

## Projet du module “Conception de Programmes à Objet”

## Oh non, des mulots !\*

## 1 Présentation

Dans l’aventure qui nous attend, le Chi, un fameux utilisateur maladroit d’ordinateurs, furieux de ne pas arriver à diriger son mulot, l’a fracassé violemment contre l’écran de sa machine. Du coup les mulots de tous les PC de l’immeuble, pris de panique, se sont enfuis. Il faut en sauver le plus grand nombre en les guidant vers la sortie.

Ces mulots sont les héros d’un jeu que vous devez programmer en Java et qui est inspiré du fameux “Lemmings” de Psygnosis.

### Règles du jeu

*Notez bien que ces règles ne sont pas identiques à celles du jeu original “Lemmings” : elles sont plus simples et parfois un peu différentes.*

Le jeu comprend une série de niveaux de plus en plus complexes. La réussite d’un niveau permet de passer au niveau suivant. A chaque niveau, des mulots apparaissent à l’écran depuis une entrée et avancent ensuite bêtement tout droit, horizontalement (de gauche à droite). Le but du joueur est, à chaque niveau, de faire sortir un certain nombre de mulots, nombre précisé lorsque le niveau commence. Des obstacles et pièges divers sont disposés entre l’entrée où arrivent les mulots et la sortie du niveau. Ces obstacles peuvent bloquer les mulots, les tuer ou leur faire faire demi-tour. Heureusement il y a des moyens de les éviter, si le joueur est suffisamment malin, sans qu’un trop grand nombre de ces charmantes bestioles ne passe de vie à trépas.

Chaque mulot a une *profession*. Lorsqu’ils apparaissent dans le niveau, normalement en tombant de l’entrée sur une plate-forme plus ou moins grande, les mulots sont des **marcheurs** : ils avancent bêtement sur le terrain de gauche à droite et font demi-tour lorsqu’ils rencontrent un mur. Si la plate-forme n’est pas bordée de murs, les malheureux tombent alors dans le vide et si la chute est trop haute ils s’écrasent sur le sol.

Pour essayer de les rapprocher de la sortie, le joueur peut faire changer de profession à certains mulots, chaque profession ayant un comportement qui lui est propre. Les différentes professions sont expliquées ci-dessous. Le joueur ne peut transformer qu’un mulot à la fois. Pour cela il sélectionne une profession puis un mulot. La façon d’effectuer la sélection de la profession et du mulot est discutée plus bas.

- Les **grimpeurs** : ils sont capables de grimper le prochain mur qu’ils rencontrent, sinon, sur un terrain horizontal, ils marchent normalement. Si le mur monte jusqu’au plafond ils ne sont pas capable de marcher la tête en bas et donc ils retombent alors à terre et font demi-tour (en s’éloignant du mur). Attention néanmoins: si le mur est trop haut et qu’il y a un plafond, la chute est mortelle. Après avoir escaladé un mur (avec chute éventuelle) ils redeviennent des marcheurs (s’ils survivent à la chute éventuelle).
- Les **parachutistes** : ils ne s’écrasent pas, même s’il tombent de haut, à leur prochaine chute. Sur le plat ils marchent normalement. Après une chute, haute ou non, ils redeviennent des marcheurs.
- Les **maçons** : ils se mettent immédiatement à construire un escalier de hauteur fixe (disons 5 marches par exemple) mais s’arrêtent de construire dès qu’un obstacle les empêche de continuer (un mur notamment). Ils redeviennent alors de simples marcheurs.
- Les **creuseurs** : ils se mettent immédiatement à creuser un trou vers le bas, du moins si le sol n’est pas trop dur (métal). Quand ils rencontrent le vide ils tombent et redeviennent alors des marcheurs (ou s’écrasent si le sol est trop bas). Evidemment les mulots qui les suivent tombent dans le trou.
- Les **perceurs** : ils percent horizontalement le prochain mur qu’ils rencontrent, jusqu’à tomber sur du vide. Ils redeviennent alors de simples marcheurs.

