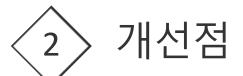
# 게임프로그래밍

응용 9-2-1









## 원본(응용9-2-1)

```
66
```

#### 디지털 레시피

- 1. 레시피
- 2. 단위환산
- 3. 프로그램 종료

메뉴번호 입력>

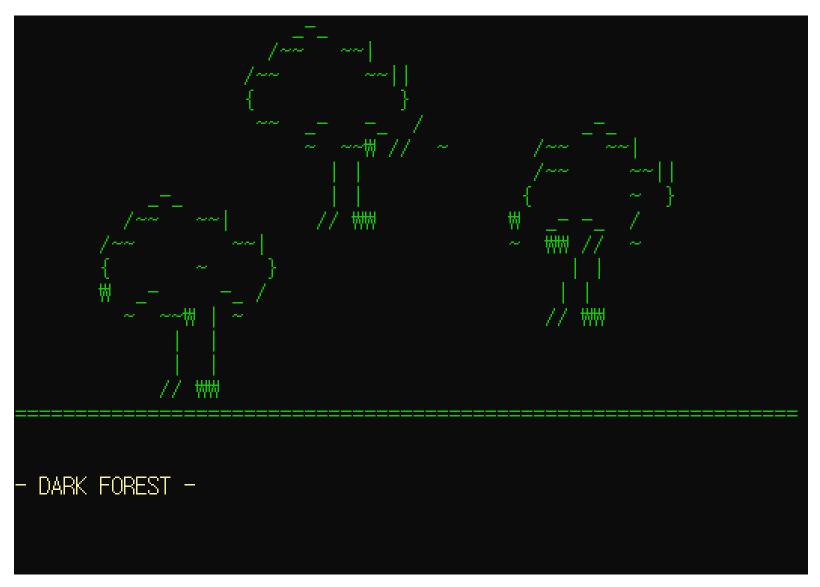
레시피출력프로그램

1 저장된 레시피 출력

2 단위 환산

# 수정본





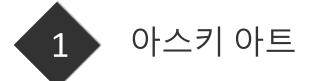
#### 간단한 스토리 게임

1 번호를 선택하여 진행

2 선택에 따른 엔딩

## 추가요소

```
static void printFrame0(void)
    printLine3("
                                                     ");
                                                    ");
");
");
    printLine3("
    printLine3("
    printLine3("
    printLine3(" : ' . ",
    printLine3(" `
    printLine3(" . ' . ", "|===|
    printLine3(" `
                                                    (");
");
");
    printLine3(" '
    printLine3(" :
   printLine3(" : . ' ",
    printLine3("
                                                     ");
    printLine3("
                                                     ");
    printLine3("
    printLine3("
                                                      ");
```



2 음악 삽입

3 선택에 따른 엔딩

3 미니게임추가

```
#ifndef ENABLE_VIRTUAL_TERMINAL_PROCESSING
#define ENABLE_VIRTUAL_TERMINAL_PROCESSING 0x0004
#endif
#define ANSI_RESET "\x1b[0m"
#define ANSI_YELLOW "\x1b[93m"
#define ANSI_RED "\x1b[91m"
#define ANSI_GREEN "\x1b[92m"
#define ANSI_BROWN "\x1b[38;5;94m"
```

1. Define ANSI로 색을 정해두고 필요할 때마다 호출

```
choice = key - '0';

// 유효 범위 체크

if (choice >= min && choice <= max)
    return choice;

// 잘못된 숫자 입력 시 선택지 다시 출력

system("cls");
slowPrint("을 바른 선택지를 입력하세요.", 15);
printf("\n\n");
if (printOptions != NULL)
    printOptions();
printf("\n 선택 > ");
}
```

- 1. 숫자가 아닌 값이 입력되면 되돌아감
- 2. 아스키 코드 (입력값 0)이 1~3이 아닐 경우 다시 선택지로 되돌아감

- 1. i%2로 두 그림을 번갈아 출력
- 2. Sleep(300) 프레임간 시간 0.3으로 설정

```
void printOptions_askBirds(void)
{
    printf(" 1. 작은 새\n");
    printf(" 2. 큰 새\n");
    printf(" 3. 긴 새\n");
    if (heardAboutBigBird == 1 && heardAboutLongBird == 1 && heardAboutSmallBird == 1)
        printf(" 4. 이제 충분히 들은 것 같다.\n");
}
```

1. 모든 설명을 듣고 나서야 입력 가능한 선택지

```
system("cls");
switch (choice)
   case 1:
       slowPrint("긴 새 : \"더
       slowPrint("당신은 긴 새
       break;
   case 2:
       slowPrint("긴 새 : \"언
       printf("\n");
       slowPrint("긴 새 : \"앞
       printf("\n");
       slowPrint("당신은 긴 새
       printf("\n");
       animateScales();
       badChoices++;
       break;
   case 3:
       slowPrint("긴 새 : \"다
       slowPrint("당신은 긴 새
       break;
```

