

# 단위 변환 프로그램 업그레이드

송채원

2023964054

깃허브

[gameprogramming/report/Proj1 at main · ssongchaewon/gameprogramming](#)

# 기본 프로그램

게임의 핵심 C 코드로, 사용자 입력에 따라 단위 변환을 수행하며 인터랙티브한 쿼즈 게임의 기반이 됩니다.

이 `measure_calculate` 함수는 다양한 길이 단위를 상호 변환하며, 게임 내 모든 단위 변환 계산의 기초 역할을 합니다.

```
길이에 대한 도량형 환산 프로그램

0:센티미터   1:미터   2:킬로미터
3:인치       4:피트   5:야드
6:마일       7:자(척) 8:간
9:리

기본단위와 변환단위 입력 후 Enter (예 : 1 5) > 1 5
값을 입력하고 Enter > 100

100.00      미터는      109.36133      야드
```

```
#include <stdio.h>
char *ms_char[10]={"센티미터", "미터", "킬로미터", "인치", "피트",
                  ,"야드", "마일", "자(척)", "간", "리"};
void print_distance_unit(void);
void measure_calculate(int basic, int transe, double measure);

> int main(void){ ... }

void print_distance_unit(void)
{
    int i;
    for(i=0; i<=9; i++)
    {
        printf("%d:%-8s ", i, ms_char[i]);
        if (i%3==2)
            printf("\n");
    }
}

void measure_calculate(int basic, int transe, double measure)
{
    //[함수 11.4.1]의 정의 부분 참고
    double msre[10], unit, result;

    msre[0]=392772; //centimeter
    msre[1]=msre[0]/100.; //meter
    msre[2]=msre[1]/1000.; //kilometer
    msre[3]=msre[0]/2.54; //inch
    msre[4]=msre[3]/12.; //feet
    msre[5]=msre[4]/3.; //yard
    msre[6]=msre[5]/1760.; //mile
    msre[7]=12960.; //자
    msre[8]=msre[7]/6.; //간
    msre[9]=msre[8]/(6.*36); //리

    unit=msre[basic];
    result=measure*(msre[transe]/unit);
    printf("%.2f %8s는 ", measure, ms_char[basic]);
    printf("%15.5f %8s\n", result, ms_char[transe]);
}
```

# 인트로 화면 (intro\_screen 함수)

## 게임 시작 전 시각적 효과

인트로 화면은 게임 시작 전 사용자에게 시각적 흥미를 제공합니다.

### ASCII 아트 제목

게임 제목을 ASCII 아트로 크게 표시하여 콘솔 게임 분위기를 연출합니다.

### 로딩 애니메이션

점(.)을 찍는 애니메이션으로 로딩 중임을 표시합니다. (색상 코드 14, 노랑)

### 시작 안내

"Enter를 눌러 시작하세요..." 메시지로 게임 진입을 유도합니다.



```
void Intro_screen(void) {
    system("cls");
    system("title 단위 계산기 - 로그라이크 게임");
    SetConsoleTextAttribute(hConsole, 11);

    printf("      ----                               \n");
    printf("    / ---|  _ _ _ _ _ _ _ _ _ _ _ _ _ _ \n");
    printf("  | | _ / _ | ' _ ' _ _ _ _ _ _ _ _ _ _ \n");
    printf("  | | | ( _ | | | | | | | | | | _ / _ _ \n");
    printf("  _ _ _ _ | _ _ _ _ _ | | | | | | _ _ _ _ _ \n");
    printf("                                UNIT CONVERTER ROGUELIKE GAME \n");
    printf("-----\n");

    SetConsoleTextAttribute(hConsole, 14);
    printf("로딩 중 ");
    for (int i = 0; i < 5; i++) {
        printf(".");
        fflush(stdout);
        Sleep(300);
    }

    SetConsoleTextAttribute(hConsole, 7);
    printf("\n\nEnter를 눌러 시작하세요...");
    (void)getchar(); // 경고 제거
    system("cls");
}
```

# 메뉴 화면 (menu 함수)

사용자가 게임의 주요 기능을 선택할 수 있는 중앙 허브입니다. 선택에 따라 `start_game()`, `show_unit_info()`, `outro_screen()` 함수로 분기됩니다.

1

## 1. 게임 시작

새로운 퀴즈 세션을 시작하고 점수와 레벨을 초기화합니다.

