

# REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



Université des sciences et de la technologie Houari Boumediene  
Département d'Informatique

## PROJECT POO LA REALISATION D'UN JEU DE CULTURE GENERALE



Binôme :

- MENGHOUR Walid
- ABB HAMOU Nouredine

Encadre par :

✓ Nabila Salmi

Année Universitaire : 2019/2020

# REMERCEMENT

A la fin de ce projet, nous remercions ALLAH qui nous a aidé et nous a donné la

Patience et le courage durant ce semestre qui était en rebondissement.

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont

Apporté leur aide et qui ont contribué à la mise en œuvre de notre projet et à la réussite de

Cette formidable expérience.

.....

J'adresse mes remerciements aux personnes qui m'ont aidé dans la réalisation de ce Project

Je Remercie Notre charge de cours Mme NABILA SALMI a tous le choisie qui elle été fait Just pour communiqué l'information dans cette dure de virus COVID-19 . Par email par capture vidéo par document PDF par webex ,Merci .

Je souhaite particulièrement remercier mon binôme NOUREDDINE ABA HAMOU a ses efforts dans cette dure de la réalisation de ce projet par distance.

.....

Je remercie tout d'abord notre charge de cours madame Nabila Salmi à ses efforts, son orientation, son assistance et le choisissent de ce projet magnifique qui il nous a aidé à maîtriser les concepts de POO

Ensuite, Je remercie mon binôme MENGHOUR WALID à ses efforts FANTASTIQUES pendant cette durée de travail...

Merci à tous

# CONTENU

<b>INTRODUCTION.....</b>	<b>6</b>
<b>ARCHITECTURE DE LA SOLUTION.....</b>	<b>1</b>
DIAGRAMME UML PACKAGE COM.USTHB .....	1
DIAGRAMME UML PACKAGE COM.USTHB.MODELES .....	2
DIAGRAMME UML PACKAGE COM.USTHB.DESSIN.....	3
<b>PACKAGE COM.USTHB. MODELES.....</b>	<b>7</b>
CLASS EUREKA .....	7
<b>PAKAGE COM.USTHB. MODELES.....</b>	<b>8</b>
CLASS JOUEUR .....	8
CLASS ENFANT .....	9
CLASS ADULTE .....	9
CLASS THEMEJEU .....	9
CLASS QUESTION.....	10
CLASS QUESTIONENFANT. ....	10
CLASS QUESTIONADULTE .....	10
CLASS PARTIEJEU.....	11
<b>PAKAGE COM.USTHB. DESSING .....</b>	<b>12</b>
BIENVENU .....	12
.....	12
SECONNECTERINT .....	13
INSCRIPTIONINT .....	15
SELECTTHEMEINT .....	16
PARTIEDEJEUINT .....	16
LJOUEURSINSCR .....	16
<b>RÉSUMER LES STRUCTURES DE DONNÉES GLOBALES UTILISÉES :.....</b>	<b>17</b>
<b>OUTILS DE DÉVELOPPEMENT UTILISÉS.....</b>	<b>18</b>
<b>OUTILS DE DESIGN UTILISE.....</b>	<b>18</b>
<b>OUTILS DE COMMUNICATION ET TRAVAILLE.....</b>	<b>18</b>
<b>CONCLUSION .....</b>	<b>19</b>
REMARQUES SUR MA REALISATION.....	19
CE QUE JE PROPOSE COMME AMELIORATION DE L'APPLICATION .....	19
CE QUE J'AI APPRIS DU PROJET .....	19

## Table des Figure

<b>FIGURE 1 : DIAGRAMME UML GENERAL DE GESTION EUREKA.....</b>	<b>1</b>
<b>FIGURE 2 SCHEMA DE LES COMPOSANT GRAPHIQUE.....</b>	<b>2</b>
<b>FIGURE 3 DIAGRAMME UML DE CLASS PAKAGE COM.USTHB.DESSIN.....</b>	<b>3</b>
<b>FIGURE 4: SCHEMA REPRESENT LES DIFFERENTES ETAPES DE L'INSCRIPTION ET LA CONNEXION JUSQU'A LANCER UNE PARTIE DE JEU.....</b>	<b>4</b>
<b>FIGURE 5 SCHEMA REPRESEN LES DIFFERENTES ETAPES DE PARTIE JEU .....</b>	<b>5</b>
<b>FIGURE 6: SCHEMA D'UTILISATION DES FICHIERS.....</b>	<b>6</b>

## Introduction

Ce jeu est destiné à différents groupes d'âge, enfants et adulte, L'Objectif de EUREKA est de tester les différentes connaissances de le Jouer d'après une suite des questions sur un thème spécifique choisie au début par le Jouer est les thèmes définis dans cette Jeu sont (HISTOIRE, GEOGRAPHIE, SANTE, CULTURE GENERALE, ISLAM) si le Jouer Gagné les 5 niveaux du thème Choies elle peut choisie autre Thème est Jeu ainsi de suite jusqu'à le gagé tous les niveaux des Thème, Eureka en Performance d'amélioration d'un Le niveau de réflexion Et analyse des différentes connaissances en Qualité et réduction de défaillances Techniquement est Acquérir de nouvelles connaissances .

