

```
1  cmake_minimum_required(VERSION 3.5)
2
3  project(hw1 VERSION 0.1 LANGUAGES CXX)
4
5  set(CMAKE_INCLUDE_CURRENT_DIR ON)
6
7  set(CMAKE_AUTOUIC ON)
8  set(CMAKE_AUTOMOC ON)
9  set(CMAKE_AUTORCC ON)
10
11 set(CMAKE_CXX_STANDARD 11)
12 set(CMAKE_CXX_STANDARD_REQUIRED ON)
13
14 find_package(QT NAMES Qt6 Qt5 COMPONENTS Widgets REQUIRED)
15 find_package(Qt${QT_VERSION_MAJOR} COMPONENTS Widgets REQUIRED)
16
17 set(PROJECT_SOURCES
18     src/main.cpp
19     src/mainwindow.cpp
20     include/mainwindow.h
21     ui/mainwindow.ui
22     resources/resources.qrc
23 )
```

- 파일 생성시 나오는 기본 코드
- 파일 경로 PDF처럼 수정

```
if(${QT_VERSION_MAJOR} GREATER_EQUAL 6)
    qt_add_executable(hw1
        MANUAL_FINALIZATION
        ${PROJECT_SOURCES}
    )
set_property(TARGET hw1 PROPERTY AUTOUIC_SEARCH_PATHS "${CMAKE_CURRENT_SOURCE_DIR}/ui")
# Define target properties for Android with Qt 6 as:
# set_property(TARGET hw1 APPEND PROPERTY QT_ANDROID_PACKAGE_SOURCE_DIR
#             ${CMAKE_CURRENT_SOURCE_DIR}/android)
# For more information, see https://doc.qt.io/qt-6/qt-add-executable.html#target-creation
else()
    if(ANDROID)
        add_library(hw1 SHARED
            ${PROJECT_SOURCES}
        )
    # Define properties for Android with Qt 5 after find_package() calls as:
    # set(ANDROID_PACKAGE_SOURCE_DIR "${CMAKE_CURRENT_SOURCE_DIR}/android")
    else()
        add_executable(hw1
            ${PROJECT_SOURCES}
        )
    endif()
endif()
```

- 파일 생성시 나오는 기본 코드
- Ui 경로
- 파일 생성시 나오는 기본 코드

```
target_include_directories(hw1 PRIVATE "${CMAKE_CURRENT_SOURCE_DIR}/include")

target_link_libraries(hw1 PRIVATE Qt${QT_VERSION_MAJOR}::Widgets)

set_target_properties(hw1 PROPERTIES
    MACOSX_BUNDLE_GUI_IDENTIFIER my.example.com
    MACOSX_BUNDLE_BUNDLE_VERSION ${PROJECT_VERSION}
    MACOSX_BUNDLE_SHORT_VERSION_STRING ${PROJECT_VERSION_MAJOR}.${PROJECT_VERSION_MINOR}
    MACOSX_BUNDLE TRUE
    WIN32_EXECUTABLE TRUE
)

if(QT_VERSION_MAJOR EQUAL 6)
    qt_finalize_executable(hw1)
endif()
```

