

Categoría: Trabajos de Cátedra

Título del Trabajo: Mavaru-Online

Autores: estudiantes de Ingeniería en Informática de la  
Universidad Austral

Santiago Bestani: [santor1210@gmail.com](mailto:santor1210@gmail.com)

Juan Ignacio Molteni: [juan.ignacio.molteni@gmail.com](mailto:juan.ignacio.molteni@gmail.com)

María Julia Raca: [juliracca@gmail.com](mailto:juliracca@gmail.com)

Santiago Raca: [santiracca3@gmail.com](mailto:santiracca3@gmail.com)

Cátedra: Laboratorio III

Dra. Gabriela Robiolo: [grobiolo@austral.edu.ar](mailto:grobiolo@austral.edu.ar)

Ing. Daniel Grane: [daniel.grane@gmail.com](mailto:daniel.grane@gmail.com)

## Mavaru-Online

**Abstract:** The online social networks open a new window to human interaction. However, the poor dynamic interaction offered by them motivated us to think about a new solution. We believe that creating a graphic world where users can interact with their friends in a funner more ergonomic way. Mavaru-Online, is a network where he or she will not only be able to make friends and communicate, but also enjoy an experience with a brand new level of interaction. This will include actions like leaving messages, chatting between friends, editing your avatar to look more like you, creating new friendships as well as maintain old ones in a new, more dynamic, visual, and richer experience.

Keywords: visual social network, extreme programming, web application.

### 1 Introduction

Internet growth has modified almost all dimensions of social relations. Human beings, middle and end of social relations, have adapted to the social networks socially embedded behaviors and patterns

The Internet adds new dynamics to social exchange as the notions of space and time seem meaningless in the virtual world [1].

But these networks offer poor dynamic interaction which affects every user, impacting the limit of its usage because of its staginess. This motivated us to think about a successful solution that would be creating a graphic world where users can interact with their friends in a new way, as in a third person 3D game.

So, the main idea of our product is taking the everyday experience on a social network and make it visual. The user will be able to connect to de Mavaru-Online network, where he or she will not only be able to make friends and communicate, but also enjoy this experience with a brand new level of interaction. This will include actions like leaving messages, chatting between friends, editing your avatar to look more like you, and many more. If this wasn't enough already the user may also explore the world finding new places and interacting with your friends in them. The Mavaru-Online world will also offer the opportunity to face off with your friends in different minigames and 3D activities as well as with other people you might not know. This type of interaction will create new friendships as well as maintain old ones in a new, more dynamic, visual, and richer experience.

Maravuv-Online is visual and 3D. It is oriented to everyone with Internet, who uses social networks daily, which offers a dynamic way of interacting, unlike Facebook, which facilitated static interaction.

There are two possible applications that could compete Mavaru-Online-Online. Imvu [2] social network made in Brasil, aiming older people; is mostly focused on dating. The second option is Club Penguin [3], which aims at a younger audience building an environment where kids can play games, earn points, with which they will buy things. We think that Mavaru-Online-Online can attract the audience that is in between these two ages because it aims to integrate itself with Facebook and other popular social networks and show them in a way that has never been seen before.

Our product is described in the following sections beginning in section one with the context, section two the product description where the structured, implementation environment and the external design are reported. Finally a brief of the lessons learned.

## **2 Context**

Our product was developed in the context of academic subject of the 4th year of Informatics Engineering at Austral University in Buenos Aires, which is called Third Laboratory. During subject we developed a product beginning with an idea and ending with a prototype.

The students formed groups with three or four members. Each group was balanced in certain aspects such as work experience and finals owed, in such a way that they were comparable between each other. By doing so, the difference between projects could be reduced to the different methods of programing employed and the way they were carried out.

