

# SNESGSS

## Guide complet de composition, samples, effets et export SPC

Pipeline pratique pour composer et graver proprement des musiques SNES avec SNES Game Sound System

### Position de départ

SNESGSS n'est pas un tracker confortable comme OpenMPT. C'est plutôt un outil de gravure SNES : banque sonore commune, données musicales légères, export SPC et fichiers prêts pour homebrew. Son intérêt majeur : une conversion BRR/SPC souvent très propre quand les samples sont bien préparés.

Version respirée - juin 2026

# Sommaire

---

1. Ce qu'est vraiment SNESGSS
2. Les fichiers et leur rôle
3. Préparer les samples correctement
4. Importer instruments .gsi et WAV
5. Comprendre le Song Editor
6. Notes, instruments, volumes et arrêt de note
7. Rows infinies, sections, longueur et loop
8. Lecture, navigation et édition
9. Effets SNESGSS : codes et exemples
10. Pan, pseudo-tremolo, detune, legato et astuces
11. Export SPC et export pour intégration SNES
12. Workflow recommandé DAC MASTER
13. Anti-pièges
14. Fiche rapide
15. Sources et notes de méthode

# 1. Ce qu'est vraiment SNEGS

---

SNEGS signifie SNES Game Sound System / Solution. L'outil est pensé pour produire de la musique et des effets sonores pour du homebrew SNES. Il n'est pas conçu comme un environnement de composition autonome et confortable.

La différence avec OpenMPT est fondamentale : OpenMPT est un atelier de composition tracker généraliste, tandis que SNEGS est plus proche d'un éditeur de données musicales pour un moteur SPC700.

## **Idée centrale**

OpenMPT aide à penser la musique. SNEGS aide à produire une sortie SNES propre. Si la conversion OpenMPT + SNESMOD clique, SNEGS peut devenir le meilleur outil pour la version finale.

## 2. Les fichiers et leur rôle

Format	Rôle
.gsm	Projet SNESGSS éditable. C'est ton fichier source de travail.
.gsi	Instrument SNESGSS déjà préparé. Ce n'est pas un WAV brut.
.wav	Sample source à importer : mono, 16-bit, idéalement 32000 Hz.
.spc	Fichier musical SNES autonome, écoutable dans un lecteur SPC ou émulateur.
spc700.bin	Driver SPC700 compilé + samples + données SFX chargées au démarrage.
music_N.bin	Données musicales d'un sous-morceau non marqué comme sound effect.
sounds.asm	Fichier d'intégration WLA-DX avec incbin du driver et des musiques.
sounds.h	Aliases C : noms des musiques, pointeurs musicData, constantes.

Dans ton test, sounds.asm référence spc700.bin et music\_1.bin, tandis que sounds.h définit MUS\_UNTITLED et un tableau musicData. Cela confirme que SNESGSS produit déjà une structure exploitable dans une ROM.

```
sounds.asm :
    spc700_code_1: .incbin "spc700.bin"
    music_1_data: .incbin "music_1.bin"

sounds.h :
    enum { MUS_UNTITLED = 0 };
    const unsigned char* const musicData[MUSIC_ALL] = { music_1_data };
```

### 3. Préparer les samples correctement

C'est la partie la moins confortable, mais c'est aussi celle qui explique la propreté de la conversion. SNESGSS attend que les samples soient déjà resamplés et loopés dans un éditeur externe : Audacity, Wavosaur, Ableton ou équivalent.

Paramètre	Recommandation
Format	WAV
Canaux	Mono
Résolution	16-bit
Fréquence	8000, 16000 ou 32000 Hz ; 32000 Hz est préférable
Sample mélodique	B +21 cents
Si le sample source est en C	baisser de 79 cents : C -> -79 cents
Boucle	à préparer avant import, sur une zone stable

#### Pourquoi B +21 cents ?

Le BRR SNES travaille par blocs de 16 samples. L'accordage B +21 cents aide certaines formes d'onde simples à tomber plus proprement sur des multiples de 16, donc à mieux looper. Mentalement c'est étrange, mais musicalement ça peut réduire les artefacts.

Avec ton Korg i5S, le plus simple est de sampler un C propre, puis de le pitcher à -79 cents dans Audacity ou Ableton. L'autre méthode est de sampler directement un B, puis de l'ajuster à +21 cents.

