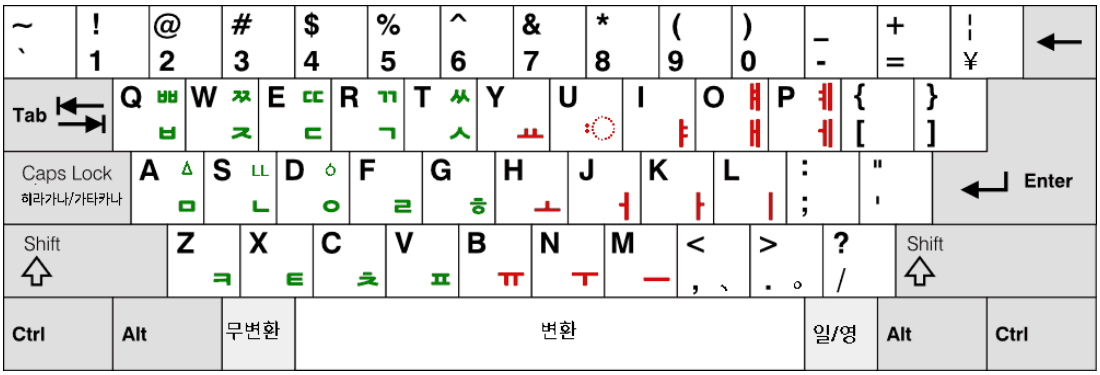
1. 개념

한글 사용자들을 위해 가나 문자 입력 방식으로 일본어 발음을 로마자로 입력하는 것을 대신하여 가나 발음을 한글로 입력함으로써 일본어 문서의 생산을 가능케 하는 것이 이 "hanjp im"의 기본 개념이다. "hanjp im"에서는 기존의 시스템들과 달리 옛한글을 도입하면 일본어의 표기가 쉬워진다는 것을 착안해 옛한글을 도입해 정확하고 편한 입력 시스템을 구축하고자 한다. 그리고 한글 사용자를 배려한 편리한 기능을 구현하여 실용적이고 합리적인 입력 체계를 구축하고자 한다.

2. 자판 배열

다음은 기초적으로 KS X 5002 한글 두벌식 배열 사용자를 고려해 편의성을 절충하여 나온 자판이다.



위 자판을 보면 일본어 입력 모드와 영어 입력 모드에서 몇 개의 키 바인딩을 다르게 했고 'ㅿ', 'ㅥ', 'ㆁ'등의 몇 개의 옛한글 자소를 도입하여 정확하게 일본어 음을 구분했다. 그리고 일본어에서 사용되지 않는 'ㅖ'음 자리에는 방점을 대신 넣어 쉬운 장음 입력을 하게 했다.

3. 기초 규칙

과 → ごぁ

기본적으로 모음이 자음 없이 있는 경우에는 'ㅏ, ㅣ, ㅜ, ㅖ, ㅗ'는 'ぁ, ぃ, ぅ, ぇ, ぉ' 와 같이 작은 가나 문자로 바뀌고 자음과 조합되는 경우에는 아래 '키 조합 매칭'에서 소개되는 것과 같이 변환이 된다. 그리고 '과', '궤', '괴' 같은 이중모음은 'ㅗㅏ', 'ㅜㅖ', 'ㅗㅣ'와 같이 분리되어 최종적으로 위의 그림과 같은 형태로 변환된다.

4. 키 조합 매핑

키 조합 매핑은 오십음도에 따라서 하되 일본어에 없는 모음인 'ㅓ'는 'ㅗ'로, 'ㅕ'는 'ㅛ'로, 'ㅖ/ㅒ'는 'ㅣ'+'ㅐ'로 조합해서 쓴다. 또 요음은 'ㅣ' + '해당 모음'으로 한다. 그리고 한글에 없는 자음인 'w'는 '와', '오'로만 조합하게 하여 해결한다. 이것을 따르며 각각의 자모의 인덱스를 찾아서 2D 배열에서 문자를 찾는다. 다음은 키 조합 표이다.

