Playing with the words of science online: birds of a feather flock together

Llop Agramunt, David
Curs 2013-2014

Directors: VANESA DAZA i ROSA ESTOPÀ
GRAU EN ENGINYERIA TELEMÀTICA

Treball de Fi de Grau
PLAYING WITH THE WORDS OF SCIENCE ONLINE: 
BIRDS OF A FEATHER FLOCK TOGETHER

David Llop Agramunt

FINAL DEGREE PROJECT
TELEMATIC ENGINEERING
ESCOLA SUPERIOR POLITÈCNICA UPF
2014

DIRECTOR OF THE PROJECT
Rosa Estopà, Vanesa Daza
To my lovely partner Núria and her future children.
Acknowledgement

Thanks to Miquel Cormudella for his patience and knowledge, to Núria Espuña for her love and guidance, to my family for support and to Rosa Estopà and Vanesa Daza for letting me being part of their project.
Abstract

This Project is the web implementation of a board game made by Rosa Estopà’s Team in the “Jugant a definir la ciència” (Estopà, 2013) project context. The aim of the project was to develop a tool for teaching entities, for children to use it using the new technologies and the games as a media to get children’s attention and help them learn about science. The game format is online and allows the user to play competing with a friend, the computer or himself, and it includes the choosing of the difficulty of the game. This document puts the game in the context of “Jugant a definir la ciència”, describes the design, the technical implementation of the game which has been done using the object oriented programming paradigm, and includes a user guide for the game.

Resum

Aquest projecte és la implementació web d’un joc de taula elaborat en el marc del projecte “Jugant a definir la ciència”(Estopà, 2013). El projecte s’ha desenvolupat amb intenció de ser una eina per a entitats educatives d’infants, que pugin utilitzar les noves tecnologies i els jocs com un recurs per atraure l’atenció dels infants i ajudar-los a treballar el lèxic científic. El joc és en format online, i permet a l’usuari jugar amb diverses modalitats: jugar individualment o competir amb un amic, amb l’ordinador i, a més a més, inclou la possibilitat de seleccionar la dificultat de la partida. Aquest document posa el joc en el context de “Jugant a definir la ciència”, descriu el disseny i la implementació tècnica del joc que ha estat feta usant el paradigma de la programació orientada a objectes i inclou una guia d’usuari del joc.

Resumen

Este proyecto es la implementación web de un juego de mesa elaborado en el marco del proyecto” Jugant a definir la ciencia”(Estopà, 2013). El proyecto se ha desarrollado con la intención de ser una herramienta para las entidades educativas de niños, que puedan usar las nuevas tecnologías e los juegos como un recurso para atraer la atención de los niños y ayudarles a trabajar el léxico científico. El juego es en formato online, y permite al usuario jugar en diversas modalidades: jugar individualmente o competir con un amigo o con el ordenador y además incluye la posibilidad de seleccionar la dificultad de la partida. Este documento pone el juego en el contexto de “Jugant a definir la ciencia”, describe el diseño y la implementación técnica del juego que ha sido hecho usando el paradigma de la programación orientada a objetos e incluye una guía de usuario del juego.
Summary

Abstract........................................................................................................................................vii

List of figures...................................................................................................................................xi

1. INTRODUCTION ...................................................................................................................... 1
   1.1 Motivation .............................................................................................................................. 1
   1.2 Objectives ................................................................................................................................ 1
   1.3 Context of the project .............................................................................................................. 1
   1.4 The original board Game ........................................................................................................ 2

2. PLANIFICATION ....................................................................................................................... 3
   2.1 The Scheduling Process .......................................................................................................... 3
   2.2 The Development Process ........................................................................................................ 3
       a) Analysis of the information system .................................................................................... 3
       b) Design of the game ............................................................................................................. 3
       c) Implementation of the game ............................................................................................. 3

3. TOOLS AND RESOURCES ..................................................................................................... 7
   3.1 The Java Programming Language ......................................................................................... 7
   3.2 Vaadin ..................................................................................................................................... 8
   3.3 The Eclipse tool ..................................................................................................................... 9

4. DESIGN ....................................................................................................................................... 11
   4.1 Project Requirements ............................................................................................................. 11
       a) Functional requirements ...................................................................................................... 11
       b) Non-functional requirements ............................................................................................ 12
   4.2 Functionalities ....................................................................................................................... 12
   4.3 Design decisions .................................................................................................................... 12
   4.4 Use Cases .............................................................................................................................. 13
       a) Use case: Choose game mode ........................................................................................... 14
       b) Use case: Choose game couples ....................................................................................... 15
       c) Use case: Play time attack ............................................................................................... 15
       d) Use case: Play versus a friend ......................................................................................... 16
       e) Use case: Play Versus computer ..................................................................................... 16

5. IMPLEMENTATION ................................................................................................................... 19
   5.1 Structure of the code .............................................................................................................. 19
   5.2 Implementation process ......................................................................................................... 21
       a) The Start class .................................................................................................................. 21
       b) The Block class ................................................................................................................ 21
       c) The Game class ................................................................................................................. 22
   5.3 Implementation handicaps ..................................................................................................... 24
   5.4 Evaluation and testing .......................................................................................................... 24

6. USER GUIDE ............................................................................................................................ 25
   6.1 Preparing the game ............................................................................................................... 25
   6.2 General in-game features ..................................................................................................... 27
6.3 Playing Time Attack or freely ................................................................. 27
6.4 Playing versus a friend ............................................................... 29
6.5 Playing versus the computer.................................................. 31

7. FINAL RESULT ......................................................................................... 33

8 CONCLUSIONS ....................................................................................... 35

9 FUTURE WORK ......................................................................................... 37

Bibliography ................................................................................................. 39

Annex ............................................................................................................. 41
List of figures