- 파일 생성시 나오는 기본 코드
- 위젯 관련 코드
- 기본코드

```

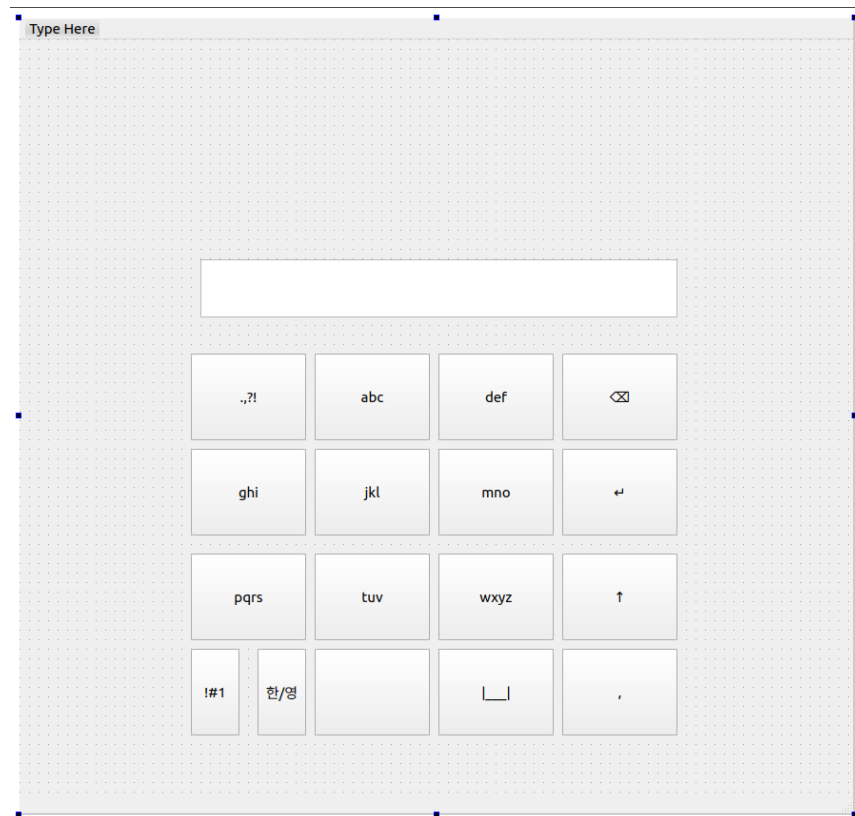
1  #ifndef MAINWINDOW_H
2  #define MAINWINDOW_H
3
4  #include <QMainWindow>
5  #include <QListWidgetItem>
6  #include <vector>
7  #include <QTimer>
8  #include <QFile>
9  #include <QTextStream>
10 #include <QMessageBox>
11
12 QT_BEGIN_NAMESPACE
13 namespace Ui { class MainWindow; }
14 QT_END_NAMESPACE
15
16
17
18 class MainWindow : public QMainWindow
19 {
20     Q_OBJECT
21
22 public:
23     MainWindow(QWidget *parent = nullptr);
24     ~MainWindow();
25
26 private slots:
27     void on_pushButton_clicked();
28     void on_pushButton_2_clicked();
29     void on_pushButton_3_clicked();
30     void on_pushButton_4_clicked();
31     void on_pushButton_5_clicked();
32     void on_pushButton_8_clicked();
33     void on_pushButton_6_clicked();
34     void on_pushButton_7_clicked();
35     void on_pushButton_9_clicked();
36     void on_pushButton_12_clicked();
37     void on_pushButton_10_clicked();
38     void on_pushButton_11_clicked();
39     void on_pushButton_14_clicked();
40     void on_pushButton_17_clicked();
41     void on_pushButton_15_clicked();
42     void on_pushButton_13_clicked();
43     void on_pushButton_16_clicked();

```

헤더파일

- List위젯
- 벡터
- 타이머
- 파일
- 문장 쉽게 저장하는 것
- 메시지 박스 생성

- 버튼 1~16



```
46 private:
47     Ui::MainWindow *ui;
48     QTimer *timer;
49     int timeout=1;
50     int lg=0;//0=영어 1= 한국어 -1= 다른 입력상태
51     int sim=-1;//0=숫자 1..=문자 -1= 다른 입력상태
52     int shift=0;//0=소문자 1=대문자 1번 2=대문자 연속
53
54     std::vector<QString> data;
55
56
57     void updateListWidget();
58     void update();
59
60
61 };
62
63
64 #endif // MAINWINDOW_H
65
```

- 화면 ui
- 타이머
- 언어 상태
- 기호 패널
- 영어에서 shift 사용했는가
- 화면에 띄울 글자 저장하는 벡터
- 화면 위젯(text)
- 타이머용 함수

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    timer=new QTimer;
    timer->setInterval(1500);
    timer->setSingleShot(true);
    connect(timer, &QTimer::timeout, this, &MainWindow::update);
}

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

void MainWindow::update(){timeout=1;}
```

- Ui용 기본 명령어
- 타이머 생성
- 타임아웃 1.5초로 설정
- 한번만 신호 보내기
- 타임 아웃하면 update 함수 실행
- 타임 아웃하면 timeout 변수 1로 바꿈

