

IHM du Pokédeck

TP 5

Programmation objets, web et mobiles en JAVA
Licence 3 Professionnelle – Multimédia

WALTHER Laureen

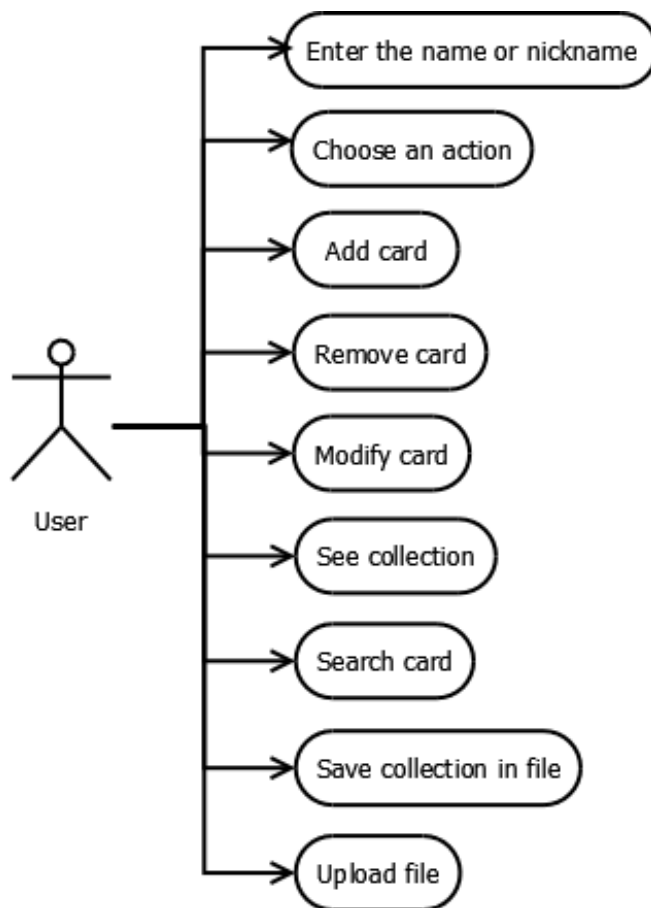
Université Pierre et Marie Curie



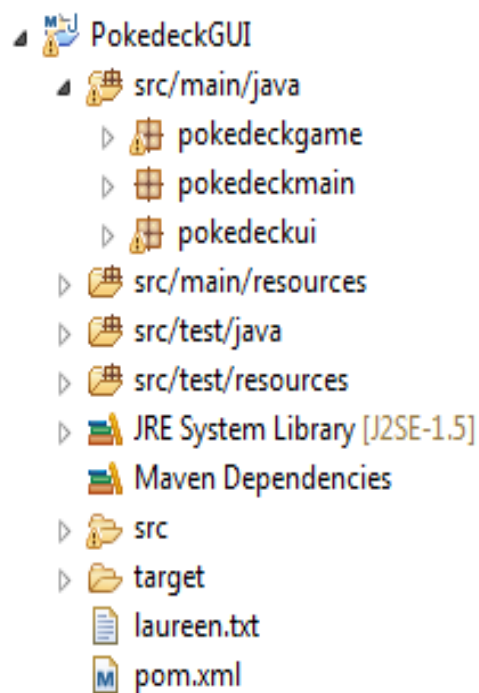
The aim of this work was to implement a Pokedeck, that is to say, software to manage their Pokemon cards.

Github : https://github.com/laureenw/Pokedeck_IHM

1) Use case



2) Code architecture









src/main/java:

- **pokedeckgame** contains all the business classes doing the calculations on the data of the game
- **pokedeckmain** contains the main
- **pokedeckui** contains all the classes of the GUI



src/main/resources: stock images
















3) Packages description

a. pokedeckgame

- ▲  pokedeckgame
 - ▷  Card.java
 - ▷  Player.java
 - ▷  Pokedeck.java
 - ▷  PokemonType.java
 - ▷  UserChoice.java

b. Pokedeckmain

- ▲  pokedeckmain
 - ▷  PrincipalGame.java

- ▲  pokedeckui
 - ▷  AddCardPanel.java
 - ▷  CardPanel.java
 - ▷  FilePreview.java
 - ▷  ImagePanel.java
 - ▷  Menu.java
 - ▷  ModifyCardPanel.java
 - ▷  PokedeckUI.java
 - ▷  PrincipalContent.java
 - ▷  PrincipalPanel.java
 - ▷  RemoveCardPanel.java
 - ▷  SaveCollectionPanel.java
 - ▷  SearchCardPanel.java
 - ▷  SeeCollectionPanel.java
 - ▷  UploadCollectionPanel.java

1. Card

- Contains constructor to create card with parameters num and name
- **Attributes :**
 - num : in type int

- name: in type String
- pokemon_type: in type String corresponding to the enum PokemonType
- pokemon_image: in type String
- **Methods :**
 - Show() : to display card
 - toString() : return string used to describe object
- 2. **Player** : contains player name
 - Contains constructor with parameter **name** (in type String)
 - **Attributes:** name (in type String)
 - **Accesseur** getName()
- 3. **Pokedeck**: contains flow of the game
 - **Attributes:**
 - collectCard (in type ArrayList<Card>) : initialize tab of cards
 - numCard (in type int) : initialize num card to zero
 - nameCard (in type String) : initialize name card to blank
 - myCard (in type Card) : corresponds to the current card
 - cardDelete (in type Object) : corresponds to the deleted card
 - cardUpdate (in type Object) : corresponds to the updated card
 - numCardSearch (in type int) : corresponds to the desired card number
 - nameCardSearch (in type String) : corresponds to the desired card name
 - pokemon_type (in type String) : initialize pokemon type card to blank
 - pokemon_type_search (in type String) : corresponds to the desired pokemon type card
 - pokemon_image (in type String) : initialize image card to blank
 - pokemon_image_search (in type String) : corresponds to the desired image card
 - **Objects:**
 - Random random : used for a random draw
 - Player p : contains the current player
 - **Methods:**
 - **WriteCollectCardInFile()**
 - FileOutputStream file: creates a file output stream to write
 - ObjectOutputStream oos: open stream on file
 - writeObject(): object serialization
 - flush(): empty the writing buffers
 - close(): close stream
 - **ReadCollectCardInFile()**
 - FileInputStream file: creates a file input stream to read
 - ObjectInputStream: open stream on file
 - readObject(): object deserialization
 - **AddCard()**: add description of new card
 - **Test:** as the collection contains card name selected by user, program asks card name
 - **Test:** if the collection contains card number, add 10 at the card number

- Add new card on the collectCard, increments card number, display card
- **RemoveCard():** delete card
 - Request the number card to remove, remove corresponding card on the collectCard
- **ModifyCard():** update card
 - Request the number card to update, request new card name, request new pokemon type, request new image, replace corresponding card with new card name and other statements
- **SearchCard():** search card
 - Request the number card to search, request the name card to search, request the pokemon type to search, request the image to search
 - **Test:** if collectCard contains number card, name card, pokemon type and image : display corresponding card

