

Comment j'ai codé mac Gyver : le jeu

Lien vers repo GitHub : https://github.com/GMBAMorera/OC_projet_3

1. Transformer le fichier lab.txt en carte

Ma première étape pour concevoir le jeu a été de définir à quoi devra ressembler le fichier .txt contenant le labyrinthe. Utiliser les lignes et les colonnes du fichier pour placer cases pour représenter les murs et le sol permettait d'avoir immédiatement une vision de ce à quoi allait ressembler le labyrinthe.

La question suivante était de savoir comment extraire les données du fichier et les utiliser. Mon premier réflexe a été de placer le fichier sur un seul string puis de déplacer le personnage en utilisant la division euclidienne. Puis j'ai découvert la méthode de open « readlines » qui redistribuait immédiatement mon fichier en lignes de texte dans une liste et me permettait d'avoir immédiatement une manière de carte découpé, comme un tableau en deux dimensions. Il m'a donc paru plus naturel de l'utiliser.

Quel que soit la manière dont je découpais la carte, cela demandait cependant que le labyrinthe soit prédisposé dans un rectangle parfait, j'ai donc introduit une exception chargé de vérifier si le labyrinthe était bien valide. Cela fournissait aussi rapidement un outil à destination des joueurs souhaitant modifier le labyrinthe afin de les aider à se conformer aux attentes du jeu.

Enfin, le défaut de cette découpe en liste de lignes est de parcourir moins efficacement le fichier à la recherche des objets. J'ai donc rajouté deux méthodes pour changer la carte en liste de caractères et vice-versa, particulièrement utile pour remarquer les problèmes de caractères sur la carte et aussi pour distribuer les objets aléatoirement.

2. Déplacer mac Gyver

Maintenant que le labyrinthe était mis en place, il fallait réfléchir aux déplacements du héros. L'idée d'avoir un objet à prendre en compte, mac gyver, et des méthodes à lui donner, tel que se déplacer, récupérer des objets, vaincre le gardien, m'a orienté vers la programmation orienté objet. Il m'a paru naturel de construire une classe Character sur laquelle instancier mac Gyver. Dans un souci de symétrie que j'ai plus tard remis en cause, la classe initiale contenait Mac Gyver, mais aussi le gardien et même durant un temps les objets à récupérer. Comme cela complexifiait plus le projet que cela n'aidait, j'ai finalement retiré les objets, puis le gardien qui s'est mué en simple attribut de mac Gyver.

Une autre question que je me suis posé à ce moment-là était de savoir si je devais retirer les objets du sol au passage de mac Gyver ou si je pouvais les laisser. Coder le fait de les retirer du jeu me paraissait une possible source d'erreur et de conflit, j'ai donc initialement conservé leur position et joué de la décomposition en nombre premier pour savoir si l'objet avait été pris au moins une fois. Ce choix n'a malheureusement pas survécu à l'affichage de l'inventaire du héros. Il en reste néanmoins la manière « multiplicative » de compter les objets de mac Gyver.

3. Afficher le jeu

L'étape suivante était d'afficher le jeu. L'affichage est découpé en plusieurs fonctions qui sont autant de sous-étapes dans la conception. Premièrement, l'affichage initial est obtenu en parcourant la « carte » du labyrinthe et en posant un sprite correspondant au contenu de la case du labyrinthe. Sont ainsi déposés mac Gyver, le gardien, les objets, les murs et les sols.

Une fois cela fait, il a fallu prendre en compte les déplacements de mac Gyver. Mac Gyver se déplace en fait à la fois sur le damier graphique et sur la carte du labyrinthe. Il modifie les graphismes en vérifiant les informations notés sur la case sur laquelle il veut se déplacer.

Enfin, afin de tenir compte de l'avancé, j'ai rajouté un cadre supplémentaire pour servir d'inventaire et montrer les objets déjà récupérés. C'est à ce moment-là que j'ai décidé de retirer de la carte les anciens objets en même temps qu'ils étaient retirés de l'affichage. J'ai aussi mis en place les crédits et la possibilité de relancer ou de quitter le jeu.

L'ultime étape a été de s'occuper de la sensibilité de la souris que j'ai réglée en rajoutant deux délais dans ma fonction, avant et après chaque déplacement.