

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

*Agustin Cartaya*

- 1) Expliquer les attributs que vous avez déclarés dans la classe « JeuMemory » et expliquer également le constructeur de cette classe.

Atributos :

`private` LesPersonnages `persos` ;

Variable de tipo LesPersonnages que hace referencia a una instancia de la clase LesPersonnages que contiene una lista con todos los personajes que están en la mesa de juego y métodos para la manipulación de la lista

`private` LesJoueurs `joueurs` ;

Variable de tipo LesJoueurs que hace referencia a una instancia de la clase LesJoueurs que contiene una lista con todos los jugadores que forman parte del juego y métodos para la manipulación de la lista

`private` InitDlg `initDlg`;

Variable de tipo initDlg la cual hace referencia a una instancia de la clase initDlg que contiene un formulario para establecer los parámetros principales del juego (dificultad de juego) y también permite agregar Jugadores por defecto, entre 4 que son propuestos

`private` VisuJoueursDlg `visuJoueursDlg`;

Variable de tipo VisuJoueursDlg la cual hace referencia a una instancia de la clase VisuJoueursDlg que muestra a los jugadores actuales del juego en forma de mosaico y permite ver la información de cada uno seleccionando su imagen

`private` SaisieJoueurDlg `saisieJoueurDlg`;

Variable de tipo SaisieJoueurDlg la cual hace referencia a una instancia de la clase SaisieJoueurDlg que contiene un formulario que permite crear un nuevo jugador con su seudónimo, su foto y su familia preferida y agregarlo a la lista de jugadores

`private` VisuPersonnagesDlg `visuPersonnagesDlg`; *//non utilise*

Variable de tipo VisuPersonnagesDlg la cual hace referencia a una instancia de la clase VisuPersonnagesDlg que muestra en forma de mosaico los personajes del jugador actual, su nombre y su puntuación. **no utilizada por el momento.**

`private` JButton [] `cartes`;

Arreglo de tipo JButton que contiene todas las referencias a los botones (cartas) del tablero de juego

`private` int `nbCartes`;

variable de tipo int que contiene la cantidad de cartas presentes en el tablero

Constructeur de la classe « JeuMemory »

```
//Inicio Del constructor
public JeuMemory() {

    //Inicialización de los componentes fijos de la ventana
    initComponents();

    //Creacion de los botones (cartas) del tablero de juego
    //se han puesto aquí por cuestión de estética, aun no tienen
    //ninguna utilidad
    initButtons( 20 );

    // Inicialización de la variable joueurs
    this.joueurs = new LesJoueurs();

    // Inicialización de la variable persos
    this.persos = new LesPersonnages ();

    // Inicialización de la variable initDlg
    this.initDlg = new InitDlg(this, true, this);

    // Inicialización de la variable visuJoueursDlg
    this.visuJoueursDlg = new
    VisuJoueursDlg(this,true,this.joueurs);

    // Inicialización de la variable saisieJoueurDlg
    this.saisieJoueurDlg = new SaisieJoueurDlg(this,true,
    this.joueurs);

}
```

2) Pour la boîte de dialogue « SaisieJoueurDlg »

- a. Expliquer son rôle  
ventana JDialog con un formulario que permite crear un nuevo jugador con su seudónimo, su foto y su familia preferida y agregarlo a la lista de jugadores
- 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 tipo Frame que hacer referencia a la ventana padre, la cual es la ventana principal de la aplicación, es utilizada para llamar al constructor de la clase JDialog
  - modal:  
variable de tipo boolean modo de apertura, si es true la ventana padre queda bloqueada hasta el cierre de esta, sino se puede utilizar ambas a la vez, es utilizada para llamar al constructor de la clase JDialog
  - lj:  
variable de tipo LesJoueurs que hace referencia a todos los jugadores del juego, es utilizada para agregar en nuevo jugador creado
- Depuis l'extérieur
  - Pseudo :  
Seudónimo del jugador a crear, contenido en el JTextField tfPseudo y es pasado directamente al atributo pseudo del jugador creado en el momento de validación
  - Famille préféré:  
Familia preferida del jugador a crear, es seleccionado del JComboBox ltFamilles y pasada directamente al atributo famillePref del jugador creado en el momento de validación
  - Image:  
URL de la imagen del jugador, es obtenida gracias a un JFileChooser que permite recorrer las carpetas del pc, con esta URL se crea la instancia de tipo Imagemcon para establecer la imagen del jugador en el momento de validación

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

Boite de dialogue → application principale

La transferencia de información siguiendo este sentido se hace gracias al constructor de la clase SaisieJoueurDlg que pide como parámetro la lista de jugadores al momento de la instanciación.

Application principale → Boite de dialogue

La transferencia de información siguiendo este sentido se hace gracias a la referencia obtenida por el constructor de la lista de jugadores.

