Miryoku's Bilateral Combinations
for Home Row Mods in QMK
as tamed by Suraj N. Kurapati

https://sunaku.github.io/home-row-mods.html

any key is tapped

mod key is held

mod key is released

are we active?
- yes → **EXTEND**
- no → are we in the middle of a typing streak? (check activity timer, as configured by...)

configured by → #define BILATERAL_COMBINATIONS_ TYPING_STREAK_TIMEOUT (amount of time in milliseconds)

are we in the middle of a typing streak?
- yes → do the held key's mods match this bitwise mask? (as configured by...)
- no → **BEGIN**

do the held key's mods match this bitwise mask? (as configured by...)
- configured by → #define BILATERAL_COMBINATIONS_ TYPING_STREAK_MODMASK (bitwise mask of modifier flags)
- no → **BEGIN**
- yes → UPDATE activity timer

EXTEND → is the held key on the same side as the beginning key?
- no → CONVERT hold to tap
- no → TRACK typing streak

are we active? (left)
- yes → has the beginning key been held long enough? (as configured by...)

has the beginning key been held long enough? (as configured by...)
- configured by → is the tapped key on the same side as the beginning key?
- no → FLUSH chord keys as taps

is the tapped key on the same side as the beginning key?
- yes → #define BILATERAL_COMBINATIONS_ ALLOW_SAMESIDED_AFTER (amount of time in milliseconds)
- no → #define BILATERAL_COMBINATIONS_ ALLOW_CROSSOVER_AFTER (amount of time in milliseconds)

CONVERT hold to tap
- #1 → FLUSH chord keys as taps
- yes → FLUSH chorded mods
- #2 → CALL tap_code()

FLUSH chord keys as taps → did we flush anything yet?
- no → CONVERT chord to taps
- yes → FLUSH chorded mods

CONVERT chord to taps
- #1 → SET flushed = true
- #2. for each key in chord → CALL del_mods()
- #3. for each key in chord → CALL tap_code()

FLUSH chorded mods
- #2 → SET flushed = true
- #1 → APPLY chorded mods

TRACK typing streak
- #2 → CALL tap_code()

BEGIN
- #5 → UPDATE activity timer
- #4 → APPEND key to chord
- #3 → STORE this key as the beginning key
- #2 → SET flushed = false
- #6 → SCHEDULE mods registration
- #1 → SET active = true

APPEND key to chord
- configured by → #define BILATERAL_COMBINATIONS_ LIMIT_CHORD_TO_N_KEYS (maximum number of keys in chord)

SCHEDULE mods registration → do the held key's mods match this bitwise mask? (as configured by...)
- configured by → #define BILATERAL_COMBINATIONS_ DELAY_MODS_THAT_MATCH (bitwise mask of modifier flags)
- yes (delay mods registration) → CALL defer_exec()
- no (don't delay registration) → APPLY chorded mods

CALL defer_exec()
- configured by → #define BILATERAL_COMBINATIONS_ DELAY_MATCHED_MODS_BY (amount of time in milliseconds)
- call (later on... as scheduled) → CALLBACK for defer_exec()

CALL cancel_deferred_exec()
- cancel → CALLBACK for defer_exec()

CALLBACK for defer_exec() → are we active?
- yes → APPLY chorded mods
- no → IGNORE

APPLY chorded mods → did we flush anything yet?
- yes → PASSTHROUGH key will be registered with the current mods
- no → CALL register_mods()

END
- #2 → CALL cancel_deferred_exec()
- #1 → SET active = false

END (mod key is released branch)

are we active? (right)
- yes → is the released key the same one as the beginning key?
- no →

is the released key the same one as the beginning key?
- yes → are the released key's mods the same as the beginning key's mods?
- no → RELEASE

are the released key's mods the same as the beginning key's mods?
- yes → #3 CALL unregister_mods()
- no → RELEASE

RELEASE
- #2 → CALL unregister_mods()
- #1 → DELETE key from chord