# Architecture de la solution

Voici un schéma General de Project de Gestion Eureka d'un frome minimal

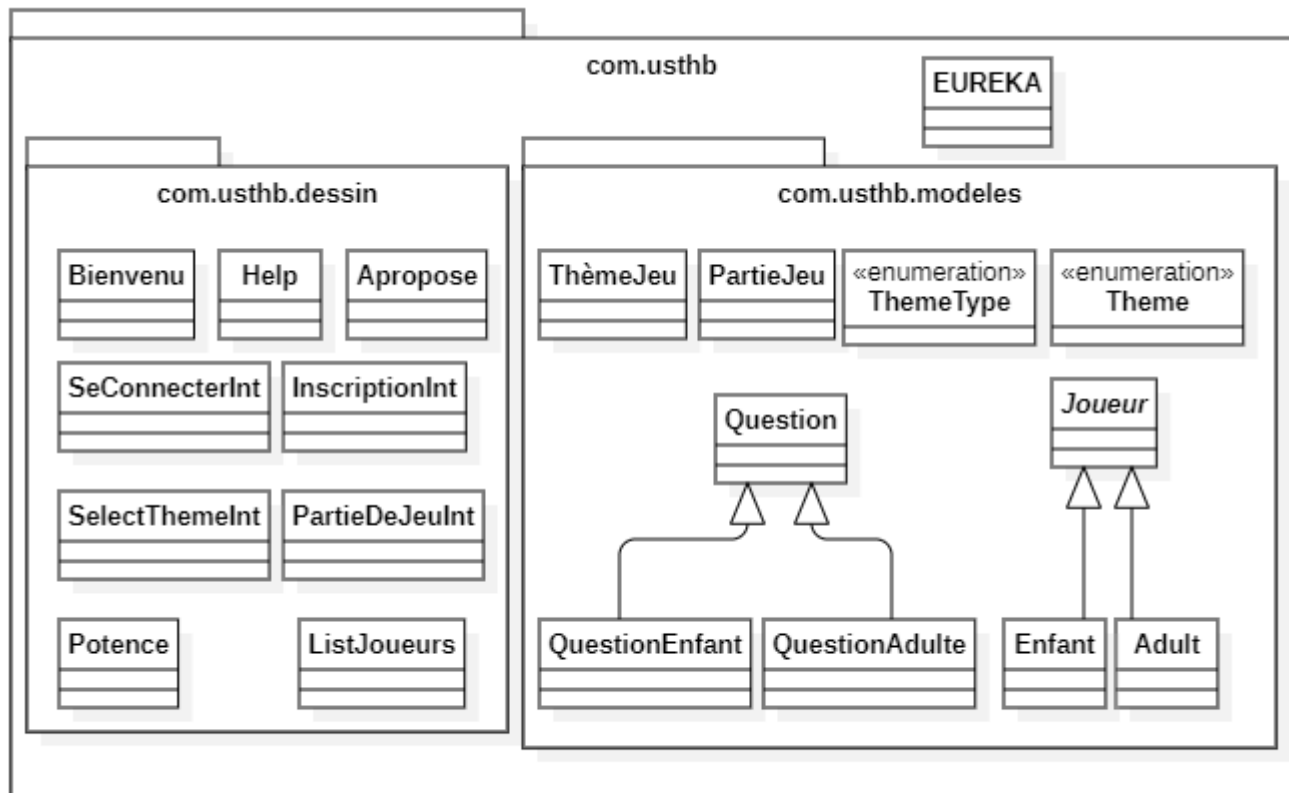


Figure 1 : Diagramme UML général de Gestion Eureka

## DIAGRAMME UML PACKAGE COM.USTHB

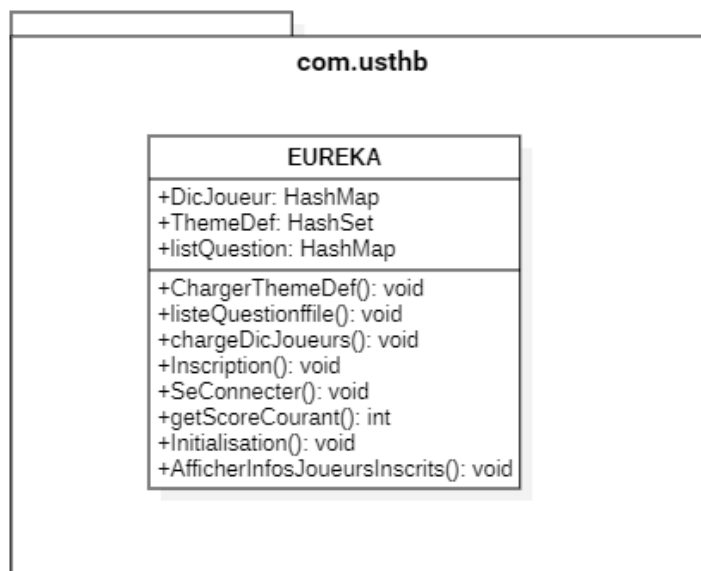


Figure 1 : Diagramme UML de class Eureka

## DIAGRAMME UML PACKAGE COM.USTHB.MODELES

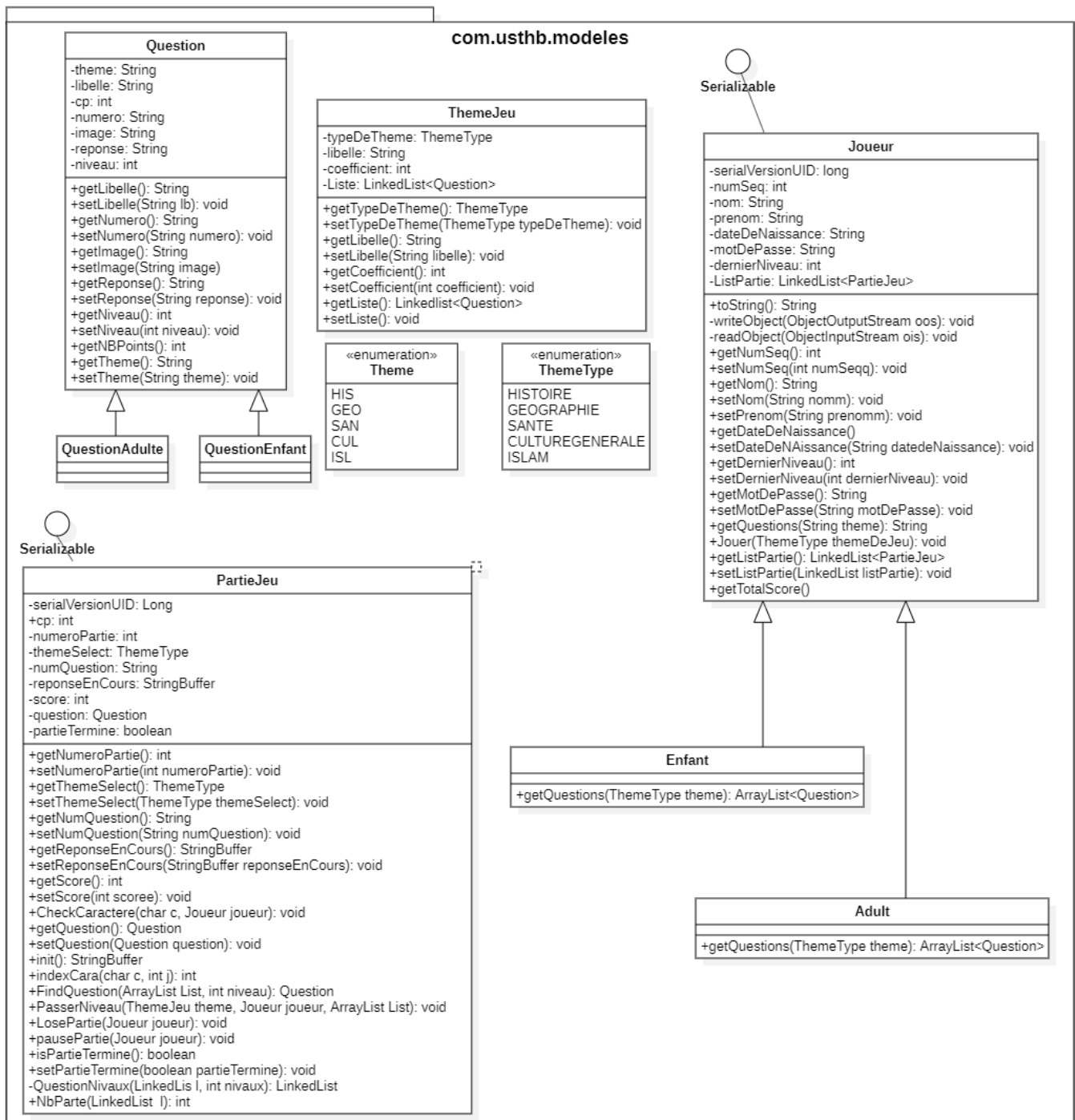
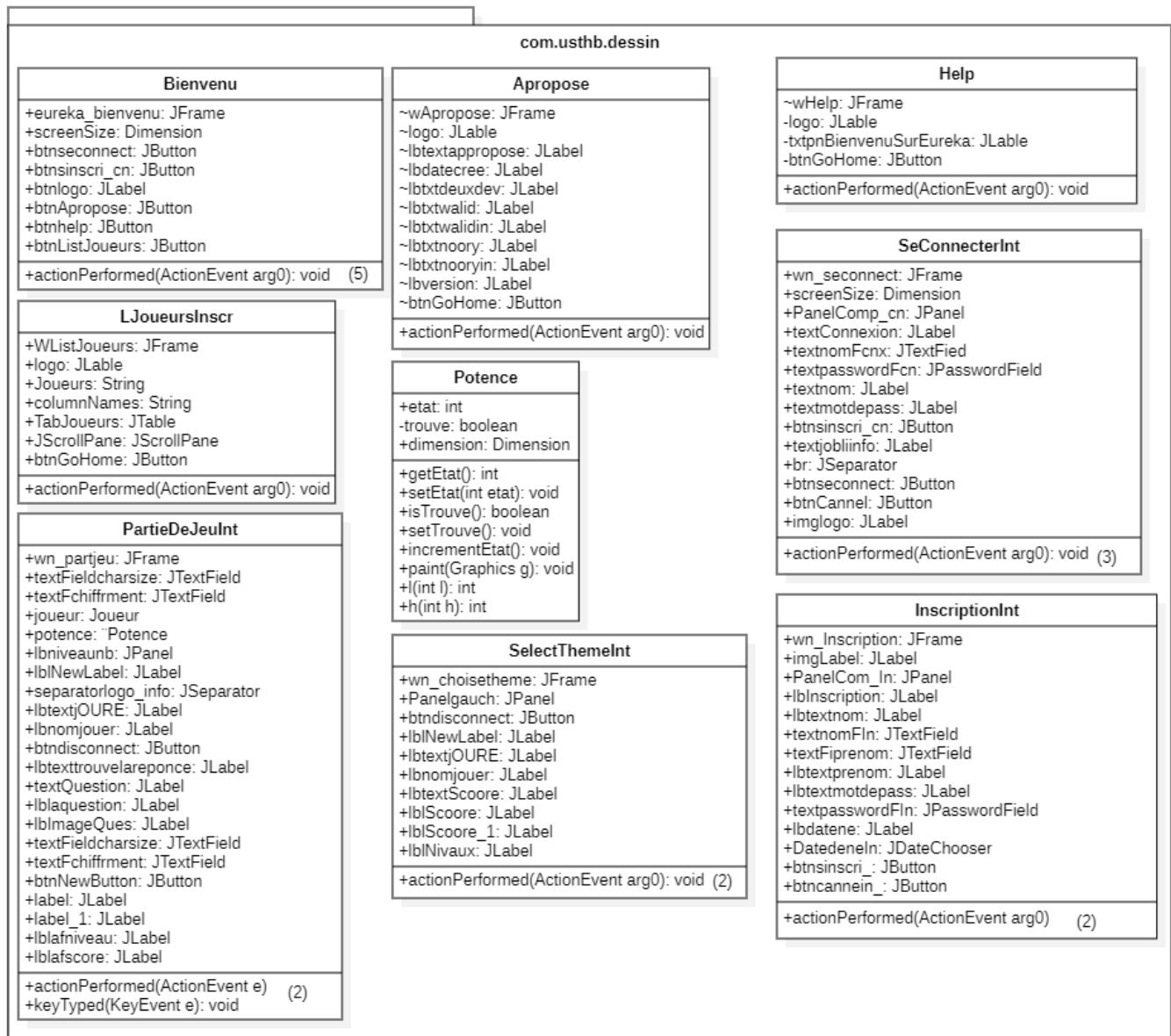


