

TD5 : Le « Démineur »

Ce TD a pour objectif de vous faire mettre en oeuvre les différentes nouvelles notions (JavaBean, Observateur-Observé, MVC, ...) abordées dans ce cours. Ce travail, ainsi qu'un autre que nous aborderons ultérieurement, sera à présenter pour votre examen.

La lecture du « Guide du graphiste, Graphics2D » peut-être bénéfique pour les détails ... graphiques.

1 LE PROJET: UN JEU DU DÉMINEUR

Nous allons réaliser l'interface graphique du jeu du **Démineur** tel que nous le connaissons. Si tel n'est pas le cas, les règles du jeu du Démineur sont expliquées sur [Wikipédia!](http://fr.wikipedia.org/wiki/D%C3%A9mineur), nous vous invitons à les lire.

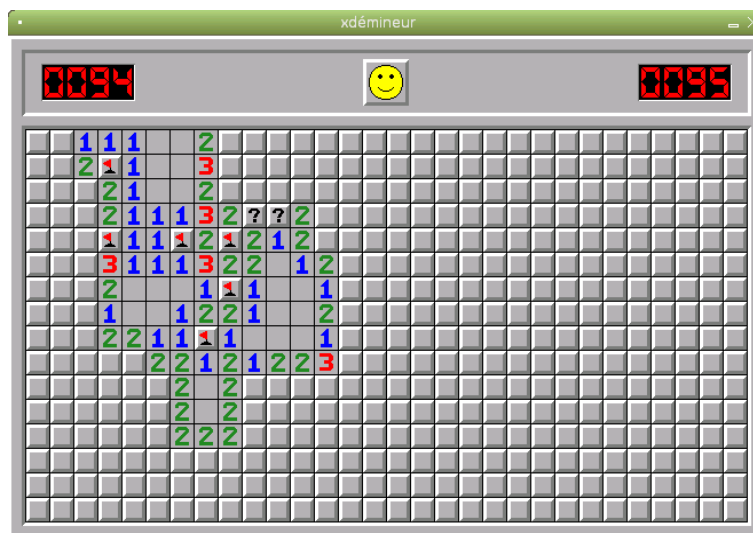


Figure 1 : Exemple de démineur (xdemineur)

Vous devrez prévoir une vue permettant de jouer.

Le modèle, sous la forme d'un *jar*, ainsi que sa javadoc, vous est fourni en annexe. Il vous suffira d'instancier un objet `Démineur` comme modèle. Ce modèle vous permet d'obtenir l'état de chacune des cases du démineur (les mines), il vous permet de marquer une case comme mine (supposée) et de marquer une case comme étant soupçonnée d'être une mine, ...

1 Si le lien n'est pas cliquable, [http://fr.wikipedia.org/wiki/Démineur](http://fr.wikipedia.org/wiki/D%C3%A9mineur) (jeu)

2 JAVABEAN, CASEDEMINEUR

Vous aurez à concevoir un composant graphique `CaseDemineur` représentant une case du jeu du démineur. Une case de démineur peut être dans un des états suivants (ces états sont proposés par le modèle):

1. CACHÉE : case non dévoilée et sur laquelle aucun soupçon ne plane, c'est l'état de départ,
2. MARQUÉE : case marquée comme étant une mine par le joueur (se représente généralement affublée d'un drapeau),
3. SOUPCONNÉE : case sur laquelle un soupçon plane (peut se représenter marquée par un point d'interrogation),
4. VISIBLE : case découverte, par défaut le nombre de mines voisines est affiché,
5. EXPLOSÉE : case représentant une mine explosée, **état terminal**²,
6. MINE_VISIBLE: case représentant une mine non explosée³, **état terminal**,
7. EN_ERREUR : case non explosive ayant été marquée comme mine par le joueur, **état terminal**

Un clic gauche sur une `CaseDemineur` la fera passer de l'état *caché* à l'état *visible* ou *explosée* (cette transition n'arrive qu'une fois) tandis qu'un clic droit la fera cycliser des états: *cachée* -> *marquée* -> *soupçonnée*.

Vous testerez votre bean en créant un `JFrame` à l'allure suivante⁴

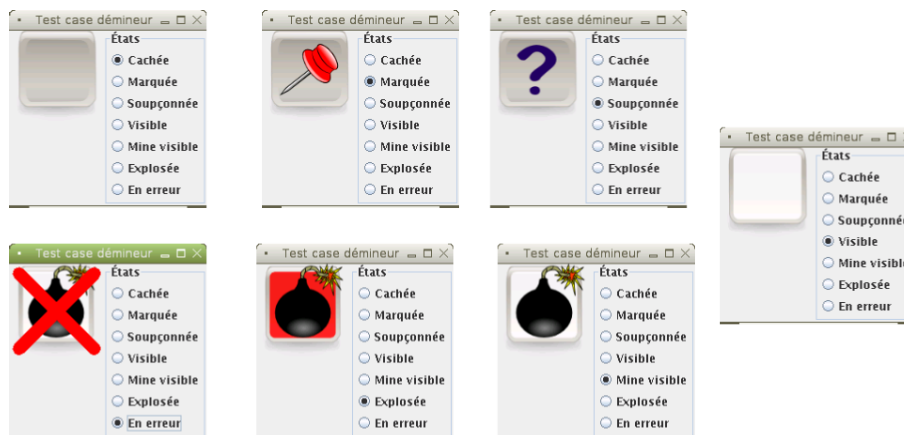


Figure 2 : Différents screenshots de la frame de test

Vous êtes bien entendu libre de choisir vos images comme bon vous semble. Soyez cependant attentif au copyright des images⁵.

Remarque⁶: Pour inclure vos images dans votre projet (et plus particulièrement dans votre `jar` .. afin qu'elles soient diffusées avec celui-ci) créez un package dans votre projet intitulé `images` (par exemple) et ajoutez-y une classe vide intitulée `Images` (par exemple).

2 Un **état terminal** est un état qui n'est atteint que pour l'affichage du jeu ... après qu'il soit fini.

3 Les perceptions du jeu peuvent ici diverger. L'un pourrait dire qu'une seule mine explose et que l'on montre alors l'emplacement des autres mines qui n'ont pas explosés. L'autre pourrait dire que dès qu'une mine explose toutes les autres explosent et dans ce cas il n'y a pas de mines non explosées ... les deux manières de voir les choses sont justifiables mais en général le fond est rouge pour la mine explosée.

4 Il doit être clair que nous vous montrons 6 états différents du même `JFrame`.

5 Les miennes proviennent du package `openclipart`.

6 Merci à `sun` et `adt` pour l'astuce.

Ajoutez vos images dans le répertoire du package (probablement src/images) et créez vos images avec des instructions de la forme:

```
private final URL URL_IMG_BUTTON =  
    Images.class.getResource("case_vide.png") ;  
private final Image IMG_BUTTON =  
    getToolkit().createImage(URL_IMG_BUTTON);
```

3 LA VUE

Concevez une vue permettant de jouer le plus simplement possible à ce jeu.

Vous prévoyez les options suivantes dans un menu

- à tout moment le joueur doit pouvoir recommencer une partie,
- il doit lui être possible de choisir le nombre de lignes et de colonnes du jeu.

Pour ceux qui veulent aller plus loin, vous pouvez envisager les options suivantes :

1. Permettre de choisir la taille des cases (petites, normales, grandes)
2. Permettre de choisir différentes représentations des états des cases (images).
3. Permettre de découvrir toutes les cases qui se trouvent autour d'une case découverte.
4. ...