

## Fiche TP numéro 7

### Le démineur

Pour ceux dont le fonctionnement de la classe `Demineur` n'est pas satisfaisant, un fichier `demineur.py` est disponible sur Moodle.

Dans ce TP, nous allons maintenant réaliser complètement le jeu du démineur, en créant une interface graphique et en faisant la liaison avec le jeu que vous déjà avez développé.

**Exercice 1 :** On complète d'abord un peu le démineur déjà développé (le modèle).

- Ajoutez à la classe `Case` la méthode `valeur(self)` qui retourne la valeur contenue dans la case ;
- Ajoutez à la classe `Demineur` les méthodes suivantes :
  - la méthode `valeur(self, l, c)` qui retourne la valeur de la case du démineur aux coordonnées  $(l, c)$  ;
  - la méthode `lig(self)` qui retourne le nombre de lignes du plateau du démineur ;
  - la méthode `col(self)` qui retourne le nombre de colonnes du plateau du démineur ;
  - la méthode `est_cache(self, l, c)` qui indique si la case aux coordonnées  $(l, c)$  du démineur est cachée ou pas.

**Exercice 2 :** Écrire une classe `VueDemineur` pour le jeu du démineur. Elle doit posséder :

- un constructeur, qui prend en paramètre une instance de la classe `Demineur` (le modèle que vous avez déjà écrit - n'oubliez pas de faire un `import` de votre module). L'interface graphique doit être construite de telle sorte :
  - qu'elle affiche `lig × col` images placées sur des boutons (`lig` et `col` étant les dimensions du `Demineur`). Tous les boutons affichent l'image `cache.gif`
  - qu'elle conserve dans une liste les images du démineur (le premier élément de la liste, à l'indice 0, doit être l'image `rien.gif`, le dernier élément de la liste doit être l'image `mine.gif`). `cache.gif` doit être mémorisée à part
  - qu'il y ait un bouton pour quitter le jeu et un bouton pour recommencer une partie.
- Seul le bouton pour quitter l'application sera associé à une action dans un premier temps.
- le modèle (l'instance du `Demineur`), la liste des boutons qui affichent les images, la liste des images, l'image `cache.gif` seront conservés dans des attributs.



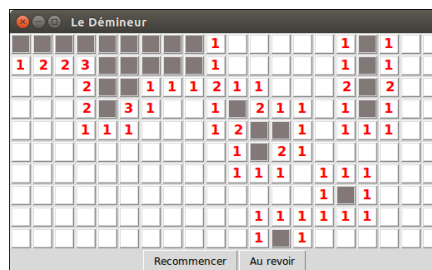
**Exercice 3 : Le contrôleur de réinitialisation.** Spécifiez puis écrivez la méthode `ctrl_reinit(self)` qui est associée au bouton *Recommencer*. Cette méthode demande au modèle de se réinitialiser puis demande à la vue de se remettre complètement à jour.

Testez votre application.

**Exercice 4 :** Spécifiez puis écrivez la méthode `demarre(self)` qui fait un appel à `ctrl_reinit` puis lance la boucle d'écoute des événements.

**Exercice 5 : Le contrôleur de découverte des cases.** Spécifiez puis écrivez la méthode `creer_ctrl_montre_case(self, lig, col)` qui retourne une fonction. La fonction retournée (que vous devrez spécifier) s'appelle `ctrl_montre_case()` et demande au modèle de montrer la case `(lig, col)` (uniquement si celle-ci n'est pas cachée). Si la découverte a pu avoir lieu, alors la fonction demande à la vue de se mettre à jour :

- tous les boutons doivent être redessinés : vous créez une nouvelle méthode dans `VueDemineur` `redessine(self)` qui interroge le `Demineur` pour savoir quelle image afficher sur chaque bouton : `cache.gif` ou une image correspondant à la valeur de la case
- si la partie est finie, alors un message au vainqueur est affiché
- si une bombe a été découverte, alors c'est un autre message qui est affiché.



C'est dans la méthode `ctrl_reinit` précédemment écrite que vous lierez chaque bouton du plateau à son contrôleur `ctrl_montre_case`.

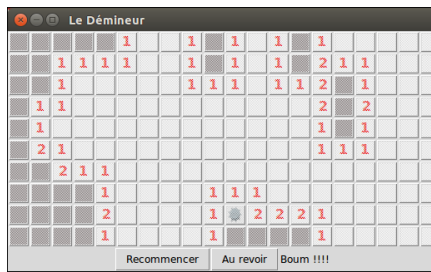
Testez votre application.

**Exercice 6 :** Pour peaufiner : écrivez une méthode `disable_buttons` qui met les boutons du plateau de jeu dans un état disable (regardez les exemples du cours). Cette méthode doit être appelée à bon escient dans le contrôleur `ctrl_montre_case`.



**Exercice 7 : Bonus pour peaufiner...** Récupérez l'image du drapeau sur Moodle... Pour permettre à l'utilisateur de marquer les endroits qui doivent contenir une mine, vous devez :

- enregistrer l'image du drapeau dans un attribut spécifique de la classe `VueDemineur` (comme `cache.gif`)



- ajouter un attribut booléen `__drapeau` à la classe `Case`, et les méthodes :
  - `est_drapeau(self)` qui retourne `__drapeau`
  - `pose_drapeau(self)` qui pose `__drapeau` à vrai (uniquement si la case est cachée)
  - `enleve_drapeau(self)` qui pose `__drapeau` à faux
- ajouter les méthodes suivantes à la classe `Demineur` :
  - `est_drapeau(self, l, c)` qui retourne la valeur du `__drapeau` de la case aux coordonnées `(l, c)`
  - `pose_drapeau(self)` qui pose `__drapeau` à vrai (uniquement si la case est cachée) sur la case aux coordonnées `(l, c)`
  - `enleve_drapeau(self)` qui pose `__drapeau` à faux sur la case aux coordonnées `(l, c)`
- lier un click droit (événement `<3>`) sur chaque bouton du plateau de jeu à une méthode `drapeau` qui pose ou qui enlève un drapeau sur une case cachée
- ne pas oublier de mettre à jour la méthode `redessine` !

