

Rapport Projet

Manuel d'utilisateur:

Dungeon est un jeu style roguelike dans lequel vous allez explorer des salles à la recherche de l'escalier pour descendre à l'étage suivant. Vous rencontrerez des monstres qui gardent des coffres pendant votre exploration, affrontez et triomphez de ces monstres pour gagner de l'expérience et gagner des niveaux. Chaque niveau vous donne 3 points à attribuer dans un stats au choix entre Attaque, Intelligence et Défense. Faites usage des équipements et potions que vous trouverez dans les coffres pour survivre et descendre le plus profond possible !

Pour lancer le programme tapez “`./bin/dungeon`”.

Pour lancer le programme avec la carte en ASCII et avoir accès à la map graphique (touche m) tapez “`./bin/dungeon --map ou -m`”.

Pour afficher les commandes tapez “`./bin/dungeon --help ou -h`”.



Cliquez sur NEW GAME pour commencer une nouvelle partie.
Cliquez sur PLAY pour reprendre la partie à partir de la sauvegarde précédente. (Si il n'y a pas de sauvegarde, une nouvelle partie sera créée.
Cliquez sur EXIT pour quitter le programme.

Dans le jeu:



- 1: Votre barre de vie, si cette barre est déplétée, vous perdrez le jeu.
- 2: Votre barre de mana, ressource pour utiliser des attaques magiques.
- 3: Votre barre d'expérience, lorsqu'elle est remplie, vous montez d'un niveau
- 4: Le niveau de votre personnage
- 5: Les stats de votre personnage
- 6: Cliquez sur ce bouton pour attribuer des points dans des stats.
- 7: Le bouton de sauvegarde, créer un fichier de sauvegarde et ferme le jeu. Vous pouvez reprendre où vous vous êtes arrêté en cliquant sur PLAY à la prochaine lancer du programme !
- 8: Appuyez sur ce bouton ou sur A pour attaquer un ennemie en face de vous
- 9: Appuyez sur ce bouton ou sur E pour attaquer un ennemie en face de vous en utilisant de la magie (Utilise de la mana)
- 10: Appuyez sur ce bouton ou sur I pour accéder à votre inventaire.

L'inventaire:



Vous pouvez vous déplacer dans l'inventaire à l'aide des flèches directionnelles ou en cliquant directement sur l'icône des objets. Appuyer sur l'icône du doigt ou sur U pour utiliser la potion ou équiper l'équipement sélectionné. Cliquez sur la croix pour jeter l'objet sélectionné.

Lorsque vous ouvrez un coffre, l'inventaire du coffre s'ouvre à côté de votre inventaire et affiche les objets un par un. Pour récupérer l'objet il faut appuyer sur la flèche orange, pour le jeter appuyer sur la croix, une fois l'action choisie c'est le deuxième objet qui apparaît. Le petit nombre au-dessus des pièces d'équipement représente leur qualité. Plus leur qualité est haute, plus l'équipement sera fort !



Un gardien de trésors, ces chevaliers ne bougent pas d'un pouce et se tiennent toujours en face d'un trésor. On se demande à quoi ils peuvent

bien penser... La barre en rouge au-dessus de lui représente sa barre de vie. Battez le pour accéder au coffre qu'il défend ! Pour initier un combat, déplacez-vous vers lui, puis utilisez des attaques au corps à corps et des attaques magiques pour le vaincre. Si lors d'un combat vous affrontez deux ennemis en même temps cliquez sur l'ennemi que vous voulez toucher pour vous retourner vers lui.



Un coffre au trésor, classique...
Peut contenir les objets suivant:



Une armure, augmente votre défense.



Une épée, augmente votre attaque.



Une baguette magique, augmente votre intelligence.



Une potion de mana, restaure votre mana de 10%.



Une potion d'expérience, +30% d'exp pendant 30 tours.



Une potion de régénération, +20hp et +10 mp par tour pendant 30 tours.

