

# 게임프로그래밍 (바이오리듬)

2021963016 김용효

# 원본 코드의 핵심 로직

c



// 원본의 핵심

**total\_days()** - 날짜를 일수로 변환

**bio\_status()** - 사인 함수로 리듬 상태 계산 (+, -, D)

**print\_biorhythm()** - 결과를 텍스트로 출력

생년월일을 입력받아 한 달간의 바이ורי듬을 텍스트로 출력하는 간단한 프로그램

# 게임설명

바이오리듬기반 턴제 rpg 게임

생년월일을 바탕으로 현재 바이오리듬을 계산하고  
바이오리듬을 전략적으로 변화하면서  
그에 따라 캐릭터의 능력치가 변화하는 턴제 rpg 게임

# 바이올리듬이란?

바이올리듬은 인간의 신체적, 감정적, 지적 상태가 출생일을 기준으로 해서 일정한 주기로 변동한다는 가설적 개념

변동되는 상태로는 최고조상태(+), 임계점(\*), 최저상태(-)가 있음

- 각 주기 계산
- 신체 리듬 23일 주기
- 감정 리듬 28일 주기
- 지성 리듬 33일 주기

(+ 상태) = 신체적·정신적 컨디션 상승

(\* 상태) = 주의 필요한 기간

(- 상태) = 신체적·정신적 컨디션 저하

# calculate\_biorhythm

현재 날짜 기준으로 바이ोर리듬을 계산

```
c
void calculate_biorhythm(int byr, int bmon, int bday, Biorhythm *bio) {
    time_t t = time(NULL);
    struct tm tm = *localtime(&t);

    // 오늘 날짜와 생년월일의 차이 계산
    long total = total_days(tm.tm_year + 1900, tm.tm_mon + 1, tm.tm_mday) -
        total_days(byr, bmon, bday);

    // 각 주기로 남은 나머지 계산
    bio->physical = total % 23;    // 신체 주기
    bio->emotion = total % 28;    // 감정 주기
    bio->intellect = total % 33;  // 지성 주기

    // 각 리듬의 상태 판정
    bio->phys_status = bio_status(bio->physical, 23);
    bio->emo_status = bio_status(bio->emotion, 28);
    bio->intel_status = bio_status(bio->intellect, 33);

    // 생년월일 저장
    bio->birth_year = byr;
    bio->birth_month = bmon;
    bio->birth_day = bday;
}
```

현재 날짜 가져오기

```
c
time_t t = time(NULL);
struct tm tm = *localtime(&t);
```

일수 차이 계산

```
c
long total = 오늘 날짜 - 생년월일
```

각 주기 계산

신체 리듬: total%23(23일 주기)

감정 리듬: total%28(28일 주기)

지성 리듬: total%33(33일 주기)

# apply\_biorhythm\_to\_character

바이오리듬에 따라 캐릭터 스탯을 동적으로 조정

```
// 기본 스탯 설정
player->max_hp = 100;
player->max_mp = 50;
player->attack = 15;
player->magic_attack = 12;
player->defense = 10;
player->magic_defense = 8;
player->crit_chance = 5;
player->stamina = 100;

// 신체 리듬 보너스
if (bio->phys_status == '+') {
    player->max_hp += 30; // HP +30
    player->attack += 8; // 공격력 +8
}
else if (bio->phys_status == '*') {
    player->max_hp += 10; // HP +10
}
else if (bio->phys_status == '-') {
    player->attack -= 5; // 공격력 -5
}

// 지적 리듬 보너스
if (bio->intel_status == '+') {
    player->max_mp += 30; // MP +30
    player->magic_attack += 8; // 마법공격력 +8
}
else if (bio->intel_status == '*') {
    player->max_mp += 10; // MP +10
}
else if (bio->intel_status == '-') {
    player->magic_attack -= 5; // 마법공격력 -5
}

// 감정 리듬 보너스
if (bio->emo_status == '+') {
    player->crit_chance += 30; // 치명타 확률 +30%
}
else if (bio->emo_status == '*') {
    player->crit_chance += 10; // 치명타 확률 +10%
}
else if (bio->emo_status == '-') {
    player->crit_chance = 0; // 치명타 확률 0%
}

// HP/MP 최대치 초과 방지
if (player->hp > player->max_hp) player->hp = player->max_hp;
```

