Académie de Montpellier

Université Montpellier 2

— Sciences et Techniques du Languedoc —

M.Sc. Informatics

Travail d'Etude et de Recherche

Amazonia Serious Game

Guillaume Dalichoux Alexandre Foucher
Fabien Hervouet Panupat Patramool

Advisor	:	Stefano A. Cerri
Co-advisor	:	Tiago Garcia de Senna Carneiro
Co-advisor	:	Ines Di Loreto
Referee	:	Mathieu Lafourcade
Listener	:	Michel Leclère

Abstract

This project was carried out by students during their first year of Master Informatics. The original idea was to create an educational game – also called "serious game" – with the scope of making people aware of what is happening right now in *Amazonia*. The originated game meant to be a MMORPG, literally Massively Multiplayer Online Role-Playing Game, which allows players to play a character who evolves into a persistent virtual world within the framework of resources, actors and actions relating to *Amazonia*. In this report, we provide a complete analysis, both from a conceptual and a formal point of view – full game design and informatical modeling schemas – but we also abundantly explain how we thought to implement technical specifications. The developed game is based on the client / server model, respectively programmed in *Flex* and *Python*.
Since a complete game is not something that can be realized in three months, we decided to focus our work on the specifications and simply implement a very first prototype. In this way, further implementations that will have only to follow the existing specifications.

Ce projet a été développé par des étudiants durant leur première année de Master Informatique. Le principe était de créer un jeu destiné à l'éducation – que l'on appelle également "jeu sérieux" – avec l'idée d'essayer d'informer les gens sur ce qui se passe actuellement en *Amazonie*. Ce jeu se veut être un MMORPG – jeu massivement multijoueur en ligne – qui permet aux joueurs d'endosser le rôle d'un personnage qui évolue dans un monde virtuel persistant ayant pour cadre les ressources, les acteurs et les actions relatifs à l'*Amazonie*. Nous fournissons dans ce rapport une analyse complète, à la fois conceptuelle et formelle – le "game design" complet et les schémas de modélisation – mais nous expliquons également en détail la façon dont nous avons pensé l'implémentation des spécifications techniques. Ce logiciel est basé sur le modèle client / serveur, qui sont respectivement programmés en *Flex* et *Python*.
Etant donné que faire un jeu complet n'est pas une chose réalisable en trois mois, nous avons décidé d'axer notre travail sur les spécifications et d'implémenter un tout premier prototype. L'implémentation future n'aura qu'à suivre nos spécifications.

Keywords : Amazonia, serious game, MMORPG, Flex, Python

Acknowledgements

Acknowledgements is a very particular section, maybe the most difficult to write, because we want to thank every person who dedicated more or less time to this project, trying not to forget anyone. Let's try.

So we firstly thank very much both proponents of this very exciting project that was so rewarding, both for the collaboration that has been installed between all participants, and the knowledge that it gave us at all levels : **Pr. Stefano A. Cerri** as our advisor at the *University of Montpellier 2*, and **Dr. Tiago Garcia de Senna Carneiro** as our advisor at a distance at *Universidade Federal de Ouro Preto*.

How could we sufficiently thank the chance that led **Ines Di Loreto** to come for a few months in *Montpellier* to finish writing her thesis ? Especially because besides her professionalism, her game design skills, her inexhaustible enthusiasm, her good advice, her encouragement, her talent for drawing and her charming Italian accent, she made us find out more Italy, its specialties and wines ! What more could we ask for ? Thank you a lot **Ines**.

We are not used to deal with social scientist in our curricula. Their presence has contributed to the fact that this project has been very interesting. That's why we absolutely want to say that we are very grateful that **Roberto Araújo de Oliveira Santos Júnior**, a Brazilian anthropologist from *Belem*, spent time during the weekend to explain us what is currently happening in Amazonia.
We also warmly thank **Viviane Boyer-Araújo**, who made the full size drawings of the characters and the introductive storyboard, despite her courses and other projects under way in her cursus.

There is another man who helped us by setting up the basic project communication and collaboration tools : **Pascal Dugénie**. His kindness and availability affected us a lot, and enabled us to use powerful tools.

And of course, we thank a lot every people who helped us to a greater or lesser scale and that we would have had the awkwardness of forgetting to mention.

Contents

Chapter 1

Introduction

In this section, we will introduce you to the general framework of this project, the origins, and why we wanted to do it. We will then explain more precisely the subject, and we will see the requirements document we wrote during the first month of work on this project.

1.1 General issue

Within the framework of our project of first year of Master degree in Informatics – which consists in 10 european credits (ECTS) – we were proposed to take part in the design process of a global project about **Amazonia**.
We are four students from three different routes in the vast domain of Informatics : *Conception and modeling*, *Web oriented development* and *Artificial Intelligence*. This has been a real force for the team work, because everybody has different knowledge, some of us were more accurate in certain domains, and moreover this provided new points of view.
It is interesting to note that rather than apply on existing topics proposed by teachers, we went ourselves to see Pr. *Stefano A. Cerri* whether he had a possible topic in mind. The origins of the project are explained in the following 1.2 section.

1.2 The subject

1.2.1 Current situation in Amazonia

The fact that the subject of the project relates to an ecological concern has contributed heavily to our infatuation. While the Amazon's deforestation problem is a complex one, it is worth to spend some lines describing it for better understanding the issues of the largest and most species-rich tract of tropical rainforest in the world.

Figure 1.1: Localization of Amazonia ecoregions as delineated by the WWF

History

Prior to the early 1960s, access to the forest's interior was highly restricted, and the forest remained basically intact. Actually, the key point in deforestation of Amazonia was when colonists established farms in the forest during the 1960s. Their farming system was based on crop cultivation and the slash and burn method. Farmers are therefore constantly moving to new areas and clearing more and more land. Amazonian colonization was ruled by cattle raising – because ranching required little labor while grass can grow in the poor soil – generated decent profits, and awarded social status in the community.
However, the results of the farming lead to extensive deforestation and caused extensive environmental damage. Some experts estimate that 30% of the deforestation is due to small farmers. This stresses the importance of distributing previously cleared land for agricultural purpose, rather than the typical easiest political and less expensive research way to grant still-forested areas to agricultors.

Actors in the deforestation

Understanding the forces driving deforestation is a top priority because any changes in policies designed to contain deforestation are likely to fail if they are not based on realistic scenarios of how the social processes involved operate today and how they may change in the future.
A detailed actor-based approach is needed if the social processes underlying deforestation are to be understood and eventually modified to contain clearing. Types of actors cover a full range in terms of wealth, legality, and the intensive or extensive nature of their activities. [2]

1.2.2 Origin of the project

In November 2008, a conference was held in Manaus, where the key actors committed to the management of scientific knowledge by means of ICT met. Among them, Pr. *Stefano A. Cerri* & Dr. *Tiago Garcia de Senna Carneiro*[1] were invited as speakers. The main topic issue was the sustainable development. The talks of these two informaticians gave them the idea and the desire to collaborate.
Returned to France, *Stefano A. Cerri* has reworked the idea of collaboration. The trigger was the fact that we, students, came to ask him a subject. He proposed us four subjects linked with Brazil and technologies he likes very much – such as *Agora GSD*[2] – but the project which excited us, was the one dealing with serious game, with its ecological aspects and the educational part.

1.2.3 Main objectives

The aims of this project were numerous, but the first one was **collaboration at a distance**, since the project has been advised by Pr. *Stefano A. Cerri* and a PhD student *Ines Di Loreto* at *LIRMM* in *Montpellier*, but also by Dr. *Tiago Garcia de Senna Carneiro*, researcher in *Ouro Preto* (Brazil). This collaboration contributed to design a serious game, which could be seen as an educational game to sensitize people to what is happening everyday in Amazonia. Moreover, the game had to simulate the **behaviors** of the main actors responsible for **deforestation** in Amazonia, while letting players evolve with their liking.

1.3 The first requirements document

1.3.1 Objectives, needs and constraints

We considered *Tiago Garcia de Senna Carneiro* as our "client" for the needs and the constraints of this project. That is the reason why we really needed a first video conference with him before we start to work, to be sure to write the right requirements document.
With his collaboration, we understood that the mission will deal with understanding by simulation what people think about Amazonia occupation, having targets for the potential users at three different levels : local (Amazonian people), national (Brazilian people) and international (the whole world).
For our project we have identified three major objectives :

- As we said, collaboration is very important within the participatory design process, especially if the customer works in Brazil while we are in France. To communicate with our "client" and between us, we decided to use

[1] Computer Science Department at the Federal University of Ouro Preto (UFOP) – Brazil; Head of TerraLAB – Laboratory for Modeling and Simulation of Earth Systems, National Institute of Space Research (*INPE*). Research (INPE)

[2] Experimental platform for the deployment of collaborative spaces. Agora's originality lies in its architecture to facilitate access to resources distributed over several sites.

asynchronous and synchronous communication tools which are explained in the A.2 section. The collaboration process will take an important place in the gameplay too. Indeed, if players want to communicate in order to cooperate, or haggle over their lands and/or services, etc... (2.2.5).

- The second objective was to make a list of the main actors who influence the deforestation process in Amazonia. In order to define them, *Tiago* gave us the contact of an anthropologist of his, Roberto Araújo de Oliveira Santos Júnior[3]. The defined actors have different behaviors, and their own way to approach the resources of the forest, respecting it or not, being honest or not... But any action carried out will have a positive or negative consequence on the evolution of the game. So we decided to make a list of rules with "ACTORS / ACTIONS / CONSEQUENCES". For instance :
  - The government decides to build a new road, which implies the destruction of a few trees.
  - A farmer sees a new road and decides to install a new farm, which implies the destruction of additional trees.
  - A drug trafficker finds a farm already built accessible with a perfect road, he decides to kill the farmer and take the farm.
- The third objective was to find a way to make those behaviors designable, and to integrate them in a game. This game would allow players to incarnate a kind of character with its own rules, but likely to evolve according to his goodwill.

1.3.2 Possible solution(s)

The solution considered consisted in designing three linked parts about the game design.

