

최종 결과 보고서

(2021학년도 1학기)

과제명	내 꿈은 영국신사! 텍스트 어드벤처 게임		
과목 /담당교수	프로그래밍설계 / 정복래 교수님		
제출일자	2022. 06. 20		
연구참여자 (담당분야)	팀장: 20190895 김찬영 (총괄, 보고서 작성 및 System.c 설계) 팀원: 20190898 김태현 (서기, System 및 Mini game 함수 설계) 팀원: 20190919 이상욱 (자료조사, Event 함수 설계) 팀원: 20190954 허진환 (ppt 제작, Mini game 함수 설계)		
팀명	비틀즈	팀 번호	07조

목 차

1. 설계과제 제목	3
2. 연구목적	3
3. 설계과제의 필요성	3
4. 설계과제 내용 및 목표	4
5. 설계과정	6
6. 제작	10
7. 시험	15
8. 평가	21
9. 추진체계	24
10. 설계 추진일정	24
11. 결론	25

1. 설계과제의 제목

내 꿈은 영국신사! 텍스트 어드벤처 게임

2. 연구목적

본 프로젝트를 추진을 통해 궁극적으로 얻고자 하는 내용은 다음과 같다.

- 1) Visual C를 이용해 여러 라이브러리 함수들의 이해 및 응용
- 2) 게임 개발 및 제작에 대한 학습 및 이해
- 3) 설계한 게임의 구현을 위한 알고리즘 작성 능력 향상
- 4) 라이브러리 함수를 사용하여 모듈화 프로그래밍 능력 향상
- 5) 모듈화 프로그래밍을 통하여 협동 프로젝트 수행

3. 설계과제의 필요성

여태까지 1학년 과정까지는 Visual C를 이용해 프로그래밍을 진행할 때, 개인으로 진행하는 경우가 대부분이었으며 팀 단위로 프로젝트를 진행한 경험은 없었다. 하지만 개인 단위가 아닌, 팀 단위로 프로젝트를 진행하는 것은 의견 조율부터 시작해서 효율적인 프로그래밍을 위한 역할 분담, 소통이 계속해서 요구되고 사회에서 원하는 프로그래머 또한 그렇다. 따라서 이 과제를 수행하면서 그동안 경험할 수 없었던 팀 프로젝트를 진행하고 진행하는 과정에서 많은 것들을 배울 수 있다.

4. 설계과제 내용 및 목표

4.1 설계과제 내용

Visual C를 기반으로cmd 창 내에서 텍스트 출력 및 선택지 기반의 게임을 개발한다. 게임의 제목은<내 꿈은 영국신사!>이며 장르는 텍스트 어드벤처이다. 게임의 목적은 사용자가 게임 속 캐릭터를 여러 가지 사건으로부터 살아남고, 캐릭터의 능력치를 키우는 것이다.

포함되는 기능들은 캐릭터의 선택, 스토리를 화면에 출력, 미니게임의 플레이, 여러 이벤트에 대한 선택지 제시, 캐릭터의 현재 상태에 따라 게임의 종료가 포함된다.

게임의 기본적인 구조의 경우, cmd 화면 내에서 사용자가 게임에 몰입할 수 있도록 문자열 출력에 시각적인 효과를 주고, 사용자의 입력을 받아 결과를 처리하며 이벤트 및 미니게임의 출력을 무작위로 발생할 수 있게끔 제어하고, 전반적으로 게임을 시작-진행-종료하게끔 진행시켜 준다.

캐릭터 선택 기능의 경우, 초기 능력치가 다른 5종류의 캐릭터를 설정한다. 캐릭터는 총 4가지의 능력치를 갖고 있으며, **골드, 체력, 민첩, 힘**으로 구성된다.

각각의 캐릭터는 초기 능력치 및 클리어 조건 능력치에 있어서 차이를 가지고 있으며, 이에 따라 쉬움, 보통, 어려움, 매우 어려움, 지옥 총 5가지 난이도를 가지게 된다. 사용자는 이를 직접 선택해서 플레이하게 한다.

스토리를 화면에 출력하는 기능의 경우, 화면을 제어하는 함수를 사용하여 가독성을 높여 사용자에게 스토리 및 게임, 사건의 내용을 정확하게 전달한다.

미니 게임의 플레이 기능의 경우, 간단한 게임들을 사용자가 직접 플레이하게 하고, 플레이 결과에 따라 캐릭터의 상태에 영향을 미치게 한다. 미니게임의 종류에는 카드를 이용한 게임(인디언 포커, 할리갈리 등), 슬롯머신 게임 등 간단하게 즐길 수 있는 게임들이 있다.

여러 이벤트에 대한 선택지 제시 기능의 경우, 스토리가 진행되면서 캐릭터는 무작위로 여러 가지 이벤트를 마주치게 되는데, 그때 선택지를 고르게 해서 캐릭터의 상태 및 게임의 엔딩에 영향을 미치는 기능이다. 이벤트를 진행하면서 캐릭터의 기본 4가지 능력치에 증감을 줘서 배드 엔딩 또는 해피 엔딩으로 다가갈 수 있게 한다.

조건에 따라 게임의 종료 기능은, 여러 가지 조건을 고려하여 엔딩을 결정하는 기능이다. 우선 이벤트 및 미니게임 플레이가 일정 조건을 만족하게 되면, 캐릭터의 체력 및 골드 상태를 확인하는데, 이때 일정 기준 이상이면 바로 해피 엔딩으로 게임이 종료된다. 만약, 조건을 충족하지 못했거나 게임의 종료 판단 시점 이전에 캐릭터의 기본 4가지 능력치 중 하나의 값이 0이 되면 배드 엔딩으로 게임이 종료된다.

4.2 설계과제 목표

본 과제에서 달성하고자 하는 목표는 다음과 같으며, 이외에도 본 과제에서 아래와 같은 현실적인 제한요소들을 달성한다.

