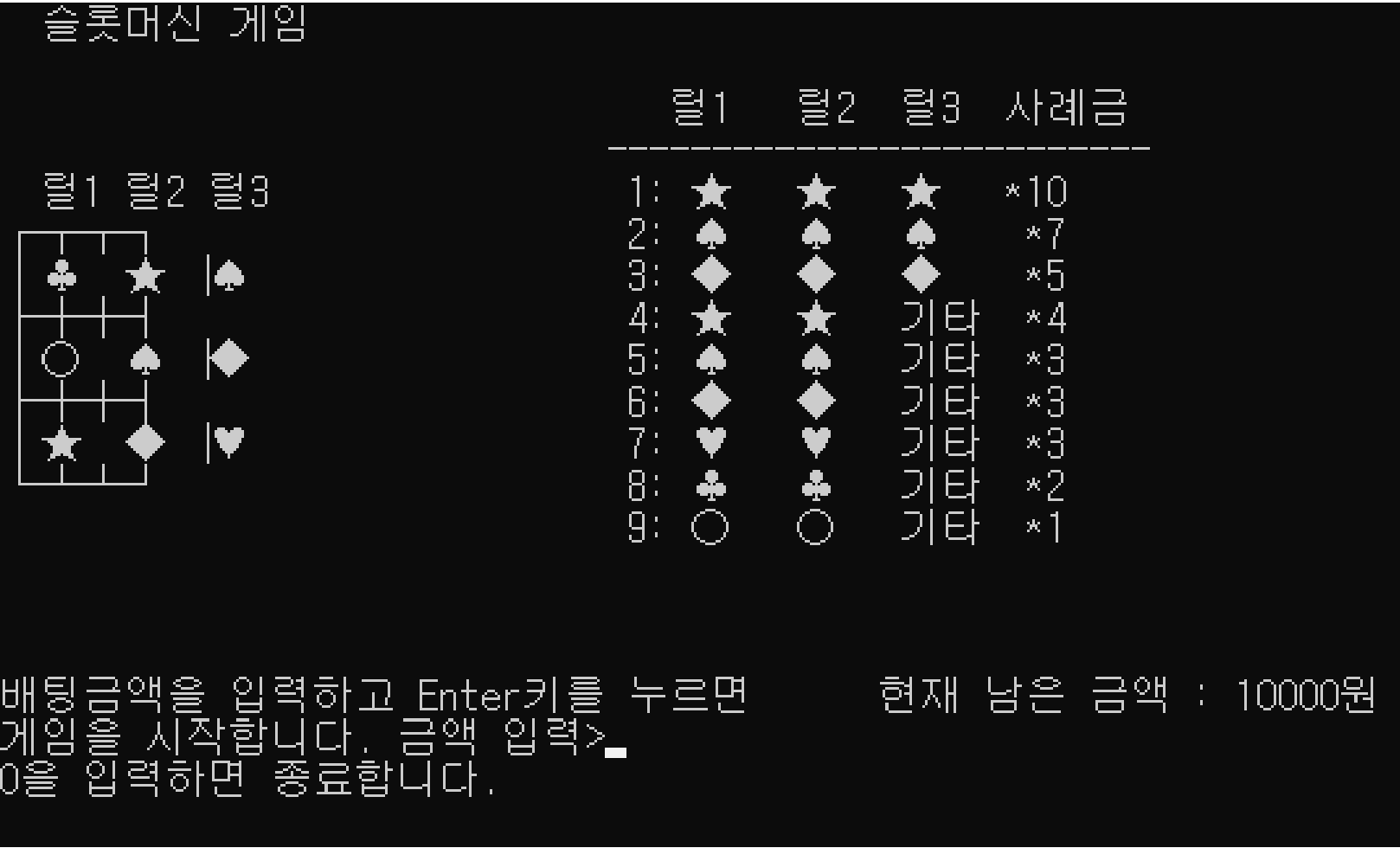

게임프로그래밍

슬롯머신 게임

기존 소스

01



기존 슬롯머신 게임 화면

심플하고 있어야 할 기능들로
잘 갖춰진 기본적인 슬롯머신 프로그램

기존 소스

01

```
int main(void)
{
    char reel[6][3]={"★", "♠", "♦", "♥", "♣", "○"};
    int i, reel_num[3][3];
    int money=10000;
    srand(time(NULL));
    printf(" 슬롯머신 게임\n\n");
    display_rule();
    gotoxy(3, 5);
    printf("릴1 릴2 릴3");
    gotoxy(1,6);
    draw_check02(3, 3);
    for(i=0;i<3;i++)
        reel_num[0][i]=rand()%5;
    reel_series(reel_num);

    do
    {
        game_control(reel, reel_num, &money);
    }while(money);
    gotoxy(3,20);
    return 0;
}
```