```

{
    ui->listWidget->clear();
    QString textToShow;
    for(const QString &s : data) {
        textToShow.append(s);
    }
    ui->listWidget->addItem(textToShow);

    if(shift==1){
        ui->pushButton_11->setStyleSheet("background-color: ■rgb(153, 193, 241);");
    }
    else if(shift==2){
        ui->pushButton_11->setStyleSheet("background-color: ■rgb(100, 100, 250);");
    }
    else{
        ui->pushButton_11->setStyleSheet("");
    }

    if(lg==0){
        if(shift==0){
            ui->pushButton->setText(". , ? !");
            ui->pushButton_2->setText("abc");
            ui->pushButton_3->setText("def");
            ui->pushButton_5->setText("ghi");
            ui->pushButton_8->setText("jkl");
            ui->pushButton_6->setText("mno");
            ui->pushButton_9->setText("pqrs");
            ui->pushButton_12->setText("tuv");
            ui->pushButton_10->setText("wxyz");
            ui->pushButton_11->setText("↑");
            ui->pushButton_16->setText(",");
            ui->pushButton_15->setText(" ");
        }
        else {
            ui->pushButton->setText(". , ? !");
            ui->pushButton_2->setText("ABC");
            ui->pushButton_3->setText("DEF");
            ui->pushButton_5->setText("GHI");
            ui->pushButton_8->setText("JKL");
            ui->pushButton_6->setText("MNO");
            ui->pushButton_9->setText("PQRS");
            ui->pushButton_12->setText("TUV");
            ui->pushButton_10->setText("WXYZ");
            ui->pushButton_11->setText("↑");
            ui->pushButton_16->setText(",");
            ui->pushButton_15->setText(" ");
        }
    }
}

```

화면 업뎃

- 리스트 위젯 초기화
- QString 형식의 변수 생성
- 위젯에 vector에 저장된 글자 QString으로 옮긴 후 모두 추가

Hw1

- Shift 상태 1이면 shift키 연한 파랑

- 2면 진함 파랑

- 아니면 기본

- Shift 아닐 때 버튼들 내부 내용

- Shift일 때 버튼들 내부 내용

- 언어 설정1(한국어)일때 상태
 - 숫자 패드 설정(sim 0)일때,
 - 문자 설정일 때(sim>0) 상태들
- (위 내용과 동일하여 생략)

```

else if(lg==1){
    ui->pushButton->setText("|");
    ui->pushButton_2->setText("·");
    ui->pushButton_3->setText("—");
    ui->pushButton_5->setText("ㄱ");
    ui->pushButton_8->setText("ㄴ");
    ui->pushButton_6->setText("ㄷ");
    ui->pushButton_9->setText("ㅁ");
    ui->pushButton_12->setText("ㅂ");
    ui->pushButton_10->setText("ㅅ");
    ui->pushButton_11->setText("·,?!");
    ui->pushButton_16->setText("한자");
    ui->pushButton_15->setText("ㅇ");
}
else if(lg == -1) {
    if (sim == 0) {
        ui->pushButton->setText("1");
        ui->pushButton_2->setText("2");
        ui->pushButton_3->setText("3");
        ui->pushButton_5->setText("4");
        ui->pushButton_8->setText("5");
        ui->pushButton_6->setText("6");
        ui->pushButton_9->setText("7");
        ui->pushButton_12->setText("8");
        ui->pushButton_10->setText("9");
        ui->pushButton_15->setText("0");
        ui->pushButton_11->setText("·,-/");
        ui->pushButton_16->setText("%");
    } else if (sim == 1) {
        ui->pushButton->setText("!");
        ui->pushButton_2->setText("?");
        ui->pushButton_3->setText(".");
        ui->pushButton_5->setText(",");
        ui->pushButton_8->setText("(");
        ui->pushButton_6->setText(")");
        ui->pushButton_9->setText("@");
        ui->pushButton_12->setText(":");
        ui->pushButton_10->setText(";");
        ui->pushButton_15->setText("1/3");
        ui->pushButton_11->setText("·,?!");
        ui->pushButton_16->setText("%");
    } else if (sim == 2) {
        ui->pushButton->setText("/");
        ui->pushButton_2->setText("-");
        ui->pushButton_3->setText("*");
        ui->pushButton_5->setText("_");
        ui->pushButton_8->setText("%");
        ui->pushButton_6->setText("~");
        ui->pushButton_9->setText("^");
        ui->pushButton_12->setText("#");
        ui->pushButton_10->setText("+");
        ui->pushButton_15->setText("2/3");
        ui->pushButton_11->setText("·,?!");
        ui->pushButton_16->setText("%");
    } else if (sim == 3) {
        ui->pushButton->setText("=");
        ui->pushButton_2->setText("\");
        ui->pushButton_3->setText("");
        ui->pushButton_5->setText("<");
        ui->pushButton_8->setText(">");
        ui->pushButton_6->setText("{");
        ui->pushButton_9->setText("}");
        ui->pushButton_12->setText("[");
        ui->pushButton_10->setText("]");
        ui->pushButton_15->setText("3/3");
        ui->pushButton_11->setText("·,?!");
        ui->pushButton_16->setText("%");
    }
}

```