4) Menu

Code:

```
public enum UserChoice {
    AddCard ("Add new card"),
    RemoveCard ("Delete card"),
    ModifyCard ("Update card"),
    SeeCollection ("See collection"),
    SearchCard ("Search card"),
    SaveCollection ("Save collection"),
    UploadCollection ("Upload collection"),
    Stop ("Exit");

    private String name = "";

    UserChoice(String name) {
        this.name = name;
    }

    public String toString() {
        return name;
    }
}

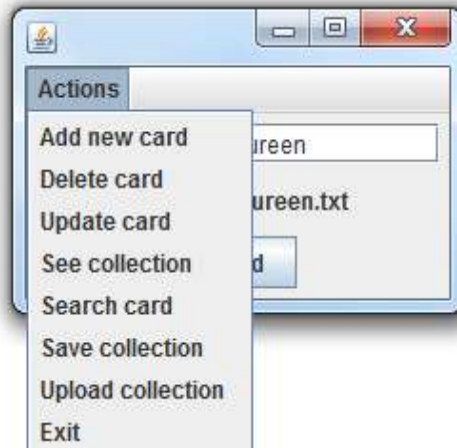
JMenuBar menuBar;
JMenu actions;
JMenuItem addCard, removeCard, modifyCard, seeCollection, searchCard, saveCollection,
uploadCollection, stop;

public Menu() {
    menuBar = new JMenuBar();
    actions = new JMenu("Actions");
    addCard = new JMenuItem(""+UserChoice.AddCard);
    removeCard = new JMenuItem(""+UserChoice.RemoveCard);
    modifyCard = new JMenuItem(""+UserChoice.ModifyCard);
    seeCollection = new JMenuItem(""+UserChoice.SeeCollection);
    searchCard = new JMenuItem(""+UserChoice.SearchCard);
    saveCollection = new JMenuItem(""+UserChoice.SaveCollection);
    uploadCollection = new JMenuItem(""+UserChoice.UploadCollection);
    stop = new JMenuItem(""+UserChoice.Stop);
    actions.add(addCard);
    actions.add(removeCard);
```

```

        actions.add(modifyCard);
        actions.add(seeCollection);
        actions.add(searchCard);
        actions.add(saveCollection);
        actions.add(uploadCollection);
        actions.add(stop);
        menuBar.add(actions);
    }

```



Menu

5) Principal Content

Window where there will be the content:

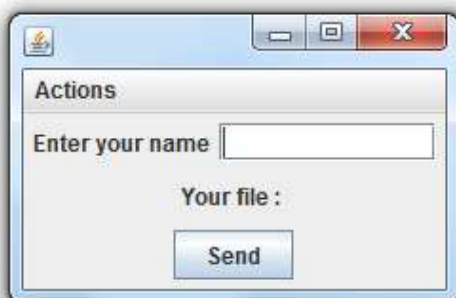
Code:

```

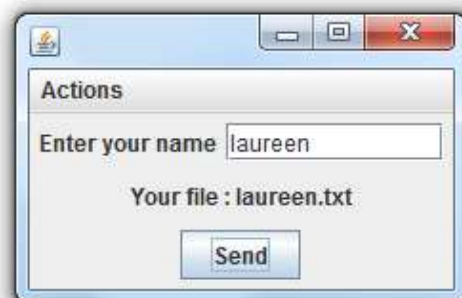
public class PrincipalContent {
    JPanel principalPanel;
    JPanel addCardPanel;
    JPanel removeCardPanel;
    JPanel modifyCardPanel;
    JPanel seeCollectionPanel;
    JPanel searchCardPanel;
    JPanel saveCollectionPanel;
    JPanel uploadCollectionPanel;
    BorderLayout login;
    public PrincipalContent() {
        principalPanel = new JPanel();
        login = new BorderLayout();
    }
}

