Let’s go on Holidays

CPM Project

Paula Díaz Álvarez UO294067

Table of contents

Introduction

Problem

I had to develop an app that allows the user to:

- **Play a gam**e to gain discounts.
- **Making reservations** for castles with the characteristics that the user wants.

The gaining discount process is as follows:

1. **Play the game**: you move the ghostbusters according to the value of the dice to eliminate the ghosts in the board.
2. **At the end of the game**: if you gain a discount (25% if you eliminate the ghost leader and the gangs are not complete, or 10% if only the gangs are not complete) the user can save the discount if he wants. If it doesn't provide an id to save or the id has an already saved discount, the new discount will be lost.

The reservation process is as follows:

1. The user will **see all the available castles** according to the criteria he choses (country, enchantments and price)
2. The user can click on the castle he wants, and he will see the **detailed information of the castle**.
3. The user will **indicate the date of the reservation, the number of people, the number of rooms and the number of days**. It must be verified that there cannot be more than 2 people in a room
    a. The user will **introduce an id, and can claim a discount** if he wants (and has one). The reservation price will be updated after applying the discount
    b. The user can change the details of the reservation at any time before it is formalized
4. Before formalizing, **a summary of the reservation will be shown** with the castle they are reserving, the number of rooms, people and days, and the total price of the reservation.
5. If the user accepts the reservation, the **formalization** window will be shown so the user can fill: the name, surname, contact email and comments (the comments are optional)
6. If he finally formalizes, **the reservation will be saved and the discount applied (if any) won't be able to be used again**.

Implementations to take into account

- Internationalization.
- Use of a look and feel.
- Use of a JCalendar so the user can choose the starting date.
- Use of music while playing the game

Application development

Logic

To resolve the logic of the reservation part, we have the following classes:

- **Castle**: It represents each castle and stores all the information of the castle
- **Discount**: It represents the discount: it has an id associated and its value (the code)
- **DiscountCode**: It's the different values of the discount: EXTRA25, EXTRA10

- **Reservation**: It's the reservation that the user makes for a castle. It contains all the information for saving the booking, the discount applied, room price and number of people
  - *calculateFinalPrice(int numberOfRooms)*: Calculates the final price as the roomPrice * numberOfRooms * numberOfDays * discount applied (if there is any)
  - *makeReservation(String id, String castleCode, Date reservationDate, int days, int numberOfRooms, int numberOfPeople)*: Sets the given information of the reservation
- **TypeOfEnchantment**: It's the list of enchantments: Ap, De, En, Ob, Ol, Ru
- **FileUtil**: It's the file util for loading the castles and discounts to the service, and saving the reservation and discounts
  - *loadCastleFile(String fileName, List<Castle> castlesList)*: Adds to the given castlesList all the castles loaded from the file with the given filename
  - *saveReservationToFile(String fileName, Reservation reservation)*: Appends the given reservation to the file with the given fileName
  - *loadDiscounts(String fileName, List<Discount> discountList)*: Adds to the given discountList all the discounts loaded from the file with the given filename
  - *updateDiscounts(String fileName, List<Discount> discountList)*: Writes all the discounts in the given list in the file with the given fileName
  - *saveDiscount(String fileName, Discount discount)*: Appends the given discount to the file with the given fileName
- **Castlevania**: It's the class representing the castle reservation business. It has the list of castles and discounts; the reservation and chosen castle; and the enchantment and price filter
  - *Castlevania()*: It creates the business app. It initializes all the lists, adds all the enchantments to the filter and loads the discounts
  - *createReservation()*: Creates a new reservation
  - *makeReservation(String id, String castleCode, Date reservationDate, int days, int numberOfRooms, int numberOfPeople)*: Sets the given information to the reservation
  - *addEnchanmentToFilter(TypeOfEnchantment enchantment)*: Adds the given enchantment to the filter if it doesn't contain them already
  - *addAllEnchanmentToFilter()*: Adds all the enchantments to the filter
  - *removeEnchanmentToFilter(TypeOfEnchantment enchantment)*: Removes the given enchantment to the filter
  - *removeAllEnchanmentToFilter()*: Removes all the enchantments on the filter
  - *loadCastles(String fileName)*: Loads the castles from the given file to the list of castles, and calculates the limits for the price filer
  - *loadDiscounts()*: Loads the discounts from the discounts file to the list of discounts
  - *saveDiscount(Discount discount)*: Saves the given discount on the discounts file and add it to the list of discounts
  - *removeDiscount()*: Removes the discount from the list of discounts, and updates the discounts file
  - *saveReservation()*: Saves the current reservation to the bookings file

    - *getCountries()*: returns the sorted list of all the countries from the list of castles
    - *filterByCountry(List<Castle> castles, String selectedCountry, int selectedIndex)*: Filters the given list of castles by the selectedCountry. If the selected index is 0, the list of castles is return as it is (the default option is chosen)
    - *calculateLimitsPrice()*: It calculates the limits for the price filter by picking the minimum price and maximum price of the castles
    - *filterByPrice(List<Castle> castles, int selectedIndex)*: Filters the given list of castles by the price. From the selectedIndex we obtain the maximum value for the price (same index as in priceFilter) and the minimum value (index-1 as in priceFilter)
    - *getCastleList()*: returns the list of castles filtered by enchantments
    - *hasDiscount(String dni)*: returns if there is a discount with the dni associated
    - *searchDiscount(String dni)*: returns the discount associated with the given dni. If there is no discount, it returns null

To resolve the logic of the game part, we have the following classes:

