**Notation 2**

This is the standard format used, with the extension **.song**

The song is constructed as a sequence of grammatical entities separated by vertical bars. Bars must be complete on each line. This means the last character of each line (apart from spaces and comments) must be a vertical bar.

Lyrics

Lyrics are specified using speech marks e.g. “Hello world”. The presence of an asterisk in the lyric forces spaces to be inserted at that point, thus allowing right justification “\*Hello” or to force words to opposite ends of the bar “end.\*My”. Lyrics are optional.

Notes

Individual played notes are done using a combination of the digits 0-9,X,T,E,W the ^ character, the + character and a comma. The comma is used for clarity (e.g. 060 must be 0,6,0 but 120 would be analysed as 12,0 (because 12 is a legal fret)). 1 2 0 would have to be encoded as 1,2,0 or 1,20

Notes are left justified with either X or 0 – depending on assignment “drone” (which defaults to Yes). So 2,4 would be actually X,2,4 and 11 would be X,X,11

The ^ character acts as a string shift by appending X to the end instead (so ^4 is the same as 4X), each ^ shifts to the next string in.

Notes always advance the time by one beat, this can be modified with the adjusters (see below)

TEW are supported as shorthand for 10,11,12 for backwards compatibility but their use is not advised.

Adjusters

The standard position adjusters and post note modifiers are used:

|  |  |  |
| --- | --- | --- |
| Character | Offset | Description |
| o | +1 | Advance one beat |
| & | +1 | Advance one beat (rest) |
| . | +0.5 | Advance half a beat |
| - | -0.5 | Back half a beat – note to quaver. |
| = | -0.75 | Back three quarters of a beat – note to semiquaver. |

Rhythm Sections

@n is a bar sets the current rhythm patterns (which defaults to 1). These patterns are set in the assignments using pattern\_<n> , v n and . The pattern must be twice the length of the number of beats (e.g. pattern\_1 := v.v.vnv.)

Actual rhythm bars are set using square brackets with one entry per beat. To space between entries use a -. There need not be an entry for every beat (so [G – Bm] will play G on beats 1+2 and Bm on beat 3+4 with 4 beats in a bar). [] is also permitted which means “play the same rhythm using the last chord in the previous bar”.

Chord name are built up using a name “Bb7” and a hidden tag (so multiple D chords can be used, but they are all labelled “D”. The separator is a . (e.g. F#5.wierd) – but it will *only* display the F#5

Chords are defined by being entered as assignments e.g. chord\_F#5.wierd := 3,2,3 using the same format as notes.

Bars must be either rhythm or chords, they cannot be both. After checking for pattern, chord definitions and lyrics, if there is anything left in the bar, and chords are played, an error will occur.

If lyrics contain a forward slash (e.g.) [G] “hello world/I like/you” this will cause three bars to be generated [G]”hello world” [G]”I like” [G]”you”.

Macros

Macro definitions are lines that begin with a colon, and are named a-z e.g.

:a <stuff>

From that point on rather than being processed it is added to the macro with a space separator (removing comments), until a line is found ending in a semicolon (which can be the first line).

When processing ; a line must only contain $<name> except spaces and comments, if this is found the line compiler is called recursively once with the contents of the macro. The macro in total must therefore conform to the same syntax as the line.

(and yes, I do like FORTH ….)