

-hw_01 : 천지인키보드 구현 -

<목차>

p.2 - 문제 분석

- 1. 문제요지
- 2. 필수조건
- 3. 핵심요약

p.4 - 알고리즘 설명 및 구현

- 1. 영어
- 2. 한글
- 3. 기타

p.6 - result screen

과제1

천지인 키보드 과제

[필수조건]

- 대소문자 구현
- 입력창에 실시간으로 작성중인 문자 출력
- 엔터가 눌리면 문장 txt파일 마지막줄에 추가

[추가점수]

- 한글자판 구현
- 숫자 및 특수문자 구현
- 입력 받은 버튼의 눌림 처리(배경색상 변화 등)
- 기타 실제 천지인 키보드와 유사할수록 추가점 부여

*** mp4파일로 문장 출력 결과영상 첨부할것**

1. 문제 요지

천지인 키보드를 소프트웨어로 구현하는 과제다. 기본적으로 입력창에 문자를 실시간으로 출력하고, 엔터 키 입력 시 작성된 문장을 txt 파일에 저장해야 한다. 대소문자 전환을 포함해 키보드 기능을 모사하는 것이 핵심이다.

2. 필수조건

대문자/소문자 입력 기능 구현

입력창에 실시간으로 작성 중인 문자 출력

엔터 입력 시, 입력된 문장을 txt 파일의 마지막 줄에 추가

3. 핵심 요약

기본적으로 "영문 대소문자 입력 + 실시간 표시 + 엔터 시 txt 저장"을 만족해야 하고, 추가 점수를 위해 한글 자판, 숫자·특수문자, 키 입력 시 시각 효과, 실제 천지인과 유사한 동작을 구현할 수 있다.

핵심 흐름

버튼 입력 => 상태 결정

한글 자음: MultiTapHandler

클릭 길이와 연타 간격으로 c1/c2/long 선택.

타이머 만료 시 확정 입력. 만료 전 연속 클릭이면 치환.

확정 시 MainWindow::insertJamo() 호출.

한글 모음: VowelComposer

pressFloorU/pressDotA/pressSunl → commitCombine()

Out{replaceLast,ch} 반환. replaceLast==true면 직전 글자 삭제 후 치환 입력.

영문: handleT9()

같은 키를 탭 간격(ENG_TAP_MS) 안에 반복하면 사이클 인덱스만 증가하고 직전 문자를 삭제 후 치환.

합성·치환 => 문서 반영

insertJamo(QChar j)

직전 글자가 한글 음절이면 분해(L,V,T).

모음 입력 => 받침 있으면 받침을 다음 음절 초성으로 이동 후 새 음절 구성.

자음 입력 => 받침 비어 있으면 받침에 결합. 이미 받침 있으면 새 글자 시작.

최종 문자 QTextCursor로 삽입.

제어 키 입력

Space/Enter/Delete → 즉시 삽입·삭제.

이 때 한/영 T9 상태, 특수문자 사이클, 모음 pending 모두 초기화.

특수문자 사이클

.,?! 버튼은 내부 카운터로 순환. 같은 버튼 연속이면 직전 기호 삭제 후 치환.

주의 포인트

페이지 전환 함수가 뒤바뀜

on_kor_eng_change_kor_clicked()가 pageEn으로,

on_kor_eng_change_eng_clicked()가 pageKo로 전환함. 서로 바뀌어야 정상.

UI 객체명 오타자

connect(ui->shfit_capslock, ...) 오타자. .ui의 objectName이 shift_capslock인지 확인하고 통일.

Enter 시 파일 저장 미구현

요구사항: 엔터 누르면 현재 줄을 txt 마지막 줄에 추가.

제안: on_enter_*에서 append("") 뒤

QFile(QStandardPaths::writableLocation(AppDataLocation)+"/output.txt", Append)로 저장 루틴 호출.

모음 3타 규칙 검증

DDF→ㅈ 등 커스텀 규칙 존재. 실제 천지인과 다르면 채점 시 감점. 명세와 일치 확인 필요.

겹받침 분해 미완성

insertJamo()의 받침 분해는 단일 받침 위주. ㄱ, ㄴ, ㄷ, ㄹ ... 케이스 분해·이동 로직 보강 필요.

초성 단독 글자 처리

“자모 단독 → 모음 입력 시 합성” 분기 단순화 주석. 실제로는 직전 글자가 음절인지 자모인지 확실히 판별해야 오동작 방지.

상태 초기화 일관성

Space/Enter/Delete에서 resetVowel(), clearKorDotState(), lastT9Key.clear()를 모두 호출해 교차 상태 잔류 방지.

타이밍 상수 튜닝

LONGPRESS_MS=280, MULTITAP_MS=270, ENG_TAP_MS=700은 기기·과제 기준에 맞게 조정. 멀티탭·롱프레스 오인식 시 체감 품질 저하.

커서 경계 안전성

치환 시 !cur.atStart() 검사 이미 존재. 추가로 선택 영역이 있을 때 덮어쓰기 동작 정의 필요.

