

Projet rapport intermédiaire

Constructeur LeJeuLoto

Dans cette partie on s'intéresse à comment l'application « LeJeuLoto » gère les joueurs. La classe « Joueur » est créée pour représenter un individu possédant ces caractéristiques. Ensuite la classe « LesJoueurs » a été construite pour mettre ensemble tous ces joueurs dans une liste de joueur.

La classe « LeJeuLoto » est la classe principale de notre programme.

Dans ce jeu on veut ajouter des joueurs, il nous faut donc une liste de joueur. Pour cela on déclare donc un attribut `lstJ` de type « LesJoueurs ».

La méthode **`iniComponents()`** construit notre interface.

D'autres attributs seront rajoutés au fur et à mesure comme les options de jeu, les tailles des cartes...

```
public class LeJeuLoto extends javax.swing.JFrame {  
  
    private LesJoueurs lstJ;  
  
    public LeJeuLoto(){  
        initComponents();  
        this.lstJ = new LesJoueurs();  
    }  
}
```

AjoutJoueurDlg

Le menu « Ajout » comporte une option « Ajouter un joueur » qui permet la saisie d'un nouveau joueur à l'aide d'une boîte de dialogue. Il est possible de saisir le pseudo et de choisir une couleur, le solde du joueur étant fixé à 20.0€ par défaut et est non modifiable.

Le but de cette boîte de dialogue est d'entrer un nouveau joueur au début de la partie de notre jeu du loto. Ce nouveau joueur aura un pseudo et une couleur choisi par l'utilisateur, de plus le solde du joueur est fixé par les règles du jeu à 20 euros en début de partie.

Pour d'écrire les actions : quand l'utilisateur va vouloir ajouter un nouveau joueur il va aller dans le menu "Ajout" de notre application principale « LeJeuLoto » et sélectionner "Ajout Joueur". Ensuite une nouvelle fenêtre s'ouvrira et à ce moment il pourra compléter les différents champs nécessaires à la création d'un joueur, sauf pour la photo qui est ajoutée par défaut par le programme et le solde (en somme il ne rentrera que le pseudo et pour choisir

une couleur grâce à une palette de couleur). Ensuite pour valider sa sélection l'utilisateur doit cliquer sur le bouton "Valider". Ensuite l'utilisateur voit son ajout sur les boutons de joueur avec le pseudo et la couleur qui l'a choisi pour son joueur.

⇒ Pour cela il nous faut créer une boîte de dialog `JDialog` que l'on nommera

« `AjoutJoueurDlg` » qui dérive de notre `JFrame` « `LeJeuLoto` ».

- Pour créer un joueur il nous faut aussi un objet de type `Joueur`, pour cela une classe « `Joueur` » a été créée en amont. Un `Joueur` se caractérise par : un pseudo, une couleur, un solde et une photo (pour le moment).

⇒ Lors du clic sur l'option "Ajouter" de « `LeJeuLoto` » on crée une instance de la boîte de dialogue d'ajout, on affiche cette boîte de dialogue. Lors de sa fermeture il y a récupération d'un indicateur pour savoir si la boîte a été fermée par le bouton « Valider » ou « Annuler ».

- Notre classe « `AjoutJoueurDlg` » dispose donc d'un attribut `ok` de type **Boolean**.

⇒ Si on clique sur « Valider », on récupère les informations sur le joueur que l'utilisateur a saisie, il y a alors création d'un objet « `Joueur` » qui sera stocké dans notre attribut `lstJ` de type **LesJoueurs** de la classe « `LeJeuLoto` ». Une condition est rajoutée pour que le nombre de joueurs ne dépasse pas quatre (donc si au dernier ajout la liste était de 3 on peut ajouter un joueur et donc à la fin avoir une liste de quatre joueurs, sinon si la liste était de quatre en ajoutant un nous fera une liste de cinq joueurs et ce n'est pas ce que l'on veut). On gère ceci avec la taille de notre liste de joueurs.

- Notre classe « `AjoutJoueurDlg` » dispose donc aussi d'un attribut `nj` de type **Joueur**
- Une méthode `BCouleurActionPerformed` est écrite pour que quand l'utilisateur clique sur le bouton "Choisir couleur" une boîte de dialogue s'ouvre (`JColorChooser`) et donne une couleur au joueur.
- Une méthode `BValiderActionPerformed` est écrite pour que quand l'utilisateur clique sur le bouton "Valider", notre attribut `nj` est les valeurs sélectionnées.
- Il nous faut donc un objet de type **LesJoueurs** dans la classe « `LeJeuLoto` » pour cela une classe « `LesJoueur` » a été créée en amont. C'est une liste des joueurs (que l'utilisateur a entrée).
-

⇒ Ensuite on cherche à 'voir' notre joueur sur notre interface principale « `LeJeuLoto` ».

