



# 줄다리기 게임

2021663028 박도현



# 8\_10\_1

줄다리기 게임

3번 싸워서 2번 먼저 이긴 팀이 승자입니다.

참가자는 두 팀입니다.

1번 참가팀의 이름을 입력하고 Enter>청

2번 참가팀의 이름을 입력하고 Enter>백

줄다리기 게임의 제한 시간(초 단위)을 입력하고 Enter>10|

제한시간 : 10.0 초

경과시간 : 3.4초

청  
0승, 0패  
□□□□◆□□□□■□□□□◆□□□□

기준점  
▼  
▲

백  
0승, 0패

청팀이 이겼습니다.  
아무키나 누르세요. |

아쉬운 점

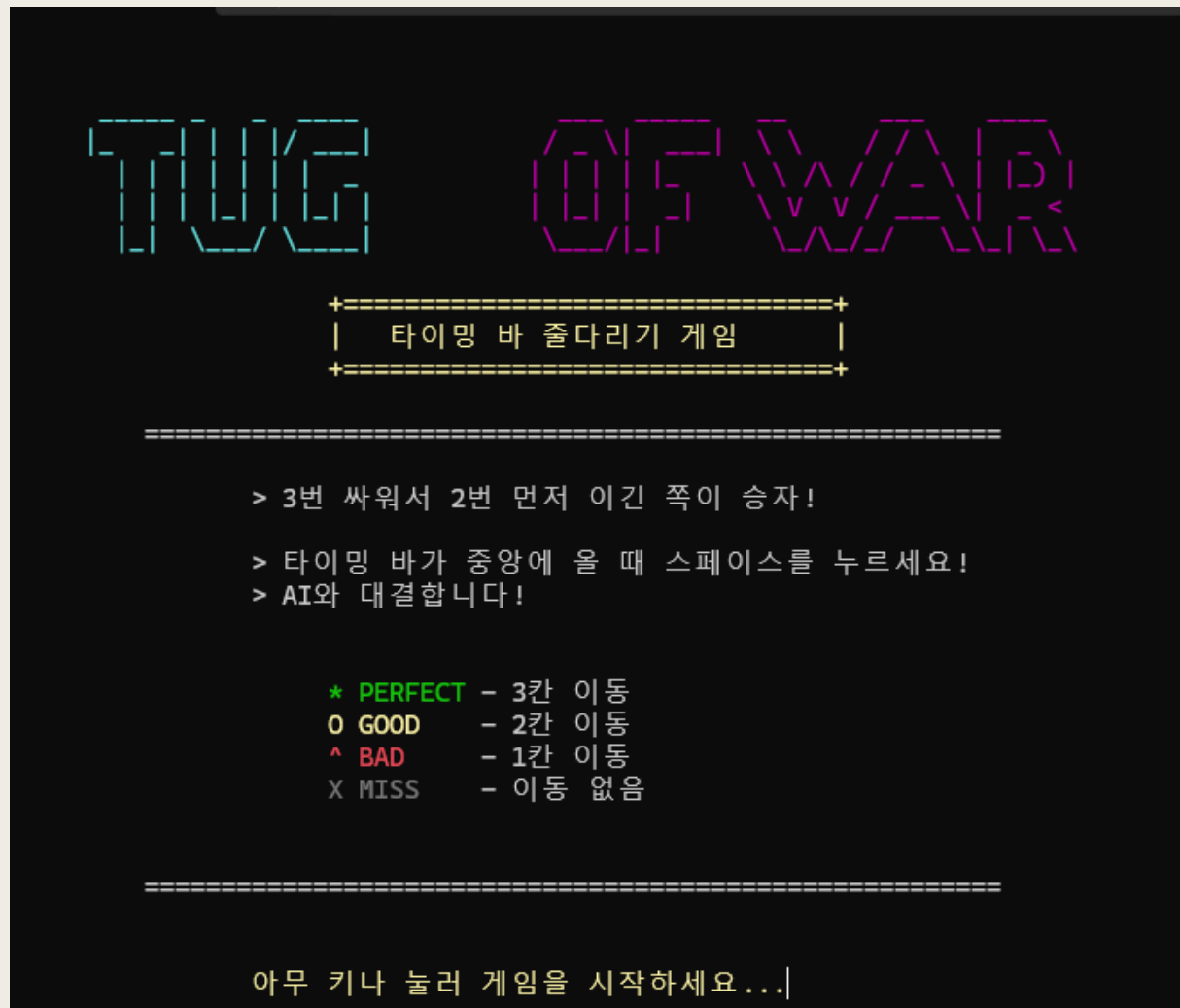
시각적 단조로움

참여도 낮음

몰입감 부족

# UpGrade(인트로)

## 개선된 점



컬러와 로고를 통해 시각적 주목도 상승

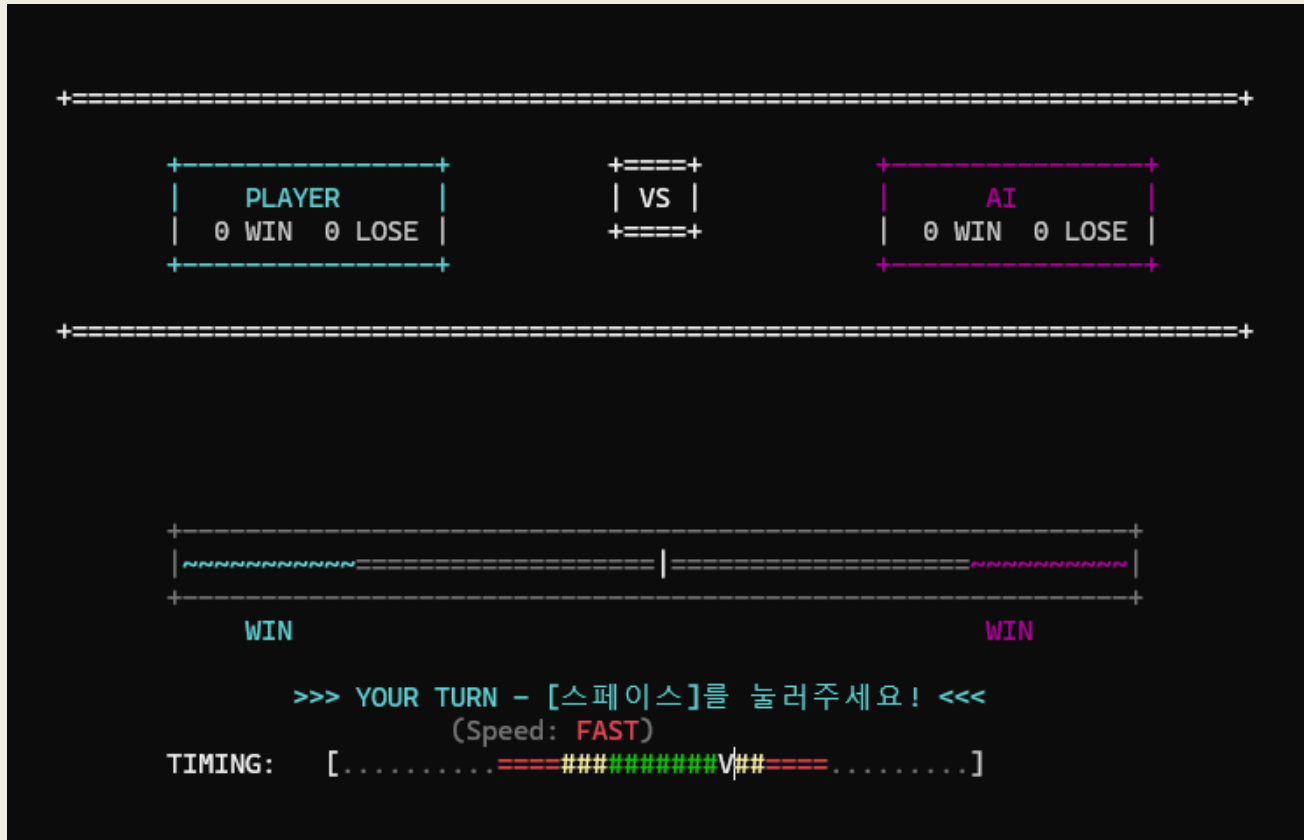
게임의 목적과 조작법 이해

# UpGrade(게임)

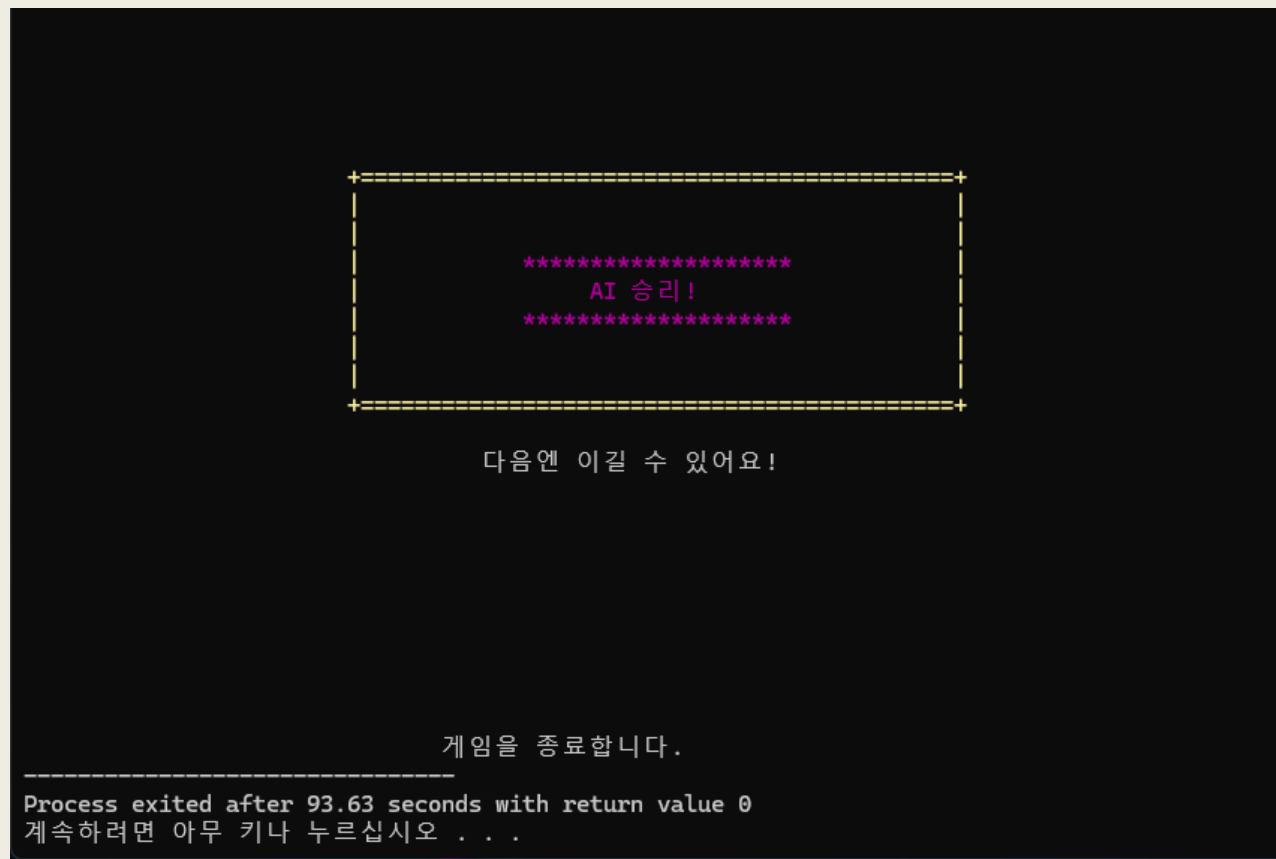
## 개선된 점

참여형 게임 구조

UI를 통한 직관성 강화



# Upgrade(승리)



개선된 점

시각적 완결감

플레이어의 여운 제공

성취감 및 재도전 의욕 유도

# 정리

항목	업그레이드 전	업그레이드 후
게임 방식	시간 제한 자동 진행	턴제 타이밍 게임
플레이어 역할	관전만 가능	직접 참여 (스페이스바)
상대	팀 이름 입력	AI 대전
난이도	고정	3단계 속도 (SLOW/NORMAL/FAST)
UI/UX	단순 텍스트	컬러 구분 + 타이밍 바

# 핵심 코드

- 턴 방식

- 타이밍 바

- AI

# 턴 방식

```
int main(void)
{
    ...

    int turn_count = 0;

    ...

    do
    {
        system("cls");
        draw_border();
        r_start = 20;
