

Rapport intermédiaire sur le projet à rendre pour le lundi 30 mars

Agustín Cartaya

- 1) Expliquez les attributs que vous avez déclarés dans la classe " JeuMemory " et expliquez également le constructeur de cette classe.

Attributs :

private LesPersonnages *persos* ;

Variable de type LesPersonnages qui fait référence à une instance de la classe LesPersonnages qui contient une liste de tous les personnages qui sont sur la table de jeu et aussi contient les méthodes pour manipuler cette liste

private LesJoueurs *joueurs* ;

Variable de type LesJoueurs qui fait référence à une instance de la classe LesJoueurs qui contient une liste de tous les joueurs qui font partie de la liste et aussi contient les méthodes pour manipuler cette liste

private initDlg *initDlg*;

Variable de type initDlg qui fait référence à une instance de la classe initDlg qui contient un formulaire pour définir les paramètres principaux du jeu (difficulté de jeu) et permet également d'ajouter des joueurs par défaut, parmi 4 qui sont proposés

private visuJoueursDlg *visuJoueursDlg*;

Variable de type VisuJoueursDlg qui fait référence à une instance de la classe VisuJoueursDlg qui montre les joueurs sous forme de mosaïque et permet voir les informations du joueur en cliquant sur son image

private SaisieJoueurDlg *saisieJoueurDlg*;

Variable de type SaisieJoueurDlg qui fait référence à une instance de la classe SaisieJoueurDlg qui contient un formulaire qui permet de créer un nouveau joueur avec un pseudonyme, photo et famille préférée et l'ajouter à la liste des joueurs

private visuPersonnagesDlg *visuPersonnagesDlg*; *//non utilisé*

Variable de type VisuPersonnagesDlg qui fait référence à une instance de la classe VisuPersonnagesDlg qui montre les personnages du joueur actuel en forme d'une mosaïque d'autres informations sont aussi montrées tels que, son pseudonyme, et son score. **Non utilisé pour le moment**

private JButton [] *cartes*;

Array de type JButton contenant toutes les références aux boutons (cartes) sur le plateau du jeu

private int *nbCartes*;

Variable de type int contenant le nombre de cartes présentes sur le tableau

Constructeur de la classe " JeuMemory "

Initialisation du constructeur

```
public JeuMemory () {  
  
    Initialisation des composants fixes de la fenêtre  
    initComponents();  
  
    Création des boutons (cartes) sur le plateau de jeu  
    c'est juste pour l'esthétique, pas de vrai utilité  
    initButtons( 20 );  
  
    Initialisation de la variable joueurs  
    this.joueurs = new LesJoueurs();  
  
    Initialisation de la variable persos  
    this.persos = new LesPersonnages ();  
  
    Initialisation de la variable initDlg  
    this.initDlg = nouveau InitDlg (ce, vrai, ceci);  
  
    Initialisation de la variable visuJoueursDlg  
    this.visuJoueursDlg = new VisuJoueursDlg (this, true, this.joueurs);  
  
    Initialisation de la variable saisieJoueurDlg  
    this.saisieJoueurDlg = new SaisieJoueurDlg (this, true, this.joueurs);  
  
}
```

2) Pour la boîte de dialogue " SaisieJoueurDlg "

a. Expliquez son rôle

Fenêtre JDialog avec un formulaire qui permet de créer un nouveau joueur avec un pseudonyme, une photo et une famille préférée et l'ajouter à la liste des joueurs.

b. Détailler les informations qu'elle reçoit, celles qu'elle renvoie et celles qu'elle utilise (selon les cas), expliquer comment ces informations sont transmises de la boîte de dialogue vers l'application principale ou vice-et-versa. Expliquer le rôle des attributs de la classe et comment sont réalisés les échanges d'informations en illustrant avec votre code,

Informations qu'elle reçoit

- Par l'appel du constructeur

parent :

Variable de type Frame qui fait référence à la fenêtre parente, qui est la fenêtre de l'application principale. Est utilisée pour appeler le constructeur de la classe Jdialog.

modal :

Variable de type boolean qui établit mode d'ouverture, si elle est true la fenêtre parente est bloqué jusqu'à la fermeture de la cette fenêtre Est utilisée pour appeler le constructeur de la classe Jdialog.

lj:

Variable de type LesJoueurs qui fait référence à tous les joueurs du jeu, est utilisé pour ajouter dans la liste le nouveau joueur créé.

