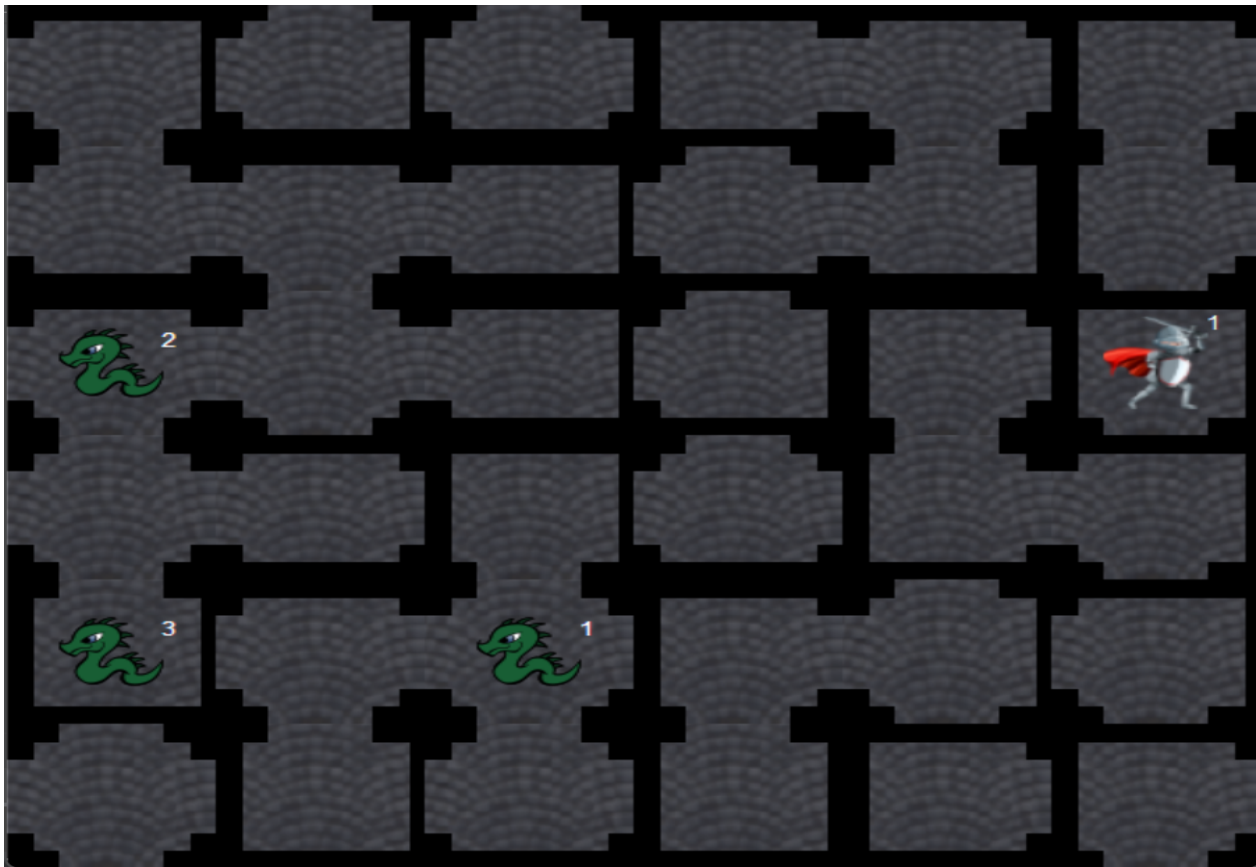


Wall Is You

Rapport de la création du jeu “Wall Is You” sur Python 3



Introduction

***Objectif :**

L'objectif est de recréer, avec l'aide de python, le jeu que nous avons appelé “Wall Is You”, un jeu puzzle. C'est un jeu dans lequel on incarne le donjon et où l'on cherche à aider l'aventurier à progresser.

Table des Matières

Introduction	1
Table des Matières	2
I - Guide d'utilisation	3
A - Lancement du jeu	3
B - Comment jouer	3
II - Fonctionnement de l'équipe et organisation	4
A - Organisation	4
B - Répartition	4
C - Difficultés	5
III - Avancement du projet	5
A - Avancement des tâches	5
B - Choix de structure de données	6
C - Impressions	7

I - Guide d'utilisation

A - Lancement du jeu

Le but du jeu est de faire en sorte que l'aventurier tue tous les dragons sans mourir, lorsque tous les dragons sont morts, la partie est gagnée. Si l'aventurier tombe sur un dragon qui est d'un niveau supérieur au sien, la partie est perdue.

Au lancement du jeu, il vous est demandé de choisir le donjon sur lequel vous souhaitez jouer. Il existe 4 donjons différents, chaque donjon a une architecture propre et l'emplacement des dragons est différent en fonction du donjon choisi.

B - Comment jouer

Commandes : Clic gauche de la souris - pivoter une salle, Espace - appliquer le chemin / fin du tour

Pour commencer à jouer une fois le donjon choisi, il faut tout d'abord repérer où se trouve notre aventurier et les dragons puis les Treasures s'il y en a. Ensuite le jeu débute, il faut donc cliquer sur les salles pour les faire pivoter pour que l'aventurier atteigne un dragon. Lorsque l'aventurier peut atteindre un dragon, un trait rouge apparaît entre les deux, il vous suffit donc d'appuyer sur la touche espace pour mettre fin à son tour de jeu et donc appliquer ce que l'aventurier prévoyait de faire.