Figure 2: Diagramme UML des Classes du Package com.usthb.modeles

## DIAGRAMME UML PACKAGE COM.USTHB.Dessin



**Figure 3 Diagramme UML de Class Package com.usthb.dessin**

### REMARQUE

- 1- Les numéro entre parenthèsereprésenté le nombre dèss fonctions dans la classe ayant des corps différents.



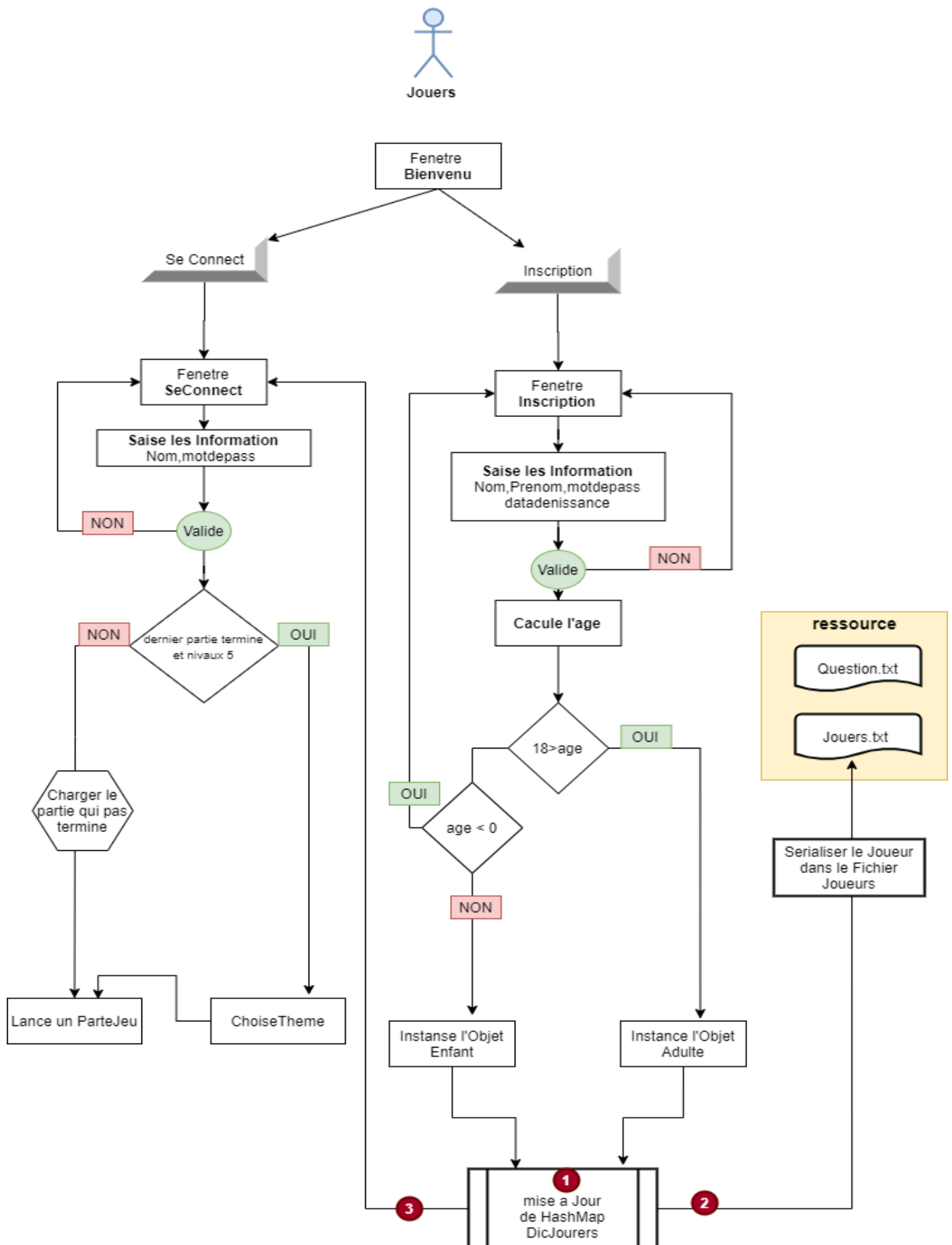


Figure 4: Schéma Représente les différentes étapes de l'inscription et la connexion jusqu'à lancer une partie de jeu

