

C언어 게임

학 번 : 2019975070

학 과 : 소프트웨어학과

이름 : 한재훈

목차

- 원본 코드 설명

1. 코드 설명
2. 코드 출력 결과
3. 코드 주요 함수 설명

- 업그레이드

1. 업그레이드 방향 및 게임 기획
 2. 코드 설명
 3. 추가 된 기능들
 4. 코드 출력 결과
-

원본 코드 설명

- 이 코드는 간단한 스톱워치 프로그램입니다.
기능은 다음과 같습니다:
- 시간을 시작, 일시 중지, 재시작하고 종료할 수 있습니다.
- 현재 경과 시간을 "시", "분", "초"로 표시합니다.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4 #include <math.h>
5 #include <conio.h>
6
7 void time_pass(time_t start);
8 void display_time(long nhour, long nmin, long nsecond);
9 void control_watch(void);
10
11 int main(void)
12 {
13     display_time(0, 0, 0);
14     getch();
15     control_watch();
16     return 0;
17 }
18 void control_watch(void)
19 {
20     char key;
21     time_t start;
22     start=time(NULL);
23     time_pass(start);
24     do
25     {
26         key=getch();
27         switch(key)
28         {
29             case 10 :
30                 time_pass(start);
31             case 32 :
32                 start=time(NULL);
33                 time_pass(start);
34             case 27 :
35                 exit(0);
36                 break;
37         }
38     }while(key!=27);
39 }
40
41 void time_pass(time_t start)
42 {
43     double hour, min, second;
44     long nhour, nmin, nsecond;
45     time_t now;
46     while(!kbhit())
47     {
48         now=time(NULL);
49         second=difftime(now, start);
50         hour= (second/3600.);
51         nhour= (long) hour;
52         min= fmod(second, 3600.)/60;
53         nmin= (long) min;
54         second= fmod(second, 60);
55         nsecond= (long) second;
56         display_time(nhour, nmin, nsecond);
57     }
58 }
59
60 void display_time(long nhour, long nmin, long nsecond)
61 {
62     system("cls");
63     printf("stopwatch\n\n");
64     printf("%.21d시 %.21d분 %.21d초 \n\n", nhour, nmin, nsecond);
65     printf("Enter:(시 작 /정 지), SPACE:재 시 작, Esc:종 료 \n");
66 }
67 }
```

원본 코드 출력 결과

