

객체 지향 프로그래밍 ASSN4

20180285 이신범

무은재학부

nm160

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

나는 이 프로그래밍 과제를
다른 사람의 부적절한 도움 없이
수행하였습니다.
이신범

(1) 개요

어싸인 4는 C++ 중 '연산자 오버로딩'에 대한 내용을 잘 숙지하고 있는지를 묻는 문제와 상속의 개념을 이용해 두 클래스를 virtual method로 묶어줄 수 있는지에 대한 문제, 그리고 QT의 전반적인 GUI의 구현과 SIGNAL & SLOT에 대한 지식을 묻는 문제였다.

1. 어싸인 4를 수행하기 위해 알아야 했던 사항은 다음과 같다.
2. Istream, ostream 연산자 오버로딩을 이해하고 파일 입출력을 실행할 수 있는가?
3. QT의 기본적인 CLASS들과 SIGNAL, SLOT 개념을 이해했는가?
4. 상속을 이용해 두 클래스를 virtual method로 선언할 수 있는가?

(2) 사전 지식

1. 연산자 오버로딩

.seq라는 파일에 대해 파일입출력을 진행할 수 있도록 연산자를 오버로딩 시켜주는 작업을 진행했다.

Seqio.h , .cpp

Songio.h, .cpp 에는 << 연산자와 >> 연산자를 포함시켜줬고

Seq.h, .cpp

Song.h, .cpp 에는 == 연산자를 friend로 포함시켜줬다.

(이 이유는 아직도 잘 모르겠습니다. 단지 인터넷에 존재하던 자료 중, == 연산자를 friend a operator == (...) 로 정의하는 것들이 있었습니다.)

2. QT class에 대한 이해

본 QT 프로그램 구현을 위해 사용된 QT class는 다음과 같다

QTimer , 일정한 주기에 맞춰 slot을 실행하는 위젯임을 이해했다.

Qsoundeffect , 저장된 URI에 따라 .wav 파일을 실행해줄 수 있었다.

[이 때 어려웠던 점은, Qsoundeffect * a = new Qsoundeffect 와 같이 동적 할당을 해주어야 a->play() 에서 소리가 났다.]

본 QT프로그램 구현을 위해 사용된 GUI class는 다음과 같다.

Qtable, 음표를 출력할 표를 화면에 나타냈다. 적절한 slot을 받아 클릭된 cell의 row와 column을 인식할 수 있었다.

QPushbutton, 버튼이 click 되면 slot에 해당하는 동작을 진행한다.

QtabWidget, drumtrack과 melodytrack을 바꿔주는 역할을 진행했다.

QColor , cell이 클릭되면, cell을 색칠해줌으로써 cell에 note가 존재하는지 존재하지 않는지를 판단했다.

말로 설명하니 다소 쉬운 것 같지만, 사실 익숙치 않은 프로그램을 통해 GUI와 코딩을 모두 하려다 보니 어싸인을 도와주신 조교님과 친구들이 없었다면 혼자서는 결코 하지 못했을 것 같다.

(3) Class 구성

1. Prob 1에 대해서는 as4:: 의 하위항목인 seqio.h, .cpp과 songio.h, .cpp 그리고 seq.h, .cpp 과 song.h, .cpp에 올바른 연산자 오버로딩을 작성해줌으로써 해결했다.

이 때 fin과 fout을 해주기 위해 for문을 돌렸는데, sequence의 size를 file에 저장할 때 같이 저장해놓음으로써 for문을 돌릴 횟수를 결정할 수 있었다.

== 연산자 오버로딩 같은 경우, == 연산자를 선언할 위치를 찾는 것이 아직도 잘 이해가 되지 않는다.. 어려웠던 점은, const Song& song 으로 선언되어서, song.GetMelodySeq()->getordered()를 부를 수 없었는데, 이는 getordered() 함수가 const로 선언되지 않았기 때문이었다. 이를 해결하기 위해 const로 선언된 getordered를 하나 더 선언해주던가, 혹은 Song1을 const_cast 해주는 방법이 존재했다. 나는 const_cast를 해주는 방법을 이용했다!