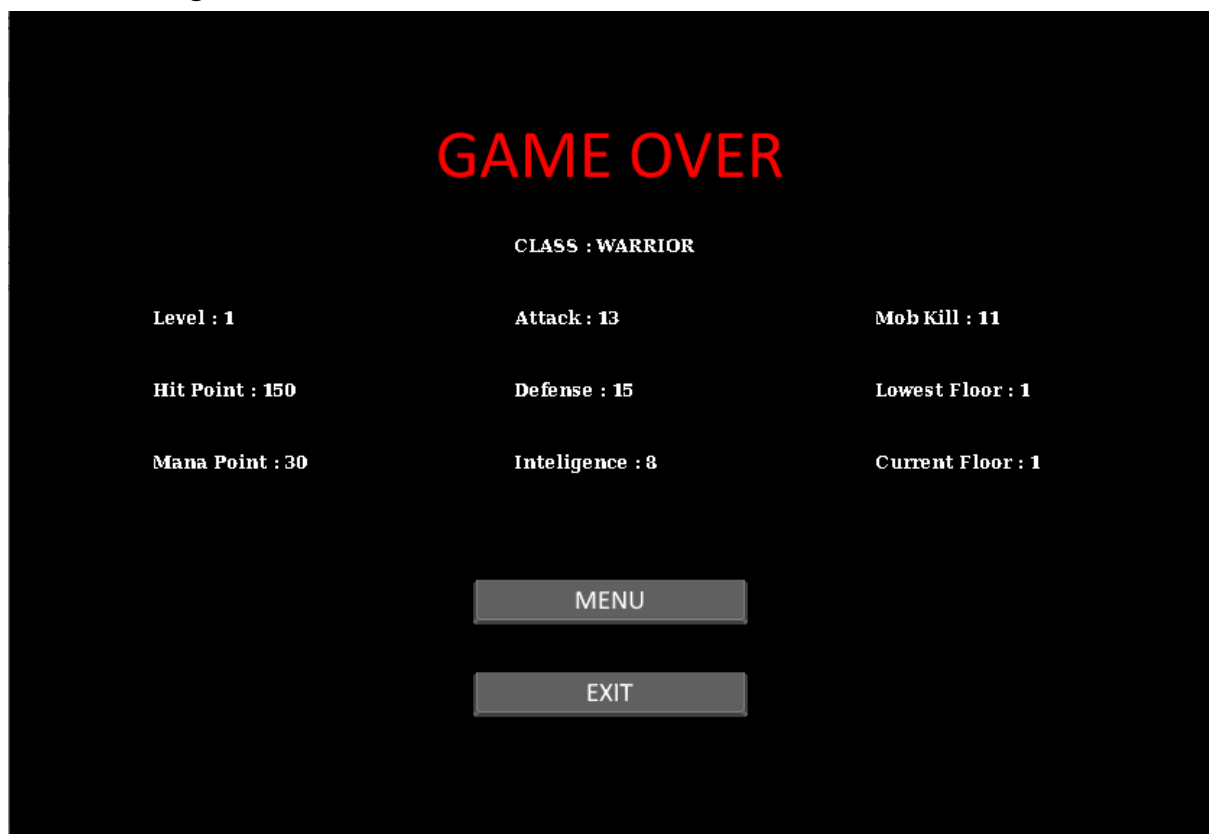


Une potion de soin, restaure votre vie de 10%.



Une potion de précision, augmente votre taux de coup critique de 5% à 15% pendant 30 tours.

L'écran de game over:



C'est un écran de game over qui résume votre partie.
Cliquez sur MENU pour retourner au menu principal.
Cliquez sur EXIT pour fermer le programme.