```
stopwatch
```

```
00시 00분 00초
```

```
Enter:(시작/정지), SPACE:재시작, Esc:종료
```

원본 코드 주요 함수 설명

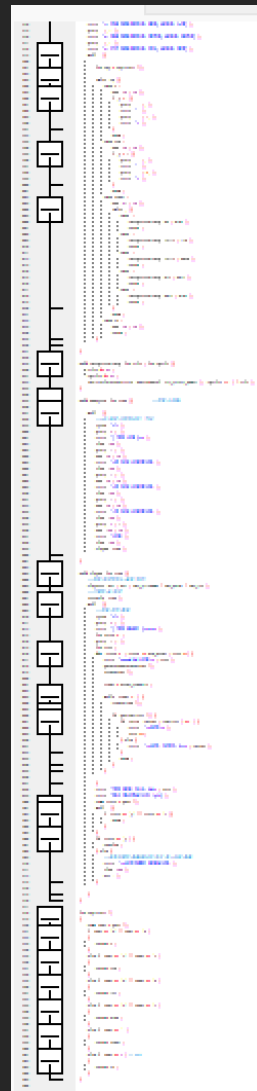
- **main():** 이 함수에서는 프로그램이 시작됩니다. 초기에 0시 0분 0초를 출력하고, 사용자의 입력을 기다립니다. 그리고 **control_watch()** 함수를 호출하여 스톱워치를 제어합니다.
 - **control_watch():** 이 함수에서는 스톱워치의 제어가 이루어집니다. 시작 시간을 저장하고, 사용자의 키 입력에 따라 동작을 수행합니다.
Enter(ASCII 값 10) 또는 Space(ASCII 값 32)를 누르면 시간이 재설정되고 스톱워치가 다시 시작됩니다.
Esc(ASCII 값 27)를 누르면 프로그램이 종료됩니다.
 - **time_pass():** 이 함수에서는 경과 시간을 계산하고 화면에 출력합니다. 현재 시간과 시작 시간 사이의 차이(second)를 계산하여 그 값을 "시", "분", "초"로 변환하여 출력합니다.
 - **display_time():** 이 함수에서는 주어진 '시', '분', '초' 값을 화면에 출력합니다.
-

업그레이드 방향 및 게임 기획

- 먼저 원본 코드에는 사용자의 입력 키 값을 입력 받아 스톱워치를 제어하는 기능만 있습니다. 저는 키를 입력 받는 것을 활용하여 만들 수 있는 패턴 기억해서 맞추기를 제작하겠습니다. 원본 소스에는 없었던 게임 인트로와 사운드 재생등을 추가하려고 합니다. 추가로 "텍스트 파일"을 읽어오는 기능등을 추가하겠습니다.
-

결과물 코드 설명

- 이 코드는 패턴 맞추기 게임 프로그램입니다.
- 원본 소스에는 단순히 스톱워치를 구현한 프로그램이기에 게임의 느낌이 없었으니 원본 코드의 타이머 기능만 가져와 게임의 느낌을 줘봤습니다.



주요 코드 설명 (헤더부분)

- **<stdio.h>:**
C 표준 입출력 함수를 사용할 수 있도록 해주는 헤더 파일입니다. **printf**, **scanf** 등의 함수를 사용하기 위해 포함됩니다.
 - **<stdlib.h>:**
표준 라이브러리 함수를 포함하는 헤더 파일입니다. **malloc**, **free**, **exit** 등의 함수를 사용할 수 있도록 합니다.
 - **<stdint.h>:**
정수형 데이터 유형을 위한 정의를 제공합니다. 정수형 변수의 크기와 부호 여부를 명확하게 정의할 때 유용합니다.
 - **<time.h>:**
시간 및 날짜 관련 함수를 제공하는 헤더 파일입니다. **time**, **ctime** 등을 사용하여 시간 정보를 다룰 수 있습니다.