⇒ On sait que l'on a une 'liste de joueur' contenu dans un objet « `LesJoueurs` », les joueurs ont donc un indice dans notre liste, grâce à cet indice on dira que le joueur ayant un indice '0' s'affichera sur le bouton 'Joueur 1' avec sa couleur, le joueur ayant un indice '1' s'affichera sur le bouton 'Joueur 2' avec sa couleur... ,

- Une méthode `ItemJoueurActionPerformed` est écrite dans notre classe « `LeJeuLoto` » pour que notre joueur entré soit ajouté à notre liste. On crée donc la boîte de dialogue pour ajouter le joueur, on met une condition pour voir si

cette boîte de dialogue est bien ouverte ou non et si la liste des joueurs est inférieure à quatre, on crée un joueur à partir de ce que la boîte de dialogue nous envoie, on l'ajoute à la liste.

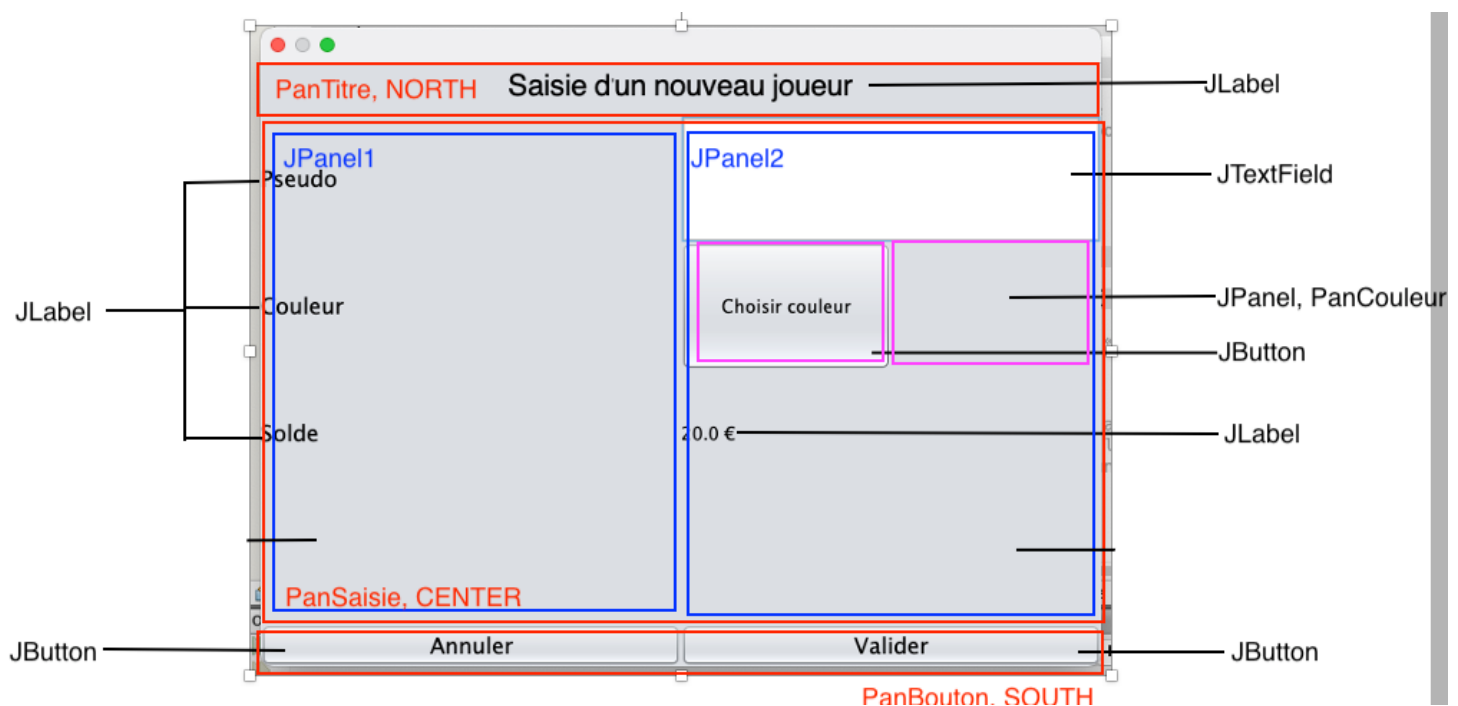
⇒ Ceci modifie donc le text de notre bouton avec le pseudo du joueur, par exemple "Joueur1" devient "Clothilde" puis sa couleur est également modifiée et donc le fond de couleur du bouton devient rose.

