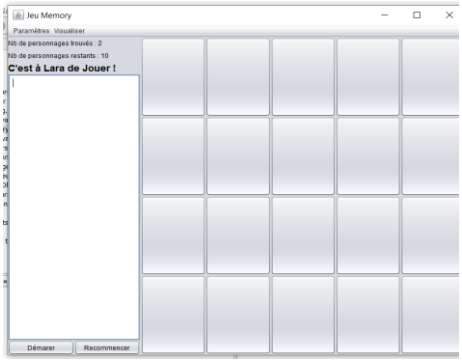
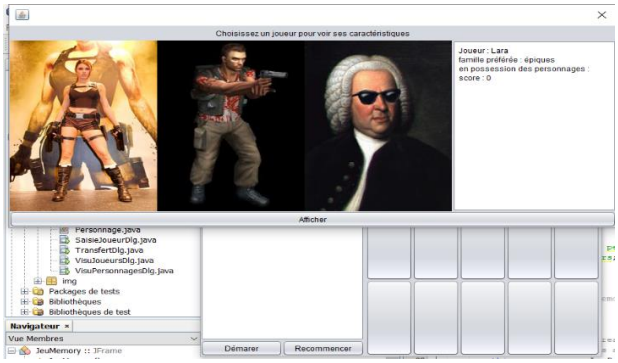
Nom : Ouattara Prenom : Ismael Simeon Herve Groupe : MI2-I2

Rapport de Projet:





1-explication des attributs dans la classe <<JeuMemory>> et explication le constructeur de cette classe.

private LesPersonnages persos; //attribut Persos de type LesPersonnages qui gere l' ensemble des personnages

private LesJoueurs joueurs; //attribut Joueurs de type LesJoueurs qui gere un ensemble de joueurs

private Joueur j; //attribut j de type Joueur qui gere un joueur

-Constructeur de cette classe

public JeuMemory() {

Creation de l'interface

initComponents() ;

this.persos = new LesPersonnages(); //initialisation de l'attribut persos et creation d'une instance de <<lespersonnage>>

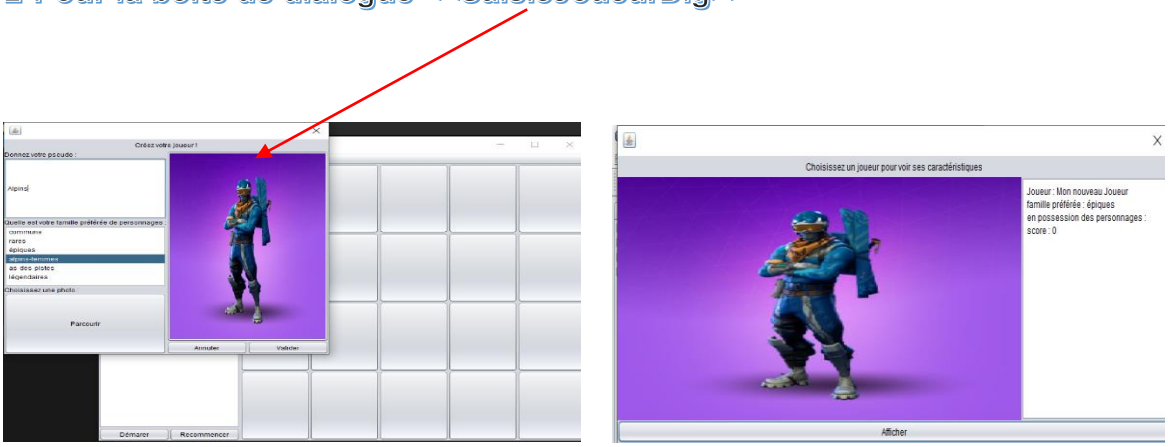
this.joueurs = new LesJoueurs(); //initialisation de l'attribut joueurs et creation d'une instance de <<LesJoueurs>> .

this.j = new Joueur("FanMemory", "commun"); //creation d'un joueur avec des cartes

this.j.initPaquetTest(); //appel a la methode du joueur permettant d'afficher ses cartes

}

2-Pour la boite de dialogue <<SaisieJoueurDlg>>



a – Explication du role de cette classe

Cette classe permet de créer un **nouveau joueur** par la saisie des informations relatives à ce joueur. Ce dernier sera ajouté à la liste des joueurs dans l'application principale à la fermeture de la fenêtre.

b – Detaillons les informations qu'elle recoit et qu'elle renvoie et ce qu'elle utilise (selon le cas) expliquer comment ces informations sont transmises de la boite de dialogue vers l'application principale ou vise versa. Expliquer le role des attributs de la classe et comment sont realisés les echanges d'informations en illustrant avec votre code.

