

Interface graphique Swing

Package javax.swing.*

Projects

- AppDebug
- AppliJeu
 - Source Packages
 - <default package>
 - applijeu
 - fenetre
 - AjoutHeros.java
 - JAjoutJoueur.java
 - JMenu.java
 - lesException
 - option
 - Test Packages
 - Libraries
 - Test Libraries
 - Bd
 - BD-AppelJava
 - CVDATPTEST
 - Source Packages
 - Exception
 - cvdatptest
 - Bibliotheque.java
 - CVDATPTEST.java
 - ListePersonne.java
 - Livre.java
 - Personne.java
 - Test Packages
 - <default package>
 - BibliothequeTest.java
 - ListePersonneTest.java
 - Libraries
 - Test Libraries
 - Devoir
 - Geometrie
 - GestionException

Source Design History

To change layout manager of a container use Set Layout submenu from its context menu.

Création d'un héros

Nom

Points de vie

Défense

Poids

Agilité

Ajouter

Palette

Swing Containers

- Panel
- Tabbed Pane
- Split Pane
- Scroll Pane
- Tool Bar
- Desktop Pane
- Internal Frame
- Layered Pane

Swing Controls

- Label
- Button
- Toggle Button
- Check Box
- Radio Button
- Button Group
- Combo Box
- List
- Text Field
- Text Area
- Scroll Bar
- Slider
- Progress Bar
- Formatted Field
- Password Field
- Spinner
- Separator
- Text Pane
- Editor Pane
- Tree
- Table

Swing Menus

Swing Windows

Swing Fillers

AWT

Beans

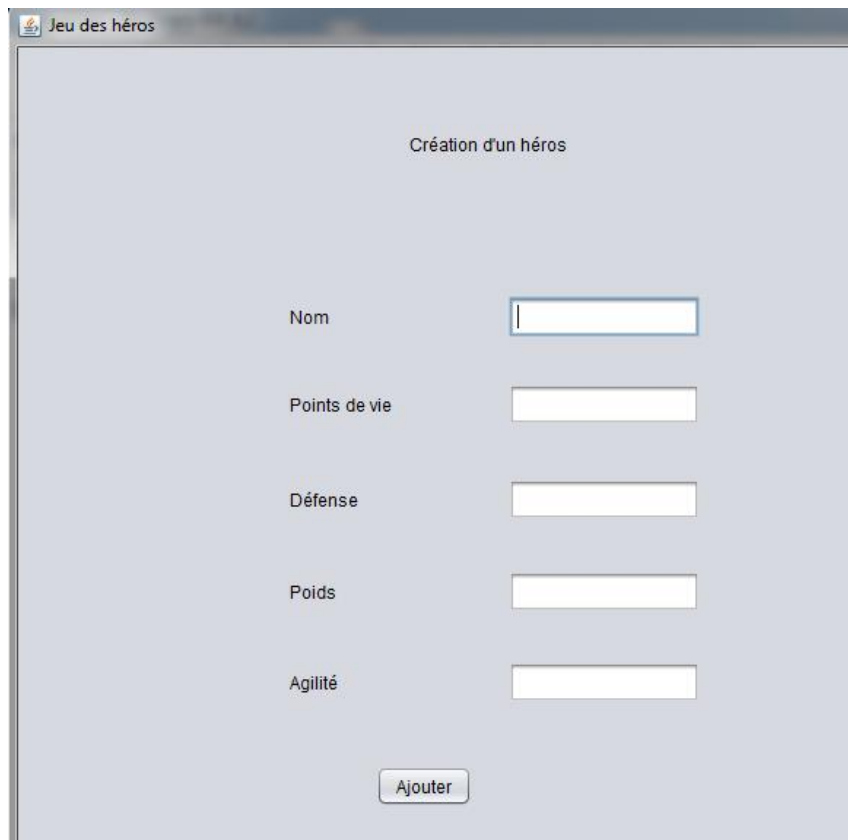
Java Persistence

AjoutHeros.java - Properties

Properties

Name	AjoutHeros
Extension	java
File Size	9388
Modification Time	2 mai 2018 17:39:29
All Files	D:\Dropbox\NetBeansProject\AppliJ...
Classpaths	
Compile Classpath	
Runtime Classpath	D:\Dropbox\NetBeansProject\AppliJ...
Boot Classpath	C:\Program Files\Java\jdk1.8.0_131\...

