Unity PSG Player - MML reference

Format Basics

MML is fundamentally composed of a sequence of command blocks consisting of "command symbol + parameter," processed sequentially from the first character.

Spaces, line breaks, or characters not defined as commands, as well as numbers not recognized as parameters, are ignored between blocks. (*Excluding Single-line comments and the Tie command "&".)

However, if spaces or other characters appear between the command symbol and its parameters, the parameter value will not be recognized, and the command's default value will be applied.

You can use this limitation to improve readability by inserting spaces between each beat or adding symbols like "|" between measures.

The alphabet as command symbols distinguishes between uppercase and lowercase letters.

Note that even the same alphabet letter functions differently depending on whether it is uppercase or lowercase.

Numeric values treated as parameters can only be processed in decimal number format.

Additionally, parameter values must be integers only; they cannot handle digits like 0.5 or fractions like 1/3.

About Note Length

The length representation <len> associated with notes and rests corresponds to the notation of nth-value notes in the score.

A whole note is represented as "1", a half note as '2', a sixteenth note as "16", and so on.

For example, a triplet of eighth notes contains 12 notes within one measure (three notes within a quarter note), so it is equivalent to a twelfth note and is therefore written as "12".

By default, PSG Player calculates quarter notes as 960 ticks. This causes subtle multi-channel timing shifts for tuplets like 7th notes that aren't divisible by 960.

This discrepancy grows progressively larger during loop playback.

To avoid this, either set PSG Player's tickPerNote (resolution) to a multiple of the tuplet length, or use a divisible length to create a pseudo-tuplet.

(Example: 7 triplets of 16th notes (28 beats) × 7 --> 3 triplets of 32nd notes (24 beats) × 3 + 32nd note × 4)

Additionally, placing a "." (dot) after a number creates a dotted note, adding half the note's length to its length.

Multiple dots can be specified.

(Example 4... = quarter note + eighth note + sixteenth note + thirty-second note)

Command List

- **Note / Rest**
  - **c,d,e,f,g,a,b<sign><len>** Note
  - **r<len>**   Rest
  - **n<num>,<len>** Note Number

- **Time, Note Length**
  - **t<num>** Tempo
  - **l<len>** Default Note Length
  - **G<num>** Gate
  - **&** Tie
- **Pitch**
  - **o<num>** Octave
  - **>** Octave Shift (Up)
  - **<** Octave Shift (Down)
  - **T<num>** Tuning
  - **S<sign><num>** Sweep
  - **M<num>** LFO(Vibrato)
- **Volume**
  - **v<num>** Volume
  - **V<num>** Volume Envelope
- **Tone**
  - **@<num>** Tone
- **Control**
  - **L** Loop
  - **[〜]<num>** Repeat
- **Data Definition**
  - **V<num>{<vol>,〜 ,<|>, 〜,<vol>}** Envelope Definition
  - **M<num>{<delay>,<deapth>,<speed>}** LFO Definition
- **Comment**
  - **/* 〜 */** Block Comment
  - **; or //** Single-line comment
- **Truck Header**
  - **<ch> <string>** Channel MML

Command Details

Note / Rest

---

c,d,e,f,g,a,b<sign><len> Note

- Parameters
  - <sign> *+,-Flat/Sharp (do nothing when omitted)*
  - <len> *Note length [1–128] (default length when omitted)*

Plays a note in the specified scale.
Adding a "+" immediately after the command raises the note by a semitone, while adding a "-" lowers it by a semitone.
"+" and "-" can be written consecutively to represent double sharps, double flats, and others.
(Example: b-ag+gf+ = g+++g++g+gg- // The left and right sides form the same scale.)

Note that noise timbres use a 16-step scale. To determine pitch, convert the note to the note number (where o4c is 60) and use the remainder when divided by 16 (0–15).

Additionally, note that noise frequencies are assigned in descending order from the highest frequency.

<len> denotes note length, written as a number indicating the fraction of a whole note.
(Example: f1 // whole note, c8 // eighth note)

If <len> is omitted, the default note length set by the "l" command will be used.
Dotted notes are written with a dot ("."), adding half the length to the note length.
(Example c4. // quarter note + eighth note length)

Dotted notes can be written consecutively, supporting notes such as double dotted notes.
(Example: a2.. // Length of a half note + quarter note + eighth note)

---

r<len> Rest

- Parameters
    - <len> *Note length [1–128] (default length when omitted)*

It will not play sound for the specified length. <len> is the sound length.
If <len> is omitted, the default note length set by the "l" command will be used.

