Architecture du projet TheBigAdventure

Nous avons articulé notre projet avec plusieurs packages :

`fr.uge.bigadventure`

Ce package contient les classes `GameMap` et `Input`.

- La premiere est la classe qui prend la grid et la liste d'elements du fichier .map.
- La seconde est la classe qui permet les opérations sur les touches du clavier

`fr.uge.bigadventure.analyser`

Ce package contient les classes nécessaires au parsing du fichier .map

- `Lexer` : Le lexer de Rémi Forax, un peu modifié par nous pour inclure le numéro de ligne
- `Token` : Enum utilisé pour le lexer, par Rémi Forax
- `Result` : Record utilisé pour le lexer, par Rémi Forax
- `Parser` : La classe pour parser un fichier .map, qui donne un `.map`

Les méthodes pour parser le `data` de la map et pour parser un `element` de la map, sont un peu trop longues, mais je n'ai pas trouvé comment les raccourcir 😕

`fr.uge.bigadventure.element`

Ce package contient les éléments, les personnages(joueur, ennemis, amis) même si les amis ne sont pas implémentés par manque de temps, ni les objets spéciaux comme Book et aussi les Door.

- `Element` : Soit un `Entity`, soit un `Item`, détaillé ci-dessous.
- `Behavior` : Enum pour le comportement des ennemis, n'est pas utilisé car on n'est pas allé au delà du comportement stroll.
- `Kind` : Enum pour le kind du fichier .map
- `GridElement` : Un bloc de la grille, soit un `Obstacle` ou une `Decoration`
- `Item` : Un élément de la map (ou de l'inventaire) qui est soit un `Weapon` soit un `InventoryItem` s'il ne donne pas de dégâts

- `Entity` : Soit un `Enemy` , soit un `Friend` (non implémenté, on a pas eu le temps), soit un `Player` , qui est configuré dans le `Main` pour être unique.

Remarque : On a pas le temps d'implémenter la zone pour l'ennemi ;/

`fr.uge.bigadventure.graphic`

Ce package contient ce qui attrait à l'interface graphique, faire scroller la map, faire apparaitre les elements, les blocs, les personnages, la map.

- `Graphic` : Fonctions pour preload les images, print la map, un bloc, un personnage, ou un élement
- `Main` : Contient le main, permet de prendre en charge les arguments `--level` , `--validate` et `--dry-run` de ligne de commande, d'exécuter les fonctions de `Graphic` et de gérer de manière expérimentale (uniquement un ennemi possible) la collision entre l'ennemi et le joueur

Amelioration apportées depuis la soutenance bêta

Le projet a bien avancé depuis la soutenance bêta, avec un parseur qui marche, néanmoins je n'ai pas trouvé d'autre moyen de voir si un objet de la grid était un obstacle ou une décoration autrement qu'en cherchant dans le filesystem, ce qui est couteux.

Nous avons aussi éclairci les classes/interfaces utilisées.