Figure 2.1: Gant diagram......................................................... 4
Figure 2.2: Gant diagram graphic.............................................. 5
Figure 3.1: Car class example.................................................. 8
Figure 3.2: Capture of Eclipse webpage................................. 9
Figure 3.3: Capture of the Eclipse text editor..................... 10
Figure 4.1: User case diagram................................................ 14
Figure 5.1: Class diagram.......................................................... 18
Figure 6.1: Initial Layout......................................................... 23
Figure 6.2: Initial Layout zoom in......................................... 23
Figure 6.3: Couple choosing layout........................................ 24
Figure 6.4: Couple choosing layout zoom in ......................... 24
Figure 6.5: Time Attack with three couples......................... 25
Figure 6.6: Time Attack with three couples zoom in........... 26
Figure 6.7: Time Attack finished............................................. 26
Figure 6.8: Versus a friend with six couples....................... 27
Figure 6.9: Player 2 turn............................................................. 27
Figure 6.10: Player 1 wins......................................................... 28
Figure 6.11: Draw................................................................. 28
Figure 6.12: Versus the computer with ten couples.......... 29
Figure 6.13: Versus computer middle game....................... 29
Figure 6.14: Versus computer. End of user’s turn............. 30
Figure 6.15: Versus computer end.......................................... 30
1. INTRODUCTION

This section explains the motivation for doing this project, its objectives and in which context is this project developed.

1.1 Motivation

The criteria for choosing a final degree project was based on two personal decisions. The first one was an interest on participating in a social project, where my work could help on providing tools to help society in some way, like helping in the development and education of children as in this case. The second one is to expand the knowledge I started learning in the subject “Programació Orientada a Objectes”, a subject that I enjoyed very much. I think that to work efficiently is important to enjoy while working and that’s why I wanted to practice more deeply what I learned. The subject was an introduction to the Object Oriented Programming paradigm. It awakened an interest in me that I couldn’t practice as much as I would have liked while I was studying it.

I took this project as a test to see how creative and efficient I could be, and how I want to orientate my future in the programming world, but also as a contribution to the children education in the country.

1.2 Objectives

The main objective of the project is to analyze, design, develop and evaluate an interactive application. The application will be used as educational tool for the scientific lexical learning oriented to children in Primary School. This tool will be a game application that will be played online from the “Jugant a definir la ciència” webpage. The game is an adaptation of the classic game “Memory”.

As part of the main objective, the game must be developed in the Object Oriented programming paradigm. The technical nature of the project makes the development of it a chance to practice the methodology of technical projects implementation.

1.3 Context of the project

IULA (University Institute of Applied Linguistics) is a research center of the Universitat Pompeu Fabra that unites investigators, collaborators and trainees, and participates in national and international organizations and networks, promoting research seminars and scientific activities periodically.

The IULA investigator Rosa Estopà and her team created the “Jugant a Definir la ciència” project with the initiative of creating an environment of learning science for children. In the “Jugant a definir la ciència” project they assume that the bases of the specialized knowledge are acquired during the first living years of a person, and that we humans have significant conceptions that will help in the development of the scientific knowledge. Main study focus is the basic words of science in the context of children
between 6-8 years old, like water, space, star, brain, ice, death, sun, heat, speed, air, life, etc.

The main goal is to unite material that allows the study of what children know about the basics of science in their first years of school and provide proper materials for them, such as ludic material to work with scientific vocabulary, elaborating a dictionary done with school definitions and creating a catalog of the cognitive representations of the scholars previous to any scientific knowledge.

Nowadays, the project counts with a purchasable collection of dictionaries (“My first Dictionary of Science” (Estopà, 2013)[1]), games (“La Maleta Viatgera”, “Juga amb les paraules de la ciència: Taller de jocs gegants”), applications (“Club LEXIC”, “El Microscopi”) and publications (“Jugant a definir la ciència: un dicionari de mots de ciència fet per I per a nens I nenes”(Estopà, 2011)[2], “El Microscopi. Banc obert de definicions terminològiques I catàleg de representacions” (Cornudella, 2013) [3], “Recursos per treballar el lèxic acadèmic col·laborativament” (Estopà, 2013) [4]).

The game “The travelling suitcase” is in fact a pack of five different playable board games in a “suitcase”. These ones are intended to exercise the skills on relations, memory, concentration, speed, visual speed, analysis, construction and mathematic analysis. The board game that specifically focuses on memory and concentration is the “Cada ovella amb la seva parella” (The expression “Birds of a feather flock together”), an adaptation of the classical Memory Game.

In this context, the project “Playing with the words of science Online: Birds of a feather flock together” detailed in the next chapters is the computer version of the “Cada ovella amb la seva parella” game, which will be available to play Online.

1.4 The original board Game

This game uses the same game structure as the classic Memory game, where the goal is to remember what’s on the different cards and try to find the cards that are a couple. The classical Memory game consists just on finding the same image two times, fomenting only the memory exercise. The game “Cada ovella amb la seva parella” goes further, using studied concepts, images and drawings that make the users develop not only the memory but also the relations between images that represent concepts and words.

This document is structured in sections, we will see the planification of all the project in section 2, the tools and resources used in section 3, the design made in section 4, the explanation of the implementation in section 5, a detailed user guide in section 6 and then a valuation of the final result on section 7. The conclusions of the project are in section 8 and section 9 considers future work related to the project. Not considered as sections, in the end the bibliography and annexes can be found.
2. PLANIFICATION

As a general approach, I have used the waterfall model for the project planification. This is because the project is based on the completion of different phases in a certain order, but in this case is not a pure waterfall model because the phases are not completed strictly one after the other. I’ve used some Rapid Application Development (RAD for short) methodology keys for the planification of the project because I think that the most effective key of the developing of a game is the iterative development and the construction of prototypes. This attempts to reduce inherent project risk by breaking a project into smaller segments and requires active user involvement.

2.1 The Scheduling Process

Different phases of the project are decided during the scheduling process. After this phase, the Development Process starts.

2.2 The Development Process

a) Analysis of the information system

For this project, the information system refers to the technology used, the software and development tools. The steps consist on learning about this information system and getting to know what implementations can be done with it. Afterwards, the step is to install these tools needed and finally to get in contact with the theoretical subjects looking for the maximum knowledge available to be able to start the design.

b) Design of the game

The first step of designing the game is understanding how it should work in the end. The design is based on the structure of how the elements used behave and how they relate between them.

