

QTCreator 보고서

20기 인턴 강은구

목차

- 1 HW1
- 2 HW2

HW1

1. 코드 동작 흐름

- ① 실행 시 기본 UI 표시(한글 자판)
- ② 버튼 클릭-> labelOutput에 입력 표시
- ③ 같은 버튼 반복 클릭-> 문자 변경
- ④ Backspace버튼->마지막 문자 삭제
- ⑤ Space버튼->다음 문자 입력 또는 공백 반복 입력
- ⑥ Enter버튼->줄 바꿈
- ⑦ Save버튼->현재 입력된 문자열 txt파일로 저장

2. 클래스 설명

```

1  #ifndef MAINWINDOW_H
2  #define MAINWINDOW_H
3
4  #include <QMainWindow>
5  #include <QWidget>
6  #include <QFile>
7  #include <QTextStream>
8
9  QT_BEGIN_NAMESPACE
10 namespace Ui { class MainWindow; }
11 QT_END_NAMESPACE
12
13 class TextSave {
14 public:
15     static void saveFile(const QString &text);
16 };
17
18 class MainWindow : public QMainWindow
19 {
20     Q_OBJECT
21
22 public:
23     MainWindow(QWidget *parent = nullptr);
24     ~MainWindow();
25     int number=0;
26     QString txt_re, txt, txt_s;
27
28 private slots:
29     void on_btnSwitch_clicked(); // 버튼 눌렀을 때 호출
30     void on_e11_clicked(); // 버튼 눌렀을 때 호출
31     void on_back_clicked();
32     void on_space_bar_clicked();
33     void on_Enter_clicked();
34     void on_Save_clicked();
35
36 private:
37     QTimer *cursorTimer;
38     bool cursorVisible = false;
39     Ui::MainWindow *ui;
40 };
41
42 #endif // MAINWINDOW_H
43

```

Zoom: 110%

① MainWindow 클래스

- Qt의 메인 윈도우 클래스(QMainWindow)
- 전체 UI 제어 및 버튼 이벤트 처리

② Textsave 클래스

- 정적 함수 saveFile(QString) 저장
- 문자열을 txt파일에 저장

3. 함수별 설명

```

15 MainWindow::MainWindow(QWidget *parent)
16 : QMainWindow(parent)
17 , ui(new Ui::MainWindow)
18 {
19     ui->setupUi(this);
20     ui->labelOutput->setStyleSheet("color: black; background-color: white;");
21
22     // 타이머 생성
23     cursorTimer = new QTimer(this);
24     cursorTimer->setInterval(500); // 0.5초마다 실행
25
26     connect(cursorTimer, &QTimer::timeout, this, [=]() {
27         QString text = ui->labelOutput->text();
28         if (cursorVisible) {
29             if (text.endsWith("_")) {
30                 text.chop(1);
31                 ui->labelOutput->setText(text);
32             }
33             cursorVisible = false;
34         } else {
35             ui->labelOutput->setText(text + "_");
36             cursorVisible = true;
37         }
38     });
39     cursorTimer->start();
40
41     // 시작: 한글 위젯 보이게
42     ui->widgetKorean->move(400, 300);
43     ui->widgetEnglish->move(400, 300);
44     ui->widgetEnglish_2->move(400, 300);
45     ui->widgetKorean->show();
46     ui->widgetEnglish->hide();
47     ui->widgetEnglish_2->hide();
48
49     QPushButton *buttons[] = {
50         ui->e1, ui->e2, ui->e3, ui->e4, ui->e5, ui->e6,
51         ui->e7, ui->e8, ui->e9, ui->e11, ui->e12, ui->e13,
52         ui->E2, ui->E3, ui->E4, ui->E5, ui->E6,
53         ui->E7, ui->E8, ui->E9
54     };
55
56     for (QPushButton *btn : buttons) {
57         connect(btn, &QPushButton::clicked, this, [=]() {
58             if (btn == ui->back || btn == ui->Enter || btn == ui->space_bar || btn == ui->e11) {
59                 return;
60             }
61             QString current = ui->labelOutput->text();
62             txt = btn->text();
63             if(current.length()-1>=0 && current[current.length()-1]=='_'){
64                 current.chop(1);
65                 ui->labelOutput->setText(current);
66             }
67             if(txt.length()==1){
68                 ui->labelOutput->setText(current + txt[0]);
69                 return;
70             }