-

주요 코드 설명 (헤더부분)

- **<math.h>:**
 - 수학 함수와 수학 상수를 포함하는 헤더 파일입니다. **sin**, **cos**, **sqrt** 등의 수학 연산 함수를 사용할 수 있습니다.
 - **<conio.h>:**
 - Microsoft Visual Studio 및 Windows 환경에서 사용되는 헤더 파일입니다. 콘솔 입출력 함수 및 그래픽 관련 함수를 제공합니다. (**_getch** 등)
 - **<process.h>:**
 - 프로세스 관리 및 제어 함수를 포함하는 헤더 파일입니다. **exit**, **system** 등의 함수를 사용하여 프로세스를 다룰 수 있습니다.
 - **<windows.h>:**
 - Windows API를 사용하기 위한 헤더 파일입니다. Windows 환경에서 시스템 및 그래픽 관련 작업을 수행하는 데 사용됩니다.
 - **<mmsystem.h>:**
 - Windows에서 멀티미디어 시스템을 다루기 위한 헤더 파일입니다. 음악 및 사운드 관련 기능을 사용할 수 있도록 해줍니다.
-

주요 코드 설명 함수 부분

- `init(void)`:
 - 프로그램 초기화 함수입니다. 콘솔 창 크기를 설정하고 커서를 숨깁니다.
 - `gotoxy(int x, int y)`:
 - 콘솔 화면에서 커서의 위치를 (x, y) 좌표로 이동하는 함수입니다.
-

주요 코드 설명 함수 부분

- **soundPlay(int index):**
지정된 인덱스에 따라 다른 음악 파일을 재생하는 함수입니다. **index** 인자에 따라 다른 음악 파일을 재생합니다.
 - **intro(void):**
게임 시작 전 인트로 화면을 출력하는 함수입니다. 게임 제목과 메뉴를 표시하며, 메뉴 선택을 처리합니다.
 - **backgroundChange(int color, int bgcolor):**
콘솔 화면의 배경색과 글자색을 변경하는 함수입니다.
-

주요 코드 설명 함수 부분

- **titledraw(void):**

게임 인트로 화면에서 게임 제목을 출력하는 함수입니다.

- **menudraw(void):**

게임 인트로 화면에서 메뉴를 출력하고 사용자의 입력을 받아 메뉴 선택을 처리하는 함수입니다.

- **infoDraw(void):**

게임 정보를 화면에 출력하는 함수입니다. 게임 조작법과 개발자 정보를 표시하며, ESC 키를 누를 때까지 대기합니다.

- **settingDraw(void):**

게임 설정 화면을 출력하고 사용자가 테마를 선택할 수 있도록 합니다. 선택한 테마에 따라 배경색과 글자색을 변경할 수 있습니다.