- Depuis l'extérieur

pseudo :

Pseudonyme du nouveau joueur, il est contenu dans le TFPpseudo JTextField TFPpseudo et est transmis directement aux attributs du joueur créé au moment de la validation.

Famille préféré :

Famille préférée du nouveau joueur, sélectionnée du JComboBox LtFamilies et passe directement aux attributx du joueur créé au moment de la validation.

Image :

URL de l'image du joueur, est obtenue grâce à un JFileChooser qui permet de parcourir les dossiers du PC, avec cette URL une variable de type Imagen est créé pour définir l'image du joueur créé au moment de la validation.

Comment ces informations sont transmises de la boîte de dialogue vers l'application principale ou vice-versa

Boite de dialogue → application principale

Le transfert d'informations dans ce sens se fait grâce au constructeur de la classe SaisieJoueurDlg qui demande comme paramètre la liste des joueurs au moment de l'instanciation.

Application principale → Boite de dialogue

Le transfert d'informations dans cette direction est effectué grâce à la référence obtenue par le constructeur de la liste de joueurs.

Note.

Je ne sais pas si c'est la meilleure façon de communiquer un JFrame avec un JDialog, puisque je n'ai pas fait usage de la variable Boolean "ok" proposé dans l'énoncé du TP, J'aimerais connaître la bonne façon de le faire, j'ai cherché sur Internet et je n'ai pas encore trouvé une réponse concrète.

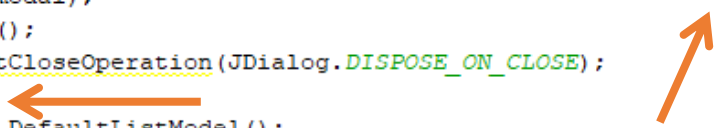
Expliquez le rôle des attributs de la classe et commentez les échanges d'informations en illustrant avec votre code

Private LesJoueurs lj;

Variable de type LesJoueurs qui fait référence à l'instance de type LesJoueurs qui contient les joueurs dans le jeu.

Echange avec l'application principale

```
public SaisieJoueurDlg(java.awt.Frame parent, boolean modal, LesJoueurs lj) {  
    super(parent, modal);  
    initComponents();  
    this.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);  
    this.lj = lj;  
    this.dlm = new DefaultListModel();  
    this.ltFamilles.setModel(this.dlm);  
    initListeFamilles();  
}
```



Echange lors de la création du nouveau joueur

```
private void btValiderActionPerformed(java.awt.event.ActionEvent evt) {  
    if(this.tfPseudo.getText() != null && !this.tfPseudo.getText().equals("")  
        && this.ltFamilles.getSelectedIndex() != -1  
        && this.btPhoto.getIcon() != null ){  
  
        this.newJoueur = new Joueur(this.tfPseudo.getText(),  
            this.dlm.getElementAt( this.ltFamilles.getSelectedIndex()).toString());  
        this.newJoueur.setPhoto( (ImageIcon)this.btPhoto.getIcon() );  
        this.newJoueur.initPaquetTest();  
        this.lj.ajouteJoueur(this.newJoueur); ←  
        this.dispose();  
    }else{  
        JOptionPane.showMessageDialog(this, "SVP saisir les informations nécessaires ",  
            "Error", JOptionPane.ERROR_MESSAGE);  
    }  
}
```

Private DefaultListModel dlm;

Variable de type DefaultListModel qui s'occupe de la gestion des objets contenus dans la liste des familles de personnages.