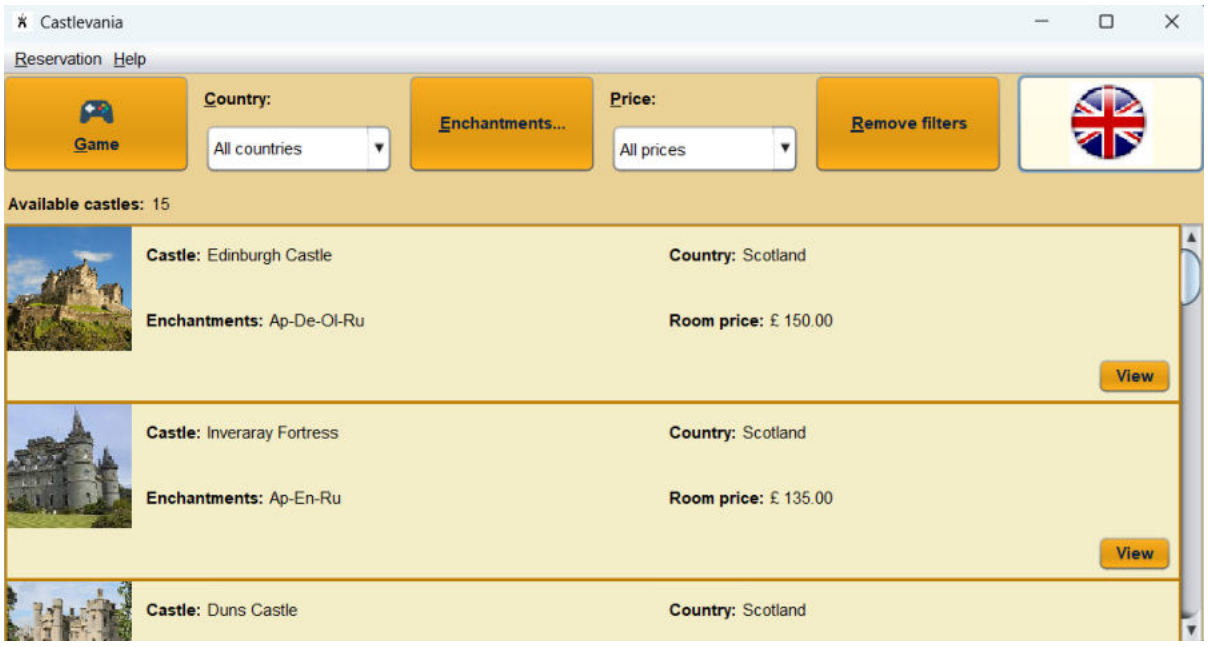
- **Board**: It's the board of the game with the matrix of squares, the list of gangs and the leader ghost
    - *Board()*: it creates the list of gangs with each type, initializes each square and it puts each entity in the corresponding position
    - *localizeGhost(int[] pos)*: Localizes a random ghost in a random position, and it's added to its corresponding gang. If the gang is full another type of ghost is chosen
    - *swap(int iGb, int iG, int j)*: It swaps the position of the given ghostbuster and the given ghost (it eliminates the ghost, it moves the ghostbuster to the ghost' square)
    - *areGangsNotCompleted()*: return true in all the gangs at least one ghost has been eliminated
    - *hasLeaderBeenEliminated()*: if the leader ghost has been eliminated
- **Dice**: It's the dice for the game
    - *launch()*: It generates randomly 1 or 2
- **Entity**: It represents each creature of the board: ghosts and ghostbusters
- **Gang**: It's the group of maximum ghosts of the same type
    - *addGhost(Ghost ghost)*: Adds the given ghost if the ghost is of the same type as the gang and the gang is not full
    - *isCompleted()*: returns if none of the ghost in the gang have been eliminated
- **Ghost**: Represents the entity Ghost
    - *isEliminated()*: returns if the ghost has been eliminated by a ghostbuster
    - *setEliminated(boolean eliminated)*: Sets if the ghost has been eliminated or not
- **Ghostbuster**: It represents the ghostbusters
- **Square**: It represents each square of the board of the game, and it contains the entity that is in the square
    - *getEntity()*: returns the entity in the square
    - *setEntity(Entity entity)*: Sets the entity that is in the square to the given one

- **TypeEntity**: It's the type of entity that can be in a square: ghost_1, ghost_2, ghost_3, ghost_4, ghost_5, ghost_leader, ghostbuster
- **Game**: It's the game. It contains the board and the number of throws made
  - *hasWonExtra25()*: returns if the user has won a 25% discount
  - *hasWonExtra10()*: returns if the user has won a 10% discount
  - *hasFinish()*: returns if the game has finish (there are no more throws)
  - *getRemainingThrows()*: returns the remaining throws (throws - throws made)
  - *increaseNumberOfThrowsMade()*: Increases by one the number of throws made
  - *swap(int iGb, int iG, int j)*: Swaps the ghostbuster and the ghost to eliminate the ghost, and returns if they have been swapped
  - *getDiscount(String id)*: returns the discount gained by the given id
  - *initialize()*: It initializes the game: it creates a new board and sets the number of throws made to 0

For the music:
I use the class of MusicPlayer used in the labs and add a method to play the song I wanted as music. In the GameDialog I created a thread to just continuously play the music while you are playing the game, and stop when you go out of the game view.

Interface

Main Window

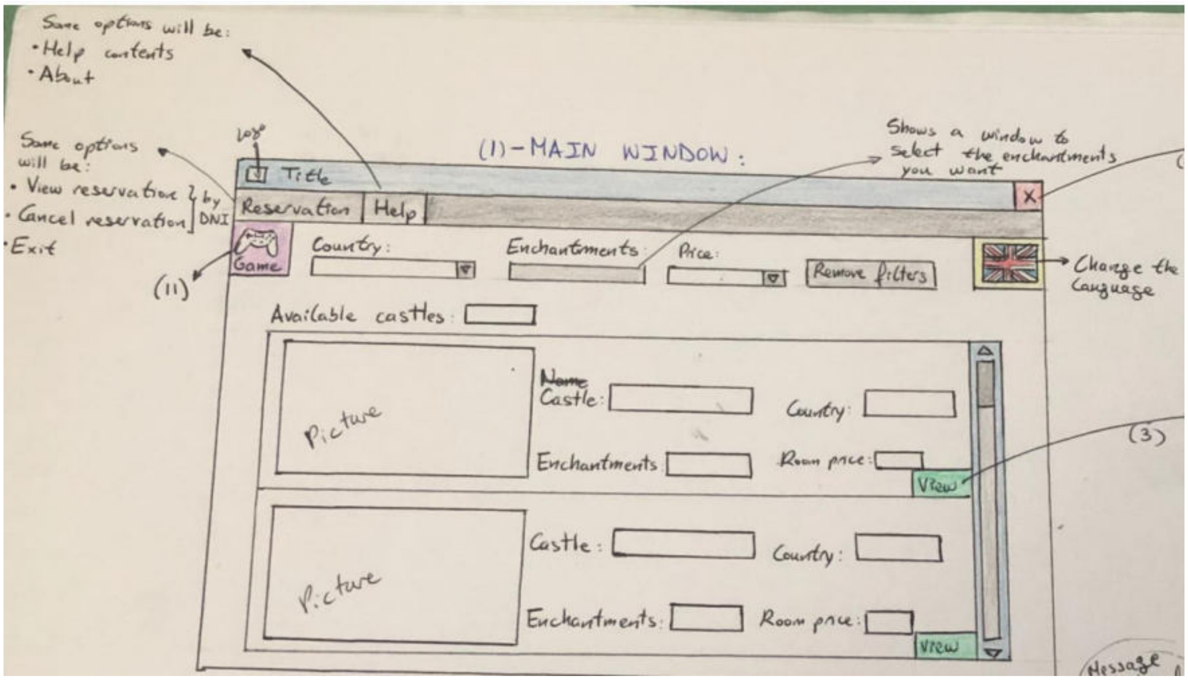


Available castles: 15

Castle: Edinburgh Castle **Country:** Scotland
Enchantments: Ap-De-Ol-Ru **Room price:** £ 150.00