- Ensuite on fait un switch avec l'indice de notre joueur pour qu'il soit bien complété dans les boutons souhaités.

⇒ Si on clique sur le bouton « Annuler » il n'y a pas de traitement des informations et donc aucun objet n'est créé.

- Une méthode `BAnnulerActionPerformed` est écrite pour que quand l'utilisateur clique sur le bouton "Annuler", aucun objet de type **Joueur** ne soit créé et que la fenêtre se ferme.

→ Arborescence de l'application



AjoutJoueurDlg `JDialog` (cette fenêtre est une boîte de dialogue, elle hérite du type `JDialog`)

BorderLayout

JPanel PanTitre, NORTH

FlowLayout

`JLabel` `LTitre` « Saisie d'un nouveau joueur »

JPanel PanSaisie, CENTER

GridLayout(1,2)

JPanel JPanel1

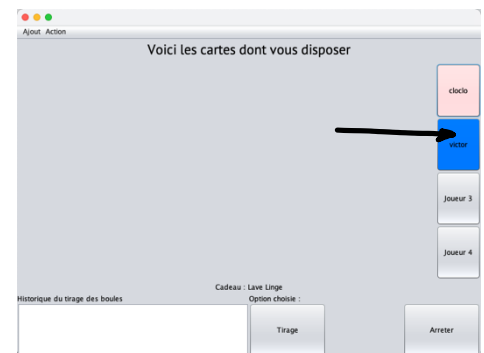
GridLayout(4,1)

```
JLabel LPseudo « Pseudo »
JLabel LCouleur « Couleur »
JLabel LSolde « Solde »
JPanel JPanel2
    GridLayout(4,1)
        JTextField Pseudo
        JPanel PanChoixCoul
            GridLayout(1,2)
                JButton BCouleur « Choisir couleur »
                JPanel PanCouleur
                    JLabel Solde « 20.0 € »
JPanel PanBouton, SOUTH
    GridLayout(1,2)
        JButton BAnnuler « Annuler »
        JButton BValider « Valider »
```

VisuJoueurDlg

Le but de cette boîte de dialogue est de visualiser les informations relatives à un joueur en cliquant sur le bouton correspondant à ce joueur dans notre application « LeJeuLoto ». On pourra donc connaître son pseudo, sa couleur, son solde et voir s’il a gagné un lot auparavant.

Pour d’écrire les actions : quand l’utilisateur va vouloir visualiser un des joueurs, il pourra cliquer sur le bouton qui est attribué au joueur. Une boîte de dialogue va alors s’ouvrir et affichée les informations relatives au joueur. L’utilisateur pourra fermer cette boîte de dialogue en appuyant sur le bouton ‘Fermé’ de cette dernière.



⇒ Pour cela il nous faut créer une boîte de dialog JDialog que l’on nommera « VisuJoueurDlg ».

⇒ Lors du clic sur le bouton de l’un des joueurs de « LeJeuLoto » on crée une instance de la boîte de dialogue de visualisation, on affiche cette boîte de dialogue.

⇒ On affiche les informations sur joueur.

- Notre classe dispose donc d’un attribut **lejoueur** de type **Joueur**.
- D’une méthode **initInfos()** qui affiche le pseudo, le solde... par le biais de set sur les différents labels, panneau et bouton de l’interface. Puis pour la zone d’édition, on la vide puis on l’a rempli du texte “Aucun lot gagné”. Puis on fait en sorte que l’utilisateur ne puisse pas modifier cette zone d’édition.