Lors de l'ouverture de cette **JDialogue:**

-On va **saisie** Le Pseudo du Joueur

-Choisir la famille préférée du personnage (qui n'autorisation pas de doublon) dans la liste des famille(JList) .

Utilisation de L'attribut **lp** de type **<<LesPersonnages>>** (c 'est a dire Utilisation de la classe **LesPersonnages**) qui va etre initialiser dans le constructeur Pour permet d'initialiser la liste de famille existantes.

-**Choisir la photo** du joueur grâce au bouton **Parcourir** qu'il va appelé le gestionnaire du bouton parcourir qui lui , permet d'aller dans la boîte de dialogue «JFileChooser» est la boîte de sélection d'un fichier, elle retourne un objet de type «File». La méthode «getPath()» de cette classe «File» donne le chemin complet du fichier.

-La photo du Joueur sera **reçu** et **afficher** dans le **Photo** qui est un Bouton.

Ensuite

-Le Clic sur le bouton **Annuler,** va fermer la boite de Dialogue **SaisieJoueurDlg** et annuler tous les informations saisie dans cette JDialogue.

-Le clic sur le bouton **Valider** , recuperation de tous les informations de cette JDialogue a savoir (Pseudo du joueur, la photo et le nom de famille du joueur) vont etre envoyer vers l'application priincipale apres verification de certaines conditions comme **pas de doublon** du nom de famille.et envoyer lorsque **le boolean est true**, c est a dire le joueur est crée apres les conditions des differents informations.

-on ajoute un parametre **LesJoueurs lj** recupere les joueurs a afficher qui sont des informations qui proviennent de l'application principales.

-le transfere des informations de SaisieJoueursDlg vers l'application principale :

On Ajouter dans la classe principale **«JeuMemory»** le gestionnaire d'évènement du clic sur la sous-option «Ajout Joueur» de l'option «Paramètres» du menu. Ce gestionnaire ouvre la boîte de dialogue, et si elle est fermée par «Valider», ajoute le joueur saisi dans l'ensemble des joueurs.

Fenetre qui ajoute mode d'ouverture permet d'avoir acces aux joueurs

```
public SaisieJoueurDlg ( java.awt.Frame parent, boolean modal, LesJoueurs lj ) {

      super(parent, modal); // appel du constructeur avec parametre de la JDialogue ( classe mere)

….

}
```

```
   private Joueur joueur;  //attribut Joueur de type joueur

   private boolean ok;   // attribut ok de type boolean qui sera l'indicateur de fermeture de cette  JDialogue. True  : valider  et false : false.

   private ImageIcon photo; // attribut photo de type ImageIcon qui permet de gere l'image

   private LesJoueurs lj; //attribut lj de type LesJoueurs qui permet de gere les joueurs

   private LesPersonnages lp; // attribut lp de types les Personnages qui permet de gerer les personnages
```