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

int game_progress(int money)
{
    int bet;
    gotoxy(43, 17);
    printf("현재 남은 금액 : %d원", money);
    gotoxy(1, 17);
    printf("배팅금액을 입력하고 Enter키를 누르면 ");
    gotoxy(1, 19);
    printf("0을 입력하면 종료합니다.");
    gotoxy(1, 18);
    printf("게임을 시작합니다. 금액 입력>");
    gotoxy(30, 18);
    scanf("%d", &bet);
    if (bet==0)
        exit(0);
    gotoxy(1, 19);
    printf("아무키나 한번만 키를 누르면 릴이 멈춥니다.");
    return bet;
}

void display_reel(char rc[][3], int r[][3], int index)
{
    int i, j;
    for(i=0;i<3;i++)
        for(j=index;j<3;j++)
        {
            gotoxy(3+j*4, 7+i*2);
            printf("%s", rc[r[i][j]]);
        }
}
```

```
void game_control(char reel[][3], int reel_num[][3], int *money)
{
    int i, j, bet, case_num=0, thank, num[3];
    double pst;
    clock_t start, end;

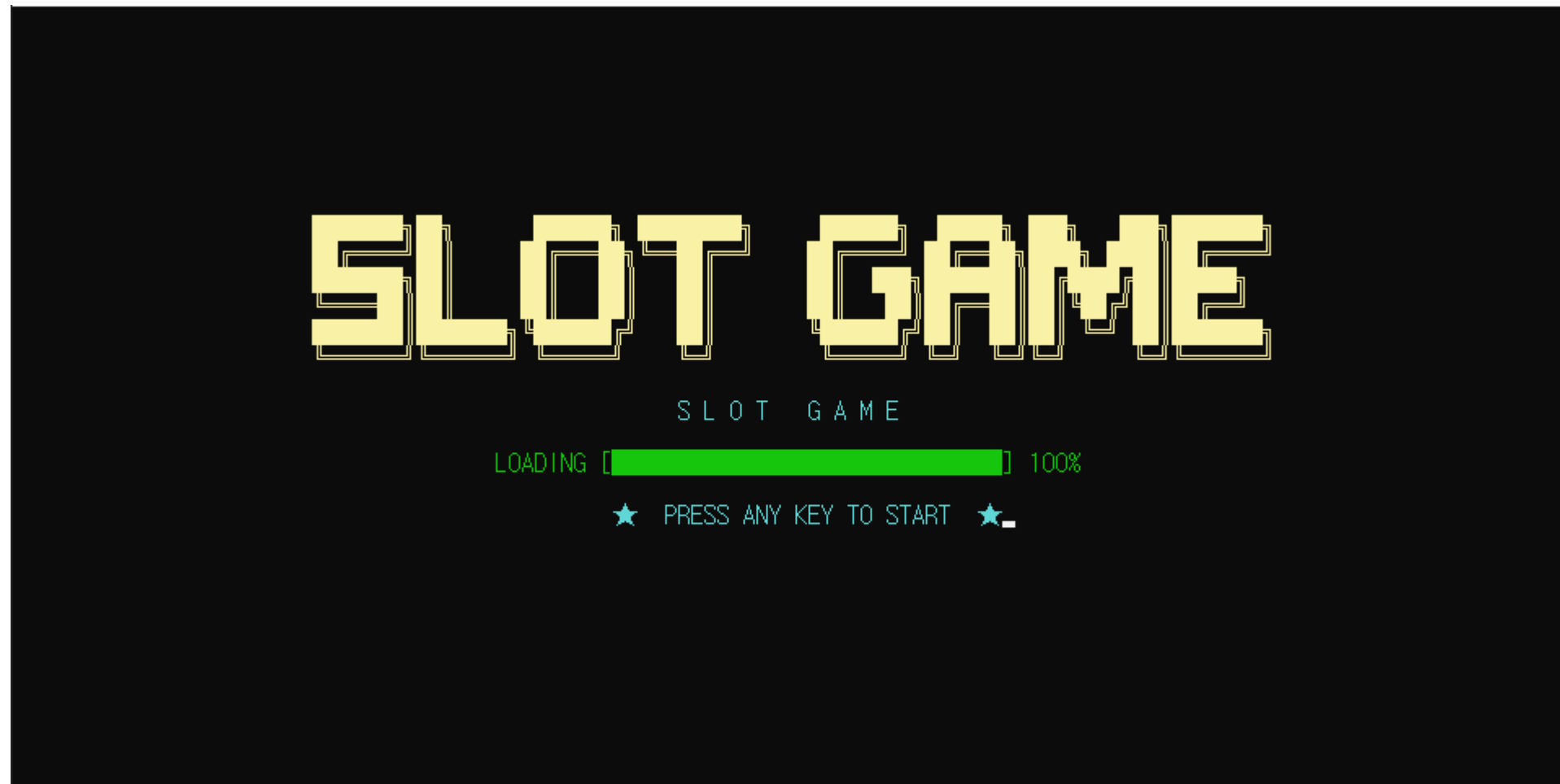
    display_reel(reel, reel_num, 0);
    bet=game_progress(*money);