- 1) Visual C를 이용해 여러 라이브러리 함수들의 이해 및 응용
- 2) 게임 개발 및 제작에 대한 학습 및 이해
- 3) 설계한 게임의 구현을 위한 알고리즘 작성 능력 향상
- 4) 라이브러리 함수를 사용하여 모듈화 프로그래밍 능력 향상
- 5) 모듈화 프로그래밍을 통하여 협동 프로젝트 수행

표 1. 본 프로젝트의 현실적 제한요소 항목

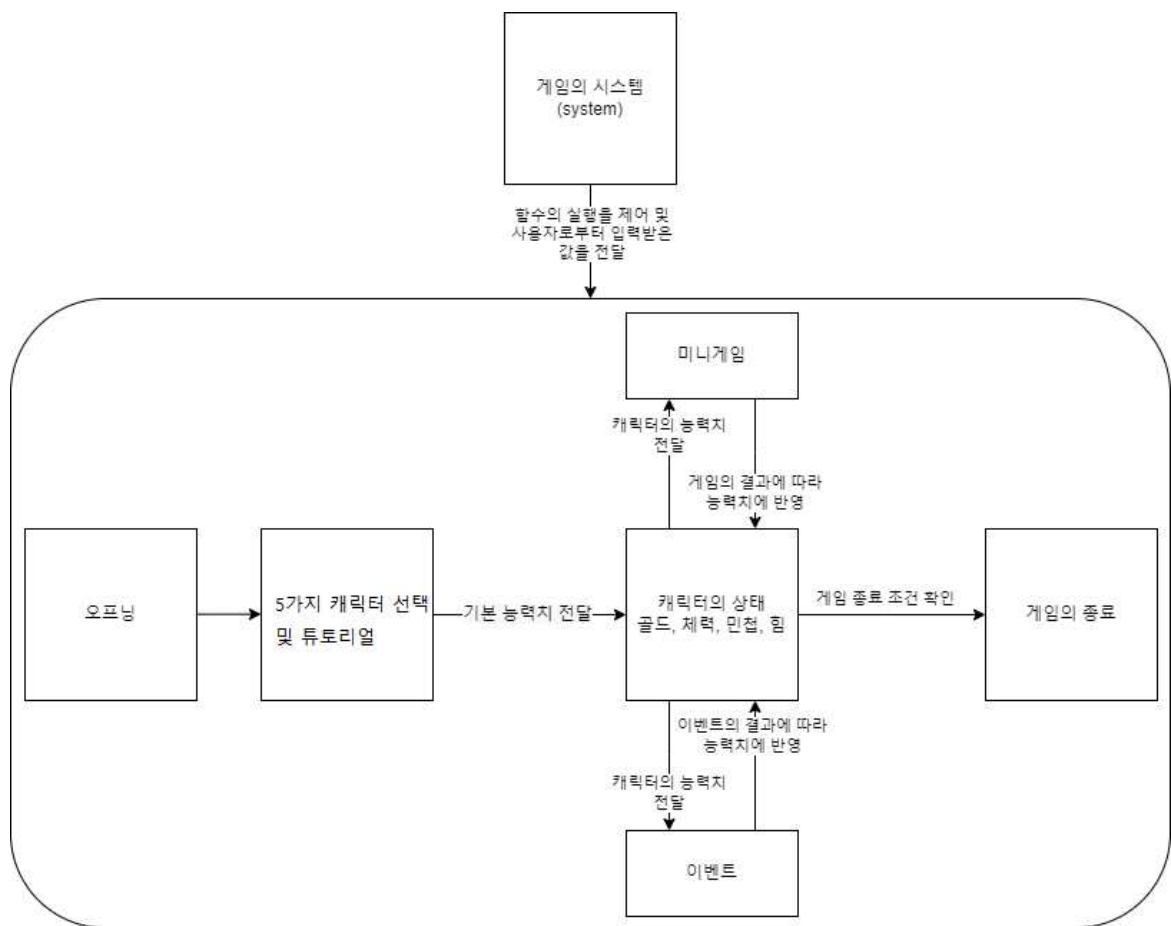
현실적 제한 요소들	내 용 (Content)
경제	<ul style="list-style-type: none"> - 개발 제품의 가격 경쟁력 확보 - 낭비되는 메모리 없이 최소한의 리소스만을 사용하여 프로그램을 구현해 경제성 고려
편리	<ul style="list-style-type: none"> - cmd 창 내부에서 최대한 사용자의 가독성을 고려하여 개발 (문자열의 출력, 화면의 제어 등)
윤리	<ul style="list-style-type: none"> - 게임 내 사행성 요소가 포함되어 있기에 게임물관리위원회의 기준에 따라 윤리성 고려 (게임 이용 가능한 나이) - 악의적인 사용을 방지하기 위한 2차 창작의 금지
사회	<ul style="list-style-type: none"> - 게임 중독 예방을 고려하여 게임 실행 도중 실행 시간의 알림

5. 설계과정

5.1 설계 기초이론

- 5.1.1 데이터 구조 : 카드게임 설계 시 필요
- 5.1.2 Visual C 프로그래밍 : 전반적인 프로그래밍 시 필요
- 5.1.3 게임 시나리오 제작 : 이벤트 설계 및 게임 밸런스 조정 시 필요

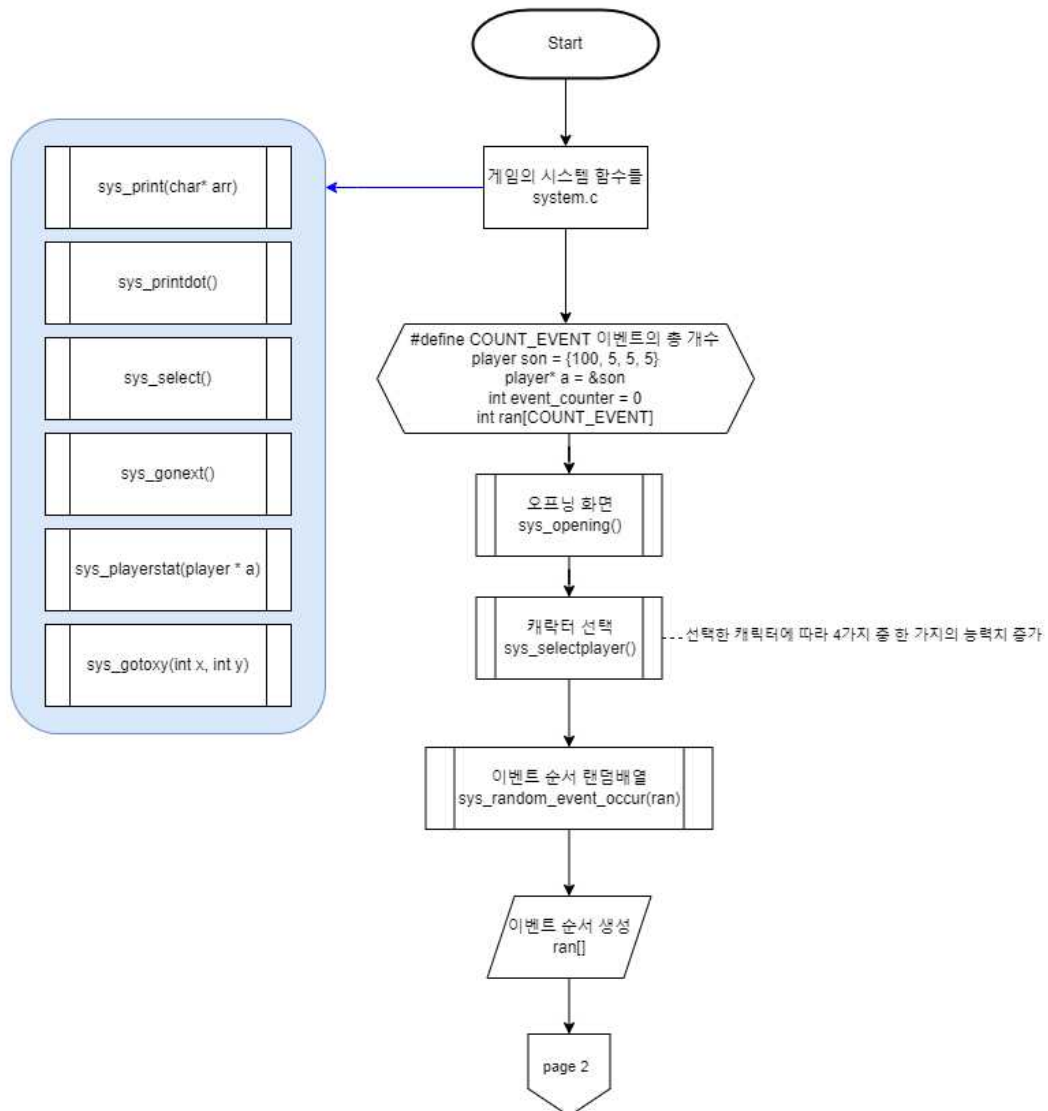
5.2 소프트웨어 기능블록도 (개념설계)



