FORMATION DA-PYTHON / Projet 3

Lien Github : http://github.com/Neltarim/project_3

Préambule :

Ce projet m'a beaucoup appris sur la gestion d'une interface graphique. Il s'agissait de mon plus gros point faible et j'y ai acquis beaucoup de connaissances et compétences sur le sujet ainsi que sur l'approche orientée objet au fil de ce projet.

Architecture :

un fichier py principal (get_out.py) qui lance le programme et gère les events.

Un dossier contenant les ressources (png et jpg de chaque~~s~~ objets et level.txt qui est la structure initiale de la map).

un fichier texte de 15 par 15 représentant la map.

Package Labyrinth :

level.py, sert à générer la map via level.txt et la sauvegarde dans un tableau, donne les différentes positions initiales aux items de manière aléatoire, ainsi qu'aux personnages et aux murs. Il sert aussi à afficher le background, les murs et à gérer l'évolution du niveau durant la partie.

Characters.py, class permettant de créer et gérer le personnage ainsi que le garde (mouvements, affichage, ramassage d'objets etc).

items.py, class pour les différents items répartis sur la map.

const.py, liste des différentes constantes du programme, permet de gérer et changer plus facilement certains paramètres ainsi que les utiliser de manière globale dans tous les fichiers.

Difficultés et problèmes rencontrés :

Ayant commencé avec une méthode procédurale pour get_out.py, les inconvénients se sont vites faits ressentir principalement sur la gestion de la map. C'est alors que j'ai appris que même le programme main pouvait être un objet, cette perspective à totalement modifiée ma façon de penser et j'ai donc décidé de réécrire tout mon programme en suivant cette approche.

Une fois mon package créé, mon programme réécrit en objet, j'ai pu enfin passer à la phase la plus délicate : le debug.

A l'exception de quelques bugs mineurs (concernant des erreurs syntaxiques), l'un d'eux m'a posé certaines difficultés. Il s'agissait d'une erreur lors du lancement de la méthode blit() de la fenêtre avec pygame. Après recherches, il me fallait simplement inverser l'axe X et l'axe Y.

Conclusion :

Ce projet m'a demandé plus de travail que je ne pensais. Avant de commencer, j'avais déjà une idée précise concernant la construction globale du programme. En l'écrivant tout d'abord de manière procédurale pour la gestion globale du programme (get_out.py), il m'a fallu réadapter ma manière de penser et de rédiger ce programme vers la manière orientée objet complet.

Cette nouvelle méthode m'a permis de faciliter la mécanique globale, notamment d'avoir chaque objets en attribut pour ne pas avoir à les renvoyer en paramètre dans certaines fonctions.

J'ai donc maintenant une méthode d'écriture plus adaptée selon moi et je compte bien l'utiliser à l'avenir. J'envisage de la pratiquer plus fréquemment afin de l'améliorer au fil de ma formation. Je sera alors pour être capable de l'utiliser d'un manière efficace sur de futurs projets.