Castle: Inveraray Fortress **Country:** Scotland
Enchantments: Ap-En-Ru **Room price:** £ 135.00

Castle: Duns Castle **Country:** Scotland

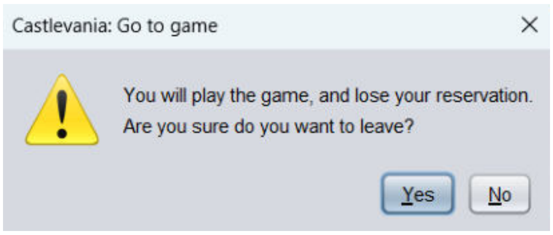


Changes:

- I change the options in the reservation menu: instead of the view reservation and cancel reservation by ID (as they are new functionalities of dealing with the bookings file, and the statement said to not implement them), I substitute them by **the menu Gain discount** that allows the user to play the game at any time. For example, if the user thought that he had a discount but in reality not, he could play the game without going back to the main window so he can gain a discount.
- Instead of having several windows, I mainly divided the app by 2: one window with **CardLayout** for the whole reservation process, and another window with CardLayout for the whole gaming part. I made this decision so the user doesn't have many windows on the screen

Most relevant elements:

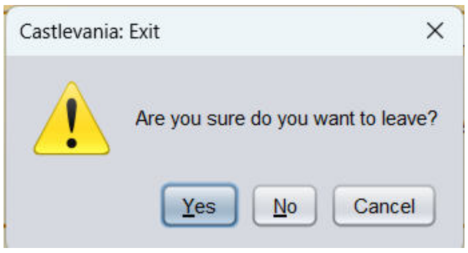
- The **menu bar** with the options:
  - **Reservation**
    - **New**: initializes the app again
    - **Gain discount**: allows the user to play the game at any time, but the user will lose the current reservation, so a JOptionPane warns the user before going to the game



Castlevania: Go to game

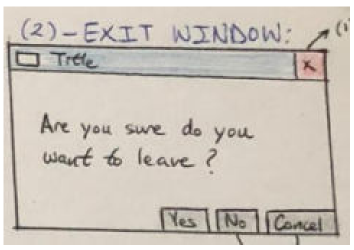
You will play the game, and lose your reservation.
Are you sure do you want to leave?

    - **Exit**: It's a JOptionPane that checks if the user really wants to close the app



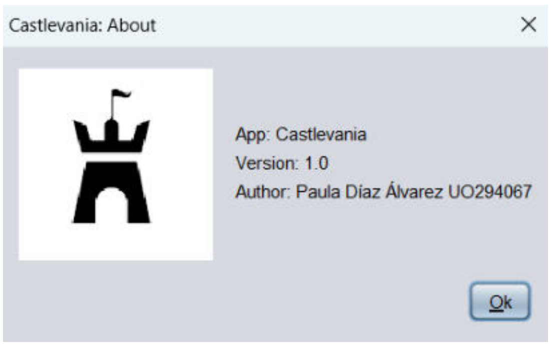
Castlevania: Exit

Are you sure do you want to leave?



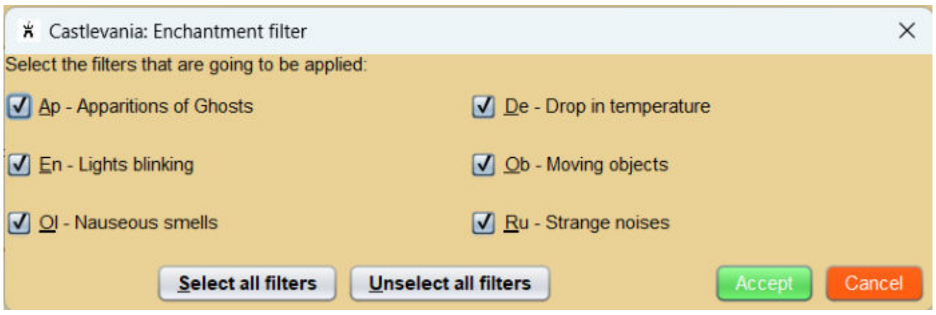
  - **Help**

        - **Help contents**: It shows the help of the app
        - **About**: It's a JOptionPane that shows the app name, version and author


Castlevania: About

App: Castlevania
Version: 1.0
Author: Paula Díaz Álvarez UO294067

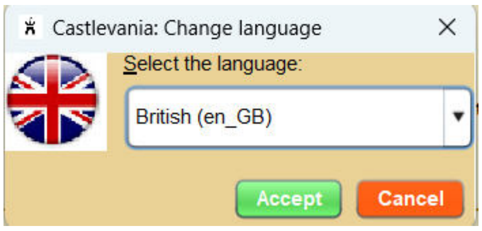
- The **north panel** with: button for playing the game, the filters and the language button
    - For the **filters by country and price**, I decided to use 2 combo boxes, so the user can easily identify the different options, and as it a simple filter (you only select one option)
    - For the **filter by the enchantments**, I decided to use another window that shows all the enchantments. So the user can select any enchantment (as they are not exclusive, I use checkboxes). I also put 4 buttons:
        - One for **selecting all** the enchantments and another for **unselecting**, that would be very useful so the user doesn't have to go one by one selecting them or unselecting them.
        - One for **accepting the changes and** other to **cancel** them if the user regrets what he has just chosen


Castlevania: Enchantment filter

Select the filters that are going to be applied:

☑ Ap - Apparitions of Ghosts
☑ De - Drop in temperature
☑ En - Lights blinking
☑ Ob - Moving objects
☑ Ol - Nauseous smells
☑ Ru - Strange noises