Throughout the year we worked with a method called Extreme Programming (XP), that is a software development methodology which is intended to improve software quality and sensitivity to changing customer requirements [4,5]. As a type of agile software development, it advocates frequent "releases" in short development cycles, which is proposed to improve productivity and introduce checkpoints where new customer requirements can be adopted. Also, it considered avoiding programming of features until they are actually needed, expecting changes in the customer's requirements as time passes and the problem is better understood, simplicity and clarity in code, frequent communication with the customer and among programmers, programming in pairs, doing extensive code review, unit testing and a flat management structure.

Critics have noted several potential drawbacks, including problems with unstable requirements, no documented compromises of user conflicts, and a lack of an overall design specification or document [6].

Each sprint in the agile methodology implemented a set of user stories, which are assigned a number of hours to be worked on. One of the key factors of Extreme programming is the quality of the code obtained doing pair programming and the amount

of time that we save on debugging such code. The time which is saved can be attributed to the benefit of having two people working on the code at the same time, if one of them cannot debug a certain piece of code the other one might, because he or she will probably see the problem from a different angle.

## 2.1 Project execution

The participants played the different roles such as:

- a. Students were Programmers, which had the following responsibilities: 1) create the objects needed in the program 2) make them interact with each other 3) add functionality 4) maintain the data base 5) create and maintain the product server and 6) making sure the program is free of bugs. Also, they were Animator, which was in charge of the animation for the objects used in the application, adapting the design models and verifying that they were animated in a correct and fluent way.
- b. Professors were Clients, Tracker, Coach and Big Boss, who guided thought the development of the product, offering feedback, controlling that it is delivered on time, introducing new requirements.

Figure 1 shows the workloads and deadlines for sprints and Figure 2 the estimated and actual hours of each user story that were implemented in a particular sprint.

Sprint N	1	2	3	4
Duration	6 weeks	4 weeks	4 weeks	5 weeks
Total Hours	43 hours	12 hours	21 hours	33 hours
Start	May 4, 2011	Aug 10, 2011	Sep 14, 2011	Oct 19, 2011
Finish	Jun 22, 2011	Aug 31, 2011	Oct 12, 2011	Nov 16, 2011
Presentation	Jun 22, 2011	Sep 7, 2011	Oct 12, 2011	Nov 16, 2011

Fig. 1. sprints workloads and deadlines

User Story	Estimated Hours	Real Hours
<b>Sprint 1</b>	<b>43</b>	<b>47</b>
Create Character	10	25
Delete My Character	0	2
Enter World	9	6
Walk	11	3
Have a House	13	11
<b>Sprint 2</b>	<b>12</b>	<b>15</b>
Register & Login	7	7
Customize My Character	5	8
<b>Sprint 3</b>	<b>21</b>	<b>24</b>
Play in the Web	5	5
Interact with World	4	3
Messaging	12	16
<b>Sprint 4</b>	<b>33</b>	<b>31</b>
See Other Characters	15	13
Chat with Your Friends	11	9
Go to Friend's House	4	6
Have a Common Place to Hang out	3	3

Fig. 2. User story estimated and actual hours of each

### 3 Product description

Basically, users will have the possibility to interact with others as a common social network, but with the added potential of being able to view the whole experience. Some of these additional functionalities would be the ability to walk around a 3D world and talk to friends, comment on their virtual wall, visit their home and play an set of games.

#### 3.1 Architecture

Mavaru-Online Online structure consists of classes that correspond to the Graphic User Interface (View), others to Entities persisted in the database and their attributes (Model), and the Service classes that are responsible of connecting those explained before (Controller) which are shown in Figure 3.

In the View section (GUI) we can find classes such as Chat, FriendList, Message-BoardUI, which are in charge mainly of showing graphical objects to the user as the game goes on. Each of the View classes will contain an OnGUI function which will be responsible of redrawing each object every time the game is updated so that it is shown correctly on the screen.

On the other hand, in the Model area, we will find classes referred to Entities that may be persisted in the game database, such as User and Message. But since we used Unity3D for developing this game, then, we will also need objects to match these classes in the game, as Character and MainCharacter. Considering that Mavaru-Online is a

massive multiplayer game, we will also need classes to distinguish between my player(MainCharacter) and the rest drawn on my screen(Character), which belong to people playing somewhere else.