This phase is critical, because a bad or not optimized design affects deeply in the project. The usual consequences of a bad design are visible in the implementation phase, where the bad design implies an impossible or non-optimized implementation, meaning a mandatory re-design and the loss of resources. As a final step of the “Design of the game” and as a link to the “Implementation of the game” phase there is the realization of a first non-functional prototype of the interface.

c) Implementation of the game

The implementation of the game is the execution of the design made in the previous phase. First step of the phase is creating a primitive initial prototype, then a functional prototype until a final prototype. As it has been said, the design has a lot of influence in the implementation steps. This phase includes the testing of the game in the functional and final prototype.
A documentation has been made as part each phase to follow the evolution of all the tasks and help with the realization of this document. With the previous stages of the planification decided, I created a Gantt diagram to have a clear and reliable planification (see Figure 2.1 and Figure 2.2)

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Degree Project</td>
<td>504 hrs</td>
<td>Wed 12/02/14</td>
<td>Mon 12/05/14</td>
</tr>
<tr>
<td>Scheduling Process</td>
<td>8 hrs</td>
<td>Wed 12/02/14</td>
<td>Thu 13/02/14</td>
</tr>
<tr>
<td>Study of the tasks and phases of the project</td>
<td>2 days</td>
<td>Wed 12/02/14</td>
<td>Thu 13/02/14</td>
</tr>
<tr>
<td>Development Process</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis of the information system</td>
<td>496 hrs</td>
<td>Thu 13/02/14</td>
<td>Mon 12/05/14</td>
</tr>
<tr>
<td>Initial Abstract</td>
<td>136 hrs</td>
<td>Thu 13/02/14</td>
<td>Mon 10/01/14</td>
</tr>
<tr>
<td>Analis of the technology to apply</td>
<td>0 days</td>
<td>Thu 13/02/14</td>
<td>Thu 13/02/14</td>
</tr>
<tr>
<td>Search and installation of the development Software</td>
<td>4 days</td>
<td>Thu 13/02/14</td>
<td>Mon 17/02/14</td>
</tr>
<tr>
<td>Study of the utilization of the software</td>
<td>10 days</td>
<td>Mon 17/02/14</td>
<td>Mon 24/02/14</td>
</tr>
<tr>
<td>Documentation</td>
<td>20 days</td>
<td>Mon 24/02/14</td>
<td>Mon 10/03/14</td>
</tr>
<tr>
<td>Design of the game</td>
<td>34 days</td>
<td>Mon 10/03/14</td>
<td>Mon 10/03/14</td>
</tr>
<tr>
<td>Design of the User Interface</td>
<td>140 hrs</td>
<td>Mon 10/03/14</td>
<td>Thu 03/04/14</td>
</tr>
<tr>
<td>Design of the Classes</td>
<td>5 days</td>
<td>Mon 10/03/14</td>
<td>Thu 13/03/14</td>
</tr>
<tr>
<td>Prototype of the interface non-functional</td>
<td>25 days</td>
<td>Mon 13/03/14</td>
<td>Mon 31/03/14</td>
</tr>
<tr>
<td>Documentation</td>
<td>5 days</td>
<td>Mon 31/03/14</td>
<td>Thu 03/04/14</td>
</tr>
<tr>
<td>Implementation of the game</td>
<td>35 days</td>
<td>Mon 10/03/14</td>
<td>Thu 03/04/14</td>
</tr>
<tr>
<td>Initial Prototype</td>
<td>220 hrs</td>
<td>Thu 03/04/14</td>
<td>Mon 12/05/14</td>
</tr>
<tr>
<td>Functional Prototype (Beta)</td>
<td>16 days</td>
<td>Thu 03/04/14</td>
<td>Tue 15/04/14</td>
</tr>
<tr>
<td>Final Prototype</td>
<td>24 days</td>
<td>Tue 15/04/14</td>
<td>Thu 01/05/14</td>
</tr>
<tr>
<td>Documentation</td>
<td>15 days</td>
<td>Thu 01/05/14</td>
<td>Mon 12/05/14</td>
</tr>
<tr>
<td>Documentation</td>
<td>55 days</td>
<td>Thu 03/04/14</td>
<td>Mon 12/05/14</td>
</tr>
</tbody>
</table>

Figure 2.1: Gant diagram.
Figure 2.2: Gant diagram graphic.
3. TOOLS AND RESOURCES

The Tools and Resources section describes all the software used during the development of the project. Microsoft Word 2013 has been used to write the documentation of the project, and MS Project 2013 to create the Gantt diagram. The programming tools and resources used to program in the Java programming language have been the Eclipse programming tool and the Vaadin plug-in for Eclipse.

3.1 The Java Programming Language

During my university years I’ve seen several ways of programming, including languages like C, C++, C#, Java, Octave, Phyton, Scilab or Visual Basic.NET. I’ve specially developed an interest in the Object Oriented Programming paradigm.

Different programming paradigms exist in the world of programming. Each of them are based on rules and statements that define how the programmer must order his ideas to create programs. One of these paradigms in the programming world is the Object Oriented Programming paradigm. As its name says, it is based on the existence of “objects”. The idea of this paradigm is having different “objects” that specialize in different functions, and the interaction between them allow the program to execute complex tasks. The “objects” have data fields known as attributes that describe the object and associated procedures known as methods. Some of the languages that use this paradigm are C++, Objective-C, C#, Perl, Python, Ruby, PHP, Smalltalk or Java. Specifically, the language chosen for this project is the Java language, one of the most popular programming languages in use. Two books to get started on understanding Java are “Java Programming Language (4th Edition) [(Ken Arnold, 2005) [5] and “Head First Java (2nd Edition)” [(Sierra, 2005) [6].

Java is a concurrent, class-based, object-oriented computer programming language, and its creation was made looking forward to let application developers “write once, run anywhere”, or WORA, which means that the code that runs on one platform doesn’t need to be recompiled every time that has to be run on another platform. The nature of Java is that is compiled to bytecode (or else named “class file”), and this means that can run on any JVM (Java Virtual Machine, a virtual machine that can execute Java bytecode).