Création de JFrame



The screenshot shows a Java Swing window titled "Jeu des héros". Inside the window, the text "Création d'un héros" is centered at the top. Below this, there are five labels on the left, each followed by a text input field on the right: "Nom", "Points de vie", "Défense", "Poids", and "Agilité". At the bottom center of the window, there is a button labeled "Ajouter".

- Chaque JFrame possède une méthode main (Run ou classe de démarrage)
- Propriétés



Properties	
defaultCloseOperation	DISPOSE
title	Jeu des héros

Classes - interface graphique

nomClasse	Permet de gérer
JFrame	Une fenêtre
JButton	Un bouton
JTextField	Un champ de saisie
JRadioButton	Un bouton de sélection
JComboBox	Une liste déroulante
JCheckBox	Une case à cocher
JMenu	Un menu affiché dans la fenêtre
JDialog	Une boîte de dialogue

Comportement des objets

Propriétés

Properties	Binding	Events	Code
[-] Properties			
editable		<input checked="" type="checkbox"/>	...
background		<input type="checkbox"/> [255,255,255]	...
columns		0	...
document		<default>	...
font		Tahoma 11 Plain	...
foreground		<input checked="" type="checkbox"/> [0,0,0]	...
horizontalAlignment		LEADING	...
text			...
toolTipText			...
[-] Other Properties			

Evènements

Properties	Binding	Events	Code
[-] Events			
actionPerformed		<none>	...
ancestorAdded		<none>	...
ancestorMoved		<none>	...
ancestorMoved		<none>	...
ancestorRemoved		<none>	...
ancestorResized		<none>	...
caretPositionChanged		<none>	...
caretUpdate		<none>	...
componentAdded		<none>	...
componentHidden		<none>	...

Gestion des propriétés

- Gestion par l'interface graphique de l'environnement de développement
- Gestion dans le code avec les accesseurs sur les propriétés

Génération de code



```
public class AjoutHeros extends javax.swing.JFrame {

    /**
     * Creates new form AjoutHeros
     */
    public AjoutHeros() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jTextNom = new javax.swing.JTextField();
        jTextPoints = new javax.swing.JTextField();
        jTextDefense = new javax.swing.JTextField();
        jTextPoids = new javax.swing.JTextField();
        jTextAgilite = new javax.swing.JTextField();
        jButtonAjout = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
    }
}
```

← Non modifiable

Gestion des objets

Classe Infos

- Créer la classe Infos
- Créer 2 listes public static à initialiser :
 - lesHeros
 - lesJoueurs

Utilisation

- Dans les fenêtres
- Importer la classe
- Utilisation des listes
 - Infos.lesHeros.methode
 - Infos.lesJoueurs.methode

Exemple

Pour accéder à la valeur d'un champ JTextField :
`nomChamp.getText();`

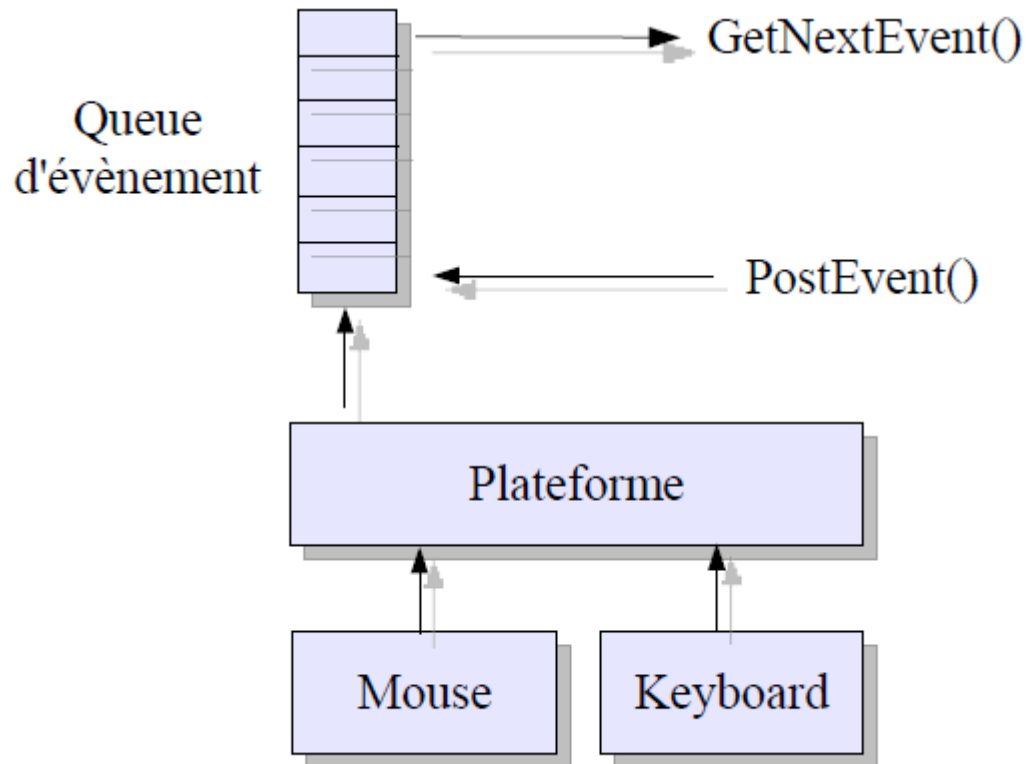
- Créer un package fenetres
- Créer la fenêtre JAjoutHeros qui permet d'ajouter un héros de terre au jeu
- Un clic sur le bouton Ajouter doit créer un objet de la classe HerosTerre et l'ajouter à la liste lesHeros