5.3 기능적 요구사항 명세서

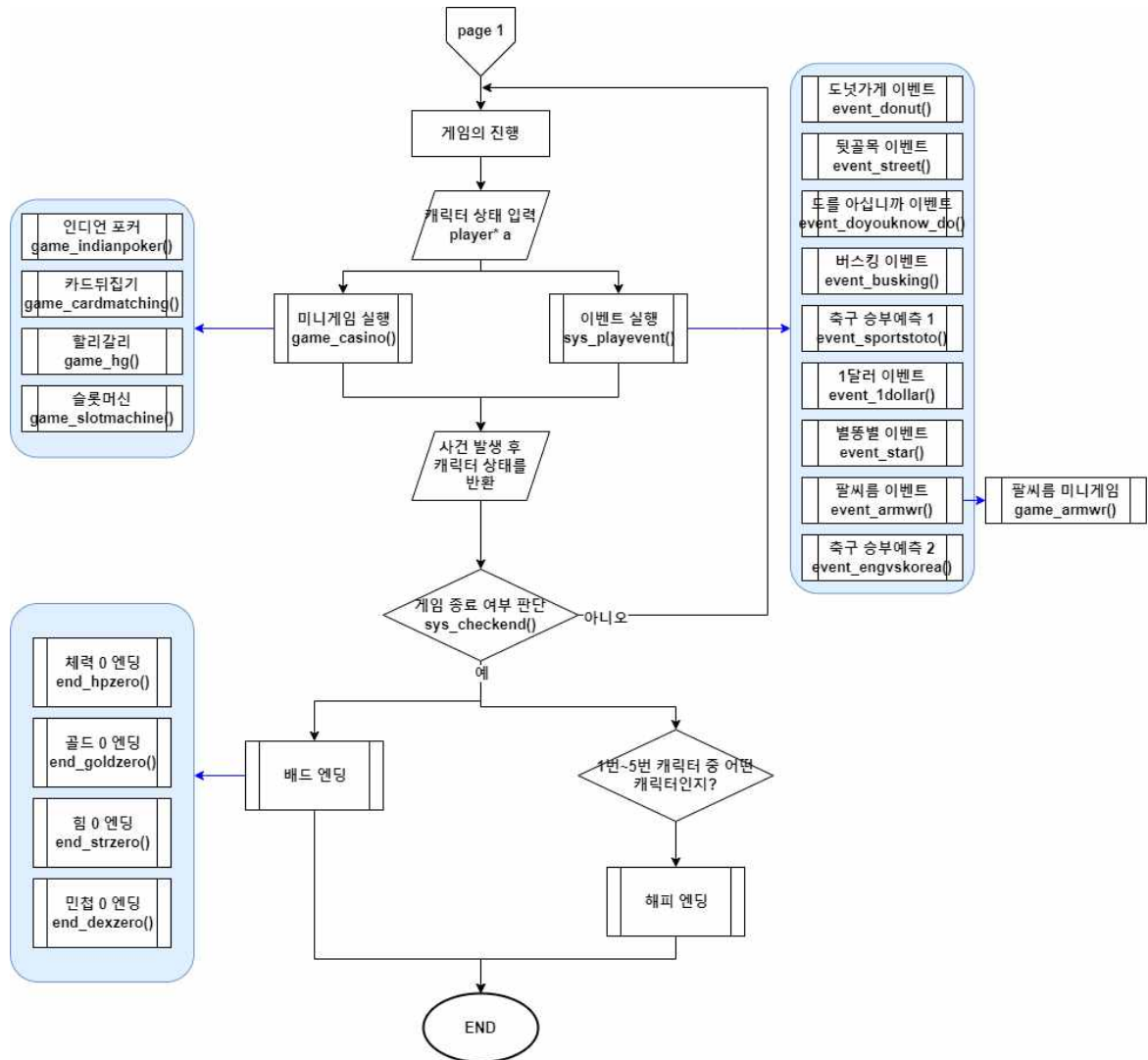
ID	요구사항	내용	설명	우선순위
p_01	게임의 시작 및 튜토리얼	게임 시작 시 게임 타이틀 화면을 출력하고, 게임을 설명한다.	프로그램 실행 시 타이틀 화면을 출력하고, 게임 시작을 선택하면 게임의 배경을 설명해야 한다.	4
p_02	캐릭터 선택 기능	유저가 캐릭터를 선택하게 한다.	각각 기본 능력치 및 클리어 목표가 다른 5가지 종류의 캐릭터를 사용자가 직접 선택한다.	5
p_03	게임의 종료	특정 조건을 만족할 경우 게임을 종료시킨다.	게임 진행 상태 및 캐릭터의 능력치에 따라 게임의 엔딩이 달라져야 하며 여러 가지 엔딩을 포함하여 사용자에게 다양한 경험을 제공한다.	6
p_04	캐릭터의 상태 알림	캐릭터의 현재 능력치를 모두 표시한다.	게임의 종료에 가장 크게 영향을 미치는 캐릭터의 4가지 요소를 직관적으로 사용자에게 알려주어야 한다.	7
p_05	게임의 시스템 (system)	텍스트를 화면에 출력하고, 게임을 하면서 발생하는 데이터를 관리한다.	cmd 화면 내에서 사용자에게 그저 문자열 출력을 하는 것이 아닌 여러 가지의 시각적인 효과를 주고, 직접 미니게임 및 이벤트를 플레이하여 몰입감을 높일 수 있어야 한다. 또한, 사건의 발생 및 게임의 진행을 제어할 수 있어야 함	1
p_06	여러 가지 미니게임 플레이	미니게임 플레이를 통해 캐릭터에게 상태 변화를 준다.	미니게임의 결과에 따라 캐릭터의 골드 및 체력에 영향을 주고, 게임의 난이도를 조정하여야 한다.	2
p_07	여러 가지 이벤트의 발생	여러 가지 이벤트의 발생을 통해 게임의 배경 및 스토리를 설명한다.	이벤트를 통해 사용자는 게임의 스토리를 알 수 있어야 하고, 결과에 따라 캐릭터의 골드 및 체력에 영향을 줄 수 있어야 한다.	3

