

Développement d'Application Python : Projet 3 « Aidez McGyver à s'échapper »

Lien vers repo : <https://github.com/woowoo91/projet3>

Pour la réalisation de ce projet, étant complètement débutant en programmation, je me suis inspiré de programmes existants sur internet (dans le milieu des développeurs de « roguelike » où j'ai trouvé les textures de mon « donjon »), j'y ai trouvé différentes méthodes, différentes manières de programmer, beaucoup de choses très intéressantes mais difficiles à appliquer à mon niveau.

Je me suis intéressé aux différents algorithmes de générations de labyrinthe mais je leur ai trouvé plusieurs inconvénients :

- Ils génèrent des couloirs répétitifs ennuyeux, pas de grandes salles
- Trop peu de cases dans le niveau de notre projet pour en justifier l'emploi
- Sans occultation du niveau on sait tout de suite où se trouve la sortie

J'ai donc décidé de conserver une structure de niveau faite main et de rendre l'exploration du niveau intéressante en l'occultant.

J'ai cherché différents algorithmes de gestion de la visibilité et de l'éclairage, je n'en ai trouvé qu'un seul de vraiment intéressant par son rendu mais c'est aussi le plus complexe et difficile à intégrer au jeu.

J'ai voulu occulter le niveau avec l'ajout d'un rectangle noir de la taille du niveau et un masque représentant le champ de vision du personnage et autres sources de lumières parsemées dans le niveau mais difficile de créer un gradient radial sous paint, j'ai tenté avec gimp aussi, autrement le résultat est moche et sans intérêt.

J'en suis donc resté à un programme très simple.

J'utilise un tableau pour créer le niveau en faisant une liste de rangées contenant une liste de caractères chargés un par un depuis le fichier texte, l'édition du niveau est très simple et il y a assez de caractères sur le clavier pour afficher un nombre important de textures et d'objets différents.

J'ai préféré charger le niveau sans position de départ et d'arrivée, je les fais apparaître après avec une méthode que j'espérais réutiliser pour faire apparaître de nouveaux sprites en fonctions d'évènements.

J'ai pensé que la meilleure façon de gérer l'inventaire du joueur était de créer une classe dédiée et d'utiliser un set, j'ai rencontré plusieurs bugs qui m'ont fait supprimer la classe et opter pour un dictionnaire.

Comme compteur d'objets, j'ai décidé de rajouter une ligne au fichier texte utilisé pour la création du niveau et de simplement y déplacer les objets récupérés.

J'ai ajouté des écrans au lancement (titre et instructions) et à la fermeture du programme (victoire ou défaite).