# DSL KeyPad — инструмент для многоязычного письма, требующий лишь QWERTY/ЙЦУКЕН.

«DSL KeyPad» — инструмент, дающий пользователю возможность вводить широкий спектр символов юникода посредством ряда функций, описанных ниже. Написан на AutoHotkey V2.

Основной фокус инструмента — расширение доступных ко вводу символов латиницы и кириллицы, при этом требуя от пользователя наличия только стандартных QWERTY/ЙЦУКЕН раскладок.

Идея наличия в системе ряда раскладок для сходных на письме языков кажется мне неудобством — зачем, если можно поверх «единого фундамента» добавить какие захочешь закорючки? И они всегда будут располагаться на одном месте вне зависимости языка. Кто-то может и такое посчитать неудобным, но выбранный мной способ решения проблемы многоязычного ввода для меня же кажется более привлекательным.

## Краткая «история»

Изначально инструмент родился из собственной нужды вводить ряд нестандартных символов. По началу я использовала «Типографскую раскладку» Ильи Бирмана, но в итоге и этого мне оказалось мало. Мне нужен был простой способ вводить шпации, диакритику, расширенную кириллицу и латиницу — для написания лора собственного вымышленного мира и использования сихъ знаковъ в писательстве.

Первая версия исполнения была простой как пробка: ~20 строчек простых биндов вида «<^<!e:: Send("{U+045E}") ; (Ў)». Однако со временем приходили новые идеи и желание сделать что-то более сложное, что могло бы покрыть больше моих же потенциальных потребностей. Как итог — процесс сильно затянулся и получился какой-то монструозный ООП’оид, но результатом я довольна.

Влетала я в это со слабыми навыки в написании кода и постепенно училась в процессе разработки этой штуки её, собственно, и разрабатывать. Конечно, с использованием нейронок, но скорее как указатель на ошибки, справочник и «я тут накакала, сделай по красивее» — «вайбкодить» у меня получалось крайне редко, не умею я с железякой контактировать… и хорошо.

## Краткая сводка фичей инструмента

1. Поддержка широкого спектра символов юникода. На данный момент — более 3300 символов, включая, помимо озвученного: пунктуация, математические символы, символы валют, различные специальные символы, алхимические/астрологические/астрономические символы.
2. GUI для отображения списков доступных символов с горячими клавишами или последовательностями ввода (см. далее); окно настроек; окно для просмотра, создания/изменения/удаления пользовательских рецептов.
3. «Многослоный» набор горячих клавиш, по большей части раздельных для английского и русского языков (например, [англ.] RAlt + A = Ă; RLAlt + A = Â; RAlt + RLShift + A = Ã; [рус.] RAlt + Ф = Ѳ; RAlt + Я = Ѧ).
4. Режим композиции, позволяющий из последовательности одних символов получить какой-то другой (например: AE → Æ, OE → Œ, TH → Þ). По мере ввода будет будет отображаться подсказка с совпадающими последовательностями.
   1. Включает поддержку «пользовательских рецептов». Пользователь может создать как простой рецепт с выводом одного символа, так и рецепт, результатом которого будет вывод многострочного текста, вплоть хоть до сниппета кода или главы книги.
5. «Альтернативные режимы» для ввода форм письменности, отличных от латиницы и кириллицы (Греческая письменность, Германские Руны, Глаголица, Финикийское письмо, Древнесеверноаравийское и т.д.), а также для ввода Международного фонетического алфавита.
6. Режимы для ввода альтернативных форм символов (например: 𝑎, 𝐚, 𝒂, 𝔞, 𝖆, 𝒶, 𝓪, 𝕒, 𝚊, ᴀ).
7. Возможность переключать тип ввода — HTML-энтити/мнемоника и LaTeX (если доступно) вместо символов юникода.
8. Возможность искать символы по их «тегам» в локальной библиотеке (например: «latin capital ligature AE with macron» → Ǣ; запрос может содержать неполный тег: «cap ae mac» → Ǣ; «одал» → ᛟ).
9. Возможность добавлять символы библиотеки в избранное — это позволит всегда отображать их рецепты в подсказке режима композиции, а также отметит избранные символы в GUI звёздочкой.
10. «Внутренние» раскладки клавиатуры (QWERTY, Dvorak, Colemak | ЙЦУКЕН, Диктор ЙІУКЕН 1907) (инструмент не очень хорошо работает с системными раскладками, отличными от QWERTY/ЙЦУКЕН).
    1. Включает поддержку «пользовательских раскладок».
11. Поддержка пользовательских биндов, однако ограниченных символами локальной библиотеки.
12. Отдельные вспомогательные режимы для облегчения/ускорения письма на вьетнамском языке/пиньине.
13. Прочие различные возможности…

## Подробнее об основных фичах

### Горячие клавиши (они же — «Быстрые ключи»)

Их много, просто невероятно много, и опытный пианист, безусловно, справится с ними. А если серьёзно — на каждую клавишу может приходиться до 10–11 возможных комбинаций (но в среднем — около 8) (латинская I в этом рекордсмен: Ĭ Ï Ī Ĩ Î Ǐ Į İ Í Ì Ȉ).