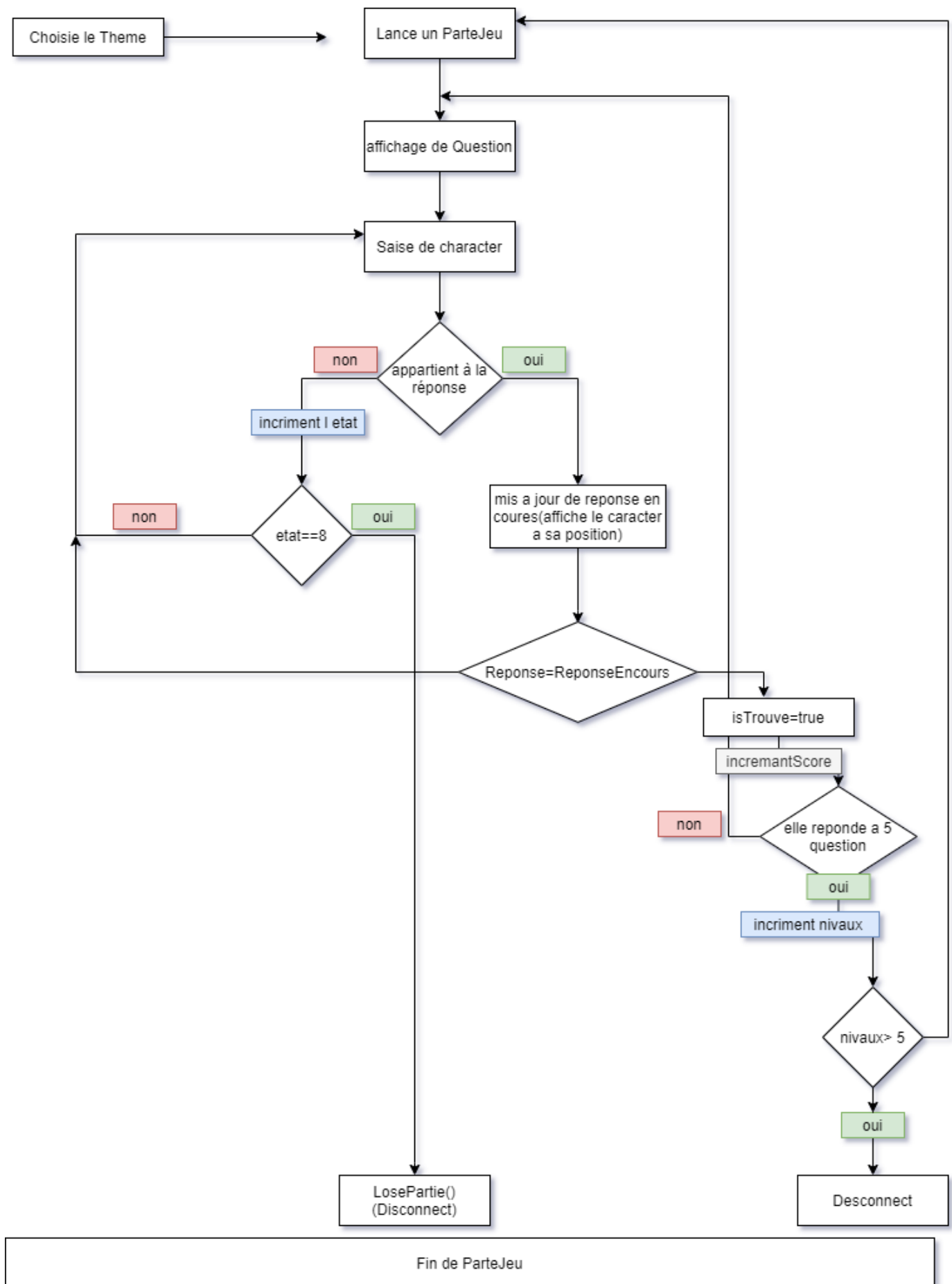


Figure 5 Schéma Représent les différentes étapes de partie Jeu

## Le Concept de la Serialisation

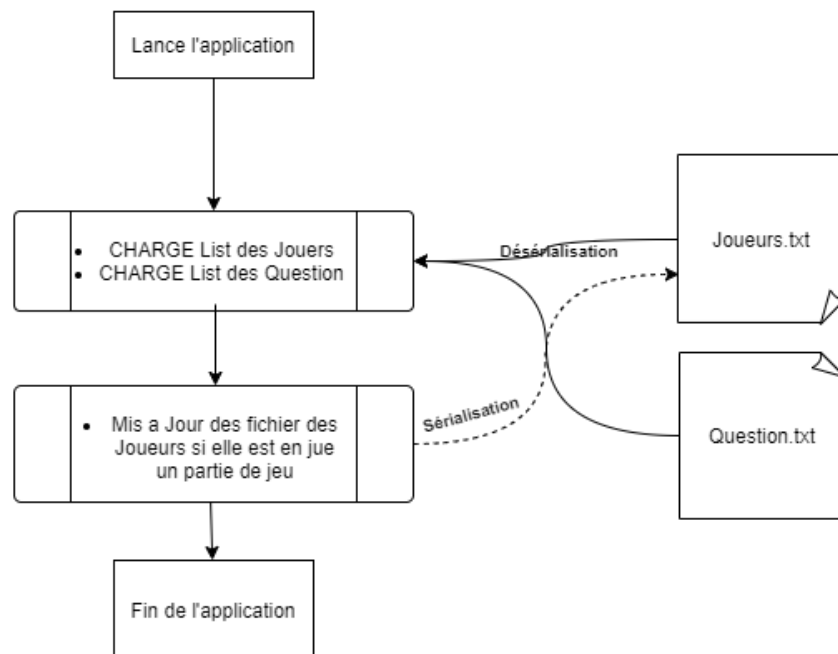
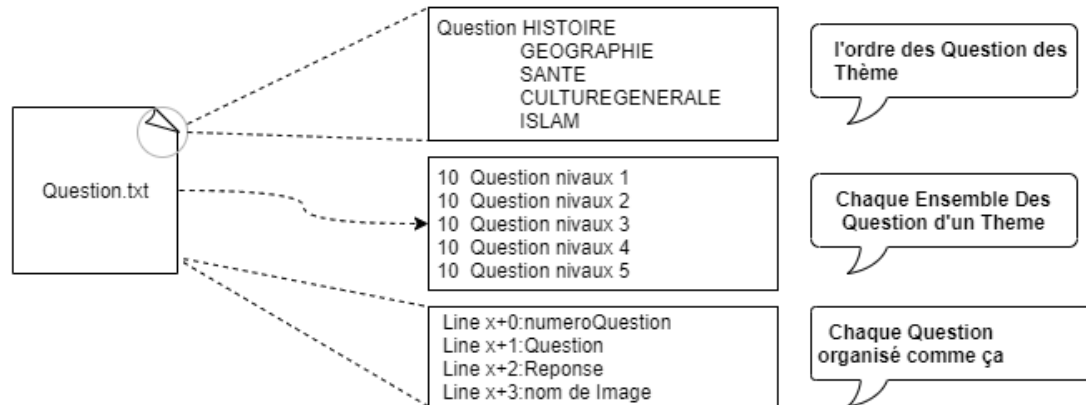
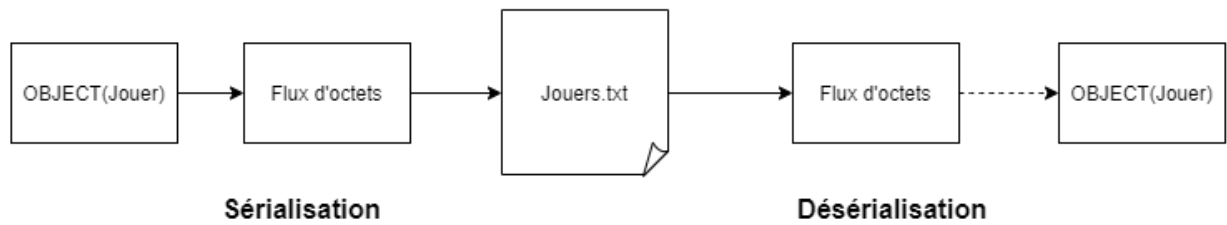


Figure 6: Schéma d'utilisation des Fichiers

Voici une description des tous les composant de les class des méthodes et des attributs et sa fonctionnalité :

# PACKAGE COM.USTHB. MODELES

## Class Eureka

Attribut	Rôle
<b><i>public static HashMap&lt;Integer, Joueur&gt; DicJoueur = new HashMap&lt;Integer, Joueur&gt;();</i></b>	<i>Regroupe Tous les Joueurs Inscrit dans le Jeu Eureka Elle charges par la méthode :« public static void chargeDicJoueurs() » dans le démarrage de l'application .</i>
<b><i>public static HashSet&lt;ThemeJeu&gt; ThemeDef = new HashSet&lt;ThemeJeu&gt;();</i></b>	<i>Regroupe-les différent thèmes définis dans le Jeu Des thèmes pour les Joueurs Enfant et des Joueurs Adulte elle charge par la méthode« public static void ChargerThemeDef() » dans le démarrage de l'application et après le chargement de liste question .</i>
<b><i>public static HashMap&lt;String, LinkedList&lt;Question&gt;&gt; listeQuestion = new HashMap&lt;String, LinkedList&lt;Question&gt;&gt;();</i></b>	<i>Regroupe des linkedList des Object Question des différent thème, chaque linkedList contient tous les question de le thème , elle charge par un fonction « public static void listeQuestionffile() » dans le démarrage de l'application .</i>

