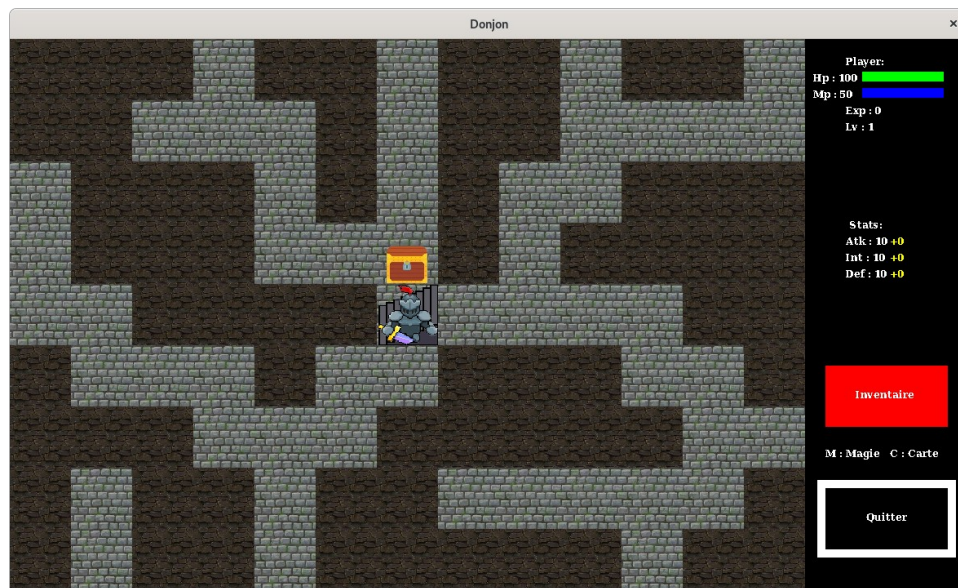


Rapport du projet Donjon Rogue



Introduction :

Le but de ce projet est de développer en langage C un jeu de donjon de type « Roguelike ».

Le joueur va explorer les différents niveaux de donjon, généré aléatoirement. Pour chaque niveau, plusieurs monstres et trésors sont aussi générés.

Le joueur gagne des points d'expérience en éliminant les monstres, et lorsque le joueur obtient un certain point, le joueur passe à un niveau supérieur en choisissant quel attribut il veut évolué.

Pour l'interface graphique, on utilise le bibliothèque MLV.

Répartition des modules :

Pour ce projet, on a réparti en 6 modules :

- Main pour que le programme soit exécuter ;
- Graphique qui s'occupe des interfaces graphiques ;
- Plateau qui se charge des générations du plateau du jeu ;
- Jeu qui gère tous les mouvements du jeu ;
- Elements qui permet de gérer tous les informations du joueur (l'état, le port des objets etc.) ;
- Donjon permet de stocker tous les informations de jeu et de le sauvegarder.

Explication des structures :

Dans la structure Save, nous avons 4 éléments pour sauvegarder les informations du jeu :

- un structure Niveau qui stocke les informations du plateau ;
- un structure Character qui stocke les éléments du joueur ;
- un entier qui signifie l'étage actuel ;
- un structure Inventaire qui stocke les informations des objets que le joueur à obtenu.

La structure Niveau possède aussi 4 éléments :

- le niveau d'étage ;
- un structure Plateau qui possède tous les informations du niveau générés ;
- un structure Case qui stocke la position de l'escalier pour aller au niveau suivant ;
- un pointeur de lui-même pour effectuer une liste.

