

CSED232 Object-Oriented Programming

Spring 2019

Programming Assignment #4

Problem Solving

무은재 새내기학부

20180038

박형규

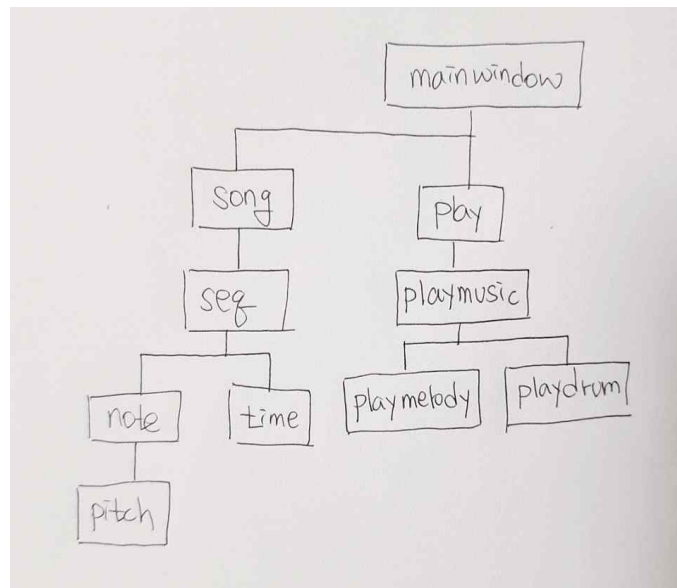
POVIS ID : hyeongkyu

Honor Code : 나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

1. 프로그램 개요

이번 과제는 총 3개의 Problem으로 구성되어있다. Problem1은 Operator Overloading을 사용하여 Qt의 Test를 통과시키는 Problem이고, Problem2와 3은 Qt Framework를 사용하여 GUI 어플리케이션 제작을 실습한다. 이번에 구현할 GUI 어플리케이션은 MUSIC SEQUENCER로 음표를 조작하여 음악을 작성하고 편집할 수 있게 해주는 소프트웨어이다. 이번 프로그램에서는 마우스 클릭을 통해 음표를 생성, 삭제, 재생할 수 있는 피아노 롤 형태의 간단한 소프트웨어를 구현한다.

2. 전체 구조



이번 프로그램에서는 mainwindow.cpp/h가 Ui를 관리하며 프로그램을 총괄적으로 관리한다. mainwindow 의 private member로 Song class의 객체가 존재하여 melody track과 drum track의 Sequence를 저장하는 역할을 하고, signal and slot으로 play버튼을 누르면 play.cpp/h의 start() SLOT이 호출되어 음악이 재생되게 된다. Song class는 두 개의 Vector Seq로 이루어져 있으며 Seq는 Note들의 vector로 이루어져 있고, Note는 두 개의 integer member와 Pitch member로 이루어져 있다.

또한 Problem1을 진행하기 위한 테스트 코드가 있다. as4에서 io파일에 존재하는 seqio.h/cpp, songio.h/cpp파일을 이용하여 test_io.cpp코드가 컴파일되어 Test의 통과 여부를 확인할 수 있다. Test를 통과시키기 위하여 seqio와 songio에서 각각 <<, >>, ==의 operator들을 overloading시켰다. 또한 test_io.cpp에서 <<와 >>는 using namespace as4::io::operators가 사용된 블록 안에서 쓰이지만 QCOMPARE는 블록 밖에 존재하기 때문에 seqio와 songio에서 as4::io에 해당하는 block안에 정의

와 선언하였다.

