

Rétrospective

Ce projet fut le premier programme que j'ai eu à développer.

De ce fait, un certain nombres de difficultés se sont présentées dans la phase d'analyse et de conception. Principalement lié au manque d'expérience dans la manière de structurer un programme de ce type, ces difficultés furent surmontées en trouvant quelques tutoriels de création de jeux avec pygame.

Une fois passer cette difficulté de conception, j'ai décidé de structurer l'algorithme principal du programme autour d'une série de boucle. À l'origine, mon algorithme consistait en une boucle principale responsable de l'initialisation du module pygame et de deux sous-boucles. Une sous-boucle était en charge de l'affichage du menu/page de garde du programme ainsi que des interactions dans cette page. L'autre était responsable de l'affichage et de la gestion de l'interaction avec le niveau. Cette seconde boucle était la seule à faire des appels aux autres modules du programme afin de gérer la création et l'affichage du niveau, des objets, et des personnages.

Ayant été informé par mon mentor que la longueur de ma fonction main() (qui à ce point contenait plus de 100 lignes d'instruction) était beaucoup trop importante, j'ai procédé à une refonte de cet algorithme. J'ai donc ajouté dans des modules séparés la gestion du menu et celle du niveau. Mon algorithme a de ce fait été réduit à une boucle pour afficher le menu et une sous-boucle pour le niveau. La boucle principale affiche et ferme le menu, toujours utilisé comme page de garde de l'application. La sous-boucle est elle une fois de plus en charge de l'affichage du niveau et des interactions nécessaires pour jouer.

Au cours de ce projet, j'ai rencontré trois difficultés principales :

Premièrement, la question de la création de la structure du labyrinthe à partir d'un fichier a dû être résolue. Afin de pouvoir avoir un labyrinthe de 15 cases par 15 cases, les données du fichier devaient être importées sous la forme d'une matrice de cette même taille. Afin de créer cette matrice, j'ai choisie comme solution de créer une liste de liste. La méthode de création parcourt donc le fichier ligne par ligne et ajoute les éléments de chaque ligne dans une liste. Chaque ligne est donc contenue dans une liste et ajoutée ligne par ligne dans une autre liste.

Ensuite, le problème fut de savoir comment afficher de manière aléatoire les trois objets nécessaires à la victoire de telle sorte que :

1. ces objets soient affichés à chaque parcours de la boucle
2. ces objets ne soient affichés qu'une fois par type d'objet
3. ces objets ne soient plus affichés une fois que le personnage les a ramassés

Afin de se conformer à ces restrictions, j'ai pris comme solution de créer une méthode pour modifier directement la matrice structurale du labyrinthe en y ajoutant des marqueurs de position pour chacun des objets. Cette méthode permet d'avoir un repère permanent pour la localisation de

chaque objets tant qu'il est nécessaire de l'afficher. Et cette modification peut être inverser lorsque le personnage se trouve sur le marqueur, permettant ainsi de ne plus afficher les objets ramassés.

Enfin, le dernier problème rencontre lors de ce projet fut de simplifier ma fonction main() originelle. Ceci fut fait en créant deux nouveaux modules, l'un pour la gestion du menu et l'autre pour la gestion du niveau. Une fois ces deux modules créés, il m'a fallut retravailler mon algorithme afin de pouvoir :

1. afficher le menu lors du lancement du programme
2. pouvoir lancer l'affichage du niveau depuis le menu
3. revenir au menu lorsque le niveau entre en statut de fin de jeux

Ceci fut accompli en prenant le menu comme fondation du programme et donc la boucle d'affichage de celui-ci comme lié à l'ouverture du programme. Le module gérant le niveau affiché est quand à lui lié à une sous-boucle pouvant être ouverte ou fermer selon les actions de l'utilisateur.