II - Fonctionnement de l'équipe et organisation

A - Organisation

Pour l'organisation, nous avons décidé de tout faire pendant les séances de travaux pratiques jusqu'aux vacances scolaires. Ensuite lorsque les vacances sont arrivées, nous avons uniquement communiqué avec l'application Discord pour travailler et expliquer les incompréhensions. Puis à la fin des vacances, nous avons peaufiné notre projet lors de la dernière séance de TP. Donc au sujet du code, nous avons tous les deux travaillé dessus.

B - Répartition

Il restait alors à faire le rapport et la partie graphique avec le module ftk. Nous nous sommes donc réparti le travail, comme Mouayad est un peu plus à l'aise avec le module ftk que Loïc, Mouayad s'est occupé de l'interface de jeu et Loïc s'est occupé des menus de début et de fin. Loïc s'est chargé de la rédaction du rapport.

Donc pour la répartition du travail en termes de pourcentages on a :

- Loïc : 40%
- Mouayad : 60%

C - Difficultés

La seule difficulté rencontrée est au niveau du code car l'organisation n'était pas à plaindre. Pour le code, nous avons rencontré plusieurs problèmes, nous amenant à le changer en partie, voire totalement.

III - Avancement du projet

A - Avancement des tâches

Au début du projet, nous avons commencé par les fonctions utiles pour jouer au jeu, comme la fonction `pivoter` (qui sert à faire pivoter une salle du donjon), ainsi que la fonction `connect` (qui sert à voir si deux salles sont connectées entre elles).

Après cela, nous avons fait la fonction `charger` pour pouvoir récupérer les informations disponibles sur un fichier, ainsi que la fonction `charger_fichier_enregistrer` qui elle, permet de charger le fichier d'un jeu en cours.

Ensuite nous nous sommes attaqués à la plus grosse partie du projet : la fonction `intention` (c'est elle qui permet de déterminer l'intention de l'aventurier avant qu'elle soit appliquée par la fonction `appliquer_chemin`).

Puis nous avons fait les dernières fonctions avant de commencer l'interface graphique, soit les fonctions `cases_voisines` (regarder si deux cases sont collées), la fonction `application_chemin`, la fonction `rencontre` (qui sert à observer la rencontre entre un dragon et l'aventurier), la fonction `partie_gagne`.

Ensuite nous avons commencé l'interface graphique avec l'aide du module `fltk`. Pour la création du donjon, nous avons donc traduit les informations que l'on pourrait récupérer dans les fichiers donnés, en salle de donjons. Chaque élément présent dans ces fichiers correspond à un type de salle précis.

Après cela, il était temps de créer le programme principal, c'est grâce à lui que l'on peut lancer une partie. C'est lui qui sert à relier entre elles toutes les fonctions décrites précédemment et donc de les utiliser afin de jouer une partie du fameux jeu : "Wall Is You".

B - Choix de structure de données

Parlons maintenant de notre choix de structure de données. Après plusieurs réflexions, nous avons décidé de garder la structure proposée par le professeur étant donné que nous n'en trouvions pas d'autre plus appropriée. Donc pour représenter ce magnifique donjon, la structure globale est une "liste de listes de tuples", ce qui signifie que chaque salle est représentée par un tuple de 4 booléens, s'il y a un True c'est qu'à cet endroit là, la salle est ouverte, si c'est False c'est que la salle est fermée ici.

Ces tuples qui représentent les salles sont placés dans une liste qui représente la ligne du donjon contenant ces salles. Puis enfin ces listes sont elles-mêmes dans une grande liste qui représente le donjon. Par exemple, pour un donjon de 6 salles, une des structures possible serait :

```
[[ (False, True, True, False), (False, False, True, True), (True, False, False, True)],  
 [ (True, True, False, True), (True, True, True, False), (False, True, False, False)],  
 [ (False, False, False, True), (True, False, True, False), (False, True, True, True)]]
```

C - Impressions

Ce que nous avons aimé, c'est la partie du codage de l'intention de l'aventurier car cette partie était vraiment délicate car il fallait vraiment réfléchir pour coder cette fonction ainsi qu'utiliser la récursivité pour y parvenir. Nous avons aussi apprécié le fait que certaines fonctions étaient plutôt simples à coder d'un premier abord, mais que par la suite, nous avons rencontré des difficultés que l'on a fini par surmonter. Cela nous a montré notre évolution en programmation ce qui est assez plaisant.

Ce que nous avons un peu moins aimé c'est le fait que la structure de données nous soit directement conseillée, on aurait peut être préféré la chercher par nous même, ce qui aurait rajouté de la difficulté au projet mais aussi un côté intéressant.

Pour finir, le fait que ce jeu soit plutôt dur à coder, nous a permis d'utiliser tout ce que l'on a pu voir sur les deux premiers semestres, ce qui est plutôt stimulant, car on remarque que certaines choses que l'on a appris ont une utilité que l'on ne soupçonnait pas.