

[illegible]

Les caractères que vous ne souhaitez pas définir devront être remplacé par 8 octets à 0, voir exemple ci-dessus.

## La Tilesmap, fichier .TLM :

Ce fichier va vous permettre de définir votre map de jeu. La map que vous définirez ne doit pas excéder plus de 65535 Tiles, c'est-à-dire que la largeur de la map X la hauteur de la map doit être inférieur à 65535.

Le fichier va être décomposé en trois parties, l'entête, les données et les propriétés des tiles.

Dans l'entête nous allons avoir les informations suivantes :

- Octets 0 et 1 -> définira le nombre total de tile en largeur de la map codé sur 16 bits
- Octets 2 et 3 -> définira le nombre total de tile en hauteur de la map codé sur 16 bits
- Octet 4 -> définira la largeur en pixel d'un tile, les valeurs possibles sont 8 ou 16
- Octet 5 -> définira la hauteur en pixel d'un tile, la valeur maximum possible est 16
- Octet 6 et 7 -> nombre de tiles différents utilisés dans la map codé sur 16 bits. Si la map utilise 100 fois le même tile, la valeur ici sera alors 1.
- Octet 8 et 9 -> réservé.

Nombre de tiles en largeur de la map 0x0028

Nombre de tiles en hauteur de la map 0x001C

Largeur en pixel d'un tile 0x08

Hauteur en pixel d'un tile 0x07

Nombre de tiles différents dans la map 0X0026

Deux octets non utilisés 0XFFFF

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 00 28 00 1C 08 07 00 26 FF FF 00 00 00 00 00 00  (.....&yy.....
00000010 01 03 03 03 03 03 03 03 03 02 00 00 00 04 00 07  .....
```

La map sera codée ligne par ligne en se déplaçant de la gauche vers la droite. Dans l'exemple ci-dessus une ligne aura 0x0028 (40) valeurs et nous auront au total 0x001C (28) lignes, ce qui représente 2560 valeurs de tiles dans la partie donnée.

Le codage des données sera différent si l'on utilise des tiles de 8 pixels de large ou de 16 pixels de large comme défini avec l'octet 4 de l'entête.

- Pour des tiles de 8 pixels de large, les numéros des tiles du fichier Tilesmap seront codés sur 1 octet, ce qui autorisera 255 tiles différents dans la map.
- Pour des tiles de 16 pixels de large, les numéros des tiles du fichier Tilesmap seront codés sur 2 octets, ce qui autorisera 65535 tiles différents dans la map.

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 00 28 00 1C 08 07 00 26 FF FF 00 00 00 00 00 00  (.....&yy.....
00000010 01 03 03 03 03 03 03 03 03 02 00 00 00 04 00 07  .....
```

Chaque encadré ci-dessus représente un numéro de tile sur 1 octet (donc tiles de largeur de 8 pixels), on peut voir que le début de la ligne 1 commence par 6 tiles numéro 0, suivi d'un tile numéro 1, suivi de 8 tiles numéro 3 et ainsi de suite.

*Le numéro de tile 0 signifie pas de tile à afficher, c'est-à-dire qu'à l'emplacement du tile 0, on verra l'arrière-plan.*

A noter que si les tiles avaient été d'une largeur de 16 pixels en encadré bleu regrouperait deux octets pour définir le numéro du tile.

Les propriétés au nombre de trois, vous permettront de définir des caractéristiques pour chaque tile. A noter que le tile de numéro 0 n'a pas de propriétés associées. Par exemple si les tiles ayant le numéro 18, 34, 45, 46, 47 correspondent à des obstacles, vous pourrez définir une valeur dans la propriété 1 indiquant que ces tiles sont des obstacles au lieu de vérifier leurs numéros dans votre programme, ce qui permet de simplifier la programmation et de gagner du temps processeur. Vous êtes libre de coder ce que vous désirez dans ces 3 octets.

La partie propriété du fichier Tilesmap est codé sur 8 octets par tiles différents, soit si votre map compte 30 tiles différents, la section propriété comptera  $30 \times 8 = 240$  octets.

Codage des propriétés pour 1 tile :