```
void MainWindow::on_pushButton_4_clicked()
{
    if (!data.empty()) {
        data.pop_back();
    }
    updateListWidget();
    timeout=1;
}
```

```
void MainWindow::on_pushButton_14_clicked()
{
    if (lg != -1) {lg = -1;sim = 0;}
    else {
        if (sim == 0) {sim = 1;}
        else {sim = 0;}
    }
    timeout = 1;
    shift=0;
    updateListWidget();
}
```

```
void MainWindow::on_pushButton_17_clicked()
{
    lg++;
    if(lg>1){lg=0;}
    sim=-1;
    timeout = 1;
    shift=0;
    updateListWidget();
}
```

Backspace

- 벡터에서 뒤에 값 지우기

- 위젯 업데이트

문자 버튼

- 문자 상태 아니면 숫자패드 꺼내기

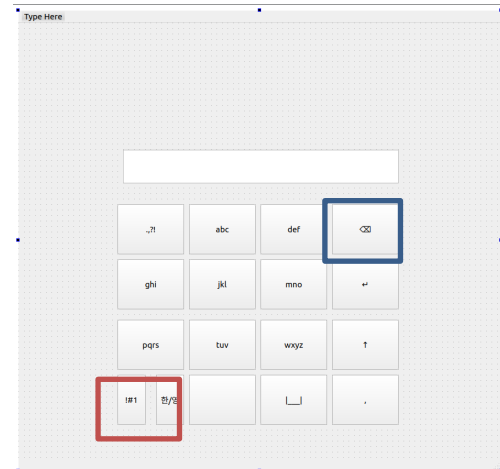
- 숫자패드면 문자패드로 넘김

- Shift 0

한영 버튼

- 언어 변수 1 추가
- 언어 변수가 2이상이면 0으로 돌려놓음
- 문자 변수 -1

- Shift 0



Enter

```
void MainWindow::on_pushButton_7_clicked()
{
    QString textToSave;
    for(const QString &s : data) {
        textToSave.append(s);
    }

    QFile file("/home/david/intern_ws/qt/day3/hw1/hw1.txt");
    if(!file.open(QFile::Append | QFile::Text)){
        QMessageBox::warning(this,"title","file is not open");
        return;
    }
    QTextStream out(&file);
    out << textToSave << "\n";
    file.close();

