

# HW1

과목명: 게임프로그래밍

담당교수: 박종승 교수님

학과: 컴퓨터공학부

학년: 4

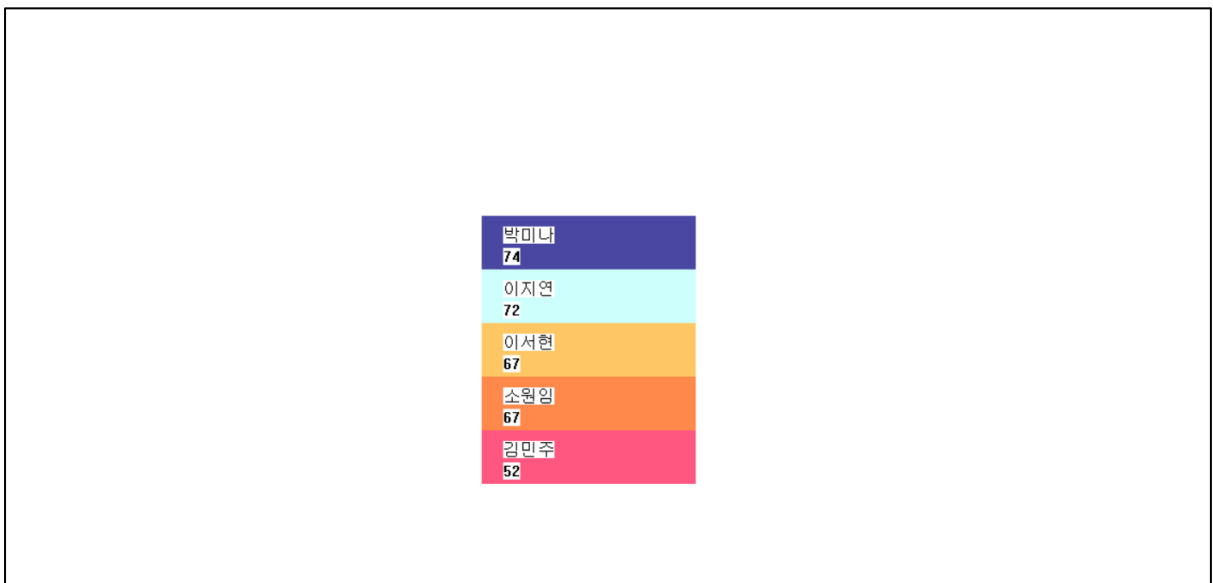
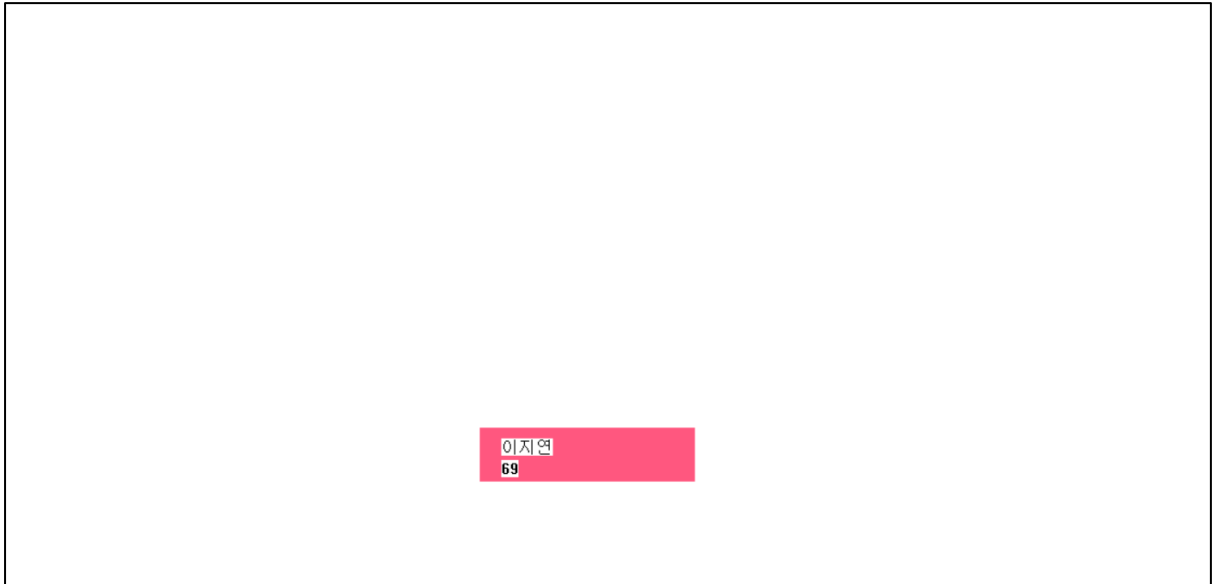
학번: 201801567

