

Jeu de jetons

Note : Seulement les noms de méthodes sans leur paramètre seront utilisés pour l'explication du concept.

Conteneur principal et sous conteneur

Cette interface graphique utilise un `JDesktopPane(mainFrame)` comme conteneur principal. Ce conteneur contient deux `JInternalFrame` soit :

- 1) Pour afficher le sommaire de l'aide. (`resumeWindow`)
- 2) Pour afficher la fenêtre d'ouverture ou d'enregistrement de fichiers. (`fileChooser`)

Panneau de dessin

D'autre part, un panneau (`gameWindow`) est également intégré au `JDesktopPane` afin d'obtenir un espace de dessin. Le contexte graphique sera généré par un objet (`openGame`) permettant de dessiner la grille du jeu (`buildGrill`) et tous les jetons à l'intérieur de ce panneau (`paintComponent`). Cet objet permettra de contrôler l'issue de la partie à l'aide de plusieurs méthodes :

SetXYJetonLocation :

Permet d'insérer et de dessiner les jetons dans le contexte graphique en appelant la méthode `repaint` qui à son tour appelle la méthode `paintComponent`.

Permet également de compter le nombre de jetons des joueurs et de vérifier l'état de la partie en appelant la méthode `verifyStateGame`.

Le principe utilisé pour savoir si il y a un gagnant repose sur les quatre méthodes de recherche suivantes :

SearchRightPosition : Recherche à droite du premier jeton rencontré.

SearchRightBottomPosition : Recherche en diagonale vers le bas et à droite du premier jeton rencontré.

SearchBottomPosition : Recherche en dessous du premier jeton rencontré.

SearchLeftBottomPosition : Recherche en diagonale vers le bas et à gauche du premier jeton rencontré.

Fenêtres de messages d'erreurs

L'utilisation de `JOptionPane.showMessageDialog` à servie pour l'affichage les différents messages d'erreurs possibles et pour le sous menu d'aide À propos.

Principe de l'Interface Graphique

Presque toutes les composantes utilisées dans cette application sont reliées à un gestionnaire d'événement, qui à leur tour appelle les méthodes qui exécutent le code nécessaire pour chacune des actions à effectuer. Cependant, les composantes qui utilisent des options à multiples choix comme les boutons radio et les listes utilisent des gestionnaires définies par une classe qui implémente ActionListener. Le cœur de cette application est contrôlé par quatre méthodes soit :

newGame_actionPerformed :

Cette méthode démarre la partie en créant un objet **openGame** qui permet de construire un contexte graphique (Construit la grille et ces bordures), d'initialiser beaucoup de paramètres à l'aide de l'appel à la méthode **resetParametre**, et de supprimer les composantes reliées au contexte graphique de son conteneur afin de pouvoir réinitialiser le tout au moment du démarrage d'une nouvelle partie etc.

validPlayer1JButton_actionPerformed et validPlayer2JButton_actionPerformed :

Ces deux méthodes contrôlent l'insertion de jetons et font appel à la méthode **gameControler** pour la gestion des minuteries, des coordonnées X, Y, du nombre de jetons joués par le joueur, des différents affichages etc. et de vérifier s'il y a un gagnant. Par conséquent, presque tout est relié à ces deux méthodes afin de contrôler l'historique du jeu.