

## Programming Assignment #5

### Qt 타자 연습

Due Date: 2022년 6월 10일 금요일 23:59

담당조교: 남석현 (obiwan96@postech.ac.kr)

#### 1. 과제 목표

본 과제에서는 C++와 Qt framework를 이용한 GUI Application 제작을 실습해본다. Qt Creator를 사용해 간단한 타자 연습 게임을 만들어보는 것을 목표로 한다.

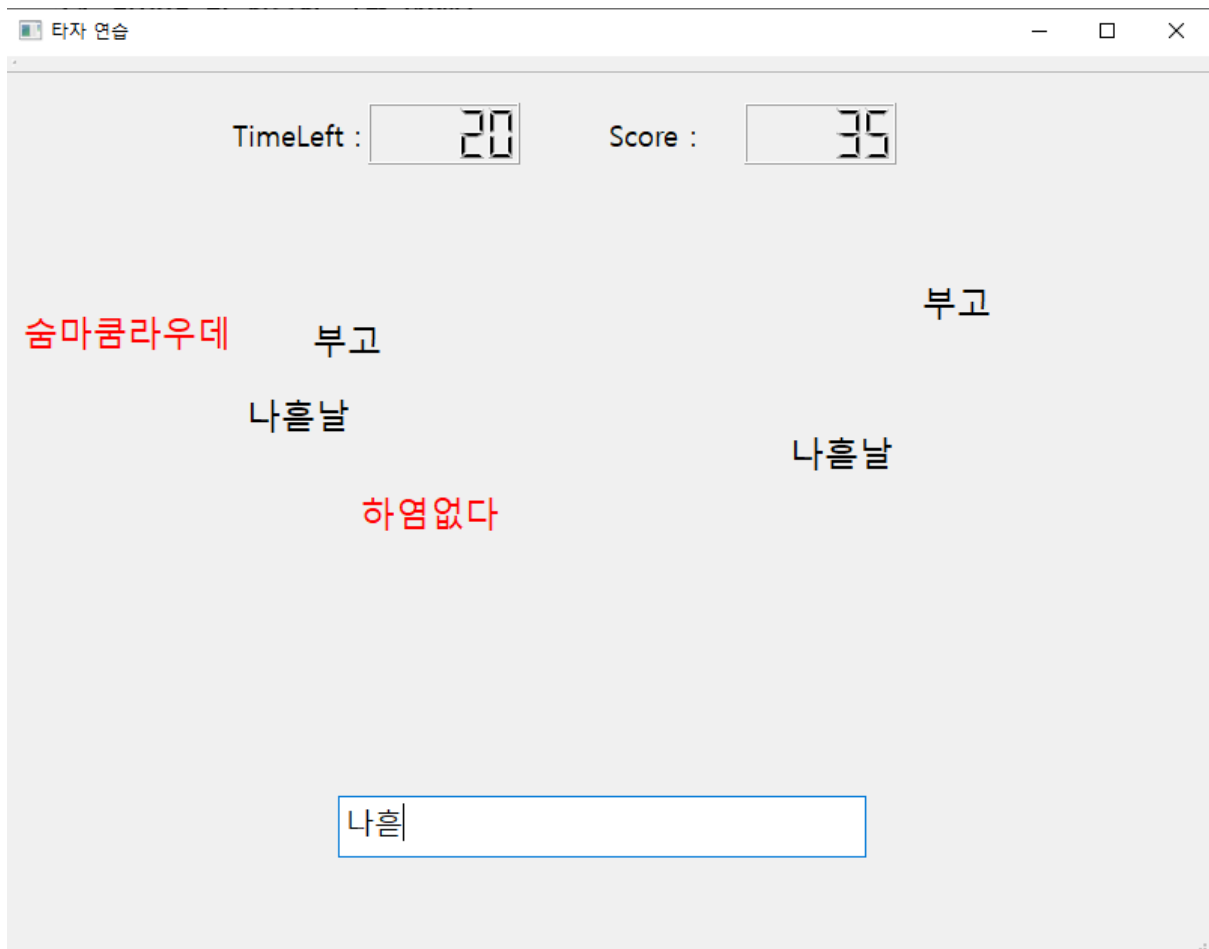


Fig. 1. Qt로 제작한 타자 연습

## 2. 과제 설명

### 2.1. 개요

본 과제에서는 타자 연습을 GUI Application 형태로 구현한다.

