

Roguelike

2 novembre 2012

1 Système et concepts :

1.1 Invariants

- Un seul joueur par partie : toutes les autres entités sont simulées par une intelligence artificielle.
- Tour par tour, chaque tour étant déclenché par l'action du joueur.
- Les cartes sont des grilles 2D ; le plateau de jeu d'une partie est composé de plusieurs cartes qui ont des points de passages entre elles (portes, escaliers, portails, à sens unique ou non) ; typiquement, un donjon composé de plusieurs étages.
- Un seul personnage par case.
- Le joueur ne connaît de la carte que ce qu'il en a exploré.
- Le joueur ne voit que les objets et entités qui sont à sa portée.
- Même lorsque le joueur ne les voit pas, le comportement des objets et entités reste cohérent.

1.2 Jeu

Le joueur peut effectuer des actions toujours possibles (se déplacer), utiliser des objets de son inventaire (boire une potion) ou interagir avec des objets extérieurs (ouvrir une porte), dans le but d'explorer son monde, de vaincre des ennemis ou d'accomplir des quêtes.

1.3 Temps

Chaque action a une certaine durée.

Un tour de jeu correspond à une action du joueur. Un tour de jeu n'est donc pas une unité de temps fixe, mais peut être plus ou moins long dans le temps du jeu selon l'action effectuée, bien que cela soit transparent pour le joueur.

Durant ce tour, cette action est effectuée, et le temps de l'action s'écoule dans le jeu (les créatures susceptibles de se déplacer se déplacent, des événements peuvent se produire, etc).

Rien ne se produit dans le jeu sans action du joueur, qui peut donc prendre autant de temps qu'il le souhaite pour jouer, sans conséquence.

2 Choix d’architectures et d’implémentation

2.1 Actions

2.2 Objets

2.2.1 Objets mutables et constants

Les objets peuvent être séparés en deux catégories dont le traitement sera différent : les objets qui ont un comportement propre, et les objets « constants », i.e. dont l’état peut changer, mais pas de leur propre initiative.

Les premiers agissent à chaque tour de jeu ; ils doivent donc être parcourues à chaque tour et sont donc stocké dans une structure de type `List`. Ils contiennent une référence sur leur emplacement actuel. Lors de ce parcours, le moteur de jeu garde une référence vers les objets qui sont à portée du joueur.

Les seconds sont plus statiques, mais doivent pouvoir être récupérés rapidement depuis leur position : ils sont donc stockés dans une structure de type `Map`.

2.2.2 Fonctionnalités

Une fonctionnalité d’un objet correspond à un trait dont il bénéficie. Voir le fichier `/src/draft/traits.scala`, par exemple.

2.2.3 Position d’un objet

La position d’un objet peut être :

- sa position sur une carte donnée – i.e. un triplet (x, y, z) , z référencant la carte.
- une référence vers un objet de trait `container`.

3 Questions en suspens

- Représentation de la carte ?

4 Extensions envisagées