- Modeling characters' behavior : the player can choose a character among a list. Every PC (Player Character) should have specific soft and hard skills. The player will obviously be able to make evolve a character apart from his usual behaviors.
  Modeling NPC behavior could be great. To add interesting interactions we could insert NPCs (Non Player Characters) who would evolve in the game and randomly trigger events (we need to implement rules of events for each NPC). This would lengthen the life time of the gameplay because every game session would be different, depending on the events triggered.
- Modeling land changes : studies made by biologists who work with *Tiago* would be very useful. This part would use a database that we could design (using a system possibly based on rules of evolution) and we would have to design/program agents for the modeling of the environment evolutions.

[3]Anthropologist, professor and researcher from the Emilio Goeldi Museum and the Federal University of Pará (Belém, Brazil)

- Modeling relationships : the modeling of socialization will be at two levels. First, players could decide to collaborate in the game for making more money which could be very interesting from the gameplay point of view. Secondly, players could use communication tools during the game (synchronous and/or asynchronous). Moreover, players can influence the way other players evolve.

For the technical part, we more or less decided that the graphical part of the game will be in 2D, or probably in 3D isometric. *Flash* was evoked as a possible development platform, almost for the ease of distribution (internet) and compatibility (the only need is a web browser). It has been necessary for us to get more information about this technology to weigh the pros and cons. For this reason we decided to find a game engine based on it in order not to reinvent the wheel.

1.3.3 The spiral model

The spiral model is a specific software development process. It combines elements of both design and prototyping, in an effort to combine advantages of top-down and bottom-up[4] concepts. This model of development is intended for large, expensive and complicated projects (and this project is large and complicated...).
It relies on an incremental model, based on a series of iterations, and it is constructed to best manage the complete development of the software. Iterations are typically six months to two years long. Each phase starts with a design goal and ends with – in the commercial sense – the client reviewing the progress thus far.
The different steps in this model are generally :

- Requirements are defined in as much detail as possible, interviewing future end users.
- First draft design is created.
- First prototype is constructed from the preliminary design.
- Second prototype is evolved considering a few points :
  - Evaluation of the strengths and weaknesses of the first prototype
  - Requirements of the second prototype;
  - Designing the second prototype;
  - Second prototype is constructed and tested

Moreover, the spiral model guaranties the quality through prototyping at each stage in systems development.

[4]Strategies of information processing and knowledge ordering, mostly involving software, but also other humanistic and scientific theories

For all these, from the start we thought that the development of this game will perfectly fit the spiral model. In fact, such a project will take a long time in order to be finished, and will probably picked up by other developers than us, its parents.

Chapter 2

Analysis

The analysis is the key point of any project. In this project which is a game, there was a lot of research work and deepening on the part of the game design, which is an extremely important part to us. In this section are firstly presented the specifications of the project, then the way we wanted to make the user's side of the game, and finally how we have modeled these different aspects in a more formal informatical way.

2.1 Identification of specifications

This part is one of the most important of the project, and it took us a long time. First, we had to understand what is happening in Amazonia. Processes involved in the deforestation are multiple, almost as much as the number of people who apply them.
We documented a lot about Brazil, especially Amazonia. We have been dealing with an ethnologist / anthropologist[1], who explained us in detail the various actors working in these regions, and also the roles they play. We tried to understand the life chances of these people.
Then according to their means, they can evolve into various "professions", from the simple picking to survive – many of them do not exceed this stage – to the mechanized agriculture. We had to pick some of the main actors, because they are very numerous, and we chose five of them, which are described in 2.2.3.

2.2 Game design

2.2.1 Framework of the game

This game takes place in the Amazon context. The different processes and people involved in the deforestation are the starting point of this game. The

[1] Roberto Araújo de Oliveira Santos Júnior who is one of Tiago GSC's colleague.

player will be shown immediately in a tragic context. To do this, we developed a short series of drawings that explain how some people (currently our main character) are brought in Amazonia to work as slaves for a pittance – when they are paid.
In a more gaming formal way, this game is a kind of MMORPG[2], it means that multiple players will have the same main goal in the same virtual persistent world : making quests, winning experience points, and try to evolve to a more comfortable role. That is the essence of any RPG.
But there is a great innovation compared to traditional RPG, because in our game the player may at some time, change the style of play and start playing like in a strategy game. This possibility is given when the player reaches the highest role of the game : *Fazendeiro*. We naturally believe that this perspective can greatly increase the duration of a game, because the one who does not wish to change will just have to remain at a lower level, but those who would like to change, will be able to use this new mode and continue to play in a very different way, enabling them to see the world in which they has evolved in a different light.

2.2.2 Gameplay

Different factors

The player will have to deal with various factors that constitute his primary concern throughout the game :

Money It is a factor which is very common in the real life, and as we want to describe the real life of actors evolving in Amazonia, this factor has its place among others. Money will be required to buy something, and will be earned by performing quests, stealing, or offering services to other players.

Experience points They are needed in order to evolve. The player will win these experience points by performing quests, but lose in deforestation – among others. Experience is a common feature of almost every RPG that exist.

Health Another common factor in this kind of game. This is a very strong feature, because it shows when the player is tired, sick – the health rate will decrease according to this. However, the player may treat himself to recover the lost health and even increase its capacity for general health in cases where certain quests succeed.

[2]Game in which a large number of players interact with one another within a virtual game world.

Mood This feature is less prevalent in games, but it exists in *The Sims*[3]. This is an idea to show how a good mood is important to perform action faster and with more accuracy. This factor decreases, for instance, with lands deforestation around player's home, or if he brings little to eat to his family, or according to external events caused by *chance cards* – explained in 2.2.4.

Socialization rate This is also a new kind of feature which is not used in classical RPG. It is linked with our wish to enhance interactions between players. This feature introduces the fact that if players don't interact enough with others, this rate will decrease and will ultimately penalize him. This, with the idea of social quest, was created to avoid people playing just to evolve alone in their own world.

Interactions

The general framework for interactions in the game will be the map. As the player will evolve in an environment with many objects around him, he will be able to make a lot of interactions. But the main principle is that the player will have to deal with the paradigm of being. It means that objects will have a very important place in the game. In a graphical way, it means that the player will firstly have to click on the "object" – this includes natural products of the environment, property of the player, and even other players (2.2.5) – he wants to interact with, and then he will see a contextual menu appear.
Moreover, there will be four main tools accessible whenever the player wants to. These tools represents the main interactions with the environment :

Point This is what we have just explained, a pointer which allows to "point" an "object" of the environment to interact with it merely.

Move The player will naturally have to move in the world, and this is thanks to this feature. It allows you to click a point on the map, and the character will go there by calculating the shortest way (3.2.3).

Browse If the player just wants to browse the map in order to be aware of where he is or where he can find a specific resource, he will be able to do it by using this tool.

Deforest & Reforest The fourth feature is dual, but is the one that points out the more the main subject of this project : ecological awareness. The fact that these two actions are put at the same ergonomical level than the three others, it means a lot to us. We wanted to show to players the importance of these actions, and putting them very accessible denotes precisely the way how

[3] A strategic life-simulation computer game developed by *Maxis* and published by *Electronic Arts*. It is a simulation of the daily activities of one or more virtual persons in a suburban household.

current actors in Amazonia think : deforest and reforestation are two actions that people there use very often. Thus the paradox between the fact that in Europe, the school educates us on the problems of deforestation, and to give free access to a very simple tool to do it in a game is very interesting to a point of psycho-sociological view in order to see how ecological sensitized people use them.

Constructions

As we said previously, the player will be able to interact with already existing "objects" in the world. But he will also be able to construct some of them, in order to improve his life quality – a better built and more comfortable house – or tools to help him in its agricultor role – cultivating fields, raising fields, etc... – and the like.
These constructions will require, as almost all actions, a certain amount of experience points, money and time to perform it.

2.2.3 Characters in the game

We decided to create only one beginning character : the *Collector*. Player's choices will totally influence the evolution of his career. There may be multiple paths to access another higher job, but each path has advantages and drawbacks. Moreover, each evolution needs some own goods for the player, and a certain amount of money. Moreover, any career development can not happen unless the player decides for it by himself, although changes will of course be proposed, as the program of quests.
At first, we thought that it would be better to allow players to become an illegal man in itself. But we found out that there was not much interest in a role dedicated to that. We preferred to establish a system of status. The player will have the opportunity to make illegal actions, but they will directly – in terms of their seriousness – increase their own "crime rate in the blood". The higher the rate is, the more difficult it becomes to make illegal actions without being apprehended by police.
Moreover, we wanted to involve disruptive elements that we call outside forces. For instance, the government will be an external force in the way that you will be able to ask him for more lands, for a title deed, or on the contrary you may be asked to pay some taxes, etc...

Collector

This is the lowest step of the game. The main quest for the Collector will be to quit this "job" and to make his career evolve.
Collectors are populations in search of land for farming. They use only simple tools like ax, shovel, or pickaxe. They just collect some resources in the forest (leather, rubber, small wood, etc...) to survive, and sometimes to sell in order to buy food.

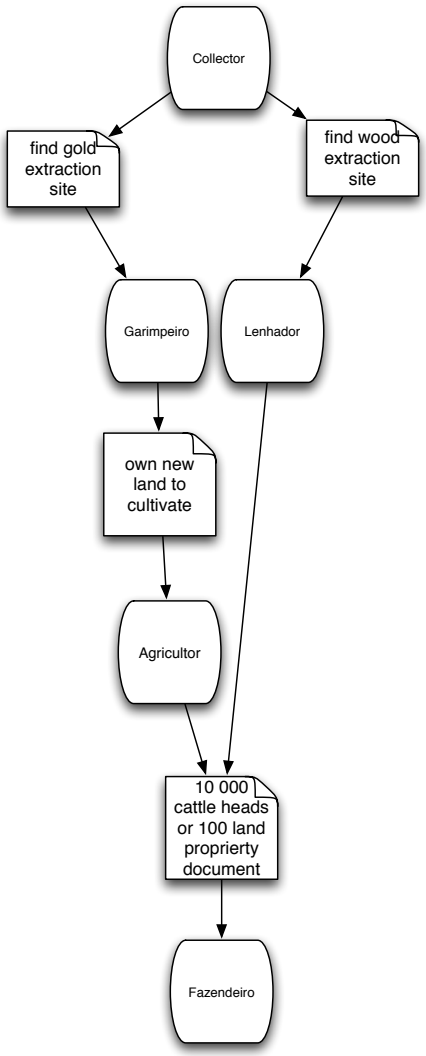

Figure 2.1: Graphical evolution of the career (one possible scenario)

Garimpeiro

Garimpeiros are gold miners. It means they search for minerals, fossils, precious metals or mineral specimens. The activity can be performed in the open-pit, after floods that let emerge minerals or rocks, or in mines. Garimpeiros work in a place called "garimpo". It can be a highly predatory activity for the environment if it is not done with proper environmental management.