Finally, in the Controller side, we will find classes that handle the connection of new players, such as NetworkManager and CharacterNetworkManager; others that manage database transactions, as DatabaseManager; and those ones which deal with the characters interactions, CharacterHolder for instance.

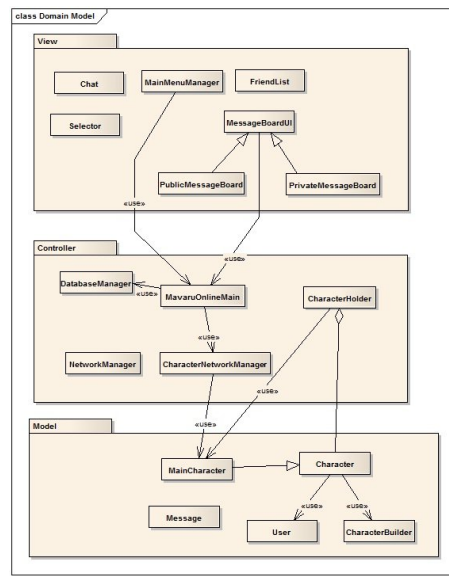


Fig. 3. Class Diagram

### 3.2 Implementation Environment

The developing environment was a hard task, in which we need to take advantage of several advanced tools. Therefore we used Maya as well as Google Sketchup for characters and world's graphics design, and Unity 3D to develop interaction between objects. Finally, we used Monodevelop for the object's script implementation. All of this was wrapped around a MySQL database that can connect with our C# code in Monodevelop via NHibernate.

The tools used were:

- a. Unity 3D [7], is an integrated authoring tool for creating 3D video games or other interactive content such as architectural visualizations or real-time 3D animations. Unity's development environment runs on Microsoft Windows and Mac OS

X, and the games it produces can be run on Windows, Mac, Xbox 360, PlayStation 3, Wii,iPad, iPhone, as well as the Android platform. It can also produce browser games that use the Unity web playerplugin, supported on Mac and Windows but not Linux. The web player is also used for deployment as Mac widgets.

Unity consists of both an editor for developing/designing content and a game engine for executing the final product. Unity is similar to Director, Blender game engine, Virtools,Torque Game Builder, and Gamestudio, which also use an integrated graphical environment as the primary method of development.

b. MonoDevelop [8], is an open source integrated development environment for the Linuxplatform, Mac OS X, and Microsoft Windows, primarily targeted for the development of software that uses both the Mono and Microsoft .NET frameworks. MonoDevelop integrates features similar to those of NetBeans and Microsoft Visual Studio, such as automatic code completion, source control, a graphical user interface (GUI) and Web designer. MonoDevelop integrates a Gtk# GUI designer called Stetic. It currently has language support for C#, Java, Boo, Visual Basic.NET, Oxygene, CIL, Python, Vala, C and C++.

c. NHibernate [9], is an object-relational mapping (ORM) solution for the Microsoft .NET platform: it provides a framework for mapping an object-oriented domain model to a traditional relational database. Its purpose is to relieve the developer from a significant portion of relational datapersistence-related programming tasks. NHibernate is free as open source software that is distributed under the GNU Lesser General Public License. NHibernate is a port of the popular Java O/R mapper Hibernate to .NET. NHibernate provides transparent persistence forPlain Old CLR Objects (POCOs).

d. MySQL [10], is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. It is named after developer Michael Widenius' daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla,WordPress, MyBB, phpBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google (though not for searches), Facebook, and Twitter.

### 3.3 External Design

The external design simulates a more realistic interaction between the persons as the avatars take the place of the people. Figure 4 shows the Login screen, Figure 5 shows the screen which let to Customize the user's character, Figure 6 shows Go home or Go to Friends Home.