---

n<num>,<len> Note number

- Parameters
    - <num> *Note number [1-127] (Omitted: 60)*
    - <len> *Note length [1–128] (default length when omitted)*

It plays the note specified by the note number.
The note number is the number assigned to each key on the keyboard, with C in the fourth octave (o4c) designated as 60.
For noise sounds, the pitch is determined by the remainder (0–15) when the note number is divided by 16.

Note number and note length are separated by comma ",".

<len> is the sound length.
If <len> is omitted, the default note length set by the "l" command will be used.
When omitting <len>, you may also omit the comma ",".
(Example: l16 n60,8. n69)

Time, Note Length

---

t<num> Tempo

- Parameters
    - <num> *Tempo [1-255] (Omitted: 120)*

Specify the tempo (number of quarter notes per minute).

---

l<len> Default length

- Parameters
    - <len> *Default note length [1-128] (Omitted: 4)*

Specify the default note length.
(This is the note length when <len> is omitted for notes and rests.)

G<num> Gate

- Parameters
    - <num> *Gate ratio (%) [1-100] (Oitted: 100)*

Specifies the percentage of the specified note length that is actually played.
Since the note length remains unchanged, after playing for the percentage specified by the gate ratio, the remaining percentage is silent.
(Example: "G50 l4 c" has a duration of 50% of a quarter note, so it is equivalent to "G100 l8 c r")

Reducing the ratio allows for staccato handling.
Additionally, depending on the pitch, consecutive notes may sometimes sound connected.
In such cases, slightly shortening the articulation with a gate can separate them.
(Example: t120 o4 l4 G100 aaaa G99 aaaa // The first half sounds connected)

& Tie

- Parameters
    - None

Connect the note preceding this symbol to the note following it.
However, only notes on the same key can be connected.
(Example: a4& a16 // Play the note A for a quarter note plus a sixteenth note.)

Additionally, writing "&" immediately after a previous note without inserting a space or other character will temporarily set the gate of the previous note to 100%.
At this point, the gate will only be active for the note length of the next note.
(Example: "G50 a4& a8" and "G50 a4." have the same note length, but the actual sound duration is 1.25 beats for the former and 0.75 beats for the latter.)

Note that using "&" to connect sounds does not reset the sweep.
(Example: l4 S+10 c S-10 c // Ascending from C, descending from C
l4 S+10 c& S-10 c // Start from C, ascend, then descend from the ascended point)

Pitch

---

o<num> Octave

- Parameters
  - <num> *Octave [2-8] (Omitted: 4)*

Specify the octave directly.

> Octave Shift (Up)

- Parameters
  - None

Raise the octave one step.

< Octave Shift (Down)

- Parameters
  - None

Lower the octave one step.

T<num> Tuning

- Parameters
  - <num> *o4a (A4) frequency (Hz) [400–480] (Omitted: 440)*

Directly specify the frequency of the A note in the fourth octave (o4a), which serves as the reference point for the scale.
When playing the same note with the same tone across multiple channels, shifting the tuning adds depth to the sound.

S<sign><num> Sweep

- Parameters
  - <sign> *Plus or minus sign (+, -) (Omitted: +)*
  - <num> *Change amount (cents) [0–1200] (Omitted: 0)*

The pitch changes by the specified amount every 1/60th of a second.
The change amount is measured in cents, which is the value obtained by dividing a semitone by 100.
The change amount can be specified as a positive or negative value. A positive value causes the pitch to rise, while a negative value causes the pitch to fall.
To stop the sweep, set the change amount to 0 ("S0"; "S" alone is also acceptable).

M<num> LFO(Vibrato)

- Parameters
  - <num> *LFO Number [0–127] (Omitted: 0)*

Number 0 disables the LFO
Set an LFO (vibrato) that modulates pitch up and down over time.
If an undefined LFO number is specified, it will not be enabled.
For details on LFO content, refer to the LFO Definition.
To disable the LFO, set it to "M0" (or just "M").