Lenhador

Lenhador are wood loggers. They work for sawmill owners. Deforestation has been estimated (using LANDSAT imagery) to an average of 15 383 km2/year over the 2000–2002 period, excluding the state of Amazonas. Yet, the impact of logging is greatly magnified because of the increased probability of forest fires entering the logged areas [2].

Agricultor

The livestock is a specific part of agriculture. The practice of producing and reproducing livestock is a vital skill for many farmers. They are mostly families (recent immigration in order to have land concession). They rise children in order to have more workers. They don't have the money to hire additional men (so they cannot open great surfaces in a year). If they are not able to bear a dividend, the government can take back the land.

Fazendeiro

The player is in charge of a very big farm and has many people working for him. So he has livestock, fields to be cultivated, and a lot of mechanized farming tools. He builds illegal roads and sell pieces of land to other players. Sometimes workers can revolt and stop to work asking for higher wages, this is a part of the *chance cards* explained in 2.2.4.
This is the final evolution for the player. At this step, objectives of the game change. Indeed, before it was a RPG, and the player tried to make his career evolving. But from the moment he became *Fazendeiro*, he will have to manage people who will work for him, to decide when to buy or sell resources collected by them, and to manage its lands. That is what it usually called a strategy game.

2.2.4 Chance cards & Quests

Self-generated chance cards principle

The idea is to create random events just as chance cards in some famous games such as *Monopoly*[4]. Those events would happen to players and they may propose

[4]Famous board game published by *Parker Brothers*, a subsidiary of *Hasbro*. The game is named after the economic concept of monopoly, the domination of a market by a single entity.

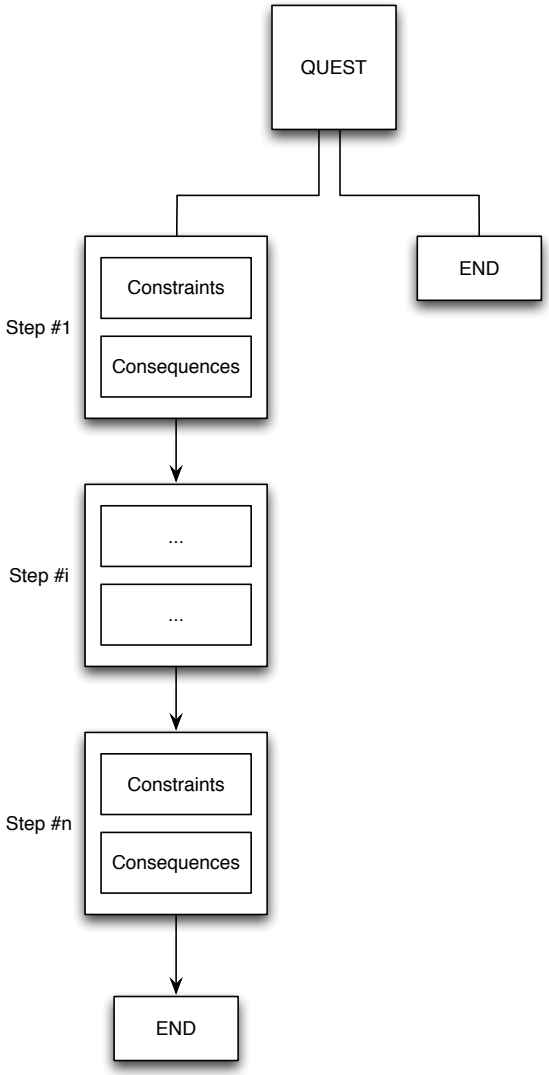

Figure 2.2: General schema of a quest

help. But they may also ask for a quest (i.e. "Your aunt is very sick, you have to find two heads of cattle to heal her.", or "The Government decided to give you another land of 100 ha."). Those "cards" are external forces that make the game a bit more exciting. They should consider your cash and your role in order not to ask for impossible quests.

Kinds of self-generated quests

In order to avoid to implement one by one each quest we thought to create a sort of *quest generator*. In this way the quest will be randomly generated starting from a set of elements, following a set of algorithms. Each of this quests will have constraints and consequences. Thus we propose four kinds of self-generated quest templates in order to be the more adaptable as possible.

Get a thing The general template of this quest would be to propose the player to get a thing which would be more or less rare in the world.

> Hi `<player name>` ! I'm looking for someone that will `<action>` a `<amount>` of `<object>`. I will give you `<reward>`. Would you like to do it ?

Legend This kind plays more with the legend part and the beliefs of amazonian people. The template should be like this :

> Hi `<player name>` ! There is a problem ! `<problem description>` ! Would you please go and `<action>` ? I will give you `<reward>`.

Particular constraint : you need high mood in order to do this kind of quest, so part of the quests could be improve your mood.

Social or individual This one proposes the player to perform a task, with the choice to try to convince another player to perform it with him, in order not to lose time, or to do it on his own.

> Hi `<player name>` ! You need a `<object>` otherwise you cannot `<action>`. For doing this you need `<items>`. Once you have the items it will take you `<amount of time>` to `<action over the object>`. But if you find someone else to help you it will take half of the time.

Mandatory social These quests should be performed mandatory by several players. It enforces the impression of community and support among the players who all have something to gain.

> Hi `<player name>` ! There is `<particular object>` in `<place>` ! But it's too big to `<action>` it alone ! If we can find `<n>` other people maybe we can `<action>` it !

Self randomly generated objects

As the generation process of chance cards and quests, we needed to be a bit more independent and not to only allow players to modify the virtual world. In a persistent world, if people play from all over the real world, some of them could be playing while others could be sleeping, so the interaction is difficult between them. We decided to generate automatically and randomly some objects.

> Each n time unit, the object Ω will be generated at the (x, y) coordinates on the map.

This simply means that each n hours / minutes / seconds, for example, a tree will product fruits, a place in the river will product fishes, a rock will product gold automatically. The player will explore the map and interact with these automatically generated objects.

2.2.5 Collaboration in the game

Collaboration is the watchword of the project. That is why we wanted to introduce it much in the game. We therefore propose several tools related to the issues involved – some more traditional than others – allowing players to interact among themselves in an intelligent and practical way.

Classical text chat

This is the very classical way to communicate. A multiplayer game without at least text interactions between players would be totally inconceivable. We decided not to make an exception to this rule. Because people like it very much for its simplicity and the fact that it is non-intrusive. The player chooses when he wants to begin or to end a communication.

Audio & Video chat

In order to complete the classical text chat, we thought that it could be great for players wishing to be able to interact more efficiently with others, to propose them to use the audio and video technologies. These technologies – which are not often directly integrated in video games – tend to gain ground, especially as VoIP software are increasingly used by the general public. So we decided to provide audio and video multi-conferencing, based on private chat : it means that only players who are "friends" can play it.

The market place

The principle of this feature is quite simple, because we can compare it to an existing website : *eBay*[5]. So the interest of this feature is to allow players to buy and sell goods according to the market principle : if the request for a resource

[5] An online auction and shopping website in which people and businesses buy and sell an incredibly wide variety of goods and services worldwide.

is high, the players may have to sell it at a high price. Conversely, the player who seeks to buy a resource that nobody wants will be able to get it for cheap. More in details, people will be able to set some of their goods for sell, specifying a minimum price. From this moment, these goods appear on the market, and other players can buy them just by clicking on – verifying their amount of money – and accepting the transaction. It is worth noting that if the buyer wants to give more money than the minimum price, he will be able to do it.
It is evident that all these transactions do not require both parties to be connected at the same time, because it would be too restrictive for players. The seller can put on sale something, and then log out, the buyer will be able to acquire it whenever he wants to, at least until the object has not been purchased by someone else or withdrawn from sale by its owner. And on the contrary, if a player seeks for a particular good, he can use in the game a special mechanism that allows him to buy automatically this good for a fixed maximum price that he decides himself.
We can moreover note that a player can directly interact with another one to sell him specifically certain objects for a "friend price".

Collaboration for quests

As we discussed it in 2.2.4, the quests are really present in this kind of games. And, in our humble opinion, the main drawback of many RPG is that quests are only solo challenges. Therefore, we thought that it would be better for players to interact with each other to succeed in achieving a common goal. That is what we called "Collaboration for quests". It means that some quests are unachievable without the help of other players. In this case, the player who is chosen for the quest has to find people who are ready to help him, knowing that they will gain experience points too and share the benefits. We assume that this kind of quest will ensure virtual links between players, and will contribute to enhance the community.

2.3 Modeling

The modeling we made is mainly about the "functional kernel" of the game. We distinguished three main parts : the player, the land and the objects modeling. Diagrams are available in the appendix B.

2.3.1 Players

The *Player* class is the base of our game. Its properties are the common characteristics of the player – name, money, health, etc... – and it implements some methods to interact – `buy(good: Good)`, `performAction(action: Action)`, `sendMessage(recipient: Player, message: Message)`, etc...
The *Player* owns a list of *Good* objects – which simply are *GameObject* with a

price. There are also four classes – that correspond to the five levels of characters to which the player tries to reach – that specialize the *Player* class : *Collector*, *Garimpeiro*, *Lenhador*, *Agricultor* and *Fazendeiro*. B.1

2.3.2 Map & Lands

The *Map* class has a few properties such as the name of the region – optional, it is just be more realistic – a matrix containing *LandSquare* objects, and the total number of players. The *LandSquare* class has properties to know its owner, the object or the player on it, and if it is crossable. There are a lot of classes which specialize the *LandSquare* class, such as *Road*, *Field*, *Forest*, *Water*, etc...
It is important to note that in the *LandSquare* class, there is a very important method that returns the possible actions that the given *Player* can perform on it, considering the player's properties, objects and characteristics. B.2

2.3.3 Actions

At first, we thought about putting every possible action performable by a player as simple method in the right character class. But it appeared that this was not the right choice. We wanted to be very flexible, and to allow anyone to perform any action – according to the needs of the action. So we decided to design an *Action* interface that only specifies a simple `execute()` method that is due to be redefined in specialized classes. We found out three main kinds of action which the real actions derive from : *UseSomething*, *CollaborateOne*, *CollaborateMany*. For instance, we have the *Harvest* class that derives from *UseSomething*, and *WorkFor* that derives from *CollaborateOne*. B.3

2.3.4 Objects

The *Objects* diagram is quite simple. There is only one mother class named *GameObject*, and all the different objects specialize it. The trick lies in the fact that we have designed interfaces for each type of object. Using interfaces allows to create objects that can implement multiple interfaces in order to simulate multiple inheritance. For instance, it is important to create a *Boat* object that implements both *Buildable* and *Usable* interfaces. B.4

Buildable : objects that the player can build (*Road*, *House*, etc...).