The screenshot shows a Java Swing window titled "Jeu des héros". Inside the window, there is a section titled "Création d'un héros". Below this title, there are five text input fields, each preceded by a label: "Nom", "Points de vie", "Défense", "Poids", and "Agilité". A blue arrow points to the "Nom" input field. At the bottom of the form, there is a button labeled "Ajouter".

Exemple d'évènements

Evènement	Action associée
actionPerformed	Clic sur l'objet
focusGained/focusLost	Focus sur l'objet ou focus perdu par l'objet
keyPressed/keyReleased	Touche appuyée ou relachée
mouseClicked/mouseEntered/mouseExited...	Actions de la souris sur l'objet

Traitement des évènements



Principe

- Création d'un écouteur (listener) en attente d'un évènement sur une ressource
- Une action de l'utilisateur est captée par le listener correspondant qui déclenche le programme associé

Génération automatique avec Swing

```
jButtonAjout.setText("Ajouter");  
jButtonAjout.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButtonAjoutActionPerformed(evt);  
    }  
});
```

Ajout d'un listener

Définition de l'action
écoutée

Méthode exécutée lorsque
le listener détecte l'action

Capter des évènements

- Plusieurs évènement peuvent être captés
- Action réalisée : clic
- Focus sur ou hors de l'objet
- Actions de la souris :
 - Bouton pressé
 - Déplacement
 - Pointeur entré dans une zone
 - Pointeur sortir d'une zone

Exemple

- Faire afficher l'aide lors d'un survol des boutons (évènement FocusGained)
- Propriété ToolTipText à modifier

