
        if (rankA->remaining_time < rankB->remaining_time) {
            return 1;
        }
        else if (rankA->remaining_time > rankB->remaining_time) {
            return -1;
        }
        else {
            return 0;
        }
    }
}
```

```
void save_rankings(Rank rankings[], int count) {
    FILE *fp = fopen(RANKING_FILE, "w");
    if (fp == NULL) {
        printf("랭킹 파일을 열 수 없습니다.\n");
        return;
    }

    for (int i = 0; i < count; i++) {
        fprintf(fp, "%s %d %.1f\n", rankings[i].name, rankings[i].difficulty, rankings[i].remaining_time);
    }

    fclose(fp);
}

int load_rankings(Rank rankings[]) {
    FILE *fp = fopen(RANKING_FILE, "r");
    int count = 0;

    if (fp == NULL) {
        return 0; // 파일이 없으면 랭킹이 없음
    }

    while (fscanf(fp, "%s %d %.1f", rankings[count].name, &rankings[count].difficulty, &rankings[count].remaining_time) == 3) {
        count++;
        if (count >= MAX_RANK) {
            break;
        }
    }

    fclose(fp);
    return count;
}
```



=== 랭킹 ===

1위: 랭킹용 - 난이도: 2, 남은 시간: 3.1초
2위: d - 난이도: 1, 남은 시간: 10.7초

아무 키나 눌러서 메인 화면으로 돌아가세요.

추가 기능 : 게임방법 보기

```
else if (key == '2') {  
    show_instructions();  
  
    printf("\n\n아무 키나 눌러서 메인 화면으로 돌아가세요.")  
    _getch();  
    system("cls");  
    continue;  
}
```

```
void show_instructions() {  
    system("cls");  
    set_text_color(14);  
    gotoxy(40, 5);  
    printf("=== 게임 방법 ===");  
    set_text_color(7);  
  
    gotoxy(20, 7);  
    printf("1. 게임이 시작되면, 제한 시간 내에 스페이스바를 빠르게 눌러 줄을 끌어당기세요.");  
    gotoxy(20, 9);  
    printf("2. 난이도가 높을수록 제한 시간이 짧아지고 상대 팀이 더 강해집니다.");  
    gotoxy(20, 11);  
    printf("3. 제한 시간이 끝나거나 상대 팀에게 끌려가면 라운드에서 패배합니다.");  
    gotoxy(20, 13);  
    printf("4. 2라운드를 먼저 승리하면 게임에서 승리합니다.");  
}
```



2. 게임 방법 보기

=== 게임 방법 ===

1. 게임이 시작되면, 제한 시간 내에 스페이스바를 빠르게 눌러 줄을 끌어당기세요.
2. 난이도가 높을수록 제한 시간이 짧아지고 상대 팀이 더 강해집니다.
3. 제한 시간이 끝나거나 상대 팀에게 끌려가면 라운드에서 패배합니다.
4. 2라운드를 먼저 승리하면 게임에서 승리합니다.

아무 키나 눌러서 메인 화면으로 돌아가세요.



추가 기능 : 난이도

```
display_title();
getch();

system("cls");
game_time = intro_game(team_name);

int difficulty;

printf("\n\n\n");
printf("          난이도를 선택하세요 (1: 쉬움, 2: 중간, 3: 어려움): ");
if (scanf("%d", &difficulty) != 1 || difficulty < 1 || difficulty > 3) {
    printf("잘못된 난이도를 입력했습니다. 게임을 종료합니다.\n");
    return 1;
}

if (difficulty == 1) game_time = 15;
else if (difficulty == 2) game_time = 12;
else if (difficulty == 3) game_time = 10;

countdown();

int computer_move = (rand() % 2) + 1; // 컴퓨터가 당기는 양 (1~2)
Sleep(120 - (difficulty * 20)); // 컴퓨터가 더 빠르게 당김.

*position += computer_move; // 컴퓨터가 줄을 당기는 방향
if (*position >= LINE_LENGTH) *position = LINE_LENGTH - 1;
```



팀 이름을 입력하세요 : 겐프

난이도를 선택하세요 (1: 쉬움, 2: 중간, 3: 어려움): |



추가 기능 : 카운트다운

```
void countdown() {  
    system("cls");  
    gotoxy(48, 12);  
    for (int i = 3; i > 0; i--) {  
        set_text_color(11); // 밝은 파란색 텍스트  
        printf("%d", i);  
        Sleep(1000);  
        gotoxy(48, 12);  
        printf(" ");  
    }  
    system("cls");  
    gotoxy(48, 12);  
    set_text_color(10); // 밝은 초록색 텍스트  
    printf("시작!\n");  
    gotoxy(38, 15);  
    printf("스페이스바를 연타해주세요!!");  
    Sleep(1000);  
    set_text_color(7);  
    system("cls");  
}
```



3|

2|

1|

시작!

스페이스바를 연타해주세요!!|



추가 기능 : 비프음

```
printf("시 작 !\n");  
Beep(750, 500); // 시작 문구와 함께 beep을 추가
```



시작!

스페이스바를 연타해주세요!!

추가 기능 : UI