Collectable : objects that will be collected (*Guarana*, *Rubber*, *Fish*, etc...).

Cultivable : objects that the player will be able to cultivate (*Manioc*, *Rice*, etc...).

Raisable : objects that will be raised (*Livestock*, *Children*, etc...).

Usable : objects that the player can use to perform an action (*Axe*, *Shovel*, *Boat*, etc...).

Chapter 3

Architecture

How did we solve the technical issues ? How did we decide to develop the different parts of the project ? The answers are in this section which explains first the server side, and then the different aspects of the work made to the client side.

3.1 The server

3.1.1 Connection process

In order to connect to the game server, player must create an account using his e-mail address which will be used as an identifier. In the game homepage, an interface invites players to enter the e-mail and password they have chosen during the inscription process, if they're already registered. Otherwise, they can get registered on this same page.
Once the information required for the identification process is sent to the server via the interface, the server connects to a database to verify the accuracy of the information. If everything is fine, the player is connected and ... fun fun fun for everyone !
Once one player has passed the identification process, the server launches a thread specially dedicated to the communication with this player. Because during the game there will be a lot of exchanges of *XML* messages for almost everything the player will do.
The server doesn't only receive *XML* messages but also sends them to the clients. For example, to inform that the information entered by the player are correct and he can enter the game, or also for transfer all instant messages.

3.1.2 Parsing XML

To parse *XML*, the server which is written in *Python* uses the *SAX* module. We have to define different handlers to suit the actions of the clients. Some actions made by players need the server to update or check information in the

database. For instance, in order to login, the client sends an XML tag like this one to the server : `<login mail="mail@address.com" pwd="mypass" />`
The server parses it and retrieves the e-mail and password. This information will be compared with information stored in the database to make sure of the existence of this player. For more information about *SAX*, check out the API [16].
Thereafter are some xml tags we use :

Action	XML tag
To send a message	`<msg from="" to="" content"" />`
To sale or buy something	`<chgOwner buyer="" saler="" object_ID="" price="" />`
To put an object for sale	`<forSale mail="" object_ID="" price="" />`
To move a character	`<move player="" fromX="" fromY="" toX="" toY="" />`
To build an object	`<build mail="" object_ID="" />`
To pick a thing	`<pick player="" object_ID="" quantity="" />`

3.1.3 Database

The database is used to store game data. The database structure tries to follow *UML* diagrams in the modeling part. The database stores information about the players and also every useful data for the game such as details about all characters, what they own, what they buy or sale etc...
In order to connect to the database and make queries, the server uses the *MySQLdb* module in Python [17], which is quite easy to use.
The database schema could be like this :

Character (id, name, description, image)

User (id, email, pwd, age, sex, country, character_id)

Player (user_id, character_id, health, mood, hunger, money, xp)

Good (id, name, description, price, image)

GoodOwned (player_id, good_id, forSale)

Resource (id, name, description)

LandType (id, name, description, crossable)

Land (x, y, type, owner_id, resource_id, resource_amount, player_id, good_id)

GetAThingQuest (id, user_id, action, amount, good_id, reward)

LegendQuest (id, user_id, problem, action, reward)

SocialQuest (id, user_id, good_id, x, y, action, nplayers)

SocialOrIndividualQuest (id, user_id, good_id, needs, time)

ActionPerformed (id, user_id, action, timestamp)

3.2 The client

3.2.1 Graphical User Interface

The graphical interface of a game is a delicate part of its achievement. We wanted to make an attractive and pleasant interface, both easy to use and with a pleasant design.
The language chosen to implement the interface is *Flex*, developed by *Adobe* to create web applications using *Flash* technology. Like *Flash*, *Flex* creates *SWF* files that are rendered by *Flash Player*. *Adobe* provides a free and open-source development kit – *Flex SDK*[1] – that includes predefined class libraries, application services and a standalone compiler. Building *Flex* applications implies the use of two languages :

1. **MXML** which is an *XML* based markup language that is primarily used to layout application display elements.

2. **ActionScript 3** which is an *ECMAScript* compliant object oriented programming language.

But above all, we chose *Flex* – indeed *Flash* – for the ease of distribution of the bytecode. *Flash Player* can be installed on many platforms[2] – such as Windows, Linux and MacOS – and we wanted to create a game embedded in a browser to avoid installation problems. The main mandatory elements for the achievement of the game were found quickly : a "Home" part for the presentation of the game and allowing the user to log in, register and manage their profile and personal information. Then a "Game" part, dedicated to the game itself, including the game map, a menu to access the different options available, and a sign listing all the player's properties (health, money, etc.).

Home

For this part, the first important point was to facilitate and simplify as much as possible the log in and the registration mechanisms. This is the reason why we chose only to ask just for an email address and a password in order to complete the registration. Always with a view to simplify, the registration button "Need to register" was included in the log in panel. There is a simple effect that expands the panel with another text field to confirm the password for the registration. In order to increase the attractivity, some graphical effects are used in the interface.
The second important point was for the "Header". It has been designed with the idea of putting the (future) player in the atmosphere of Amazonia. That's why you can see photos of different local activities performed, and an aerial view of the broader region.

[1] The *Flex SDK* is available on *Adobe*'s servers here : http://www.adobe.com/cfusion/entitlement/index.cfm?e=flex3email

[2] According to *Adobe*, almost 98% of internet users have *Flash Player* installed in their web browser.

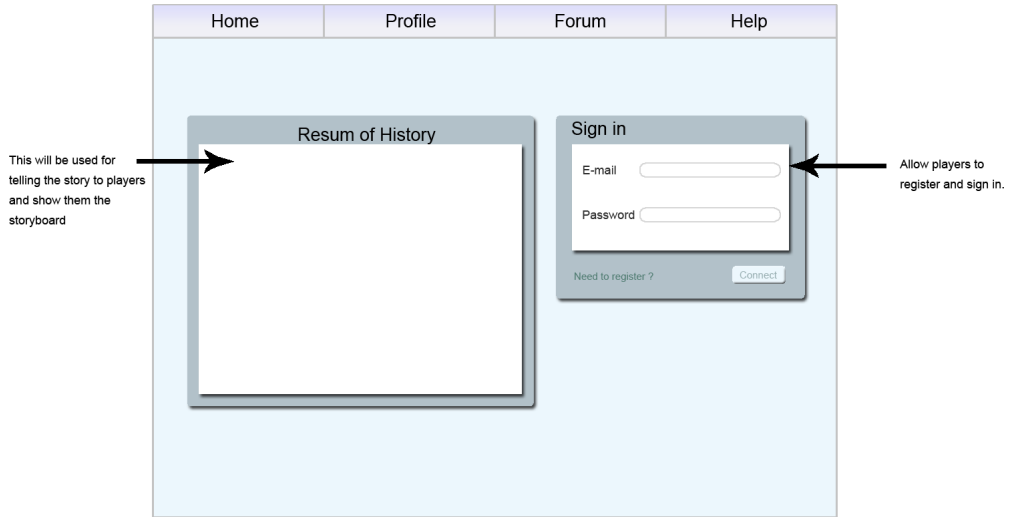

Figure 3.1: Global view of the Home part

Game

For this part the most important element was the design and implementation of the menu. In fact, to meet our goals we wanted an easy to use menu with a pleasant design. To do this, we chose a simple animation, allowing with a simple click to enlarge and reduce our own menu items. Five different menus were created for the possible actions.

Journal The Journal menu lists the various quests carried out and ongoing. It allows to easily view them as they are arranged chronologically.

Career This menu shows all characters available in the game. They are enabled as the player wins experience, money and other parameters. A description for each character appears after it has been clicked.

Build Here are gathered, organized by category, all the objects that you can build. Their description are available on the right.

Market Two views are available for the market place. The first one proposes to list players who have something to sell, while the second one orders the list by objects for sale. For the first one if you click on a player icon, it will popup the objects he sells. By clicking on an object in the second views, it will popup the different players who sell this type of good. Moreover, there is a search bar for easier access to a player or a specific object.

