

hw1 보고서

20기 인턴 송수민

- 코드 분석

1. 헤더파일

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QPushButton>
#include <QTextEdit>
#include <QTimer>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    void handleButtonClicked(); // 숫자/문자 버튼 클릭 처리
    void handleBackspace();
    void handleEnter();
    void handleSpace();
    void handleClearFile();
    void handleShift();
    void handleSymbol();
    void handleChangeLanguage();

private:
    Ui::MainWindow *ui;
    bool shiftOn;
    bool koreanMode;
    bool symbolMode;

    QPushButton* lastButton;
    qint64 lastPressTime;
    int multiTapIndex;
};

#endif // MAINWINDOW_H
```

QMainWindow : Qt에서 기본 제공하는 메인 윈도우 클래스. 메뉴, 툴바, 상태바 등을 쉽게 사용 가능.

QPushButton : 버튼 위젯.

QTextEdit : 여러 줄 입력 가능한 텍스트 편집 위젯.
QTimer : 시간 기반 이벤트 처리용 클래스.
그리고 각 함수들의 선언과 변수들을 정의해 두었다.

2. mainwindow.cpp 파일

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QDateTime>
#include <QTimer>
#include <QFile>
#include <QTextStream>

MainWindow::MainWindow(QWidget *parent)
: QMainWindow(parent)
, ui(new Ui::MainWindow)
, shiftOn(false)
, koreanMode(false)
, symbolMode(false)
, lastButton(nullptr)
, lastPressTime(0)
, multiTapIndex(0)
{
    ui->setupUi(this);

    // 버튼 텍스트 설정
    ui->button1->setText(".,?!");
    ui->button2->setText("ABC");
    ui->button3->setText("DEF");
    ui->button4->setText("GHI");
    ui->button5->setText("JKL");
    ui->button6->setText("MNO");
    ui->button7->setText("PQRS");
    ui->button8->setText("TUV");
    ui->button9->setText("WXYZ");
    ui->button0->setText("0");

    // 버튼 연결
    QList<QPushButton*> buttons = {
        ui->button1, ui->button2, ui->button3, ui->button4,
        ui->button5, ui->button6, ui->button7, ui->button8,
        ui->button9, ui->button0
    };
};
```

```

for (QPushButton* btn : buttons) {
    connect(btn, &QPushButton::clicked, this, &MainWindow::handleButtonClicked);
}

// 특수 버튼
connect(ui->back_space, &QPushButton::clicked, this, &MainWindow::handleBackspace);
connect(ui->enter, &QPushButton::clicked, this, &MainWindow::handleEnter);
connect(ui->space, &QPushButton::clicked, this, &MainWindow::handleSpace);
connect(ui->reset, &QPushButton::clicked, this, &MainWindow::handleClearFile);
connect(ui->shift, &QPushButton::clicked, this, &MainWindow::handleShift);
connect(ui->change_langue, &QPushButton::clicked, this, &MainWindow::handleChangeLanguage);
connect(ui->boutton_symbol, &QPushButton::clicked, this, &MainWindow::handleSymbol);
}

MainWindow::~MainWindow()
{
    delete ui;
}

// ----- 버튼 동작 -----

// 버튼 클릭 (멀티탭 방식)
void MainWindow::handleButtonClicked()
{
    QPushButton *btn = qobject_cast<QPushButton*>(sender());
    if (!btn) return;

    QString chars = btn->text();
    qint64 now = QDateTime::currentMSecsSinceEpoch();

    if (lastButton == btn && now - lastPressTime < 1000) {
        multiTapIndex = (multiTapIndex + 1) % chars.length();

        QString content = ui->textEdit->toPlainText();
        if (!content.isEmpty()) {
            content.chop(1); // 마지막 문자 지우기
        }
    }
}

```

```

        QChar ch = chars.at(multiTapIndex);
        if (shiftOn) {
            ch = ch.toUpper();
        } else {
            ch = ch.toLower();
        }

        content.append(ch);
        ui->textEdit->setPlainText(content);
        ui->textEdit->moveCursor(QTextCursor::End);
    } else {
        // 새 입력
        multiTapIndex = 0;
        QChar ch = chars.at(multiTapIndex);
        if (shiftOn) {
            ch = ch.toUpper();
        } else {
            ch = ch.toLower();
        }

        ui->textEdit->insertPlainText(QString(ch));
    }

    lastButton = btn;
    lastPressTime = now;
}

```

클래스가 생성될 때 각각의 변수들에 필요한 값을 저장하고 버튼 ui와 함수들을 연결해주었다.