In Java, the programmer creates “classes”, and every class has “attributes” and “methods”. Attributes can be, among others, integers, floats, Booleans or strings, and they have a certain value. Methods are functions in charge of “doing” things, so by definition are the ones that interact with the attributes of the instances. If classes are the objects that define “what kind of characteristics”, “instances” of those classes are the ones that specify the values of these characteristics. For example, one could have the class “Car”, with the attribute “weight” as float and “brand” and “color” as strings, and the method “getcolor” that returns the value of the “color” attribute. An instance of this class would be one with “Ferrari” as the value for the string attribute “brand”, “red” for the “color” attribute and “100” for the value of the “weight” attribute.

In a few words, this means that the class defines how the instances of that class will be, and the instances are just “real examples” of the class that can interact between them.
A visual example of a class “Car” in Figure 3.1:

![Car class example](image)

Figure 3.1: Car class example.

### 3.2 Vaadin

Vaadin is an open source Web application framework. It has the singularity of working with server-side Java code, differentiating it from client-oriented java frameworks such as GWT. The major part of the logic runs on the servers, and it is defined as a robust architecture for rapid application development thanks to the tools in it, that helps the programmer to build web user interfaces extremely fast. Vaadin is designed to be compatible with all the internet browsers in order to let the programmer focus on developing the application.

Built-in themes like Reindeer, Runo or Cameleon Theme, the option of customizing own themes, creating own components and using and creating add-ons are features of Vaadin that follow the goal of focusing in the easy development of the application. The documentation, tutorials and community makes relatively easy to learn how to use Vaadin. For this project, the tutorials, the Book of Vaadin¹ and the forum have been particularly useful. The tutorials are clearly structured, with sound and image quality, and with the steps specified to start a project with Vaadin. The Book of Vaadin is the full reference for Vaadin. It shows you how to get started and is a written guide of all the features that Vaadin has, their functionalities and how to use them. The Community disposes of a forum, a blog, and a wiki that contain many common code doubts and problems solved and allows anyone to participate in them by registering in.

“Vaadin is a Java framework for building modern web applications that look great, perform well and make you and your users happy.”²

When I learned about Vaadin, I realized that it was the kind of tool that I needed to implement the game. Since Vaadin is also open source, it has lots of published information about how does it work and what can you do with it.

---

² [www.vaadin.com](www.vaadin.com)
The main goal of Vaadin is to help programmers to create user interfaces. This is done through the hundreds of components that are available and the agility that provides the fact that it is built on HTML5. This allows the programmer to move applications to the web without having to do installations or plug-ins. The Server-Client concept is used with the intention of having the benefit of the speed of the server combined with the added flexibility of client-side solutions.

There are several ways of getting Vaadin into a project including Maven, the Eclipse plug-in or the All-in-One archive, but for this project the chosen one has been the Eclipse plug-in because I am familiar with Eclipse and feel comfortable working with it.

### 3.3 The Eclipse tool

The Eclipse Tool is a software composed by a group of multiplatform open-source programming tools that are capable of designing Client applications, typically used to develop IDE’s (Integrated Development Environment). This tool has a text editor with syntax exaltation that allows the user to write the code in java. It compiles in real time and can create projects and classes, making it a useful tool to program in Java.

The Eclipse Tool is part of the “Eclipse community”, developed by the Eclipse Foundation, an open source community (see webpage capture in Figure 3.2)

![Eclipse webpage](image)

3 www.eclipse.org
The following capture (Figure 3.3) is from the Eclipse interface. The automatic identification of certain concrete words, numeration of the lines and tabulation makes the understanding of the code a lot easier, and so makes the programming more agile.
4. DESIGN

This part of the document explains the criteria used in every step of the design of the game, the requirements asked, the functionalities, the relations between the elements and its behavior.

The design is the most important phase of the project. The reason is that the implementation of the game is strictly related to the design, so if the design is wrong the implementation can reveal that the game can’t accomplish the functionalities expected, and then a re-design is required and a new implementation as well, loosing valuable resources.

A good design implies a clear and optimized structure of the elements that are involved and how are they related. There are different ways to program and different designs that can have the same results, but the best design is the one with the qualities of efficiency and scalability.

4.1 Project Requirements

The first thing done in the project was establishing the project requirements. These were accorded with the tutor Rosa Estopà and his colleague Miquel Cornudella. The requirements are the following:

- The game: Cada Ovella amb la seva parella
- The platform: WEB
- The software: Eclipse and Vaadin

The first requirement was deciding what game was to be done. As it has been said in the context of the project, there exist various games, but the one decided was the “Cada Ovella amb la seva parella”. The second requirement was on what platform would the game be available, and since the internet is nowadays the most accessible resource (not everyone has devices like Tablets, but most of the schools and houses have internet⁴) it was established that the game had to be played through a webpage.

In order to develop a game playable in a webpage, the tools Eclipse and Vaadin were proposed, because they’re tools used to create web applications with easy interface features, and the “Jugant a definir la ciència” project webpage had already been developed with these tools.

a) Functional requirements

The functional requirement is describing the behavior of the system as it relates to the system's functionality. For this project, the system must allow the user to navigate between different scenarios and choose game parameters, and then it must behave as specified depending on the buttons and cards pressed.

⁴ http://data.worldbank.org/indicator/IT.NET.USER.P2
b) Non-functional requirements

The non-functional requirement elaborates a performance characteristic of the system. The performance requirements are having access to a computer with internet connection, platform constraints are having any of the common internet browsers specified (Firefox, Chrome, Explorer).

4.2 Functionalities

The main functionalities obtained from the resulting code are the following:

- Capability of launching the application on Internet Explorer 8, Firefox, Chrome, Safari, Opera and all the new versions of the browsers.
- Launch the game from different devices and browsers with complete independence.
- Play a single game, choosing between “time-attack” and “versus the computer”.
- Play a multiplayer game, choosing “versus a friend”.

4.3 Design decisions.

Being part of “Jugant a definir la ciència” has conditioned and facilitated the design of the game. The design needed to be strictly related to the design of the original game and webpage⁵.

For the interface design of the application I have used the images annexed to this document selected by Rosa Estopà, and the cards that are played during the game are the same as the ones used in “Cada ovella amb la seva parella”.

