

게임 프로그래밍 자리 배치 프로그램

2021963057 장문용

목차

1. 원본 실행 화면
2. 업그레이드 화면 및 실행
3. 코드 소스 설명

원본 실행 화면

```
C:\Users\j2256\OneDrive\Desktop\VSCode\2025GameProgramming\1010\Seating_Program\Original\10_4_1.exe

자리배치 프로그램

좌석배치에 대한 행과 열을 입력해야 합니다.
학생수는 최대 100명입니다.(10X10)
행의 수를 입력하고 Enter>4
열의 수를 입력하고 Enter>4



|    |    |    |    |
|----|----|----|----|
| 1  | 6  | 9  | 5  |
| 4  | 13 | 7  | 3  |
| 15 | 8  | 10 | 16 |
| 12 | 2  | 11 | 14 |



-----
Process exited after 67.73 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
```

업그레이드 화면 및 실행

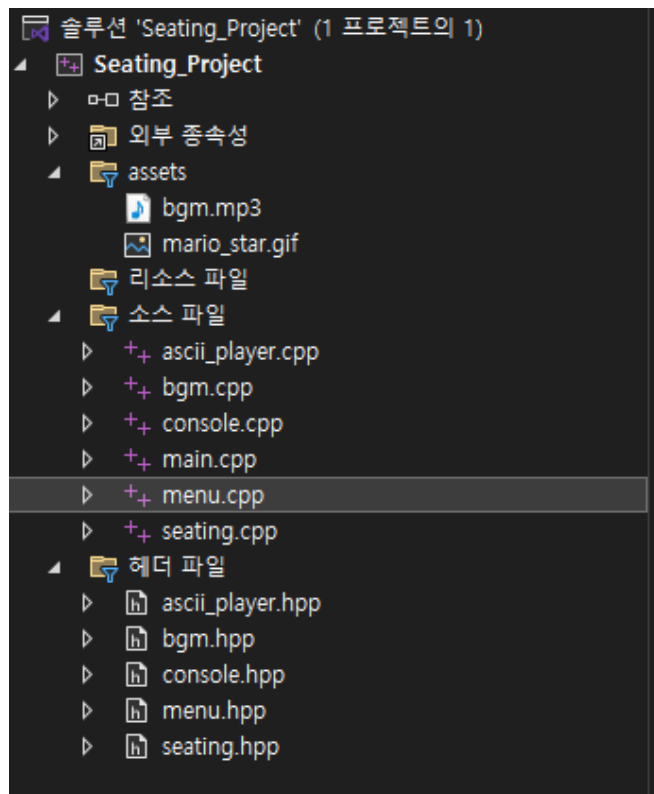
- 다운로드 경로

https://github.com/GUBBIB/2025GameProgramming/tree/main/1010/Seating_Program/Own

업그레이드 화면 및 실행

- 실행

코드 설명



6개의 소스파일
5개의 헤더 파일
2개의 미디어 파일

```

#pragma once
#include <windows.h>
#include <wchar.h>

namespace Console {

    // 전역 핸들
    extern HANDLE HOUT;
    extern HANDLE HIN;

    // 초기화: 코드페이지/타이틀/창 크기 등
    void init(int cols = 150, int lines = 50, const wchar_t* title = L"자리 배치 프로그램");

    // 기본 유틸
    void setAttr(WORD attr);
    // 포인터 위치 이동
    void gotoxy(int x, int y);
    // 포인터 숨김
    void hideCursor(bool hide);
    // 현재 포인터 위치 얻기
    void get_cursor_lbased(short* x, short* y);
    // x * y 위치의 width 칸 지우기
    void clearAt(int x, int y, int width = 80);
    // x * y 위치에 문자열 출력
    void wprintAt(int x, int y, const wchar_t* text);
    // 현재 콘솔 창 크기 얻기
    void getConsoleSize(int& cols, int& rows);
    // 문자열의 콘솔 셀 너비 계산
    int consoleCellWidth(const wchar_t* s);
    // 창 크기와 버퍼 크기 설정
    void setWindowAndBuffer(int winCols, int winLines, int bufCols, int bufLines);
    // 초기 설정 복원
    void restoreDefaults();
    // 한 줄 가운데 출력
    void wprintCentered(int y, const wchar_t* text);

}