```
void display_line(int position, char team_name[]) {
    int line[LINE_LENGTH] = {0};
    line[CENTER] = 1; // 줄 기호점
    line[USER_START] = 2; // 유저 팀 위치
    line[COMP_START] = 3; // 컴퓨터 팀 위치

    gotoxy(20, 10);
    printf("[User Team]");
    gotoxy(70, 10);
    printf("[Computer Team]");

    gotoxy(30, 11);
    for (int i = 0; i < LINE_LENGTH; i++) {
        if (i == position) {
            printf("|"); // 현재 줄의 위치 표시
        } else if (line[i] == 1) {
            printf("■"); // 중앙 기호점
        } else if (line[i] == 2) {
            printf("*"); // 유저 팀
        } else if (line[i] == 3) {
            printf("*"); // 컴퓨터 팀
        } else {
            printf("-"); // 빈 줄
        }
    }
}

void display_start(char team_name[], int wins, int losses, int round) {
    gotoxy(0, 0);
    printf("=====\\n");
    gotoxy(42, 1);
    printf("Tug of War Game\\n");
    gotoxy(0, 2);
    printf("=====\\n");
    gotoxy(20, 3);
    printf("%s팀 승리 : %d", team_name, wins);
    gotoxy(70, 3);
    printf("컴퓨터팀 승리 : %d", losses);
    gotoxy(45, 4);
    printf("Round %d\\n", round);
    gotoxy(0, 5);
    printf("=====\\n");

    display_line(CENTER, team_name);

    gotoxy(35, 22);
    set_text_color(12);
    printf("스페이스바를 눌러 줄을 당기세요.");
    set_text_color(7);
}

void make_decision(int position, int *wins, int *losses, char team_name[], int time_out) {
    clear_line(14);
    clear_line(15);

    if (position <= 0) { // 유저 승리
        (*wins)++;
        gotoxy(35, 14);
        printf("%s팀이 이겼습니다!", team_name);
    } else if (position >= LINE_LENGTH - 1 || time_out) { // 컴퓨터 승리
        (*losses)++;
        if (time_out) {
            gotoxy(35, 14);
            printf("시간 초과! 컴퓨터팀이 이겼습니다.");
        } else {
            gotoxy(35, 14);
            printf("컴퓨터팀이 이겼습니다.");
        }
    } else {
        gotoxy(35, 14);
        printf("비겼습니다.");
    }

    // 게임이 끝나지 않았을 때만
    if (*wins < 2 && *losses < 2) {
        gotoxy(35, 16);
        printf("계속하려면 아무 키나 누르세요.");
        getch();
        Sleep(1300); // 다음 라운드에 들어가기 전에 1초 지연
    }

    // 게임이 끝났을 때는 바로 결과 화면으로
}
```



```
Tug of War Game
제한시간 : 14.0초
컴프팀 승리: 0                      Round 1                      컴퓨터팀 승리: 0

[User Team]          [Computer Team]
-----*-----■-----*-----|-----

스페이스바를 눌러 줄을 당기세요.
```

```
Tug of War Game
컴프팀 승리: 1                      Round 2                      컴퓨터팀 승리: 0

[User Team]          [Computer Team]
-----*-----|-----*-----

스페이스바를 눌러 줄을 당기세요.]
```

추가 기능 : 승패 연출

```
void display_result_screen(int user_won) {
    system("cls");
    if (user_won) {
        set_text_color(10);
        gotoxy(38, 12);
        printf(" _ _ _ _ _");
        gotoxy(38, 13);
        printf(" \\ \\ // \\ \\ / ( ) | |");
        gotoxy(38, 14);
        printf(" \\ \\_/_ _ _ \\ \\_/_/_/_ | |");
        gotoxy(38, 15);
        printf(" \\ /_\\_||| \\ \\_\\_|||'_\\_||");
        gotoxy(38, 16);
        printf(" | | ( | | | \\ \\_/_/_||| | |");
        gotoxy(38, 17);
        printf(" | | \\_\\_||| \\ \\_\\_||| | |");
        gotoxy(57, 18);
        printf(" \\ \\_\\_||| \\ \\_\\_||| | |");
        gotoxy(57, 19);
        printf("You Win!");
    }
    else {
        set_text_color(12);
        gotoxy(35, 12);
        printf(" _ _ _ _ _");
        gotoxy(35, 13);
        printf(" /_\\_|||_/_/_/_/_/_/_/_/_/_");
        gotoxy(35, 14);
        printf(" | |_/_/_/_/_/_/_/_/_/_/_/_");
        gotoxy(35, 15);
        printf(" | | ( | | | | | | | | | | | |");
        gotoxy(35, 16);
        printf(" \\ \\_\\_||| \\ \\_\\_||| \\ \\_\\_|||");
        gotoxy(57, 18);
        printf("Game Over!");
    }
    set_text_color(7);
}
```



```

You Win!
```

```

Game Over!
```



추가 기능 : 재시작

```
// 결과 화면 표시
display_result_screen(score_wins == 2);

printf("\n게임을 다시 시작하려면 'y' 키를 누르고, 종료하려면 'n' 키를 누르세요.\n");

while (1) {
    int key = _getch();
    if (key == 'n' || key == 'N') { // 'n' 키
        return 0;
    } else if (key == 'y' || key == 'Y') { // 'y' 키
        break; // 게임 재시작
    }
}

return 0;
}
```



You Win!

게임을 다시 시작하려면 'y' 키를 누르고, 종료하려면 'n' 키를 누르세요.

Game Over!

게임을 다시 시작하려면 'y' 키를 누르고, 종료하려면 'n' 키를 누르세요.



팀 이름을 입력하세요 :



점수

10점

THANK YOU

