

Report Java Project - Mystic Tarot

Valère Preney

December 2020

1 Introduction

Before begin any typing, I've made some research in order to know more about this "practice". And I've found all of what I needed here : <https://www.apprendre-tarotdemarseille.com/> (All the images I used for the card are from this website by the way). The I've decided to start with Arcanes Mineures and Arcanes Majeures. And as I thought two classes were not enough for basis of the project, I added one own-maded class. Arcanes Majestueuse is it.(More details in this report below - Classes Section) In a first hand I wrote all the classes that doesn't belong to the view part. So are they Player class, Card(Abstract class), ArcanesMajeures, Arcanes Mineures, Arcanes Majestueuses. All of them extended from Card Class. And all the view part was coded in a one single page. As I thought it wasn't really clean and properly coded I've decided to start all the project over again. Then I've coded the project by using the MVC design pattern, and the Observer principle. And there is differents classes in the view part that represents differents components of the application (Also more details in the report below)

2 UML Diagram

See the Class Diagram png in RapportProjet directory

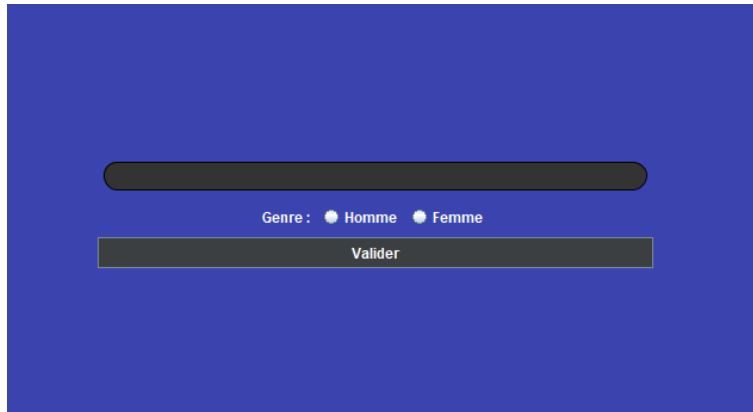
3 How to use it

3.1 Form Welcome Panel

Once the user has started the program, he has got to fill a welcome form. He must enter his name and choose betwin two choices of a two radioButton couple. Then he press enter and go into in.

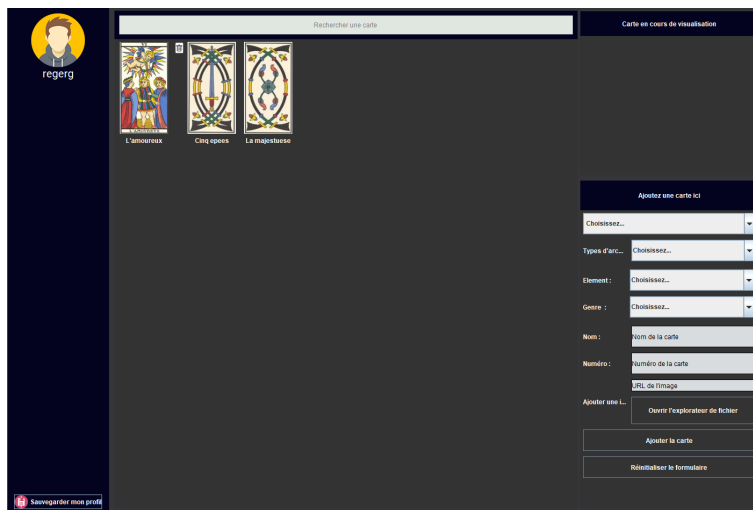
3.2 Main Window

Then, once it's done, the user can see the program break down into segments.



A dark blue rectangular panel with a dark grey rounded rectangle at the top. Below it, the text "Genre : " is followed by two radio buttons. The first radio button is selected and is followed by the word "Homme". The second radio button is unselected and is followed by the word "Femme". Below these options is a dark grey rectangular button with the word "Valider" in white text.

Figure 1: Form Welcom Panel



The main page has a dark blue background. On the left is a sidebar with a yellow circular profile icon containing a person silhouette, labeled "regerg" below it. At the bottom of the sidebar is a red button with a white floppy disk icon and the text "sauvegarder mes profils". The main area has a header bar with a search icon and the text "Rechercher une carte". Below the header, three tarot cards are displayed: "L'Amoureux", "Cinq épées", and "La mort". The right sidebar is titled "Carte en cours de visualisation" and contains a section "Ajoutez une carte(s)" with several dropdown menus: "Choisissez...", "Types d'arc...", "Choisissez...", "Element:", "Choisissez...", "Genre :", "Choisissez...", "Nom:", "nom de la carte", "Numero:", "numéro de la carte", and "URL de l'image". Below these is a button "Ouvrir l'explorateur de fichiers". At the bottom of the right sidebar are two buttons: "Ajouter la carte" and "Réinitialiser le formulaire".