Exercice

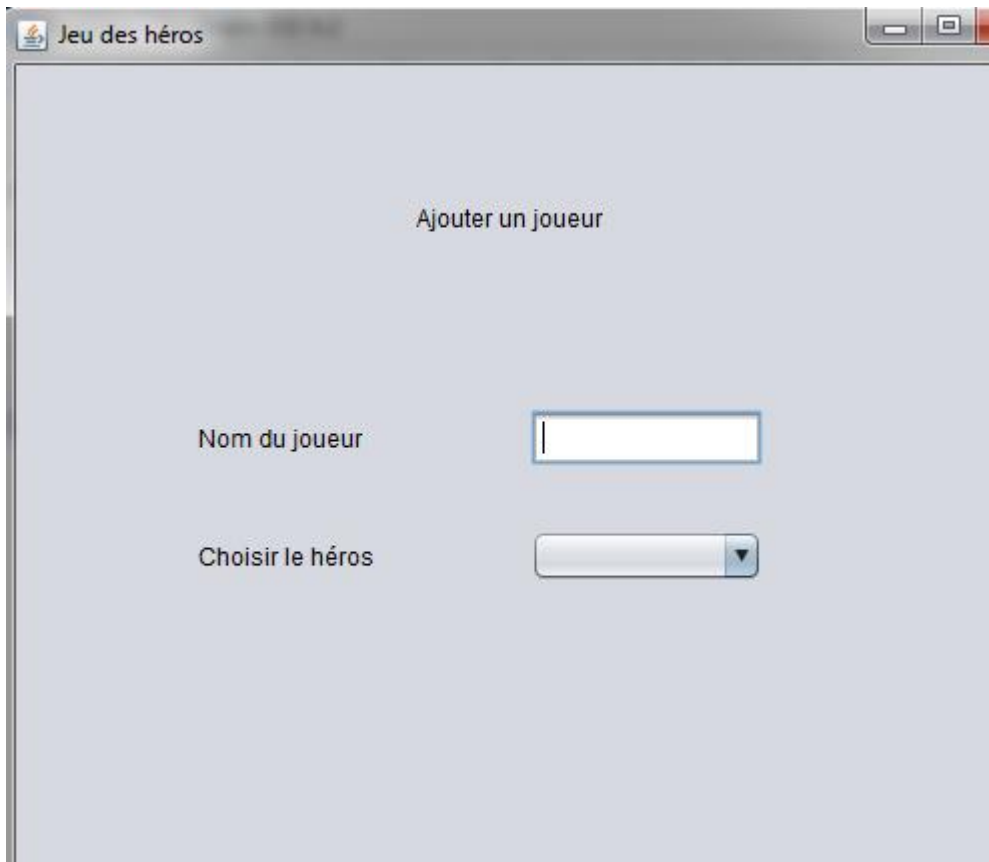
- Ajouter une aide dans la fenêtre JAjoutHeros pour indiquer à l'utilisateur les informations attendues dans chacun des champs
- Au choix
 - Utilisation des propriétés ToolTipText des champs
 - Modification d'un label d'aide intégré dans la fenêtre

Ouverture de fenêtres

- Créer la fenêtre JMenu suivante
- Créer les fenêtres JAjoutJoueur et JDemarrer vides
- Configurer la méthode actionPerformed afin d'afficher les fenêtres correspondant à chaque bouton lors d'un clic



Fenêtre JAjoutJoueur



The image shows a Java Swing window titled "Jeu des héros". Inside the window, the text "Ajouter un joueur" is centered. Below this, there are two input fields. The first is labeled "Nom du joueur" and is a text box. The second is labeled "Choisir le héros" and is a dropdown menu.

Ajouter un joueur

Nom du joueur

Choisir le héros

Gestion de radio bouton

Ajout d'un héros

Création d'un héros

☒ Heros Terre ☐ Heros Mer ☐ Heros Feu

Nom

Points de vie

Défense

Poids

Agilité

- Ajout d'un groupe de boutons (invisible)
- Ajout de 3 radio boutons associés au même groupe de boutons (option selected pour HerosTerre)
- Renommer les champs label qui affichent le poids et l'agilité

Gestion du changement d'option

- Gérer l'évènement clic (actionCommand) pour chaque radio bouton
- Modifier le libellé des 2 dernières propriétés selon le type de héros choisi
- Heros Terre
 - Poids
 - Agilité
- Heros Mer
 - Nombre de nageoires
 - Vitesse
- Heros Feu
 - Taille
 - Puissance de feu

