Sound

COLLABORATORS			
	TITLE : Sound		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		January 19, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Sound

1.1 Sound V1.00

```
PureBasic - Sound V1.00

  Cette bibliothèque est destinée à gérer les sons sous le
  PureBasic. Et croyez-nous, on a passé un temps fou à optimiser
  toutes les fonctions disponibles. Il en résulte une gestion
  des sons qui respecte le système, ultra rapide et qui utilise
  les 4 canaux audios de l'Amiga. Le format des sons utilisé
  est l'IFF/8SVX.

Commands summary:

   ChangeSoundPeriod
   ChangeSoundVolume
   CreateSound
   DecodeSound
   FreeSound
   GetSoundLength
   InitSound
   LoadSound
   PeekSoundData
   PlaySound
   PokeSoundData
   SaveSound
   SetSoundChannels
   SetSoundPeriod
   SetSoundVolume
   SoundFilter
   StopSound
   UseSoundChannels

   Sound Demo
```

1.2 changesoundperiod

```
  Syntaxe
```

```
Résultat.w = ChangeSoundPeriod(#Sound.w,Period.w)

  Résumé
Change en temps réel la période du son spécifié (qui doit être
en train d'être joué). Cette fonction renvoie le masque des
canaux audios affectés par ce changement de période.

La période originale du son (définie par  SetSoundPeriod() )
n'est pas affectée par cette fonction.


#Sound: Identifiant numérique du son

Period: Nouvelle période du son
```

1.3 changesoundvolume

```
  Syntaxe
Résultat.w = ChangeSoundVolume(#Sound.w,Volume.w)

  Résumé
Change en temps réel le volume du son spécifié (qui doit être
en train d'être joué). Cette fonction renvoie le masque des
canaux audios affectés par ce changement de période.

Le volume original du son (défini par  SetSoundVolume() )
n'est pas affecté par cette fonction.

#Sound: Identifiant numérique du son

Volume: Nouvelle valeur du volume
```

1.4 createsound

```
  Syntaxe
Résultat.l = CreateSound(#Sound.w,Length.l)

  Résumé
Crée un nouveau son entièrement vide. La période, le volume et
les canaux sont initialisés à 0. Pour écrire dans la partie
'Data' du son et ainsi pouvoir créer réellement le son, vous
pouvez utiliser la commande  PokeSoundData() . Si le résultat
de cette fonction est NULL, alors le son n'a pas pu être créé.


#Sound: Identifiant numérique de l'objet à créer.

Length: Longueur du son (longueur des 'Data')
```

1.5 decodesound

```
  Syntaxe
Résultat.w = DecodeSound(#Sound.w,Pointer.l)

  Résumé
Initialise un son à partir d'un fichier IFF/8SVX inclu
dans le programme via la commande 'IncludeBinary'.
Si le Résultat est NULL, alors le son n'a pas pu être
initialisé correctement.

* Le volume de ce son est mis à 64.
* Les canaux sont mis à 15.


#Sound: Identifiant numerique du son

Pointer: Pointeur vers le son IFF (principalement '?label').
```

1.6 freesound

```
  Syntaxe
FreeSound(#Sound.w)

  Résumé
Libère un son de la mémoire et détruit toutes les informations
le concernant.


#Sound: Indentifiant numérique du son
```

1.7 getsoundlength

```
  Syntaxe
Résultat.l = GetSoundLength(#Sound.w)

  Résumé
Retourne la longueur du son spécifié. Utile pour éviter
d'écrire en dehors de la zone de 'Data' lorsqu'on
utilise la commande  PokeSoundData() .


#Sound: Identifiant numérique du son.
```

1.8 initsound

```
  Syntaxe
Résultat.w = InitSound(Objects.l)

  Résumé
Initialise l'environnement nécessaire à la gestion des sons.
```

```
Vous devez appeler cette fonction avant d'appeler une autre
fonction de cette bibliothèque. Si le 'Résultat' est NULL,
alors il n'y a plus assez de mémoire libre.


Objects: Nombre maximal de sons à gérer.
```

1.9 loadsound

```
  Syntaxe
Résultat.b = LoadSound(#Sound.w,FileName$)

  Résumé
Charge le son IFF/8SVX spécifié en mémoire. Le son est
prêt à être joué. Si le résultat est NULL, alors le son
n'a pas pu être chargé.


* Le volume du son est mis à 64.
* Les canaux audios sont initialisés à 15.


#Sound: Identifiant numérique su nouveau son.

FileName: Chemin et nom du fichier son IFF/8SVX
```