Améliorations

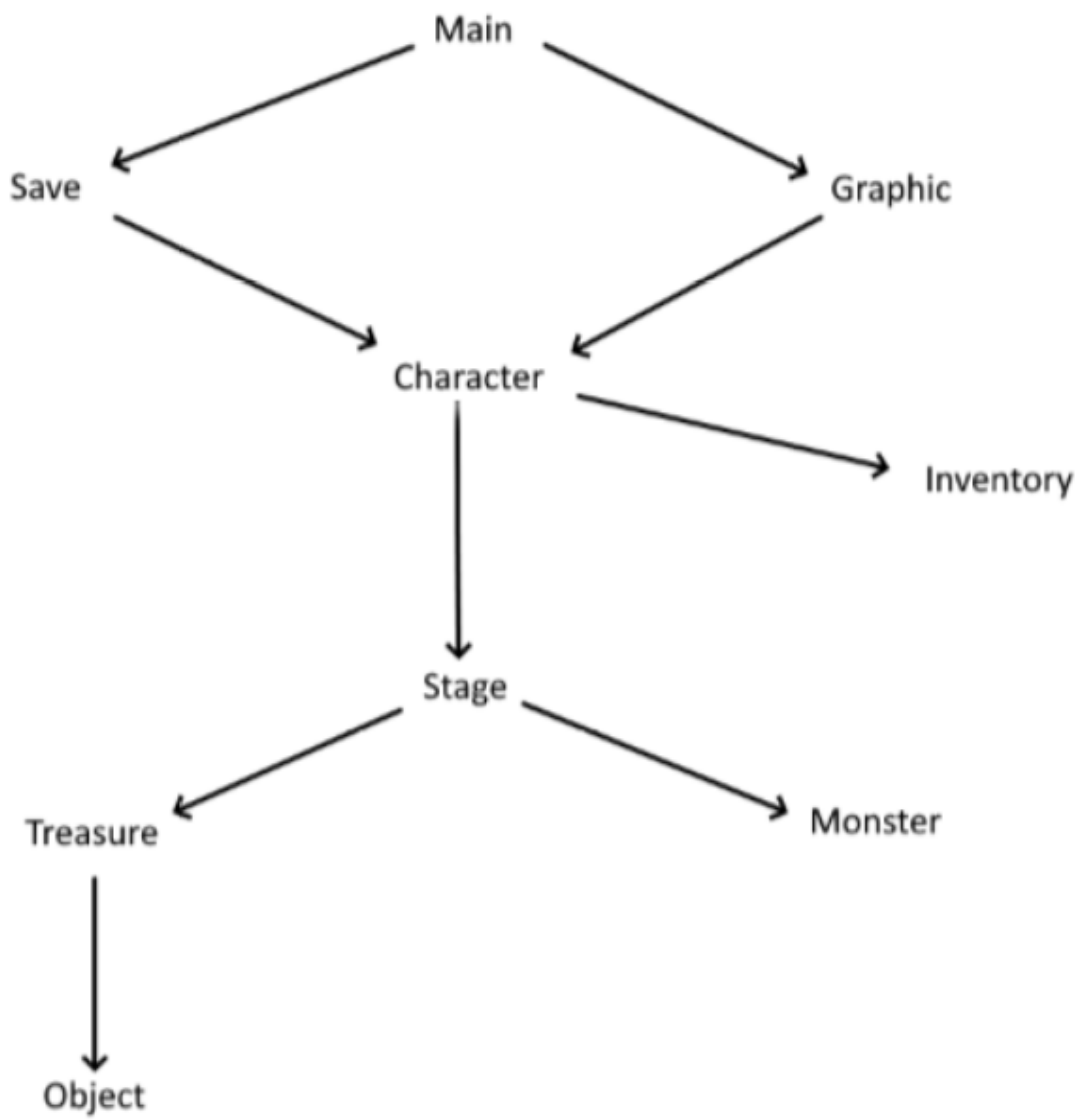
- Lorsque vous lancez vous pouvez choisir une classe parmi les 3 proposés : Traveler (Statistique de base), Warrior (Qui a plus de vie, plus d'attaque mais moins d'intelligence) et Mage (qui beaucoup plus d'intelligence mais moins d'attaque et de défense)
- Ajout d'une musique d'ambiance
- Les étages sont dans des thèmes différents (4 thèmes différents qui se répète)

Si vous avez lancé le programme avec l'option -m ou --map vous aurez accès à la carte ACSII dans le terminal et en appuyant sur m à la map graphique directement dans le jeu appuyez sur m pour l'enlever.



Rouge = monstre, Jaune = trésor, Vert = escalier montant, Violet = escalier descendant, Bleu = joueur.

Modularisation:



STAGE

Nous avons commencé par créer la fonction `generate_initial_stage` qui remplit le stage de mur (WALL) . Nous avons par la suite créé la fonction `generate_adjacent` qui renvoie une liste de 4 cellules. Ces 4 cellules sont les cellules adjacentes de la cellule dont les coordonnées ont été mis en paramètre. La fonction `alea_cell` va recevoir une liste de cellule et va choisir aléatoirement une d'entre elle pour l'extraire de la liste et la renvoyer.

La fonction `is_allowable` va vérifier si la case choisie peut devenir une ROOM à l'aide de la distance de manhattan que l'on a défini.

La fonction `is_not_at_edge` va simplement vérifier si la cellule sélectionnée n'est pas au bord du stage.

La fonction `add_toexpand` va ajouter une cellule à la liste `to expand` mis en paramètre.

Enfin la fonction `three_cell_wall` va identifier toute les cases ayant 3 murs adjacents et va changer ces cases en murs.

Pour finir nous avons créé la fonction `generate_maze` utilisant toutes ces fonctions afin de pouvoir modifier le stage en labyrinthe. La fonction génère une liste de 4 cellules adjacentes du centre du stage. Choisit une cellule aléatoire si elle est valable on la change en ROOM et on ajoute les salles adjacentes de celle-ci à la liste et on continue sinon on choisit aléatoirement une cellule tant qu'elle n'est pas valable ou que la liste est vide.

Ensuite il a fallu rajouter les entités (monstres et trésors), la fonction `room_to_treasure` va remplacer une ROOM par un TREASURE en générant un treasure, de même avec `room_to_monster` pour les monstres.

`start_treasure` va créer le coffre de départ, coffre étant placé à côté du joueur.

La fonction `add_all_entity` va faire appelle aux fonctions précédentes afin d'appeler le coffre départ au premier étage. et d'ajouter un coffre dans les cases ayant 3 murs adjacents et y placer un monstre devant.

La fonction `generate_stair_down` va identifier toutes les cases à une distance de manhattan de plus de 25 et en choisir une aléatoirement pour y mettre l'escalier descendant.

generate_dungeon va créer et initialiser un donjon qui aura au début 1 seul étage.

add_floor va ajouter au donjon un étage.

