



Forum: Moteur de jeu GameBlender et alternatives

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

Subject: Re: Python - l'instruction return

Posté par: DaWaaaaghBabal

Contribution le : 25/3/2018 21:53:22

Citation :

Vous avez des tutos sur la POO à recommander ?

... Bonne question. Je n'ai pas appris en ligne, donc je n'en ai pas sous la main. Une recherche rapide m'a permis de trouver [celui-ci](#) et [celui-ci, un peu plus détaillé](#). Les deux s'appliquent au Java, ce qui m'arrange en fait : ce qui compte, c'est les concepts, pas la syntaxe.

Mais ce ne sont pas vraiment des tutoriels, plus des présentations des concepts. Après, il y a [celui-ci](#) qui est un peu plus tourné vers la pratique. Il se finit par [ce bonus](#), que tu devrais lire avant tout le reste en fait.

Mais un tutoriel pour apprendre à programmer, ça revient un peu à demander un tutoriel pour apprendre à réfléchir. Or il y a un seul moyen d'apprendre à réfléchir : réfléchir. Donc, ce dont tu as besoin, ce n'est pas de tutoriels, c'est 1/ de bien comprendre ce que représentent les concepts de base du langage, dans l'abstrait, pas dans un cas concret dans un langage (Python) et un environnement (le BGE) spécifiques ; et 2/ de [faire des katas](#). Et pour l'instant, code en-dehors du BGE.

Citation :

De ton point de vue, c'est pas logique. Pour moi, ça l'est. C'est comme ça et pas autrement, j'interprète comme ça.

Je ne t'accuse pas d'incohérence ou d'illogisme, seulement de fonder ton raisonnement sur des bases imparfaites.

Citation :

En fait, ma question était dans le sens du "c'est comme ça et pas autrement ?".

Ah, je vois. Disons qu'il y a des tas de façons de faire, là je suis resté proche de ce que tu avais au départ, en essayant de respecter ce que je devinais de ta logique. En résumé, c'est la meilleure façon de faire ce que tu cherchais apparemment à faire. J'avais détaillé pourquoi, mais ce message commence à être très long donc si tu veux le raisonnement pas-à-pas, demande.

Citation :

En incluant l'outil de recherche, pour être honnête, oui, c'est ma façon à moi de procéder. J'aime bien avoir tout sous la main d'un coup. Même si on peut avoir autant de fenêtres Blender que l'on veut.

Ah, il y a aussi une question d'outils. Un bon éditeur de code présente une arborescence des fichiers du projet, donc en découpant tout la navigation devient plus simple puisqu'au lieu de chercher dans 4 000 lignes, tu cherches le bon fichier parmi les 5 qui sont dans le bon sous-dossier. Et tu peux garder plusieurs fichiers dans plusieurs onglets.

Mais si tu tiens à utiliser Blender comme éditeur de code... Je ne veux pas dire du mal de notre logiciel préféré, mais ça fait partie, avec le café, des choses qu'il ne fait clairement pas bien.

Après, je ne connais pas d'IDE Python, peut-être que Google en dira plus.

Juste une remarque : "je garde tout sous la main un coup"... Et ça te sert souvent ? Tu lis souvent tout une traite ? Tu travailles souvent sur tout le code à la fois ?

Note : j'avais aussi écrit un gros bloc concernant la séparation des responsabilités appliquée à l'ensemble du projet, mais peut-être qu'avant de se poser la question de l'architecture d'ensemble d'un gros système, il faudrait avoir une compréhension en profondeur des briques de base. On verra ça plus tard.