    - The **remove filters** is just a button that sets all the filters to their default option, so the user can eliminate all easily
    - For **changing the language**, I use another window with a combo box with all the languages sorted and not translated so the user can recognize them easily. For example: british, español… It also has 2 buttons to accept and cancel the changes made

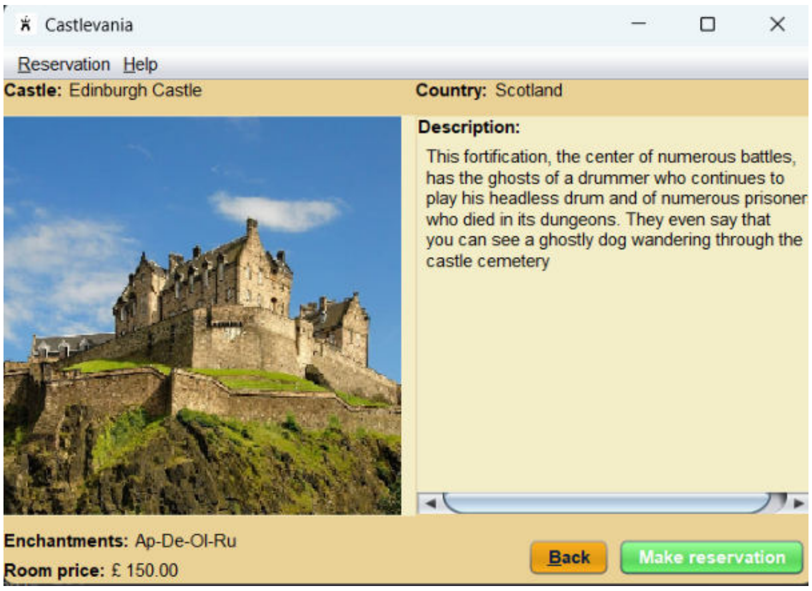

Castlevania: Change language

Select the language:

British (en_GB)

- The center of the panel that shows **the list of castles** uses a scrollPane and a gridLayout so all the panels for each castle have the same height. The panel for each castle has a BorderLayout with the image, the button View to see the detailed information, and a GridLayout with the castle name, country, list of enchantments and room price.

Castle Window



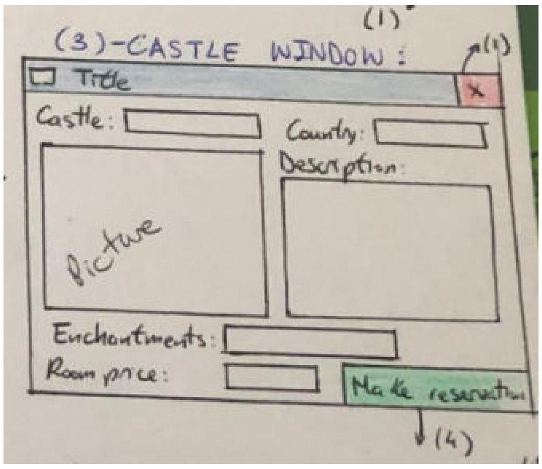
Castle: Edinburgh Castle **Country:** Scotland

Description:

This fortification, the center of numerous battles, has the ghosts of a drummer who continues to play his headless drum and of numerous prisoner who died in its dungeons. They even say that you can see a ghostly dog wandering through the castle cemetery

Enchantments: Ap-De-Ol-Ru

Room price: £ 150.00



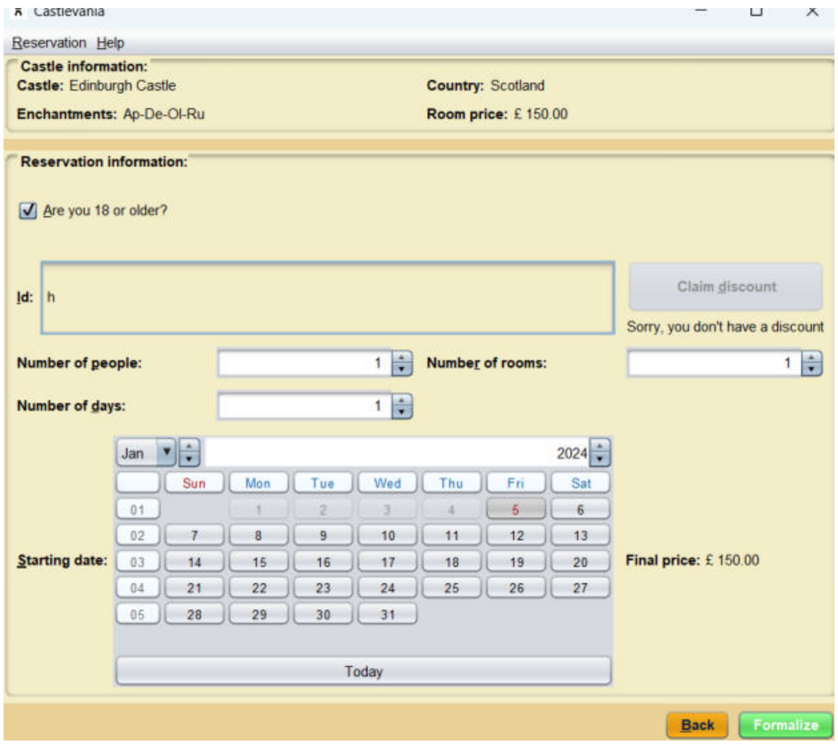
Changes:

- Add the button Back, because before (when not using CardLayout) it was an independent window and the user coil close it, but now I need a button to go back to the panel before

Most relevant elements:

- The **north panel** has the name of the castle and the country
- The **center panel** has the image, and a text area and scroll pane with the description
- The **south panel** is a GridLayout has: one panel with GridLayout for the enchantments and room price; and another panel with GridLayout for the buttons:
  - **Make reservation**: It goes to the Reservation window with that castle as the chosen one
  - **Back**: It goes back to the Main Window panel but you don't lose the information of reservation

Reservation Window



Castle information:

Castle: Edinburgh Castle **Country:** Scotland

Enchantments: Ap-De-Ol-Ru **Room price:** £ 150.00

Reservation information:

☑ Are you 18 or older?