Gestion du héros à ajouter

Bouton ajouter

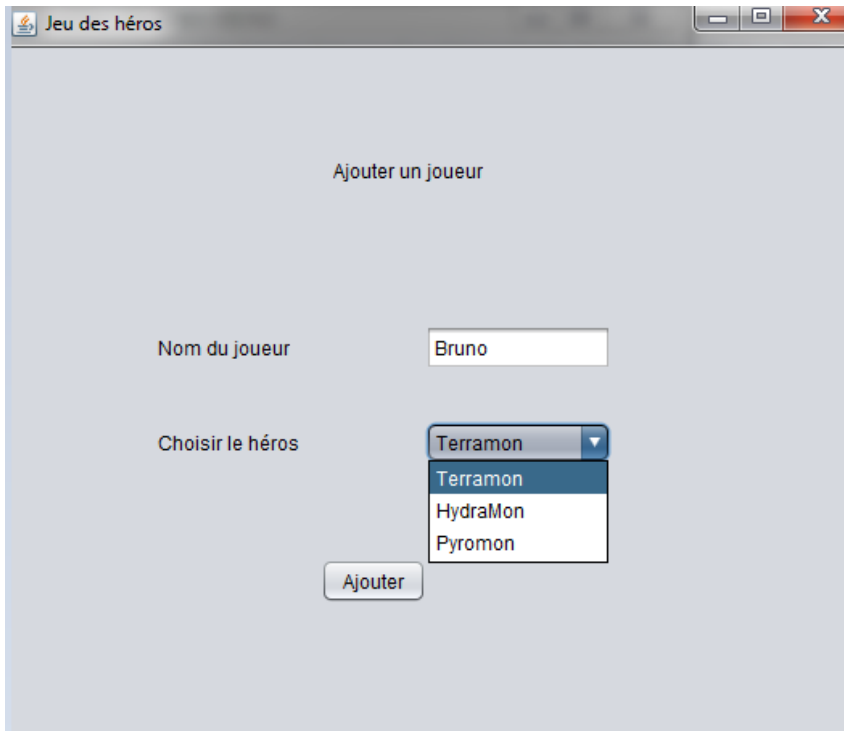
- Créer un HerosTerre, un HerosMer ou un HerosFeu à ajouter à la liste de Heros selon le choix de l'utilisateur

Gestion des radio Bouton

- Méthode isSelected() : boolean pour savoir si le bouton est sélectionné

Gestion d'une liste déroulante

Ajout d'un joueur



The screenshot shows a Java Swing window titled "Jeu des héros". Inside the window, there is a label "Ajouter un joueur" at the top. Below it, there are two input fields. The first is labeled "Nom du joueur" and contains the text "Bruno". The second is labeled "Choisir le héros" and is a JComboBox with a dropdown menu open, showing four options: "Terramon", "Terramon", "HydraMon", and "Pyromon". Below the input fields is a button labeled "Ajouter".

- Création d'une combo Box pour afficher la liste des héros
- La liste des héros est remplie lorsque la fenêtre gagne le focus

Dans la fenêtre AjoutJoueur

A l'initialisation de la fenêtre

- Remplir la liste déroulante avec les noms des héros
- Utilisation de la méthode `addItem(String info)` de la classe `JComboBox`

Bouton ajouter

- Crée le joueur en lui associant le héros sélectionné
- Utilisation de la méthode `getSelectedIndex()` de la classe `JComboBox` pour récupérer l'index de l'élément sélectionné
- L'index dans la liste déroulante correspond à l'index de l'objet `Héros` sélectionné dans la liste de `Heros`

Gestion d'une table

Classement des joueurs

Classement

Rang	Joueur	Points
1	sophie	0
2	Alain	0
3	Pierre	0
4	Fanny	0

- Ajouter une JTable inclue dans un objet ScrollPane

Classement des joueurs

AfficheClassement

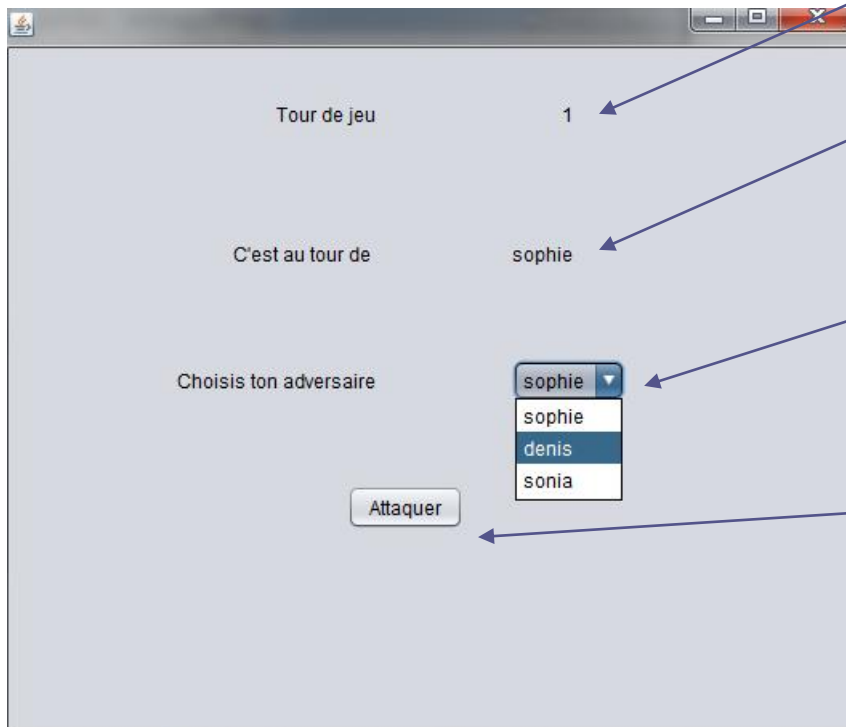
- Reprendre la méthode afficheClassement du moteur de Jeu
- Appeler la méthode à l'initialisation de la fenêtre pour remplir la JTable