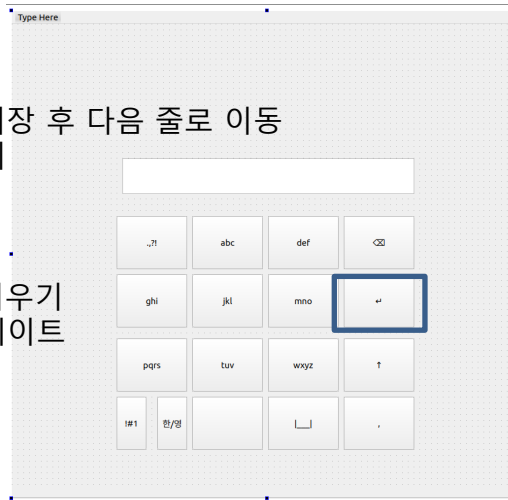
    data.clear();
    updateListWidget();
    timeout=1;
}
```

- QString 형식 변수 생성
- 해당 변수에 vector 요소 삽입

- 절대 경로에 위치한 파일
- 파일 못 찾았다고 띄우기

- 파일에 저장 후 다음 줄로 이동
- 파일 닫기

- Vector 비우기
- 위젯 업데이트



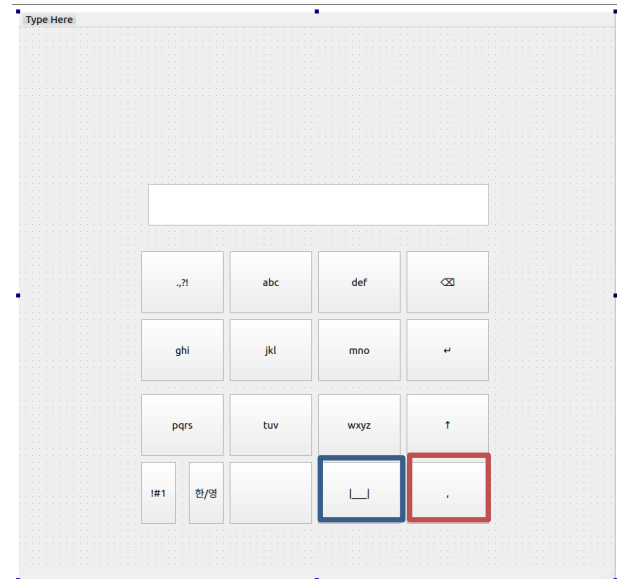
띄어쓰기

- 백터에 공백 넣기

- 입력 시간 초기화

- 영어일 때 ','

- 숫자, 기호일 때 &



```
void MainWindow::on_pushButton_13_clicked()
{
    data.push_back(" ");
    updateListWidget();
    timeout=1;
}
```

```
void MainWindow::on_pushButton_16_clicked()
{
    if(lg==0){data.push_back(",");timeout=1;}
    else if(lg==1){timeout=1;}
    else if(lg == -1){data.push_back("%");timeout=1;}
    updateListWidget();
}
```

```

void MainWindow::on_pushButton_8_clicked()
{
    if(lg==0){
        if(data.empty()){ if(shift==0){data.push_back("j");} else{data.push_back("J");} }
        else{
            if(shift==0){
                if(timeout==0){
                    if(data.back()=="j"){data.back()="k";}
                    else if(data.back()=="k"){data.back()="l";}
                    else if(data.back()=="l"){data.back()="j";}
                    else{data.push_back("j");}
                } else{data.push_back("j");}
            } else{
                if(timeout==0){
                    if(data.back()=="J"){data.back()="K";}
                    else if(data.back()=="K"){data.back()="L";}
                    else if(data.back()=="L"){data.back()="J";}
                    else{data.push_back("J");}
                } else{data.push_back("J");}
            }
        }
        if(shift==1){shift=0;}
        timeout=0; timer->start();
    }
    else if(lg==1){
        if(data.empty()){ data.push_back("L"); }
        else{
            if(timeout==0){
                if(data.back()=="L"){data.back()="R";}
                else if(data.back()=="R"){data.back()="L";}
                else{data.push_back("L");}
            } else{data.push_back("L");}
        }
        timeout=0; timer->start();
    }
    else if(lg == -1){
        if(sim == 0) { data.push_back("5"); }
        else if(sim == 1) { data.push_back("("); }
        else if(sim == 2) { data.push_back("%"); }
        else if(sim == 3) { data.push_back(">"); }
        timeout=1;
    }
    updateListWidget();
}

```

5 L R 쪽 버튼

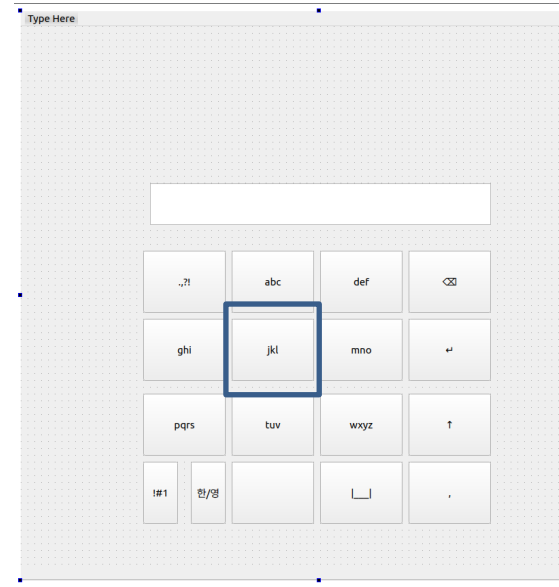
- 비어있다면 언어와 쉬프트 상태를 보고 추가
- Timeout 상태 아니고 동일 버튼 내에 있는 단어면 변경

- Timer 작동

- 영어와 동일

- 숫자 키패드면 5
- 이외에 맞게 설정
(바뀌는 단어가 없어 그냥 push함)

- 화면 업데이트