```

6) Principal Panel

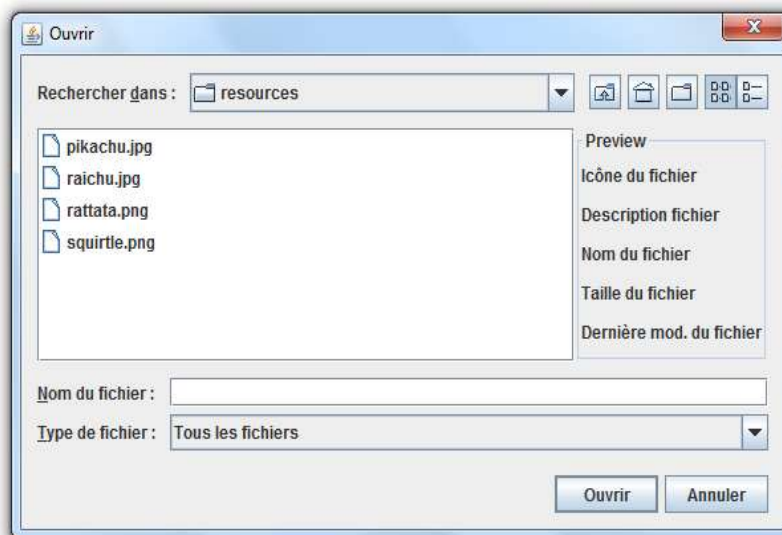


First interface

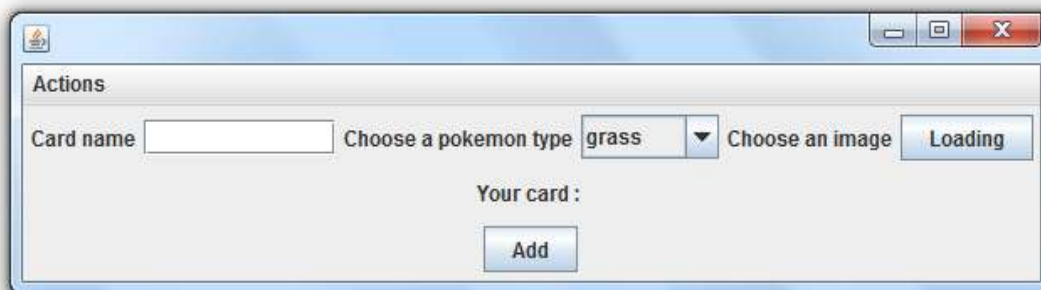


The user has entered his name and the program tells him that the file belongs to

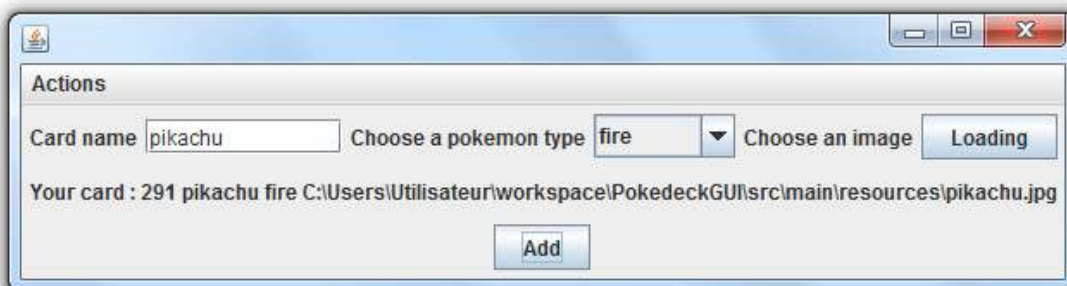
7) File Preview : upload an image



8) Add Card Panel



Add new card



The user has entered the name of a card; the user has chosen the pokemon type, image and the program displays the card

Code:

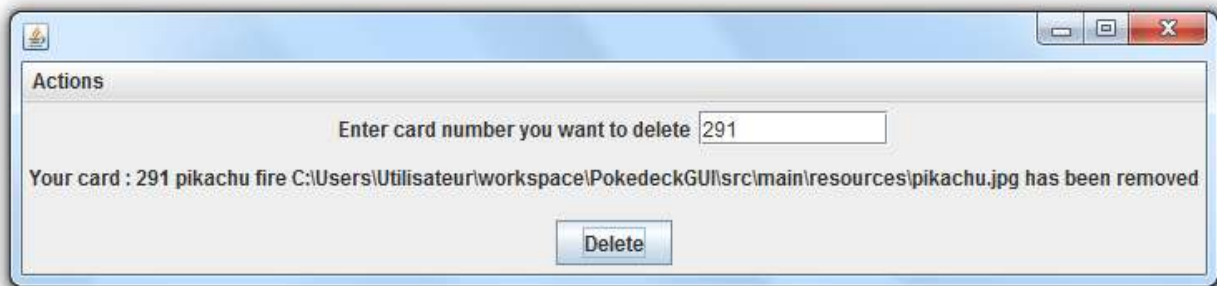
```
//addCard panel
menu.addCard.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        AddCardPanel addcardpanel = new AddCardPanel();
        window.getContentPane().removeAll();
        window.getContentPane().add(addcardpanel.principalcontent.addCardPanel,
        BorderLayout.PAGE_START);
    }
});
```

```

        window.getContentPane().add(addcardpanel.card_addcardpanel,
BorderLayout.CENTER);
        window.getContentPane().add(addcardpanel.button_addcardpanel,
BorderLayout.PAGE_END);
        window.getRootPane().repaint();
        window.getRootPane().revalidate();
    }
});