Id: h

Sorry, you don't have a discount

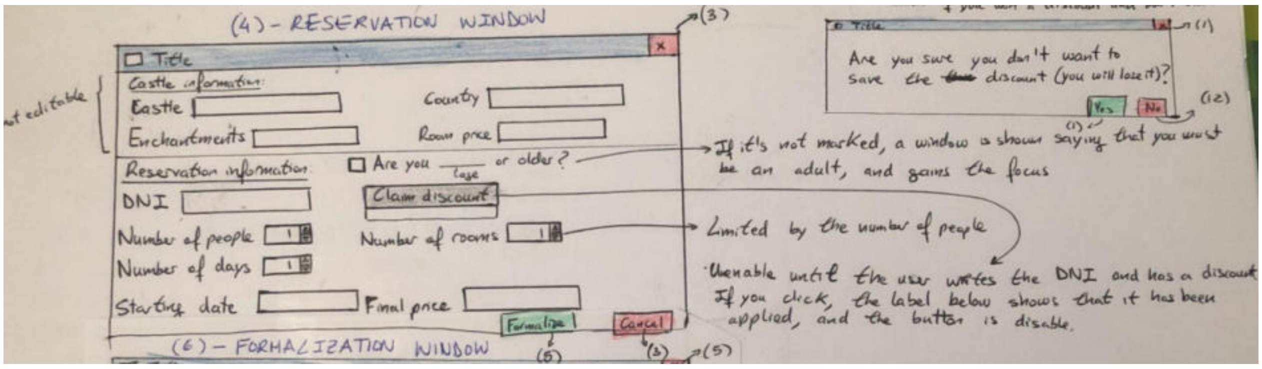
Number of people: 1 **Number of rooms:** 1

Number of days: 1

Starting date: Jan 2024

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
01		1	2	3	4	5	6
02	7	8	9	10	11	12	13
03	14	15	16	17	18	19	20
04	21	22	23	24	25	26	27
05	28	29	30	31			

Final price: £ 150.00

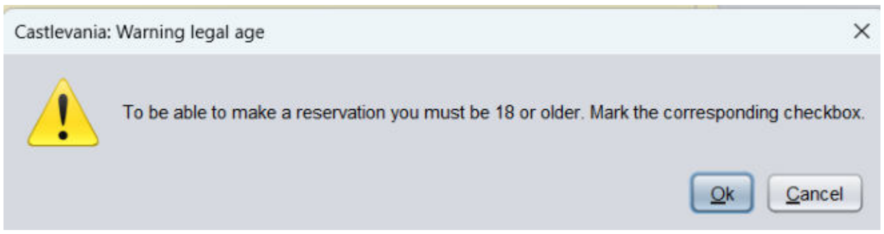

(4) - RESERVATION WINDOW
Title
not editable
Castle information:
Castle
Country
Enchantments
Room price
Reservation information
Are you 18 or older?
DNI
Claim discount
Number of people
Number of rooms
Number of days
Starting date
Final price
Formalize
Cancel
(3)
(5)
(6) - FORMALIZATION WINDOW
Title
Are you sure you don't want to save the discount (you will lose it)?
Yes
No
(1)
(12)
If it's not marked, a window is shown saying that you must be an adult, and gains the focus
Limited by the number of people
Unenable until the user writes the DNI and has a discount
If you click, the label below shows that it has been applied, and the button is disable.

Changes:

- I change the Cancel button for a **Back button** and change the order (to be consistent and predictable), as I use a CardLayout, the user won't see lots of windows, he would feel that he is passing different pages of a form, and I think that Back is a more suitable word in this view as the user doesn't lose the information of the reservation if he goes to the previous view, and the user with Cancel will feel that he loses the whole reservation.
- **Notify the user when it doesn't have a discount**, so the user can know when it doesn't have a discount, if it has a discount and when the discount is applied.
- I substitute the text field of the starting date by a **JCalendar** to be able to check the date is correct (in format and that the date is not less than today), and depending on the locale, change the calendar so everyone can understand it.

Most relevant elements:

- A panel with BorderLayout containing:
  - A panel with GridLayout with a **summary of the castle**
  - A panel with GridLayout with **all the form to be filled** (with 2 panels inside)
    - It has a **checkbox to check that the user is 18 or older**, so he can make a reservation. If not, a warning JOptionPane will appear. If it presses Ok, he will go back to the reservation view to mark the checkbox, if not he would go to the Castle Window view

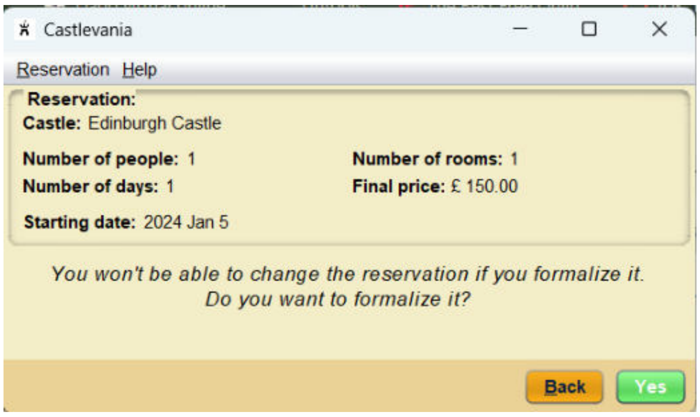

Castlevania: Warning legal age

To be able to make a reservation you must be 18 or older. Mark the corresponding checkbox.

    - It has a **DNI text field** that doesn't allow ';' for security of the files as it's the separator of fields
    - It has a **Claim discount button** and a label that will notify the user if it has a discount so he can claim it, if not or if the discount has been applied
    - It has **3 spinners: number of people, rooms and days**. It's the best option so the user can introduce a valid number within a range. The number of rooms minimum and maximum is determined every time the number of people is changed, so the user cannot enter an invalid value
    - It has a **JCalendar** to introduce the starting date, and it's limited by the today's date
    - The **text field of the final price** is updated every time the number of rooms or days changes

- A panel with the buttons:
    - **Formalize**: It goes to the Formalize warning window
    - **Back**: It goes to the Castle Window without losing the information filled

Formalize warning Window



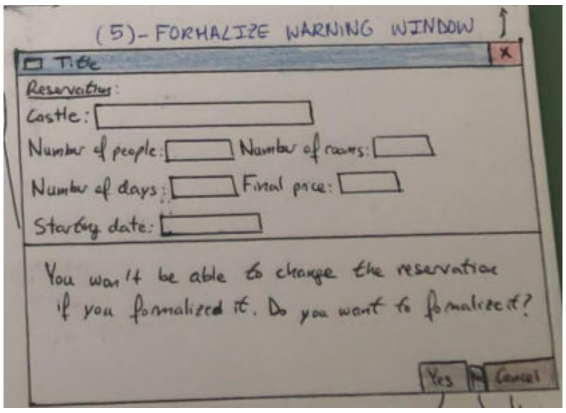
Castlevania

Reservation Help

Reservation:
Castle: Edinburgh Castle
Number of people: 1 **Number of rooms:** 1
Number of days: 1 **Final price:** £ 150.00
Starting date: 2024 Jan 5

You won't be able to change the reservation if you formalize it.
Do you want to formalize it?