Volume

---

v<num> Volume

- Parameters
  - <num> *Volume [0～15] (Omitted: 15)*

Set the volume.

V<num> Volume Envelope

- Parameters
  - <num> *Envelope number [0-127] (Omitted: 0)*

Number 0 disable the envelope
Set a software envelope that changes volume over time.
If an undefined envelope number is specified, it will not take effect (the currently set volume will continue).
For details on envelope contents, refer to the Envelope Definition.
To disable the envelope, set "V0" (or just 'V') or "v" (volume).

Tone

@<num> Tone

- Parameters
  - <num> *Tone number [0-6] (Omitted: 2)*

Set the sound tone.

Number	Wave form
0	Pulse wave (12.5%)
1	Pulse wave (25%)
2	Square wave (50%)
3	Pulse wave (75%)
4	Triangle wave

Number	Wave form
5	Noise
6	Short-cycle noise

Control

L Loop

- Parameters
  - None

When the sequence reaches the end, it returns to the "L" (loop) position.
The loop continues until playback is stopped.

[〜]<num> Repeat

- Parameters
  - <num> *Repeat count [1–128] (Omitted: 1)*

Repeat the content between the brackets "[ ]" the specified number of times.
Repeat count 1 does not repeat.
~~Nesting is not possible.~~
`v0.9.4beta` Nesting of repeat commands is now available.

Data Definition

V<num>{<vol>,〜 ,<|>, 〜,<vol>} Envelope Definition

- Parameters
  - <num> *Envelope number [1-127] (Required)*
  - <vol> *Volume [0-15] (Omitted: 15 initially; subsequent settings revert to the last volume set)*
  - <|> *Loop position (Omitted: Maintain the last volume level)*

Envelope number 0 cannot be defined.
Define a software envelope that changes volume over time.
Set the envelope number to recall it with the "V" (envelope) command.
Every 1/60th of a second, the volume changes from left to right according to the values within the "{ }" (braces).
If a volume is omitted, the first volume becomes 15, and subsequent volumes keeps to the immediately preceding volume setting.
(Example: V1{,,14,,13,} = V1{15,15,14,14,13,13} // The left and right sides have the same envelope.)

If a "|" is present within "{ }", when the envelope data reaches the end during playback, it will return to the position of the "|" and continue processing.

(Example: V1{15,13,11,|,9,10} // Volume changes to 15, 13, 11, then repeats 9 and 10.)

If "|" is omitted, the sound will play at the volume maintained after the envelope changes.

M<num>{<delay>,<deapth>,<speed>} LFO Difinition

- Parameters
    - ○ <num> *LFO number [1-127] (Required)*
    - ○ <delay> *LFO delay time [0–255] (Omitted: 0)*
    - ○ <deapth> *Pitch range [0–255] (Omitted: 0)*
    - ○ <speed> *Pitch change speed [1–255] (Omitted: 1)*

*LFO number 0 cannot be defined.
Define an LFO (vibrato) that modulates pitch up and down over time.
Set the LFO number to recall it with the "M" (LFO) command.
<delay> sets the time from the start of the sound until the LFO begins to affect it, in units of 1/120th of a second.
<depth> is the pitch modulation range, measured in cents.
<speed> is the modulation rate, specifying the LFO frequency in units of 1/16th of a hertz.

Comment

/* ～ */ Block Comment

Content enclosed within "/* ～ */" is treated as a comment.

; or // Single-line Comment

The text from the character ";" or "//" onward until the end of the line is treated as a comment.

Truck Header

<ch> <string> Channel MML

- Parameters
    - ○ <ch> *Transmit Channel [A, B, C...]*
    - ○ <string> *MML string*

*Only for MMLSplitter
Specify the channel to send to with <ch>, and <string> is the MML to send to each channel.
To distinguish between the channel and the MML, please insert a space between them.

Since the header is processed line by line, <ch> must be written at the beginning of each line.

The channel number is determined by the number of channels set in MMLSplitter, and channels are assigned in ascending order using uppercase letters starting with "A".
Additionally, <ch> can be used to specify multiple channels consecutively, such as "ABC".
In this case, the same MML is sent to each of the "A", 'B', and "C" channels.

Lines without headers will be sent to the destination specified immediately before them.