Из-за своего же числа, по умолчанию — выключены, но переключаются на RAlt + F1. Состояние вкл/выкл сохраняется в конфигурационном файле.

Делятся комбинации на ряд условных подгрупп, по «иницирующим» модификаторам: LCtrl LAlt-группа; LAlt-группа; RAlt-…; LShift-…; RShift-….

Больше всего привязок включает в себя RAlt-группа, и она же является основной. На втором месте LCtrl LAlt-группа, чей фокус сосредоточен на комбинируемой диакритике, но ею не ограничивается. Остальные — «по остаточному принципу».

Для привязок латинских букв я попыталась сделать какую-то логику для большей интуитивности, например: все комбинации RAlt + LAlt + <БУКВА> = <БУКВА> с циркумфлексом (ŴÊŶÛÎÔÂŜĜĤĴẐĈ). Однако это не заимело полноценной реализации — части символов с соответствующими диакритическими знаками в юникоде по просту не существует, и в то же время есть буквы с более высоким приоритетом над другими. Так, например, буквы Å и Ů заменили буквы Ǎ и Ǔ на позиции RAlt + LAlt + LShift <БУКВА>, изначально задуманной для букв с гачеком (ĚŘŤǏǑŠĎǦȞǨĽŽČŇ), так как, по моей оценке, надобность в их вводе будет возникать значительно чаще, чем в случае A и U с гачеком. В случае же кириллицы — там тот ещё разброд и шатание, лишь в единичных случаях пытающийся соответствовать описанной выше логике: RAlt + RShift + <БУКВА> = ӢӮ ([лат.] ĀĒȲŪĪŌḠ).

Карту привязок можно представить следующим образом:

Пришлось несколько «нестандартно» отобразить альты и шифты, чтобы уместить на одном изображении сразу кучу комбинаций, а потому небольшое пояснение: если какой-то цвет встречается на позиции альтов и/или шифтов — значит их и нужно нажимать для получения соответствующего символа.

< ВСТАВИТЬ КАРТИНКИ С БИНДАМИ >

### «Плавильня»

Упомянутый выше «режим композиции» является основным способом использовать то, что я назвала «Плавильней» — фичу для «переплавки» последовательности одних символов в другие. Вдохновляясь, конечно, Compose key из Линукса.

Композиция активируется двойным нажатием на RAlt — ввод пользователя будет отображаться во всплывающей подсказке, крепящейся к полю ввода (или к курсору, если по какой-то причине поле ввода не обнаружено). Там же, по мере ввода, будут отображаться предложения совпадающих рецептов (\*последовательностей), при этом рецепты избранных символов будут отображены всегда.

< ВСТАВИТЬ АНИМАЦИЮ КОМПОЗИЦИИ >

«Плавильня» позволила значительно расширить возможности ввода. Клавиатура не резиновая и бесконечное количество привязок, увы, сделать не получится, тогда как композиция принимает любой ввод (в том числе с «Быстрых ключей») и позволяет печатать прямо с клавиатуры ощутимо больше символов.

Нюанс — режим композиции триггерится на первое точное совпадение рецепта, что означает невозможность ввести, например, Ǣ. Однако композицию можно поставить на паузу нажав клавишу… паузы. В таком состоянии можно будет вводить любые схожие рецепты не опасаясь, что ввод сорвётся первым же совпадением. Для выхода из паузы нужно повторно нажать паузу или нажать энтер.

Вместо паузы при вводе можно использовать «оператор разделения» — символ «`». Это не предотвратит триггер на первое точное совпадение, но позволит продолжить ввод последовательности, если есть рецепты, начинающиеся с результата первого совпадения, например, AE → Æ, AE<МАКРОН> → Ǣ; тогда — RAlt×2: A`E → (Æ) → Æ<МАКРОН> → Ǣ.

< ВСТАВИТЬ АНИМАЦИЮ С ` >

С активной паузой можно использовать ещё несколько «операторов»: (ЧИСЛО), (~) и (ЧИСЛО~) — пишутся в начале строки и отбиваются пробелом от ввода; «ЧИСЛО» — количество повторений результата (например, «(4) alc sulfur» → «🜍🜍🜍🜍»); тильда же позволяет обрабатывать комбинации внутри какой-либо фразы, например, попытка ввести «apa<КРАТКА>» приведёт к ошибке «рецепт не найден», но «(~) apa<КРАТКА>» успешно вставит румынское «apă» (вода).

И ещё есть оператор «##», который нужен для предотвращения обработки нежелательных последовательностей: ввод «(~) tho<АКУТ>rr» приведёт к вставке «þórr», но если мы хотим оставить «th», тогда вводить следует «(~) t##ho<АКУТ>rr» — «thórr».