리듬	상태	보너스효과
신체	+	HP+30, 공격력+8
신체	*	HP+10
신체	-	공격력-5
지적	+	MP+30,마공+8
지적	*	MP+10
지적	-	마공-5
감정	+	치명타+30%
감정	*	치명타+10%
감정	-	치명타0%

# change\_rhythm

```
void change_rhythm(Character *player, Biorythm *bio) {  
    // MP 체크  
    if(player->mp < 10) {  
        printf("MP가 부족합니다!");  
        return;  
    }  
  
    // 생년월일 변경  
    printf("1. 리듬 변경 (생년월일 조정)\n");  
    printf("2. 바이오리듬 계산기\n");  
    printf("3. 취소\n");  
  
    int choice;  
    scanf("%d", &choice);  
  
    if(choice == 2) {  
        rhythm_calculator(bio);  
        return;  
    } else if(choice == 3) {  
        return;  
    }  
  
    // 일수 입력 받기  
    printf("연감할 일수 (1~3, 음수는 과거로): ");  
    int days;  
    scanf("%d", &days);  
  
    if(days < -3 || days > 3 || days == 0) {  
        printf("잘못된 입력!");  
        return;  
    }  
  
    // MP 소모  
    player->mp -= 10;  
  
    // 생년월일 조정  
    bio->birth_day += days;
```

```
// 바이오리듬 재계산  
calculate_biorhythm(bio->birth_year, bio->birth_month, bio->birth_day, bio)  
  
// 스코어 재계산 (HP/MP 비율 유지)  
int hp_ratio = (player->hp * 100) / player->max_hp;  
int mp_ratio = (player->mp * 100) / player->max_mp;  
  
apply_biorhythm_to_character(player, bio);  
  
player->hp = (player->max_hp * hp_ratio) / 100;  
player->mp = (player->max_mp * mp_ratio) / 100;  
  
// 결과 출력  
// ...  
}
```

생년월일을 조정하여 바이오리듬을 변경하는 스킬

일수 조정: -3 ~ +3일 범위 내에서 조정 가능

# battle

```
int battle(Character *player, Monster *enemy, Biorythm *bio, int stage) {
    int selected = 0; // 선택된 스킬 인덱스

    while(player->hp > 0 && enemy->hp > 0) {
        draw_ui(player, enemy, bio, stage);
        draw_skill_menu(selected, player);

        char key = _getch(); // 키 입력 대기

        // 키 입력 처리
        if(key == 'c' || key == 'C') {
            rhythm_calculator(bio); // 바이오리듬 계산기
            continue;
        }
        else if(key == 'x' || key == 'X') {
            // 제한 옵션 확인
            printf("개임을 종료하시겠습니까? (Y/N)");
            char confirm = _getch();
            if(confirm == 'Y' || confirm == 'y') {
                exit(0);
            }
        }
        else if(key == 72) { // 위 방향키
            selected = (selected - 2 + 4) % 4;
        }
        else if(key == 80) { // 아래 방향키
            selected = (selected + 2) % 4;
        }
        else if(key == 75) { // 왼쪽 방향키
            if(selected % 2 == 1) selected--;
        }
        else if(key == 77) { // 오른쪽 방향키
            if(selected % 2 == 0) selected++;
        }
        else if(key == 'z' || key == 'Z') {
            // 스킬 실행
            if(selected == 1) { // 리듬 변경
                change_rhythm(player, bio);
            }
        }
    }
}
```

```
if(enemy->hp > 0) {
    int enemy_damage = enemy->attack - player->defense / 2;
    if(enemy_damage < 1) enemy_damage = 1;
    player->hp -= enemy_damage;
    printf("%s의 반격! %d 데미지!", enemy->name, enemy_damage);
}
continue;
}

// 공격 데미지 계산
int damage = 0;
int actual_def = enemy->defense;
int actual_mdef = enemy->magic_defense;

// 보스의 숨겨진 방어력
if(enemy->is_boss) {
    if(enemy->hidden_type == 0) {
        actual_def = 30;
        actual_mdef = 8;
    } else {
        actual_def = 8;
        actual_mdef = 30;
    }
}

switch(selected) {
    case 0: // 물리 공격
        damage = player->attack - actual_def / 2;
        if(damage < 1) damage = 1;

        // 크리스탈 공격
        if(rand() % 100 < player->crit_chance) {
            damage *= 2;
            printf("크리탈힐!");
        }
        enemy->hp -= damage;
        break;

    case 2: // 마법 공격
        damage = player->magic_attack - actual_mdef / 2;
        if(damage < 1) damage = 1;
        enemy->hp -= damage;
        break;
}
```

```
case 3: // 회복
    if(player->hp >= 20) {
        player->hp += 30;
        if(player->hp > player->max_hp)
            player->hp = player->max_hp;
    } else {
        printf("MP가 부족합니다!");
    }
    break;
}

// 적 공격 (회피 30)
if(enemy->hp > 0 && selected != 3) {
    int enemy_damage = enemy->attack - player->defense / 2;
    if(enemy_damage < 1) enemy_damage = 1;
    player->hp -= enemy_damage;
}

// 전투 결과
if(player->hp <= 0) {
    return 0; // 패배
} else {
    // 보스 인전 공격
    if(enemy->is_boss) {
        if(enemy->hidden_type == 0) {
            printf("보스의 경재: 물리킬어칠 (DEF 30, M.DEF 8)");
        } else {
            printf("보스의 경재: 마법킬어칠 (DEF 8, M.DEF 30)");
        }
    }
    return 1; // 승리
}
}
```