5.4 전체 순서도 (상세설계)



우선c언어에서 텍스트 어드벤처 장르를 구현하기 위해 시각적인 부분 또는 게임 시스템적인 부분에서 기본적인 틀을 잡아야 한다고 생각했기에, system.c라는 소스파일을 하나 만들어 게임 실행 및 진행에 필요한 여러 함수들을 포함시켜야 한다고 설정하였다.

또한, 게임을 매번 실행할 때마다 항상 똑같은 순서로 이벤트나 게임이 실행되면 사용자에게 재미를 줄 수 없다고 생각했기 때문에, 게임 실행 시 매번 랜덤한 순서로 이벤트가 발생하도록 이벤트 랜덤 순서 배열 및 실행에 관한 함수들을 system.c에 포함시켰다.



앞서 설명한 바와 같이, 게임을 진행하면서 여러 이벤트들 및 미니게임들을 플레이하게 되는데, 모든 이벤트 및 미니게임들의 함수는 struct player* a를 인수로 받아 사건의 결과가 캐릭터의 상태에 영향을 미친다.

미니 게임이나 이벤트의 실행이 완료되면 캐릭터의 상태에 변화가 생기기 때문에, 항상 게임의 종료 여부를 확인하는 과정을 거친다. 이 때 조건을 만족했을 경우, 각 조건에 맞는 엔딩이 출력되면서 게임이 종료되고, 어떠한 조건도 만족하지 못했을 경우 계속해서 미니 게임 및 이벤트를 반복하게 된다.

6. 제작(Implementation)

6.1 제작과정

(전체 소스코드는 사이버캠퍼스 업로드 시 첨부하여 올려놓았음)

팀 프로젝트에서 가장 중요한 것은 효율적으로 조원 모두가 프로그래밍에 참여하는 것이라고 생각했기에, 각자의 장점을 살릴 수 있는 역할을 맡아서 제적을 진행했다.

소스 코드 및 헤더 파일을 포함하면 총 11개의 .c파일을 프로그래밍했다.

1) **main.c** : 메인 함수가 포함된 소스코드

2) **system.c** : 게임의 전반적인 틀을 잡는 함수가 포함된 소스코드

void sys_gotoxy(int x, int y) : gotoxy함수와 같음

void sys_print(char* print) : 문자열을 말 하는 것처럼 출력

void sys_printdot() : 점을 출력하며 로딩효과를 준다

int sys_select() : 선택 번호를 입력받아 반환하는 함수

int sys_reselect() : 다시 선택 번호를 입력받아 반환

void sys_playerstat(player* a): 캐릭터의 능력치를 말해주는 함수

void sys_printstat(player* a) :캐릭터의 능력치를 상단에 표시

void sys_gonext() : 키 입력을 받기 전 까지 화면을 멈춤

void sys_random_event_occur(int ran[]) :

이벤트의 개수에 따라 랜덤한순서를 생성하는 함수

int sys_playevent(int eventcount, int ran[], player* a):

랜덤 순서로 배열된 이벤트를 실행하는 함수. 코드를 한 번 실행할 때 마다

그 순서에 맞는 이벤트가 실행된다.

void sys_selectplayer(player* a): 캐릭터 선택 함수

void sys_opening() : 오프닝 출력 함수

void sys_tutorial(player* a) : 튜토리얼 실행 함수

3) **ending.c** : 게임의 클리어 조건을 확인하는 함수 및 엔딩장면을 출력하는 함수 포함

void end_print_clear() : 클리어 화면 출력

void end_print_gameover() : 게임오버 화면 출력

void end_goldzero() : 골드가 0일 때 배드엔딩

void end_hpzero() : 체력이 0일 때 배드엔딩

void end_strzero() : 힘이 0일 때 배드엔딩

void end_dexzero() : 민첩이 0일 때 배드엔딩

void sys_checkend(player* a) :

엔딩 조건을 확인한 뒤, 이에 해당되면 해피 엔딩을 보여주고 아닐 경우 계속 게임을 진행하게 한다.

4) event.c : 이벤트 함수들이 포함된 소스코드

event_donut(a); 도넛가게 이벤트
 event_street(a); 뒷골목 이벤트
 event_doyouknow_do(a); 도를 아십니까? 이벤트
 event_busking(a); 버스킹 이벤트
 event_sportstoto(a); 첼블루VS갱스날 승부예측 이벤트
 event_1dollar(a); 1달러만 이벤트
 event_star(a); 별똥별 이벤트
 event_armwrestle(a); 팔씨름 이벤트
 event_engvskorea(a); 한국VS영국 승부예측 이벤트

5) game_armwr.c : 팔씨름 게임. 팔씨름 이벤트로부터 호출된다.

6) game_casino_enter.c : 카지노 입장하는 함수. 카지노 내에서 인디언 포커, 카드 뒤집기, 할리갈리, 슬롯머신을 실행할 수 있다.

7) game_indianpoker.c : 인디언 포커 게임

void make_deck(card* deck) : 덱 생성 함수
 void shuffle_deck(card* deck) : 덱 셔플 함수
 int is_empty_deck() : 덱에 카드가 0장인지 확인하는 함수
 void init_deck() : 덱을 초기화하는 함수
 card draw_card() : 스택 자료구조에서의 pop()과 동일. 카드 한장 반환
 void make_standard_deck() : 덱 초기화 후 덱을 생성하여 셔플하는 함수
 void print_card(card ex) : 카드 숫자와 모양을 출력
 void print_card_number(card ex) : 11, 12, 13번째 카드를 J, Q, K로 출력해야 함
 int precedence_card(card ex) : 카드 모양의 우선순위. 숫자가 높을수록 높다
 void print_indianpoker() : 인디언포커 글자를 ASCII ART로 출력함
 void print_table(card a) : 카드의 모양을 화면에 출력해줌
 int print_my_card(card a) : 플레이어의 카드를 화면에 출력해주는 함수
 int rank_card(card me, card dealer) : 베팅 후 카드의 승패를 결정해주는 함수
 void game_indianpoker(player* a) : 인디언포커 게임 실행 메인 함수