버튼 클릭 시 sender()을 통해서 어떤 버튼이 클릭되었는지 확인 후 잘못된 경우 함수를 종료하며, 올바른 경우 버튼에 설정된 문자열을 가져온다. 이후 여러 번 클릭 시 다음 문자로 바꾸기 위해 현재 시간을 밀리초 단위로 받아온 후 같은 버튼의 클릭 간격이 1초 이내일 시 멀티탭으로 판단하고 멀티탭 변수를 증가시켜 다음 문자를 선택하며 % chars.length()로 순환하면서 문자를 선택한다. 멀티탭을 통해서 선택된 문자가 기존 문자의 마지막을 대체하도록 하기위해 삭제한다. shift 상태일 시 대문자로 변환한다. 이후 변환된 문자를 텍스트 박스 끝에 추가하고 커서를 마지막 위치로 이동시킨다. 1초가 지나거나 다른 버튼 클릭 시 새 입력으로 처리한다. 처리 이후 상태를 업데이트 한다.

```

// 백스페이스
void MainWindow::handleBackspace()
{
    QString content = ui->textEdit->toPlainText();
    if (!content.isEmpty()) {
        content.chop(1);
        ui->textEdit->setPlainText(content);
        ui->textEdit->moveCursor(QTextCursor::End);
    }
}

// 엔터
void MainWindow::handleEnter()
{
    QString content = ui->textEdit->toPlainText().trimmed();

    if (!content.isEmpty()) {
        QFile file("log.txt");
        if (file.open(QIODevice::Append | QIODevice::Text)) {
            QTextStream out(&file);
            out << content << "\n";
            file.close();
            ui->statusbar->showMessage("추가되었습니다");
        } else {
            ui->statusbar->showMessage("파일 열기 실패");
        }
    }

    ui->textEdit->clear(); // 화면 비우기
}

// 스페이스
void MainWindow::handleSpace()
{
    ui->textEdit->insertPlainText(" ");
}

```

백스페이스 클릭 시 현재 텍스트 박스의 문자열을 받아와서 문자열 존재 시 마지막 문자를 삭제하고 수정된 문자열을 텍스트 박스에 다시 설정하고 커서를 마지막 위치로 옮겨 한 글자씩 지우는 기능을 구현하였다.

엔터 클릭 시 텍스트 박스의 문자열을 받아와서 공백을 제거하고 비어있지 않을 시 log 파일에 추가 모드로 내용을 저장한다. 결과를 상태바에 출력하고 입력 후 텍스트 박스를 초기화 한다.

스페이스 클릭 시 공백을 삽입한다.

```

// 파일 초기화
void MainWindow::handleClearFile()
{
    QFile file("log.txt");
    if (file.open(QIODevice::WriteOnly | QIODevice::Truncate)) {
        file.close();
        ui->statusbar->showMessage("파일이 초기화되었습니다");
    } else {
        ui->statusbar->showMessage("파일 초기화 실패");
    }
}

// Shift
void MainWindow::handleShift()
{
    shiftOn = !shiftOn;

    if (shiftOn) {
        ui->shift->setStyleSheet("background-color: lightblue;");
    } else {
        ui->shift->setStyleSheet("");
    }
}

```

리셋 실행 시 파일일 쓰기 전용 모드 + 기존 파일을 모두 삭제하고 새로 작성하는 truncate 옵션으로 열어서 파일을 초기화하고 결과를 출력한다.

shift키 클릭 시 변수를 통해 쉬프트 상태를 저장하고 버튼 색상도 바꾼다. 버튼은 토글로 작동한다.


```

// 한/영 전환 (기본은 영어)
void MainWindow::handleChangeLanguage()
{
    koreanMode = !koreanMode;
    if (koreanMode) {
        ui->button1->setText(" | ");
        ui->button2->setText(" . ");
        ui->button3->setText(" - ");
        ui->button4->setText("ㄱ ㅋ ㆁ");
        ui->button5->setText("ㄴ ㄷ");
        ui->button6->setText("ㄷ ㅌ ㅅ");
        ui->button7->setText("ㅂ ㅃ ㅍ");
        ui->button8->setText("ㅇ ㆁ ㅈ");
        ui->button9->setText("ㅊ ㅌ ㅍ");
        ui->button0->setText("ㅇ ㅁ ㅎ");
        ui->statusbar->showMessage("Korean Mode");
    } else {
        ui->button1->setText(", . ? !");
        ui->button2->setText("ABC");
        ui->button3->setText("DEF");
        ui->button4->setText("GHI");
        ui->button5->setText("JKL");
        ui->button6->setText("MNO");
        ui->button7->setText("PQRS");
        ui->button8->setText("TUV");
        ui->button9->setText("WXYZ");
        ui->button0->setText("0");
        ui->statusbar->showMessage("English Mode");
    }
}

void MainWindow::handleSymbol()
{
    symbolMode = !symbolMode;
    if (symbolMode) {
        ui->button1->setText("1");
        ui->button2->setText("2");
        ui->button3->setText("3");
        ui->button4->setText("4");
        ui->button5->setText("5");
        ui->button6->setText("6");
        ui->button7->setText("7");
        ui->button8->setText("8");
        ui->button9->setText("9");
        ui->button0->setText("0");
        ui->statusbar->showMessage("Number Mode");
    } else {
        ui->button1->setText("!@#%&");
        ui->button2->setText("^&+()");
        ui->button3->setText("_+~");
        ui->button4->setText("|###/");
        ui->button5->setText(";#'"<");
        ui->button6->setText(", . ? !");
        ui->button7->setText("...° · ■");
        ui->button8->setText("×, €$");
        ui->button9->setText("¥#&@");
        ui->button0->setText("@™☑");
        ui->statusbar->showMessage("Symbol Mode");
    }
}

```

스위치에 모드에 알맞은 문자를 저장한다.

3. main.cpp파일

```

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.setWindowTitle("천지인 키보드");
    w.show();
    return a.exec();
}

```

창 이름 바꾸고 실행한다.