이름: 성은지

제출일: 2021.03.26

## 키 조작법

1. 왼쪽 마우스 클릭 시 학생 상자 하나 추가



2. ESC 클릭 시 종료

## 마우스 이벤트 처리

### 1. 학생 구조체

학생 정보 구조체는 성명(string)과 성적(int) 두 필드로 이루어져 있다. 이 구조체는 student.h에 저장되어 있다.

```
#include "stdafx.h"
using namespace std;

// 학생 구조체
struct student {
    string name;
    int score = 0;
};
```

### 2. 학생의 성명

학생의 성명은 3자리 한글이다. 이것은 windowMain.cpp에 있다.

```
#define MAX_COUNT 8
string name[STUDENT_NUM] = { "성은지", "이지연", "이서현", "조유진", "허가현", "김민주", "유현아", "이동은", "소원임", "박미나" };
```

### 3. 학생의 성적

학생의 성적은 50에서 99 사이의 정수이다. 학생 벡터를 하나 만들 때 점수는 랜덤으로 생성된다. 'rand() % 50'을 하면 0~49까지 정수가 생성되기 때문에 +50을 하면 범위에 충족된다.

```
student tmpStudent;
srand((unsigned)time(NULL));
tmpStudent.name = name[rand() % (STUDENT_NUM - 1)];
tmpStudent.score = rand() % 50 + 50;
```

#### 4. sort 알고리즘 사용

학생 데이터가 벡터에 추가될 때마다 학생 정보들이 성적에 따라 정렬되도록 `sort()`를 사용한다. 이를 위해 `algorithm`을 추가한다. 그림을 그리기 전에 먼저 `sort`를 한다. `struct`를 정렬하기 때문에 `stable_sort()`를 사용한다. 아래와 같은 `cmp()`를 사용하면 학생들의 정보가 점수가 낮은 순서로 정렬된다.

```
#include <algorithm>
```

```
case WM_PAINT:
    hdc = BeginPaint(windowHandle, &ps);
    stable_sort(vStudent.begin(), vStudent.end(), cmp);
    drawRectangle(windowHandle, hdc);
    EndPaint(windowHandle, &ps);
    break;
```

```
bool cmp(const student& p1, const student& p2) {
    if (p1.score < p2.score) return true;
    else return false;
}
```

#### 5. 학생 정보는 최대 8개까지만 허용된다.



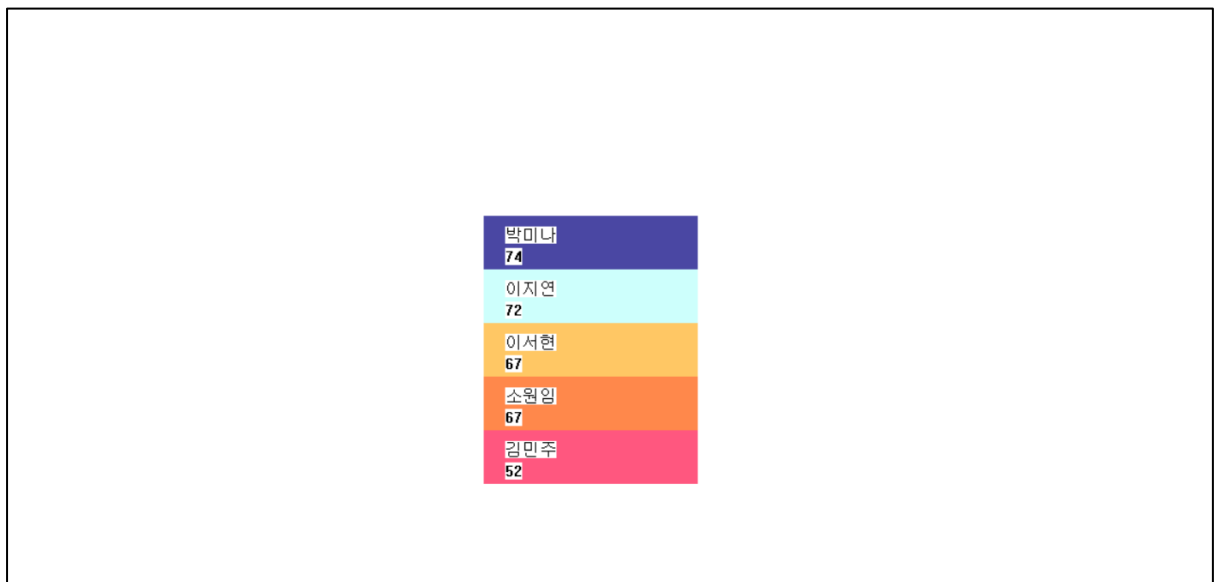
```
// 8개가 최대
if (vStudent.size() == 8) {
    ::MessageBox(0, L"Size of the struct is 8. So you can't make it.", L"Out of Size", MB_OK);
    return 0;
}
```

## 디스플레이

각 상자 안에는 한 학생의 정보가 표시된다.