Since the users will mostly be children, the different scenarios on the game are very similar, and the user can interact with a different number of buttons, making the interaction with the game graphically very clear and easy for the user.

The different scenarios in which the user can navigate share the same background but the elements change. Also, the cards change every time that a scenario with cards is generated, so that the game has the maximum variability and randomness to increase the “play again” value.

What this design accomplishes is an easy understanding of how to play the game and at the same time, the techniques that make a game able to be replayed, encouraging the users to use the application as long as wanted.

⁵ http://defciencia.iula.upf.edu/index.htm
There are three main buttons to choose the game mode:

- **Jugar a contrarelle o lliure**: “Play freely or in time-attack mode”. This button prepares the game to be played in a scenario where only one player can play, freely or in time-attack mode.
- **Jugar amb un amic**: “Play with a friend”. This button prepares the game to be played in a scenario where two people can compete against each other.
- **Jugar contra l’ordenador**: “Play against the computer”. This button prepares the game to be played competing against the computer. The computer will play with artificial intelligence, trying to play like a real person.

As parameters of the game scenario, the number of couples must be decided:

- **2 parelles, 3 parelles, 4 parelles, 5 parelles, 6 parelles, 7 parelles, 8 parelles, 9 parelles, 10 parelles**: Buttons to select how many couples the user wants to play with.
- **Torna al lloc anterior**: Button to return to the previous scenario.

In the time-attack or Free scenario

- Card button: Button that contains the card which the user plays with.
- **Torna al lloc anterior**: Button to return to the previous scenario.
- **Juga un altre cop**: Button to restart the current scenario.
- **Comença el temps!**: Button to start the time counter.
- **Fi!**: Button to stop the time counter and to show the time that has passed from the time that the Comença el temps! button was pushed. This button is initially disabled to avoid confusing the user, and gets enabled once the Comença el temps! button is pushed.

In the versus a friend scenario

- Card button: Button that contains the card which the user plays with.
- **Torna al lloc anterior**: Button to return to the previous scenario.
- **Juga un altre cop**: Button to restart the scenario.

In the versus the computer scenario

- Card button: Button that contains the card which the user plays with.
- **Torna al lloc anterior**: Button to return to the previous scenario.
- **Juga un altre cop**: Button to restart the scenario.
- **Seguent ronda**: Button that lets the computer move when it is his turn.

### 4.4 Use Cases

Use cases show the reactions to the actions that the users can do over the application. The use cases show the behavior of the game during all its life, and allows to analyze the specific functionality of it. To understand the use cases relations I made the user case diagram shown in Figure 4.1
The Use Case template is formed by a number of fields next described that define the information on the use cases:

- **Use Case**: Title of the use case.
- **Actor**: An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks.
- **Description**: Provides a brief description of the sequence of actions and the outcome of executing the use case.
- **Preconditions**: List of activities that must take place, or any conditions that must be true, before the use case can be started.
- **Postconditions**: Describes the state of the system at the conclusion of the use case execution.
- **Normal course of events**: Provides a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions.
- **Alternative courses**: Documentation of other legitimate usage scenarios that can take place within this use case separately in this section.
- **Includes**: List of any other use cases that are included or called by this use case.
- **Notes and Issues**: Additional comments about this use case.

![User case diagram](image)

Figure 4.1: User case diagram.

**a) Use case: Choose game mode**

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Choose game mode</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>User</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>The user has to choose one of the three buttons corresponding to the three game modes</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>The user has started the application to play the game</td>
</tr>
<tr>
<td><strong>Postconditions</strong></td>
<td>The configuration related to the game mode selected is prepared and a new scenario is charged.</td>
</tr>
</tbody>
</table>
b) Use case: Choose game couples

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Choose game couples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>User</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>The user has to choose the quantity of couples for the game.</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>The user has chosen a game mode.</td>
</tr>
<tr>
<td><strong>Postconditions</strong></td>
<td>All the game features have been chosen, the game will start.</td>
</tr>
<tr>
<td><strong>Normal course of events</strong></td>
<td>The user clicks on the button that represents the number of couples desired, the scenario has all the information ready to start the game. A label indicates the game mode chosen.</td>
</tr>
<tr>
<td><strong>Alternative courses</strong></td>
<td>The button to go to the previous scenario is pushed, the actual scenario is deleted and the previous scenario is charged again.</td>
</tr>
<tr>
<td><strong>Includes</strong></td>
<td>If alternative course, “Choose game mode” use case.</td>
</tr>
<tr>
<td><strong>Notes and issues</strong></td>
<td>-</td>
</tr>
</tbody>
</table>

c) Use case: Play time attack

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Play Time Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>User</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>The number of couples and the game mode is displayed and ready to start playing.</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>The user selected the Time-Attack mode and a number of couples.</td>
</tr>
<tr>
<td><strong>Postconditions</strong></td>
<td>Push the “Fi” button and check the time.</td>
</tr>
<tr>
<td><strong>Normal course of events</strong></td>
<td>The number of couples selected is displayed. A label with the game mode is displayed. The buttons “Torna al lloc anterior”, “Juga un altre cop” and “Comença el temps!” are displayed. The “Fi!” button is displayed but disabled. The user clicks the “Comença el temps!” button to start the Time-Attack and starts to play.</td>
</tr>
<tr>
<td><strong>Alternative courses</strong></td>
<td>The user can push the “Torna al lloc anterior” button to get back to the previous scenario. The user can push the “Juga un altre cop” button to start a new game with the</td>
</tr>
</tbody>
</table>
d) Use case: Play versus a friend