```

void MainWindow::on_pushButton_11_clicked(){
    if(lg==0){
        shift++;
        if(shift>2){shift=0;}
    }
    else if(lg==1 || (lg == -1 && sim > 0)){
        if(data.empty()){data.push_back(".");}
        else{
            if(timeout==0){
                if(data.back()==""){data.back()=",";}
                else if(data.back()==""){data.back()="?";}
                else if(data.back()=="?"){data.back()="!";}
                else if(data.back()=="!"){data.back()=".";}
                else{data.push_back(".");}
            } else{data.push_back(".");}
        }
        timeout=0;
        timer->start();
    }
    else if(lg == -1 && sim == 0){
        if(data.empty()){ data.push_back("."); }
        else{
            if(timeout==0){
                if(data.back()==""){data.back()=",";}
                else if(data.back()==""){data.back()="-";}
                else if(data.back()=="-"){data.back()="/";}
                else if(data.back()==""){data.back()=".";}
                else{data.push_back(".");}
            } else{data.push_back(".");}
        }
        timeout=0;
        timer->start();
    }
    updateListWidget();
}

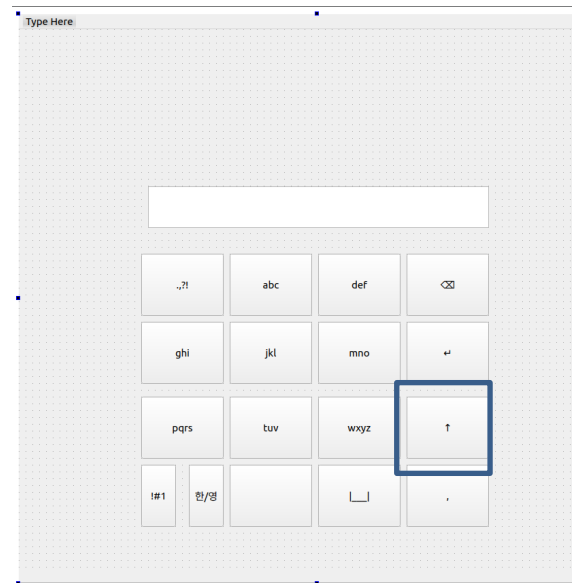
```

Shift 버튼

- 영어 일때만 shift 변경
- 한국어 일때 일반 버튼처럼 작동(똑같으니 생략)

- 숫자일 때 해당하는 버튼으로 바뀜

- 문자 키패드는 추후에 넣음
- 화면 업데이트



```

void MainWindow::on_pushButton_15_clicked()
{
    if(lg==0){data.push_back(" ");timeout=1;}
    else if(lg==1){
        if(data.empty()){ data.push_back("○"); }
        else{
            if(timeout==0){
                if(data.back()=="○"){data.back()="□";}
                else if(data.back()=="□"){data.back()="○";}
                else{data.push_back("○");}
            } else{data.push_back("○");}
        }
        timeout=0;
        timer->start();
    }
    else if(lg == -1){
        if (sim == 0) {data.push_back("0");}
        else if (sim == 1) {sim = 2;}
        else if (sim == 2) {sim = 3;}
        else if (sim == 3) {sim = 1;}
        timeout=1;
    }
    updateListWidget();
}

```

0번 버튼

- 영어 일 때 아무기능 없음(수정함)
- 한국어 일 때 일반 버튼처럼 작동

- 숫자일 때 해당하는 버튼으로 바뀜
- 문자 패드일 때 누르면 sim 번호 바뀜

- 화면 업데이트