# 인트로 화면

```
# #####
#                                     ??????????????????????????
#                                     ?   바이오리듬 RPG 게임   ?
#                                     ??????????????????????????
#
# ■ 게임 소개 :
#     당신의 생년월일을 바탕으로 현재 바이오리듬을 계산하고 ,
#     그에 따라 캐릭터의 능력치가 변화하는 턴제 RPG 게임입니다 .
#
# ■ 바이오리듬 시스템 :
#     신체(+) : 체력 +30, 공격력 +8           신체(-) : 공격력 -5
#     지적(+) : 마나 +30, 마법공격력 +8       지적(-) : 마법공격력 -5
#     감정(+) : 치명타 확률 +30%              감정(-) : 치명타 확률 0%
#
# ■ 조작법 :
#     방향키 : 스킬 선택
#     Z 키 : 선택 / 공격
#     X 키 : 게임 종료
#     C 키 : 확인 / 계속
#
# ■ 팁 :
#     - 물리방어가 높은 적은 마법으로, 마법방어가 높은 적은 물리로 공격하세요 !
#     - '리듬 변경' 스킬로 생년월일을 조정하여 바이오리듬을 바꿀 수 있습니다 .
#
#                                     아무 키나 눌러 시작...|
```

# 실제 바이오 리듬 변환 과정



리듬 변경 스킬 클릭

# 실제 바이오 리듬 변환 과정

```
????????????????????????????????
?   리듬 변경 스킬   ?
????????????????????????????????
```

현재 생년월일: 1990년 5월 29일

■ 설명:  
생년월일을 1~3일 단위로 조정하여 바이오리듬을 변경할 수 있습니다.  
변경된 바이오리듬에 따라 스탯이 즉시 재계산됩니다.

변경할 일수를 입력하세요 (1~3, 음수는 과거로): -3

바뀐 스테이터스

```
HP: [=====]123/130
#####
MP: [=====]40/50
#
#   ★PLAYER★   #
#               #
#               #
#               #
#               #
#               #
#               #
#####
```

ATK:23 M.ATK:7  
DEF:10 M.DEF:8  
CRIT:0%  
신체:~ 감정:- 지성:-

```
#####
#   ■ 변경 완료! (MP -10)   #
#   새로운 생년월일: 1990년 5월 26일   #
#   신체:~ 감정:- 지성:-   #
#   C 키를 눌러 계속...|   #
#####
```

# 성과 및 개선점

## 성과

단순 계산기 출력 프로그램에서 전략적 rpg 게임으로 바꾸어  
서 사용자가 재미를 느낄 수 있는 프로그램으로 변화시킴

## 개선점

rpg라고 하기에는 모험 요소가 아직 적음

모두 +상태일 때 밸런스 깨짐

몬스터 패턴의 단조로움

# 출처

- 챗gpt
- 클로우디 ai

감사합니다