1.10 peeksounddata

```
  Syntaxe
Résultat.b = PeekSoundData(#Sound.w,Position.l)

  Résumé
Retourne la valeur du 'Data' du son spécifié à une position donnée.
Cette valeur est comprise entre -128 et +127


#Sound: Identifiant numérique du son.

Position: Position à l'intérieur du 'SoundData'
```

1.11 playsound

```
  Syntaxe
Résultat.w = PlaySound(#Sound.w,Repeat.w)

  Résumé
Joue le son spécifé en tenant compte de sa période, de son
volume et de ses canaux. Cette fonction renvoie le masque des
canaux audios affectés par ce changement de période. Si
la valeur retournée est NULL alors le son ne peut pas
```

être joué.

#Sound: Identifiant numérique du son à jouer.

Repeat: Nombre de répétions pour ce son. Si vous voulez
qu'il soit répété à l'infini, utiliser une valeur
négative.

1.12 pokesounddata

Syntaxe
PokeSoundData(#Sound.w,Position.l,Data.b)

Résumé
Ecrit une valeur dans le 'SoundData' à la position donnée

#Sound: Identifiant numérique du son

Position: Position à laquelle la valeur doit être écrite.

Data: Valeur à écrire (comprise entre -128 et +127).

1.13 savesound

Syntaxe
Résultat.b = SaveSound(#Sound.w,FileName$)

Résumé
Sauvegarde le son spécifié à l'endroit spécifié par 'Filename'. Si
le 'Résultat' est NULL, alors le son n'a pas pu être sauvegardé
à cet endroit.

#Sound: Identifiant numérique du son à sauvegarder.

FileName: Chemin et nom du fichier où le son sera sauvegardé.

1.14 setsoundchannels

Syntaxe
SetSoundChannels(#Sound.w,Channels.w)

Résumé
Change l'affection des canaux pour le son spécifié. Chaque
son a son propre paramétrage des canaux sur lesquels il sera
joué (par la commande PlaySound()).

```
#Sound: Identifiant numérique du son

Channels: Masque representant un canal ou une combinaison
          de canaux:

Valeur du masque:

1 = seulement le canal 0
2 = seulement le canal 1
4 = seulement le canal 2
8 = seulement le canal 3

Exemples de combinaisons:

 5 = canal 0 (valeur 1) et canal 2 (valeur 4)
15 = tous les canaux (1+2+4+8)
```

1.15 setsoundperiod

```
  Syntaxe
SetSoundPeriod(#Sound.w,Period.w)

  Résumé
Change la période interne du son qui sera utilisée
par la commande  PlaySound() .


#Sound: Identifiant numérique du son.

Period: Nouvelle période du son.
```

1.16 setsoundvolume

```
  Syntaxe
SetSoundVolume(#Sound.w,Volume.w)

  Résumé
Change le volume interne du son spécifié qui sera utilisé
par la commande  PlaySound() .


#Sound: Identifiant numérique du son.

Volume: Nouveau volume.
```

1.17 soundfilter

```
  Syntaxe
SoundFilter(State)
```

Résumé
Active ou desactive le filtre passe-bas interne de l'Amiga
(coupe les hautes fréquences).

State: 1 = Filtre activé, 0 = Filtre désactivé.

1.18 stopsound

Syntaxe
StopSound(#Sound.w)

Résumé
Arrête le son spécifié en train d'être joué.

#Sound: Identifiant numérique du son à stopper.

1.19 usesoundchannels

Syntaxe
Résultat.w = UseSoundChannels(Channels.w)

Résumé
Alloue ou libère les canaux sonores voulus. Si vous avez alloué
tous les canaux (Channel = 15) et que vous voulez en libérer
2, par exemple les canaux 2 et 3 vous pouvez appeler cette
fonction avec 'Channels = 3' (canaux 1 et 2). Ainsi les canaux
0 et 1 restent alloués tandis que les canaux 2 et 3 sont
libérés. Si le 'Résultat' est FALSE, alors les canaux n'ont
pas pus être alloués correctement.

Channels: Masque représentant les canaux à allouer.