UFR ST - Besançon- L2 Info - Année 2015/16 Programmation par Objets

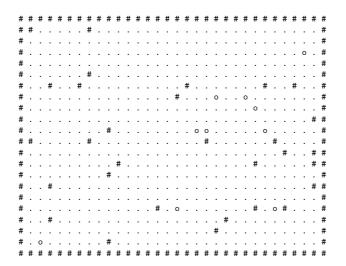
TP 8 - Héritage, polymorphisme : préparation au projet

Le but de ce TP est de préparer les développements pour le projet.

Il s'agit simplement dans ce TP d'afficher un damier avec :

- soit des cases de type Vide : .
- soit des cases de type Mur : #
- soit des cases de type Balle : o

comme illustré dans l'exemple suivant :



Vous mettrez cependant en place une modélisation de ce problème mettant en jeu les notions d'héritage, de classe abstraite et de polymorphisme.

Le damier est une matrice (un tableau à deux dimensions) de types Case, où Case est une classe abstraite dont dérivent tous les objets que l'on peut placer sur le damier.

Classe abstraite Case

Une case possède une position, définie par deux entiers, et représentant ses numéros de ligne et de colonne dans le damier. Vous définirez une classe Position pour enregistrer une telle position.

Une méthode isFree(), abstraite dans Case, indique par un booléen si une case est libre ou pas sur le damier. Les cases de type Vide seront des cases libres, mais les autres cases (de type Mur ou Balle) ne seront pas des cases libres.

Sous-classes concrètes de Case : Vide, Mur, Balle

Les cases de type Vide, Mur et Balle sont à coder comme des sous-classes concrètes de Case. Cela signifie qu'elles héritent de Case mais ne sont pas abstraites. Chacune d'elles doit donc redéfinir la méthode isFree(). De plus, chacune d'elle implante la méthode toString() de la manière suivante :

```
dans Vide, toString() renvoie "."
dans Mur, toString() renvoie "#"
dans Balle, toString() renvoie "o"
```

Les cases Balle, en plus d'une position, possèdent une direction car elles seront amenées à se déplacer.

La direction sera codée par une classe Direction, sous la forme d'un couple d'entiers, chacun d'eux valant soit 1, 0 ou -1. Ce codage permettra dans le projet d'indiquer la variation à appliquer à la composante correspondante de la position : la valeur 1 indique que la valeur (en x ou en y) de la position doit être augmentée, la valeur -1 qu'elle doit être diminuée, et la valeur 0 qu'elle doit rester inchangée. Par exemple, avec un point de référence situé en haut à gauche, la direction nord est codée par le couple (0, -1), et la direction sud par le couple (0, 1). La direction ouest est codée par le couple (-1, 0). La direction sud-est est codée par le couple (1, 1). Etc.

Classe Damier

Un damier peut être instancié avec un nombre de lignes et de colonnes fournis en paramètre. La matrice de Case est un attribut de cette classe.

Les bords du damier sont des murs. L'intérieur est constitué de cases vides, sauf pour certaines qui sont des murs ou des balles disposés aléatoirement.

Une méthode getRandomFreePosition() doit permettre de retourner une Position correspondant à une case vide du damier, et choisie aléatoirement parmi les cases libres.

Une méthode setRandMur() permet de positionner une case Mur au hasard (selon une position choisie par getRandomFreePosition()). Même chose pour une case Balle avec une méthode setRandBalle().

Enfin, le toString() de Damier construit une chaîne représentant le damier. Il suffit pour cela de parcourir le damier en invoquant le toString() de chaque case, qui par polymorphisme renvoie la chaîne la représentant. N'oubliez pas d'insérer dans la chaîne le caractère de saut de ligne '\n' à la fin de chaque ligne du damier.

Programme principal

Votre programme principal consistera simplement à instancier un damier, à y disposer aléatoirement des cases de type Mur et des cases de type Balle, puis à l'afficher (en affichant la chaîne construite par son toString()).

Par exemple, pour obtenir l'image du damier présenté en exemple au début de ce sujet, on a créé un damier de 22 lignes et 32 colonnes. Comme les bords sont des murs, il reste un espace de 20 lignes par 30 colonnes pour ajouter des obstacles (Mur) et des balles. Dans cet espace, on a disposé aléatoirement 30 cases de type Mur et 10 cases de type Balle. Les autres cases sont des cases de type Vide.