```

9) Remove Card Panel



Delete card: if the card number exists in the collection, the card is removed

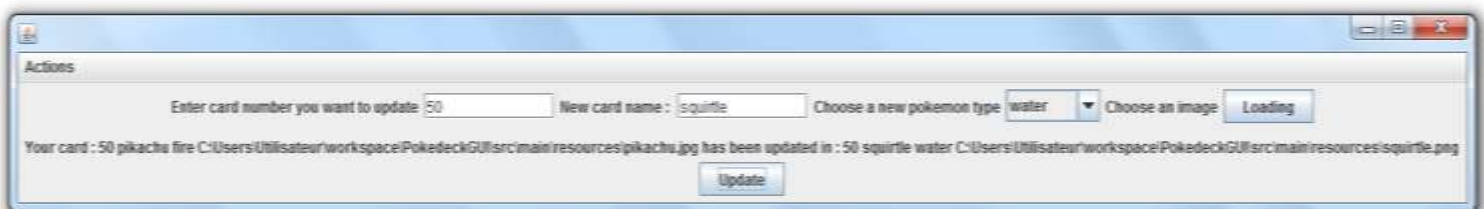
Code:

```

//removeCard panel
menu.removeCard.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        RemoveCardPanel removecardpanel = new RemoveCardPanel();
        window.getContentPane().removeAll();
        window.getContentPane().add(removecardpanel.principalcontent.removeCardPanel,
BorderLayout.PAGE_START);
        window.getContentPane().add(removecardpanel.card_removecardpanel,
BorderLayout.CENTER);
        window.getContentPane().add(removecardpanel.button_removecardpanel,
BorderLayout.PAGE_END);
        window.getRootPane().repaint();
        window.getRootPane().revalidate();
    }
});

```

10) Modify Card Panel



Update card: enter the card number, the new name, chooses the new pokemon type and new image

Code:

```

//modifyCard panel
menu.modifyCard.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        ModifyCardPanel modifycardpanel = new ModifyCardPanel();
        window.getContentPane().removeAll();
    }
});

```



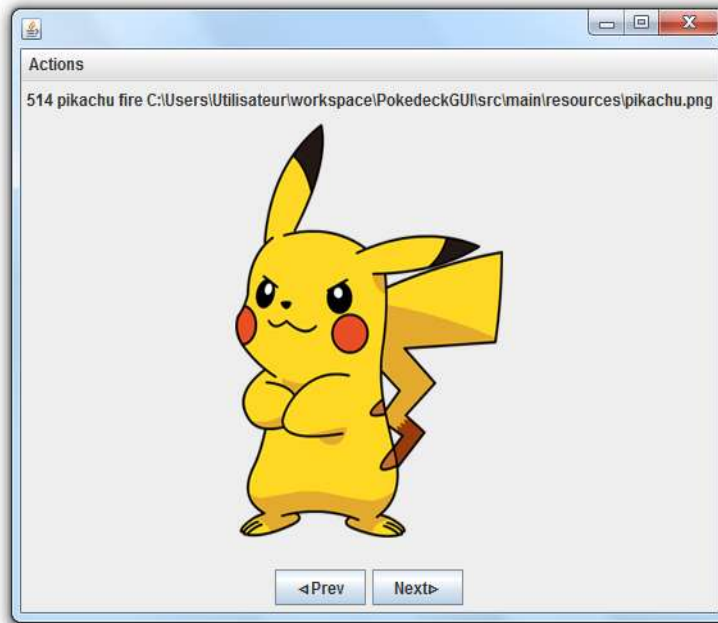
```

        window.getContentPane().add(modifycardpanel.principalcontent.modifyCardPanel,
BorderLayout.PAGE_START);
        window.getContentPane().add(modifycardpanel.card_modifycardpanel,
BorderLayout.CENTER);
        window.getContentPane().add(modifycardpanel.button_modifycardpanel,
BorderLayout.PAGE_END);

        window.getRootPane().repaint();
        window.getRootPane().revalidate();
    }
});