8) game_matchingcard.c : 카드 뒤집기 게임

void Setchoicecard() : 14장의 카드를 순서대로 만든다. 11223344556677 총 14장 생성

void Random() : 위에서 생성한 카드를 셔플

void Initcard() : 14장의 카드에 초기값 할당

void Displaynum() : 카드의 숫자를 보여주는 함수

void Displaycardopen() : 선택한 카드를 오픈해주는 함수

int PlayGame() : 게임을 진행하게 하는 함수

int game_matchingcard : 카드뒤집기 게임 메인 실행 함수

9) game_hg.c : 할리 갈리 게임

void make_hg_deck(card* deck) : 각 모양의 카드를 1부터 5까지, 2장씩 총 40장의 카드 덱 생성

void shuffle_hg_deck(card* deck) : 카드 뭉치를 셔플하는 함수

void shuffle_player_deck(card* player, int count) : 종을 올린 후 바닥의 카드 뭉치를 승자가 가져가는데, 그렇게 합친 카드 뭉치를 다시 섞는 함수

void deal_hg_deck(card* deck, card* player1, card* player2) : 플레이어와 딜러가 서로 카드를 나눠가짐

double hg_dspped() : 딜러의 벨 누르는 속도 랜덤값으로 반환 (0.6초~1.2초 사이)

int check_hg() : 할리갈리 조건을 만족할 경우를 확인하는 함수

void print_interface_hg() : 할리 갈리 인터페이스 출력 함수

void print_hg(card table1, card table2) : 카드 모양을 화면에 출력

void init_table() : 처음 게임 실행 시 바닥의 카드는 0장으로 설정

void game_hg(player* a) : 할리 갈리 메인 실행 함수

10) game_slotmachine.c : 슬롯 머신 게임

void display_rule(void) : 당첨 사례와 사례 금액을 출력하는 함수

void reel_series(int r[][3]) : 각 릴에 대한 기호의 순서를 결정하는 함수

int game_progress(int money) : 게임 진행상황과 배팅 금액을 입력받아 반환하는 함수

void display_reel(char rc[][3], int r[][3], int index) : 회전순서에 따라 릴을 화면에 출력하는 함수

void clear_text(void) : 화면에 표시된 지시어들을 지우는 함수

int game_control(char reel[][3], int reel_num[][3], int* money) : 슬롯머신게임을 제어하는 함수

int return_money(int r[], int betting, int* case_num) : 당첨사례따라배팅 금액과 사례금을 계산하는 함수

int play_slotmachine(void) : 슬롯머신 메인 실행 함수

11) 헤더.h : 통합 헤더파일을 코딩해 #define문을 모든 소스코드에 일괄적으로 적용될 수 있도록 사용했다.

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <Windows.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>
#include <mmsystem.h>
#pragma comment(lib, "winmm.lib")

#define COUNT_EVENT 9          //총 구현한 이벤트 개수

//캐릭터 1번 = 이윤재 클리어 조건
#define CHARNUM1_CLR_GOLD 600

//캐릭터 2번 = 손우민 클리어 조건
#define CHARNUM2_CLR_GOLD 700
#define CHARNUM2_CLR_DEX 9

//캐릭터 3번 = 기신용 클리어 조건
#define CHARNUM3_CLR_GOLD 1000
#define CHARNUM3_CLR_HP 9
#define CHARNUM3_CLR_STR 9

//캐릭터 4번 = 간두리 클리어 조건
#define CHARNUM4_CLR_GOLD 1000
#define CHARNUM4_CLR_HP 10
#define CHARNUM4_CLR_DEX 10
#define CHARNUM4_CLR_STR 10

//캐릭터 5번 = 밥취용 클리어 조건
#define CHARNUM5_CLR_GOLD 1500
#define CHARNUM5_CLR_HP 10
#define CHARNUM5_CLR_DEX 10
#define CHARNUM5_CLR_STR 10