Méthode pratique Korg i5S :

1. Enregistrer un C tenu propre.
2. Passer en mono.
3. Convertir / exporter en 16-bit, 32000 Hz.
4. Appliquer Change Pitch = -0.79 demi-ton, donc -79 cents.
5. Créer la loop dans l'éditeur audio.
6. Importer le WAV dans SNESGSS.

## 4. Importer instruments .gsi et WAV

---

Il faut bien distinguer instrument et sample brut. Un .gsi est un instrument SNESGSS. Un .wav est une matière sonore à convertir/importer dans un instrument.

```
F3 = Instrument editor

.gsi  -> charger comme instrument
.wav  -> importer comme sample source dans un instrument
F1    -> revenir au Song editor
```

Point essentiel : être sur l'instrument 2 dans F3 ne suffit pas. Dans le Song Editor, chaque note doit porter son numéro d'instrument dans la colonne ii.

```
C-4 01 80 .... -> joue l'instrument 1
C-4 02 80 .... -> joue l'instrument 2
C-4 03 80 .... -> joue l'instrument 3
```

## 5. Comprendre le Song Editor

---

SNESGSS n'a pas la logique patterns + order list d'OpenMPT. Il fonctionne comme une longue timeline verticale avec des sections, une longueur de morceau et un point de boucle.

```
RRRR SS [CH1..CH8]
```

```
RRRR = numéro de ligne
```

```
SS = speed par ligne, de 1 à 99
```

```
1 = plus rapide, 99 = plus lent, 20 = valeur par défaut
```

```
CH1..CH8 = champs de notes des 8 canaux SNES
```

Le champ de note a cette forme :

```
C#3 ii VV E vv
```

```
C#3 = note
```

```
ii = instrument, de 1 à 99
```

```
VV = volume, de 0 à 99
```

```
E = effet
```

```
vv = valeur de l'effet, de 0 à 99
```

## 6. Notes, instruments, volumes et arrêt de note

---

La piste ne choisit pas l'instrument automatiquement. L'instrument est écrit note par note. C'est la colonne ii qui décide.

```
Exemple batterie / basse :  
CH1 : C-3 01 85 .... -> basse instrument 1  
CH7 : C-3 10 99 .... -> kick instrument 10  
CH8 : D-3 11 99 .... -> snare instrument 11
```

Pour arrêter une note loopée ou tenue, il faut insérer une rest note / keyoff dans la colonne note. Le readme indique : 1 ou A = rest note. Attention : sur certains claviers AZERTY, la touche physique peut être différente, et le pavé numérique 1 règle l'octave au lieu d'insérer une rest.

```
C-4 01 80 .... -> note tenue  
... ..  
1.. .. -> rest / keyoff / arrêt de note
```

### Piège clavier

Numpad 1..8 sert à choisir l'octave. Pour une rest note, il faut la touche 1 de la rangée haute ou A, selon le mapping. Sur AZERTY, tester aussi la touche physique équivalente au A QWERTY.

## 7. Rows infinies, sections, longueur et loop

---

Les lignes semblent infinies, mais ton morceau réel existe seulement entre son début, sa longueur et son point de boucle. Il faut déclarer explicitement la fin et le retour de loop.

```
Ctrl+Home = définir la position de boucle  
Ctrl+End  = définir la longueur du morceau
```

Exemple simple :

```
0000 INTRO  
0032 LOOP START  
0128 END  
  
Sur 0032 : Ctrl+Home  
Sur 0128 : Ctrl+End
```

Pour ne pas te perdre, utilise les sections.

```
Space = poser une séparation de section  
`     = nommer une section  
Home  = section précédente  
End   = section suivante  
Ctrl+F = liste des sections
```

## 8. Lecture, navigation et édition

---

### Tableau (1/3)

Action	Raccourci
Song editor	F1
Song list editor	F2
Instrument editor	F3
Info	F4
Play depuis le début	F5
Play depuis la ligne courante	F7
Stop	F8
Maintenir pour jouer depuis la position	Enter

### Tableau (2/3)

Action	Raccourci
Mute canal	Ctrl+1..8
Solo canal courant	Ctrl+0
Auto step	[ ]
Octave	Numpad / * ou Numpad 1..8
Sélection	Shift + curseur
Copier / coller	Ctrl+C / Ctrl+V
Coller par-dessus	Ctrl+B
Transpose demi-ton	Ctrl+F1 / Ctrl+F2

### Tableau (3/3)

Action	Raccourci
Transpose octave	Ctrl+F3 / Ctrl+F4
Expand / Shrink	Ctrl+E / Ctrl+S

#### Mode semi-interactif

SNESGSS ne réagit pas comme OpenMPT. Les changements ne s'entendent qu'après redémarrage de la lecture. Même les mutes peuvent nécessiter de relancer. Travailler souvent avec F7 ou Enter depuis la ligne courante.

