



**Forum: Moteur de jeu GameBlender et alternatives**

**Topic: Sauvegarder des propriétés : modifier un script python via le GE**

**Subject: Re: Sauvegarder des propriétés : modifier un script python via le GE**

Posté par: Yunshi

Contribution le : 9/7/2011 20:06:00

Merci beaucoup pour ta réponse claire et rapide

Toutefois, elle ne répond pas exactement à mes attentes

Je vais essayer d'être le plus explicite possible (au temps pour moi si j'ai manqué de rigueur et de clarté dans mes propos) :

Je crée un jeu de plateaux par exemple. Je crée un fichier python, dans lequel j'écris clairement une variable à laquelle j'attribue la valeur "0" (on a donc nombre\_de\_plateaux\_finis = 0). Maintenant j'aimerais, par exemple, que si je finis le plateau Numéro 1 (j'en sais rien moi, avec une simple Collision avec un tel ou tel objet finish;), cette valeur s'incrémente de 1. On aura donc nombre\_de\_plateaux\_finis = 1. Le joueur n'aurait pas à sauvegarder sa progression, elle serait donc effectuée à chaque fois qu'il remplit un plateau. Et le but serait que si je relance le jeu (je manipule les actuators pour avoir une relecture à chaque fois du fichier python), ce dernier se souvienne, via la variable, quel plateau j'ai su atteindre. En ouvrant le fichier, Blender comprendrait donc explicitement : le joueur a déjà fini le plateau Numéro 1, il peut commencer directement au 2.

Je maîtrise les briques logiques de Blender, faire un perso et des plateaux, je peux m'en sortir allègrement. Ce qu'il me manque, c'est une sorte de fichier "mémoire" dans lequel une variable indique la progression dans le jeu, et qui est lu à chaque lancement du jeu. Et c'est là que, clairement, je ne connais pas la manipulation à établir.

Ce qui m'intéresse donc, c'est ce bout de code que tu m'as gentiment proposé :

```
# Systeme de sauvegarde/chargement:
nomFichier = "nom_fichier.extension"

# Est ce que le sensor sav est activé ?
if sav.positive == 1:

    # Dans ce cas, ouvrir le fichier en mode ecriture seulement ("w" = write)
    fichier = open(nomFichier, "w")
    # J'écris la position:
    fichier.write(pos)
    # Et je ferme le fichier:
    fichier.close()

if load.positive == 1:
```

```
# Là, j'ouvre le fichier en mode lecture seulement ("r" = read)
fichier = open(nomFichier, "r")
# Je lit la position:
posF = fichier.read()
# Je ferme le fichier, j'ai eu ce que je voulais
fichier.close()

# Et enfin j'assigne la position à l'objet:
obj.position = eval(posF)
```

Sauf qu'à la place de la position, j'aimerais sauvegarder une propriété d'un objet. La valeur de cette même propriété permettrait de repérer la progression du joueur. Si cette propriété est égale à 2, je sais que le joueur a complété les deux premiers niveaux. Via mes briques logiques je pourrai facilement faire en sorte que le joueur ait accès aux plateaux 1, 2, et naturellement le 3.

Bref, je pense que le vif du problème est tout là

[EDIT] : Voilà un blend cadeau (super méga basique, soyons d'accord) qui je l'espère, peut vous éclaircir davantage.

<http://dl.free.fr/mC2eoaMw2>

J'ai laissé comme propriété de base : disponible = 1. Seul le premier plateau est accessible. Essayer vous-même en lançant le jeu, il n'y a rien de sorcier là-dessous. Si vous quittez le jeu et que vous attribuez la valeur 2 à la propriété 'disponible', deux plateaux seront alors accessibles. Même principe pour la valeur 3 (mais je me suis arrêté ici hein

). Utilisez les flèches Haut et Bas pour diriger le curseur.

Il faudrait qu'à chaque lancement du jeu, l'objet 'menu' exécute une lecture du fichier python contenant la variable concernant les niveaux préalablement terminés, permettant d'attribuer à sa propriété 'disponible' la valeur indiquée dans ce fichier python. Les niveaux accessibles dépendent directement de la propriété 'disponible' détenue par l'objet 'menu'.

C'est pas très esthétique, mais il n'y a aucune raison pour que ce concept de sauvegarde ne marche pas