

Projet Programmation 2 : Rapport Partie 3

MANGEL Léo et MICHEL Thomas

1 Fonctionnalités ajoutées

1.1 Génération des salles

Pour générer des salles, on utilise des fichiers d'extension `.rdf` pour `room description file`. Les fichiers contiennent des presets de salles qui sont composés de trois parties qui sont chacune séparées par `'%'`. La première partie donne des informations sur la taille de la salle ainsi que l'emplacement des sorties de la salle, la seconde décrit la disposition de la salle et la troisième donne des informations sur le contenu de la salle.

Pour générer une salle, l'objet `RoomGenerator` sélectionne un des presets du fichier, puis utilise un objet de la classe `RoomParser` pour lire le preset et créer un objet de la classe `Room` qui stocke la position de tous les éléments d'une pièce par rapport à l'entrée de la pièce.

Pour donner les informations sur le contenu, nous avons décidé de parser un langage que nous allons décrire à l'aide des outils proposés par Scala. Voici la grammaire utilisée :

```
program → expr | expr conjunction program
      expr → number specialCharacter element
      element → character | item | "wall" | "lever" | "lockedDoor" | "lock" | "elevator"
character → enemy | "vending machine" | "computer"
      enemy → "easy" | "normal" | "hard" | "boss"
```

De plus, `number` contient l'ensemble des nombres entiers strictement positifs et `specialCharacter` contient toutes les lettres en majuscule et en minuscule.

Chaque expression de la forme `n c e` du programme ajoute l'élément `e` sur `n c` de la salle. Si il y a plus de `n c` dans la salle, on en choisit aléatoirement `n`.

Par exemple, dans une salle qui contient 4 `d`, l'expression 2 `d easy` va choisir 2 `d` parmi les 4 et placer deux ennemis faciles à ces emplacements.

1.2 Génération des niveaux

Afin de pouvoir incorporer les salles customisées dans les niveaux, nous avons du recommencer à zéro la génération de niveau. En effet, avec l'ancienne méthode, lorsqu'on plaçait les chemins entre les salles, on ne vérifiait pas que l'on ne recouvrait pas une autre salle.

Désormais, nous générons une première salle de départ et nous gardons en mémoire la liste des sorties non utilisées. A chaque étape, on choisit aléatoirement une sortie non utilisée qu'on retire de la liste et on essaye de placer une salle aléatoire à proximité. Si la salle recouvre une case déjà modifiée précédemment, on ne la place pas et on essaie à la place de relier cette sortie à une autre sortie non utilisée à proximité. Lorsqu'on a atteint le nombre de salle demandé ou que toutes les sorties sont utilisées, on s'arrête. Si le nombre de salle demandé n'est pas atteint, on recommence.

De plus, on place parfois des salles spéciales. On peut placer des salles avec des leviers et, si il y a des leviers non utilisés, des salles au trésor qui contiennent une porte qui sera liée à un levier. Enfin, la dernière salle de l'étage contient un ascenseur pour passer à l'étage d'après. Il y a trois types de salles finales : les salles avec un verrou, les salles avec des portes fermées. Les salles avec un verrou fonctionnent de la même manière que dans la partie précédente. Les portes fermées des salles finales peuvent se déverrouiller de trois manières distinctes : soit en tuant un boss se trouvant dans la salle, soit en activant un levier, soit en tuant tous les ennemis de l'étage.

2 Difficultés rencontrées