### 2.2. 시작 화면

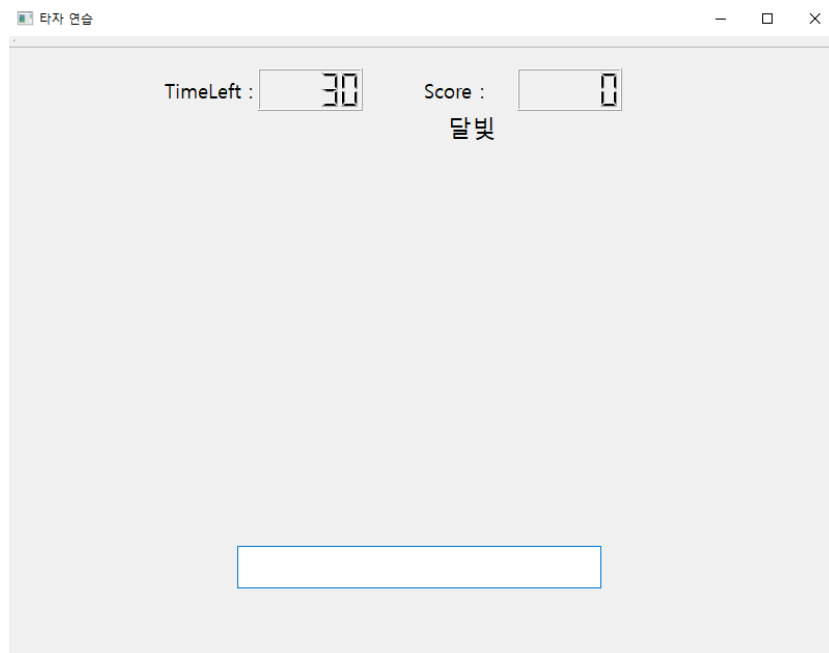


Fig. 2. 시작 화면

Application을 실행했을 때 나타나는 화면이다. 아래쪽의 텍스트 입력 창과 상단의 30초로 설정된 타이머와 0점으로 초기화되어 있는 점수 창이 있으며, 연이어 단어가 생성된다. 타자로 단어를 정확히 치면 해당 단어가 사라지며, 점수를 얻는다.

## 3. 구현 요구사항 (80%)

### 3.1. 타이머 및 점수창 구현 (10%)

- Application이 시작되면 상단에 30초로 설정된 타이머가 나타난다. (5%)
  - 시간은 1초마다 줄어든다.
  - Player가 단어를 놓치면 남은 시간이 3초가 줄어든다.
  - 타이머의 시간은 음수로 표기되지 않는다. (예외처리 할 것)
- Application이 시작되면 상단에 0점으로 설정된 점수창이 나타나며, 플레이어의 진행에

따라 변경된 점수가 나타난다. (5%)

### 3.2. 단어 박스 구현 (30%)

- 게임 시작과 함께 단어 박스가 1초마다 새로 생성된다. (10%)
  - 단어는 다음 기본 단어 집합 중 랜덤으로 선택한다.
  - `QString easy_words[30] = {"명징", "직조", "갈증", "비행기", "나흘날", "비둘기", "웅달샘", "줄이다", "거울", "집오리", "물리학", "토마토", "올리브", "은박지", "문지방", "물병", "양치질", "어린이", "집무실", "달빛", "다르다", "부고", "낫", "서당", "참외", "사다리", "천지", "인바디", "어금니", "추격"};`
  - 총 화면상에 존재하는 단어 박스의 수는 6개를 넘지 않도록 한다.
  - 사용자의 입력으로 인해 단어가 6개보다 적어지면 다시 단어를 생성한다.
  - 초기 Y축 좌표는 타이머 아래의 맨 위이며, X축 좌표는 랜덤으로 선택된다.
  - 생성된 단어 박스는 타이머 및 점수창과 겹치지 않도록 하며, 글자가 화면 밖으로 나가서 가리는 일이 없게 한다.
  - 각 단어 박스가 겹치지 않도록 하거나 같은 단어가 중복돼서 생성되지 않도록 하는 기능은 구현할 필요 없다. (화면 상에 같은 단어가 두 개여도 상관없다.)
  - 단어 '박스'라고 해서 테두리를 입힐 필요는 없다.
- 단어 박스가 생성될 때 20%의 확률로 어려운 단어 박스가 생성된다. (5%)
  - 단어는 다음 어려운 단어 집합 중 랜덤으로 선택한다.
  - `QString crazy_words[10] = {"숨마쿰라우데", "우거지다", "토지소유권", "하염없다", "도통하다", "넙적부리", "타란티노", "방울토마토", "객체지향수업", "컴퓨터공학과"};`
  - 위치는 일반 단어 박스와 같은 기준을 따른다. (화면 밖으로 단어가 나가면 안된다!)
  - 어려운 단어는 빨간색으로 나타난다.
- 생성된 단어 박스가 떨어진다. (10%)
  - 각 단어 박스들은 Y축을 따라 떨어지며, 최소 1초에 한 번씩 위치를 업데이트 해야 한다.
  - 떨어지는 속도는 직접 플레이해보며 적당한 값으로 설정할 것.
  - 어려운 단어 박스는 창 중앙까지는 빠르게(기본 속도의 1.5배) 떨어지며, 중앙 이후로는 기본 단어 박스와 같은 속도로 떨어진다.
- 각 Player가 입력하지 못한 단어 박스들은 입력 창보다 약간 위에서 사라지며, 단어박스가 사라질 때마다 타이머의 시간은 3초 줄어든다. (5%)