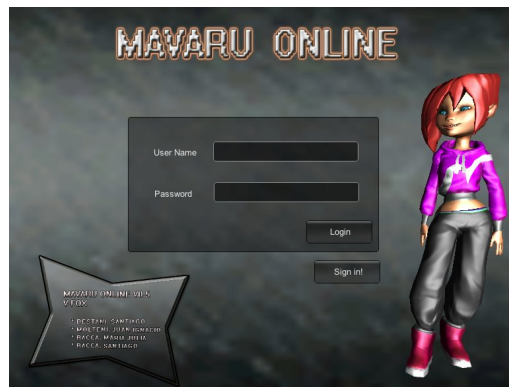


Fig. 4. shows the Login screen



Fig. 5. Customize you character





**Fig. 6.** Home

Figure 7 shows Go to public place, Chat facilities and also a social event that is See friends. Figure 8 shows the billboard that have the options of Write on walls, Read walls and Write private message.



**Fig. 7.** Go to public place, Chat and See friends



Fig. 8. Write on walls, Read walls and Write private message

#### 4 Lessons Learnt

Having this subject at the University made us work a lot in groups, helping us to get to know each other better, as well as encourage us to help others with their weaknesses as they helped us with ours. We have got to improve ourselves, as we were forced to develop more understandable and cleaner code, so others could understand it as quick as possible. That way, the project would evolve faster and in a more efficient way. The only way to get to this was by hard work and opened mindedness to changes and other points of view. This might be one of the most important things learned throughout the whole process.

Whilst working on this project we found in pair programming a great method to complement our work as a team. Pair programming consists of putting together groups of two for each user story, and working side by side on such story. This system facilitated parallel thinking which, by consequence brought faster programming, and reduced errors. It also allowed us to see each user story from two different perspectives that could be meshed together to create a better outcome, and also help with the debugging process. This method of teamwork aided us in joining our knowledge, and in building a final product that is far more superior to one without this group effort. Many people would ask themselves why it is that working two on each computer can increment productivity. If we think it straight, if each programmer worked in their own computer twice the amount of code could be generated. But this is not as real as it sounds. While working in pairs, one generally gets involved completely with the coding and programming, but the other gets a complete perspective of the issue to solve. Many times when we are programming, we immerse ourselves on what we are doing and forget about the problem itself, so we try and keep trying to solve some-

thing that maybe we cannot find the answer to because we do not have a complete vision of the program or module we are working on. So we lose time wondering how to handle each issue when someone, with a new perspective, could solve it almost immediately. That happens every time we work, that's why group programming is far better than doing it on your own, because you always need different points of view. This is what pair programming enhances the most, you get to make a better, more reusable, cleaner, and much more understandable code to others unrelated to it. Regarding technical knowledge, we enriched ourselves a lot with new technologies, such as Unity3D, MonoDevelop, NHibernate. These are really powerful tools for developing games that you might never learn to use, if you do not get involved with them and struggle to make a game as best as you can. We really wanted to highlight this. What we have made with Mavaru-Online will surely open a lot of doors in game programming. This version of Mavaru-Online was just the first step and we are really proud of it. There is a lot of work to do on it, and we are willing to improve ourselves more and more. The hardest part has finished, now is just a matter of time for its exponential growth.

## 5 References

1. Schiavone, E. A. (2011) Redes sociales on line. Trabajo de Grado, Universidad Austral
2. Imvu, <http://www.imvu.com>
3. Club Penguin, <http://www.clubpenguin.com>
4. Extreme programming <http://www.extremeprogramming.org>
5. Beck, K and Andres, C. (2005) Extreme Programming Explained: Embrace Change (2nd Edition), Pearson Education.
6. Coram, M and Bohner, S (2005) The Impact of Agile Methods on Software Project Management. Department of Computer Science, Virginia Polytechnical Institute and State University, Blacksburg, Virginia.
7. Unity 3D, <http://www.unity3d.com>
8. MonoDevelop, <http://www.monodevelop.com>
9. NHibernate, <http://www.nhforge.org>
10. MySQL, <http://www.mysql.com>