typedef struct player {        //캐릭터 구조체
    int charnum;              //고유 캐릭터 번호
    int gold;                 //골드
    int hp;                   //체력
    int dex;                  //민첩
    int str;                  //힘
}player;
```

6.2 제작 시 문제점 및 개선사항

6.2.1 입력 버퍼에 관한 문제

특정 상황에서 선택지 입력 혹은 게임을 플레이하고 나서 버퍼가 초기화되지 않고 계속 남아있기 때문에, 다음 선택지를 고를 때 원하지 않은 값이 들어가는 문제가 발생하였다. 이를 해결하기 위해 `fflush()`, `getch()`, `rewind()` 함수 등 여러 라이브러리 함수들을 시도했지만, 결국 `getch()` 함수를 사용해 해결하였다.

6.2.2 입력 버퍼를 초기화했어도 발생하는 문제

입력 버퍼를 초기화한 다음, 선택지에 원하는 값을 입력하기 위해 키보드의 숫자를 입력하여도 ENTER가 입력되는 문제가 특정 경우에 발생하는 것을 확인하였다. `sys_gonext()` 함수에서 `while(!kbhit())`이란 조건으로 화면을 멈추는 방식을 사용했는데, 이 과정에서 ENTER를 입력하여 프로그램을 진행시킬 경우 발생하는 문제라고 판단하였다.

이를 해결하기 위해 인터넷 검색 및 여러 방법을 시도해봤지만 찾을 수 없었고, 결국 ENTER 이외의 키를 입력해 게임을 진행하는 대안을 찾았지만 근본적인 문제 해결을 할 수 없었다.

6.3 지도교수 지도내용 및 지적사항 조치결과

1) 게임이 텍스트로만 진행되면 너무 단조로울 것 같음

이벤트를 진행하면서 여러 장소에 도착하는 것을 ASCII ART로 표현하였다. 또한, 캐릭터를 선택한 다음 캐릭터의 얼굴을 화면에 ASCII ART로 표현하였다.

게임 중간중간 출력되는 글자들이 그저 문자열로 출력되는 것이 아니라, 말을 하는 것처럼 표현이 된다면가 아니면 아래와 같은 방식으로 표현하여 시각적인 효과를 주었다.



2) 게임 중 사운드의 출력이 있으면 좋겠음

오프닝 화면, 각 이벤트 및 미니게임을 플레이할 때 상황에 맞는 BGM을 선별하여 게임에 추가하였다.

7. 시험

아무래도 우리가 설계한 것이 ‘게임’이다 보니, 오류 없이 플레이어가 선택한 방향으로 게임이 흘러가는 것이 가장 중요하다고 생각했다. 따라서, 팀원 모두가 계속해서 플레이하면서 모든 경우의 수를 테스트하였다. 그 과정에서 (보고서의 6.2.2 입력 버퍼를 초기화했어도 발생하는 문제)를 확인하였고, 이를 해결하기 위해 여러 방법을 시도했지만 해결할 수 없었다.

위 문제 이외에의 모든 경우의 수에 대해 확인을 진행한 결과, 우리가 정의한 함수들이 정상적으로 의도한 바와 같이 작동하는 것을 확인할 수 있었다.

또한, 게임에서 밸런스와 같은 부분도 빼놓을 수 없는데, 5가지 캐릭터를 쉬움, 보통, 어려움, 매우 어려움, 지옥 총 5가지 난이도로 나누었다. 이 난이도가 적절한지를 확인 및 수정하기 위해 계속 실행해본 결과 팀원 모두가 만족할만한 난이도 설정을 마칠 수 있었다.

그림1. 오프닝 출력 화면



그림2. 캐릭터 선택 화면



그림3. 튜토리얼 실행 화면

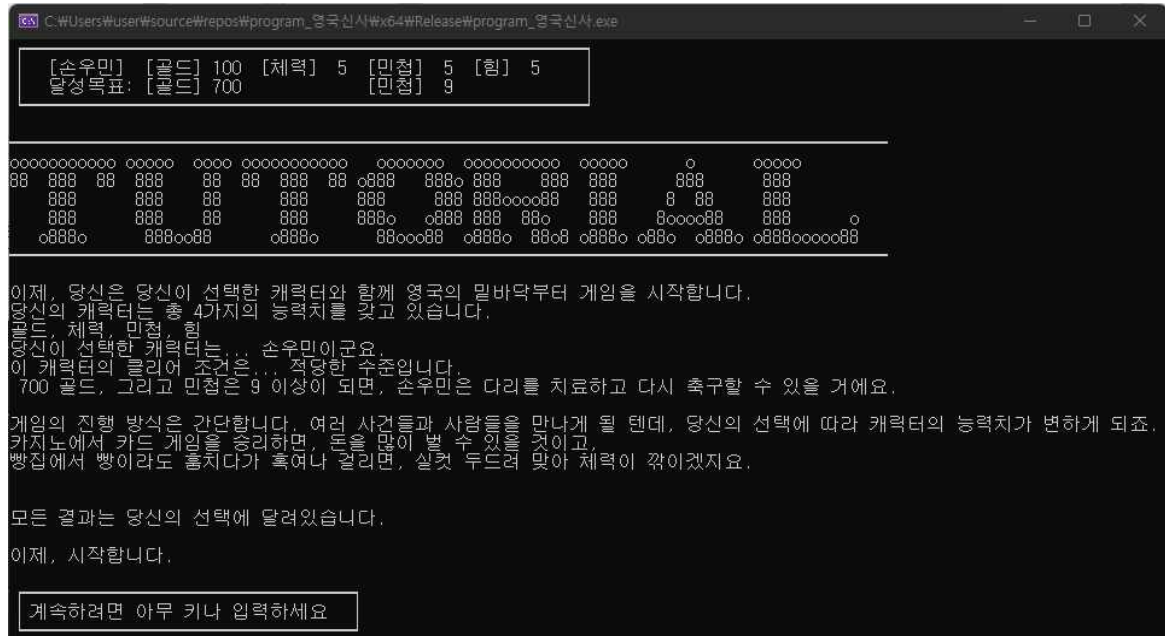
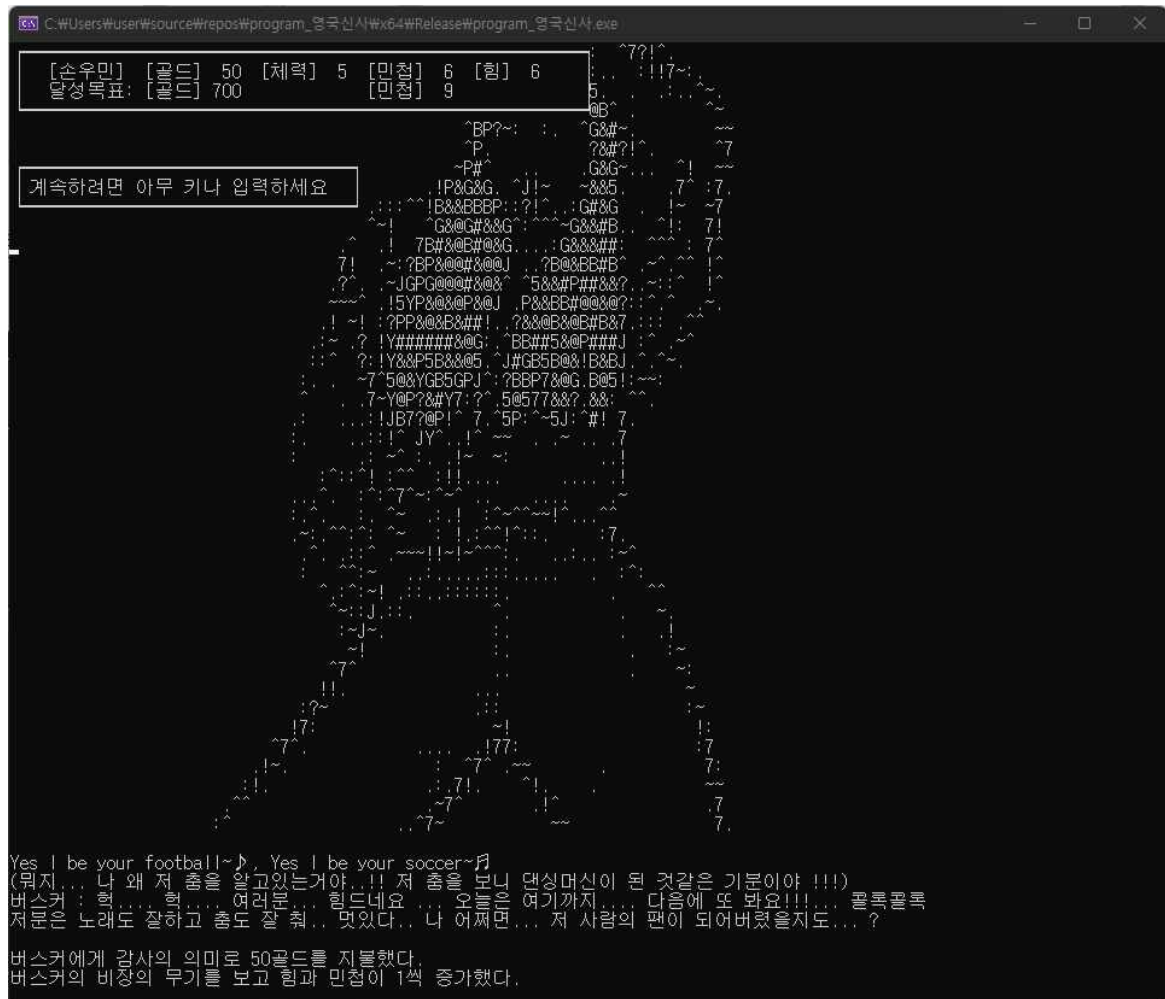


그림4. 이벤트 중 버스킹 이벤트 실행 화면



C:\Users\user\source\repos\program_영국신사\#x64\Release\program_영국신사.exe

