

게임프로그래밍

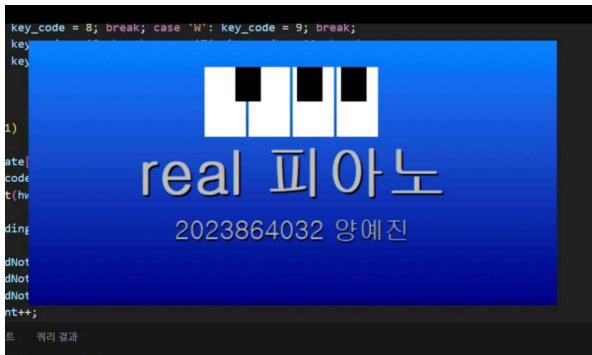
2023864032 양예진

<https://github.com/yejin1015/gameprogrammig>

WinAPI

- Windows 운영체제에서 애플리케이션을 만들기 위한 c언어 기반 함수들의 집합
- Windows의 창, 버튼, 메뉴 등 모든 그래픽 요소를 직접 제어할 수 있는 가장 기본적인 방법
- 윈도우 프로시저(WindowProc): 창에서 발생하는 모든 이벤트(메시지)를 처리하는 함수
- 메시지 루프(Message Loop): 운영체제로부터 "마우스 클릭!", "키보드 입력!" 같은 메시지를 받아 윈도우 프로시저에게 전달하는 무한 루프

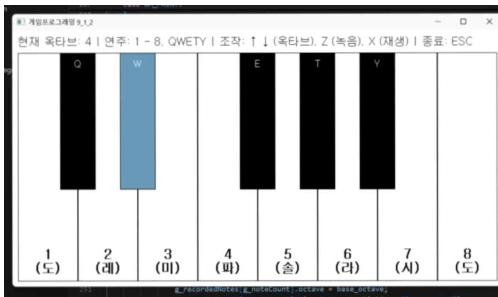
인트로 화면 및 소리



소리 - <https://freesori.tistory.com/19>하프등장음

폰트- 클로바 나눔손글씨 MaruBuri-Regular

피아노 구현



실제 피아노와 비슷한 모양으로 만들기 위해
흑건백건 구현

왼쪽상단에 기능 설명

사용자가 키보드(1~8, q~y)를 누르면 건반의
색이 파란색으로 바뀌면서 시각적인 효과

<https://ggonggi.tistory.com/entry/피아노-음계-소리>

폰트- 클로바 나눔손글씨 MaruBuri-Regular

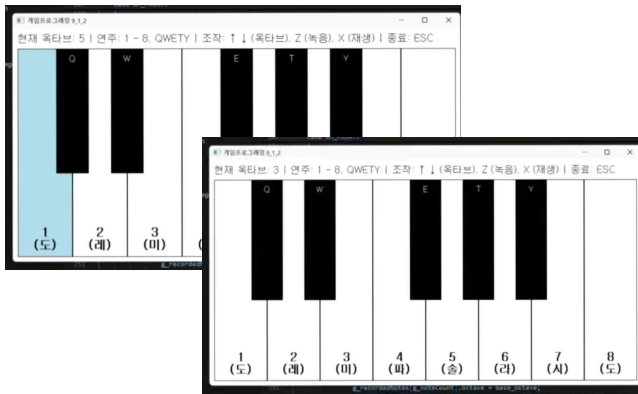
건반 클릭 시 색바뀜

```
case WM_KEYUP: // 키보드에서 손을 뗄 때
{
    int key_code = -1;
    // ... (눌렀던 키와 동일한 로직으로 key_code 찾기) ...

    if (key_code != -1) {
        // 100ms(0.1초) 후에 타이머 이벤트를 딱 한 번만 발생시킴
        SetTimer(hwnd, IDT_KEY_TIMER_BASE + key_code, 100, NULL);
    }
}
return 0;

case WM_TIMER: // SetTimer로 설정한 시간이 되면 발생
{
    // 키를 뗄 때 발생시킨 타이머인지 확인
    if (wParam >= IDT_KEY_TIMER_BASE && wParam < IDT_KEY_TIMER_BASE +
        NUM_TOTAL_KEYS)
    {
        int key_code = wParam - IDT_KEY_TIMER_BASE;
        key_visual_state[key_code] = 0; // 건반 상태를 원래대로 복원
        KillTimer(hwnd, IDT_KEY_TIMER_BASE + key_code); // 타이머 삭제
        InvalidateRect(hwnd, NULL, FALSE); // 화면 다시 그리기
    }
    // ... (후략) ...
}
```