c)Donner sous forme d'une Arborescence la description de l'interface

- JDialog SaisieJoueurDlg
  - BorderLayout
  - JPanel jPanel 1 , NORTH

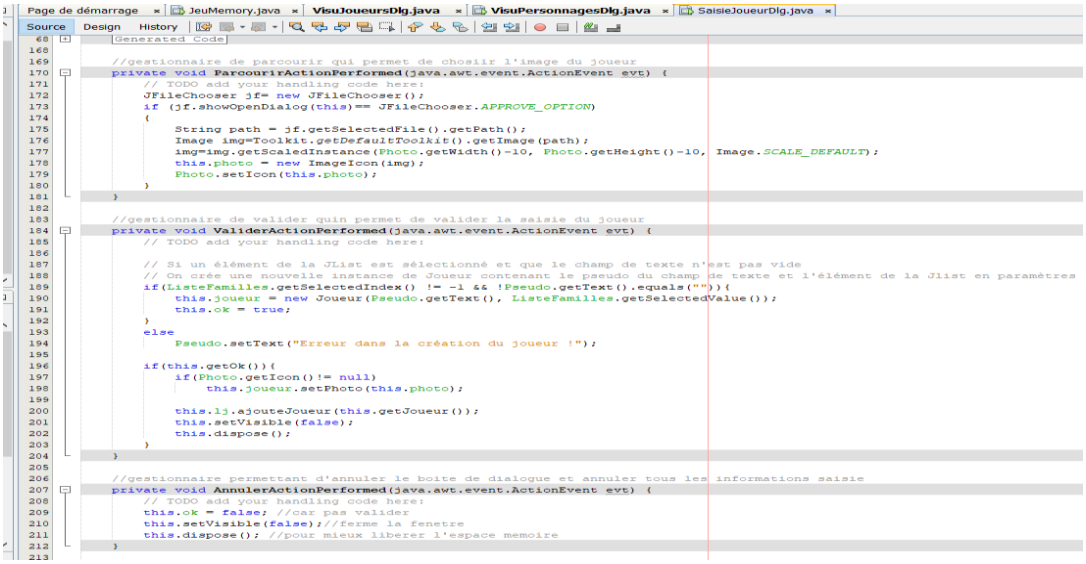
- FlowLayout
- JLabel jLabel1 «Creez votre joueur ! »

JPanel jPanel2 , CENTER

- GridLayout(1,2)
- JPanel jPanel3
  - GridLayout(3,1)
  - JPanel jPanel4
    - BorderLayout
    - JLabel jLabel2 « Donnez votre pseudo : »
    - JTextField Pseudo
  - JPanel jPanel5
    - BorderLayout
    - JLabel jLabel2 « Quelle est votre famille préférée de Personnages ? »
    - JScrollPane jScrollPane1
      - JList ListeFamilles
  - JPanel jPanel6
    - BorderLayout
    - JLabel jLabel2 « Choisissez une photo : »
    - JButton Parcourir « Parcourir »
- JPanel jPanel7
  - BorderLayout
  - JButton Photo « »
  - JPanel jPanel8 , South
    - GridLayout(1, 2)
      - JButton Annuler « Annuler»

↳ JButton Valider « Valider »

d)Expliquons de facon détaillée mais avec des phrases les evenements gérés par la boite de dialogue



```
Page de démarrage | JeuMemory.java | VisuJoueursDlg.java | VisuPersonnagesDlg.java | SaisieJoueurDlg.java
Source | Design | History
    Generated Code

    //gestionnaire de parcourir qui permet de chosiir l'image du joueur
    private void ParcourirActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        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(Photo.getWidth()-10, Photo.getHeight()-10, Image.SCALE_DEFAULT);
            this.photo = new ImageIcon(img);
            Photo.setIcon(this.photo);
        }
    }

    //gestionnaire de valider quin permet de valider la saisie du joueur
    private void ValiderActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        // Si un élément de la JList est sélectionné et que le champ de texte n'est pas vide
        // On crée une nouvelle instance de Joueur contenant le pseudo du champ de texte et l'élément de la Jlist en paramètres
        if(ListeFamilles.getSelectedIndex() != -1 && !Pseudo.getText().equals("")){
            this.joueur = new Joueur(Pseudo.getText(), ListeFamilles.getSelectedValue());
            this.ok = true;
        }
        else
            Pseudo.setText("Erreur dans la création du joueur !");

        if(this.getOk()){
            if(Photo.getIcon() != null)
                this.joueur.setPhoto(this.photo);

            this.lj.ajouteJoueur(this.getJoueur());
            this.setVisible(false);
            this.dispose();
        }
    }

    //gestionnaire permettant d'annuler le boite de dialogue et annuler tous les informations saisie
    private void AnnulerActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        this.ok = false; //car pas valider
        this.setVisible(false);//ferme la fenetre
        this.dispose(); //pour mieux liberer l'espace memoire
    }
```

permet l'ajout de la photo dans le bouton Photo

-Private void ParcourirActionPerformed (java.awt.event.ActionEvent evt){

…img =img.getScaledInstance(…) ;

Photo.setIcon(photo) ; }

Cette methode permet de créer une version reduite de l'image de l'objet.

le gestionnaire du bouton parcourir , permet d'aller dans la boîte de dialogue «JFileChooser» est la boîte de sélection d'un fichier, elle retourne un objet de type «File». La méthode «getPath()» de cette classe «File» donne le chemin complet du fichier. Et qui permettra de choisir la photo.

