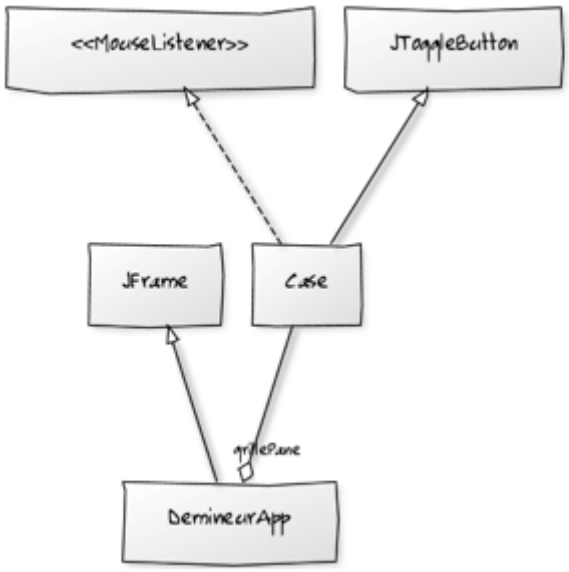
TP JAVA : Demineur

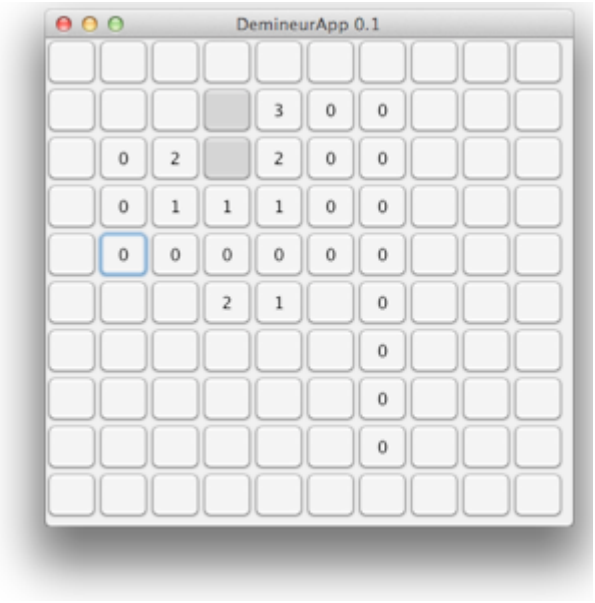
Notions : Définition de classes, Swing

Nous voulons réaliser un démineur en Java. Nous vous proposons de découvrir une autre type de bouton, le `JToggleButton`. La particularité de ce bouton est de posséder un état : appuyé ou non.

Le principe du jeu est très simple : on propose à une personne de découvrir petit à petit une zone "minée" en signalant à chaque case, le nombre de mines adjacentes. Il faut découvrir toute la zone sans tomber sur une mine. Pour une plus grande sécurité, on peut en général marquer la zone où l'on est persuadé qu'il y a une mine.

Voici l'arborescence que nous vous proposons ainsi que le résultat que l'on cherche à obtenir :





1. Classe principale

Nous allons tout d'abord nous intéresser à la classe `DemineurApp`. Cette classe possède une *grille* de jeu carrée de *taille* donnée. Les éléments de la grille de type entier et auront la signification suivante : -1 pour une mine et le nombre de mines adjacentes à la case sinon.

Ecrire le constructeur de la classe qui initalisera la grille (taille et nombre de mines donnés en paramètres), placera aléatoirement les mines et calculera le nombre de mines adjacentes à chaque case.

Proposer une méthode `main()` qui permettra de lire les arguments en ligne de commande et de proposer des valeurs par défaut le cas échant (10 mines pour une grille 10x10).

2. Interface graphique

L'interface graphique utilisateur est là encore très simple : nous allons utiliser les fameux boutons à état.

Je vous propose de créer votre propre classe spécialisée de `JToggleButton` : nommons la `Case`. Ce bouton aura pour particularité de connaître la valeur de la grille associée. Au départ, rien n'est affiché !

Pour placer les éléments, il faudra le bon `Layout` ! Le GridLayout semble tout adapté, mais vous pouvez vous contenter d'un layout null.

Si vous le désirez, vous pouvez mettre une barre de menu pour "Recommencer" / "Quitter" ou simplement afficher une boîte de dialogue d'aide (on peut même afficher une page au format HTML) et une barre de statut pour afficher le nombre de cases découvertes...

3. Moteur de jeu

3.1. Interaction utilisateur

Un `JToggleButton` est un `JButton` qui a deux états (repérés visuellement par un changement de couleur et un attribut associé `selected`) ; le bouton passe d'un état à l'autre en un simple clic.

Je vous propose de coder le schéma suivant :

- L'utilisateur clique avec le bouton gauche sur une case : cela affiche la case (le nombre de bombes adjacentes) ou cela entraine la fin du jeu
- L'utilisateur clique sur le bouton droit sur une case : la case reste sélectionnée(gris foncé) et ne peut faire exploser la grille, cela peut également entrainer la fin du jeu si l'utilisateur a trouvé toutes les bombes sans en faire exploser une seule.

3.2. Fin du jeu

Pour détecter la fin du jeu, on peut doter la classe `Case` d'un attribut de classe « restant » qui indique le nombre de cases qu'il reste à marquer. Quand ce nombre est à 0, le jeu est terminé (gagnant).

Pour indiquer la fin du jeu, "Gagné" ou "Perdu", on propose d'utiliser une boite de dialogue de `JOptionPane`.