**Nota.**

No sé si esta es la mejor manera de comunicar un JFrame con un JDialog, ya que no he hecho uso de la variable booleana "ok" propuesta en el enunciado del TP, estaría muy agradecido si me mostraran la forma correcta ya que he buscado por internet y aun no he encontrado nada.

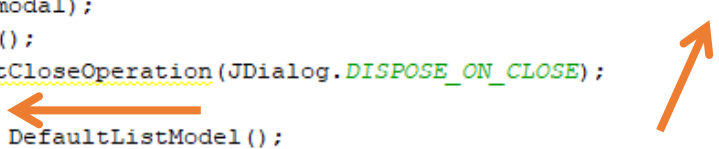
*Expliquer le rôle des attributs de la classe et comment sont réalisés les échanges d'informations en illustrant avec votre code*

`private LesJoueurs lj;`

Variable de tipo LesJoueurs que hace referencia a la instancia de tipo LesJoueurs que contiene los jugadores del juego

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 tipo DefaultListModel encargada de gestionar los objetos contenidos en la lista de familias de personajes.

Rellenando la lista

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

Obteniendo familia preferida para crear el jugador

```
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 tipo ImageIcon encargada de crear la foto del jugador

```
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);  
    }  
}
```

```
private Joueur newJoueur;
```

Variable de type Joueur encargada de almacenar el nuevo jugador creado

```
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 boolean ok; //non utilisé
```

```
private LesPersonnages lp; //non utilisé
```

estas dos últimas variables, aunque fueron propuestas en el enunciando no se utilizaron que no personalmente no encontré la utilidad por el momento

- c. Donner sous forme d'une arborescence (complète 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 personmag..? " 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`

Se ejecuta al iniciar la ventana, sirve para rellenar la lista de familias con todas las familias existentes de los personajes, esto se logra creando una instancia tipo `LesPersonnages` con el máximo numero de personajes y obteniendo todos los nombres de sus familias

#### `btParcourrirActionPerformed`

Se ejecuta cuando se hace clic en el botón “Parcourrir” se crea un `JFileChooser`, para buscar la URL de la imagen del nuevo jugador en los archivos de la computadora, luego con esta URL se crea un objeto de tipo `ImageIcon`, se escala y se muestra en el botón para previsualizar la foto.

#### `btAnnulerActionPerformed`

Se ejecuta cuando se hace clic en el botón “Annuler” y cierra la ventana sin hacer nada mas.

#### `btValiderActionPerformed`

Se ejecuta cuando se hace clic en el botón “Valider” verifica que se haya escrito el seudónimo, se haya seleccionado la familia favorita y se haya elegido una imagen para el nuevo jugador, luego se crea el nuevo jugador con todos estos atributos, se añade a la lista de jugadores del juego y se cierra la ventana, si alguna de los atributos requeridos hace falta una ventana emergente señalando un error es mostrada

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

Esta ventana es llamada cuando se hace clic en la sub opción “Ajout Joueur” de la opción “Parametres” de la barra de herramientas, lo que se hace dentro del método que gestiona la acción establecer la ventana como visible utilizando la siguiente línea de código

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

### 3) Pour la boîte de dialogue « VisuPersonnagesDlg »

- a. Expliquer son rôle  
ventana JDialog que muestra en forma de mosaico los personajes del jugador actual, su nombre y su puntuación.
- 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  
Parecido que la ventana vista anteriormente recibe como parámetro la ventana padre y el modo de apertura  
  
joueur:  
variable de tipo Joueur que hace referencia al jugador actual del juego

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

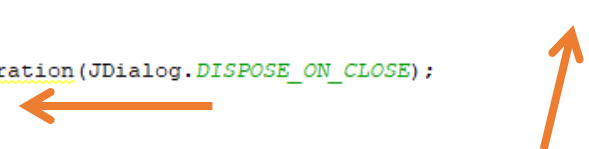
Boite de dialogue → application principale  
**Sin transferencia de información**

Application principale → Boite de dialogue  
**Por el constructor cuando se crea la instancia de esta ventana**

*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 (complète 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

Se ejecuta al iniciar la ventana, sirve para crear los botones que contendrán los personajes del jugador actual, primero se obtienen los personajes, luego se calcula como estos van a ser distribuidos intentado crear una tabla cuadrada, luego por cada personaje del jugador se crea un botón el cual mostrara la imagen del personaje, por ultimo se muestra el nombre del jugador su puntuación total

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

Esta ventana es llamada cuando se hace clic en la sub opción "Cartes" de la opción "Visualiser" de la barra de herramientas, lo que se hace dentro del método que gestiona la acción es crear la ventana pasando los parámetros necesarios, entre ellos el jugador actual y hacerla visible utilizando la siguiente línea de código

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