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Play Versus a friend</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>Two users</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>The number of couples and the game mode is displayed and ready to start playing.</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>The user selected the versus a friend mode and a number of couples.</td>
</tr>
<tr>
<td><strong>Postconditions</strong></td>
<td>A warning informs of who is the winner.</td>
</tr>
<tr>
<td><strong>Normal course of events</strong></td>
<td>The number of couples selected is displayed. A label with the game mode is displayed. Two labels “Player 1” and “Player 2” are displayed with its score. The buttons “Torna al lloc anterior” and “Juga un altre cop” are displayed. Every user has 2 clicks available in their turn.</td>
</tr>
<tr>
<td><strong>Alternative courses</strong></td>
<td>The user can push the “Torna al lloc anterior” button to get back to the previous scenario. The user can push the “Juga un altre cop” button to start a new game with the same game features that were selected.</td>
</tr>
<tr>
<td><strong>Includes</strong></td>
<td>For the first alternative course, “Choose couples” use case. For the second alternative course, “Play Time-Attack” use case.</td>
</tr>
<tr>
<td><strong>Notes and issues</strong></td>
<td>The buttons specified can be pushed any time.</td>
</tr>
</tbody>
</table>

e) Use case: Play Versus computer

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Play Versus computer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor</strong></td>
<td>User</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>The number of couples and the game mode is displayed and ready to start playing.</td>
</tr>
<tr>
<td><strong>Preconditions</strong></td>
<td>The user selected the Play Versus Computer mode and a number of couples.</td>
</tr>
<tr>
<td><strong>Postconditions</strong></td>
<td>A warning informs of who is the winner.</td>
</tr>
<tr>
<td><strong>Normal course of events</strong></td>
<td>The number of couples selected is displayed. A label with the game mode is displayed. Two labels “Player 1” and “Computer” are displayed with its score. The buttons “Torna al lloc anterior” and “Juga un altre cop” are displayed. Every time the user finishes his turn, the button</td>
</tr>
<tr>
<td>Alternative courses</td>
<td>“Següent ronda” is enabled. The user can push the “Torna al lloc anterior” button to get back to the previous scenario. The user can push the “Juga un altre cop” button to start a new game with the same game features that were selected.</td>
</tr>
<tr>
<td>Includes</td>
<td>For the first alternative course, “Choose couples” use case. For the second alternative course, “Play Versus computer” use case.</td>
</tr>
<tr>
<td>Notes and issues</td>
<td>The buttons specified can be pushed any time.</td>
</tr>
</tbody>
</table>
5. IMPLEMENTATION

This chapter describes how the game was structured and implemented, explaining the criteria followed and detailing the code parts that help to understand the operation of the game.

5.1 Structure of the code

The structure of the code is based on the design parameters and functionalities. For this project and the way it has been approached, a not very complex structure has been made. The complexity of the code is visible in the implementation of the in-game behavior and the detailed artificial intelligence criteria.

Three classes have been designed and details are explained in Section 5.2.

The class Start has been designed to start running the game, generating the layout and setting the game ready. The class Block manages the structure and behavior of the main element of the game, the cards that will be used to play. The most complex class is the Game class. This class defines the layout characteristics and manages all the game mode rules and elements and the artificial intelligence of the computer player.

In the structure, these class characteristics are reflected on the number of elements conforming each class (See class diagram in Figure 5.1)
Figure 5.1: Class diagram.
5.2 Implementation process

Once the structure was defined, the implementation started. The understanding of the relations and dependencies between the elements of the code defined the implementation process schedule.

As it was planned, in this part of the project the methodology was to develop a functionality, test it, and then evaluate how this functionality influences the game. Deciding to include or eliminate functionalities was done after seeing the result of the implementation, and that meant modifying the design as well.

In this part of the document, a non-technical language is used to explain the code characteristics.

a) The Start class

The first part implemented was the file Start.java that contains the “Start” class. This class is the first code part of the game, so it contains the method that executes the code.

The function of this class is to start the application and create the initial layout of the game. To do this, the method InitialMenu was created.

InitialMenu: This method creates an instance of the class Game and it sets the layout element as main layout to set the background layout of the game. Then, adds the three buttons that allow the user to choose the game mode to the layout.

b) The Block class

In the structure it was decided that a class “Block” would be the best approach on managing the card elements of the game. A block represents two cards that form a couple, and during the game this class has to be instanced very often to generate cards.

The methods of this class are getback, and the buttons of the two cards.

getback: This method turns around the two buttons that represent a couple of cards. This is done by taking the reverse image of a card and setting it as the button image.

“Card” button: This method is launched every time that the button related to the card is pressed. When is called, the card checks if the other card of the same block has been pressed, considering the couple completed. It also calls a method checkreset from the class Game to check the state of the remaining cards on the board.
c) The Game class

The game class is the class that represents all the interaction between the user and the game. Most part of the code is in this class because the interaction with the game and the artificial intelligence are the densest parts. This is due the specific behaviors of the game in certain points that imply exceptions instead of rules.

SecondMenu: When one of the three first buttons (Time-Attack, Versus a friend, Versus the computer) is pushed, this method is called. The first thing that the method does is removing the three previous buttons, then it recognizes the game mode selected and indicates it in a label, adds the button “Torna al lloc anterior” and adds the buttons corresponding to the number of couples to choose.

“Torna al lloc anterior” button: This button removes the elements on the layout and adds the game mode buttons, recreating the starting scenario.

“couples” button: This buttons all call a method named thebuttoneffect. Depending on which button is pressed, the method gets an input (an integer) representing the number of couples chosen.

thebuttoneffect : This method creates the game board. It removes the previous components of the layout and then adds a grid created with getGameGrid that is full with the cards created with createBlocks and AddBlocksToUI. Then, depending on the game mode chosen it customizes the layout with customizegame and also adds the buttons “Juga un altre cop” and “Torna al lloc anterior”.

getGameGrid : It creates a grid and dimensions it depending on the couples chosen.

createBlocks: Depending on the number of couples chosen, this method randomly chooses couples from a list of possible cards in the game and generates the buttons representing the cards.

addBlocksToUI: Method in charge of putting the cards in the grid.

“Juga un altre cop” button: The button to restart the game. When pressed, this button restarts the variables of the game selected and calls the thebuttoneffect method.