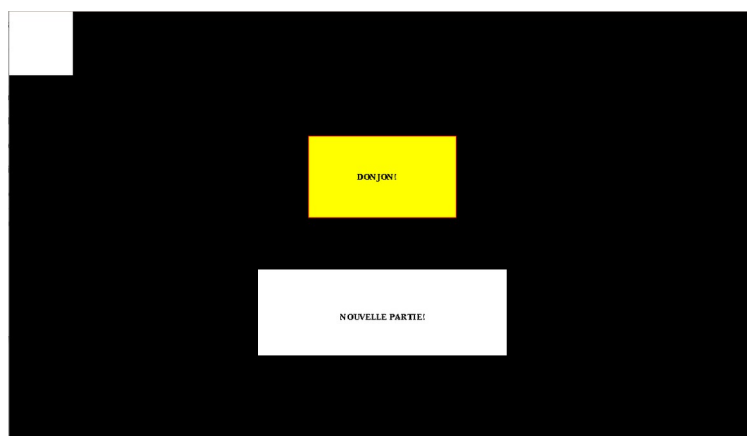
Le structure Character comporte 7 entiers (pour signifier son point de vie, de magie, le point d'attaque, d'intelligence, de défense, le point d'expérience et son niveau), le structure Case pour signifier sa position, un structure d'énumération pour signifier sa direction, et un structure Equipement pour signifier les équipements du joueur.

Cet structure est équivalent au structure Monster qui signifie les données de monstres.

Le structure Object comporte une chaîne de caractère et de la structure Equipement, qui est composé d'une chaîne de caractère (signifiant le nom), de structure Quality qui signifie la qualité d'équipement (Attaque, Intelligence ou Défense) et un entier pour signifier combien de point évolue en portant cet équipement.

Le structure Plateau comporte un tableau à deux dimensions de la structure Cell, qui comporte un structure d'énumération Celltype qui permet de distinguer le type de cette case, et un type union Entity si la case est un trésor ou un monstre.

Explication de certains fonctions :



Titre et menu d'écran

Au début du jeu, un menu d'écran apparaît dans une fenêtre. Lorsque le joueur sélectionne une nouvelle partie, on génère les informations du joueur avec la fonction `initPerso`, l'étage courant avec la fonction `insere_C`, qui prend en paramètre un pointeur de Donjon (équivalent d'un pointeur de Niveau), et un entier, et on le met la structure de sauvegarde.

La fonction `insere_C` permet d'allouer un niveau, en initialisant et générant le plateau, générant les entités et l'escalier pour aller au niveau suivant, au de le pointer au pointeur Donjon (sinon il pointe au pointeur suivant de Donjon, qui est une liste).

La fonction `etage_donjon`, qui prend en paramètre un pointeur du Niveau et un pointeur d'un entier, permet de mettre en point l'étage actuel.

Après l'affichage de l'écran avec une fonction de Graphique, `afficheMLV`, on entre dans une boucle `while` et on parcourt la fonction `tour_joueur_MLV`, qui prend un pointeur du Plateau, du Character, d'Inventaire, d'entier d'étage, et 2 entiers pour signifier la probabilité de la précision et de l'apprentissage (on va expliquer après ces fonctionnements).

Le joueur peut choisir de bouger avec les flèches du clavier, ou d'utiliser les potions qui comportent différents effets.

La fonction renvoie 0 s'il n'y a pas de changement spécifique, 1 si le joueur utilise l'escalier, 2 si le joueur souhaite utiliser les potions, ou 3 si le joueur souhaite terminer la partie.

Dans le cas de 0, on parcourt 3 fonctions :