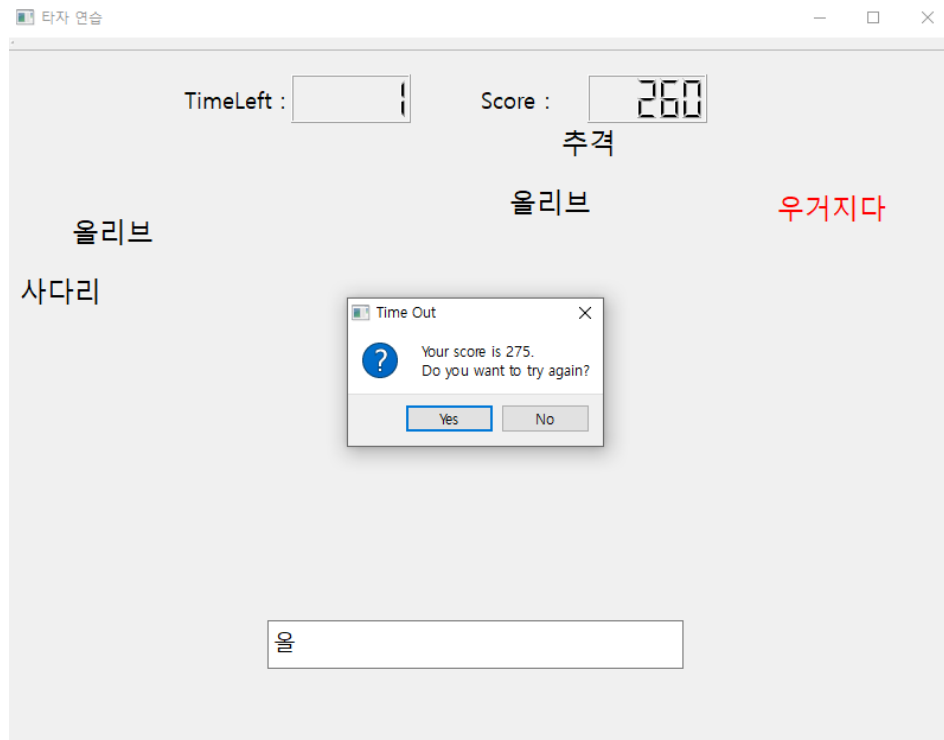
- 타이머에 출력되는 시간은 음수가 되지 않는다. (음수가 되어도 0초로 나타나며 게임이 끝나야 한다.)

### 3.3. 입력 창 구현 (30%)

- Player가 텍스트를 입력하기 위한 입력창을 구현한다. (5%)
- Player가 텍스트를 친 후 엔터를 치면 텍스트가 사라진다. (15%)
  - **힌트** : eventFilter 및 QT::Key\_Return 에 대해 찾아볼 것.
- Player가 엔터를 쳤을 때 현재 입력한 단어를 생성된 단어 박스들과 비교 후 같은 단어가 있으면 해당 단어를 지우고 점수를 업데이트한다. (10%)
  - 맞춘 단어가 일반 단어이면 10점, 어려운 단어이면 15점을 올린다.
  - Player가 친 단어가 현재 없는 단어이면 텍스트만 사라지고 아무 일도 일어나지 않는다.

### 3.4. 결과 창 (10%)

- 타이머의 시간이 0초 이하가 되면 게임이 끝나며, 팝업 알림 메시지를 띄워 점수를 알려준다. (5%)



● Fig. 3. 게임 종료 화면

- 팝업 알림 메시지에서 게임을 한번 더 할 것인지 아니면 끝 것인지 물으며 Yes를 누를

경우 새 게임이 진행되고 No를 누르면 게임이 꺼진다. (5%)

- 새 게임이 시작되면 초기 시작 화면과 같은 상태로 시작해야 한다.