Méthodes	Rôle
<b><i>public static void ChargerThemeDef()</i></b>	<i>Dans cette méthode on parcourt les différent valeurs de l'énumérateur « ThemeType » par ordre et en parcourt les différent question chargé dans « listeQuestion » et en génère pour chaque type de Joueurs le thème spécifique et dans un coefficiente et un liste des question (la moitié des question pour un enfant (les question Impair) et l'autre pour adulte (les question Paire)) en ajoute les thème génère a le HashSet « ThemeDef ».</i>
<b><i>public static void listeQuestionffile()</i></b>	<i>Dans cette fonction en déclarée 5 LinkedList des différent thème contient des Question en parcourt le fichier « Question.txt » qui localisé dans les Resource de Project les différent question sont ordonnées HISTOIRE, GEOGRAPHIE, SANTE, CULTURE GENERALE, ISLAM en fait des condition des intervalle des question de chaque thème et le condition de détermine les question de quelle niveaux en ajoute les question a LinkedList spécifique a les condition définie après la fin de lecture de fichier en ajoute les LinkedList des questions a le hashmap des question « listeQuestion »</i>
<b><i>public static void chargeDicJoueurs()</i></b>	<i>Cette méthode permet de créer un HashMap&lt;Integer, Joueur&gt; et lire un joueur a partir de le fichier Joueurs apres mettre ce HashMap dans DicJoueurs.</i>
<b><i>public static void Inscription()</i></b>	<i>Cette méthode affiche la fenêtre InscriptionInt qui est Responsable a l'inscription d'un Joueur et Qui contient plusieurs composant Graphique pour récupérer les informations d'un Joueur.</i>
<b><i>public static void SeConnecter()</i></b>	<i>Cette méthode affiche la fenêtre SeConnecterInt Qui contient plusieurs composant Graphique permettent de garantir la connexion d'un Joueur a le Jeu Eureka.</i>

<b>public static int getScoreCourant()</b>	Cette méthode fait appelle à la méthode getTotalScore () en passons par la class PartieDeJeuInt et le joueur instancie qui localise dans la classe et el addition ce dernier a les score des partie précédant
<b>public void Initialisation()</b>	Cette fonction affiche la fenêtre Bienvenu qui contient les différent composant Graphique Responsable a l'affichage de liste des Joueur Inscrit , la connexion et l'Inscription d'un Joueur .
<b>public void AfficherInfosJoueursInscrits()</b>	Cette méthode crée un tableau des chaines de caractère de 4 columen dans les quels on peut ajoute tous les Joueurs existé dans DicJoueur
<b>public static void main(String[] args)</b>	Cette méthode principale fait appelle a les méthodes : «listeQuestionffile() » «ChargerThemeDef() » chargeDicJoueurs() , pour charge les Nécessaires pour Bon Fonctionnement de le Jeu et On appelle la méthode Initialisation() pour le Démarrage de la premier Fenêtre de l' application Eureka .

## PAKAGE COM.USTHB. MODELES

### Class Joueur

Attribut	Rôle
<b>private int numSeq;</b>	C'est le numéro séquentielle de Joueur
<b>private String nom;</b>	Le nom du Joueur
<b>private String prenom;</b>	Le Pre-nom de joueur .
<b>private String dateDeNaissance;</b>	La date de naissance de Joueur
<b>private String motDePasse;</b>	Le mot de passe de Joueur
<b>private int dernierNiveau;</b>	Le dernier niveau passe par le Joueur
<b>private LinkedList&lt;PartieJeu&gt; ListPartie;</b>	Contient les parties de jeu joue par le joueur

Méthode	Rôle
<b>public abstract ArrayList&lt;Question&gt; getQuestions(ThemeType thème);</b>	C'est une méthode redéfinie dans les class fils
<b>public String toString()</b>	La Redéfinition de la fonction prédéfinie dans java toString pour l'affichage des attributs d'un Joues
<b>public int getNumSeq()</b>	Pour Récupérer le numéro séquentiel du Joueur
<b>public void setNumSeq(int numSeq)</b>	Pour changer le numéro du Joueur
<b>public String getNom()</b>	Pour Récupérer le nom du Joueur
<b>public void setNom(String nom)</b>	Pour modifie le Nom de Joueur
<b>public String getPrenom()</b>	Pour Récupérer le Prénom de Joueur
<b>public void setPrenom(String prenomm)</b>	Pour modifier le Prénom de Joueur
<b>public String getDateDeNaissance()</b>	Pour Récupérer la date de nissanaise de Joueur
<b>Public void setDateDeNAissance(String datedeNaissancee)</b>	Pour modifier la date de nissanaise de Joueur
<b>public int getDernierNiveau()</b>	Pour Récupérer le dernier niveau de Joueur
<b>Public void setDernierNiveau(int dernierNiveauu)</b>	Pour modifier le dernier niveau de joueur dans le Jeu
<b>public String getMotDePasse()</b>	Pour Récupérer le mot de passe de Joueur
<b>public void setMotDePasse(String motDePasse)</b>	Pour Modifier le Mot de passe de Joueur

<b>public void Jouer(ThemeJeu themeDeJeu)</b>	cette méthode fait appelle à les 2 méthodes <i>ListeQuestionffile()</i> et <i>ChargerThemeDef()</i> après , il cherche si la partie de jeu est terminée ou pas , si c'est le cas on relance la partie , sinon on instancier une nouvelle
<b>public LinkedList&lt; PartieJeu&gt; getListPartie()</b>	Pour Récupérer la liste des parties réaliser par le Joueur
<b>public int getTotalScore()</b>	Pour retourner le score total des partie de jeu réaliser par le joueur ;elle parcourt la liste et récupérer le score de chaque partie et fait la somme

### Class Enfant

Méthode	Rôle
<b>Public</b> <i>LinkedList&lt;Question&gt;</i> <b>getQuestions(String theme)</b>	C'est une méthode redéfinie qui retourne un <i>LinkedList</i> des question pour un Joueur Enfant a un thème passe en paramètre ,On parcourt l'ensemble des Thèmes définies dans la class Eureka «Themedef» et on test les questions définie dans chaque thème si elle est un objet de la class <i>QuestionEnfant</i> On retourne cette dernier .

### Class Adulte

Méthode	Rôle
<b>Public</b> <i>LinkedList&lt;Question&gt;</i> <b>getQuestions(String theme)</b>	C'est une méthode redéfinie qui retourne un <i>LinkedList</i> des questions pour un Joueur Adulte a un thème passe en paramètre. On parcourt l'ensemble des Thèmes définies dans la class Eureka «Themedef» et on test les questions définies dans chaque thème si elle est un objet de la class <i>QuestionAdult</i> si oui On retourne cette dernier .

### Class ThemeJeu

Attribut	Rôle
<b>private ThemeType typeDeTheme;</b>	Un attribut qui contient une valeur des thèmes définie dans enum <i>HISTOIRE</i> , <i>GEOGRAPHIE</i> , <i>SANTE</i> , <i>CULTUREGENERALE</i> , <i>ISLAM</i>
<b>private String libelle;</b>	Contient l'abréviation d'un thème par exemple si on a l'attribut <i>typeDeTheme</i> contient « <i>HISTOIRE</i> » Alors <i>libelle</i> contient « <i>HIS</i> » les trois premières lettres de chaque thème
<b>private int coefficient;</b>	Contient le coefficient spécifique pour chaque thème en utilisant ce dernier pour le calcule le point gagné dans un partie de Jeu
<b>private LinkedList&lt;Question&gt; Liste;</b>	Centienne un ensemble des question spécifique et le thème et le Joueur soit enfant ou Adulte

Méthode	Rôle
<b>public ThemeType getTypeDeTheme()</b>	Retourner le thème
<b>public void setTypeDeTheme(ThemeType typeDeTheme)</b>	Modifier le theme
<b>public String getLibelle()</b>	Retourner l'Abréviation de thème
<b>public void setLibelle(String libelle)</b>	Modifier l'abréviation de thème
<b>public int getCoefficient()</b>	Retourne le coefficient spécifique a ce thème
<b>public void setCoefficient(int coefficient)</b>	Modifier la coefficient
<b>public LinkedList&lt;Question&gt; getListe()</b>	Retourner la liste des Question
<b>public void setListe(LinkedList&lt;Question&gt; quesE)</b>	Modifier la liste des Question