-Private void AnnulerActionPerformed(java.awt.event.ActionEvent evt){

this.ok = false; //car pas valider

this.setVisible(false);// fermer la fenetre

this.dispose();// pour mieux liberer l'espace memoire

}

-Private void ValiderActionPerformed (java.awt.event.ActionEvent evt){

// Si un élément de la JList est sélectionné et que le champ de texte n'est pas vide

// On crée une nouvelle instance de Joueur contenant le pseudo du champ de texte et l'élément de la Jlist en paramètres

}

e)Expliquons la facon dont est appelée cette boite de dialogue a partir de la fenetre principale

```
201
202     //gestionnaire de la sous option - AjoutJoueur qui permet de visualisation la boite de dialogue SaisieJoueurDlg
203     private void AjoutJoueurActionPerformed(java.awt.event.ActionEvent evt) {
204         // TODO add your handling code here:
205         SaisieJoueurDlg saisie = new SaisieJoueurDlg(this, true, this.joueurs);
206         saisie.setVisible(true);
207     }
208
```

On Ajouter dans la classe principale **«JeuMemory»** le gestionnaire d'évènement du clic sur la sous-option «Ajout Joueur» de l'option «Paramètres» du menu. Ce gestionnaire ouvre la boîte de dialogue, et si elle est fermée par «Valider», ajoute le joueur saisi dans l'ensemble des joueurs. Ce gestionnaire :

Private void AjoutJoueurActionPerformed (java.awt.event.ActionEvent evt){

Ajoute un parametre au constructeur pour recupere le joueur saisi dans l'ensemble des joueurs

::SaisieJoueurDlg saisie = new SaisieJoueurDlg(this, true, this.joueurs);

saisie.setVisible(true); //ouverture la boite de dialogue

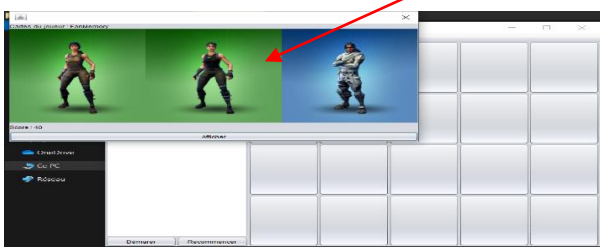
la fenetre principale gere la boite

mode d'ouverture modal

= bloquant

}

3-Pour la boite de Dialogue VisuPersonnagesDlg



a-expliquer le role

Cette boîte de dialogue permet de **visualiser** les personnages (**cartes**) gagnés par le **joueur** courant (celui qui est en train de jouer). Par exemple, lors d'une partie, avec Jack et Lara, lorsque c'est à Lara de jouer,et il est possible de visualiser les «cartes» déjà gagnées par Lara sous forme d'un «trombinoscope».

b – Detaillons les informations qu'elle recoit et qu'elle renvoie et ce qu'elle utilise (selon le cas) expliquer comment ces informations sont transmises de la boite de dialogue vers l'application principale ou vise versa. Expliquer le role des attributs de la classe et comment sont realisés les echanges d'informations en illustrant avec votre code.

```
private Joueur joueur ;    //attribut joueur de (type joueur ) dont on veut afficher les cartes qui sera recupere par le
                                                    constructeur.
```

L'instance de type « joueur »qui initialise l'attribut qui correspondant et appele la methode initPanneau

Fenetre principale qui gere la boite mode d'ouverture=bloquant

```
public VisuPersonnagesDlg(java.awt.Frame parent, boolean modal, Joueur joueur) {
    super(parent, modal); //appel du constrcuteur de la classe ancetre
    initComponents(); //creation de l'interface
    this.joueur = joueur; //initialisation de l'attribut joueur
    initPanneau(); //methode du panneau qui contient les trombinoscope
    Perso.setText("Cartes du joueur : " + this.joueur.getNom()); //permet d'afficher le Pseudo du joueur en
                                                    recuperant le Pseudo du joueur.
    Score.setText("Score : " + this.joueur.getPaquet().getScore()); //Permet d'afficher le Score du joueur en
                                                    recuperant le score du Paquet.
}
```

le constructeur a en paramètre une instance de type **«Joueur »**pour initialiser l'attribut correspondant. Il fait appel à une méthode **«initPanneau»** qui permet la création du trombinoscope. Et permet d'avoir accés aux informations de l'application principales.