```

Console.hpp

- 콘솔 및 포인터(커서) 관련 코드

```

#pragma once
#define NOMINMAX
#include "console.hpp"
#include <windows.h>
#include <gdiplus.h>
#include <vector>
#include <string>
#include <conio.h>
#include <algorithm>
#include <chrono>
#include <thread>
#include <iostream>

using namespace std;
using namespace Gdiplus;
using namespace Console;

// RGB 색 구조체
struct RGB { BYTE r, g, b; };
// 아스키 아트 프레임 구조체
struct AsciiFrame {
    vector<wstring> lines; // 한 프레임의 콘솔 라인들
    int delay_ms = 33;    // 프레임 딜레이

    vector<vector<RGB>> colors; // lines와 동일 shape, [row][col] 색
};

```

```

// 콘솔에서 GIF를 아스키 아트로 재생
// - gifPath : 재생할 GIF 경로
// - targetCols : 가로 최대 칸(문자) 수 (예: 120)
// - fps : 초당 프레임 (0이면 GIF 디폴트 딜레이 사용)
// - header : 맨 위 가운데 정렬 텍스트
// - footer : 맨 아래 가운데 정렬 텍스트
// - topMargin, midGap, botMargin : 헤더/영상/풋터 사이 여백(줄)
bool play_gif_ascii(const std::wstring& gifPath,
    int targetCols,
    int fps,
    const std::wstring& header,
    const std::wstring& footer,
    int topMargin = 2,
    int midGap = 2,
    int botMargin = 1);

// 비디오 다시 재생/일시 중지
bool ascii_player_should_stop();
void ascii_player_request_stop();
// 비디오 재생
void play_ascii_video(const std::wstring& gifPath,
    const std::wstring& header,
    const std::wstring& footer,
    int preferCols = -1, // -1이면 콘솔 폭 기준으로 자동
    int fps = 0, // 0이면 GIF 기본 딜레이 사용
    int topMargin = 1,
    int midGap = 2,
    int botMargin = 1);

// 내부 구현용
wchar_t pix2ch(unsigned char y);
// VT 모드 활성화
void enableVtColors();
// 콘솔 문자 크기에 맞춰 목표 크기 계산
void calcTargetSize(int srcW, int srcH, int maxCols, int maxRows, int& outCols, int& outRows);
// GIF 파일을 아스키 프레임 벡터로 변환
bool loadGifToAsciiFrames(const std::wstring& path, int targetCols, int maxRows, vector<AsciiFrame>& out);

```

ascii_player.hpp

assets 파일에 있는 사진/GIF 를
아스키 아트로 재생과 관련된
코드

bgm.hpp

```
#pragma once
#include <windows.h>
#include <mmsystem.h>
#include <string>
#include <sstream>
#include <cstdio>
#include <iostream>

namespace Bgm {

    /// exe와 같은 폴더 기준 기본 경로 (.\\assets\\bgm.mp3)
    std::wstring defaultPath();
    /// 파일을 열기만 함(재생은 안 함). 실패시 false.
    bool init(const std::wstring& mp3Path);
    /// 재생 시작. loop=true면 무한 반복.
    bool play(bool loop = true);
    /// 일시정지/재개/정지
    void pause();
    void resume();
    void stop(); // stop + close
    /// 0~1000 (WinMM 범위). 700 정도가 무난.
    bool setVolume(int volume0to1000);
    /// 특정 구간 반복 (밀리초 단위). 성공시 true.
    /// 예: setLoopRangeMs(0, 10000); // 처음 10초 반복
    bool setLoopRangeMs(unsigned long fromMs, unsigned long toMs);
    /// 루프 구간 해제(파일 전체 반복으로 복구)
    void clearLoopRange();
    /// 마지막 에러 문자열(있으면)
    std::wstring lastError();
    void startBgmTest();
}
```

bgm 관련 코드

menu.hpp

```
#pragma once
#include "console.hpp"
#include "seating.hpp"
#include <iostream>
#include <conio.h>
#include "ascii_player.hpp"

// 메인 화면 출력 함수
void main_screen();
```