    for(i=0;i<3;i++)
    {
        start=clock();
        do
        {
            for(j=i;j<3;j++)
                reel_num[0][j]=reel_num[1][j];
            reel_series(reel_num);
            display_reel(reel, reel_num, i);
            end=clock();
            pst=(double)(end-start)/CLK_TCK;
        }while(!kbhit() || (pst<1));
        num[i]=reel_num[1][i];
    }
    getch();
    thank=return_money(num, bet, &case_num);
    if (thank==0)
        *money-=bet;
    else
        *money+=thank;

    if (case_num)
    {
        gotoxy(43, 18);
        printf("사례:%d, 배팅:%d원, 사례금:%d", case_num, bet, thank);
    }
    clear_text();
}
```

수정 소스

02



게임 실행시 인트로 화면 추가
타이틀 문구와 로딩바가 추가된
인트로 화면

수정 소스

02

```
static int utf8_disp_width(const char* s) {
    int w = 0;
    for (const unsigned char* p = (const unsigned char*)s; *p; ++p) {
        if ( (*p & 0xC0) != 0x80 ) w++;
    }
    return w;
}

static void get_console_size(int* outW, int* outH) {
    CONSOLE_SCREEN_BUFFER_INFO csbi;
    GetConsoleScreenBufferInfo(GetStdHandle(STD_OUTPUT_HANDLE), &csbi);
    *outW = csbi.srWindow.Right - csbi.srWindow.Left + 1;
    *outH = csbi.srWindow.Bottom - csbi.srWindow.Top + 1;
}
```

수정 소스

02

[illegible]

```
const char* SUB = "S L O T   G A M E";
int subX = (W - utf8_disp_width(SUB))/2 + 1;
SetConsoleTextAttribute(h, CYAN);
gotoxy(subX, topY + LOGO_LINES + 1);
printf("%s", SUB);

int barW = (W >= 60 ? 30 : 20);
int loadY = topY + LOGO_LINES + 3;
char buf[160];
for (int i=0; i<=barW; ++i) {
    int pos = 0;
    pos += snprintf(buf+pos, sizeof(buf)-pos, "LOADING [");
    for (int j=0; j<i; ++j) pos += snprintf(buf+pos, sizeof(buf)-pos, "█");
    for (int j=i; j<barW; ++j) pos += snprintf(buf+pos, sizeof(buf)-pos, "░");
    pos += snprintf(buf+pos, sizeof(buf)-pos, "] %3d%%", (i*100)/barW);

    int loadX = (W - utf8_disp_width(buf))/2 + 1;
    SetConsoleTextAttribute(h, GREEN);
    gotoxy(loadX, loadY);
    printf("%s", buf);
    SetConsoleTextAttribute(h, WHITE);
    Sleep(45);
}

const char* PRESS = "★ PRESS ANY KEY TO START ★";
int pressY = loadY + 2;
int pressX = (W - utf8_disp_width(PRESS))/2 + 1;
int on = 1;
while (1) {
    gotoxy(pressX, pressY);
    SetConsoleTextAttribute(h, CYAN);
    if (on) printf("%s", PRESS);
    else    printf("                ");
    SetConsoleTextAttribute(h, WHITE);