2

## 2. 단위 비율 정보

주요 단위 간의 변환 비율 정보를 테이블 형태로 보여줍니다.

3

## 3. 종료

게임을 종료하고 아웃트로 화면으로 이동합니다.

□ 강조 포인트: 메뉴 UI는 테두리(청색)와 옵션(노랑)으로 색상 강조됩니다.

```
void menu(void) {
    int choice;
    do {
        system("cls");
        SetConsoleTextAttribute(hConsole, 11);
        printf("+-----+~n");
        printf("|          UNIT GAME MENU          |~n");
        printf("+-----+~n");
        SetConsoleTextAttribute(hConsole, 14);
        printf("| 1. 게임 시작                    |~n");
        printf("| 2. 단위 비율 정보              |~n");
        printf("| 3. 종료                        |~n");
        SetConsoleTextAttribute(hConsole, 7);
        printf("+-----+~n");
        printf("선택 > ");
        (void)scanf("%d", &choice);

        switch (choice) {
            case 1: start_game(); break;
            case 2: show_unit_info(); break;
            case 3: outro_screen(); break;
            default:
                printf("잘못된 입력입니다.~n");
                Sleep(800);
        }
    } while (choice != 3);
}
```

```
+-----+
|          UNIT GAME MENU          |
+-----+
| 1. 게임 시작                    |
| 2. 단위 비율 정보              |
| 3. 종료                        |
+-----+
선택 > |
```

# 게임 진행: 문제 출제 및 진행도

## 문제 출제 (ask\_question)

- 랜덤 단위 (rand() % 6)와 값 (rand() % 1000 / 10.0 + 1)을 조합하여 문제를 생성합니다.
- 정답 입력 후 calc 함수로 실제 정답을 계산하여 오차 범위 내 (fabs < 1.0)에서 정답을 판별합니다.
- 정답 시 +10점, 오답 시 -5점 처리 및 정답 공개.

## 진행도 표시 (show\_progress)

최대 9개의 문제 중 현재 진행 상황을 진행도 바 (# 또는 -)로 시각화합니다.

```
+-----+
| 레벨:1  점수:0 |
| 진행도:-----|
+-----+

문제 1: 21.00 야드를 미터로 변환하세요
정답 입력 > 1
오답! 정답은 19.20240 미터 입니다. -5점
```

### 레벨

현재 레벨을 청색으로 표시합니다.

### 점수

현재 점수를 노란색으로 표시합니다.

### 진행도

퀴즈 진행 상황을 초록색 바(#)로 표시합니다.

```
+-----+
| 레벨:1  점수:-30|
| 진행도:#####---|
+-----+

문제 7: 38.40 센티미터를 피트로 변환하세요
정답 입력 > 2
정답! +10점
```

# 핵심 함수 구현 상세

본 섹션에서는 게임의 핵심적인 기능을 담당하는 코드의 구현 사항을 자세히 설명합니다.

## 진행도 표시 코드 (show\_progress)

이 코드는 게임의 현재 진행 상황을 시각적으로 나타내는 로직을 담당합니다.

```
void show_progress(int question_num) {
    char progress[MAX_QUESTIONS + 1];
    for (int i = 0; i < MAX_QUESTIONS; i++)
        progress[i] = (i < question_num) ? '#' : '-';
    progress[MAX_QUESTIONS] = '\0';

    SetConsoleTextAttribute(hConsole, 7);
    printf("+-----+\\n");
    SetConsoleTextAttribute(hConsole, 9);
    printf("I 레벨: %-2d ", level);
    SetConsoleTextAttribute(hConsole, 14);
    printf("점수: %-3d\\n", score);
    SetConsoleTextAttribute(hConsole, 10);
    printf("I 진행도: %-9s\\n", progress);
    SetConsoleTextAttribute(hConsole, 7);
    printf("+-----+\\n\\n");
}
```