## 9. Effets SNEGSS : codes et exemples

Tous les effets utilisent des valeurs décimales de 0 à 99, pas de l'hexadécimal. Les effets sont continus : ils restent actifs sur le canal jusqu'à ce qu'on les modifie ou qu'on les coupe avec 00. Exception : le pan n'est pas coupé de la même manière, il faut le replacer explicitement.

Code	Fonction	Exemple	Usage musical
Txx	(de)Tune	T03 / T00	Désaccord constant appliqué aux notes suivantes. T00 = retour neutre.
Uxx	Slide Up	U12	La hauteur monte progressivement. 1 lent, 99 rapide.
Dxx	Slide Down	D10	La hauteur descend progressivement.
Pxx	Portamento	P20	Glisse vers la nouvelle note. Prioritaire sur U/D.
P99	Legato	P99	Changement instantané de hauteur sans nouveau key-on.
Sxx	Pan	S00 / S50 / S99	0 gauche, 50 centre, 99 droite.
Vxy	Vibrato	V32	x = vitesse 1..9, y = profondeur 1..9.
R..	Repeat	R..	Répète la section précédente dans le canal courant.

### Exemples rapides

```
Vibrato doux de lead :  
C-4 01 80 V21
```

```
Vibrato plus chantant :  
C-4 01 80 V32
```

```
Arrêt du vibrato :  
... .. V00
```

```
Pan :  
C-4 01 80 S00 -> gauche  
C-4 01 80 S50 -> centre  
C-4 01 80 S99 -> droite
```

```
Portamento :  
0000 C-4 01 80 P20  
0010 G-4 01 80 .... -> glisse vers G avec la vitesse P20
```

```
Legato :  
0000 C-4 01 80 P99  
0010 G-4 01 80 .... -> change la hauteur sans relancer l'attaque
```

## 10. Pan, pseudo-tremolo, detune, legato et astuces

---

SNESGSS ne liste pas d'effet tremolo automatique ni d'effet echo dans le readme. Il faut donc simuler certains gestes par écriture.

### Pseudo-tremolo manuel

Le tremolo est une variation de volume. Comme il n'y a pas d'effet dédié, on alterne la colonne volume.

```
C-4 01 80 ....
... .. 70 ....
... .. 80 ....
... .. 70 ....
... .. 80 ....
```

### Pseudo-delay / largeur

Sans echo programmable, tu peux doubler une note sur deux canaux avec retard, volume plus bas et pan opposé. C'est coûteux en canaux, mais efficace sur des notes importantes.

```
CH1 : C-4 01 80 S50
CH2 : .....
CH2 : C-4 01 38 S20
CH3 : .....
CH3 : C-4 01 32 S80
```

### Volume slide

Le readme n'indique pas de volume slide comme effet. La version 1.2 a ajouté une colonne volume et remplacé l'ancien Vxx. En pratique, un fade se programme à la main avec la colonne volume.

```
Fade-out manuel :
C-4 01 90 ....
... .. 78 ....
... .. 64 ....
... .. 50 ....
... .. 36 ....
... .. 20 ....
1.. .. .. ....
```

### Echo / reverb

Pas de bloc echo type SNESMOD dans le readme SNESGSS fourni. Si tu veux une vraie grande profondeur SNES programmable, SNESMOD garde un avantage. Si tu veux zéro clic et une gravure SPC propre, SNESGSS marque des points.

## 11. Export SPC et export pour intégration SNES

---

Il y a deux exports différents à ne pas confondre.