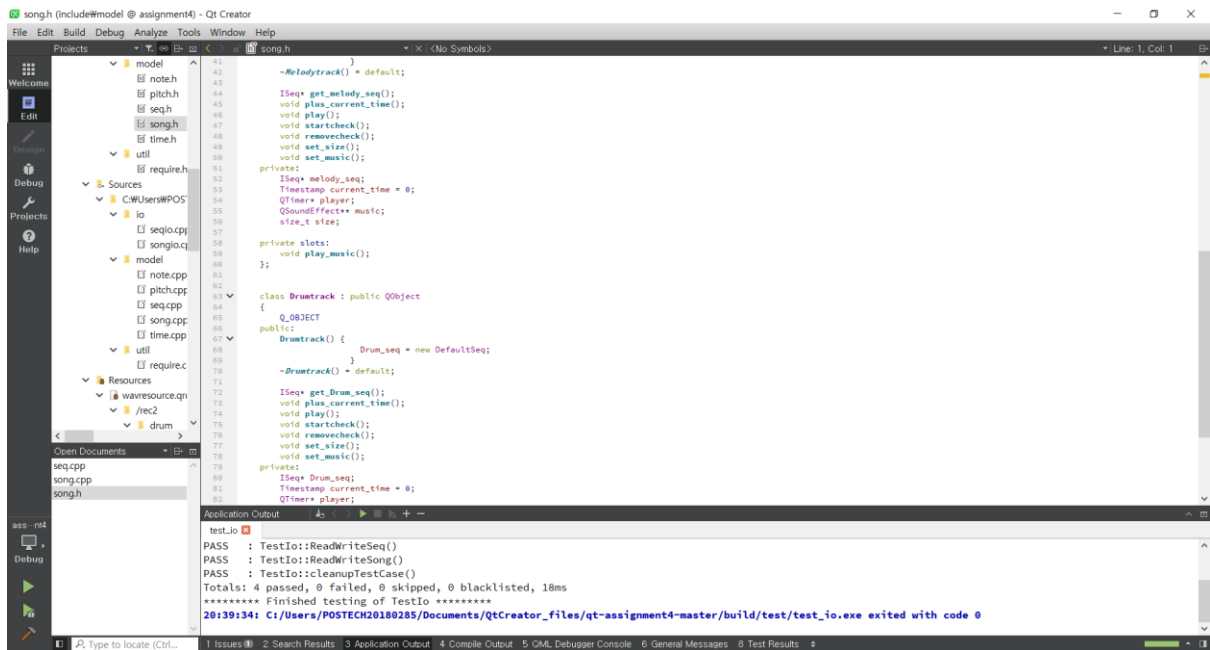
2. Prob 2에 대해서는 QT의 widget? 들인 Qsoundeffect와 QTimer를 이용하는 것이 중요했다. QTimer에서 1유닛 (125ms)마다 시그널을 발생시키면, 이를 받은 SLOT에서 Qsoundeffect를 통해 음표를 play() 하거나 stop()할 수 있도록 선언해주었다.
3. Prob 3에 대해서는 QT의 class design form을 이용했다. Qtablewidget을 이용해 note를 표시할 row 16, column 40인 칸을 선언해주었다. 또한, Qtablewidget은 SLOT을 통해 mouseclick을 받을 수 있는데, 이를 통해 note를 생성시켰다가 삭제하는 기능을 도입했다.

