

Module 3 - Aide à la Réussite : EP 2 Soutien en Programmation

Travaux Dirigés (10), Licence 1ère Année

Swing

Le but de ce TP est de programmer (une partie d') un petit jeu avec l'interface Swing.

Le jeu consiste à afficher un damier, tel celui de la figure ci-dessous, composé de 100 boîtes. Chaque boîte peut contenir une « bombe ». Le joueur clique sur les boîtes en espérant ne pas tomber sur celles contenant une « bombe », sinon ces dernières « explosent ». Le joueur gagne s'il a cliqué sur 20 boîtes et que moins de 5 ont explosé.



Mise en œuvre de la solution

1. Déclarez une classe **Boite** qui dérive de la classe **JButton** comprenant :
 - Un attribut privé **couleur** de la classe **Color**, représentant la couleur de la boîte,
 - Un attribut privé **bombe** de type **boolean**, qui indique si la boîte renferme une « bombe »,
 - Un attribut privé **ouverte** de type **boolean**, qui indique si la boîte a déjà été choisie ou pas,
 - Les getters et setters des attributs ci-dessus,
 - Une méthode **Color choixCouleur()** qui retourne une couleur choisie aléatoirement entre {**BLUE, CYAN, GREEN, MAGENTA, ORANGE, RED**}.

- Un constructeur **Boite(String n)** qui accepte en paramètre une chaîne de caractère correspondant au numéro de la boîte, qui :
 - Appelle le constructeur de la classe mère,
 - Initialise l'attribut **couleur** (appel à **choixCouleur()**),
 - Initialise la couleur de fond de l'objet **Boite** créé (**this.setBackground(couleur);**),
 - Tire un nombre aléatoire entre **0** et **99** et initialise l'attribut **bombe** à **true** si ce nombre est strictement inférieur à **20** et à **false** sinon,
 - Initialise l'attribut **ouverte** à **false**.

2. Déclarez une classe **ChampsDeMines** qui dérive de la classe **JFrame** comprenant :

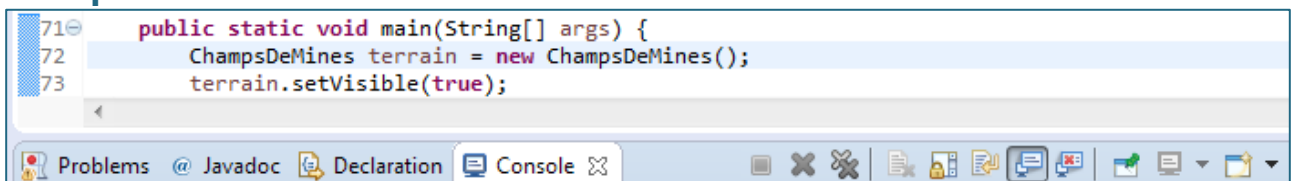
- Un attribut entier **compteur**, qui servira à numéroter les boîtes (**JButton**) à créer,
- Un attribut **boites**, tableau de deux dimensions (**10 x 10**) d'éléments **Boite**,
- Un attribut entier **valCliques**, initialisé à **0**, qui va comptabiliser le nombre de boîtes choisies (cliquées),
- Un attribut entier **valBombes**, initialisé à **0**, qui va comptabiliser le nombre de boîtes choisies (cliquées) contenant une bombe,
- Un composant **cliques** (**JLabel**) de label ("Nombre Boîtes cliquées : " + **valCliques**),
- Un composant **bombes** (**JLabel**) de label ("Nombres bombes explosées : " + **valBombes**),
- Un constructeur **ChampsDeMines()** qui :
 - Appelle le constructeur de la classe mère avec le texte "Champs de mines" comme paramètre,
 - Indique la libération des ressources lorsque l'objet **ChampsDeMines** créé est fermé (**this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);**),
 - Définit la taille de l'objet **ChampsDeMines** créé à **700 x 700** pixels,
 - Place l'objet **ChampsDeMines** créé au milieu du bureau (**this.setLocationRelativeTo(null);**),
 - Définit une stratégie de placement **BorderLayout** (**this.setLayout(new BorderLayout());**)
 - Crée un objet **score** de la classe **JPanel** qui contiendra les attributs **cliques** et **bombes**,
 - Place l'objet **score** dans la zone Nord de l'objet **ChampsDeMines** créé,
 - Crée un objet **zone** de la classe **JPanel**,
 - Définit une stratégie **GridLayout** de 10 lignes x 10 colonnes pour l'objet **zone** créé,
 - Créez **100** objets **Boite** avec chacun un numéro différent (convertir le numéro en **String**) et ajoutez-les dans le container **zone**,