메인 화면의 메뉴를 출력하는 코드

seating.hpp

자리 배치 관련 코드

```
#pragma once
#define NOMINMAX
#include "console.hpp"
#include <vector>
#include <windows.h>
#include <conio.h>
#include <random>
#include <iomanip>
#include <iostream>
#include <fstream>
#include <string>
#include <ctime>

using namespace std;
using namespace Console;

// 그리드 정보 구조체
struct Grid {
    int rows, cols; // 좌석 행/열
    int originX;    // 표 시작 X(1-based)
    int originY;    // 표 시작 Y(1-based)
    int cellW = 4;  // | + '——'
    int cellH = 2;  // 내용 1줄 + 구분 1줄
};

// 하단 메뉴 액션
enum class SeatAction { None, Reroll, Export, ChangeDir, Back };
```

```
// 좌석 배치 프로그램 메인 출력
void draw_check01();
// 그리드 그리기
void draw_check02(int L, int T, int c, int r);
// 한 좌석 렌더링
void renderSeat(const Grid& g, int r, int c, int val, bool selected);
// 좌석 배치 프로그램 메인
void seat_arrangement(int row, int column, int people);
// 하단 메뉴 표시 및 선택
SeatAction showBottomMenu(int left, int top);
// 하단 메뉴 블록 지우기
void clearBottomBlock(int lines = 4);
// 내보내기 위치 프롬프트
bool promptExportDir(std::wstring& exportDir);
// 경로 합치기
wstring joinPath(const wstring& dir, const wstring& file);
// 좌석 재배치 (locked[i] == 1 이면 고정)
void rerollSeats(vector<int>& seats, const vector<char>& locked);
// wstring을 UTF-8 string으로 변환
string w2u8(const wstring& ws);
// UTF-8 텍스트 파일로 내보내기
bool exportSeatsTxt_UTF8(const vector<int>& seats, int rows, int cols, const wstring& path);
// 축 레이블 그리기
void drawAxesLabels(const Grid& g, int rows, int cols, int step = 1);
// ESC로 취소 가능, 한 줄 입력 (공백·한글 포함)
bool readLineWithEsc(std::wstring& out, int x, int y);
```

```

int main() {
    SetWorkingDirToExeFolder();

    auto mp3 = Bgm::defaultPath(); // .\assets\bgm.mp3
    if (!Bgm::init(mp3)) {
        gotoxy(40, 40);
        wprintf(L"[BGM 실패] %s\n", Bgm::lastError().c_str());
    }
    else {
        Bgm::setVolume(700); // 0~1000
        Bgm::play(true); // 무한 반복
    }

    SetConsoleOutputCP(CP_UTF8);
    SetConsoleCP(CP_UTF8);

    Console::init(150, 40);
    main_screen();

    Bgm::stop(); // 종료 시 정리
    return 0;
}

```

main.cpp

```

#include "console.hpp"
#include "menu.hpp"
#include "bgm.hpp"
using namespace Console;

void SetWorkingDirToExeFolder() {
    wchar_t path[MAX_PATH];
    GetModuleFileNameW(NULL, path, MAX_PATH);
    std::wstring exePath(path);
    size_t pos = exePath.find_last_of(L"\\");
    if (pos != std::wstring::npos) {
        std::wstring dir = exePath.substr(0, pos);
        SetCurrentDirectoryW(dir.c_str());
    }
}

```

Main 호출 함수들이 있는 코드

감사합니다