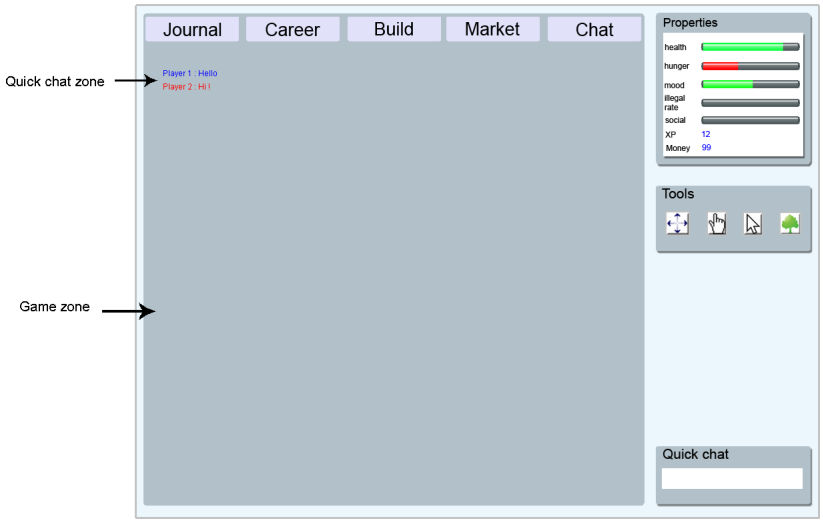

Figure 3.2: Global view of the Game part

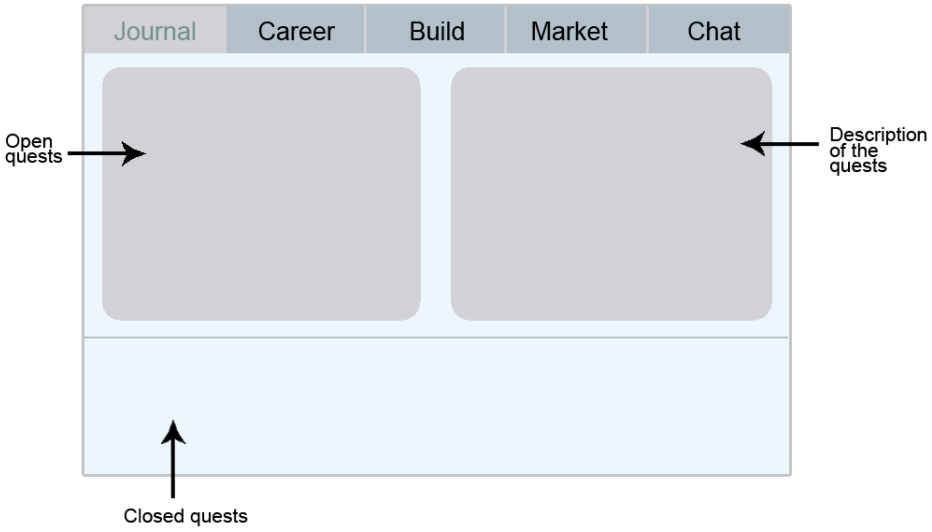

Figure 3.3: Sketch of the Journal menu

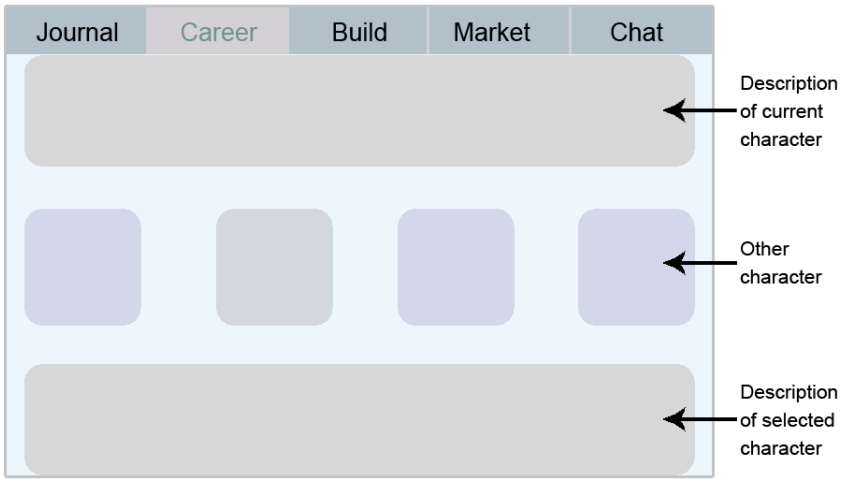

Figure 3.4: Sketch of the Career menu

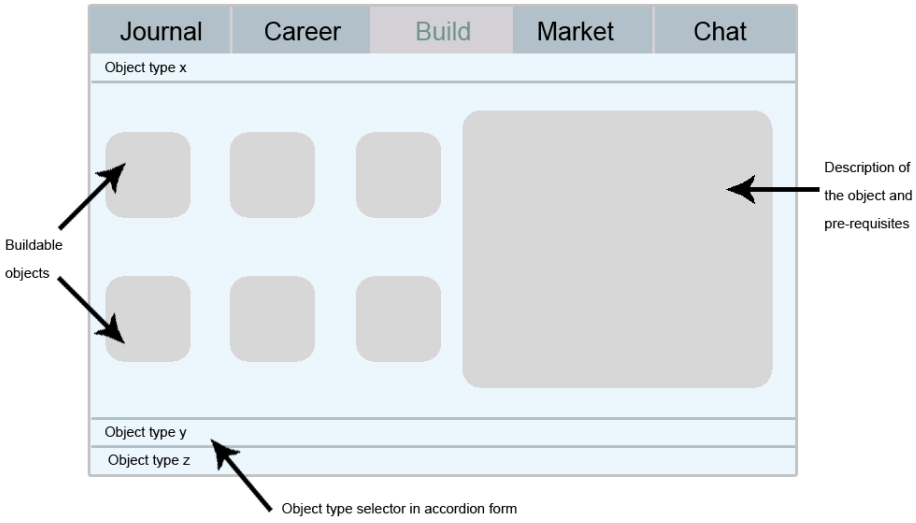

Figure 3.5: Sketch of the Build menu

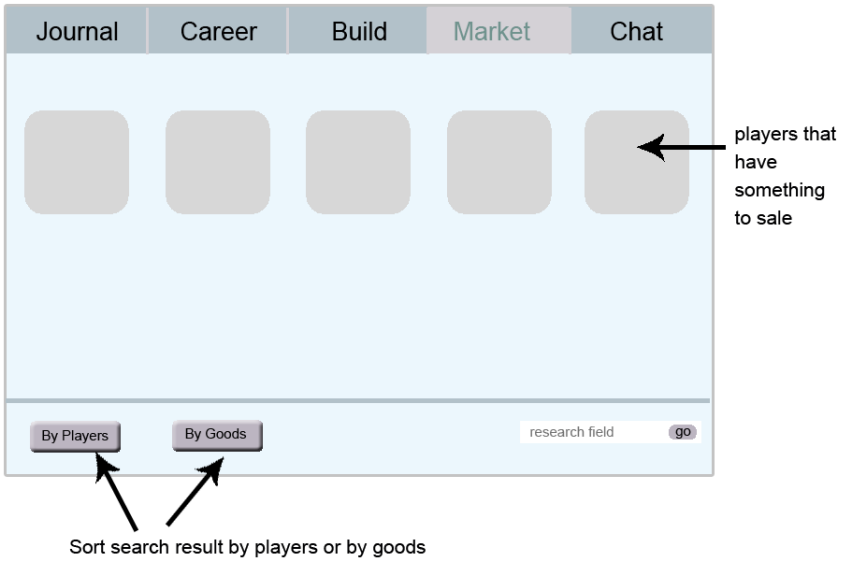

Figure 3.6: Sketch of the Market menu

Chat The chat menu provides two sections. The first one is used as a contact list which you can expand using the contact search bar. The second one is dedicated to the different conversations you may have with other players during the game.

3.2.2 Graphical engine

Since we didn't find a really convincing one, we created our own 2D graphical engine.

State machine

It is interesting to note that *Flex* includes native support for states. These states are often designed with a transition from one GUI to another in mind. But they have the functionality we needed to change between states that don't necessarily have any GUI components. Modifying the *currentState* property of the Application triggers a state change which can be used to update the internal game state to match.

Double Buffering

Double Buffering is a technique used to remove the visual tearing associated with drawing directly to the screen. It gets its name because we use two graphics buffers to draw the final image : one that resides in-memory (the back buffer),

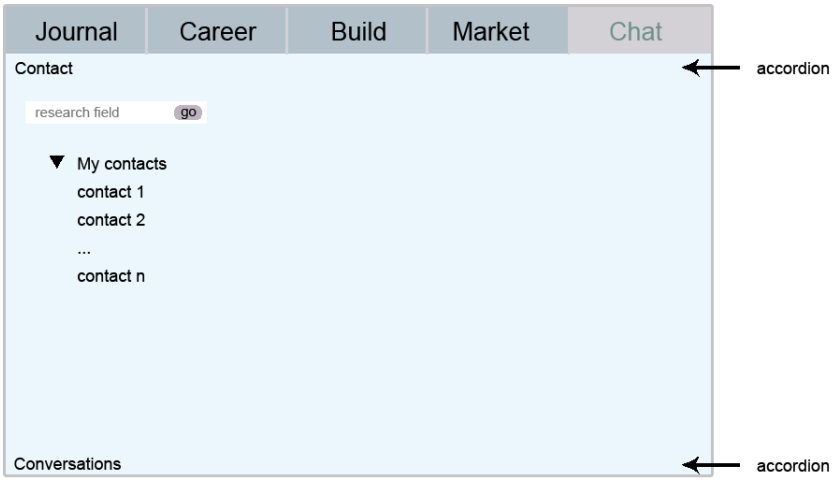

Figure 3.7: Sketch of the Chat menu

and the buffer which is displayed on the screen (the front buffer).
The back buffer can be seen as a kind of scratch pad which is built up as the individual elements that make up the final scene write to it. Once a frame has been drawn, it is copied to the front buffer in one operation. The screen then displays the contents of the front buffer.

Game Object Manager

In order to manage all the objects in the game, we created a class whose role is a kind of a conductor. It implements the *Singleton design pattern*. In this way, we can be assured that only one *GameObjectManager* will ever be created. It contains the back and front buffer of the game, a collection of all the objects visible or not in the screen and all the methods to update the display using the double buffering strategy.
The game objects themselves – characters, goods, land, etc... – have four main properties – among more :

- The position property that simply defines the position of a *GameObject* on the screen.
- The *zOrder* property that defines the "height" of the object on the screen. A *GameObject* with a low *zOrder* – like a building on the ground – will be drawn beneath a *GameObject* with a higher *zOrder* (like clouds for example).
- The graphics property which is a reference to a graphics resource. This graphics resource is the picture that gets drawn to the back buffer.

- And finally the *inUse* property which simply flags the object as being active in the game. We use this property to allow the use of resource pooling (see the next subsection).

Then the specialized classes of *GameObject* can have more attributes like speed for the moving objects or more graphics for the characters to represent them in the eight directions. [9]

About the garbage collector

One of the downsides to garbage collection is that object destruction in non-deterministic. That is to say that we have no way to know exactly when an unreferenced object will actually be destroyed. Garbage collection also incures a performance hit. Having the garbage collector trawl through hundreds of objects and destroy those that are no longer referenced may take time.
So in order to avoid destruction of objects that may be reused a few time later we set up a technique of resource pooling that solves many of these problems. Basically it keeps a collection of objects in a pool. As the objects are needed they are taken out of the pool and initialized (note that initialized is not the same as being created from scratch with the new function). Once they are no longer needed they are cleaned up and put back in the pool.
By keeping objects in a pool we avoid having to continuously create new objects. This in turn reduces the number of objects that exist in memory, which reduces the workload for the garbage collector. Also, by creating and destroying resources in a deterministic way (when the object is taken out of the pool, and before it is returned) we gain more control over the memory management process.