- 4 octets (1 mot long) -> adresse du tile dans le tileset. Mettre la valeur FFFFFFFF dans le fichier, celle-ci sera remplacée par l'adresse du tile concerné en mémoire suite au chargement du fichier des tileset.
- 1 octet -> Propriété 1 du tile
- 1 octet -> Propriété 2 du tile
- 1 octet -> Propriété 3 du tile
- 1 octet -> réservé / transparence

```

00000450 21 20 00 00 00 04 00 07 00 00 00 1F 21 21 21 21 ! .....!!!!
00000460 25 26 21 21 21 21 20 00 00 00 FF FF FF FF 02 FF %&!!!! ...YYYY.Y
00000470 FF FF FF FF FF FF 02 FF FF FF FF FF FF 02 FF YYYYYY.YYYYYYYY.Y
00000480 FF FF FF FF FF FF 02 FF FF FF FF FF FF 02 FF YYYYYY.YYYYYYYY.Y
00000490 FF FF FF FF FF FF 02 FF FF FF FF FF FF 02 FF YYYYYY.YYYYYYYY.Y
000004A0 FF FF FF FF FF FF 02 FF FF FF FF FF FF 02 FF YYYYYY.YYYYYYYY.Y
000004B0 FF FF FF FF FF FF 02 FF FF FF FF FF FF 02 FF YYYYYY.YYYYYYYY.Y
000004C0 FF FF FF FF FF FF 02 FF FF FF FF FF FF 02 FF YYYYYY.YYYYYYYY.Y
000004D0 FF FF FF FF FF FF 02 FF FF FF FF FF FF 02 FF YYYYYY.YYYYYYYY.Y

```

L'exemple ci-dessus représente dans le premier encadré vert (à la suite de la fin des données de la map) les 4 octets 0xFFFF qui sont réservés pour que le programme y inscrive l'adresse mémoire des données du tiles numéro 1.

- Le premier encadré bleu 0x02 représente la valeur de la propriété 1 du tile 1.
- Le premier encadré rouge représente la propriété 2 du tile 1 (ici 0xFF car non utilisé)
- Le premier encadré marron représente la propriété 3 du tile 1 (ici 0xFF car non utilisé)
- Le premier encadré noir représente l'octet réservé/transparence du tile 1 non utilisé dans cette version.

Les encadrés suivants représentent les mêmes informations pour le tile numéro 2 et ainsi de suite pour les tiles suivant.

## Les tiles, fichier .TLS

Le fichier TLS va contenir les données hexadécimales des tiles, sur le même principe que le fichier CRS mais sur 4 grilles (4 plans image en basse résolution) de largeur de 16 pixels et de hauteur la hauteur définie par l'octet 5 de l'entête du fichier .TLM. Quatre grilles d'une largeur de 16 pixels correspondent à 8 octets de large en données qu'il faudra multiplier par la hauteur d'un tile pour avoir le nombre d'octets utilisés pour un tile.

Exemple : tile de 12 lignes de hauteur

- Nombre d'octet par tile =  $8 \times 12 = 96$  octets.

Les données des tiles seront mises les unes à la suite des autres en entrelaçant les plans et en commençant par le tile 0 qui aura toutes ces données à 0 (le tile 0 n'est jamais affiché). Donc si on reprend l'exemple ci-dessus, le fichier va commencer avec 96 octets à 0, suivi des 96 octets du tile 1, suivi des 96 octets du tiles 2 et ainsi de suite.

- Tile 0 -> 8 x hauteur Octets à 0x00
- Tile 1 -> 8 x hauteur Octets avec entrelacement des 4 plans (codage de l'écran Atari ST)
- Tile 2 -> 8 x hauteur Octets avec entrelacement des 4 plans (codage de l'écran Atari ST)
- .....

Pour des tiles de largeurs 8 pixels, ceux-ci seront dessiner à gauche des grilles de largeur 16 pixels également et utiliseront aussi 8 octets de large.

Pour le principe de codage des données sur 4 plans images, voir Atari Compendium.

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000040	00	00	30	00	00	00	30	00	00	00	40	00	00	00	40	00	..0...0...@...@.
00000050	00	00	47	00	00	00	47	00	00	00	88	00	00	00	88	00	..G...G...^...^.
00000060	00	00	90	00	00	00	90	00	00	00	90	00	00	00	90	00	.....
00000070	00	00	F0	00	00	00	F0	00	00	00	0C	00	00	00	0C	00	..š...š.....
00000080	00	00	02	00	00	00	02	00	00	00	E1	00	00	00	E1	00	.....ă...ă.
00000090	00	00	11	00	00	00	11	00	00	00	09	00	00	00	09	00	.....
000000A0	00	00	09	00	00	00	09	00	00	00	FF	00	00	00	FF	00	.....ÿ...ÿ.
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Dans cet exemple on a codé des tiles de 8 pixels de large et de 7 lignes en hauteur. L'encadré orange représente les 96 octets du tile N°0, ces octets sont tous à la valeur 0x00.



Les octets qui suivent ont été regroupés deux à deux dans des encadrés de couleurs différents. Les encadrés rouges représentent les plans 1 du tile, les verts les plans 2, les jaunes les plans 3 et enfin les bleus les plans 4. Quatre plans pour dessiner une ligne d'un tile, qui nécessite 8 octets pour la définir.

Pour avoir les 7 lignes du tile N°1, on peut voir qu'il y a 7 groupes de quatre plans.

