

## Rapport intermédiaire n°2

Questions :

1.

Les attributs que j'ai déclarés dans la classe **JeuMemory** sont « *persos* » de type **LesPersonnages** et « *joueurs* » de type **LesJoueurs**. Nous aurons besoin pour pouvoir jouer, d'une liste de joueur donc stockée dans *joueurs*, et une liste de personnages stockée dans *persos*.

Le constructeur appelle la méthode  `initComponents()` qui construit l'interface de la JFrame. Ensuite on initialise *persos* en créant une nouvelle instance de **LesPersonnages** puis on initialise *joueurs* en créant une nouvelle instance de **LesJoueurs** afin de stocker les joueurs qui vont jouer, et les personnages des joueurs.

### 2. SaisieJoueurDlg

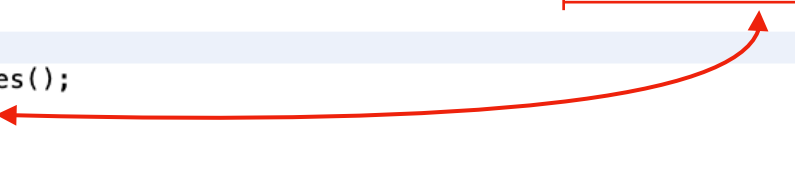
a.

La boîte de dialogue « SaisieJoueurDlg » permet à l'utilisateur d'ajouter un nouveau personnage à ceux déjà existant en personnalisant le pseudo, la famille préférée de personnages ainsi que la photo.

b.


Lors de la construction de cette boîte de dialogue, elle reçoit (via le constructeur) les paramètres **parent**, **modal** et **persos**. **parent** indique quelle est la fenêtre qui va gérer cette boîte de dialogue (en l'occurrence, la fenêtre d'où elle est appelée). **modal** détermine si la boîte de dialogue sera bloquante ou non, c'est à dire que la boîte de dialogue devra d'abord être fermée avant de pouvoir continuer sur la fenêtre précédente. Puis **persos** est l'ArrayList de personnage créée dans la fenêtre parent qui va être « transférée » vers la boîte de dialogue qui en a besoin pour que l'utilisateur choisisse sa famille préférée. Une fois cette boîte de dialogue créée, il faut définir sa visibilité sur « true ».

```
public SaisieJoueurDlg(java.awt.Frame parent, boolean modal, LesPersonnages lp) {  
    super(parent, modal);  
    initComponents();  
    lp = new LesPersonnages();  
    lp.ajoutePersos(lp);  
    initListeFamilles();  
}
```



Elle utilise un **boolean** *ok* qui va assurer que toutes les informations pour créer le nouveau personnage sont récupérées. Ensuite elle utilise **Imagelcon** *photo*, qui est facultative, et va donc servir d'avatar pour ce nouveau joueur. « ok » sera retourné à la fenêtre parent grâce à un assesseur en lecture « `isOk()` » qui retourne un booléen. Pour finir on utilise « *joueur* » de type **Joueur** qui va tout simplement référencé le joueur créé.

```
private void ajtJoueurActionPerformed(java.awt.event.ActionEvent evt) {  
    SaisieJoueurDlg diag = new SaisieJoueurDlg(this, true, persos);  
    diag.setVisible(true);  
    if(diag.isOk()) {  
        System.out.println(diag.getJoueur().getNom()+diag.getJoueur().getFamille());  
        joueurs.ajouteJoueur(diag.getJoueur());  
    }  
}
```



C.

**JFrame SaisieJoueurDlg BorderLayout**

**JPanel jPanelS GridLayout(1, 1) NORTH**

JButton jLabelTitre "Créez votre joueur !"

**JPanel jPanelNord GridLayout(1, 2) CENTER**

**JPanel jPanelGauche GridLayout(3, 1)**

**JPanel jPanelPseudo BorderLayout**

JLabel jLabelPseudo "Donnez votre pseudo : " NORTH

JTextField pseudo CENTER

**JPanel jPanelFamille BorderLayout**

JLabel jLabelFamille "Quelle est votre famille préférée de personnages ?" NORTH

JScrollPane jScrollPaneListFam CENTER

JList ListeFamilles

**JPanel jPanelPhoto BorderLayout**

JLabel jLabelPhoto "Choisissez une photo : " NORTH

JButton choixPhoto "Parcourir" CENTER

**JPanel jPanelDroite BorderLayout**

**JButton Photo CENTER**

**JPanel jPanelButton GridLayout(1, 2) SOUTH**

JButton Annuler "Annuler"

JButton Valider "Valider"

d.

La boîte de dialogue « SaisieJoueurDlg » gère trois événements qui sont tous des « ActionPerformed » sur des JButton.