Remplir la liste

```
private void initListeFamilles(){  
    //creando una nueva instancia con todos los personajes se puede  
    for(String nomFamille: (new LesPersonnages(32)).getFamilles())  
        this.dlm.addElement(nomFamille); ←  
}
```

Obtenir la famille préférée pour créer le joueur

```
this.newJoueur = new Joueur(this.tfPseudo.getText(),  
    this.dlm.getElementAt( this.ltFamilles.getSelectedIndex()).toString()); ←  
this.newJoueur.setPhoto( (ImageIcon)this.btPhoto.getIcon() );  
this.newJoueur.initPaquetTest();  
this.lj.ajouteJoueur(this.newJoueur);  
this.dispose();
```

private ImageIcon photo;


Variable de type ImageIcon qui s'occupe de la création de la photo du joueur

```
private void btParcourrirActionPerformed(java.awt.event.ActionEvent evt) {  
    JFileChooser jf= new JFileChooser();  
    if (jf.showOpenDialog(this)== JFileChooser.APPROVE_OPTION)  
    {  
        String path = jf.getSelectedFile().getPath();  
        Image img=Toolkit.getDefaultToolkit().getImage(path);  
        img=img.getScaledInstance(btPhoto.getWidth()-10, btPhoto.getHeight()-10, Image.SCALE_DEFAULT);  
        photo = new ImageIcon(img); ←  
        btPhoto.setIcon(photo);  
    }  
}
```

```
privateJoueur newJoueur;;
```

Variable de type Joueur responsable du stockage du nouveau joueur créé

```
this.newJoueur = new Joueur(this.tfPseudo.getText(),  
this.dlm.getElementAt( this.ltFamilles.getSelectedIndex()).toString());  
this.newJoueur.setPhoto( (ImageIcon)this.btPhoto.getIcon() );  
this.newJoueur.initPaquetTest();  
this.lj.ajouteJoueur(this.newJoueur);  
this.dispose();
```



```
boolean privateok; //non utilisé
```

```
privateLesPersonnages lp; //non utilisé
```

Ces deux dernières variables bien qu'elles aient été proposées dans l'état n'ont pas été utilisées car je n'ai pas trouvé l'utilité pour le moment.

- c. Donner sous forme d'une arborescence (complet comme en TD) la description de l'interface

- ✓ JDialog SaisieJoueurDlg (JDialog)
 - (BorderLayout)
 - JLabel jLabel1 "Creez votre joueur !", NORTH
 - JPanel jPanel1, CENTER
 - GridLayout(1,2)
 - JPanel jPanel2
 - GridLayout(3,1)
 - JPanel jPanel4
 - (BorderLayout)
 - JLabel jLabel2 " Donnez votre pseudo: " NORTH
 - JTextField tfPseudo, CENTER
 - JPanel jPanel5
 - (BorderLayout)
 - JLabel jLabel2 " Quelle ... de personnag..? " NORTH
 - JScrollPane jScrollPane1, CENTER
 - JList ltFamilles
 - JPanel jPanel6
 - (BorderLayout)
 - JLabel jLabel2 " Choisissez une photo: " NORTH
 - JButton btParcourrir, CENTER
 - JPanel jPanel3
 - (BorderLayout)
 - JButton btPhoto, CENTER
 - JPanel jPanel7, SOUTH
 - GridLayout(3,1)
 - JButton btAnnuler
 - JButton btValider

- d. Expliquer de façon détaillée mais avec des phrases les événements gérés par la boîte de dialogue

`initListeFamilles`

Elle s'exécute lors de l'initialisation de la fenêtre, sert à remplir la liste des familles avec toutes les familles existantes des personnages, cela est réalisé en créant une instance de type `LesPersonnages` avec le nombre maximum de personnages pour après obtenir tous les noms de leurs familles.

`btParcourrirActionPerformed`

Elle s'exécute en cliquant sur le bouton "Parcourrir". un `JFileChooser` est créé, pour trouver l'URL de l'image du nouveau joueur parmi les fichiers contenus dans l'ordinateur, puis avec cette URL un objet de type `ImageIcon` est créé, l'échelle de l'image est modifiée et après elle est montré sur un bouton

`btAnnulerActionPerformed`

Elle s'exécute en cliquant sur le bouton « Annuler » la fenêtre est fermée sans rien faire d'autre.