옥타브 변경



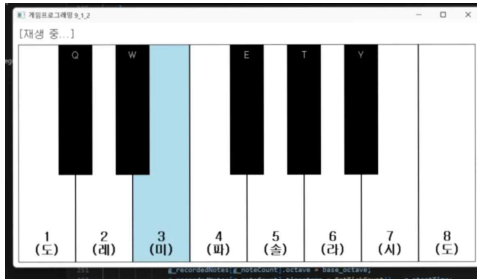
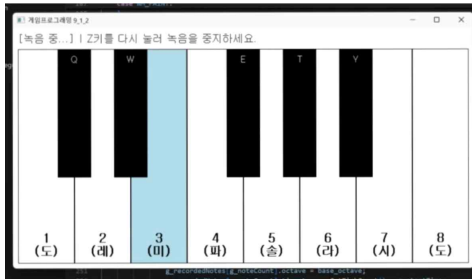
처음 화면은 4옥타브
위아래방향키 이용하여
3옥타브 5옥타브로 변경
가능

<https://ggonggi.tistory.com/entry/피아노-음계-소리>

건반 소리 재생 중요코드

```
// PlayNote 함수
void PlayNote(int key_index, int octave_override)
{
    // 3개 옥타브, 13개 건반에 해당하는 소리 파일 경로 배
    const char* sound_files[3][NUM_TOTAL_KEYS] =
    {
        { "sound/FX_piano09.wav", /* ... 3옥타브 소리 ... */ },
        { "sound/FX_piano01.wav", /* ... 4옥타브 소리 ... */ },
        { "sound/C5.wav",        /* ... 5옥타브 소리 ... */ }
    };
};
```

녹음 및 재생



<https://ggonggi.tistory.com/entry/피아노-음계-소리>

폰트- 클로바 나눔손글씨 MaruBuri-Regular

녹음 및 재생 코드

```
// WM_KEYDOWN 메시지 처리 부분
if (g_isRecording && g_noteCount < MAX_NOTES)
{
    // 1. 어떤 건반을 눌렀는지 저장
    g_recordedNotes[g_noteCount].key_index =
key_code;

    // 2. 어떤 옥타브로 눌렀는지 저장
    g_recordedNotes[g_noteCount].octave =
base_octave;

    // 3. [핵심] '언제' 눌렀는지 시간 간격을 저장
    g_recordedNotes[g_noteCount].timestamp =
GetTickCount() - g_startTime;

    g_noteCount++; // 다음 음표를 저장할 위치
로 이동
}
```

```
// WM_TIMER 메시지 처리 부분
else if (wParam == IDT_PLAYBACK_TIMER)
{
    // 1. 재생 시작 후 얼마나 시간이 흘렀는지 계산
    DWORD elapsedTime = GetTickCount() - g_playbackStartTime;

    // 2. [핵심] 현재 시간이 다음 연주할 음표의 시간보다 커졌는지 확인
    if (g_playbackIndex < g_noteCount && elapsedTime >=
g_recordedNotes[g_playbackIndex].timestamp)
    {
        // 3. '연주할 타이밍'이 되면, 저장된 정보를 꺼내서 연주!
        NoteEvent note = g_recordedNotes[g_playbackIndex];
        PlayNote(note.key_index, note.octave);

        // ... (시각 효과 코드) ...

        g_playbackIndex++; // 다음 음표를 연주하기 위해 인덱스 증가
    }

    // 4. 모든 음표를 연주했으면 타이머를 멈춤
    if (g_playbackIndex >= g_noteCount)
    {
        g_isPlaying = FALSE;
        KillTimer(hwnd, IDT_PLAYBACK_TIMER);
        // ...
    }
}
```

감사합니다

코드 제미나이

<https://ggonggi.tistory.com/entry/피아노-음계-소리>
폰트- 클로바 나눔손글씨 MaruBuri-Regular소리
<https://freesori.tistory.com/19>하프등장음