### Class Question

Attribut	Rôle
<b>Private static int numeroSeq ;</b>	C'est un numéro séquentielle spéciale .
<b>private String theme;</b>	Contient l'abréviation de thème de Question
<b>private String libelle ;</b>	Contient le nom complet de Question
<b>private String numero;</b>	Le numéro de la question compose de trois premières lettres de thème de question puis un numéro
<b>private String image;</b>	Contient le nom de l'image pour la question
<b>protected String reponse ;</b>	Contient la Réponse de la Question
<b>private int niveau;</b>	Contient le niveau de la question

Méthode	Rôle
<b>public String getLibelle()</b>	Retourne le nom complet de thème de question
<b>public void setLibelle(String lb)</b>	Modifier les nom de question
<b>public String getNumero()</b>	Retourne le numéro de la question
<b>public void setNumero(String numero)</b>	Modifier le nom de la question
<b>public String getImage()</b>	Retourne le nom de l'image de question
<b>public void setImage(String image)</b>	Modifier le nom d'image
<b>public String getReponse()</b>	Retourne la Réponse de la question
<b>public void setReponse(String reponse)</b>	Modifier la Réponse
<b>public int getNiveau()</b>	Retourne le niveau de la question
<b>public void setNiveau(int niveau)</b>	Modifier le niveaux de la question
<b>public int getNBPoints()</b>	Retourne le nombre des points pour cette question par niveaux
<b>public String getTheme()</b>	Retourne le thème de la question
<b>public void setTheme(String theme)</b>	Modifier le theme de la question

### Class QuestionEnfant.

Méthode	Rôle

### Class QuestionAdulte

Méthode	Rôle



## Class PartieJeu

Attribut	Rôle
<b>public static int cp=1</b>	Le compteur du numéro de la partie
<b>private int numeroPartie</b>	Le numéro de la partie
<b>private ThemeType themeSelect</b>	Contient le thème la Jeu qui sélection par le Jouer
<b>private String numQuestion</b>	Contient le numéro de la question pour l'affiche
<b>private static StringBuffer reponseEnCours</b>	Contient la Réponse du Joueur en cour
<b>private int score</b>	Contient le Score du Joueur dans cette partie de Jeu
<b>private Question question;</b>	Contient la question
<b>private boolean partieTermine;</b>	Pour que elle soit elle termine la partie ou non

Méthode	Rôle
<b>private StringBuffer init()</b>	Pour initialise la ReponseEnCours dans le début de Jeu pour un suite des * et ' si on a des espaces.
<b>public int getNumeroPartie()</b>	Retourne le numéro de la Partie Jeu
<b>public void setNumeroPartie(int numeroPartie)</b>	Modifier le numéro de la Partie jeu
<b>public ThemeType getThemeSelect()</b>	Retourne le Thème Sélection
<b>public void setThemeSelect(ThemeType themeSelect)</b>	Modifier le Theme .
<b>public String getNumQuestion()</b>	Retourner le numéro de la question
<b>public void setNumQuestion(String numQuestion)</b>	Modifier le numéro de la question
<b>public StringBuffer getReponseEnCours()</b>	Retourner la réponse en Cours
<b>public void setReponseEnCours(StringBuffer reponse)</b>	Modifier la réponse en Coures
<b>public static int getScore()</b>	Retourne le score de Joueur dans la partie
<b>public void setScore(int sc)</b>	Pour Modifier les Score de Joueur dans la partie de Jeu
<b>public int indexChar(char c)</b>	Pour Return les Index d'un caractère de Réponse
<b>public void checkCaractere(char c)</b>	C'est l'un des importantes méthodes de cette jeu . Sa principe consiste a vérifier si le caractère c appartient a la réponse en cours , si c'est le cas on remplace le * correspondante par ce caractère , sinon on appelle la méthode repaint() de la classe Potence . Si l'etat de potence = 8 et le Joueur n'a pas trouvé la réponse on appelle la méthode LosePartie() de cette classe .
<b>public void updatePlayer(Joueur joueur)</b>	Qui permet d'ajoute la partie jeu courante a la liste des parties réaliser par le Joueur.
<b>public void LosePartie(Joueur joueur)</b>	Permet de fini la partie elle affiche bienvenu .
<b>public boolean isPartieTermine()</b>	Pour voir si cette partie est fine ou non.
<b>Private LinkedList&lt;Question&gt; QuestionNiveaux(LinkedList&lt;Question&gt; L,int nivea) ;</b>	Cette méthode permet de récupérer les questions d'un niveau précise a partir d'une LinkedList des Questions .
<b>Public int NbParte(LinkedList&lt;ParieJeu&gt; l) ;</b>	Pour compter le nbr des partie de jeu d'un thème précise réalisées par un joueur .
<b>public void setPartieTermine(boolean partieTermine)</b>	Pour Changer l'état de la partie a termine .



# PAKAGE COM.USTHB. DESSING

## Bienvenu

- Le Premier fonction pour l'appelé aux la Fonction seconnect dans la classe Eureka se fait de la manier suivant :
- Le principe ici quand l'utilisateur click sur le Button se connect elle fait appel a la fonction Seconnect Qui Affiche la Fenêtre SeConnect pour le Saisies les bonnes informations pour commence un parte de Jeu

```
JButton btnseconnect = new JButton("Se Connect");
btnseconnect.setBounds(376, 484, 231, 29);
eureka_bienvenu.getContentPane().add(btnseconnect);
btnseconnect.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        try {
            EUREKA.SeConnector();
        } catch (Exception e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(null, e);
        }

    }
});
```

- La deuxième fonction pour l'appelé aux la Fonction Inscription dans la classe Eureka se fait de la manier suivant :
- Le principe ici quand l'utilisateur click sur le Button Inscription elle fait appel à la fonction Inscription Qui Affiche la Fenêtre Inscription pour le Saisies les bonnes informations pour l'inscription après la connexion et commence un parte de Jeu

```
JButton btnsinscri_cn = new JButton("Inscription");
btnsinscri_cn.setBounds(376, 529, 231, 29);
eureka_bienvenu.getContentPane().add(btnsinscri_cn);
btnsinscri_cn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        try {
            EUREKA.Inscription();
        } catch (Exception e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(null, e);
        }

    }
});
```

- La troisième fonction pour affiche la Fenêtre A Propose de l'application Eureka
- Le principe ici quand l'utilisateur click sur le Button A propos elle crée un Object window est-elle fait l'affichage

```
JButton btnApropose = new JButton("A propos");
btnApropose.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        Apropose window = new Apropose();
        Bienvenu.eureka_bienvenu.setVisible(false);
        window.wApropose.setVisible(true);

    }
});
```

- La quatrième fonction pour affiche la Fenêtre A Help de l'application Eureka
- Le principe ici quand l'utilisateur click sur le Button Help ( ? ) elle crée un Object window est-elle fait l'affichage

```

JButton btnhelp = new JButton("Help ( ? )");
btnhelp.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        Help window = new Help();
        Bienvenu.eureka_bienvenu.setVisible(false);
        window.wHelp.setVisible(true);

    }
});

```

- La Cinquième fonction pour Affiche la Fenêtre LJoueursInscr de l'application Eureka
- Le principe ici quand l'utilisateur Click sur le Button Liste Joueurs elle crée un Object window est-elle fait l'affichage

```

JButton btnListJoueurs = new JButton("List Joueurs");
btnListJoueurs.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        LJoueursInscr window = new LJoueursInscr();
        Bienvenu.eureka_bienvenu.setVisible(false);
        window.WListJoueurs.setVisible(true);

    }
});