`btValiderActionPerformed`

Elle s'exécute en cliquant sur le bouton "Valider" vérifie que le pseudonyme a été tapé, la famille préférée a été sélectionnée et une image a été choisie pour le nouveau joueur, puis le nouveau joueur est créé avec tous ces attributs, et ajouté à la liste des joueurs du jeu, après la fenêtre est fermée, si l'un des attributs requis n'est pas tape ou sélectionnée une fenêtre indiquant une erreur est affiché

- e. Expliquer la façon dont est appelée cette boîte de dialogue à partir de la fenêtre principale

Cette fenêtre est appelée lorsque du clic sur la sous-option "Ajout Joueur" de l'option "Parametres" sur la barre d'outils. Cette action rendre la fenêtre pour saisir un nouveau joueur visible grâce a la ligne du code suivante

```
this.saisieJoueurDlg.setVisible(true);
```

3) Pour la boîte de dialogue "VisuPersonnagesDlg"

a. Expliquer son rôle

Fenêtre JDialog qui montre sous la forme d'une mosaïque les personnages du joueur actuel, leur nom et leur score.

b. Détailler les informations qu'elle reçoit, qu'elle renvoie et qu'elle utilise (selon les cas), expliquer le comment ces informations sont transmises de la boîte de dialogue vers l'application principale ou vice-versa. Expliquez le rôle des attributs de la classe et commentez les échanges d'informations en illustrant avec votre code

Informations qu'elle reçoit

- Par l'appel du constructeur

pareil que la fenêtre vue ci-dessus reçoit comme paramètre la fenêtre parente et le mode d'ouverture

joueur:

Variable type Joueur qui fait référence au joueur actuel

Comment ces informations sont transmises de la boîte de dialogue vers l'application principale ou vice-versa

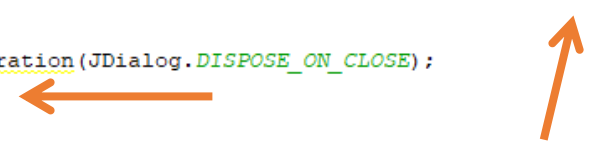
Application principale → Boîte de dialogue

Par le constructeur lors de la création de cette fenêtre

Expliquer le rôle des attributs de la classe et comment sont réalisés les échanges d'informations en illustrant avec votre code

Echange avec l'application principale du joueur actuel

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



- c. Donner sous forme d'une arborescence (complet comme en TD) la description de l'interface

- ✓ JDialog VisuPersonnagesDlg (JDialog)
 - (BorderLayout)
 - JPanel jPanel1, CENTER
 - (BorderLayout)
 - JLabel jLPseudo " Cartes du joueur: " NORTH
 - JPanel jPanel2, CENTER
 - JLabel jlScore " Score " SOUTH
 - JButton jButton1 , SOUTH //non utilise

- d. Expliquer de façon détaillée mais avec des phrases les événements gérés par la boîte de dialogue

initPanneau

Elle s'exécute lors de l'initialisation de la fenêtre, sert à créer les boutons qui contiendront les personnages du joueur actuel, d'abord les personnages sont obtenus, puis de calculs sont faits pour créer une table carrée, après pour chaque personnage du joueur un bouton qui montrera l'image du personnage est créé, enfin le nom et score du joueur est affiché.

- e. Expliquer la façon dont est appelée cette boîte de dialogue à partir de la fenêtre principale

Cette fenêtre est appelée lors du clic sur la sous-option "Cartes" de l'option "Visualiser" de la barre d'outils. Une fenêtre de type visuPersonnageDlg est créé en passant les paramètres nécessaires y compris le joueur actuel, après la fenêtre est établie comme visible.

```
(new VisuPersonnagesDlg(this, true, this.joueurs.getJoueur(this.jactuel))).setVisible(true);
```