Back Yes


(5)- FORMALIZE WARNING WINDOW
Title
Reservation:
Castle:
Number of people:
Number of rooms:
Number of days:
Final price:
Starting date:
You won't be able to change the reservation if you formalized it. Do you want to formalize it?
Yes
Cancel

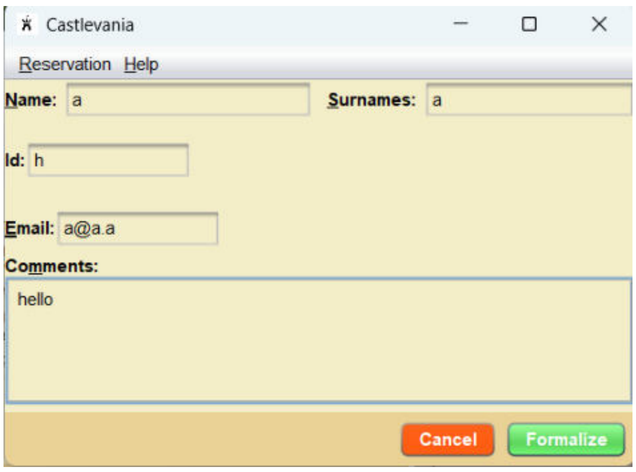
<u>*Changes*</u>:

- I change as before the Cancel button by a **Back button** and change the order of the buttons for consistency, and as the user won't lose anything if he goes to the previous view

<u>*Most relevant elements*</u>:

- A panel with Borderlayout for the **summary of the reservation**
- A panel with Borderlayout for the **buttons** (**Back**, goes to the previous view without losing the reservation, **and Yes**, it goes to the formalization window) **and the warning message**

Formalization Window



Castlevania

Reservation Help

Name: a **Surnames:** a

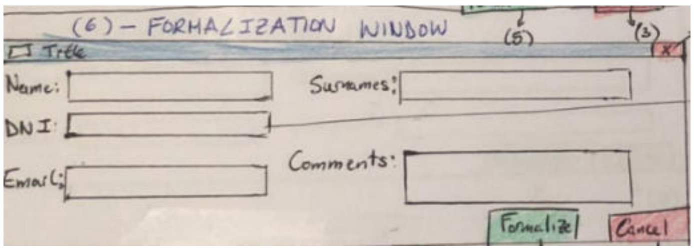
Id: h

Email: a@a.a

Comments:

hello

Cancel Formalize



<u>*Changes*</u>:

- I change the **position of the comments** as it has a greater size usually than email so it will fit better in the panel
- I **change the order of the buttons** to be consistent with the other buttons in the view (Left for cancel or back; and Right for going to the next view)

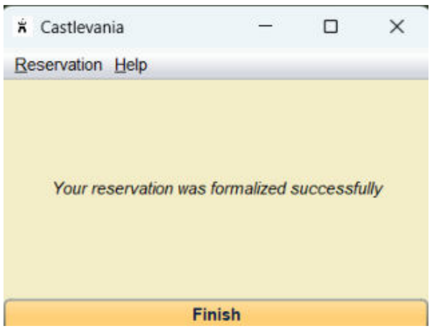
<u>*Most relevant elements*</u>:

- A panel for the **form** with a text field for every mandatory element as they only occupy one line, and a text area with a scroll pane for the comments

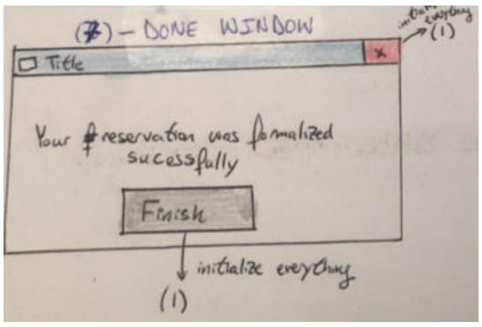
- In the **name and surname**, it won't be allowed numbers or weird characters (except blanks between words)
- In the **email** only this format will be accepted and ';' won't be accepted for security of the files: *something1*@*something2*.*something3*
- In the **comments**, the user can write as many lines as he wants with the text area, and can see the whole comment thanks to the scroll pane. ';' won't be accepted for security of the files, and also for security the new lines (\n) will be transformed to blank spaces

- A panel for **the buttons Cancel** (in this case I keep cancel as I the user goes to the previous view, he will lose the information filled in the formalization view), **and Formalize** (goes to the Done window when the form is correct)

Done Window



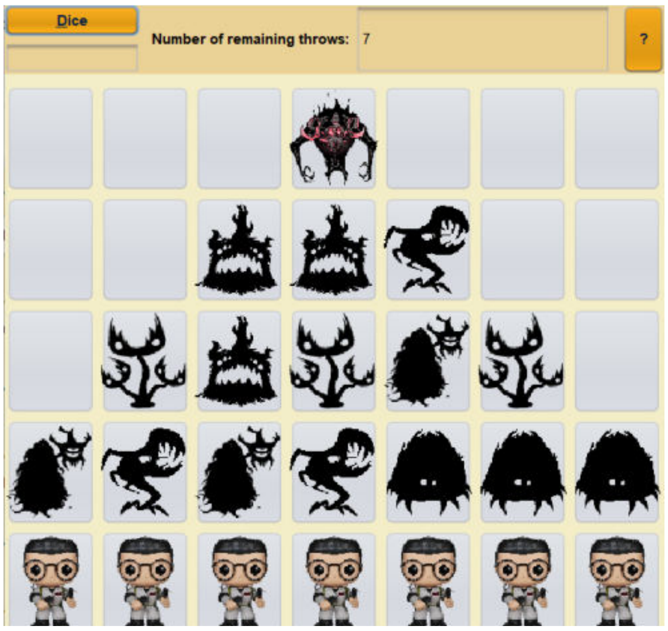
Your reservation was formalized successfully