한/영 전환 시 초기화

레이아웃 전환 직후 T9 인덱스, 모음 pending, 특수문자 사이클 카운터를 초기화해 모드 간 상태 누수 방지.

<알고리즘 설명 및 구현>

1. 영어

1) T9 멀티탭: handleT9(keyId, lower, upper)

동일 키를 ENG_TAP_MS(700ms) 내 반복 입력하면 “치환 모드”.

커서 앞 글자 삭제 후 다음 후보로 치환.

후보 인덱스는 lastT9Index로 모듈러 순환.

다른 키로 바뀌면 인덱스 초기화 후 첫 후보 삽입.

engCaps가 참이면 upper 후보군 사용. 아니면 lower.

스페이스/엔터/삭제는 “확정 입력”이라 T9 상태 초기화.

핵심: 같은 물리키 연속 입력 시 이전 문자 삭제→대체. 시간 경과로 확정.

2) 시프트/캡스

on_shift_capslock_clicked()가 engCaps 토글.

연결 위젯 이름 오탈자 가능성: ui→shfit_capslock로 연결. .ui의 objectName과 일치해야 동작.

2. 한글

1) 자음 키 입력 처리: MultiTapHandler

입력 모델: 단탭1→c1, 단탭2→c2, 길게→clong.

MouseDown에서 타이머 대기 시작. MouseRelease에서 길게 여부 결정.

길게(≥LONGPRESS_MS=280ms): insertJamo(clong) 즉시 확정. 상태 초기화.

짧게: 탭 카운트 증가. MULTITAP_MS(270ms) 내 추가 탭 없으면 onTimeout()에서 확정.

확정 규칙: 1탭이면 c1, 2탭이면 c2. 연속 멀티탭 사이 모음 상태·특수기호 상태 초기화.

핵심: 물리 버튼은 “자음 문자”만 생성. 실질 합성은 insertJamo()가 수행.

2) 모음 조합기: VowelComposer

상태: pending_에 'l(l)/D(·)/F(—)' 시퀀스를 보관.

1타(commitSingle): 해당 단일 모음 출력. pending_ 유지해서 다음 타에 치환 가능.

2타(map2 경로):

"ID"→ㅓ, "DI"→ㅕ, "DF"→ㅗ, "FD"→ㅜ, "DD"→ㅛ (특수) 등.

출력은 replaceLast=true로 직전 모음을 치환. pending_는 마지막 키 1글자만 남김.

3타 확장(commitCombine):

"IDI"→ㅓㅓ, "DII"→ㅓㅕ, "DFI"→ㅓㅗ, "FDI"→ㅓㅜ, "FI"→ㅓㅛ, "DDF"→ㅓㅛ.

매칭 실패 시 마지막 키 단독 모음으로 확정.

UI 연결: Vowels_floor(—)/sun(·)/people(l) 클릭 →

pressFloorU/pressDotA/pressSunl() → 반환값이 replaceLast면 커서 앞 글자 삭제 후 insertJamo(o.ch).

핵심: 모음은 점진적 치환 모델. 1타 찍고 2·3타에서 같은 자리 치환.

3) 음절 합성기: MainWindow::insertJamo(QChar j)

테이블: 초성 L(19), 중성 V(21), 종성 T(28) 인덱스 매핑.

make(L,V,T)로 U+AC00 블록 조합.

split(가)로 L,V,T 분해.

커서 앞 글자 조사. 한글 음절 아니면 입력만 삽입 후 종료.

입력이 모음($V_i \geq 0$):

직전 음절 $T == 0$ 이면 두 가지:

직전이 “자모 단독 자음(ㄱ 등)”이면 그 자모와 합쳐 L+V로 치환.

아니면 새 모음으로 그냥 삽입.

직전 음절 $T > 0$ 이면 받침 분리 규칙:

기존 음절을 $T = 0$ 으로 바꿔 재삽입(받침 제거).

제거된 받침을 다음 음절의 초성으로 이동하여 새 음절 `make(moveL, V_i , 0)` 생성.

여기서는 단일 받침만 이동하는 단순화. 겹받침 분해는 생략 또는 일부 매핑만 지원.

입력이 자음($L_i \geq 0$):

직전 음절 $T == 0$ 이면 가능한 경우 종성으로 흡수(`make(L,V,t)`).

이미 종성 있음($T > 0$)이면 새 자음 글자 시작.

그 외 문자는 그대로 삽입.

핵심: 자모 단독 \leftrightarrow 음절 변환, 받침 이동, 모음 치환을 최소 규칙으로 처리.

3. 기타기능

1. ., ?/.

`on_dot_com_que_exc_kor_clicked`: 같은 버튼 연속이면 직전 기호를 치환하며 .→,→?→! 순환.

2. 한/영

함수명과 주석이 상충.

`on_kor_eng_change_kor_clicked()`가 `pageEn`으로 전환.

`on_kor_eng_change_eng_clicked()`가 `pageKo`로 전환.

=> 사용 의도에 맞게 두 함수의 대상 위젯을 교체해야 일관성 확보.

<result screen>

영상 참조 바람.

- 끝 -