

# M1 Bio-info – JAVA et Programmation Objet

Projet 2013-2014  
"Paie pas de Mines"

Le but de ce projet est la programmation d'un jeu graphique.

## 1 Modalités d'évaluation

Le projet est à faire seul ou en binôme. Il fera l'objet d'une première évaluation le **22 novembre** puis d'une soutenance le **10 janvier**. Cette dernière durera 15 minutes : 10 minutes pour présenter votre projet et 5 minutes de réponse aux questions.

La note tiendra compte de la qualité de la modélisation **objet**, de la bonne programmation et du bon fonctionnement. Un code bien commenté sera valorisé. **Un projet écrit à la C et/ou ne contenant que des méthodes statiques peut très bien avoir une note en dessous de la moyenne même s'il fonctionne parfaitement.**

## 2 Cahier des charges

Vous avez cinq semaines pour réaliser un jeu graphique **tour par tour** "Paie pas de Mines" **jouable au clavier**, au concept proche de celui du démineur, et dont l'objectif est d'aider le héros à s'échapper de la carte en évitant les mines. Vous devrez respecter les spécifications suivantes :

### 2.1 Environnement

- La carte est une grille carrée.
- La carte comporte une case "Départ" dans le coin haut à gauche, et une case "Arrivée" dans le coin bas à droite.
- La carte possède un certain nombre de mines disséminées sur les différentes cases.
- La grille de jeu est affichée dans une fenêtre graphique.

### 2.2 Conditions initiales

- Au lancement du jeu, on doit arriver sur un menu qui demande à l'utilisateur la longueur du côté de la grille ainsi que le nombre de mines. Il propose de commencer

une partie selon deux options d’affichage : un mode normal et un mode ”mines visibles” de test.

- Le nombre correspondant de mines sont placées sur les cases de la carte aléatoirement.
- Le joueur commence la partie sur la case ”Départ” dans le coin en haut à gauche.

## 2.3 Gameplay

- A chaque tour, comme dans un vrai démineur, on indique sur la carte, et pour l’ensemble des cases voisines de celles déjà parcourues, le nombre de mines présentes sur les cases avoisinantes (voir schéma en annexes).
- A chaque tour, le joueur à la possibilité de marquer des cases pour signaler qu’elles sont minées.
- A chaque tour, le joueur doit obligatoirement se déplacer d’une case selon l’axe horizontal ou vertical. Attention, le joueur ne peut pas revenir sur ses pas.
- Si le joueur pénètre dans une case occupée par une mine, il est tué.
- A chaque tour, il est indiqué au joueur s’il existe au moins un chemin reliant sa position à la case ”Arrivée”. Si ce n’est pas le cas, il lui sera alors donné la possibilité de relancer une partie ou de continuer celle en cours.

## 2.4 But du jeu

- Le joueur remporte la partie lorsqu’il a rejoint la case ”Arrivée”.
- Le joueur perd la partie s’il est tué.

# 3 Programmation et conception en Java

La conception de votre programme Java doit faire appel à l’héritage. Vous devrez notamment utiliser à bon escient une ou plusieurs classes abstraites ou interfaces.

Il est recommandé de séparer dans votre code ce qui relève du graphisme et des interactions avec la machine.

Cette séparation devra se faire au moins au niveau des méthodes.

Ce projet comporte notamment deux défis à relever : d’une part comment placer aléatoirement  $n$  mines sur une carte sans risque de boucle infinie, et d’autre part comment calculer la possibilité d’un chemin entre la position courante du joueur et la case ”Arrivée”.

## 4 Pour aller plus loin

Toute implémentation de fonctionnalités supplémentaires intéressantes sera prise en compte dans la notation selon sa difficulté d’implémentation. Voici quelques idées :

- incorporer un système de ”vies” qui permet au joueur de ne pas perdre immédiatement la partie quand il marche sur une mine.

- incorporer des obstacles (murs etc..) à la carte qui bloquent les déplacements du personnage
- rendre le jeu jouable à la souris
- incorporer un chronomètre
- utilisation d'une bibliothèque graphique plus avancée (exemple : swing, SDLjava)
- système de gestion de fichiers : possibilité de sauvegarder, enregistrement des 10 meilleurs scores,...

## 5 La classe Deug

Pour réaliser ce projet, nous aurons besoin de la classe `Deug`. Cette classe contient des fonctions d'affichage graphique. Vous pouvez la télécharger depuis <http://www.lix.polytechnique.fr/~bossiere/enseignement.html>

Une documentation de la classe ainsi qu'un exemple de programme l'utilisant sont également fournis.

## 6 Evaluation

### 6.1 Première évaluation : conception

**Coefficient 2.** Vous devrez rendre sous la forme d'un document d'une ou deux pages l'architecture envisagée pour le projet accompagnée de justifications. Celle-ci pourra être présentée sous la forme d'un schéma UML (vous trouverez un très bon tutoriel sur <http://www.siteduzero.com/tutoriel-3-10363-modeliser-ses-objets-grace-a-uml.html>).

### 6.2 Seconde évaluation : soutenance

**Coefficient 5.** Vous devrez présenter seul ou en binôme votre projet pendant 10 minutes (dans le cas d'un binôme, 5 minutes chacun) : vous décrierez notamment les classes et méthodes principales utilisées. Vous exposerez les points forts et les points faibles de votre projet, ainsi que les difficultés rencontrées. Vous conclurez votre présentation par une démonstration de votre programme.

Cette présentation sera suivie de 5 minutes de questions et tests de votre programme.

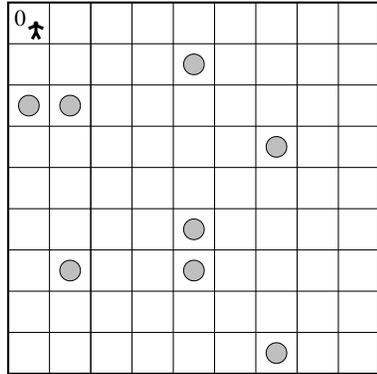


FIG. 1 – Début de partie, les mines sont indiquées par des cercles grisés, elles ne doivent apparaître que dans la version de contrôle.

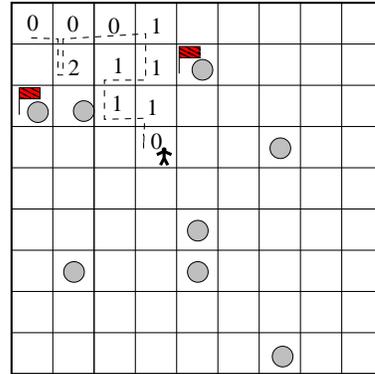


FIG. 2 – Partie en cours, ici on indique le trajet qu'a fait le joueur, ce trajet ne devra pas apparaître.

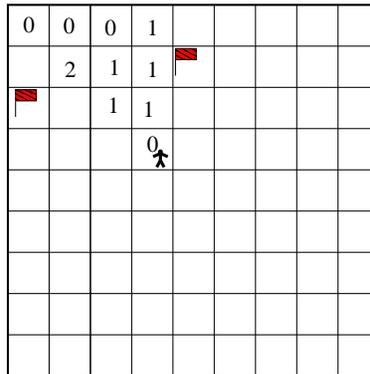


FIG. 3 – La même partie telle qu'elle devra apparaître en version de jeu normal.