```

```

71
72
73     if(txt_re[0]==txt[0]&&!txt.isEmpty()&&txt_re[0]!="\n"){
74         number++;
75         if(txt[number]==0)number=0;
76         QChar firstChar = txt[number];
77         current.chop(1);
78         ui->labelOutput->setText(current + QString(firstChar));
79     }
80     else if (txt_re[0]!=txt[0]&&!txt.isEmpty()&&txt_re[0]!="\n") {
81         number=0;
82         QChar firstChar = txt[number];
83         ui->labelOutput->setText(current + QString(firstChar));
84     }
85     txt_re = btn->text();
86 });
87
88 connect(btn, &QPushButton::pressed, this, [=]() {
89     btn->setStyleSheet("background-color: lightblue;");
90 });
91 connect(btn, &QPushButton::released, this, [=]() {
92     btn->setStyleSheet("");
93 });
94 }
95
96
97 MainWindow::~MainWindow()
98 {
99     delete ui;

```

① MainWindow::MainWindow()

- UI 초기화(ui->setupUI(this))
- 커서 깜빡임(QTimer)
- 한/영 위젯 및 초기 표시 설정
- 버튼 이벤트 연결(Lambda 사용)

```

101 void MainWindow::_on_e11_clicked()
102 {
103     if (ui->widgetEnglish_2->isVisible()) {
104         ui->widgetEnglish->show();
105         ui->widgetEnglish_2->hide();
106     }
107     else {
108         ui->widgetEnglish_2->show();
109         ui->widgetEnglish->hide();
110     }
111 }

```

② MainWindow::_on_e11_clicked()

- 영문 대-소문자 전환
- Widget show/hide를 통해 표시/숨김 구현

```

171 void MainWindow::on_btnSwitch_clicked()
172 {
173     if (ui->widgetKorean->isVisible()) {
174         ui->widgetEnglish->show();
175         ui->widgetKorean->hide();
176         ui->e1->move(420,310);
177         ui->e11->show();
178         ui->e12->show();
179         ui->e13->show();
180     }
181     else {
182         ui->widgetKorean->show();
183         ui->widgetEnglish->hide();
184         ui->widgetEnglish_2->hide();
185         ui->e1->move(690,410);
186         ui->e11->hide();
187         ui->e12->hide();
188         ui->e13->hide();
189     }
190 }

```

③ MainWindow::on_btnSwitch_clicked()

- 한글-영어 자판 전환
- Widget show/hide를 통해 표시/숨김 구현

```

113 void MainWindow::on_back_clicked()
114 {
115     QString current = ui->labelOutput->text();
116     if(current.length()!=0&&current[current.length()-1]=='_'){
117         current.chop(1);
118     }
119     if (!current.isEmpty()) {
120         current.chop(1);
121         ui->labelOutput->setText(current);
122     }
123 }

```

④ MainWindow::on_back_clicked()

- 입력된 문자열 마지막 문자 삭제
- 커서(_)가 입력되어있을 시 삭제하고 시작

```

125 void MainWindow::on_space_bar_clicked()
126 {
127     QString current = ui->labelOutput->text();
128     if(current.length()!=0&&current[current.length()-1]=='_'){
129         current.chop(1);
130     }
131     if(txt_re[0]==' '){
132         ui->labelOutput->setText(current + ' ');
133         number=0;
134         txt_re=' ';
135     }
136     else {
137         number=0;
138         txt_re=" ";
139     }
140 }
141 }

```

⑤ MainWindow::on_space_bar_clicked()

- 다음 입력 문자로 넘어가기
- 공백 반복 입력
- Txt_re과 nubmer을 초기화 하여 다음 문자 입력시 반영

```
153 void MainWindow::on_Save_clicked()
154 {
155
156
157     QString current = ui->labelOutput->text();
158     if(current.isEmpty())return;
159
160     if(current.length()!=0&&current[current.length()-1]=='_'){
161         current.chop(1);
162     }
163
164     TextSave::saveFile(current);
165     ui->labelOutput->setText("");
166 }
```

⑥ MainWindow::on_Save_clicked()

- 파일 저장 함수 호출
- 마지막 문자가 '_'일 때 삭제

```
143 void MainWindow::on_Enter_clicked()
144 {
145     QString current = ui->labelOutput->text();
146     if(current.endsWith("_")){
147         current.chop(1);
148     }
149     ui->labelOutput->setText(current + '\n');
150     number=0;
151     txt_re=" ";
152 }
```

⑦ MainWindow::on_Enter_clicked()

- 문자열 줄 바꿈

```
6 void TextSave::saveFile(const QString &text) {
7     QFile file("/home/eung/output.txt");
8     if (file.open(QIODevice::Append | QIODevice::Text)) {
9         QTextStream out(&file);
10        out << text << "\n";
11        file.close();
12    }
13 }
```

⑧ TextSave::saveFile()

- QFile, QTextStream-> 파일 append 모드로 열기
- 인자로 받은 문자열 txt파일 마지막 줄에 추가