Dans cet exemple, comme les tiles sont de 8 pixels de large, vous pouvez voir que tous les octets situés à droite de chaque plan sont à zéro, car comme déjà mentionné précédemment, les tiles sont dessinés sur la partie de gauche (sur le premier octet de chaque plan).

Si nous avions dessiné des tiles de 16 pixels de large, nous aurions utilisé les 2 octets des plans.

## L'audio DMA, fichier .ADS

Le fichier est décomposé en deux parties, une partie entête et une partie donnée.

La partie entête est composée des données suivantes :

- Octet 0 à 3 -> nombre d'octets de données audio (codé sur 4 octets)
- Octet 4 -> indique si l'audio est en Mono (0x80) ou en stéréo (0x00)
- Octet 5 -> Indique la Fréquence d'échantillonnage (0x00=6258Hz, 0x01=12517Hz, 0x02=25033Hz, 0x03=50066Hz)

La partie donnée audio contient les valeurs des échantillons audio qui devront être organisés comme ci-dessous :

- Si audio mono -> octet 1, octet 2, octet 3, .....
- Si audio stéréo -> octet 1L, octet 1R, octet 2L, octet 2R, octet 3L, octet 3R, ....

Les échantillons devront avoir des valeurs signées sur 8 bits comprises entre -128 et 127.

Nombre d'octets audio  
Audio Monophonique  
Echantillonnage à 12517 Hz

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	00	00	80	B0	80	01	FF	00	FF	00	FF	F8	EF	EB	F8	0B	..e°e.y.y.yøiëø.
00000010	0E	06	04	07	02	F8	F6	FF	06	FF	F4	F4	F5	F1	EA	F3	....øøÿ.yøøøñëó
00000020	09	10	08	05	06	05	FA	F4	FC	05	01	F7	F3	F6	F4	EB	.....úôü..÷óôøë
00000030	F1	05	10	09	03	07	05	FC	F4	F9	05	04	F9	F3	F5	F6	ñ.....úôü..úóôø
00000040	ED	ED	FE	0E	0B	04	05	07	FF	F6	F8	02	05	FB	F3	F5	íip.....ÿøø..úóô

Les octets qui suivent l'encadré marron correspondent aux valeurs d'échantillons audio.

## Les effets audio, fichier .AEF

Ce fichier permet de fournir des données au processeur audio Yamaha de l'Atari ST. Les données seront fournies durant des cycles d'interruptions générés par le timer D qui devra être configuré avec la fonction **STX\_setsoundtimer**.

Pour comprendre le fonctionnement de celui-ci reporté vous à la documentation du Yamaha 2149.

Les données de ce fichier sont organisées comme ci-dessous :

- 1 octet -> nombre de données N à lire durant l'interruption 1
- N octets -> données des différents registres avec les valeurs associées
- 1 octet -> nombre de données N à lire durant l'interruption 2
- N octets -> données des différents registres avec les valeurs associées
- 1 octet -> nombre de données à lire durant l'interruption 3
- .....
- 1 octet -> valeur de fin de fichier 0xFF

Si l'octet nombre de données N à lire durant l'interruption X est égale à 0x00, alors l'interruption sera sautée et le signal sonore sera maintenu jusqu'à la prochaine interruption.

Les données des différents registres seront codées en précisant le numéro du Canal A, B ou C suivi du numéro du registre du Yamaha 2149 concerné.

Exemple : Valeur 0x23 à appliquer au registre 1 du canal A

- 0xA1 0x23

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	06	B9	0A	B3	02	B2	7D	00	00	00	00	04	B3	01	B2	86	[.]^.^.^}.....^.*†
00000010	00	00	02	B2	15	00	04	B3	00	B2	E5	00	02	B2	C9	00	...^...^.*â...^É.
00000020	02	B2	B5	02	B2	AC	02	B2	A7	02	B2	99	02	B2	8A	00	.*u.*~.*\$.*m.*Š.
00000030	02	B2	7D	00	00	02	B2	71	02	B2	6B	02	B2	66	00	00	.*}...^q.*k.*f..
00000040	02	B2	5D	02	B2	57	00	00	02	B9	00	FF					.*].*W...^.*ÿ

Vous pouvez voir dans l'exemple ci-dessus, en rouges les octets correspondants aux nombres de données qui seront lu à chaque interruption, ceux-ci sont suivis par N octets de commandes destinés au Yamaha 2149.

En orange, on peut voir les correspondants aux nombres de données qui sont à 0, ce qui indique de ne rien changer, les données appliquées précédemment au Yamaha seront maintenues.

En fin de fichier, se trouve un octet avec la valeur 0xFF pour indiquer la fin des données.

## Les Sprites, fichier .SPT

Ces fichiers sont générés à partir du Menu Exporter de l'éditeur de Sprite SPRITEV2.PRG à utiliser sur votre émulateur ou Atari ST.