	ㅏ	ㅣ	ㅜ/ㅡ	ㅖ/ㅒ	ㅗ
ㅇ	あ[아]	い[이]	う[우]	え[에]	お[오]
ㅋ/ㄲ	か[카]	き[키]	く[쿠]	け[케]	こ[코]
ㄱ	が[가]	ぎ[기]	ぐ[구]	げ[게]	ご[고]
ㅅ/ㅆ	さ[사]	し[시]	す[수]	せ[세]	そ[소]
ㅈ	ざ[자]	じ[지]	ず[주]	ぜ[제]	ぞ[조]
ㅌ/ㄸ, ㅊ/ㅉ	た[타]	ち[치]	つ[츠]	て[테]	と[토]
ㄷ, ㅿ	だ[다]	ぢ[ㅿ]	づ[스]	で[데]	ど[도]
ㄴ	な[나]	に[니]	ぬ[누]	ね[네]	の[노]
ㅎ	は[하]	ひ[히]	ふ[후]	へ[헤]	ほ[호]
ㅂ	ば[바]	び[비]	ぶ[부]	べ[베]	ぼ[보]
ㅍ	ぱ[파]	ぴ[피]	ぷ[푸]	ぺ[페]	ぽ[포]
ㅁ	ま[마]	み[미]	む[무]	め[메]	も[모]
야	や[야]	X	ゆ[유]	X	よ[요]
ㄹ	ら[라]	り[리]	る[루]	れ[레]	ろ[로]
와	わ[와]	X	X	X	を[오]
ㅛ	ん[응]	X	X	X	X

5. hangul_to_kana 설명

hangul_to_kana 함수는 국소적으로 한글을 가나 문자로 바꿔주는 함수이다. 기본적으로 원음 + 보조음의 형식으로 바꿔준다. 예) 과 → ごぁ

hangul_to_kana의 인자는 (dest, prev, hangul, next, type)으로 처리 과정은 다음과 같다.

1) prev를 보고 hangul이 종성인지 판단

2) hangul[0]로 자음 인덱스 찾고 is_choseong_void 찾기

3) hangul[1]로 모음 인덱스 찾고 is_jungseong_void, has_contracted_sound 찾기

4) 옵션 값에 따라 자음 인덱스 바꿔주고 예외 처리

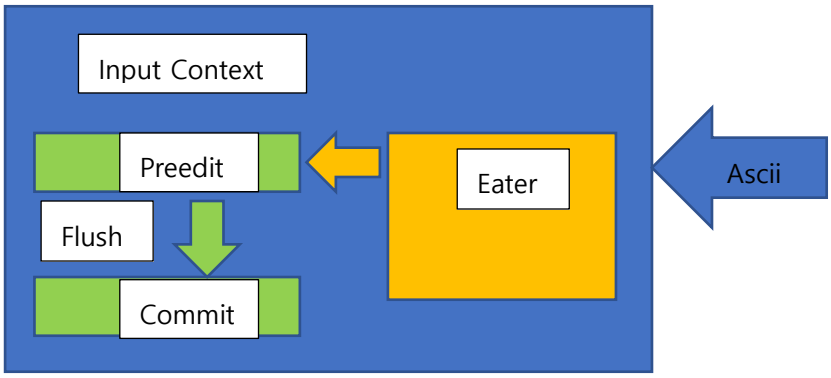
5) 모음 분리하고 보조음 찾기

6) Kana_table에서 음 찾고 보조음 덧붙이기

2-1) 종성이면 is_comport인지 판단

2-2) 받침으로 변환

6. Input Context 디자인



위는 input context의 디자인이다. Eater는 변환 로직으로서 ascii가 들어오면 hangul input context를 활용해서 한글을 입력하다 가나로 바꿔서 preeedit에 넘겨준다. 그리고 input conext에서는 flush 할 때 마다 commit string에다 옮겨준다. 위 과정을 자세히 살펴보면 다음과 같다.

1) 키배열에서 ascii를 얻는다.

2) Eater에서 Hangul input context를 이용하여 자소로 매핑하고 한글 preedit을 한다.

3) 한글 preedit string이 commit 되면 hangul_to_kana 함수를 이용해서 가나로 변환해준다.

4) Input Context의 preedit으로 옮겨준다.

5) Commit 되면 commit string으로 옮겨준다.

7. 기능 구상

Input Context에서 제공할 기능들을 정리한 것이다.

1. 히라가나 -> 한자(mozc 활용)
2. 한글 단어 -> 히라가나(단어 사전화)

   예) Ctrl +Shift 누르고 '나' -> わたし

3. 한글 -> 한자
4. 히라가나 -> 한글