```

### SeConnecterInt

- la Fonction Pour la affichage de la Fenêtre d'inscription
- Le principe ici quand l'utilisateur click sur le Button Inscription elle crée un Object window est-elle fait l'affichage de la fenêtre inscription pour le saisie des informations

```

JButton btnsinscri_cn = new JButton("Inscription");
btnsinscri_cn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        InscriptionInt window = new InscriptionInt();
        wn_seconnect.setVisible(false);
        window.wn_Inscription.setVisible(true);

    }
});

```

- Le principe ici quand le utilisateur click sur le button cannel elle supprime le contenu des textField et passwordfiéd de la fenêtre Se connect

```

JButton btnCannel = new JButton("Cancel");
btnCannel.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        textnomFcxn.setText("");
        textpasswordFcxn.setText("");
    }
});

```

- Dans la fonction de la page suivant Quand le Jouer Clicksur le Button Seconnect On test la validité de champ de saisie, On vérifie si le Joueur est déjà existe dans les Joueurs inscrit(DicJouer) on vérifie la liste des parties si elle est vide ou la dernier partie est termine on affichera la fenêtre **SelectTheme** sinon en instance la dernier partie par la appel a la fonction Jouer() qui fait ses traitement . Si le joueur n'existe pas on affiche un JoptionPane .

```

JButton btnseconnect = new JButton("Se Connecter");
btnseconnect.addActionListener(new ActionListener() {
    @SuppressWarnings("deprecation")
    public void actionPerformed(ActionEvent e) {

        Boolean JoueurExiste = false;
        // le cas de un au plusieurs champ sont vide
        if (textnomFcx.getText().equals("") && textpasswordFcx.getText().equals(""))
            || textnomFcx.getText().equals("") || textpasswordFcx.getText().equals("") {
            JOptionPane.showMessageDialog(null, "S'il vous plait Donnez tous les information",
                "EUREKA Inscription", JOptionPane.WARNING_MESSAGE);

        } else { // le cas de validation des champ

            // Cherche le Jouer dans le Dictionnaire Jouers
            Joueur joueur = null;
            for (Joueur j : EUREKA.DicJoueur.values()) {
                if (j.getNom().equals(textnomFcx.getText())
                    && j.getMotDePasse().equals(textpasswordFcx.getText())) {
                    // Jouer Existe
                    JoueurExiste = true;
                    joueur = j;
                    break;
                }
            }
            // Trite le cas de Jouer existe
            if (JoueurExiste == true) {
                // Here we make setting of Scoore and the niveaux
                LinkedList<PartieJeu> L = new LinkedList<PartieJeu>();
                L.addAll(joueur.getListPartie());
                // LE Cas de Affiche un Fentre Pour Choiser un Theme
                // Quand elle n'est Jouer pas ou que elle termine un Theme
                // un Theme Contier 25 Partie Jeu danc c un multiple de 25

                if (L.isEmpty() || L.getLast().isPartieTermine() == true
                    && L.getLast().NbPartie(joueur.getListPartie()) % 25 == 0) {
                    // Le Cas d'un Nouvelle Jouer ou Elle Termine Tous Les Partie d'un theme

                    JOptionPane.showMessageDialog(null, "Bienvenu " + textnomFcx.getText()
                        + " sur EUREKA \n\nOn vous souhaite une bonne partie");
                    LocalDate l = LocalDate.now();
                    int age = l.getYear() - (joueur.getDateDeNaissance().getYear() + 1900);

                    if (age >= 18) { // Lancer la fentre des themes
                        SelectThemeInt window = new SelectThemeInt(new Adulte(joueur.getNom(),
                            joueur.getPrenom(), joueur.getDateDeNaissance(),
                            joueur.getMotDePasse(), joueur.getNumerojouer(), joueur.get DernierNiveau(),
                            joueur.getListPartie()));
                        wn_seconnect.setVisible(false);
                        window.wn_choisetheme.setVisible(true);
                    } else {
                        SelectThemeInt window = new SelectThemeInt(new Enfant(joueur.getNom(),
                            joueur.getPrenom(), joueur.getDateDeNaissance(),
                            joueur.getMotDePasse(), joueur.getNumerojouer(), joueur.get DernierNiveau(),
                            joueur.getListPartie()));
                        wn_seconnect.setVisible(false);
                        window.wn_choisetheme.setVisible(true);
                    }
                }
            } else {

                if (!L.getLast().isPartieTermine()) {
                    wn_seconnect.setVisible(false);
                    try {
                        // appel Pour Instance la Dernier Partie Qui Pas Complite
                        joueur.Jouer(L.getLast().getThemeSelect());

                    } catch (Exception prob) {

                        JOptionPane.showMessageDialog(null, "Error Instance La Dernier
Partie",

                                "EURKA | ALTER", JOptionPane.ERROR_MESSAGE);

                    }
                }
            }
        } else {
            JOptionPane.showMessageDialog(null,
                "Votre nom ou votre mot de passe n'est pas correct \nEssayez une autre
fois \n Si vous n'avez pas d'un
compte, vous pouvez Inscri maintenant",
                "EURKA | ALTER", JOptionPane.WARNING_MESSAGE);
        }
    }
});

```

## InscriptionInt

Dans cette partie On test la validité de champ de saisie, On vérifie si le Joueur déjà existe dans les Joueurs inscrit sinon On calcule l'âge de Joueur ,si l'âge est supérieur a 18 On instance un Object Joueur adulte sinon un Joueur enfant après On crée un fichier pour cette joueur inscrit est on afficher pour lui la Fenêtre de connexion .

```
 JButton btninscri = new JButton("Inscription");
 btninscri.addActionListener(new ActionListener() {
     @SuppressWarnings({ "deprecation", "static-access", "unlikely-arg-type" })
     public void actionPerformed(ActionEvent arg0) {
         // pour get courrant annee
         LocalDate l = LocalDate.now();
         // Define Un Format de la Date "dd-MM-YYYY"
         DateFormat df = new SimpleDateFormat("dd-MM-YYYY");
         //boolean existe = false;
         if (textnomFin.getText().equals("") || textpasswordFin.getText().equals("") || textfiprenom.getText().equals("") || df.format(
             DatedeneIn.getDate()).equals("")) {
             JOptionPane.showMessageDialog(null, "S'il vous plait Donnez tous les information",
                 "EUREKA Inscription", JOptionPane.WARNING_MESSAGE);

             // ici on traite le cas de date superieur ou egala la data courrant
         }else {
             // elle est pas existe en vas ajoute a la liste des jouers
             // Calule l'age de le Joueur Inscri
             boolean JoueurExiste=false;
             try {
                 EUREKA.chargeDicJoueurs();
             } catch (Exception e) {
                 e.printStackTrace();
             }

             for (Joueur j : EUREKA.DicJoueur.values()) {
                 if(j.getNom().equals(textnomFin.getText()) && j.getPrenom().equals(textfiprenom.getText()) && j.getDateDeNaissance().
                     equals(df.format(DatedeneIn.getDate())))) {
                     JoueurExiste=true;
                     joueur = j;
                     break;
                 }
             }
             if(JoueurExiste) {
                 JOptionPane.showMessageDialog(null, "Vous avez un compte deja !!\n Essayez de Connecter directement");
             }else {
                 int age = l.getYear() - (DatedeneIn.getDate().getYear()+1900);
                 try {
                     FileReader fr = new FileReader ("Resource/NumFile.txt");
                     BufferedReader br = new BufferedReader(fr);
                     String ligne=null,num=null;
                     while((ligne = br.readLine())!= null ) {
                         num = ligne;
                     }
                     int numero = Integer.parseInt(num);
                     fr.close();
                     br.close();
                     LinkedList<PartieJeu> List = new LinkedList<PartieJeu>();
                     if(age >=18) {
                         joueur = new Adulte(textnomFin.getText(),textfiprenom.getText(),
                             DatedeneIn.getDate(), textpasswordFin.getText(),1, numero+1,List);
                         JOptionPane.showMessageDialog(null, "elle est instance un object adulte");
                     } else { if(age >0){
                         joueur = new Enfant(textnomFin.getText(),textfiprenom.getText(),
                             DatedeneIn.getDate(), textpasswordFin.getText(),1,numero+1,List);
                         JOptionPane.showMessageDialog(null, "elle est instance un object enfant");
                     }else {
                         JOptionPane.showMessageDialog(null, "Oops ! Votre date de naissance"
                             + " n'est pas correct ! Essayez d'inscrire une autre fois!");
                         joueur = null;
                     }
                 }
                 if(joueur != null) {
                     FileOutputStream f = new FileOutputStream(new File("Resource/Joueurs"+joueur.getNumSeq()+".txt"));
                     ObjectOutputStream o = new ObjectOutputStream(f);
                     //Ecriture des objets sur un fichier
                     FileWriter fw = null;
                     BufferedWriter bw = null;
                     PrintWriter pw = null;
                     try {
                         fw = new FileWriter("Resource/NumFile.txt", true);
                         bw = new BufferedWriter(fw);
                         pw = new PrintWriter(bw);
                         pw.print("\n"+joueur.getNumSeq());
                     }finally {
                         try {
                             pw.close();
                             bw.close();
                             fw.close();
                         } catch (IOException io) {
                         }
                     }
                     o.writeObject(joueur);
                     f.close();
                     o.close();
                 }
                 // AFFICHE LES INFORMATION :
                 JOptionPane.showMessageDialog(null,
                     "Bon " + textnomFin.getText().toUpperCase()
                     + " +\nvotre compte dans Eureka a Ã©tÃ© crÃ©Ã© avec succÃ©s",
                     "EUREKA Inscription", JOptionPane.PLAIN_MESSAGE);
                 SeConnecterInt window = new SeConnecterInt();
                 wn_Inscription.setVisible(false);
                 window.wn_seconnect.setVisible(true);
             } } catch (IOException i) {
                 i.printStackTrace();
             }
         }
     }
 });
```