3.2.3 Artificial Intelligence

A*

As the map is digitally represented by a matrix, we could easily implement a mechanism for pathfinding. We therefore decided to implement a simple A*[3] algorithm, which is a best-first search algorithm that finds the least-cost path from a given initial node to one goal node (out of one or more possible goals). This algorithm uses three functions to determine the path-cost : $f(x)$, $g(x)$, $h(x)$. Theses functions are linked with a sum relation :

$$f(x) = g(x) + h(x)$$

The path-cost function is $g(x)$ which is the cost from the starting node to the current node. The $h(x)$ function is an "heuristic estimation" of the distance to the goal – this is part of the final result because as any heuristic, if it is efficient, results are good, but if not, this is catastrophic.
In our case we used a very simple heuristic, due to our implementation of the

[3]Pronounced "A star"

map as a simple matrix. Thus our heuristic just uses the real distance as the crow flies, multiplied by a coefficient.

$$|current.x - arrival.x| + |current.y - arrival.y| * 10$$

Decision trees

We thought to use the decision trees mechanism for a few things in the game that required artificial intelligence. A decision tree is a decision support tool that uses a tree-like graph for decisions and their possible consequences. Decision tree learning, used in machine learning, uses a decision tree as a predictive model which maps observations about an item over conclusions about the item's target value.
The first use we thought for decision trees is for the "quest generator" and for the "chance cards", in order to create smart and less predictable algorithms. In fact, the "quest generator" and the "chance cards" need both to be configured in order to propose "suitable" quests or cards for the players. For instance, the "chance cards" should consider the player's properties – health, money, hunger, experience, etc. – and should try – for example – to help him when he is poor and inexperienced, and not necessarily encourage him when he is rich and healthy.
Beyond that, we can think to implement behaviors for bots that play roles, just as any real player. Their behavior should be interesting to model, since the number of parameters to take into account is very huge. We can start with a minimum of variables, then the implementation will be done in a few steps, by improving the decision-making part at every step. Moreover, it could be interesting to introduce a mechanism that would regulate the total population connected – by not allowing bots to play alone when there is no real player connected – in order to be more fair towards real players.

Chapter 4

Perspectives & Conclusion

It is always important to step back on our own work and wonder what are the positive and negative points. That is what we are trying to do in this section by reporting on the status of the project at the moment, and then by asking questions on the following events concerning the future of this project.

4.1 Progression of our project

4.1.1 Far from our first ideas

It might be fun to see that finally, the phase that has taken the most time in this project has been to successfully identify the real issue to deal effectively and intelligently.
Our very first idea trying to simulate the biodiversity issues in Amazonia with just the external possibility for the "player" to deal with some parameters making evolving the world differently was totally wrong.
We pre-designed a whole system using a biological database, based on a "CAUSES / CONSEQUENCES" rules system – which was very focused on deforestation. All these data could have been inserted by non-informaticians – like biologists – via a simple web interface. These data should have allowed to automatically change a virtual world where players could freely evolve.
Once we had the matter well in hand, we could develop it fairly fastly, relatively speaking. We therefore moved from a simulation with any external input of a possible player to a true RPG that puts players in Amazonia in the center of the study. The player's goal is to become an actor typically responsible for deforestation, who has to perform the actions necessary for his survival or his development according to its progress in his "career".

It made great changes in the plan, but in our discharge, the subject deserved clarification and time to grasp.

4.1.2 A long-term project

This project is obviously not intended to be completed in a short time - just three months for students who do not do this full time. The basic idea was to set up the foundations. This project already has a good foundation and is easily implementable by some motivated people, while a project badly designed and quickly programmed will never evolve in the future. We therefore very much preferred documenting up our thinking on this project as opposed to program hard things, which without any documentation would not be very useful for the future.
It is also why we definitely wanted to use and stress the importance of collaboration tools to create persistent knowledge (A.2.2), so that the fruit of our reflection could be available in a distant future for people who wish to rely on it. But we will discuss that future in the section 4.2.

4.1.3 Our prototype

We want now to explain precisely what our prototype does. As we already said, we axed the project a maximum on the analysis because we knew when we started that we would not have enough time to produce a real and well finished game. Therefore our prototype is quite simple, even tiny we might say.
Nevertheless, here is what has been done for the moment :

- A GUI that allows the user to register, connect and perform actions into the game.
- A server that allows a player to register, accepts client connections, receives and sends messages to all connected clients, and saves modifications in the database.
- A set of characters that allows the player can evolve in a virtual world and can interact with their environment.

4.2 The future

4.2.1 Students from next year ?

Shortly after the project began, we realized that we will not have a finished product, even while working on this project full time. And since that day, we decided to concentrate our efforts on the transfer of our reflection. We felt that if our thinking was well structured and well thought out, the implementation would be feasible in the future by other people. That is why we are talking about students who will come after us. The implementation could be done through a future student project for Bachelor or Master degree. The theoretical part is widely roughly trimmed, the students in question would just have to absorb it, and then implement it – knowing that our prototype source code will also remain available.

4.2.2 Next important steps

If we had had more time, or in the hypothesis that the project will be continued in the future by others, we decided to enumerate the major next steps of the project.

1. About the game design, we think about reviewing the evolution of the career by introducing new characters and new rules of evolution, in order to improve the diversity in the game among the characters as it is explained in the article [2].

2. Calculate and design the decision trees – helped by the method explained in 3.2.3 – the quest generator and the chance cards mechanisms. Then implement them because that would significantly improve the gameplay by adding self randomly generated events.

3. In the case that there would be only a few players connected at the same time, we believe that achieving the intelligence of bots playing real roles in the game would also be interesting. In this idea, we could even use decision trees or even learning, in order to provide them true good reflexes to survive in Amazonia.

4. About the graphics part, it could be great to improve them by adding some animations – when the sprites are moving for instance – and by fixing some ergonomics issues.

4.2.3 Criticism of our work

Despite missing time, it is legitimate to ask whether we could make different choices. If yes, why, and especially if these choices had been made, would they have changed our current work ?
The first is certainly the one that calls into question our main development options: *Flex* clients and a server in *Python*. We looked at the multi-agent platforms at the beginning, then we have abandoned this idea because of the specifications – a client easily disseminated via Internet – which we hastened to the *Flash* technology and *Flex* specifically. But today, we wonder if it was not necessary to take more time to look at possible multi-agents platform solutions[1] and especially take the distance to the specifications. This would surely have changed dramatically – perhaps even accelerated – the development of the software, since we could have programmed with the agent paradigm, which has its own rules. Moreover, the environment management would not have had to be made, because everything could be programmed as an agent – more or less intelligent. So the intelligent automated players programming would have been simpler.

According to Adobe, almost 98% of internet users have *Flash Player* installed in their web browser. That's why we rapidly decided to choose *Flex* as a

[1] Such as MadKit or JADE

development platform. Two other famous platforms are *Microsoft Silverlight* – proprietary – and *JavaFX* – open source. But due to the fact that there is less internet users that have *Java Runtime Environment* installed we didn't dare to take the plunge.
However, we found out using *Flex* and indeed *ActionScript 3* that it is not possible to create threads in *Flex* which we assume must be possible in *JavaFX* since it is based on *Java*. Thus, we thought of two main features that we could implement using threads in the future.

1. The first would be to think of the possibilities to distribute to the clients all over the world parts of the computation done on the server. This would be as great as a reflection task than as a feature. Distribute what is distributable...

2. The second, which is a part of an answer to the first, would be to create agents – through threads – on the client side, and use those agents to manage bots animated with decision trees (3.2.3). Since we said there would be no bot playing without a human player, it could be great to save processing power on the server this way.

Bibliography

[1] *"Causes of Deforestation of the Brazilian Amazon", World Bank Working Paper No. 2,* Sergio Margulis*, 2004*

[2] *"The Roles and Movements of Actors in the Deforestation of Brazilian Amazonia",* Philip M. Fearnside*, 2008*

[3] *"Amazon Deforestation Models", G. Camara, A. P. Aguiar, M. I. Escada, S. Amaral, T. Carneiro, A. M. Monteiro, R. Araujo, I. Vieira, B. Becker. Science, vol 307, 15 February 2005, pp. 1043-1044, http://www.dpi.inpe.br/gilberto/lucc/camara_amazon_deforestation.pdf*

[4] *"Weaving the Software Development Process Between Requirements and Architectures",* Bashar Nuseibeh*, 2001 http://www.cin.ufpe.br/ straw01/epapers/paper05.pdf*

[5] *"Geospatial Change Modeling Research Group", website of Tiago's research team, http://lucc.ess.inpe.br/doku.php?id=start*

[6] *A wiki we set up in order to create persistent knowledge about our project, http://agora.lirmm.fr/asg09*

[7] *The SVN repository for our project prototype, http://asg09.googlecode.com*

[8] *"14th International Conference on Artificial Intelligence in Education" http://www.aied2009.com/*

[9] Tutorial on Flex, *"Flash Game Development with Flex and Actionscript",* Matthew Casperson*, http://www.brighthub.com/internet/web-development/articles/11010.aspx, 2008*

[10] Wikipedia, *"The free encyclopedia that anyone can edit", http://en.wikipedia.org, 2001*

[11] The Open University, *The UK's distance learning government-supported university, http://www.open.ac.uk/, 1969*

[12] Flashmeeting, *"The Flashmeeting project : a better way of working", http://www.flashmeeting.com, 2008*

[13] GoogleCode, *Free SVN repositories for open source projects provided by Google, http://code.google.com, 2008*

[14] *"Get oriented to Flex", http://learn.adobe.com/wiki/display/Flex/Get+oriented+to+Flex*

[15] JavaFx, *Official website of JavaFX http://www.javafx.com*

[16] *The SAX library for Python http://docs.python.org/library/xml.sax.html*

[17] *Writing MySQL Scripts with Python DB-API http://www.kitebird.com/articles/pydbapi.html*

Appendix A

Organization

This section highlights the processes used in the organization of the working group. It deals with the work organization explaining what was the role of each of us and what was the timing we had to juggle with.