Le premier événement est lorsque que l'utilisateur clique sur le JButton « choixPhoto » qui va lui permettre de choisir une image via un « JFileChooser ». L'image est choisie dans l'explorateur de fichier. On crée ensuite une ImageIcon, de l'image choisie, qui sera référencée dans la variable « photo ». Pour finir, on définit l'icône du bouton « Photo » par cette même image pour que l'utilisateur voit l'image qu'il a sélectionnée.

Le second événement correspond à un clique sur le bouton « Valider » qui active le processus de création du nouveau joueur. Dans un premier temps, on vérifie que l'utilisateur a bien renseigné un pseudo. Si oui, on récupère l'indice de l'élément sélectionné dans la JList puis on vérifie que cet indice est supérieur à 0, donc que l'utilisateur a bien choisi une famille. Si oui, on récupère l'objet sélectionné dans la JList (dans notre cas un String). Puis on crée un nouveau joueur qui aura pour paramètre le pseudo écrit et la famille sélectionnée par l'utilisateur. Pour terminer, on vérifie si une photo a été importée. Si oui, on définit la photo du joueur avec cette image. Le booléen « ok » peut être défini sur true.

Si l'utilisateur :

- n'écrit pas de pseudo, on le lui rappelle en écrivant « Vous devez saisir un pseudo ! » dans la JTextField « pseudo ».
- ne sélectionne pas de famille, on change la couleur et le style de la JLabel « jLabelPseudo » en rouge et en gras.
- n'importe pas de photo, le joueur est quand même créé puis on ferme la boîte de dialogue. (le booléen « ok » sera aussi défini sur true)

Le troisième et dernier événement satisfait à un clique sur le JButton « Annuler » qui va tout simplement fermer la boîte de dialogue en définissant au préalable le booléen « ok » sur false puis en rendant la boîte de dialogue invisible et libère la mémoire utilisée par celle-ci.

e.

Cette boîte de dialogue est appelée lorsque l'utilisateur clique sur le JMenuItem « ajtJoueur ». Dans l'événement, on crée une nouvelle instance de SaisieJoueurDlg que l'on appelle « **diag** » qui prend pour paramètre **this**, **true** et **persos** (ces paramètres sont expliqués à la question 2.b.). Une fois cette boîte de dialogue créée, il faut définir sa visibilité sur « true ».

### 3. VisuPersonnagesDlg


a.

La boîte de dialogue « VisuPersonnageDlg » va permettre à l'utilisateur de visualiser les personnages gagnés par le joueur courant ainsi que son score.

b.

Lors de sa création, elle reçoit via le constructeur les paramètres **parent**, **modal** et **j**. **parent** indique quelle est la fenêtre qui va gérer cette boîte de dialogue (en l'occurrence, la fenêtre d'où elle est appelée). **modal** détermine si la boîte de dialogue sera bloquante ou non, c'est à dire que la boîte de dialogue devra d'abord être fermée avant de pouvoir continuer sur la fenêtre précédente. Puis **j** référence le joueur courant qui est nécessaire pour pouvoir afficher les cartes de son paquet.

```
public VisuPersonnagesDlg(java.awt.Frame parent, boolean modal, Joueur j) {  
    super(parent, modal);  
    initComponents();  
    this.j = j;  
    initPanneau();  
}
```



Le seul attribut dans cette classe est « j » de type Joueur qui référence le joueur courant de la classe « JeuMemory ».

La boîte de dialogue ne renverra rien à la fenêtre parent car il ne s'agit ici que d'un simple affichage.

c.

```
JFrame VisuPersonnagesDlg BorderLayout  
    JLabel jLabelTitre "Cartes du joueur :" NORTH  
    JPanel Panneau CENTER  
    JPanel jPanelInfo GridLayout(2, 1) SOUTH  
        JLabel score "Score :"  
        JButton Afficher "Afficher"
```

d.

Cette boîte de dialogue ne gère qu'un seul événement, qui correspond à un clique sur le JButton « Afficher ». Cette événement va afficher sur les JButton, précédemment créés par la méthodes `initPanneau()`, l'image correspondante aux différents personnages possédés par le joueur. L'événement `AfficherActionPerformed` procède dans un premier temps à la récupération du nombre de personnages dans le paquet du joueur. Puis pour tous les joueurs (nb), on récupère le JButton correspondant, et on change son Icon par la photo du personnage.

e.

La boîte de dialogue est créée lorsque l'utilisateur clique sur JMenuItem « VisuP ». Dans l'événement `VisuPActionPerformed*`, on va créer une nouvelle instance de `VisuPersonnagesDlg` nommée « **diag** » avec comme paramètres **this**, **true** et **j**. « j », qui correspond au personnage courant. Ces paramètres sont expliqués à la question 3.b. Puis, on définit la visibilité de la boîte de dialogue sur « true », ce qui l'affiche à l'écran.

*\*Afin de tester la fenêtre, on crée en amont un joueur j nommé « FanMemory » avec comme famille préférée « commun ». On initialise le paquet de ce joueur avec la méthode `initPaquetTest()`.*