Objectif	Chemin / résultat
Écouter / archiver une musique	Exporter un SPC. Dans ta version, l'export SPC est dans le menu Song, pas File.
Intégrer dans une ROM / homebrew	File > Export ou <code>snesgss.exe fichier.gsm -e dossier_export</code>
Sortie programme	<code>spc700.bin</code> , <code>music_N.bin</code> , <code>sounds.asm</code> , <code>sounds.h</code>

```
Commande export intégration :  
snesgss.exe filename.gsm -e [optional export path]
```

### Avant export SPC

Toujours définir Ctrl+Home pour la loop et Ctrl+End pour la longueur réelle. Sinon le SPC peut avoir une structure absurde ou une boucle mal placée.

Le driver exporté démarre en mono dans une intégration de jeu, pour compatibilité avec certains anciens téléviseurs. L'export SPC ou l'éditeur peuvent sonner stéréo, mais en ROM il faut envoyer la commande d'activation stéréo si tu l'intègres vraiment.

## 12. Workflow recommandé DAC MASTER

---

Le bon usage de SNEGSSS n'est pas forcément d'abandonner OpenMPT, mais de séparer composition et gravure.

Étape	Action
1. Idée	Composer ou esquisser dans OpenMPT si la pièce est complexe.
2. Banque	Choisir peu de samples, très propres, adaptés à la SNES.
3. Préparation	Mono, 16-bit, 32000 Hz, C -> -79 cents pour les instruments mélodiques.
4. SNEGSSS	Reconstruire la version finale en timeline SNEGSSS.
5. Contrôle	Tester souvent en SPC, pas seulement dans l'éditeur.
6. Décision	Si le SPC est propre et musical : valider. Si la composition est trop lourde : simplifier ou revenir à OpenMPT.

### Règle actuelle

OpenMPT reste meilleur pour la pensée compositionnelle. SNEGSSS devient prioritaire dès que la propreté BRR/SPC est plus importante que le confort d'édition.

## 13. Anti-pièges

---

Problème	Réponse pratique
Je change de piste mais j'entends toujours l'instrument 1.	Normal : la piste ne choisit pas l'instrument. Écris 02 ou 03 dans la colonne instrument ii.
Les rows vont à l'infini.	Normal : pose Ctrl+End pour la longueur et Ctrl+Home pour le point de loop.
J'appuie sur 1 mais ça ne fait pas de rest.	Tu utilises peut-être le pavé numérique. Le pavé numérique sert aux octaves. Essaie la rangée haute, A ou l'équivalent AZERTY.
Je change un paramètre et je n'entends rien.	SNESGSS est semi-interactif. Relance avec F7 ou Enter depuis la ligne courante.
Mon sample mélodique clique.	Prépare la loop avant import, évite l'attaque, travaille en blocs cohérents, et teste C -> -79 cents.
Je veux un echo SNES profond.	SNESGSS ne documente pas d'echo mod dans le readme. Simule avec canaux ou utilise SNESMOD si l'echo est central.
Je veux un volume slide.	Pas d'effet listé. Écris une pente de volumes dans la colonne VV.

## 14. Fiche rapide

---

```
F1 Song editor
F2 Song list
F3 Instrument editor
F4 Info
F5 Play from start
F7 Play from current row
F8 Stop
Enter Play from current row while held
```

Note field : C#3 ii VV E vv

```
ii instrument 1..99
VV volume 0..99
E effect
vv effect value 0..99 decimal
```

Effects :

```
Txx detune
Uxx slide up
Dxx slide down
Pxx portamento
P99 legato
Sxx pan : 0 left, 50 center, 99 right
Vxy vibrato : x speed, y depth
R.. repeat previous section
```

Song structure :

```
Ctrl+Home set loop position
Ctrl+End set song length
Space section break
` section name
Ctrl+F section list
```

Sample prep :

```
WAV mono 16-bit
8000 / 16000 / 32000 Hz, preferably 32000 Hz
Melodic sample : B +21 cents
If sampled in C : pitch down -79 cents
```

## 15. Sources et notes de méthode

---

Ce guide est construit à partir du readme SNESGSS v1.42 fourni en conversation, de tes fichiers d'export sounds.asm / sounds.h, et de nos tests pratiques : export SPC dans le menu Song, conversion sans clic, intérêt de SNESGSS comme atelier de gravure SPC propre.

Il ne remplace pas la documentation complète du moteur pour une intégration SNES avancée. Il sert de fiche de travail rapide pour composer, préparer les samples, utiliser les effets et sortir des SPC propres.