A.1 Work organization

A.1.1 Partitioning of concerns

There was a very good cohesion in the group, even if we did not know each other that much at the beginning of the project. We all complemented each other.
We can't say there was a real hierarchical organization, the roles changed depending on the needs of the moment.
At the partition of labor, the distribution was fairly simple by our own routes and our affinities to some particular problem we wanted to address. And thus to detail it more precisely, *Panupat* took care of the server and *Guillaume* of the client interface. *Alexandre* worked on the overall design, UML diagrams, the graphical engine and has also served as PR[1]. *Fabien* has contributed to the overall design, UML diagrams and to the Artificial Intelligence part, as well as writing almost all the documents produced for this project.

A.1.2 Time

We provided a Gantt diagram in our requirements document, but given the differences between that one and the real timing, we preferred to update it and to propose graphically the new one here.

[1] "Public Relations". The practice of managing the flow of information between an organization and its publics.

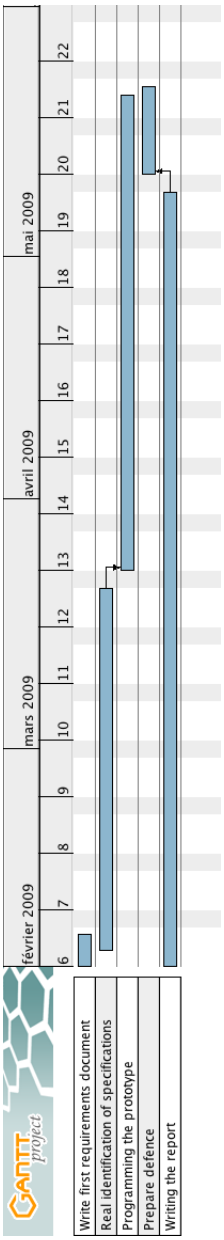

Figure A.1: The real timing of our project

A.2 Processes & tools involved in collaboration

We have distinguished two types of processes in our work of communication. On the one hand remote collaboration to communicate between the different actors of the project, and secondly the knowledge creation and sharing. Both parties are clearly explained in this section.

A.2.1 Collaborating and communicating

Meetings at LIRMM Those meetings are more recurrent than the meetings with Tiago. It is the opportunity to summarize and discuss the ideas we had since the last meeting. There, we reviewed the need and the roadmap of the project. It allows to give everyone a global vision of the project, and especially to motivate troops.

Meetings at home Meetings at home are really more informal than anywhere else, because there is no framework, but it doesn't mean that they are useless. They allow to think more easily : there are more ideas that spring, but sometimes lower quality.
But the fact is that it also allows to discuss special technical points, that it is very useless to discuss in more formal meetings we made at LIRMM. It seems that for more efficiency, the best is to work in pairs.
The process of staying together and eat after the meeting contributes to improve the cohesion of the work group, because we know each other better, and in other context.

Flash Meeting Flash meeting is a collaborative tool which allows to make meetings in real time with participants from all over the world. Flash meeting is a project initiated by the Open University[2] in UK. It is full of interesting features to help the collaboration and to forget that the meeting is virtual.
The principle is pretty simple because, you just have to book a new meeting using a time zone helper if participants are not in the same country. From this moment, the booker sends a mail to all the participants to give them the URL for the meeting. Going on this page allows you three things : if the meeting has not begun yet, you can check the time left before the beginning of the meeting, else if the meeting has begun you can access it and speak with the others, and if the meeting is over, you will be able to replay it, because every meeting made with FM is recorded.
There is some interesting points using this tool. The first one is the simplicity with which you can access the meeting, you just need a web browser using Adobe Flash © and an internet connection. Moreover, the handling of this tool is very pleasant and easy. There is a real management of the "speaking turn", and each participant can queue. But if you really want to interrupt somebody, you can.

[2]The Open University is the UK's distance learning government-supported university. It was established in 1969 and the first students enrolled in January 1971.

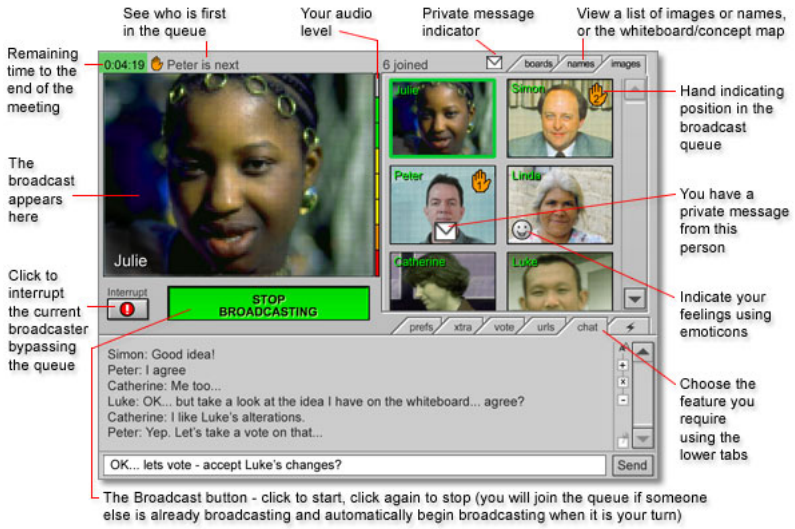

Figure A.2: Flashmeeting's user interface in the middle of a conference.

There is also a vote system, with counts very clear. While someone is speaking, the others can interact with him using smileys to show their approval or their disagreement, or using the text chat to discuss something. Sharing files and URLs has been also implemented in FM, and it's really interesting to discuss a file while others are seeing it.
For the meetings we did with FM, everybody was always present. Because of the virtual possibility to meet everyone from anywhere, sometimes we were all together at the LIRMM, and sometimes some of us were at home.
The way we used FM was very interesting because the fact that the meeting has a limited duration (one hour and a half in our case) led us to effectively prepare it, what we never did for our "informal meetings" (i.e. without Tiago). So it makes the meeting pretty effective. In our case, the meetings have been always nearly following the same pattern, because during the first meetings it was predominantly Tiago (and also Stefano because he was the only one to know Tiago) who spoke to expose us his vision of the project. After that, we spoke more and more, because we had taken the project in our hands.
And last but not least, the recording the meeting feature. It has been the most important feature of this tool because when you make a meeting using english with participants from three / four different nationalities, it is very important to listen a few times more the conference, to make sure we understood well everything.

Skype Skype is a very well known tool to the general public, because its aim is to allow people to communicate freely from all over the world with friends or family. We don't know if it's used by researchers, but it could be very interesting

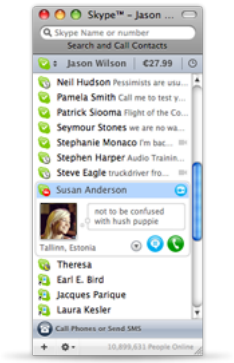

Figure A.3: Skype's user interface : contact list.

because of its simplicity and its audio quality, even in a meeting with multiple participants. Its main drawback compared to FM is the installation process and the need to register : people you want to call have to download, install, and register a Skype account just to be able to receive your call, with FM this process is simplified to the maximum for everybody's pleasure. There is also the impossibility to record the meeting (which is being to be implemented in next version of Skype...).

However, its advantage is the availability to speak everyone at the same time : you can interact more easily and quickly with your interlocutor, and this is a way to be more efficient. But the fact that maybe we less hesitate to interrupt each other, and it could be harder to follow the conversation and to let someone finish what he was speaking about.

Its main interest lies in the fact that we used it in an instant way : if we were connected and we wanted to know what some of us thought about an idea, we just had to launch Skype and to call him/her. Moreover, we were forced to meet Roberto using Skype because we couldn't book a meeting with FM, but the process is the same one. We used it very often, by pair or plus. Certain sessions were very productive thanks to the save of transport time to see each other in real.

Emails Asynchronous communication process with its advantages and drawbacks. No appointment needed to send a mail, you can send information in a message whenever you want, no waiting.

Messages are almost sent to everyone in the group, so there is a persistence of these messages. Whether for temporal information (ask for another meeting on next Wednesday) or timeless information (a new brilliant idea or a meeting report as a file), this is the perfect tool.

It is therefore more difficult to operate a mail sent/received a long time ago. And due to the non-centralization (including attachments), there are file versioning problems. The fact that emails are not always well structured can be

problematic...

A.2.2 Creating and sharing knowledge

Agora GridSharingDesktop Agora is a wonderful tool to share knowledge. In fact it lies at the crossroad of the two parts : it allows to communicate and to collaborate, but also to create and share knowledge because its main purpose is to show to multiple participants the way of doing something and / or to explain it. But the problem is that confronted to this tool, everybody is not equal : it can be a discouraging to use.
We didn't really use it. Maybe we should, particularly at one meeting with Tiago where we wanted to show him the result of our work.
Pascal Dugénie gave us a course to teach us the way how to use it. It is still remaining some bugs on it (especially scroll bars in the virtual desktop...), and the simple fact that we needed a course was not a good thing because what about non-informatician people like Roberto ? That is the reason why we think that this tool is not fully accessible for everyone.

SVN (GoogleCode) Subversion is a powerful tool to manage concurrent versions of files on a big project like this one. It suits absolutely our needs. It is pretty simple to checkout the source code files because there are a lot of free and friendly-user SVN client on every platform.
It allows two people – or more – to work at the same time without fearing data loss, because one can restore any old version of a file uploaded before.
Once installed, it works perfectly with Eclipse (the IDE we use to program the game...), and you only need two clicks to commit your changes or to update the project.
Moreover, we are getting used to it, because it is a very common tool in development, and mostly widely used.

Wiki (based on MediaWiki, provided with Agora) The wiki is not used as a communication tool, but more as a tool for future collaboration. We think that currently this is absolutely not our first way to communicate between us, we are seeing it more as a way to share the knowledge we created from our group work.
Maybe this project will go on after we stopped it, with other students next years. If so, the wiki is a huge amount of knowledge seen as a trace of what we thought, discussed and decided about it.
But also for Tiago, who doesn't follow the project daily, with this wiki he is able to be more aware of what we are designing for the game, and he can also contribute.
To sum up, it provides a well structured and centralized result. An organized persistent trace of our communicational and decisional process.