주요 코드 설명 함수 부분

- **ReadyView(int index):**

게임 시작을 준비하는 화면을 표시합니다. 3초 카운트다운 후 게임 플레이를 시작합니다.

- **PlayMap(int index):**

게임 플레이 화면을 표시하고 사용자로부터 입력을 받아 리듬 게임을 진행합니다. 게임 라운드를 반복하고 점수를 표시합니다.

- **KeyControl():**

키보드 입력을 받아와 해당하는 키에 대한 값을 반환하는 함수입니다. 게임 내에서 키 입력을 처리하는 데 사용됩니다.

주요 코드 설명 함수 부분

- **void generateRandomPattern(void)**

패턴 생성 함수입니다.

이 함수는 무작위 패턴을 생성하고, **pattern** 배열에 저장합니다.

패턴을 생성하고 싶은 방식에 따라 수정할 수 있습니다. 이 예제에서는 위 아래 패턴을 무작위로 생성합니다.

- **void showPattern(void)**

생성된 패턴을 화면에 보여주는 함수입니다.

pattern 배열에 저장된 패턴을 화면에 출력하여 사용자에게 표시합니다.

패턴을 보여주는 시간은 5초로 고정되어 있으며, **Sleep(5 * 1000);** 함수를 사용하여 일정 시간 동안 대기합니다.

-

주요 코드 설명 함수 부분

- **void updateTimer(void)**

남은 시간을 화면에 표시하는 함수입니다.

timer 변수에 저장된 시간을 화면에 출력하여 사용자에게 표시합니다.

출력 내용을 업데이트하기 위해 `\r`을 사용하여 현재 행의 처음으로 이동하고, **fflush** 함수를 사용하여 출력 버퍼를 비웁니다. 이를 통해 시간이 실시간으로 표시됩니다.

- **int getUserInput(void)**

사용자의 입력을 받는 함수입니다.

사용자가 키를 입력하면 입력된 키를 **userInput** 배열에 저장합니다.

Enter 키(ASCII 코드 13)가 입력되면 입력을 종료하고 1을 반환하며, 그 외의 키는 계속 입력을 받습니다.

입력을 받을 때 **getch** 함수를 사용하여 사용자 입력을 대기합니다.

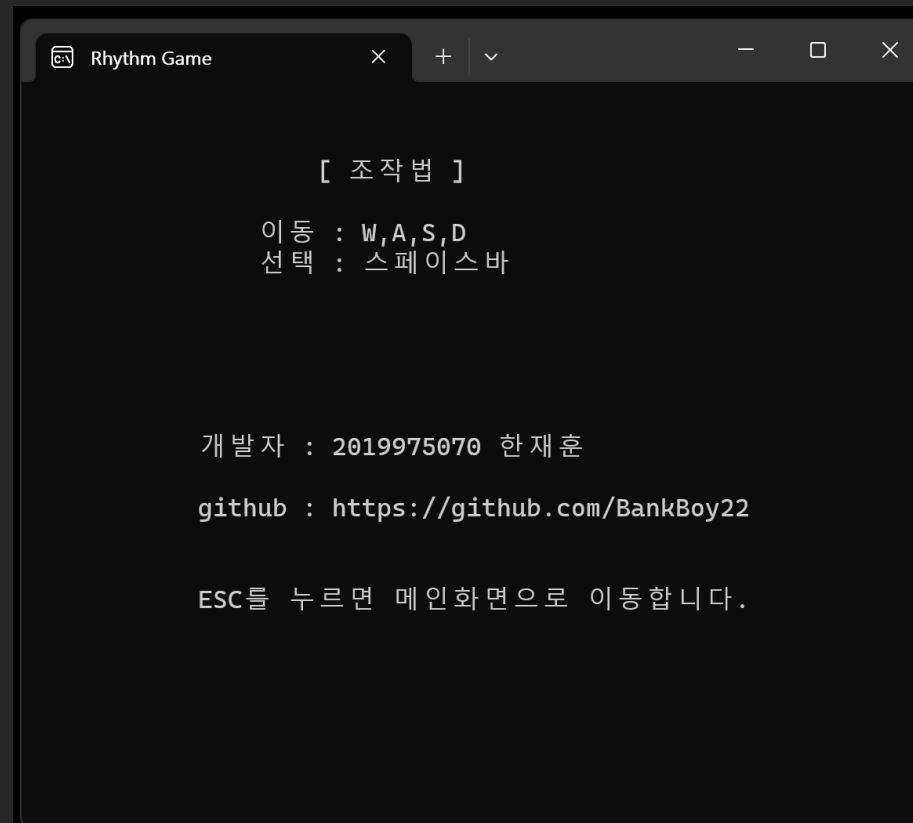
추가된 기능들

- 텍스트 파일을 읽어오는 기능 - "songlist.txt"를 불러오고 한줄 씩 출력
 - 음악 재생 기능 - Windows Multimedia API인 `winmm.lib`를 사용하며, WAV 파일을 재생
 - 인트로 화면 생성 - `intro()` 함수를 통해 인트로 화면 생성
 - 배경화면 및 폰트 색상 변경 - `backgroundChange(int color, int bgcolor)` 함수를 통해 테마 변경