-la classe principale **«JeuMemory»**

le gestionnaire d'évènement du clic sur la sous-option **«Cartes»** de l'option **«Visualiser»** du menu. Ce gestionnaire ouvre la boîte de dialogue en passant en paramètre le joueur dont les cartes doivent être affichées. Et

affiche la **JDialogue**, et lorsqu'on clique sur **« afficher »,** il r**ecupere** les informations de **l'application principale** et cela permet d'afficher les cartes personnages gagnés par le joueur.

c)Donner sous forme d'une Arborescence la description de l'interface

- JDialog VisuPersonnagesDlg
  - BorderLayout
    - JLabel Perso « Carte du Joueur : » , NORTH
    - JPanel Panneau « » , CENTER
    - JPanel jPanel1 , SOUTH
      - GridLayout(2,1)
      - JLabel Score « Score : »
      - JButton Afficher « Afficher »

d)Expliquons de facon détaillée mais avec des phrases les evenements gérés par la boite de dialogue

```
81
82      //gestionnaire d'afficher qui permet d'afficher les cartes des personnages gagnés par le joueur
83      private void AfficherActionPerformed(java.awt.event.ActionEvent evt) {
84          // TODO add your handling code here:
85          JButton jb;
86          for(int i = 0; i < this.joueur.getPaquet().getTaille(); i++){
87              jb = (JButton)Panneau.getComponent(i);
88              Image img = this.joueur.getPaquet().getPerso(i).getPhoto().getScaledInstance(jb.getWidth(), jb.getHeight(), Image.SCALE_SMOOTH);
89              jb.setIcon(new ImageIcon(img));
90          }
91      }
```

private void AffcherActionPerformed(java.awt.event.ActionEvent evt){

//ce gestionnaire permet d'afficher des cartes des personnages gagnés par le joueur

On va créer d'abord un bouton, on va cree ensuite un tableau des paquet, ce bouton sera afficher dynamiquement dans le panneau et ensuite on va redimensionner l'image du paquet du joueur qui sera afficher dans le panneau.

jb = (JButton)Panneau.getComponent(i); //le component qui a decleché l'action (quel bouton a te clique ici)

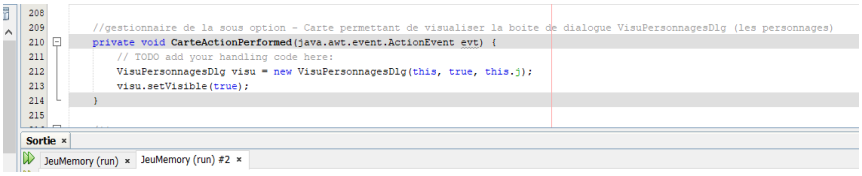
This.lj.getNbJoueurs() ; //methode de la classe LesJoueurs

ImageIcon : constructeur avec parametre de la classe ImageIcon

On va créer une instance de ImageIcon, a qui on va associer un accesseur en ecriture de l'icone.pour pouvoir associer le bouton a l'image.

}

e)Expliquons la facon dont est appelée cette boite de dialogue a partir de la fenetre principale



Dans la fentre Principale, le gestionnaire d'évènement du clic sur la sous-option **«Cartes»** de l'option **«Visualiser»** du menu. Ce gestionnaire ouvre la boîte de dialogue en passant en paramètre le joueur dont les cartes doivent être affichées. Et

affiche la **JDialogue**, et lorsqu'on clique sur **« afficher »,** il r**ecupere** les informations de **l'application principale** et cela permet d'afficher les cartes personnages gagnés par le joueur.

liste de joueur passer en parametre de type joueur

```
private void CarteActionPerformed(java.awt.event.ActionEvent evt) {
    VisuPersonnagesDlg visu = new VisuPersonnagesDlg(this, true, this.j);
    visu.setVisible(true);//permet l'ouverture de boite de dialogue
}
```

La fenetre principale gere la boite

mode d'évolution