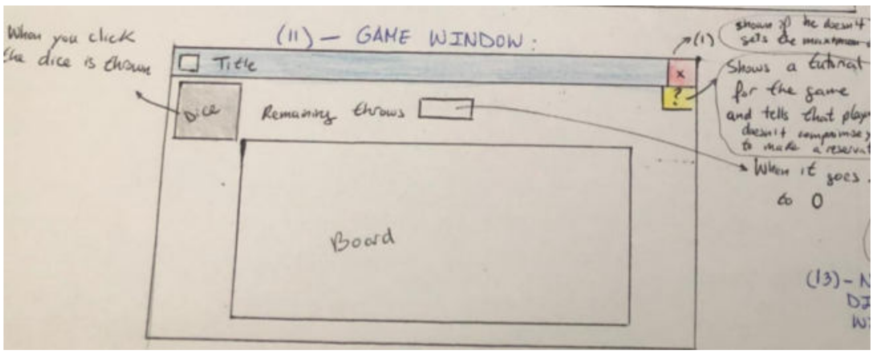


Most relevant elements:

- A label with the **message** saying that the reservation was saved
- A **Finish button** that initializes the whole app to be ready for a new user

Game Window




When you click the dice is thrown
(11) — GAME WINDOW:
Title
Dice
Remaining throws
Board
(1)
Shows a tutorial for the game and tells that player doesn't compromise to make a reservat
When it goes to 0
(13) -

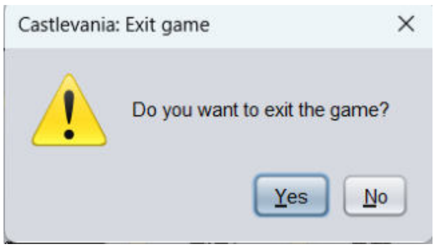
Changes:

- I added a text field with the **value of the dice**, so the user can know it
- Using **CardLayout** for the window so we have one panel for the game window, and another for notifying the user what he has gained after playing the game, and doesn't have to deal with several windows

Most relevant elements:

- **A panel for the header** with BorderLayout, where we have the dice, its value, the remaining throws and the game help button
- **A panel for the board** that is dynamically created one button per square of the board

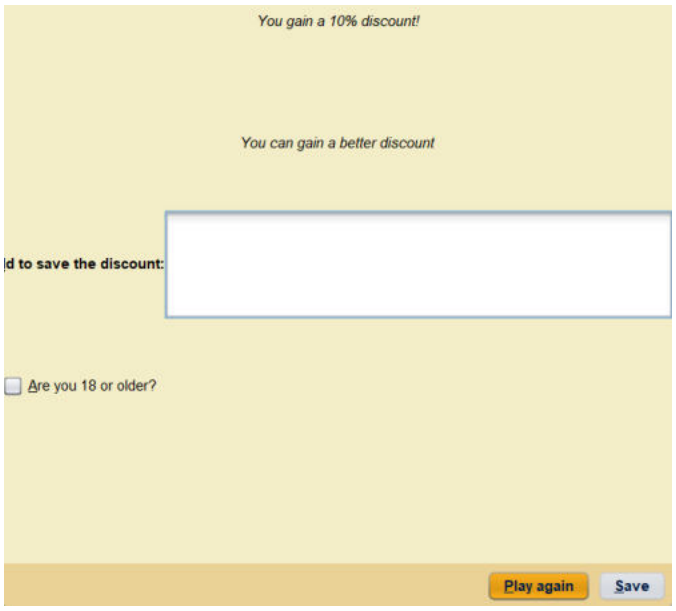
When you try to **exit the game** this window is shown:


Castlevania: Exit game

Do you want to exit the game?

If you click yes, the main window will be shown initialized

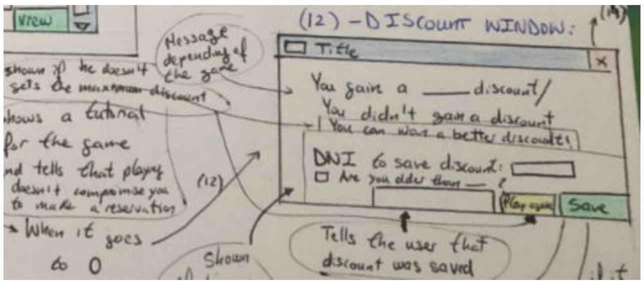
Discount window



You gain a 10% discount!

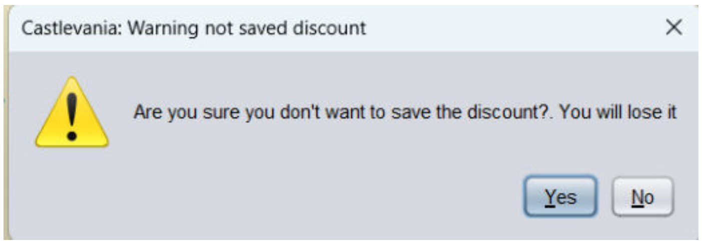
You can gain a better discount

Id to save the discount:


(12) – DISCOUNT WINDOW:
View
Title
You gain a ___ discount/
You didn't gain a discount
You can won a better discount!
DNI to save discount:
☐ Are you older than __?
Play again
Save
Message depending of the game
shown if he doesn't gets the maximum discount
shows a tutorial for the game and tells that playing doesn't compromise you to make a reservation
When it goes to 0
(12)
Shown
Tells the user that discount was saved
(14)

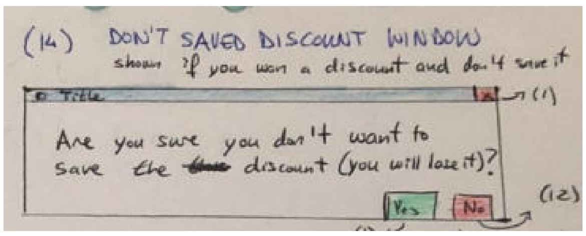
Most relevant elements:

- A panel with the buttons
  - **Play Again**: enabled when the user didn't gain the maximum discount
  - **Save**: enabled when the user gains the discount, introduces an Id and marks that he has the legal age
    - If the user **doesn't save the discount** and press a button or tries to close the window, this warning will be shown. If you don't want to save it, you will go back to the game.