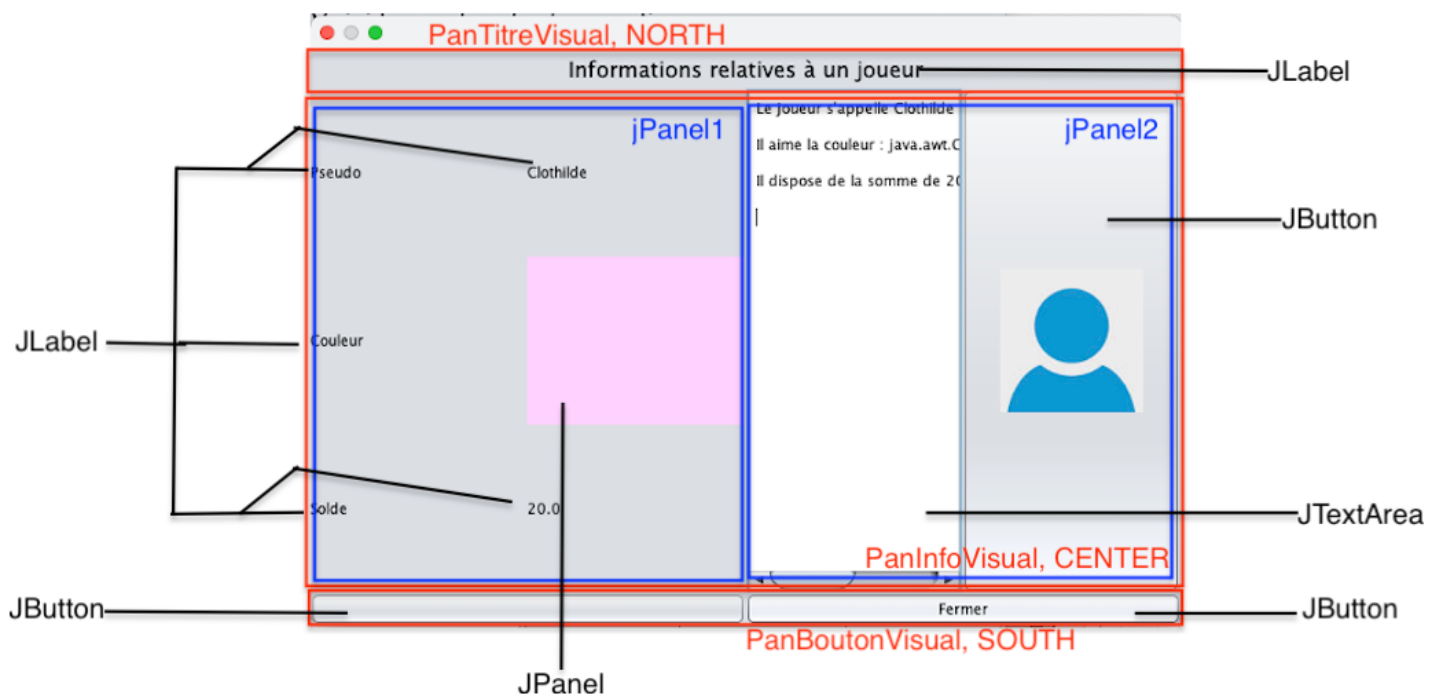
⇒ Lors du clic sur le bouton “Fermer” de la boîte de dialogue, celle-ci se ferme.

- Une méthode `BFermerActionPerformed` est écrite.

⇒ Dans l'application « LeJeuLoto », quand on clique sur un bouton de joueur on crée un objet `Joueur j` qui est par exemple pour le bouton du joueur 1, le joueur d'indice 0 de notre liste de joueur qui est dans notre attribut `lstJ`, ensuite on crée un objet `VisuJoueurDlg` auquel on passe en plus des paramètres habituels un paramètre `j` qui est notre joueur créé précédemment. Puis affiche la boîte de dialogue.

- Une méthode `BJoueur1ActionPerformed` est alors écrite pour le clic sur le bouton du joueur 1 par exemple.

→ Arborescence de l'application



VisuJoueurDlg JDialog (cette fenêtre est une boîte de dialogue, elle hérite du type JDialog)

BorderLayout

JPanel PanTitreVisual, NORTH

FlowLayout

JLabel LTitreVisual « Informations relatives à un joueur »

JPanel PanInfoVisual, CENTER

GridLayout(1,2)

JPanel jPanel1

GridLayout(3,2)

JLabel LPseudo « Pseudo »

JLabel Pseudo « »

JLabel LCouleur « Couleur »

JPanel PCouleur

```
JLabel LSolde « Solde »
JLabel Solde « »
JPanel jPanel2
GridLayout(1,2)
JScrollPane1 jScrollPane
JTextArea « Edition »
JButton Photo
JPanel PanBouton, SOUTH
GridLayout(1,2)
JButton jButton1
JButton BFermer « Fermer »
```

Pour conclure le projet sera considéré comme réussi si l'ensemble des actions décrites si dessus fonctionnent correctement.