C:\Users\User\source\repos\program_영국신사\#x64\Release\program_영국신사.exe



그림8. 카드뒤집기 게임 실행 화면

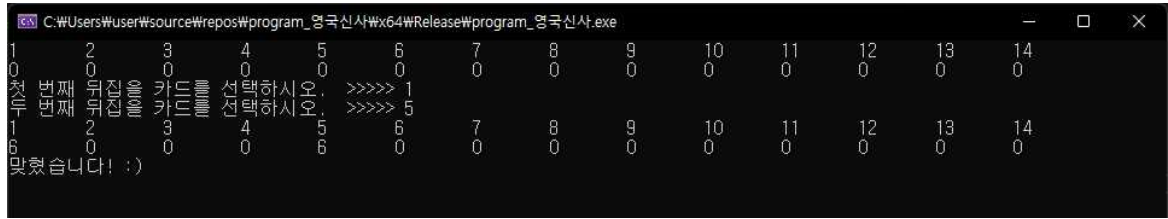


그림9. 팔씨름 게임 실행 화면

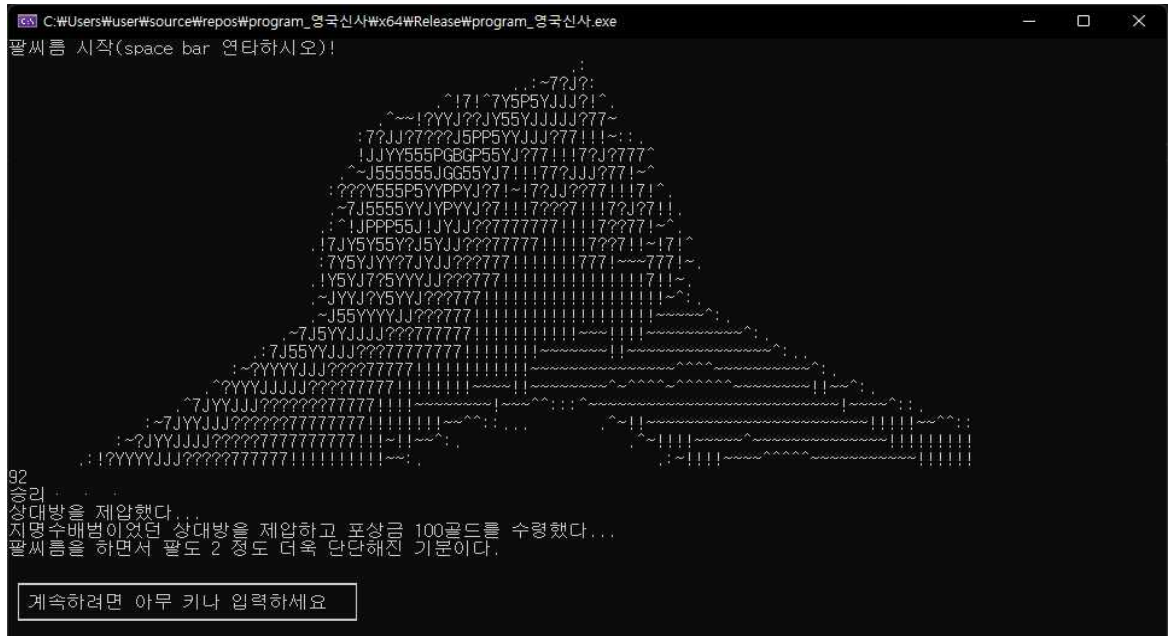


그림10. 할리갈리 게임 실행 화면



그림11. 슬롯머신 게임 실행 화면



그림12. 클리어 화면

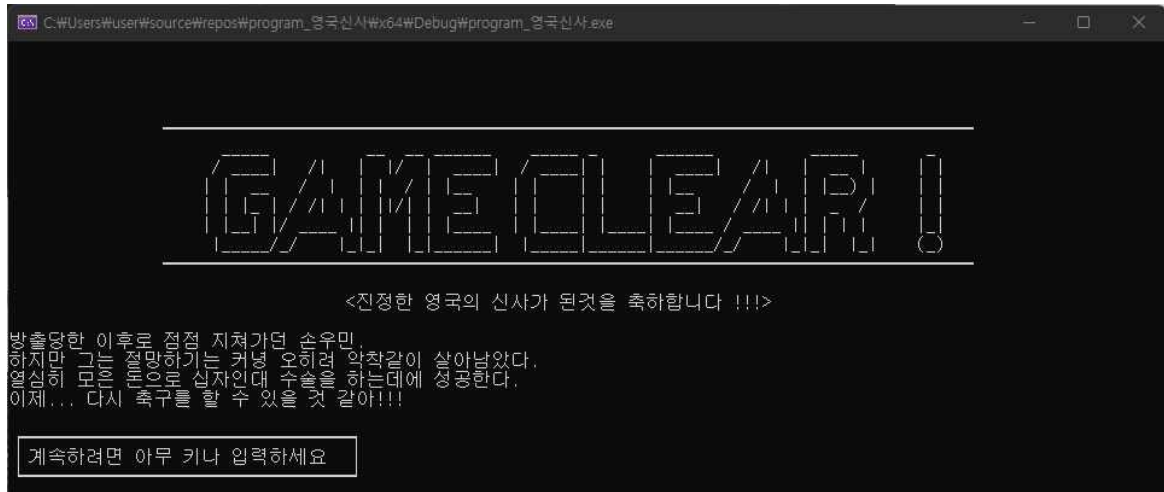


그림13. 배드엔딩 화면



8. 평가

8.1 정량적/정성적 목표달성도 평가

항목	목표값	달 성 른 (%)	비고
Visual C를 이용해 여러 라이브러리 함수들의 이해 및 응용	여러 조건문 및 반복문의 응용 및 기타 라이브러리 함수들의 사용 숙달	100%	
게임 개발 및 제작에 대한 학습 및 이해	1) cmd창 내부에서 게임처럼 보일 수 있도록 시각적인 효과를 주는 것 2) 사용자의 흥미를 극대화할 수 있도록 난이도 조절	100%	
설계한 게임의 구현을 위한 알고리즘 작성 능력 향상	1) 카드 게임의 구현을 위한 데이터 구조 학문의 이해. 2) rand()함수를 통한 난수 발생에 대한 이해 3) while(), if()함수를 사용해 조건문 및 반복문의 알고리즘 구현 이해	90%	예상치 못한 부분에서 반복문의 탈출이 불가능한 상황이 발생
라이브러리 함수를 사용하여 모듈화 프로그래밍 능력 향상	전체 프로그램을 모듈화하여 유기적으로 연결될 수 있도록 프로그래밍	100%	
모듈화 프로그래밍을 통하여 협동 프로젝트 수행	각 모듈마다 역할을 나눠 효율적으로 프로그래밍 수행	90%	효율적으로 역할 분담을 하는 것이 어려웠음

8.2 현실적 제한요소 달성도 평가

현실적 제한 요소	내 용 (Content)	달성 결과
경제	<ul style="list-style-type: none"> - 개발 제품의 가격 경쟁력 확보 - 낭비되는 메모리 없이 최소한의 리소스만을 사용하여 프로그램을 구현해 경제성 고려 	<ul style="list-style-type: none"> - 중복되는 변수나 함수 및 코드의 사용을 없애기 위해 최대한 모듈화하여 함수를 호출하는 방식으로 메모리 낭비를 줄임
편리	<ul style="list-style-type: none"> - cmd 창 내부에서 최대한 사용자의 가독성을 고려하여 개발 (문자열의 출력, 화면의 제어 등) 	<ul style="list-style-type: none"> - cmd창 내에서 게임인 것을 인지할 수 있도록 ASCII ART를 사용해 여러 글자 및 그림을 표현하였음 - sys_print()함수를 정의하여 그저 문자열을 출력하는 것이 아니라 마치 대사를 말하는 것처럼 표현
윤리	<ul style="list-style-type: none"> - 게임 내 사행성 요소가 포함되어 있기에 게임물관리위원회의 기준에 따라 윤리성 고려 (게임 이용 가능한 나이) - 악의적인 사용을 방지하기 위한 2차 창작의 금지 	<ul style="list-style-type: none"> - 게임 내 불법 도박 요소를 넣지 않았으며, 도박중독 상담 번호를 안내함.
사회	<ul style="list-style-type: none"> - 게임 중독 예방을 고려하여 게임 실행 도중 실행 시간의 알림 	<ul style="list-style-type: none"> - 난이도 조절을 통하여 게임의 플레이타임 자체를 줄이는 방식으로 프로그래밍함.

8.3 기능적 요구사항 달성도 평가

ID	요구사항	내용	우선순위	달성률
p_01	게임의 시작 및 튜토리얼	프로그램 실행 시 타이틀 화면을 출력하고, 게임 시작을 선택하면 게임의 배경을 설명해야 한다.	4	100%