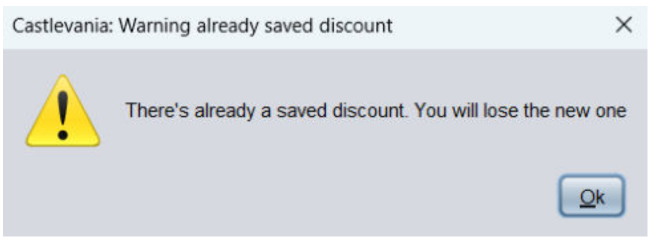


Castlevania: Warning not saved discount

Are you sure you don't want to save the discount?. You will lose it

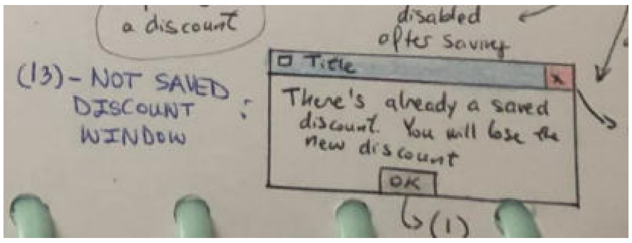

(14) DON'T SAVED DISCOUNT WINDOW
shown if you won a discount and don't save it
Title
(1)
Are you sure you don't want to save the ~~these~~ discount (you will lose it)?
Yes
No
(12)

- If the user when introduces an id, and has a **discount already saved**, this window is shown, and you will go back to the main window initialized



Castlevania: Warning already saved discount

There's already a saved discount. You will lose the new one


a discount
disabled after Saving
(13) – NOT SAVED DISCOUNT WINDOW:
Title
There's already a saved discount. You will lose the new discount
OK
(1)

- A panel with:
  - All the messages that tell the user what **discount they have won and if he can gain a better discount** (labels)
  - **Another panel that is only visible when the user gains a discount** with a text field **to enter the id, a checkbox to mark that the user has the legal age, and** a label that displays a **message when the user saves the discount**

Testing (for the Wireframe V.2)

Student developer: Paula Díaz Álvarez
Evaluating student: María Rodríguez Gomara

Person: Marga **Difficulties encountered**: ➢ The game doesn't indicate that playing doesn't compromises you to making a reservation ➢ The game after it ends, doesn't tell you that you can gain better discounts ➢ It doesn't appear a confirmation window when you don't want to save the discount **Interface improvement actions**: ➢ In the instructions must say that playing doesn't compromises you to making a reservation ➢ Play again saying that you can gain better discounts ➢ Window checking if the user doesn't want to save the discount	**Name**: Marga **Age**: 45 **Occupation**: Lawyer **Scenario**: Marga wants to surprise her partner about a trip for their anniversary. While waiting to be attended to at the travel agency, she approaches a terminal that announces an unforgettable stay and the possibility of obtaining discounts on your reservation. It is clear from the explanations on the terminal that playing does not compromise you to make a reservation, so tries her luck. There is no discount, but the app offers her to play again with no reservation commitment. Repeat the game and this time she gets a 10% discount. The application makes it clear that she could play again to obtain a greater discount, but since they are already going to attend her at the counter, she prefers to leave and not keep the discount obtained. So if she finally doesn't book a traditional trip, She will come back to try to get the maximum discount and book a haunted castle.
Person: Maya **Difficulties encountered**: ➢ It doesn't inform the user that you must be legal age to the discount **Interface improvement actions**: ➢ A notification must appear saying that she must be of legal age to save the discount (I use a checkbox)	**Name**: Maya **Age**: 11 **Occupation**: Student **Scenario**: Maya has accompanied her mother to a travel agency, where she plans to book to spend a few days together at Disneyland. While waiting, Maya notices the terminals that offer to play to get discounts on a very special stay. Play several times until she gets the maximum discount. When asking for the ID to save the discount, it clearly indicates that it must correspond to a person of legal age, so they do not save it. Out of curiosity, she checks out the list of available hotels and sees the type of experiences offered. She runs to his mother to tell her not to book at Disneyland but rather to change the trip to go to one of the castles, which she likes much more
Person: Antonio **Difficulties encountered**: ➢ After the reservation summary, it doesn't show a formalization window. **Interface improvement actions**: ➢ Extract the formalization part of the reservation window to a new window	**Name**: Antonio **Age**: 79 **Occupation**: Retired **Scenario**: Antonio is about to turn 80 and wants to celebrate with his family with a special trip that everyone will remember. He approaches one of the agency's terminals and clearly sees that before booking it is worth playing to try to get a discount. After reading the basic instructions of the game, play and

	earn a 25% discount. It is clear to him that it is the largest he can get, so he provides his ID to use later. He goes to book and analyzes the possibilities: he wants to see the castles that will offer all the enchantments except the smells, which makes him very sick. Finds one that fits him. He indicates that he is going to reserve 5 rooms for 12 people, but the system tells him that it is not possible, so he adds one more room. Antonio indicates that he wants to use the discount obtained. In the reservation summary he checks that everything is correct, with the final price already updated, so he formalizes the reservation by entering the information requested .
Person: Amy **Difficulties encountered**: None **Interface improvement actions**: None	**Name**: Amy **Age**: 22 **Occupation**: student **Scenario**: Amy has just graduated from college and her friends and she wanted to do something fun to celebrate. She played the game and won a discount. They are 5 friends and they tried to book 2 rooms for the five of them but the application didn't let them because the maximum of people per room is 2 so they had to book three rooms for the five of them.
Person: Juan **Difficulties encountered**: Checking the id (it is not required) **Interface improvement actions**: None	**Name**: Juan **Age**: 80 **Occupation**: retiree **Scenario**: Juan heard about the castles because his son told about them. His son had won a discount so Juan decided to try. When he was booking the room for his wife and him he tried to use the discount that his son had won but he couldn't use it because the discount was associated with his son DNI but not his. So he canceled everything, decided to play himself, won a discount and made the reservation with his own discount.
Person: Angel **Difficulties encountered**: None **Interface improvement actions**: None	**Name**: Angel **Age**: 45 **Occupation**: purchasing manager **Scenario**: His wife and two kids wanted to go to one of the castles. They let both of the kids play and each of them won a discount but the only one that they could keep was the first one. When they were booking the hotel room they tried to book just one for all of them but the máximum of people per room was 2. So they booked 2 hotel rooms and his wife and him each took one kid.
Person: Pierre	**Name**: Pierre **Age**: 19

Difficulties encountered: None **Interface improvement actions**: None	**Occupation**: student **Scenario**: He is a french student, he heard his Friends talking about the castles and wanted to try them. He finds it a bit difficult to understand the web page because it is in English. So he looked for a button to change the language and was able to do it. Now that he understands it he makes a reservation for a castle. A few days later he saw a castle that he liked better than the one he make the reservation to, but he couldn't change the reservation