

Jukebox

Scripts de GuardianForce

Marcoest

10.08.2007

Voici un script pour écouter de la musique dans le jeu.



Aperçu du script

Script testé, il fonctionne.

Créez un nouveau script, nommé Window_JBliste, et collez-y le code ci-dessous.

```
#=====
# ■ Window_JBliste
#-----
# Script permettant d'afficher la liste des musiques du jukebox
# Créateur : Tonyryu
# Date : 12/10/2006
# Version : 1.00
# Version : 1.01 : 06/08/2007 : Tonyryu : Correction méthode fichierMusique
#
# Attention : Ce script est ma propriété en tant que création et il est donc
# soumis au droit de la propriété intellectuelle ( http://www.irpi.ccip.fr/ ).
# En aucun cas, il ne doit être copié ou publié vers un autre forum sans en
# avoir reçu mon accord au préalable.
#
#=====
class Window_JBliste < Window_Selectable

#-----
# • initialize
#-----

def initialize
```

```
super(0, 100, 640, 380)
```

```
# Tableau contenant la liste des musique [ "nom", type, "fichier sans extension"]
```

```
# type : 1 : BGM
```

```
# 2 : BGS
```

```
# 3 : ME
```

```
# 4 : SE
```

```
@tabMusiques = [[ "Introduction", 1, "034-Heaven01"],
```

```
    [ "Theme Alex", 1, "014-Theme03"],
```

```
    [ "Combat 1", 1, "002-Battle02"],
```

```
    [ "Bruit de l'eau", 2, "010-River01"] ]
```

```
# Spécifier la fenêtre de sélection
```

```
@column_max = 2
```

```
# Récupérer le nombre de musique
```

```
@item_max = @tabMusiques.size
```

```
# Initialiser la Zone Bitmap avec une taille adapté au nombre de musique
```

```
self.contents = Bitmap.new(width - 32, row_max * 32)
```

```
# Définir la police d'écriture par défaut
```

```
self.contents.font.name = $fontface
```

```
self.contents.font.size = $fontsize
```

```
self.index = 0
```

```
refresh
```

```
end
```

```
#-----
```

```
# ● refresh
```

```
#-----
```

```
def refresh
```

```
  # Effacer la zone Bitmap
```

```
  self.contents.clear
```

```
  # Pour toutes les musiques
```

```
  for i in 0..@item_max - 1
```

```
    x = 4 + i % 2 * (288 + 32)
```

```
    y = i / 2 * 32
```

```
    # Afficher les musique en 2 colonnes
```

```

self.contents.draw_text(x, y, 212, 32, @tabMusiques[i][0], 0)

end

end

#-----

# • fichierMusique

#-----

def fichierMusique

return [@tabMusiques[@index][1], @tabMusiques[@index][2]]

end

end

end

```

```

#=====

# ■ Window_JBtitre

#-----

# Script permettant d'afficher le titre jukebox

# Créateur : Tonyryu

# Date : 12/10/2006

# Version : 1.00

#=====

class Window_JBtitre < Window_Base

#-----

# • initialize

#-----

def initialize

super(0, 0, 640, 100)

# Initialiser la Zone Bitmap

self.contents = Bitmap.new(width - 32, height - 32)

# Définir la police d'écriture

self.contents.font.name = $fontface

self.contents.font.size = 30

self.contents.font.italic = true

refresh

end

```

#-----

• refresh

#-----

def refresh

 # Effacer la zone Bitmap

 self.contents.clear

 # Ecrire le titre

 self.contents.draw_text(0, 10, 608, 50, "Jukebox", 1)

end

end

#=====

■ Scene_Jukebox

#-----

Script permettant de gérer le jukebox

Créateur : Tonyryu

Date : 12/10/2006

Version : 1.00

#=====

class Scene_Jukebox

 def main

 # Préparation des fenêtre

 @jbtitre_window = Window_JBtitre.new

 @jbliste_window = Window_JBliste.new

 # Boucle de transition, mise à jour graphique et entré clavier

 Graphics.transition

 # boucle

 loop do

 # Mise à jour graphique

 Graphics.update

 # Mise à jour des entrées clavier

 Input.update

 # Mise à jour de la class

 update

 # Si la scene n'existe plus, alors sortir de la boucle

```

if $scene != self
    break
end

end

# figer les graphiques
Graphics.freeze

# destruction des objets
@jbtitre_window.dispose

@jbliste_window.dispose

end

#-----

# • update

#-----

def update

# Mettre à jour les fenêtres de groupe et de réserve

@jbtitre_window.update

@jbliste_window.update

# si la touche B est appuyée

if Input.trigger?(Input::B)

# Alors, jouer le son "cancel"

$game_system.se_play($data_system.cancel_se)

# Revenir soit dans le menu (option 5, changeable), soit sur la map, soit sur le titre

$scene = Scene_Menu.new(5)

#$scene = Scene_Map.new

#$scene = Scene_Title.new

# Stopper tous les sons

stopper_audio

return

end

# si la touche C est appuyée

if Input.trigger?(Input::C)

# Stopper tous les sons

stopper_audio

```

```
# Récupérer le nom du fichier correspondant au choix effectué
```

```
tabFichierMusique = @jbliste_window.fichierMusique
```

```
type = tabFichierMusique[0]
```

```
nomFichier = tabFichierMusique[1]
```

```
case type
```

```
when 1
```

```
    Audio.bgm_play("Audio/BGM/" + nomFichier,100,100)
```

```
when 2
```

```
    Audio.bgs_play("Audio/BGS/" + nomFichier,100,100 )
```

```
when 3
```

```
    Audio.me_play("Audio/ME/" + nomFichier,100,100)
```

```
when 4
```

```
    Audio.se_play("Audio/SE/" + nomFichier,100,100)
```

```
end
```

```
end
```

```
end
```

```
#-----
```

```
# • stopper_audio
```

```
#-----
```

```
def stopper_audio
```

```
    # Stoper tous les sons
```

```
    Audio.bgm_stop
```

```
    Audio.bgs_stop
```

```
    Audio.me_stop
```

```
    Audio.se_stop
```

```
end
```

```
end
```

Pour appeler le script, créez un événement. Dans les commandes d'événement faites "Insérer un script", puis mettez :

```
$scene = Scene_Jukebox.new
```

Et c'est tout ! 😊