IFT1025 - Travail Pratique 1 (Énoncé partiel)

RobotFindsKitten : Super Dungeon Master 3000$_{Turbo}^{Ultra}$ Edition

Concepts appliqués : Programmation orientée objet, héritage, développement d'application de taille minimalement intéressante

Contexte

RobotFindsKitten (traduction : RobotTrouveChaton) est un jeu pour `MS-DOS` conçu en 1997 par Leonard Richardson.

Il s'agit d'une *simulation zen* : vous incarnez un Robot (`#`) qui cherche Chaton (affectueusement surnommé *Kitten* dans le reste du TP) et qui examine chaque objet sur la grille, jusqu'à ce qu'il trouve chaton (pas de contrainte de temps, pas de barre de vie, pas de score, . . .)

L'objectif principal est de passer un bon moment en contemplant les divers objets non-chatons (*Non-Kitten-Items*) qui se trouvent dans le niveau.

Exemple de jeu

Le jeu est représenté sous forme de grille en caractères ASCII :

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                        [j                            f           Y        %
%                                F                            Z             %
%   I A       Oc  k                                         Y               %
%    _    ]                             sC         #       A            G   %
%                   a            t         X                        b       %
%  k     o                         `                                 A      %
%           j                             '         Y    ;                  %
%         T Y   l   y             j                                 J I     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Chaque fois que le robot marche sur un item qui n'est pas le chaton, une description de l'item en question est affichée.

Quelques exemples :

> This is a tasty-looking banana creme pie.
>
> It's a business plan for a new startup, kitten.net.
>
> An automated robot-liker. It smiles at you.
>
> A book with "Don't Panic" in large friendly letters across the cover.

Chaque item sur la grille a une description fixe tirée au hasard en début de partie depuis une banque de descripteurs de *Non-Kitten-Items*.

Le jeu se termine lorsque le Robot trouve Chaton.

Une version web du jeu original est disponible ici : http://robotfindskitten.org/play/robotfindskitten/

Votre travail

Vous devrez recréer ce jeu, en y ajoutant quelques éléments :

- Plutôt que d'être une grande salle ouverte, la grille de jeu est divisée en pièces fermées à clé
- Le robot peut collecter des clés pour ouvrir les portes et passer à une prochaine pièce
- Chaque pièce contient quelques *Non-Kitten-Items* et une clé (représentée par une apostrophe)

De plus :

- Un des *Non-Kitten-Items* sur la grille est un *Téléporteur*, qui permet de se rendre à une position aléatoire de la grille sur demande

Le programme final attendu vous est donné en exemple avec le fichier `rfk.jar`.

Vous pouvez le télécharger et l'exécuter en ligne de commande avec :

```
java -jar rfk.jar
```

Déroulement du jeu

1. La partie commence par un message de bienvenue :

```
      Bienvenue dans RobotFindsKitten
Super Dungeon Master 3000 Ultra Turbo Edition !
```

2. La grille de jeu est générée semi-aléatoirement :
   - Des murs (`%`) sont installés autour de chaque pièce avec une porte au milieu (`!`) pour passer d'une salle à l'autre
   - Une clé est déposée sur une case au hasard dans chaque pièce
   - Les *Non-Kitten-Items* et le *Kitten* sont disposés sur la grille de façon aléatoire (la représentation des *Non-Kitten-Items* et du *Kitten* est déterminée aléatoirement parmi un ensemble de caractères ASCII)
   - En plus des *Non-Kitten-Items* un *Téléporteur* est caché quelque part sur la grille de jeu

La grille une fois générée devrait avoir l'air de quelque chose comme :

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  F       e %        P    %  ‘         % T @        %            ’%
%            %Y       V    %         M’ %      d     %             %
%    gf      !           v!    j        !  x     z   !     #     @ %
% ’          %       m    %            %  |  ’      %             %
%            % e’ ] p     %g         | % i    o      %     g      %
%%%%%%!%%%%%%%%%%%%!%%%%%%%%%%%%!%%%%%%%%%%%%!%%%%%%%%%%%%!%%%%%%
%    j       %  }      o  %           w%            %             %
%F           %         g  %            %            %          u %
%        W   !       [    !’   d       !        e  t! ?          %
%’           % N    z     %            %           % ’      v    %
%    @   t   %’      @     %    T S  h %       ’    %^       E    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

3. La grille est affichée, puis un prompt demande à l'utilisateur d'entrer son mouvement :

```
R.O.B. [0]>
```

Le prompt représente le status du robot :

`{Nom du robot} [{nombre de clés en sa possession}]`

4. Le robot peut explorer la carte en entrant les commandes `w`, `a`, `s` ou `d`, respectivement pour se déplacer dans les directions *Haut*, *Gauche*, *Bas* ou *Droite*

5. Lorsque le robot ramasse une clé, il peut s'en servir pour débarrer une des portes (qui restera débarrée pour le reste de la partie)

6. Si le robot possède le téléporteur, il peut entrer la commande `t` pour se téléporter sur une case libre aléatoire sur la grille. De plus, un `T` est ajouté à la fin du prompt pour indiquer qu'il peut se téléporter :

```
R.O.B. [0]T>
```

La grille est affichée et le prochain mouvement est demandé jusqu'à ce que Robot trouve Chaton

6. Lorsque Robot trouve Chaton, le message suivant est affiché :

```
You found kitten! Way to go, robot.
Caramel <3 R.O.B.
```

Où `Caramel <3 R.O.B.` correspond à `{Nom du Kitten} <3 {Nom du Robot}`

Et le programme se termine.