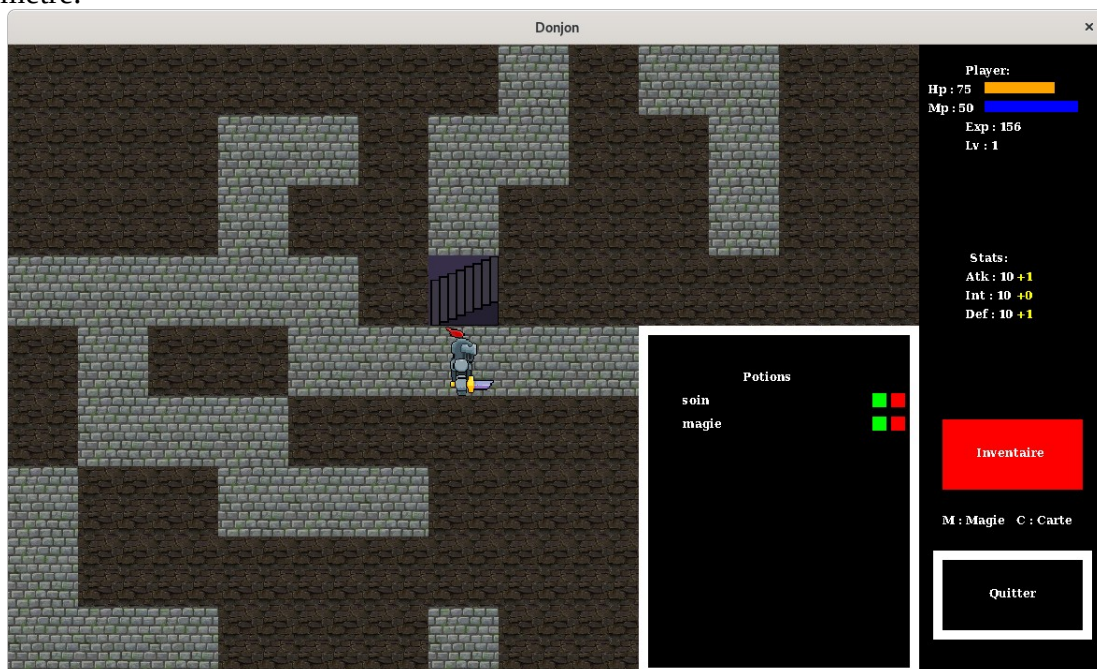
- `m_a_j_monster_treasure` qui reçoit les pointeurs de Plateau et de Character. Elle permet de changer le type de la case si le monstre est éliminé ou le trésor est ouvert ;
- `nouvelles_stats`, qui reçoit le pointeur de Character, permet de changer les stats du joueur (les paramètres, gestion des objets etc) ;
- `tour_monster`, qui reçoit les pointeurs de Plateau et de Character, pour gérer le tour de monster ;
- `verif_tour`, qui reçoit le pointeur de Character, les entiers de la probabilité de la précision et de l'apprentissage et un structure Tour (qui comporte le nombre de tour des potions), pour vérifier la validité des effets de potions.

Dans le cas de 1, si le type de la case est `STAIR_UP`, une valeur de vérification est à 1, sinon 0. Ensuite on vérifie si l'étage est changé, et on actualise l'étage avec la fonction `etage_donjon`. Si l'étage à changer, les coordonnées de joueur est à la place de l'escalier utilisé. A la fin, on vérifie le tour avec la fonction `verif_tour`.

Dans le cas de 2, une fenêtre dans l'écran est générée avec la fonction `InventaireMLV`, qui permet de proposer au joueur les potions qu'il veut utiliser.

La fonction `utilise_potion` permet de gérer l'utilisation des potions. Elle compare le nom des potion, et permet de distinguer si le joueur veut se soigner sa vie, son point de magie, ou bien se régénérer, ou augmenter sa précision d'attaque, ou avoir plus de point d'expérience.

Les tours et les valeurs sont donc stocker dans la structure Tour et les entiers, récupéré dans le paramètre.



Dans le cas de 3, on sort de la boucle et on sauvegarde la partie avec la fonction `sauvegarde_partie`. La partie sauvegardé peut être reçu lors du menu d'écran au début, en cliquant l'espace blanc en haut à gauche de l'écran.

Les options :

Nous avons fait plusieurs améliorations du jeu : la génération des plusieurs types de monstres, avec des caractéristiques différents. Et introduction des étages de boss, par exemple tous les 5 étages, dépendant des paramètres que vous avez choisis.

De plus, nous avons mis en place le map de l'étage en haut à gauche de la fenêtre pour se repérer où on est.