Écrire une méthode **main()** dans laquelle vous créerez un objet **ChampsDeMines** que vous rendrez visible. Exécutez votre **main()**.

Exemple

```

71 public static void main(String[] args) {
72     ChampsDeMines terrain = new ChampsDeMines();
73     terrain.setVisible(true);
  
```



3. Nous allons rendre les objets **Boite** réactifs aux actions de la souris.
 - Indiquez que la classe **Boite** implémente l'interface **MouseListener**,
(**public class Boite extends JButton implements MouseListener**)
 - Indiquez dans le constructeur qu'un objet **Boite** s'écoute lui-même
(**this.addMouseListener(this);**)
 - Indiquez dans la méthode **void mouseEntered(MouseEvent e)** qu'un objet doit changer sa couleur de fond en noir lorsque la souris le survole
(**this.setBackground(Color.BLACK);**)
 - Indiquez dans la méthode **void mouseExited(MouseEvent e)** qu'un objet doit recouvrir sa couleur de fond initiale lorsque la souris ne le survole plus
(**this.setBackground(couleur);**)
 -

Relancez l'exécution et vérifiez que les boites réagissent bien comme attendu.

4. Nous allons maintenant rendre le damier réactif aux cliques de la souris sur les boites qu'il contient et changer son apparence en conséquence.
 - Indiquez que la classe **ChampsDeMines** implémente l'interface **ActionListener**,
(**public class ChampsDeMines extends JFrame implements ActionListener**)
 - Dans le constructeur, après la création des boites, indiquez que chacune sera "écoutée" par l'objet **ChampsDeMines** créé
(**boites[i][j].addActionListener(this);**)
 - Implémentez la méthode **void actionPerformed(ActionEvent e)** de façon que :
 - Si le joueur clique sur une boîte et que celle-ci ne renferme pas de bombe (attribut **bombe = false**), alors son label est supprimé et sa couleur de fond devient blanc. Son attribut **ouverte** passe à **true**,
 - Si le joueur clique sur une boîte et que celle-ci renferme une bombe (attribut **bombe = true**), alors son label est supprimé et l'icône "Boum.jpg" lui est attribuée
(**Imagelcon boum = new Imagelcon("Boum.jpg");**)
(**b.setlcon(boum);**)
Note : l'icône "Boum.jpg" est disponible sur Celene.
 - Si le joueur clique sur une boîte déjà ouverte, cela ne devrait avoir aucun effet,
 - Si le joueur clique sur une boîte non ouverte, l'attribut **valCliques** est incrémenté et le label du composant **cliques** est mis à jour
(**cliques.setText("Nombre Boîtes cliquées : " + valCliques);**)
 - Si le joueur clique sur une boîte non ouverte, et qu'elle renferme une bombe, l'attribut **valBombes** est incrémenté et le label du composant **bombes** est mis à jour,
(**bombes.setText("Nombres bombes explosées : " + valBombes);**)

Relancez l'exécution et vérifiez que les boites et les labels réagissent bien comme attendu.

Vous pouvez ajouter d'autres effets lorsque le joueur gagne ou perd (modification du visuel du damier).