## 문제 출제 코드 (ask\_question)

이 코드는 난수를 사용하여 문제를 생성하고 정답 여부를 판별하는 로직을 구현합니다.

```
void ask_question(int qnum) {
    int basic = rand() % 6, transe;
    do { transe = rand() % 6; } while (transe == basic);

    double measure = (rand() % 1000) / 10.0 + 1;
    show_progress(qnum);

    printf("문제 %d: %.2f %s를 %s로 변환하세요\\n", qnum + 1, measure, ms_char[basic], ms_char[transe]);
    printf("정답 입력 > ");
    double answer;
    (void)scanf("%lf", &answer);

    double real_answer = calc(basic, transe, measure);

    if (fabs(real_answer - answer) < 1.0) {
        printf("정답! +10점\\n\\n");
        score += 10;
    }
    else {
        printf("오답! 정답은 %.5f %s 입니다. -5점\\n\\n", real_answer, ms_char[transe]);
        score -= 5;
    }

    Sleep(500);
}
```

# 단위 변환 비율 화면 (show\_unit\_info 함수)

사용자가 게임에 사용된 단위 간의 정확한 변환 비율을 확인할 수 있는 참고 자료입니다. ratio(basic, transe) 함수를 사용하여 비율을 계산합니다.

## 비율 계산

10가지 단위 중 6가지 주요 단위(0~5 인덱스) 간의 모든 변환 비율을 출력합니다.

## 테이블 구조

각 단위 쌍에 대해 1 [단위 A] = [비율] [단위 B] 형식으로 명확하게 표시합니다.

## 메뉴 복귀

정보 확인 후 Enter 키를 눌러 메인 메뉴로 쉽게 돌아갈 수 있습니다.

```
void show_unit_info(void) {
    system("cls");
    printf("+----- 단위 변환 비율 -----+\n");
    for (int i = 0; i < 6; i++) {
        for (int j = 0; j < 6; j++) {
            if (i != j)
                printf("! %-8s = %10.5f %-8s\n", ms_char[i], ratio(i, j), ms_char[j]);
        }
        printf("-----\n");
    }
    printf("\nEnter를 눌러 메뉴로 돌아갑니다...\n");
    (void)getchar(); (void)getchar();
}
```

```
1 미터      = 39.37008 인치
1 미터      = 3.28084 피트
1 미터      = 1.09361 야드
```

```
1 킬로미터 = 100000.00000 센티미터
1 킬로미터 = 1000.00000 미터
1 킬로미터 = 39370.07874 인치
1 킬로미터 = 3280.83990 피트
1 킬로미터 = 1093.61330 야드
```

```
1 인치      = 2.54000 센티미터
1 인치      = 0.02540 미터
1 인치      = 0.00003 킬로미터
1 인치      = 0.08333 피트
1 인치      = 0.02778 야드
```

```
1 피트      = 30.48000 센티미터
1 피트      = 0.30480 미터
1 피트      = 0.00030 킬로미터
1 피트      = 12.00000 인치
1 피트      = 0.33333 야드
```

```
1 야드      = 91.44000 센티미터
1 야드      = 0.91440 미터
1 야드      = 0.00091 킬로미터
1 야드      = 36.00000 인치
1 야드      = 3.00000 피트
```

```
Enter를 눌러 메뉴로 돌아갑니다...|
```



# 아웃트로 화면 (outro\_screen 함수)

게임 세션이 종료된 후 사용자에게 감사를 표하고 프로그램을 종료하는 마지막 화면입니다.

## "Thank you for playing!"

청색 테두리와 노란색 텍스트로 강조된 감사 메시지를 출력합니다.

사용자에게 Enter 키를 눌러 프로그램을 종료하도록 안내합니다.

```
void outro_screen(void) {  
    system("cls");  
    SetConsoleTextAttribute(hConsole, 11);  
    printf("+-----+~n");  
    printf("|                               |~n");  
    SetConsoleTextAttribute(hConsole, 14);  
    printf("| Thank you for playing!       |~n");  
    SetConsoleTextAttribute(hConsole, 11);  
    printf("|                               |~n");  
    printf("+-----+~n~n");  
    SetConsoleTextAttribute(hConsole, 7);  
    printf("엔터 키를 눌러 종료하세요...");  
    (void)getchar(); (void)getchar();  
}
```

Thank you for playing!



# Reference



## ChatGPT (챗GPT)

아이디어 구상, 내용 생성, 그리고 복잡한 기술적 설명을 명확하고 간결하게 다듬는 데 활용되어 콘텐츠 제작의 효율성을 극대화했습니다.



## Gamma (감마)

반응형 디자인과 시각적 스토리텔링 기능을 통해 콘솔 게임 프로젝트의 복잡한 구조와 코드 로직을 매력적이고 이해하기 쉬운 프레젠테이션으로 완성했습니다.