“Torna al lloc anterior” button: This button is the mentioned in thebuttoneffect method. In his case, the button removes all the components on the layout and then calls the SecondMenu method.

customizegame: The common parts of the layout are added in the thebuttoneffect method, but the specific elements for the different game modes are added in this method. If the game mode selected was “Time Attack”, the buttons “Comença el temps” and “Fi!” are added. “Comença el temps” starts the time counter and “Fi!” finishes the count and notifies the user the time that has passed. If he game mode selected was “Versus a friend”, this method adds two labels “Jugador 1” and “Jugador 2” and its respective score counters. Finally, if the game mode is “Versus the computer”, the method adds the button “Seguent ronda” that allows the user to go to the
next round, and also adds a difficulty box to choose the computer difficulty and the labels “Jugador” and “Ordenador” with its score counters.

getRandomList: Method that creates a list of random numbers. Used as input for the createBlocks method.

gerButtons: A method that gets a random list of cards out of the list of blocks and serves as input for the AddBlocksToUI method.

checkreset: This method is called every time that a card button is pressed. The function of the method is to control and check what the response is every time that a card is pressed. It considers the case where the first card is pressed, the second one is pressed and the third as well, executing the reactions related when there are couples found or changes of turn. Finally, it checks the game mode and calls the vsmode method if necessary.

vsmode: When playing one of the versus modes, this method is called every time that a card is pressed. It designates when the turn changes for every player, assigns points to the counters when a player discovers a couple and tells who the winner is or if the score is a draw once the game is over.

reset: When the checkreset method concludes that the cards pressed have to be turned around, the reset method is called. Its function is to turn around the cards shown if they are not a couple.

addseenlist: This method is part of the artificial intelligence created for the computer. It checks and updates a list of cards that has been seen by any player in the current game, in order to represent the memory of the computer.

playrandomly: This method is the behavior of the computer when playing with the lowest difficulty. It chooses two random cards, checks that they have not been already discovered or pressed in the current turn and then presses the cards.

artificialIntelligence: Every time that the user presses the “Seguent ronda” button, this method is called. This method defines all the behavior of the computer when playing, emulating a real player in his turn. When playing against the computer, all the cards are disabled after the user’s turn to prevent the user from cheating, so the first thing that this method does is enabling all the cards except for the couples found.

Depending on the difficulty chosen, the method decides at this point to play randomly (low difficulty) or to play smart (high difficulty). The difficulty criteria is that a random number is compared to the difficulty chosen by the user every turn. If that random number is bigger than the difficulty chosen, the computer plays randomly on that turn. The result is that for a low difficulty level, there is a low probability of smart play and high probability of random play.

If it plays smart, the next step is to update the “seen list” of the computer that works as the memory of the game. To update it, the method eliminates the already completed couples from the seen list to prevent the computer from pressing completed cards.
The method considers three possible scenarios. The first is the case where there are no cards shown before the choosing, in the second there are two cards shown before choosing and in the third case there is only one card already chosen. When there are no cards shown before choosing, the method checks if there are any seen couples to push, or if it has to push a random first card. In the “one card pressed” scenario, the method checks if this first card is couple with any of the seen list or it presses the second card randomly. For the case of two cards already shown, the method behaves very similar but avoiding to press a card already pressed because this is not a valid move in the game, so if the computer has seen a couple (including the cards already shown), it selects the couple but first pressing the card of the couple that is not shown. If there are no couples seen, it presses one card randomly, leading to the “one card pressed” scenario explained previously.

5.3 Implementation handicaps

Even aiming for the most optimistic planification possible, in the implementation phase of the project is where more trouble a programmer can have in a matter of problematic surprises. In this part of the document I explain the major implementation handicap that have affected the project scheduling.

The Vaadin plug-in for Eclipse is a software still in development, and this means that some functionalities and behaviors are not completely controlled. The Vaadin plug-in offers a specific tool to create user interfaces easily, and the original plan included using this specific tool. The tool guaranteed a fast intuitive user interface design, but the freedom is limited and the tool is not perfectly effective, forcing the programmer to dead ends.

This was solved using the classical code methodology instead of this tool. The impact was not high because of the Vaadin specific code features to facilitate the creation of user interfaces.

5.4 Evaluation and testing

To evaluate the application I looked for the critic opinion of different professionals related to technology and human matters. Those are friends, family and university colleagues that are audiovisual or informatic engineers, sociologists, anthropologists, primary school teachers, graphic designers or pediatric doctors. Every member of this group tested the application and provided a detailed feedback that helped to improve both technical and playing behaviors.

The technical testing of the application was made by pressing all the possible buttons in all the possible combinations. That way, all legal or illegal moves were checked and all the unpredicted behaviors detected. The control was made including messages in the code that were launched in the different scenarios, helping me to confirm the correct behavior of the application.
6. USER GUIDE

This user guide is an explanation of how the user will be able to use the game, showing the different scenarios that are available for the user to go through and the possible decisions that can be made.

6.1 Preparing the game.

When the application is launched, the first scenario that the user finds is an initial layout with the background of the game, and three buttons (Figure 6.1).

The user must push one of the three buttons to choose the game mode that is willing to play (Figure 6.2). The three options are “Play Time Attack or freely” (“Jugar a contrarellotge o lliure”), “Play versus a friend” (Jugar amb un amic”) and “Play versus the computer” (“Jugar contra l’ordinador”)

Once any of the three buttons is pressed, the next scenario is shown.
In here, the user can choose the number of couples to play with, from two couples to ten. The number of couples chosen indicates how many cards will be on the board, so two couples will put four cards on the board and ten couples will put twenty cards on the board.

A button to go back to the previous scenario for choosing the game mode is available (the button “Torna al lloc anterior”), and the game mode previously chosen is indicated on the top center part of the interface (See Figures 6.3 and 6.4).

With the game mode and the number of couples chosen, the user gets into the specific playing scenario.
6.2 General in-game features

For all the in-game scenarios, the “Torna al lloc anterior” button is available for the user to go back to the previous scenario. Also a “replay” button is available that when pressed, the timer is set to zero, and new cards are set on the board. The game mode that is currently played is indicated during all the game in the label on the top central part of the layout.

The cards behavior is always the same. The user can press and see two consecutive cards. If they form a couple, both cards keep shown but disabled. In case they are different, both cards are turned around when another card is pressed. When playing with two players, getting a couple right is rewarded with a point and a chance to press two more cards.

6.3 Playing Time Attack or freely

For this example, the scenario in Figure 6.5 is a game with three couples.