Figure 2: Main Page

3.2.1 The SideBar

First you've got the side bar with a little icon, the name and at save profil button at the bottom.



Figure 3: Male Icon



Figure 4: Female Icon



Figure 5: save button

3.2.2 The central panel - collection card and search bar

Then you can see a central panel. Here released a search bar and the card collection of the user. Each card will be set with a respective image and a name.

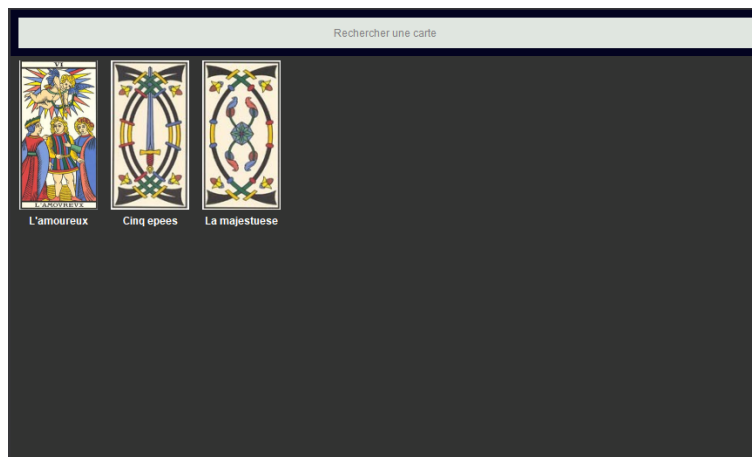


Figure 6: Collection Card



By hovering one of these card, a dustbin icon will pop just on the left of the card image so you can remove the card from you're collection.

3.2.3 The detail Panel



By clicking on one of these card you set the top-right panel : detail Panel. Here you will see the detail of the card you've just clicked on before.



Figure 7: Empty Panel Detail



Figure 8: Panel Detail

By hovering each property value on the right of the image, a pen icon will pop  so you can set the property that belong to the icon. JLabel will turn into JTextField or JComboBox. Depending on the nature of the property you're setting. To validate your change, where the edit icon was a validate icon  is instead.

3.2.4 The adding form

And finally the adding form there is. With this, you will can add new card to your collection.

There is different options :

1. You can add a card by choosing one of the default card. Once the choice done, the form will automatically be filled. Then you will be incapable to modify the fields unless you clickon the empty form button.
2. Otherwise you can create your own card by choosing the item you want in the combobox components and typing the right word in the text fields components. But the first combobox, the default card proposal, must remained unchosen. It means the first item ("choisissez une carte") must be selected.

Figure 9: Adding Form

4 The Project Structure

4.1 The View - Package

This package contains all what's important for the user interface.

4.1.1 The interfaces

These interfaces are required when the user trigger the event by using the linked component. To do so, I've been thinking like that : there is one component that trigger the event and the event listener calls the corresponding method in the linked Panel which implements the needed interface. The bound between both component (component trigger, (interface)panel calling method) is created in a static HashMap set in the View Page. Most of the time the panel linked to the trigger component is the panel that contains the trigger component. But not for one time where the panel needed to be listen is send to the class that contains the trigger component. By doing like this you have juste two lines of code to write into the EventListener Classes. It doesn't matter which component add an event to his listener.

1. IGotBtnClickable
2. IGotBtnHover
3. IGotComboxChange
4. IGotFocusComponent
5. IGotTextFieldKeyListening

4.1.2 The Classes

- View Page
 - FormPanelWelcome
 - * RoundedJTextField
 - * RoundedBorderCorner
 - PanelMainPage
 - * PanelSideBar
 - * PanelCollectionCard
 - CardPanel
 - CardIsJButton
 - * PanelDetailCard
 - AsbtractPanelForArcanesType
 - PanelForArcanesMajestueuseDetail
 - PanelForArcanesMajeureDetail
 - PanelForArcanesMineureDetail

- * PanelFormAddCard
 - ListRenderer
- EventListenerClasses
 - ButtonAction
 - ComboBoxAction
 - FocusAction
 - KeyPressedAction

4.2 The Controller - Package

This package is like a ferryman. Every change triggered in the view package (by the user) will be set in the classes model package.

4.2.1 The classes

1. AbstractControllerPlayer
2. ControllerPlayer

4.3 The Model - Package

4.3.1 The Interfaces

This interface takes care of the sustainability of the essentials data and as for it, Player is the only class to implements the interface.

1. IsSustainable

4.3.2 The Classes

1. AbstractModel
2. Player
3. Card
4. ArcanesMajeures
5. ArcanesMineures
6. ArcanesMajestueuses

4.4 The Observer - Package

This package well carrying his name. Every observed change in th model will reset the view. Once the change is set into the corresponding model class, the observable classes call the method `notifyObserver(Player p)` that inform all the observers.

4.4.1 The interface

1. Observable
2. Observer