

## Démineur en mode graphique (2/2)



### Introduction

Ce TP a pour objectif de « donner vie » à l'interface graphique en manipulant les événements/écouteurs AWT/Swing.

Vous serez évalués sur votre capacité à :

- développer des composants et/ou des traitements réutilisables,
- réduire les dépendances entre les différentes parties de votre application.

Quelques pistes pour y parvenir :

- Identifier les informations minimales devant transiter dans votre application, les émetteurs et les récepteurs (qui détient l'information ? qui en a besoin ? sous quelle forme ?)
- Il est tout à fait possible (et recommandé !) de transformer des événements de « bas niveau » émis par les composants graphiques en événements de « haut niveau » ayant un sens dans votre application.

### 1- Contrôleur graphique

- Permettre au joueur d'agir sur une cellule avec la souris :
  - Un clic gauche permet de dévoiler la cellule
  - Un clic droit permet de changer le marquage de la cellule selon le cycle :  
aucun marquage → marquage « mine » → marquage « indécis » → aucun marquage.
- On reprendra les conventions de la vue texte, sauf pour l'absence de marquage où l'on ne mettra aucun texte (au lieu de « # »)

### *Avant de continuer...*

Assurez-vous que les actions de l'utilisateur sont répercutées sur toutes les vues : les vues texte et graphique sur le nombre de mines et la vue texte sur la grille.

### 2- Panneau de configuration d'une partie personnalisée

- Permettre au joueur de définir le nombre de lignes/colonnes et le pourcentage de mines :
  - soit en déplaçant le curseur des JSliders
  - soit en entrant du texte dans les JTextField et en validant par entrée
- Pour chaque caractéristique du jeu, les valeurs représentées par le JSlider et le JTextField doivent être en permanence **cohérentes**.
- Quand le joueur change le nombre de lignes L ou le nombre de colonnes C, la borne supérieure pour le nombre de mines doit être actualisée dynamiquement pour être toujours égale à  $0.85 \times L \times C$
- Vous devez gérer le cas d'une saisie incorrecte dans un JTextField, mais la manière de le faire est laissée à votre appréciation. Elle peut différer selon le type d'erreur (valeur d'un mauvais type, valeur du bon type mais hors intervalle autorisé, etc.)

### Avant de continuer...

Êtes-vous capable d'ajouter facilement (en une ligne de code!) un troisième composant graphique (par exemple un JLabel) représentant une caractéristique du jeu ?

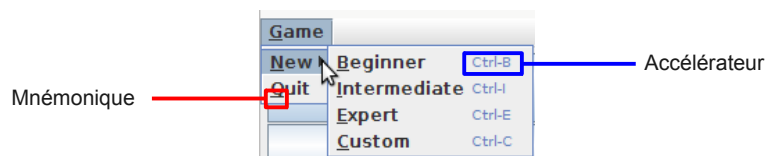
## 3- Panneau de création d'une partie

- Utiliser les événements/écouteurs AWT/Swing nécessaires pour repérer quel bouton radio est sélectionné et effectuer les traitements associés.
- Ajouter un bouton de validation au panneau de contrôle. Lorsque celui-ci est actionné, une nouvelle partie démarre avec les caractéristiques définies par le joueur.

## 4- Menu

Faire en sorte que les items du menu deviennent actifs :

- Si le joueur actionne :
  - « New » : une fenêtre contenant un panneau de création d'une partie apparaît
  - « Beginner », « Intermediate », « Expert » : une partie commence avec des paramètres prédéfinis
  - « Custom » : une fenêtre contenant un panneau de configuration d'une partie personnalisée apparaît
  - « Quit » : l'application prend fin (même effet que l'appui sur la croix de la fenêtre)
- Le joueur peut actionner un item :
  - par un clic
  - par l'utilisation des mnémoniques et accélérateurs illustrés ci-dessous :



## Conseils

- Attention à la redondance de code dans la construction du menu et la gestion des événements associés !
- Pour vérifier le faible couplage des différentes parties de votre application, vous pouvez développer un « initialiseur » de partie en mode console et vous assurer que cela n'a pas d'impact dans le reste du code.