 - 메뉴바 화살표 변경 애니메이션 - `gotoxy(int x, int y)` 함수를 통해 화살표를 이동 시킴
 - 커서 제거 - `init()` 함수 내에서 커서를 제거하는 기능을 넣었습니다.
 - 선택 및 이동 효과 사운드 - `Beep()` 기능을 통해 구현
 - 특정 키를 입력 받도록 하는 함수 - 기존 코드에서는 키를 따로 지정을 하지 않았지만 특정 키를 지정을 해줌
 - `Sleep()` 기능을 통해 대기시간 및 로딩 화면 구현
 - 스페이스바를 입력했을때 선택한 항목을 실행 시키고 `Esc`를 눌렀을때는 뒤로가기 구현
 - 패턴을 만들어주는 함수와 보여주는 함수 기능
 - 사용자의 패턴 입력을 받고 패턴의 배열과 비교하는 함수 기능
 - 게임 종료 구현
-

실행화면 (메인 화면)



실행화면 (게임 정보 화면)



실행화면 (게임 설정 화면)

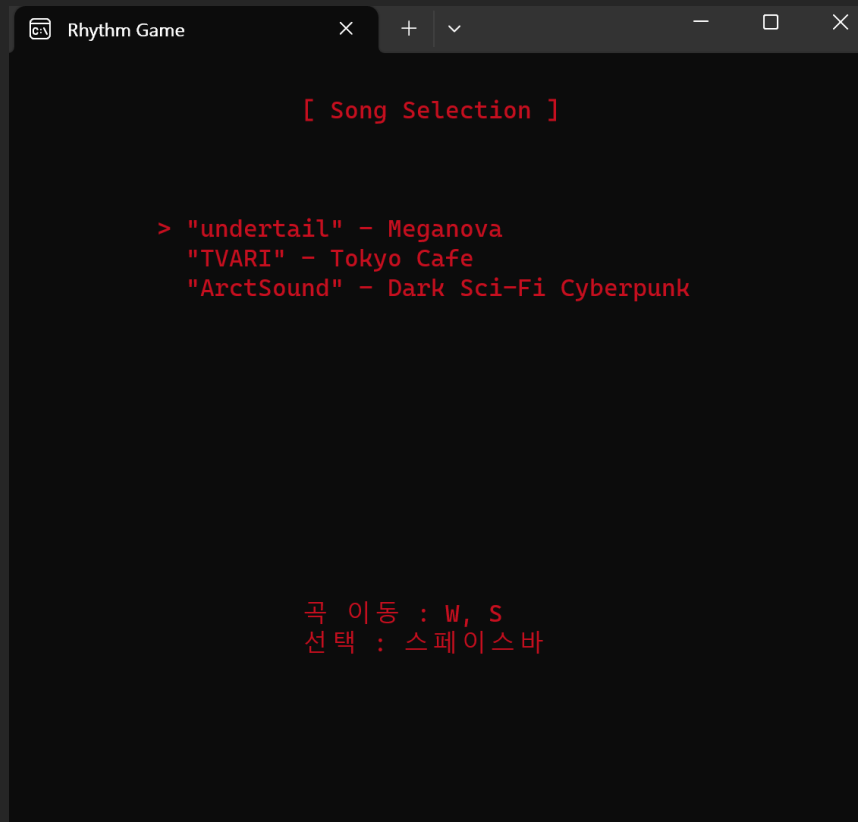
[게임 설정]

<테마 변경>

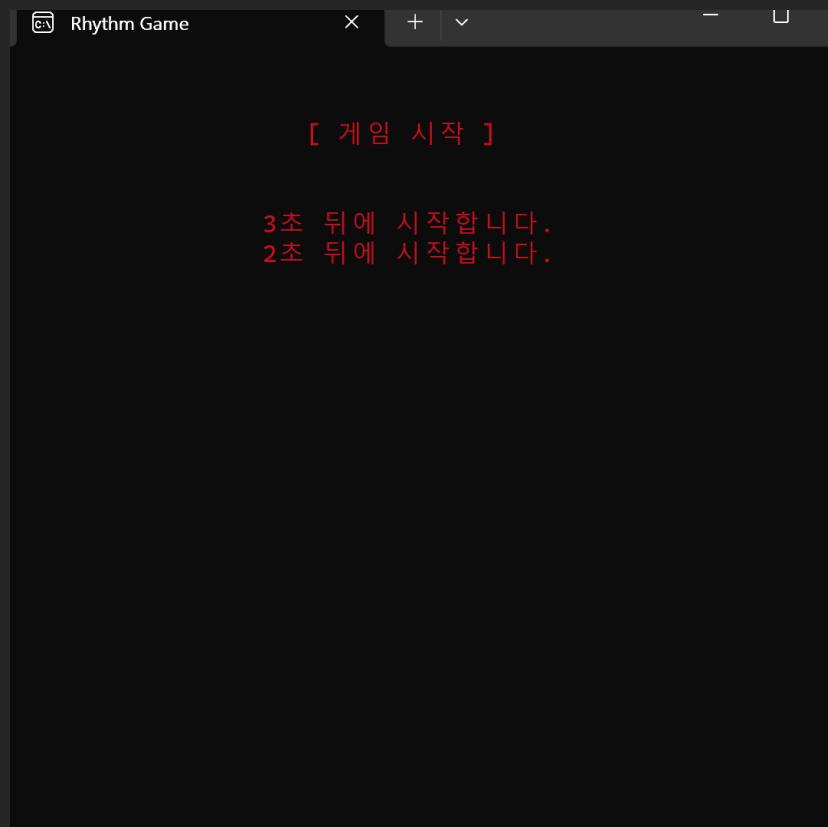
- > 1. 공포 테마(배경색: 검정, 글자색: 빨강)
- 2. 봄 테마(배경색: 하늘색, 글자색: 노랑)
- 3. 가을 테마(배경색: 갈색, 글자색: 노랑)
- 4. 겨울 테마(배경색: 하얀색, 글자색: 파란색)
- 5. 기본 테마(배경색: 검정, 글자색: 하양)

바꾸고 싶은 테마의 번호를 선택한뒤 스페이스바를 누르시오.

실행화면 (곡 선택 뷰)



실행화면 (게임 준비 뷰)



실행화면 (게임 실행)

[게임 플레이]

Round 1 시작!

패턴을 똑같이 입력하세요
남은 시간: 10 초

아쉬운점

- 시간이 너무 부족해서 원래의 목표였던 리듬게임의 구현을 다 하지 못하여서 패턴을 외우고 패턴을 똑같이 입력받는 형식의 게임으로 변경하였는데 이게 굉장히 아쉽습니다.... 그리고 .wav 형식의 파일을 사용하면 문제점이 용량이 너무 커진다는것인데 이를 해결하기 위해 mic 라는 또다른 window api를 활용할 수 있습니다. 그러나 너무 복잡한 개념이라 짧은 시간안에 이 부분을 제대로 공부하고 구현해내기가 힘들었습니다. 기회가 된다면 mic를 통해 mp3를 활용하고 중복으로 음악을 재생하게끔 하여 효과음 역시 추가로 넣고 싶습니다. 그리고 기본적인 UI 역시 시간 상 제작하지 못하였는데 이 부분 역시 더 추가하고 싶습니다. 마지막으로 여러번 시도 하다가 못한 방법인데 비트맵 이미지 역시 활용 해 보고 싶습니다.
-

감사합니다.

- -참고 자료-
 - 대부분의 자료들은 교수님의 예제들을 참고하였습니다.
 - -가져온음악출처-
 - pixabay
-