## SelectThemeInt

- Pour cette fonction si le joueur click sur le Button déconnecte elle affiche la Fenêtre de Connexion

```
btndisconnect.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        SeConnectorInt window = new SeConnectorInt  
();  
        wn_choisetheme.setVisible(false);  
        window.wn_seconnect.setVisible(true);  
    }  
});
```

- Dans cette fonction si le Joueur click sur le Button Okay on instance un nouveau thème

En fait appel a la Fonction Jouer dans la Class Joueur Qui Fait ces traitements et lance une partie de Jeu.

```
btnChoise.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
  
        ThemeJeu theme = new ThemeJeu(comboBoxJeu.getSelectedIndex().toString());  
        try {  
            wn_choisetheme.setVisible(false);  
            joueur.Jouer(theme);  
        } catch (Exception e) {  
  
            e.printStackTrace();  
        }  
    }  
});
```

## PartieDeJeuInt

- Dans cette fonction si l'utilisateur click sur valide elle fait appel à checkCaracter qui fait ses traitements après on fait la mise a jour de la réponse en cours de la TextField textFchiffrement

```
public void actionPerformed(ActionEvent e) {  
    try {  
        // verifier si le caractere correspond a un lettre de la reponse encours  
        prt.CheckCaractere((char) textFieldcharsize.getText().charAt(0), joueur);  
    } catch (Exception givechar) {  
        givechar.printStackTrace();  
    }  
    textFchiffrement.setText(prt.getReponseEnCours().toString());  
    textFieldcharsize.setText("");  
    if (prt.getReponseEnCours().toString().equals(prt.getQuestion().getReponse())) {  
        PartieDeJeuInt.potence.setTrouve(true);  
        prt.PasserNiveau(List, theme);  
    }  
}
```

Après on mis à vide pour la mise d'un autre caractère. Après on vérifie si la réponse en cours et la réponse sont le même si oui on mis a jour le potence elle est trouve et on fait appelle a la fonction passerNiveaux () avoir comme paramètre la liste des questions et le thème qui soit passe a un autre question ou a un niveau .

## LJoueursInscr

Dans le constructeur de cette classe on fait appel à la fonction AfficherInfosJoueursInscrits() qui fait les traitements nécessaires et le chargement de Tableaux Joueurs .

- Dans cette fonction quand le Joueur click sur le Button Revenir On appel la class Bienvenu

```
JButton btnGoHome = new JButton("Revenir");  
btnGoHome.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
  
        Bienvenu win = new Bienvenu();  
        Bienvenu.eureka_bienvenu.setVisible(true);  
        LJoueursInscr.WListJoueurs.setVisible(false);  
    }  
});
```



## Résumer les structures de données globales utilisées :

On a utilisé les structure des données pour la quel ses avantage qui en situe dans ce tableau la :

<b>HashSet</b>	<p>La classe HashSet est membre du framework de collections Java. Il implémente l'interface Set. Les HashSets sont utilisés pour stocker une collection d'éléments uniques.</p> <ul style="list-style-type: none"><li>• HashSet ne peut pas contenir de valeurs en double.</li><li>• HashSet autorise une valeur nulle.</li><li>• HashSet est une collection non ordonnée. Il ne conserve pas l'ordre dans lequel les éléments sont insérés.</li></ul>
<b>HashMap</b>	<p>Java HashMap est une implémentation basée sur une table de hachage de l'interface Java Map. Une carte, comme vous le savez peut-être, est une collection de paires clé-valeur. Il mappe les clés aux valeurs.</p> <ul style="list-style-type: none"><li>• Un HashMap ne peut pas contenir de clés en double.</li><li>• Java HashMap autorise les valeurs nulles et la clé nulle.</li><li>• HashMap est une collection non ordonnée. Il ne garantit aucun ordre spécifique des éléments</li></ul>
<b>LinkedList</b>	<p>Java LinkedList est une implémentation de liste doublement liée des interfaces Java List et Deque. Il fait partie des collections Java. Java LinkedList maintient l'ordre d'insertion des éléments.</p> <ul style="list-style-type: none"><li>• LinkedList peut avoir des valeurs en double et nulles.</li><li>• La classe LinkedList implémente les interfaces Queue et Deque. Par conséquent, il peut également être utilisé comme file d'attente, retrait ou empilement.</li><li>• LinkedList est plus rapide pour des opérations d'insertion et de suppression</li></ul>

## Outils de développement utilisés

- Eclipse IDE
- VS CODE
- WindowBuilder



## Outils de design utilise

- ADOBE ILLUSTRATION
- ADOBE PHOTOSHOP



## Outils de Communication et Travail

- GITHUB



- TELEGRAM



# Conclusion

## Remarques sur ma réalisation

- On a utilisé la sérialisation dans le fichier Joues seul dans le Question on a introduit directement les Question dans le Fichier ordonnes thème par thème.
- On a utilisé Just un truck pour recopier les mots de passe.
- On a utilisé JLabel pour l'emplacement des images dans les Graphique
- Pour les Question qui n'est pas une image on a mètre un image « think.jpg »

## Ce que je propose comme amélioration de l'application

- Ajoute d'un Timer pour trouver tous les caractères de la bon Réponse dans les niveaux avance.
- Remplace l'utilisation des fichiers a l'utilisation des bases de donnees parce que les base de donnees plus sécurité et facile a trouvé les donnees.
- Ajoute une méthode pour recopier le mot de passe si le Jouer oublier le mot de passe.
- Ajoute un choix a le Joueur pour jouer sur des question 3 langage Arabica, English.
- Ajoute un Button dans la parte Jeu pour affiche un Indication de la réponse.

## Ce qu'On a appris du projet

- Le concept de décompose les différent problème en des petit problème.
- Les concepts de faire tous les choisie comme des objets.
- Les différent concept de POO (Encapsulation, héritage, Polymorphisme).
- Comment travaille en Group et comment Compose le projet en Group.
- L'utilisation de GitHub la Création des Branch et Merge les Class.
- L'Utilisation des sérialisations des Object.
- L'utilisation de WindowBuilder pour faire les différent Composant Graphique.
- La Connexion à la base de donnees (On a commencé en début par utilise la base de donnees après On a changé à les fichier).