    for (int t=0; t<6; ++t) {
        if (kbhit()) { getch(); system("cls"); return; }
        Sleep(60);
    }
    on ^= 1;
}
```

수정 소스

02

슬롯머신 게임

[당첨] 패턴:13 배당×6 | 베팅:5,000원 → 당첨:30,000원

릴1 릴2 릴3

★

○

○

♣

◆

◆

♥

★

★

	릴1	릴2	릴3	배당
1:	★	★	★	×20 (잭팟)
2:	♠	♠	♠	×7
3:	◆	◆	◆	×5
4:	♥	♥	♥	×5
5:	♣	♣	♣	×5
6:	○	○	○	×5
7:	연속(예: ◆→★→♣)×4			(순환) ×4
8:	좌우대칭(예: ◆ ★ ◆)×2			×2
9:	가운데 ★			×2
10:	모두 서로 다름			×2
11:	★	두 개		×4
12:	♠	두 개		×3
13:	◆	두 개		×3
14:	♥	두 개		×3
15:	♣	두 개		×2
16:	○	두 개		×1

LUCKY [.....] 0%

베팅 금액을 입력하고 Enter 키를 누르세요 현재 보유 금액 : 40,000원

0을 입력하면 종료합니다.

게임 진행 화면

금액을 베팅하고 게임 진행 시
상단에 정보가 출력되도록 기능 추가

수정 소스

02

```
// ===== 결과 메시지 =====
{
    char mBet[32], mPay[32];
    format_money(bet, mBet);
    format_money(pay, mPay);

    clear_line_at(2);
    gotoxy(1, 2);
    if (pay > 0) {
        int mult = (bet > 0) ? (pay / bet) : 0;
        if (lucky_used) {
            printf("[당첨] (LUCKY 적용) 패턴:%d 배당:%d | 베팅:%s원 → 당첨:%s원",
                case_num, mult, mBet, mPay);
        } else {
            printf("[당첨] 패턴:%d 배당:%d | 베팅:%s원 → 당첨:%s원",
                case_num, mult, mBet, mPay);
        }
    } else {
        printf("[꽂] 베팅:%s원 차감", mBet);
    }
}

if (pay == 0) {
    G_LOSS_STREAK++;
    G_LUCKY += 20;
    if (G_LOSS_STREAK >= 3) G_LUCKY += 10; // 3연패 가산
} else {
    G_LOSS_STREAK = 0;
    if (pay >= bet * 3) G_LUCKY -= 30;
    else G_LUCKY -= 15;
}

if (G_LUCKY < 0) G_LUCKY = 0;
if (G_LUCKY >= 100){ G_LUCKY = 100; G_LUCKY_READY = 1; }

draw_lucky_meter(GAUGE_X, GAUGE_Y);

// 하단 안내/입력 영역 정리
clear_text();
}
```


수정 소스

02

슬롯머신 게임

[당첨] 패턴:13 배당×6 | 베팅:5,000원 → 당첨:30,000원

릴1 릴2 릴3

★	○	○
♣	◆	◆
♥	★	★

	릴1	릴2	릴3	배당	
1:	★	★	★	×20	(잭팟)
2:	♠	♠	♠	×7	
3:	◆	◆	◆	×5	
4:	♥	♥	♥	×5	
5:	♣	♣	♣	×5	
6:	○	○	○	×5	
7:	연속(예: ◆→★→♣)×4			(순환)	
8:	좌우대칭(예: ◆ ★ ◆)×2				
9:	가운데 ★×2				
10:	모두 서로 다름			×2	
11:	★	두 개		×4	
12:	♠	두 개		×3	
13:	◆	두 개		×3	
14:	♥	두 개		×3	
15:	♣	두 개		×2	
16:	○	두 개		×1	

LUCKY [.....] 0%

베팅 금액을 입력하고 Enter 키를 누르세요 현재 보유 금액 : 40,000원

0을 입력하면 종료합니다.

게임 진행 화면

럭키 게이지 기능 추가를 통해
성공 기회의 폭이 늘어남

수정 소스

02

슬롯머신 게임

[당첨] 패턴: 12 배당×6 | 베팅: 333원 → 당첨: 1,998원

릴1 릴2 릴3

○

♥

♥

♦

♠

♠

★

○

○

1: ★ ★ ★ ×20 (잭팟)

2: ♠ ♠ ♠ ×7

3: ♦ ♦ ♦ ×5

4: ♥ ♥ ♥ ×5

5: ♣ ♣ ♣ ×5

6: ○ ○ ○ ×5

7: 연속(예: ♦→★→♣)×4 (순환)

8: 좌우대칭(예: ♦ ★ ♦×2

9: 가운데 ★ ×2

10: 모두 서로 다름 ×2

11: ★ 두 개 ×4

12: ♠ 두 개 ×3

13: ♦ 두 개 ×3

14: ♥ 두 개 ×3

15: ♣ 두 개 ×2

16: ○ 두 개 ×1

LUCKY [.....] 0%

베팅 금액을 입력하고 Enter 키를 누르세요 현재 보유 금액 : 11,998원

베팅 금액 입력>

0을 입력하면 종료합니다.

더블업을 하시겠습니까? (Y/N) >

LUCKY [#####] 100% READY NEXT!

게임 진행 화면

더블업 기능 추가를 통해
성공 시 상금 2배 획득 가능

더블업 실패 시 럭키 게이지 상승
럭키 게이지 100% 달성 시
무조건 당첨이 되는 천장시스템 도입

수정 소스

02