        turn_count = 0;

        display_score_board(score_win, score_loose, r_start);

        gotoxy(35, 32);
        set_color(14);
        printf("아무 키나 눌러 경기 시작!");
        set_color(COLOR_RESET);
        getch();
        clear_line(32);

        while((WIN_POSITION_LEFT < r_start) && (r_start < WIN_POSITION_RIGHT))
        {
            game_control(&r_start, turn_count);
            turn_count++;
            display_score_board(score_win, score_loose, r_start);
        }

        make_decision(r_start, score_win, score_loose);

    }while((score_win[0]<2) && (score_win[1]<2));

    ...
}
```

```
void game_control(int *r_s, int turn_count)
{
    TimingBar bar;
    int current_turn = turn_count % 2; // 0: Player, 1: AI
    int timing_score, power;
    int center = TIMING_BAR_LENGTH / 2;

    init_timing_bar(&bar);

    if(current_turn == 0) {
        // 플레이어 턴
        printf(">>> YOUR TURN <<<");
        // 타이밍 바 표시
        display_timing_bar(&bar, center);
        // 입력 감지
        ...
    } else {
        // AI 턴
        printf(">>> AI TURN <<<");
        timing_score = ai_play();
        ...
    }

    // 타이밍 결과에 따라 rope 이동
    if(current_turn == 0)
        *r_s -= power;
    else
        *r_s += power;
}
```



# 타이밍 바

```
void display_timing_bar(TimingBar *bar, int center)
{
    printf("TIMING:");
    for(int i=0; i<TIMING_BAR_LENGTH; i++) {
        int dist = abs(i - center);
        if(dist <= PERFECT_ZONE) printf("#");    // PERFECT
        else if(dist <= GOOD_ZONE) printf("#");  // GOOD
        else if(dist <= BAD_ZONE) printf("=");   // BAD
        else printf(".");
    }
    printf("]");
    gotoxy(30 + 1 + bar->position, 23);
    printf("V"); // 커서
}
```

# A I

```
int ai_play()  
{  
    int random = rand() % 100;  
    if(random < 25)      return 3; // PERFECT  
    else if(random < 60) return 2; // GOOD  
    else if(random < 90) return 1; // BAD  
    else                 return 0; // MISS  
}
```

감사합니다