



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

**Topic: Bonne conception des classes (vertes, pour les enfants :-D )**

**Subject: Re: Python - l'initialisation et l'instruction return**

Posté par: Redstar

Contribution le : 25/3/2018 11:34:40

Citation :

Elle est appelée quand tu crées un objet. C'est un initialiseur...

Donc, tu veux dire que "init" n'est pas obligatoire quand on crée une classe ? Car moi, c'est ce que j'avais compris.

Si je comprends bien, ça veut dire que je pourrai très bien appeler une fonction de cette classe, tout en étant à l'extérieur (de la classe) ? (ex: ma\_classe.ajouter\_un\_texte(arg1, arg2, arg3))

J'ai testé le script (j'ai rajouté un "self" à "append", dans la fonction "add"). Le script est passé sans erreurs. Tu as dit que le "post\_draw" est un aimant à bug (ou est-ce l'ensemble ... = [write] ?), c'est bien juste ?

J'ai pas mon texte affiché dans le BGE. Comment est-ce possible d'avoir un aimant à bug si c'est lui qui permet l'affichage ? Je ne comprends pas.

Citation :

Un script n'a pas à être exécuté en boucle. Ce n'est pas la seule option. À ce stade, lis la doc.

Dans la doc python de blender ? C'est impossible. Ou as-tu trouvé cette phrase qui parle d'une telle hérésie ? Comment peux-tu, dans un environnement interactif, exécuter le script une seule fois ?

Comment peux-tu en une seule fois, virer un texte en vert si l'on place le pointeur dessus et le virer en rouge quand le pointeur est ailleurs (donc on suppose qu'il est rouge de base) ?

Citation :

Vas-y, fais-toi passer pour une victime...

Je pense que l'on a pas bien interprété ce que j'ai dit, je laisse passer.

Citation :

...une classe SceneElement qui reprend la méthode \_\_init\_\_, une classe AudioPlayer qui hérite de

SceneElement et reprend les deux autres...

Pourquoi suis-je obligé de séparer ça ? Point de vue éthique, on est d'accord, mais pourquoi créer juste une classe pour ça ? Pour le lecteur, peut-être ?

Citation :

```
super(game, self).__init__()
```

"Plus souple et plus facile à modifier par la suite"

"super" se réfère à "SceneElement" ? Mais enfin, comment est-ce possible ? Si je place "class audioplayer" en premier, c'est lui "super" alors ?

Citation :

```
current_game = Game()  
dico[current_game] = current_game
```

Seul game doit faire apparaître des créatures. Il y a bien, dans le HUD, quand j'ouvre l'inventaire du joueur, à faire apparaître les "items"... Donc, je pompe l'objet current\_game (ou game, plutôt ?) du dico, pour pouvoir faire ça ? Mais il y a plus d'arguments pour ces plans de l'inventaire.

Citation :

...minimum 40 scripts. Pas 1.

Il me semblait que c'était dangereux d'avoir beaucoup de scripts... Outre la facilité de lire ceux-ci, pourquoi faire ça ?

Donc, pour résumer le cas d'application d'une classe:

- Pas de init sauf si variable il y a.
- classe est un objet, variable = objet, donc variable est un "label", un surnom donné à la classe.
- ne sert que pour exécuter un ensemble d'actions, d'effet, comme pour une fonction qui évite les répétitions de code. Donc le reste doit rester à l'indentation zéro

J'ai oublié ou mal interpréter quelque chose ?