```
// 1) 기본 당첨 계산
pay = return_money(num, bet, &case_num);

// 2) LUCKY READY 적용 (보장/부스트)
int lucky_used = 0;
apply_lucky_effect(num, bet, &case_num, &pay, &lucky_used);

// 3) 더블업 (당첨시에만) - 최종 pay에 직접 반영
do_doubleup(&pay);

// 4) 잔액 반영 (더블업/READY 반영된 최종 pay 기준)
if (pay == 0) *money -= bet;
else *money += pay;

// 5) LUCKY 게이지 갱신 + 연패 관리 (1회만)
if (pay == 0) {
    G_LOSS_STREAK++;
    G_LUCKY += 20;
    if (G_LOSS_STREAK >= 3) G_LUCKY += 10;
} else {
    G_LOSS_STREAK = 0;
    if (pay >= bet * 3) G_LUCKY -= 30;
    else G_LUCKY -= 15;
}
if (G_LUCKY < 0) G_LUCKY = 0;
if (G_LUCKY >= 100){ G_LUCKY = 100; G_LUCKY_READY = 1; }

// 게이지 표시 갱신 (항상 동일 좌표)
draw_lucky_meter(GAUGE_X, GAUGE_Y);

// 6) 결과 메시지 (상단 한 줄)
{
    char mBet[32], mPay[32];
    format_money(bet, mBet);
    format_money(pay, mPay);

    clear_line_at(2);
    gotoxy(1, 2);
    if (pay > 0) {
        int mult = (bet > 0) ? (pay / bet) : 0; // 최종 배수(READY/더블업 포함)
        if (lucky_used)
            printf("[당첨] (LUCKY 적용) 패턴:%d 배당×%d | 베팅:%s원 → 당첨:%s원", case_num, mult, mBet, mPay);
        else
            printf("[당첨] 패턴:%d 배당×%d | 베팅:%s원 → 당첨:%s원", case_num, mult, mBet, mPay);
    } else {
        printf("[꽂] 베팅:%s원 차감", mBet);
    }
}
```

```
static void do_doubleup(int *pay){
    if (*pay <= 0) return; // 당첨 아닐 때는 무시
    int W,H; get_console_size(&W,&H);
    int qy = H - 2; // 질문 줄
    int ry = H - 1; // 결과 줄

    int accept = ask_yes_no(1, qy, "더블업을 하시겠습니까?");
    if (!accept) return;

    clear_line_at(ry);
    gotoxy(1, ry); printf("더블업 결과 확인 중...");
    Sleep(400);

    int win = rand() & 1; // 50%
    clear_line_at(ry);
    if (win) {
        *pay *= 2; // ★ 최종 pay 2배
        gotoxy(1, ry); printf("성공! 당첨 금액이 두 배가 되었습니다.");
    } else {
        *pay = 0; // 실패 → 당첨 무효
        gotoxy(1, ry); printf("실패... 이번 당첨은 무효가 됩니다.");
    }
    Sleep(600);
    clear_line_at(qy);
    clear_line_at(ry);
}
```

THANK YOU
