



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

**Topic: Les Game Property du Logic Editor.**

**Subject: Re: Les Game Property du Logic Editor.**

Posté par: Matpi

Contribution le : 22/5/2015 10:37:05

Heuuu... non...

Lorsque tu écris `mon_objet_python.ma_var = 42`, c'est une propriété de `mon_objet_python`. Le `globalDict` est un dictionnaire (un mapping) référencé comme `bge.logic.globalDict`.

On l'utilise donc comme ça:

```
bge.logic.globalDict[ma_var] = 42 print(bge.logic.globalDict[ma_var])
```

Bien sûr, on pourrait avoir bidouillé un truc pour que les attributs ajoutés à `bge.logic` (un objet de type module) soient intégrés dans le `globalDict`. Mais ce n'est pas le cas.

Précisons encore que le `globalDict`:

- 1) peut être sauvegardé en tant que fichier séparé (format pickle, si je me souviens bien, et ça justifierait le point suivant) - c'est tout son intérêt
- 2) n'accepte que des objets de type natifs, typiquement:

```
obj = bge.logic.getCurrentController().owner bge.logic.mon_obj = obj #ok  
bge.logic.globalDict["mon_obj"] = obj #erreur
```

Le `globalDict` fournit donc une possibilité de faire des sauvegardes de paramètres, etc., et il a des correspondances directement avec les briques logiques. Mais si tout ce que l'on souhaite c'est garder un objet entre deux exécutions d'un script, on préfère en général assigner directement à un module (`bge.logic. ... = ...`) ou - plus pythonique - utiliser les scripts dans le BGE en mode Module - et donc déclarer des variables globales.