NinjaTracker V2.04

1. Introduction

NinjaTracker V2.x is still a somewhat minimal music editor. Main differences to previous versions are general purpose commands (or instruments), two-column tables and a slide function that knows to stop at the target pitch.

Customization is allowed and encouraged!

```
www:   covertbitops.c64.org
email: loorni@gmail.com
```

2. How to use

2.1 General keys

F1	Play from beginning
F2	Toggle fast forward
F3	Play from mark
F4	Stop playing
F5	Enter help screen
F6	Adjust colors
F7	Octave up
F8	Octave down
←	Enter disk menu
/	Silence test notes
<>	Fast scroll up/down
0-9,A-F	Edit hexadecimal data
Cursors	Move around
Ins/Del	Delete rows
Shift+Ins	Insert rows
Shift+M	Mark copy start/end
Shift+X,C,V	Cut/copy/paste
Shift+1,2,3	Switch to tracks 1-3
Shift+4	Switch to pattern
Shift+5,6,7	Switch to tables
Shift+8	Switch to commands
Return	Goto pattern/table/command
Shift+Ret.	Point and goto next unused pattern/table/command

2.2 Track editor special keys

;:	Select sub tune
Space	Mark playing position

2.3 Pattern editor special keys

;:	Select pattern
-+	Select command number
ZSXDCVGBH..	Lower octave notes
Q2W3ER5T6..	Upper octave notes
Space	Enter key off / clear column
Shift+Space	Enter key on
Shift+Q	Transpose half step down
Shift+A	Transpose half step up
Shift+L	Toggle command legato
Shift+O	Optimize pattern
Return	Fill with above note

2.4 Command editor special keys

```
Space       Key off test note
Shift+Space Test current command
Shift+S     Smart paste (references of the source command are
            pointed to destination)
```

Testing and cut/copy/paste/ins/del works only when the cursor is over the command parameters, not command name. The test note is `C` in the currently selected octave and on the channel active in the track editor.

3. The music data

3.1 Track data

There can be a maximum of 16 different songs (sub tunes), each with 3 tracks.
All songs share the same 127 patterns, tables and 127 commands.

Values in the track data:

```
00    Loop (followed by loop position)
01-7F Pattern to play
80-BF Transpose downwards
C0-FF Transpose upwards (C0 = zero)
```

Transpose cannot be followed by loop, and the combined length of a sub tune's all tracks cannot exceed 256 bytes.

A sub tune that plays only once can be realized by playing a silent pattern (with just a long key off note) last and looping to it indefinitely.

3.2 Pattern data

A pattern consists of four columns. From left to right they are:

```
Note/Key off/Key on
Command number 01-7F, or legato 81-FF
Duration (using decimal notation)
Command name (not editable)
```

A note can range from `C-1` to `B-7`. A note without a command number will use the last used command. Similarly, if the duration column is empty, the last used duration will apply.

Command numbers `81-FF` are the commands `01-7F` called in legato mode. In legato mode hard restart, init frame waveform setup and auto key on will be skipped (when used with a note), as well as `ADSR` setup; only the table pointers are set.

Duration minimum is `03` and maximum is `65`.

Key off is shown as `---` and key on as `+++`. There is no function to let the gate mask stay in its current value, sorry!

3.3 Table data

In all tables, the left side selects the command/function, and right side has additional parameters for that function. Jump destination 00 will stop execution of that table.

Wavetable left side values:

```
00-8F Set waveform, right side is arpeggio (00-7F relative, 8C-DF
      absolute notes)
90-BF No waveform, delay arpeggio by 00-2F frames
C0-DF Vibrato with speed 00-1F, right side is depth
E0-FE Slide with speed high byte 00-1E, right side is speed low byte
FF    Jump, right side is destination, not to be entered directly from
      a command
```

Vibrato continues indefinitely. For a delay before vibrato starts, a delayed arpeggio step can be used.

When slide reaches target pitch, it jumps to the last “set waveform”-step executed before the slide started.

Pulse table left side values:

```
01-7F Modulate pulse for 01-7F frames, right side is signed mod.speed
80-FE Set pulse to right side value
FF    Jump, right side is destination, can be entered from a command
```

Filter table left side values:

```
01-7F Modulate cutoff for 01-7F frames, right side is signed mod.speed
80-FE Set passband (left nybble-8), channels to be filtered (right
      nybble) and cutoff (right side)
FF    Jump, right side is destination, can be entered from a command
```

When setting filter passband/channels/cutoff, resonance will also be set to the left nybble of the left side byte.

3.4 Command data

Commands act both as instruments (when used with a note) and as general pattern commands to alter some part of the sound (without notes). A command sets ADSR and may set any or all of wave-, pulse- and filter table pointers.

The format of a command is:

```
ADSR Wv Pu Fl
```

A pointer value 00 leaves that pointer unchanged, letting the currently running table program (if any) continue.

Commands can be named so that using them in patterns becomes easier.

To avoid setting ADSR, use the command in legato mode (cmd. numbers 81-FF).
Furthermore, the packer / relocator can optimize away the ADSR data of commands that are only used in legato mode, if they are put to the end of the command list.

You cannot directly stop pulse/filter execution from a command, but you can achieve this by pointing table execution to a FF 00 step.

3.5 Global settings

These are accessed from the disk menu and allow setting the sustain/release value used in hard restart (default 00) as well as the note init frame waveform (default 09). They are also saved with each song. To get brighter attack to noise waveform, try init frame waveform 01 (no testbit).

3.6 Playback optimizations

New note data is read from the pattern 3 frames before the note starts. On this frame slide, vibrato and pulse are all skipped.

Track data (only if necessary) is read one frame before note start. Pulse will be skipped in that case.

When executing a command without note, both pulse and wavetable execution are skipped for one frame.

To reduce the effect of optimizations, use as long note durations as possible.

4. Packing / relocating

There are two distinct modes in the packer / relocator, Normal and Gamemusic.
Normal saves the play routine with the music data, and the calls are usual:

```
  Start+0 Init, A = sub tune
  Start+3 Play, needs 2 bytes zeropage
                (chosen at relocation)
```

In Gamemusic mode, you also have to choose the start address, but the play routine is not saved with the music. This is to save disk space in a game with lots of music modules. See the Gamemusic player source code (`nt2play.s`) and the example (`example.s`) on how to use.

To adjust volume of playback, find the instructions `ORA #$0F; STA $D418` in the player code and change the value of the `ORA` instruction.

5. Closing words

See the included example tunes to best find out how this music system works in practice. Good luck, and have fun!

Version history

```
V2.0  - Original

V2.01 - Gamemusic sound effect routine optimized
      - ins2nt2 updated for different data ordering
      - Current time position in pattern is shown alongside total duration
      - Packed size ("Ps") of pattern is shown in hexadecimal
      - Testing the last edited command also works in tables
      - ProTracker and DMC note entry modes are switchable

V2.02 – Hard restart is now 2 frames by default (init waveform has gatebit off)
      - Zeropage use reduced to 2 bytes
      - Table editor limits absolute arpeggio notes to valid range

V2.03 – Hard restart is 2 frames + 1 silent frame for note init ("hifi" style)
      - Duration range changed to 3-65
      - No duration restrictions on transpose & song loop
      - Playback optimizations changed
      - Slide goes to the last waveform/arpeggio step when finished, not to
        a delayed arpeggio step anymore

V2.04 - Reset transpose when a sub tune is played from the beginning
```