각 상자의 컬러를 다 다르게 표현한다.



학생 정보는 벡터 컨테이너에 보관된다.

```
vector<student> vStudent;
```

아래의 상자가 벡터의 앞쪽에 해당하고, 위의 상자가 벡터의 뒤쪽에 해당한다. 상자를 표기할 때 사용되는 코드는 다음과 같다. 벡터 컨테이너가 시작하는 곳부터 끝까지 순서대로, 상자가 위로 올라가면서 색이 채워진다.

```
for (vector<student>::iterator iterDraw = vStudent.begin(); iterDraw != vStudent.end(); iterDraw++) {

    wstring tmpName = s2ws(iterDraw->name);
    wstring tmpScore = s2ws(to_string(iterDraw->score));

    LPCWSTR name = tmpName.c_str();
    LPCWSTR score = tmpScore.c_str();

    switch (colorNum) {
        ...
    }

    HBRUSH myBrush = CreateSolidBrush(color);
    HPEN myPen = CreatePen(PS_SOLID, 1, color);
    HGDIOBJ oldBrush = SelectObject(hdc, myBrush);
    HGDIOBJ oldPen = (HBRUSH)SelectObject(hdc, myPen);

    Rectangle(hdc, rectx, recty, rectx + 200, recty + 50);

    TextOut(hdc, rectx + 20, recty + 10, name, sizeof(name) - sizeof(' '));
    TextOut(hdc, rectx + 20, recty + 30, score, to_string(iterDraw->score).length());

    recty -= 50;

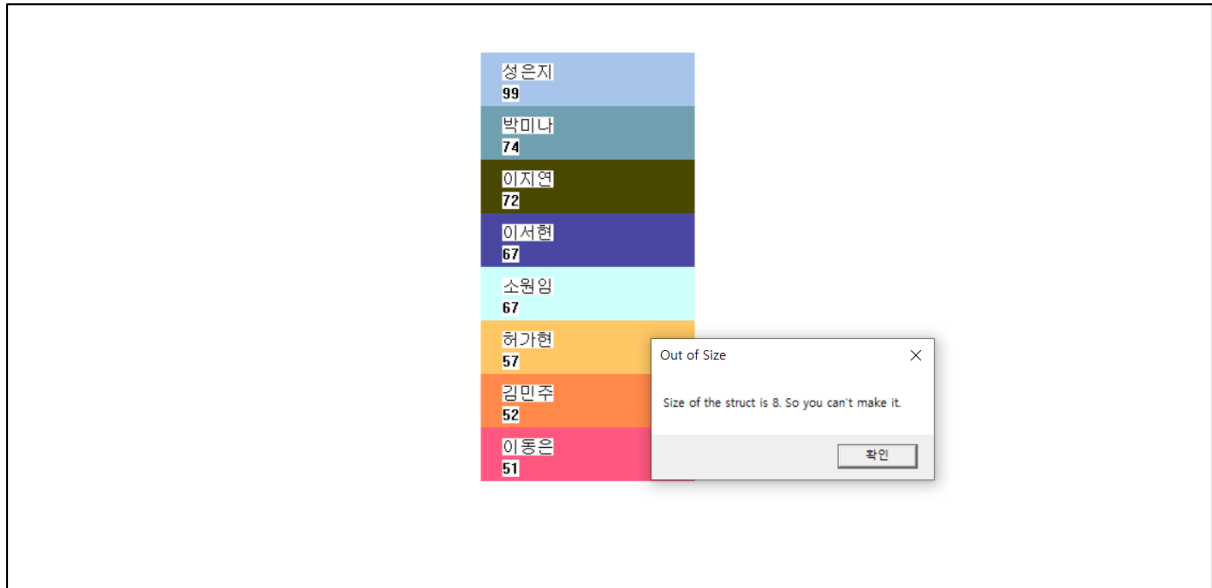
    SelectObject(hdc, oldBrush);
    SelectObject(hdc, oldPen);
    DeleteObject(myBrush);
    DeleteObject(myPen);

    colorNum++;
}
```