CHARACTER

Nous avons créé la fonction create_character qui initialise un Character ainsi que la fonction move permettant le déplacement du joueur en fonction de la direction donnée représentée ici par le x et le y. Cette fonction va également déplacer la caméra du joueur si elle n'est pas au bord du terrain.

La fonction go_down va ,si le joueur est au dernier étage, ajoute un étage au donjon. La fonction va également remplacer le joueur et sa caméra au centre.

La fonction go_up va , si le joueur n'est pas au premier étage, passer à l'étage supérieur et va retrouver l'escalier descendant de cet étage pour y placer le joueur et sa caméra. La caméra sera correctement positionnée même si l'escalier est au bord du terrain.

La fonction attack va renvoyer les dégâts de l'attaque en fonction de ces paramètres. Cette fonction est utilisée pour les caractères et pour les monstres.

La fonction effect_potion qui applique les potions qui ont plusieurs tours et décompte les tours de ces potions.

La fonction monster_attack fait attaquer tous les monstres adjacents au joueur.

La fonction interaction applique les différentes interactions en se déplaçant du joueur avec le stage (ouverture de coffre, première attaque, escalier, déplacement).

La fonction use_item permet l'utilisation d'une potion par le joueur.

La fonction level up ajoute un niveau et 3 points au joueur s'il dépasse un certain nombre d'expériences.

La fonction remove_equipment permet de déséquiper les équipements plus faibles.

La fonction chest_interaction permet de gérer les interactions lors de l'ouverture d'un coffre.

La fonction `change_direction` permet de changer la direction du joueur lorsqu'il clique sur une case du joueur permettant de changer de direction lors d'un combat.

La fonction `is_dead` détermine si le joueur est mort.

La fonction `character_interactions` regroupe toutes les interactions du joueur du jeu.

OBJECT

La fonction `generate_potion` créer une potion du type mis en paramètre. La fonction `generate_equipment` créer un équipement du type et de la qualité mis en paramètre.

La fonction `generate_object` va renvoyer un objet aléatoirement, une potion ou un équipement.

La fonction `restaure_potion` restaure une jauge (vie ou mana) en fonction d'un pourcentage mis en paramètre.

La fonction `init_stuff` initialise le stuff du joueur (EMPTY).

La fonction `equip` va équiper un équipement si il est de rareté supérieur à celui équipé actuellement.

La fonction `init_effect` initialise les effets du joueur (EMPTY)

La fonction `round_potion` ajoute les effets au joueur des potions ayant plusieurs tours. La fonction `round_count_down` diminue le nombre de tour, s'il arrive à 0 il enlève l'effet de `character`

La fonction `regeneration_potion` applique l'effet de la potion de régénération de même pour la fonction `accuracy_potion` avec la précision et la fonction `learning_potion` avec l'apprentissage.

INVENTORY

La fonction `create_inventory` créer un inventaire : un tableau de 12 objets, une variable permettant de savoir s'il est ouvert ou non et une variable permettant de savoir quelle objet est sélectionné dans l'inventaire.

La fonction `change_select` va changer l'objet sélectionné en fonction des touches directionnelles du clavier ou du clic souris.

La fonction `drop_item` remplace l'objet sélectionné par EMPTY (supprime l'objet) sauf si celui-ci est équipé.

MONSTER

La fonction `generate_monster` va créer et initialiser un monstre en fonction de ses paramètres (hp et atk).

La fonction `monster_is_dead` vérifie si le monstre est mort.

GRAPHIC

La fonction `load_all_images` va allouer l'espace nécessaire afin de charger l'ensemble des images dont on a besoin.

La fonction `free_all_image` va libérer l'espace utilisé par les images.

Toutes les fonctions sont des fonctions d'affichage.

SAVE

La fonction `save_caractere` écrit dans un fichier toute les données d'un caractère, de même pour `save_dungeon` avec le donjon.

La fonction `save` ouvre ou crée le fichier `save.bin` puis appelle les fonctions précédentes pour écrire dans le fichier. il ferme ensuite le fichier.

La fonction `load_caractere` récupère les données d'un caractère, de même pour `load_dungeon` avec le donjon.

La fonction `load` ouvre un fichier (s'il n'est pas présent retourne 0) appelle les fonctions précédente pour charger le fichier et ferme le fichier.

Difficultés rencontrées:

Nous avons eu du mal à trouver des sprites au départ, au final nous en avons créé quelques-uns à la main.

Ensuite, la taille du projet, le fait de devoir tout prendre en compte lorsqu'on modifie / ajoute de nouvelles choses a été assez difficile.

Enfin, nous avons eu un peu de mal à comprendre la sauvegarde en binaire car nos structures contenant des pointeurs vers d'autre structures qui eux même avaient des pointeurs etc...