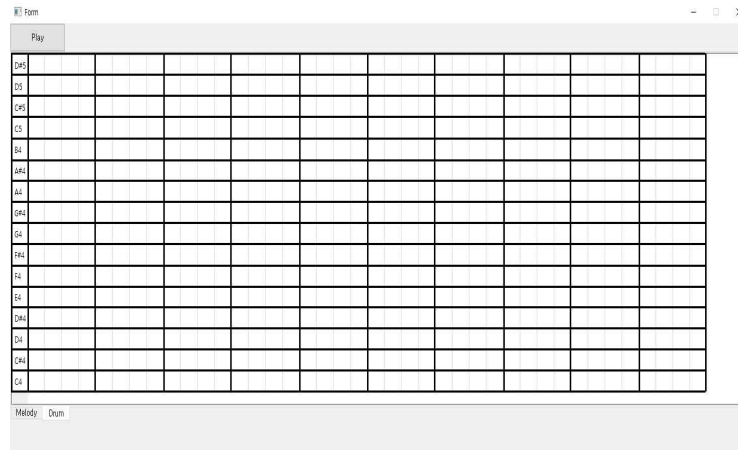
3. Class 설명

mainwindow.h/.cpp에 위치한 MainWindowWidget class는 UI를 관리하는 역할을 한다. public method로는 constructor, setforput() 함수가 있다. setforput()함수는 노트의 삽입을 준비해주는 역할을 담당하는데, cell이 클릭 될 경우 해당 cell의 row와 column을 이용하여 노트에 시작 시각, Duration, Pitch를 삽입한다. Pitch를 삽입할 때에는 as4::model::pitch에 정의된 StandardPitchStrngfer class를 이용한다. MainWindowWidget의 private slots로는 on_pushButton_released(), on_tableWidget_cellClicked(), on_tableWidget_2_cellClicked(), on_tabWidget_tabBarClicked(), on_tableWidget_cellPressed(), on_tableWidget_2_cellPresed()가 존재한다. on_pushButton_released()는 버튼을 클릭하고 떼었을 때 발생하는 slot이고, on_tableWidget_cellClicked()와 on_tableWidget_2_cellClicked()는 각각 tableWidget과 tableWidget2의 cell이 클릭되었을 때 발생하는 SLOT이다. On_tabWidget_tabBarClicked()는 tabBar가 클릭되었을 때 발생하는 SLOT이며 on_tableWidget_cellPressed()와 on_tableWidget_2_cellPressed()는 각각 tableWidget과 tableWidget2가 오른쪽 마우스 클릭 되었을 때 발생하는 SLOT이다. MainWindowWidget의 SIGNAL로는 buttonreleased()가 있다. 이는 on_pushButton_released() 슬롯에서 작업 수행을 모두 마친 뒤 발생하는 signal로 MainWindowWidget의 connect에 의한 연결에 의해 private member인 pd나 pm의 slot인 start()로 이동 시키는 역할을 한다. MainWindowWidget의 private member로는 ui, 노래를 담는 포인터인 Song pointer type의 music, melody mode인지 drum mode인지 판별해주는 mode, melody mode일 때 사용하는 포인터인 playmusic pointer type의 pm, drum mode 일 때 사용하는 포인터인 playmusic pointer type의 pd, cell의 현재 사용중(색칠) 여부를 나타내는 visit배열이 있다.

음악의 재생 역할을 담당하는 play.h/.cpp에는 playmusic, playmelody, playdrum class가 있다. Playmelody와 playdrum은 playmusic의 derived class이다. 먼저, playmusic의 public method로는 constructor, virtual destructor이 존재한다. Public slots로는 start()함수가 존재하는데, 이는 실제 재생을 할 수 있게 해주는 함수이다. Protected member로는 재생할 노트들의 목록을 Sort된 형태로 담는 vector인 result, 재생할 note의 목록을 담고있는 ISeq 포인터 타입의 track, 타이머와 재생 이벤트를 실행시켜 주는 loop, 재생할 노트의 개수를 의미하는 num이 있다. Playmelody와 playdrum class에는 각각 constructor와 destructor가 존재한다.

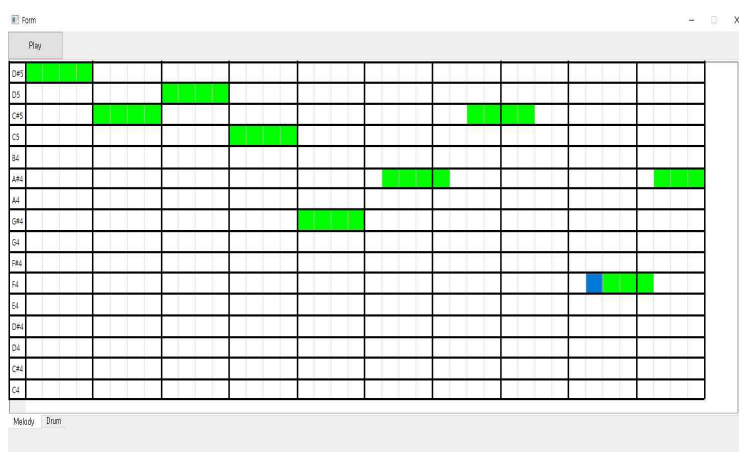
4. 프로그램 실행 방법

프로젝트를 Run하면 아래 그림과 같은 창이 나온다.