## 메시지창 표시

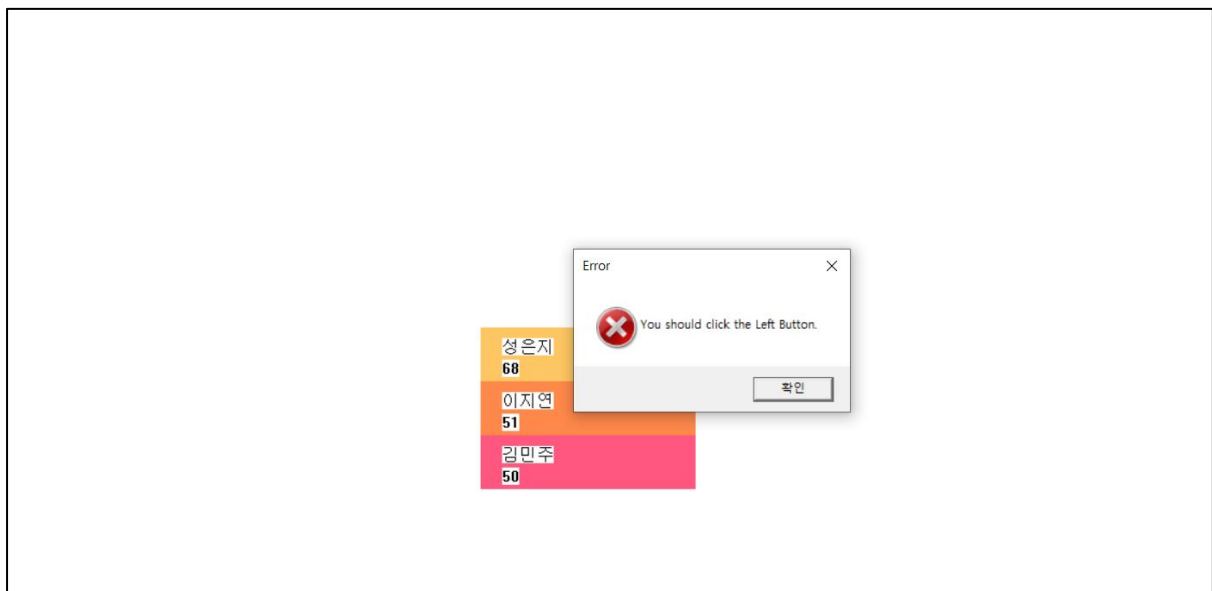
### 1. 8개 이상 추가할 수 없을 때

최대 크기는 8이기 때문에 더 이상 추가할 수 없다는 메시지를 띄운다.



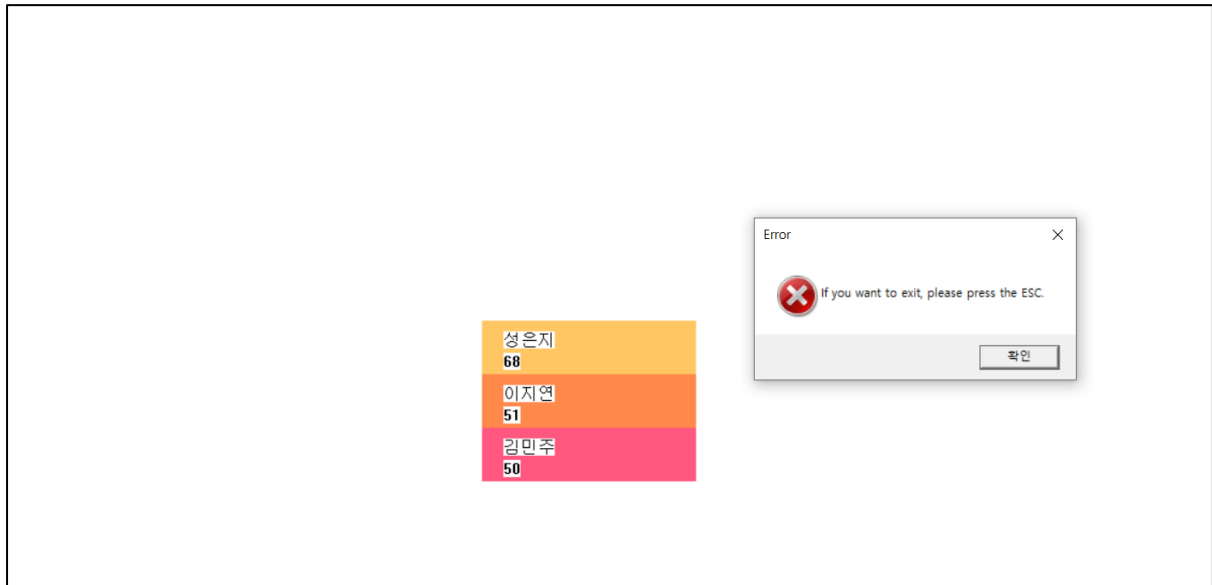
### 2. 오른쪽 마우스 버튼이 눌렸을 때

사용자가 왼쪽 마우스 버튼을 누르도록 유도한다.



### 3. ESC가 아닌 키가 눌렸을 때

'고길 원하면 ESC를 눌러주세요'라는 메시지를 띄운다.



### 4. ESC가 눌렸을 때

화면이 꺼진다.