

Rapport Projet Programmation 2

Antoine Grimod, Bastien Lhopitallier, Paul Robert

21 mai 2021

Changements dans le code de la partie 2

- Certaines textures ont été modifiées.
- Un bug affichant l'aide aux commandes par dessus du texte de la fenêtre de droite a été corrigé.
- De même pour celui autorisant l'affichage d'une zone d'attaque lorsque le joueur était à 0 PA, car cela pouvait faire croire à tort que le personnage pouvait encore attaquer.
- Il était possible de dupliquer des parties d'armure en essayant de les remplacer par d'autres, car l'ancienne armure était effectivement renvoyée dans l'inventaire mais la nouvelle n'en était pas supprimée (apparaissant donc à la fois dans l'inventaire et sur le joueur).
- Si le nombre d'objets dans l'inventaire excédait 10, les objets en trop étaient placés sur une autre page de l'inventaire, mais le curseur ne pouvait pas y apparaître, rendant ces objets inutilisables. Cela a été corrigé.

Ajouts pour la partie 3

Système de sauvegarde

Concernant le système de sauvegarde, il se déroule en deux étapes : transformer les informations présentes sur la carte en fichier JSON, puis réaliser l'opération inverse.

La carte est représentée comme suit : il s'agit d'un `Map[(Int, Int), Room]`, chaque `Room` possédant un `Map[(Int, Int), Tile]` et des `Vector` pour les ennemis, objets, portes, etc...

Les salles sont connectées entre elles au niveau des portes matérialisées dans cette version par des cadenas. Chaque `Room` possède un attribut `topLeft` indiquant les coordonnées absolues du coin en haut à gauche du plus petit rectangle contenant la salle. Les `Tile` ont ensuite des coordonnées relatives à `topLeft`.

Ainsi, pour sauvegarder la carte en un fichier JSON, on sauvegarde chaque `Room`, chacune sauvegardant ses tuiles, puis enfin chaque tuile sauvegardant ce qu'il y a dessus.

Cela génère un fichier JSON qui est envoyé aux joueurs. L'opération inverse reconstruit la carte à partir du fichier JSON créé lors de la sauvegarde en suivant le même procédé.

Mise en réseau

Grâce au système de sauvegarde développé ci-dessus, il est possible de faire dialoguer un ou plusieurs clients avec un serveur. Ce dernier ne possède pas de fenêtre graphique et va stocker la carte (il n'a pas à afficher le curseur ni les textures). Son rôle est de recevoir puis de traiter les actions effectuées par les joueurs pour ensuite leur transmettre les informations nécessaires pour qu'ils aient toujours un plateau de jeu à jour.

Ainsi, quand un joueur se connecte, il récupère la carte auprès du serveur, se voit attribuer une ID (ici, son adresse IP), puis demande ensuite à intervalles réguliers quelles ont été les dernières actions sur la carte, afin de mettre à jour sa copie. Enfin, lorsqu'il effectue une action, elle est envoyée au serveur qui vérifie sa validité, et le cas échéant l'effectue et l'ajoute à la liste des actions à renvoyer au client afin qu'il se mette à jour.

On conserve un `Map[String, Boolean]` indiquant, pour chaque ID, si le joueur correspondant a terminé son tour. Lorsque tous les booléens sont à `true`, le serveur fait débiter un nouveau tour de jeu pour tous les joueurs.

Problèmes et difficultés rencontrés

La génération aléatoire de carte était trop complexe à réaliser, notamment à cause du mécanisme de réceptacle censé apporter un peu de réflexion au jeu (placer l'objet lié au réceptacle dans la même pièce que celui-ci ne constitue pas un obstacle de taille), en effet il aurait fallu anticiper le placement des divers réceptacles et de leurs objets respectifs dans différentes salles sans bloquer l'avancée du joueur.

Les chaînes de caractères (en particulier les caractères d'échappement du type `\`) réagissent mal avec `upickle`. En particulier, nous n'avons pas réussi à transmettre la chaîne de caractères correspondant à la carte : le JSON généré est correct mais `upickle` induit un bug que nous n'avons pas réussi à corriger.

Les données étaient trop lourdes pour être directement envoyées sur le réseau, la carte dépassant la taille maximale autorisée pour un paquet. Elles sont donc maintenant compressées avant envoi.

Avec plus de temps (nous en avons perdu beaucoup à cause des problèmes de transmission de la carte), nous aurions pu réimplémenter toutes les fonctionnalités accessibles (discuter, prendre et poser des objets...). Ainsi seuls les déplacements et attaques sont disponibles pour les joueurs et les ennemis (tout est implémenté du côté serveur, mais il manque le retour client).