## 4. 참고사항

### 4.1. MinGW 위에서 Qt IDE 환경 (QtCreator) 설정하기

MSYS2 MINGW64 터미널에서, 먼저 설치된 모든 패키지를 최신 버전으로 업데이트한다.

- `pacman -Syu`

MSYS2 MINGW64 터미널에서 다음 패키지를 설치한다.

- `pacman -S mingw-w64-x86_64-toolchain`
  - C++ 컴파일러와 디버깅 툴
- `pacman -S mingw-w64-x86_64-qt5`
  - Qt5 프레임워크
- `pacman -S mingw-w64-x86_64-qt-creator`
  - QtCreator IDE

설치 완료 후 터미널에서 `"qtcreator &"` 명령어를 실행하여 IDE를 실행한다.

### 4.2. 구현 참고사항

Qt framework에서 구현을 하기 위해 사용하는 여러 Class들에 대한 정보는 아래 documentation에서 찾을 수 있다. 적절한 Class를 선택해 구현하고, 각 Class가 포함하는 method들에 대한 정보를 documentation을 참고해 활용하도록 한다.

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

참고로, 조교가 과제 예시 구현에 사용한 Class의 목록은 다음과 같다. 참고일 뿐, 정답은 아님.

<QTimer>, <QMessageBox>, <QPointF>, <QRect>, <QWidget>, <cstdlib>, <ctime>, <QtGui>, <QTextEdit>

### 4.3. 코드 작성 관련 주의사항

- 새로운 .h / .cpp 파일을 만들면 반드시 해당하는 프로젝트의 .pro 파일에 소스 코드와 헤더를 명시해줘야 한다. 이 과정을 지나칠 시 해당 소스 코드가 빌드 되지 않는다.
- 만약 소스 코드를 제대로 작성한 것 같은데도 컴파일/런타임 오류가 발생하면 프로젝트를 Clean 후 Rebuild All 하는 것이 도움이 될 수 있다.

## 5. 과제 제출 관련 유의사항 및 채점 기준

- 제출 파일은 **학번\_assign5.zip**, **학번\_assign5\_report.pdf** 형식의 두 가지 파일로 각각 업로드한다. (PLMS 이용)
- 컴파일이 되는 것을 확인한 프로젝트 폴더를 Clean 후 그대로 압축해서 보고서와 함께 제출할 것.
- 제출 보고서는 보고서 양식에 따라 필요한 내용을 중심으로 간결하게 작성하며, pdf 파일로 제출한다. **게임에서 단어를 이동시킨 방법에 대한 간단한 설명을 포함시키며, 추가로 구현한 Class 및 함수들에 대한 설명을 반드시 포함시킨다.**
- 명예서약(Honor Code): 보고서 표지 및 작성한 코드 main에 주석으로 다음 내용을 포함한다.  
**"나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다."**
- 제출 기한이 지나면 하루마다 얻은 총점의 20% 감점
- 1일 이내 지연 20% 감점, 2일 이내 지연 40% 감점, 5일 이상 지연 0점
- 제출 양식을 지키지 않을 시 과제 전체 점수 10% 감점
- **채점 기준**
  - **프로그램 구현 (80%)**
    - 자세한 기준은 위 개발 요구사항 항목 참조
  - **프로그램의 가독성 (10%)**
    - 프로그램의 소스 코드를 이해하기 쉽도록 주석을 잘 붙였는가?
    - 동적 할당된 메모리가 있다면 사용 후 이를 적절히 해제해주었는가?
  - **보고서 구성 및 내용, 양식 (10%)**
    - 보고서가 적절한 내용으로 이해하기 쉽게 잘 작성되었는가?
    - 본인이 추가로 구현한 함수 및 Class에 대한 설명이 있고 적절한가?

- 채점 환경은 MinGW 상의 QtCreator를 사용할 것임.
- Cheating시 F 학점은 물론 대학교 차원의 징계가 있을 것. 모든 코드는 인터넷 상의 코드와 대조해볼 것임. 참조 및 토의 시 이를 이해한 후 본인의 방식으로 새로 짤 것.
- 참고한 자료들 및 사이트는 보고서에 reference로 작성해야 한다.
- 과제에 대한 문의사항은 PLMS Q&A 게시판을 이용하거나 본 과제 담당 조교에게 이메일을 통해 질문한다.