p_02	캐릭터 선택 기능	각각 기본 능력치 및 클리어 목표가 다른 5가지 종류의 캐릭터를 사용자가 직접 선택한다.	5	100%
p_03	게임의 종료	게임 진행 상태 및 캐릭터의 능력치에 따라 게임의 엔딩이 달라져야 하며 여러 가지 엔딩을 포함하여 사용자에게 다양한 경험을 제공한다.	6	100%
p_04	캐릭터의 상태 알림	게임의 종료에 가장 크게 영향을 미치는 캐릭터의 4가지 요소를 직관적으로 사용자에게 알려주어야 한다.	7	100%
p_05	게임의 시스템(system)	cmd 화면 내에서 사용자에게 그저 문자열 출력을 하는 것이 아닌 여러 가지의 시각적인 효과를 주고, 직접 미니게임 및 이벤트를 플레이하여 몰입감을 높일 수 있어야 한다.또한, 사건의 발생 및 게임의 진행을 제어할 수 있어야 함	1	100%
p_06	여러 가지 미니게임 플레이	미니게임의 결과에 따라 캐릭터의 골드 및 체력에 영향을 주고, 게임의 난이도를 조정하여야 한다.	2	100%
p_07	여러 가지 이벤트의 발생	이벤트를 통해 사용자는 게임의 스토리를 알 수 있어야 하고, 결과에 따라 캐릭터의 골드 및 체력에 영향을 줄 수 있어야 한다.	3	100%

수행 내용		일정 (2주 단위)							
		1		2		3		4	
목표와 기준 설정	- 설계목표 및 주제 설정								
	- 소프트웨어 요구사항 파악 및 기능 블록도 작성								
합성	- 기능별 구현방법 결정								
	- 프로그램의 순서도 작성								
분석	- 목표달성가능성 확인								
	- 제한 요소들의 고려								
제작	- 프로그램 코딩								
시험/ 평가	- 시험 및 검증								
	- 재설계								
결과	- 결과보고 및 시연								

11. 결론

처음으로 팀 단위로 프로그래밍을 진행하면서 조원 모두가 시작부터 쉽지 않을 것이라고 생각했다. 혼자 만드는 프로그램은 자신이 생각한 대로 수정하고, 추가하고, 일정 또한 자유롭게 사용할 수 있지만 팀 단위의 프로젝트는 그렇지 않았다. 처음에 설계의 주제를 정하는 부분도 여러 회의 및 의견 교환을 통하여 정하였고, 프로그래밍을 진행하면서 팀원 모두가 동시에 상의하면서 프로그래밍을 할 수 있는 시간이 생각보다 많지 않았기 때문에 누군가가 프로그램 소스 코드를 수정하면 어떠한 부분을 수정했는지에 대해서 모두에게 전달해주지 않으면 효율성이 많이 떨어질 수 있다는 것을 가장 크게 느꼈다. 또한, 게임의 스토리 및 엔딩 조건, 이벤트에서 얼마나 체력을 깎고 골드를 얼마나 획득하고, 이런 게임의 밸런스 및 난이도적인 부분에서도 의견 조율함에 있어서 여러 난관이 있었다.

하지만 결국 완성된 결과를 보니 혼자서는 절대 이 기간 안에 구현할 수 없는 프로그램을 완성하였으며, 프로그래밍을 진행하면서 각자 잘하는 분야가 달랐기 때문에 서로의 약점을 보완하면서 프로그램을 완성할 수 있었다고 생각한다. 때로는 혼자 생각할 수 없던 것들도 회의와 소통을 하다 보면 번뜩이는 아이디어가 떠오르기도 했고, 혼자 해결할 수 없었던 부분도 팀원들과의 의사소통을 통해 해결할 수 있었다.

완성된 게임 자체만 놓고 평가하면, 처음 설계하고 싶었던 목표의 90% 이상은 구현한 것 같아 팀원 모두가 만족스러워했다. 다만 아쉬운 점은, 학기 중에 프로그래밍을 진행하는 것이다 보니 모두가 각자의 시간을 100% 프로그래밍에 투자할 수 없었기 때문에, 만약 다른 과목 없이 이 프로젝트만 수행했다면 더 좋은 결과가 나올 수 있지 않았을까 하는 아쉬움이 남는다.

부록

<07조_소스코드.zip>

*소스코드(.c), 헤더(.h), 노래(.wav) 파일 포함. 사이버캠퍼스 첨부