2-3. 3번의 play기능과 2번의 Melodytrack, Drumtrack을 연관시켜주는 것이 조금 어려웠는데,

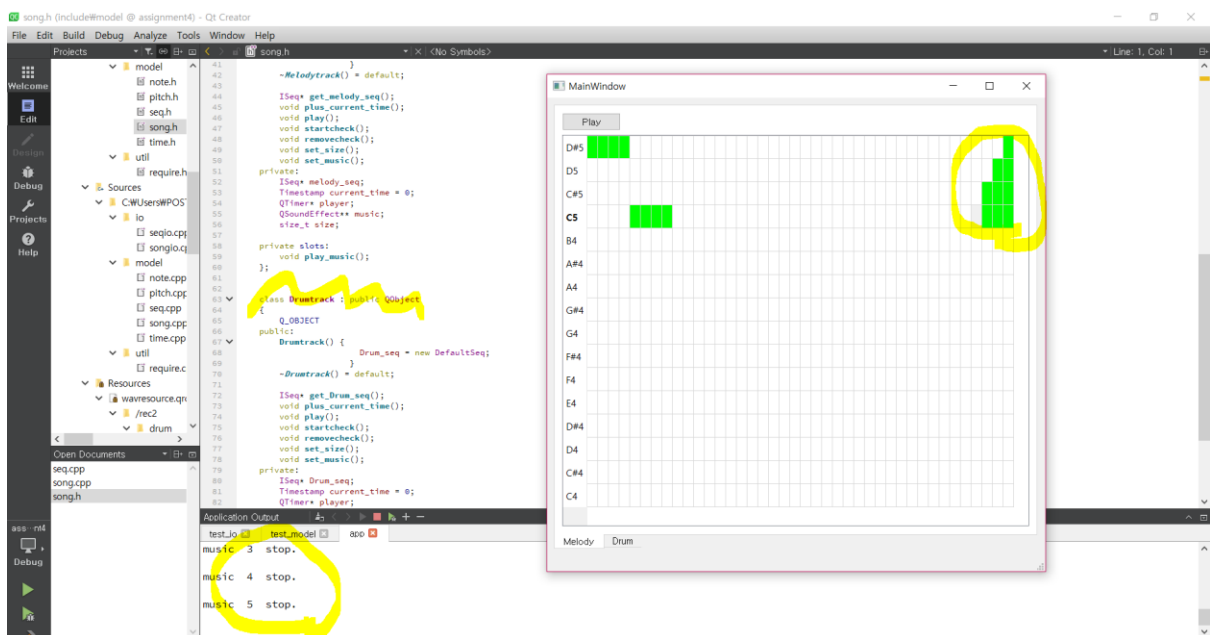
이것 역시 QTimer를 통해 색칠된 cell을 인식받아 Qsoundeffect를 play하고, stop하는 기능을 구현해야 했다.

(4) 구현에 대한 사진 첨부

Prob 1



Prob 2 and Prob 3



Polymorphism의 경우 구현하지 않았습니다.

Play를 누르면 melody_tab에서는 Melody의 음만 출력되고, Drum_tab에서는 Drum의 음만 출력된다. 또한, 왼쪽 끝과 오른쪽 끝의 경우 tablewidget의 가장자리라 오류가 생기기에 쉬운 위치인데, 그러한 자리에도 모두 구현을 완료했음을 확인할 수 있다.

(5) 토론

1. Design class를 프로젝트에 넣어주는 것이 어려웠다. 이를 해결하기 위한 방법은 2가지가 존재했다. 첫번째는 조교님께서 lms에 공지해주신 방법으로, app 프로젝트에서 ui form을 넣고, .h와 .cpp 코드를 수정해 ui파일과 연결시키는 것이었다. 두번째는 ui class를 만들면 .ui파일과 더불어 .h와 .cpp 파일이 동시에 생성되는 것을 이용한다. 이를 위해 기존에 as4에 존재했던 widgets안의 헤더와 cpp 파일을 제거하고, app 프로젝트에 ui class를 만들어준다. 그리고 app 프로젝트에 만든 .h와 .cpp파일을 as::widgets 디렉토리로 직접 넣어줌으로써 이를 해결했다.
2. 노트가 이어지는 경우 예 => 11111111의 경우 4유닛의 note가 2개 이어진 경우였다. 이를 인지해 Qsoundeffcet를 2번 실행시켜줘야 했는데, 이는 초기에 app에서 gui에 존재하는 note를 읽을 때 색칠된 부분에 다다르면 그 note를 시퀀스에 저장하고 +4칸부터 다시 check하게 함으로써 해결했다.

(6) 결론

1. 이번 어싸인을 통해 Qt를 이용해 처음으로 gui 디자인을 해볼 수 있어 좋았다. 객체 팀프로젝트를 준비하며 Qt를 어떻게 써야하나 감이 잡히지 않아 막막했는데, 그래도 assn4를 해결하는 과정에서 양질의 문제들을 많이 해결하다보니 Qt를 사용하는 방법을 알게된 것 같다(다른 Qt class의 경우 인터넷에 검색해서 비슷한 방법으로 사용해 주면 되므로)