Méthode de la classe JTable

- Utiliser la méthode setValueAt(valeur, numligne, numcolonne) pour afficher une valeur dans une case de la Jtable
 - indice de première ligne :0
 - indice de première colonne:0

Moteur du jeu

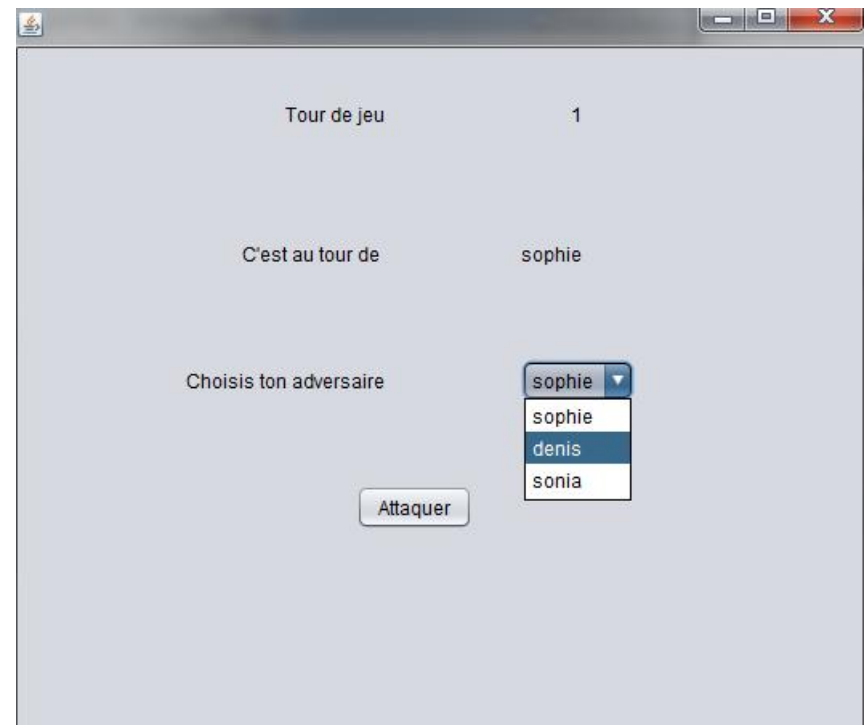
Gestion du jeu



- Affichage du tour de jeu (mis à jour à chaque tour)
- Affichage du nom du joueur (mis à jour après chaque attaque)
- Affichage de la liste des adversaires (initialisée à l'ouverture de la fenêtre)
- Ajout de 2 boutons : Attaquer et Voir le classement


Initialisation de la fenêtre

- Créer une variable static contenant l'indice de l'attaquant courant
- Initialisation du tour de jeu
- Initialisation de l'indice de l'attaquant
- Initialisation de la liste déroulante à partir de la liste des joueurs (afficher les noms)
- Masquer le bouton Voir le Classement



Bouton Voir le Classement

- Ouvre la fenêtre Classement



Classement

Rang	Joueur	Points
1	sophie	0
2	Alain	0
3	Pierre	0
4	Fanny	0

Bouton Attaquer

- Récupère le tour de jeu et le joueur adverse
- Lance l'attaque sur le joueur adverse
- Gère l'action suivante
 - On passe au joueur suivant
 - Ou on passe au tour de jeu suivant avec le premier joueur
 - Ou le jeu est terminé, on masque le bouton attaquer et on affiche le bouton Voir le classement
- Utilisation de la méthode `getSelectedIndex()` de la classe `JComboBox` pour récupérer l'index de l'élément sélectionné
- L'index dans la liste déroulante correspond à l'index de l'objet `Joueur` sélectionné dans la liste de `Joueurs`