- 1. animateScales()로 컷씬 출력
- 2. badChoices로 특정 엔딩

#### 코드 설명

```
static void printLine3(const char* left, const char* mid, const char* right)
    printf(ANSI_RED "%s" ANSI_YELLOW "%s" ANSI_RED "%s" ANSI_RESET "\n", left, mid, right);
static void printFrame0(void)
    printLine3("
    printLine3("
    printLine3('
    printLine3(
    printLine3(
    printLine3(
    printLine3(
    printLine3(
    printLine3(
    printLine3("
    printLine3(
    printLine3(
    printLine3("
                                                     ");
    printLine3("
                                                      ");
    printLine3("
static void printFrame1(void)
    printLine3("
    printLine3("
    printLine3("
    printLine3(
    printLine3(
    printLine3(
                                                     ");
    printLine3(
    printLine3(
    printLine3(
    printLine3("
    printLine3(
    printLine3("
                                                     ");
");
    printLine3("
    printLine3("
    printLine3("
```

- 1. ','를 기점으로 색상 변경
- 2. 두 개의 그림으로 움직이는 효과

```
void playIntroMelody(void)
   Beep(392, 300); Sleep(80);
   Beep(440, 350); Sleep(80);
   Beep(494, 350); Sleep(100);
   Beep(523, 400); Sleep(100);
   Beep(587, 350); Sleep(80);
   Beep(523, 350); Sleep(80);
   Beep(659, 400); Sleep(100);
   Beep(587, 450); Sleep(100);
   Beep(523, 500); Sleep(100);
    Beep(392, 650);
//플레이어가 바위에서 넘어 짐
void playFallSound(void)
   Beep(330, 120); Sleep(60);
   Beep(294, 100); Sleep(50);
   Beep(196, 200); Sleep(80);
   Beep(130, 300);
                   Sleep(100);
   Beep(98, 500);
```

#### 코드 설명

```
int playBlockGame(void)
   int block_order = 0;
   int success_flag = 0;
   int block_stack[box_length * 2 + 1] = {0};
   system("cls");
   printf("새들이 하나로 합쳐지려 하고 있습니다...\n");
   Sleep(1500);
   printf("\n");
   printf("그들의 힘을 조율하려면 최대한 높이 쌓아올리세요 \n\n");
   Sleep(2000);
   while (block order < 3)
       int w, h;
       if (block order == 0) { w = 4; h = 4; }
       else if (block_order == 1) { w = 1; h = 4; }
       else { w = 2; h = 2; }
       int temp_x = 4, temp_y = 2, temp = 2;
```

- 1. Block\_order : 현재 쌓인 블록 개수
- 2. Success\_flag: 성공 높이 확인
- 3. Block\_stack : 블록 높이 저장

```
// 블록 이동:
do {
    for (int i = 0; i < h; i++) {
        gotoxy(temp_x, temp_y + i);
        for (int j = 0; j < w * 2; j++)
            printf(" ");
    temp_x += temp;
    if (temp_x >= (box_length * 2 - (w * 2))) temp = -2;
    if (temp_x <= 4) { temp_x = 4; temp = 2; }</pre>
    for (int i = 0; i < h; i++) {
        gotoxy(temp_x, temp_y + i);
        for (int j = 0; j < w; j++)</pre>
            printf("" ");
    Sleep(60);
} while (!kbhit());
getch();
```

- 1. Kbhit : 키보드 입력 여부 확인
- 2. "" 출력으로 이전에 있던 블록 삭제
- 3. 좌우 2칸씩 이동
- 4. 새로운 좌표에 블록을 출력하여 이동을 시각화함

```
int max_h = 0;
for (int i = 0; i < w * 2; i++)
    if (block_stack[temp_x + i] > max_h)
        max_h = block_stack[temp_x + i];

int target_y = box_height + 2 - max_h - h + 1;
```

```
for (int y = 2; y < target_y; y++) {
    for (int i = 0; i < h; i++) {
        gotoxy(temp_x, y + i);
        for (int j = 0; j < w; j++)
            printf("□");
    }
    Sleep(40);
    for (int i = 0; i < h; i++) {
        gotoxy(temp_x, y + i);
        for (int j = 0; j < w * 2; j++)
            printf("");
```

- 1. 블록이 놓일 자리에 쌓인 블록 중 가장 높은 위치를 찾음
- 2. 현재 블록이 멈출 Y 좌표를 계산
- 3. Y좌표를 이동시키며 부드럽게 내려오도록 만들어줌

```
for (int i = 0; i < h; i++) {
    gotoxy(temp_x, target_y + i);
   for (int j = 0; j < w; j++)
        printf("" ");
for (int i = 0; i < w * 2; i++)
    block_stack[temp_x + i] += h;
// 누적 높이 체크(
int overall_max = 0;
for (int i = 0; i < box_length * 2; i++)</pre>
    if (block_stack[i] > overall_max)
        overall_max = block_stack[i];
if (overall_max >= GOAL_HEIGHT) {
    gotoxy(1, box_height + 5);
    printf("세 마리의 새들이 모여 종말새가 되었습니다.\n");
    printf("\n");
    success_flag = 1;
    break;
block_order++;
```

// 착지 후 고정:

- 1. 쌓은 높이가 GOAL\_HEIGHT보다 높으면 성공 판정
- 2. GOAL\_HEIGHT는 10으로 정의해둠

#### 감사합니다