```

11) See Collection Panel



See collection

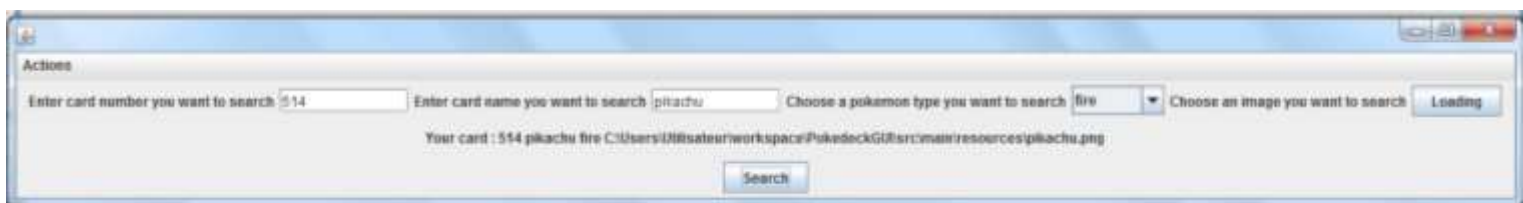
Code:

```

//seeCollection panel
menu.seeCollection.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        SeeCollectionPanel seecollectionpanel = new SeeCollectionPanel();
        window.getContentPane().removeAll();
        window.getContentPane().add(seecollectionpanel.cards,
BorderLayout.PAGE_START);
        window.getContentPane().add(seecollectionpanel.images, BorderLayout.CENTER);
        window.add(seecollectionpanel.control, BorderLayout.SOUTH);
        window.getRootPane().repaint();
        window.getRootPane().revalidate();
    }
});

```

12) Search Card Panel

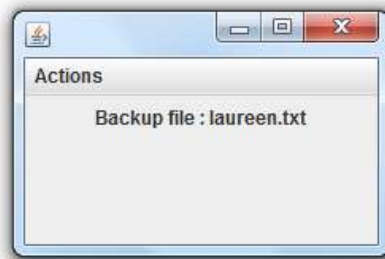


If card is in the collection

Code:

```
//searchCard panel
menu.searchCard.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        SearchCardPanel searchcardpanel = new SearchCardPanel();
        window.getContentPane().removeAll();
        window.getContentPane().add(searchcardpanel.principalcontent.searchCardPanel,
BorderLayout.PAGE_START);
        window.getContentPane().add(searchcardpanel.card_searchcardpanel,
BorderLayout.CENTER);
        window.getContentPane().add(searchcardpanel.button_searchcardpanel,
BorderLayout.PAGE_END);
        window.getRootPane().repaint();
        window.getRootPane().revalidate();
    }
});
```

13) Save Collection Panel

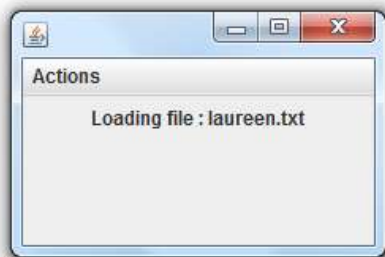


Save collection

Code:

```
//saveCollection panel
menu.saveCollection.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        SaveCollectionPanel savecollectionpanel = new SaveCollectionPanel();
        window.getContentPane().removeAll();
        window.getContentPane().add(savecollectionpanel.principalcontent.saveCollecti
onPanel, BorderLayout.PAGE_START);
        window.getRootPane().repaint();
        window.getRootPane().revalidate();
    }
});
```

14) Upload Collection Panel



Upload collection

Code:

```
//uploadCollection panel
menu.uploadCollection.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(ActionEvent e) {
        UploadCollectionPanel uploadcollectionpanel = new UploadCollectionPanel();
        window.getContentPane().removeAll();
        window.getContentPane().add(uploadcollectionpanel.principalcontent.uploadColl
        ectionPanel, BorderLayout.PAGE_START);
        window.getRootPane().repaint();
        window.getRootPane().revalidate();
    }
});
```