![Figure 6.5: Time Attack with three couples.](image)

In this game mode, the user can freely play without no pressure elements as competition or time. To play freely, no requirement is needed, the user can start pressing cards and finding the couples. Once all the couples are found or the user is ready to change, can press the “Juga un altre cop” button to play again or the “Torna al lloc anterior” button to change game characteristics.
In the example in Figure 6.6, to play in Time Attack mode, a button “Start the time” (“Comença el temps”) is available. Pressing this button starts a time counter and the user must start trying to discover all the couples. Once discovered, the user must press the “Fi!” button and a notification will show up in the middle of the screen showing the time that has passed (see Figure 6.7).

To be able to press any button, the notification must be pressed to hide it.
6.4 Playing versus a friend

For this example, the scenario shown in Figure 6.8 is a game with six pairs.

In the “Versus a friend” game mode, there is a label and a score for two players (“Jugador 1” and “Jugador 2”). When playing, the label indicates whose turn is by showing the text “ET TOCA” next to the player (See Figure 6.9)
When all the couples are found, the player with the biggest score is the winner and a notification appears to declare the winner (Figure 6.10)

![Figure 6.10: Player 1 wins.](image)

In case the same score is achieved by both players, the notification declares a draw (Figure 6.11)

![Figure 6.11: Draw.](image)

To be able to press any button, the notification must be pressed to hide it.
6.5 Playing versus the computer

For this example, the scenario in Figure 6.12 is a game with ten couples.

![Figure 6.12: Versus the computer with ten couples.](image)

When playing against the computer, a difficulty box “Escull nivell de dificultat (5 el més difícil)” is available, where the user must choose the difficulty of the game, that will be translated to how well does the computer play. This box is disabled when the user starts playing (Figure 6.13)

![Figure 6.13: Versus computer middle game.](image)
To play in this mode, the user must press two cards. When these are pressed and they’re not a couple, all the cards become disabled and a button “Següent ronda” is enabled. This prevents the user to press illegal elements and guides the user (See Figure 6.14)

Figure 6.14: Versus computer. End of user’s turn.

Pushing the “Següent ronda” button allows the computer to play his turn. The high difficulty makes the computer play as a human that remembers all the cards that sees and the computer plays with the same rules as a user, so it is normal that the computer sometimes discovers several couples in a row while playing in high difficulty, making the challenge even harder. In Figure 6.15 a computer victory is shown.

Figure 6.15: Versus computer end.
7. FINAL RESULT

The final result of the project is a computer adaptation of the game “Cada ovella amb la seva parella” that will be easily accessible via the internet with any web browser.

The game has an easy interface with intuitive usage accomplishing the requirements that imply the age of the potential users, and thanks to its design and competitive and flexible features it is a very replayable game.

Putting together these characteristics with the science learning environment of the original game, the result is a tool disguised of fun game that will have children addicted to learning science.
8 CONCLUSIONS

The planification of the process, the steps followed, the research done, the design and implementation of the code and all combined with a large number of hours invested and the guidance of professionals makes this project a major learning experience in university.

“Jugant a definir la ciència online: Cada ovella amb la seva parella” has been an excellent challenge to develop my project organization and realization skills, and a perfect chance to practice the programming knowledge learned during university.

Being part of an already developed project as “Jugant a definir la ciència” instead of creating a new one, and understanding its objectives and motivations has been a very important input to define the character of the project. “Jugant a definir la ciència online: Cada ovella amb la seva parella” has become the tool that was expected.

The result of the project is an attractive teaching science game that accomplishes the goals of teaching science thanks to the attractively entertaining game characteristics. This proofs that a well planification, structuration and implementation of the resources has been done.

Analyzing the result obtained in the project, I can say that Eclipse and Vaadin have resulted a useful and indicated tool to program the game, especially thanks to the specific features focused on the interface implementation.

This application is important to promote the learning of science knowledge and the exercising of important body functions like the memory or agility, especially for children.

As a final conclusion, I successfully analyzed, designed, developed and evaluated an interactive application with web access that hopefully will accomplish the goal “learning while playing” for children, mostly with Catalan as mother language.
9 FUTURE WORK

The adaptation made in this project it is just one of the proposals for the “Cada ovella amb la seva parella” game. The scalability of the game makes possible the extension of the game by adding other functionalities to it, like other game modes with more cards.

Some possible next features would be including the possibility of recording the score of the users in a data base and establish a score ranking, introducing a user profile structure for the users to log in, adding a feature to play versus a friend online, adding sound tracks related to the cards to hear the pronunciation of the words, add specific science area versions and implement versions of the game in other languages like Spanish, English, French, etc. The game could also be adapted to other platforms like tablets and smartphones, and specific adaptations for neurofunctional disabled people is a future work as well.

“Jugant a definir la ciència online: Cada Ovella amb la seva parella” is just one of the games proposed by the project “Jugant a definir la ciència”. In the beginning of this document other titles of “Jugant a definir la ciència” are mentioned, and those could also be adapted to the informatics world, like the dictionaries “My first Dictionary of Science” (Estopà, 2013)[1] or the other three games available in “La Maleta Viatgera”.

37
Bibliography


http://docs.oracle.com/javase/specs/ [ May 2014]

http://docs.oracle.com/javase/7/docs/api/java/awt/Button.html [February 2014]

http://es.wikipedia.org/wiki/Eclipse_(software)#Caracter%C3%ADsticas[May 2014]

http://medialab.di.unipi.it/web/IUM/Programmazione/OO/what/concepts.html [May 2014]

http://www.eclipse.org/ [June 2014]

https://vaadin.com/home [February 2014]

http://defciencia.iula.upf.edu/ [April 2014]

http://www.iula.upf.edu/breus/breu289ca.htm [April 2014]

http://stackoverflow.com/ [March 2014]

https://vaadin.com/forum# [March 2014]


[2] Estopà, Rosa “Jugant a definir la ciència: un diccionari de mots de ciència fet per i per a nens i nenes” (2013)


Annex

Background image

Card images
laboratori
lupa
mapa
microscopi
nervi
neurona
ordinador
u
relotge

telescopi

termòmetre (temperatura del cos)

volcà
planta (vegetal)
juga amb Paraules de Ciència