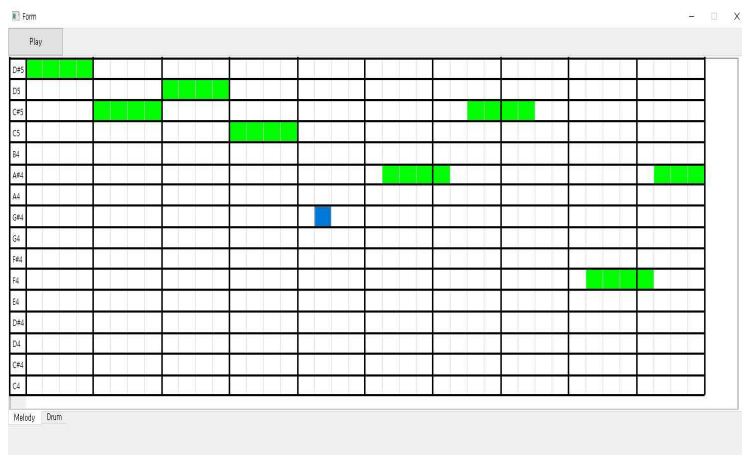


아래 Melody tab이 클릭되어 있으면 Melody 상태에서 음악을 편집할 수 있으며, Drum이 선택되어 있으면 Drum상태에서 음악을 편집할 수 있다. 각 작은 cell들을 클릭하면 1beat짜리 Note가 생성되며, 생성된 Note에 해당하는 자리가 초록색으로 바뀐다. 생성된 Note를 마우스 오른쪽 클릭하면 해당 Note가 삭제되며 해당 Note에 해당하는 자리가 다시 흰색으로 바뀐다. 왼쪽 위의 Play버튼을 누르면 현재 tab에 해당하는 음악이 재생된다. Melody 트랙의 경우 Note의 GetDuration()에 해당하는 유지시간이 끝나면 음표의 오디오 재생을 곧바로 멈추고, Drum Track의 경우 Note의 GetDuration()에 해당하는 유지시간을 무시하고 음표의 오디오 음원을 끝까지 재생한다.

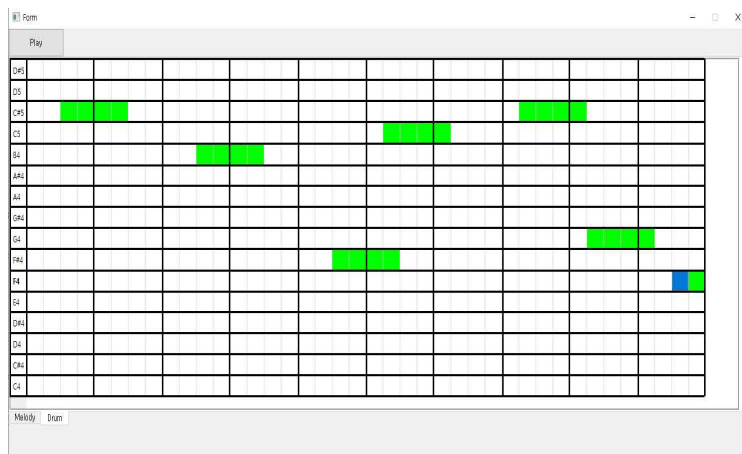
5. 프로그램 실행 예제



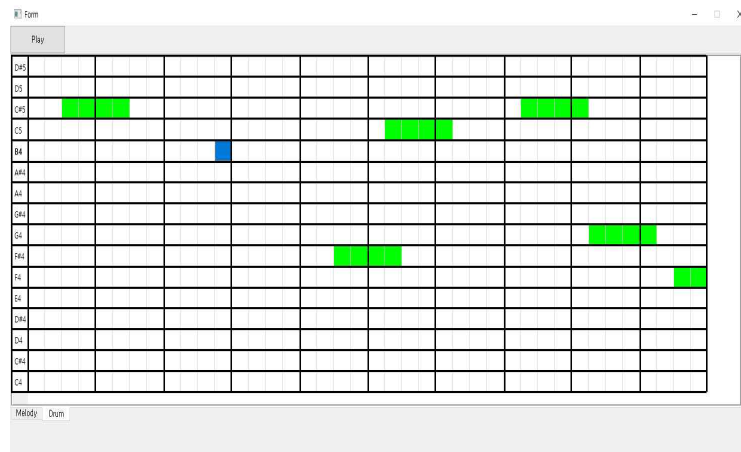
Melody Track 노트 넣은 모습



Melody Track 노트 삭제한 모습



Drum Track 노트 넣은 모습



Drum Track 노트 삭제한 모습

6. 토론, 결론 및 개선방향

이번 프로그램을 구현하며 새롭게 배우게 된 Qt에 대해 더 자세히 이해하게 되는 계기가 되었다. 특히, 처음 접해 본 QTimer, QEventLoop, QSoundEffect, QWidget 등의 class에 어떤 메소드와 멤버들이 있어 어떻게 활용해야 하는지 직접 경험해보며 느낄 수 있었다. 또한, 새롭게 접한 개념인 Signal and Slot 개념을 적용해보며 실제 객체지향 프로그래밍에서 이루어지는 프로그래밍의 개념에 대한 이해를 좀 더 심화시키는 계기가 되었다. 또한 Qt를 사용하여 보며 지금까지는 CMD창에서 문자들만을 이용하여 프로그램을 구현하고 작업하였다면, 이제는 실제 그래픽까지 사용하여 작업을 할 수 있다는 점에서 매우 새로웠다. 이번 프로그램에서는 음표의 생성과 삭제, 재생의 항목만 구현되어 실제 시장에서 사용되는 Music Sequencer보다는 조금 더 단순한 형태였다. Note의 길이를 1beat로 고정하지 않고 조정할 수 있거나, 노트가 추가된 상태에서 노트의 위치를 이동할 수 있거나, Melody와 Drum 등의 한가지 Track에서만 작업하는 것이 아니라 여러 Track에서 작업한 음악을 합쳐서 재생할 수 있다면 더 발전된 프로그램이 될 수 있다고 생각된다.

7. 참고문헌

<https://doc.qt.io/archives/qt-4.8/qeventloop.html>

<https://doc.qt.io/qt-5/qsoundeffect.html>

<https://doc.qt.io/qt-5/qtimer.html>

<https://doc.qt.io/qt-5/qwidget.html>