---

\*La version originale de ce sujet est due à Mikal Ziane

- Les **bloqueurs** : ils s'arrêtent net et restent immobiles. Ils bloquent alors les mulots qui les suivent, qui font donc demi-tour. Un bloqueur est définitivement perdu, il n'y a pas moyen de le retransformer, sauf en bombe auquel cas il meurt de toute façon.
- Les **bombes** : elles explosent immédiatement et disparaissent en creusant un trou dans le sol, ainsi que dans tout mur suffisamment proche. Le trou n'est pas très gros et sera de forme carrée pour simplifier. Une seule bombe ne creuse pas assez profond pour que le trou résultant empêche les marcheurs de passer: ils le considèrent comme un petit escalier. Par contre une deuxième bombe explosant dans le trou l'agrandit au point d'empêcher le passage des marcheurs.

A chaque niveau le joueur dispose d'un nombre limité d'utilisations de chaque profession. Ainsi par exemple il ne pourra former que 10 parachutistes et 5 grimpeurs. Bien entendu il est possible de retransformer plusieurs fois le même mulot du moment qu'il reste assez d'utilisations de la profession choisie. Le joueur devra faire un plan pour sauver le nombre de mulots requis et ce plan peut-être très subtil ce qui fait l'intérêt du jeu.

## 2 Travail à faire

Vous devez réaliser un programme permettant de jouer. Ce programme devra au minimum prendre en compte deux professions, les déplacements propres à ces professions, l'entrée et la sortie d'un mulot ainsi que le changement de profession. Vous êtes libres d'introduire de nouvelles professions (ayant des déplacements ou des agissements autres que ceux prévus ci-dessus).

### Interface utilisateur

Votre programme doit présenter une interface textuelle. L'état du jeu y sera présenté après un déplacement de chaque mulot s'y trouvant (un pas de simulation). Les mulots y seront représentés par un caractère et dans le cas où plusieurs éléments se trouvent sur une même case, un seul d'entre eux sera affiché. Des lectures non bloquantes des caractères tapées au clavier peuvent être réalisées en employant la méthode de classe `clavier()` de la classe `Outils` qui vous est fournie. Un programme illustrant l'utilisation de cette méthode est à votre disposition.

### Qui, quoi et quand?

Votre projet doit être fait en binôme. Les groupes de 3 ne seront pas acceptés. Évitez de faire votre projet tout seul (soit vous êtes très fort et des personnes ont besoin de votre aide, soit vous avez des difficultés et il faut vous faire aider).

Vous devez porter une attention particulière à la rédaction de votre dossier. Sa qualité est déterminante pour l'évaluation de votre travail. Votre dossier doit être un unique document dont la composition est la suivante :

- Une page de garde indiquant le nom et **le groupe** des membres du binôme, l'objet du dossier.
- Une table des matières de l'ensemble du dossier.
- Une brève introduction du projet.
- Le diagramme UML des classes et paquetages formant vos applications (inutile de préciser les attributs et les méthodes de chaque classe, seules les relations entre les classes doivent être reportées).
- Le listing des tests unitaires de vos classes (précisez quels tests passent et lesquels échouent).
- Le listing complet de vos sources (triez les classes selon un ordre logique).
- Un bilan du projet (les difficultés rencontrées, ce qui est réussi, ce qui peut être amélioré).

Nous vous rappelons que le critère principal de notation est la structuration de votre application. Votre rapport doit mettre en avant la qualité de celle-ci.

Vous devez rendre votre rapport complet imprimé durant la séance de travaux pratiques de la semaine du **23 mai 2016**. De plus, vous déposerez à la même date une archive portant votre nom et contenant l'ensemble de vos fichiers sources dans le puits BPO. Seules les archives au format `jar` ou `zip` seront acceptées.