DropBox This is a useful tool, which is definitively collaboration oriented. A software that allows you to create a folder which could be easily always synchronized between every collaborators, just using an Internet connection.
Every files we produced for this project has been put in this "magic folder". The great thing is that everyone can access these files at anytime. It wasn't a communication tool because each time we shared a new file with DropBox, we sent a mail to everybody to inform them of the existence of this file, but it really dealt with knowledge sharing because one of us decided to create a new file and to put some knowledge into it.

Appendix B

More informations about modeling

Here are gathered the UML class diagrams we modeled.

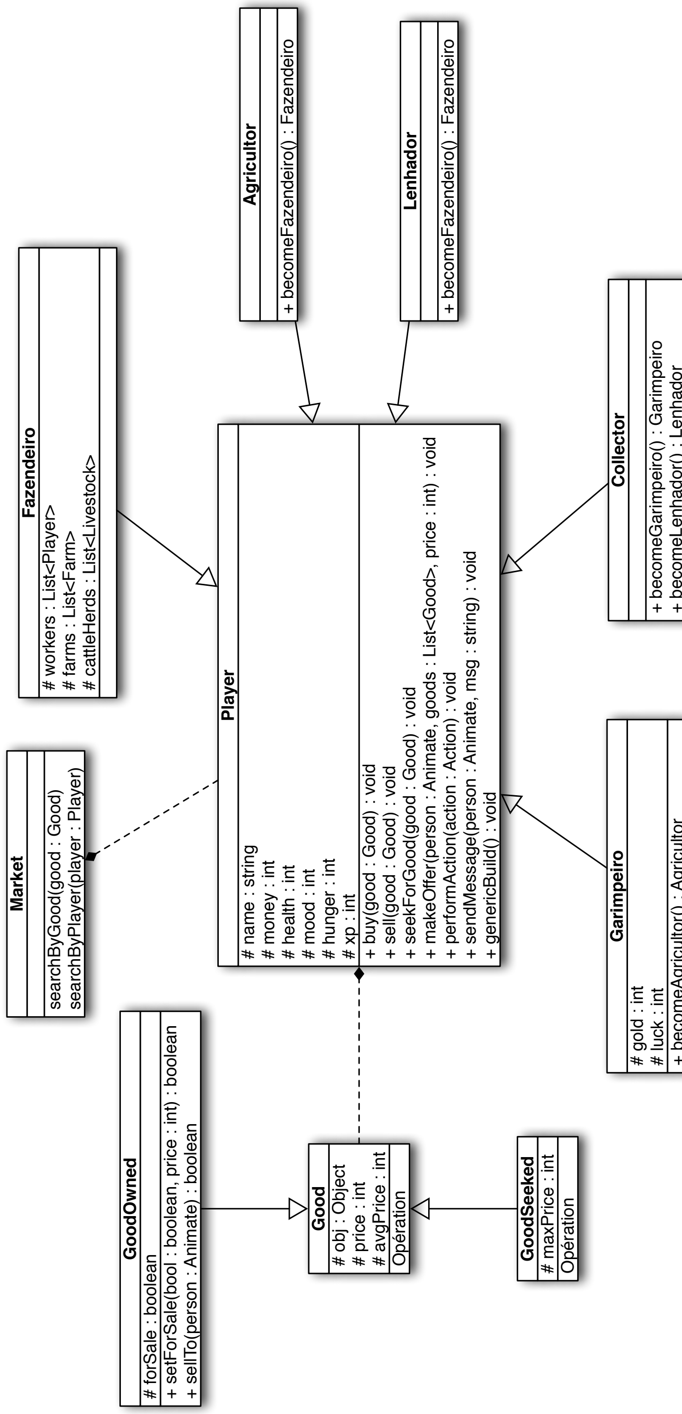

Figure B.1: UML diagram about Players

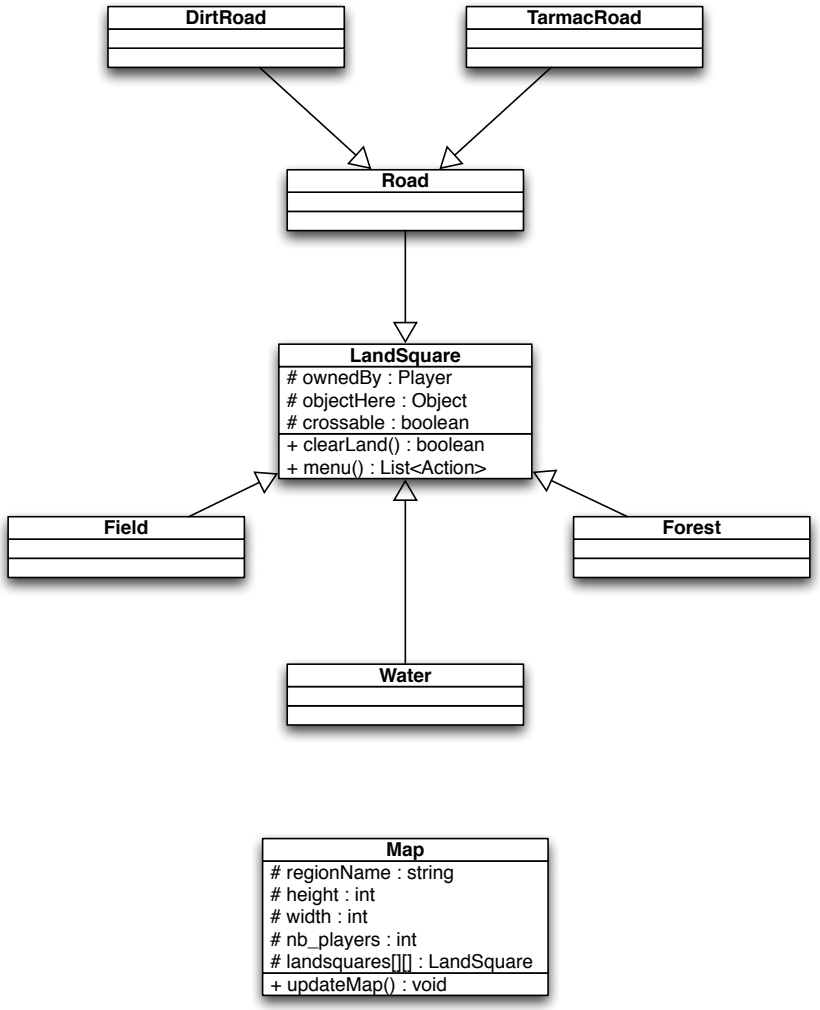

Figure B.2: UML diagram about the Map & the Lands

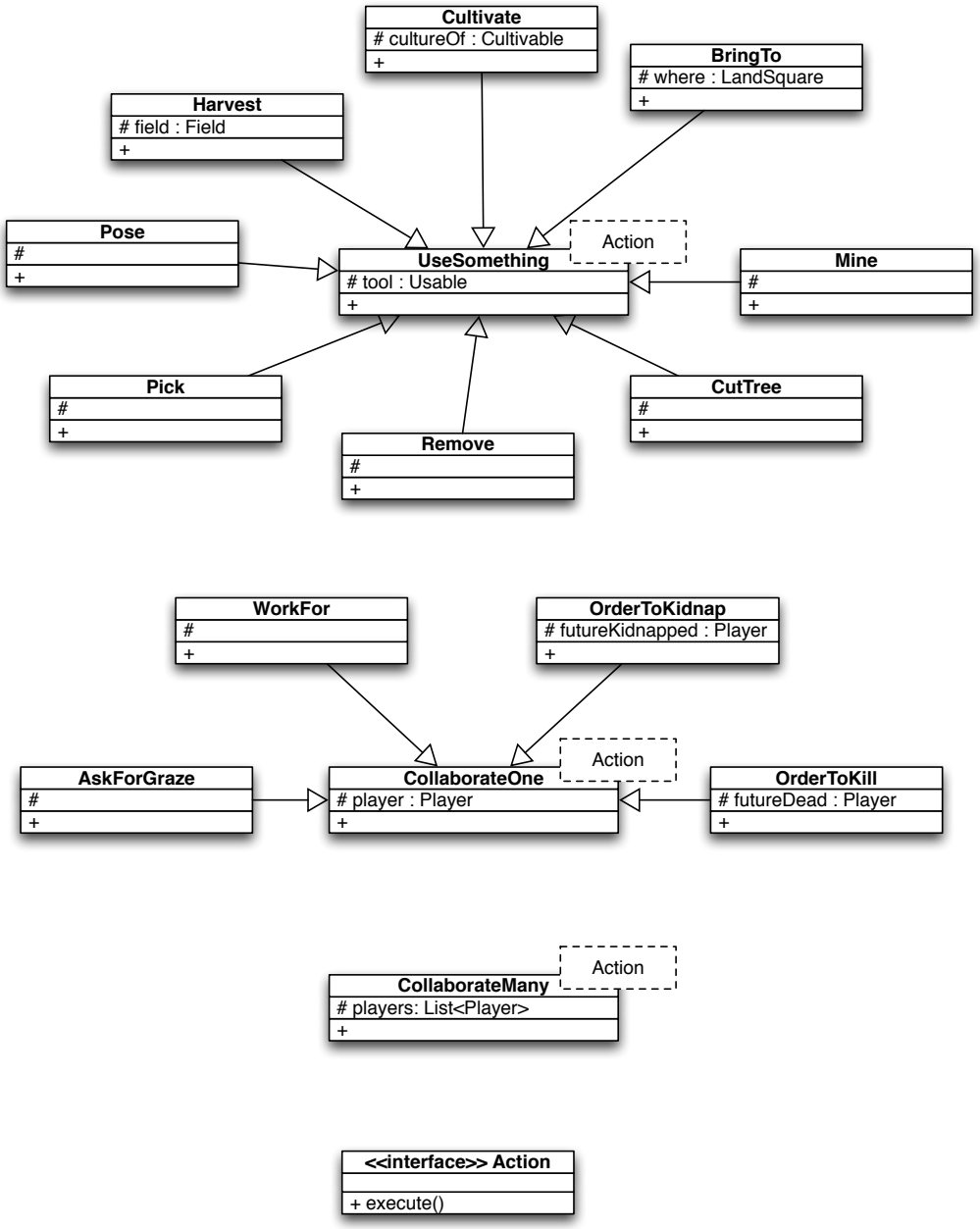

Figure B.3: UML diagram about Actions

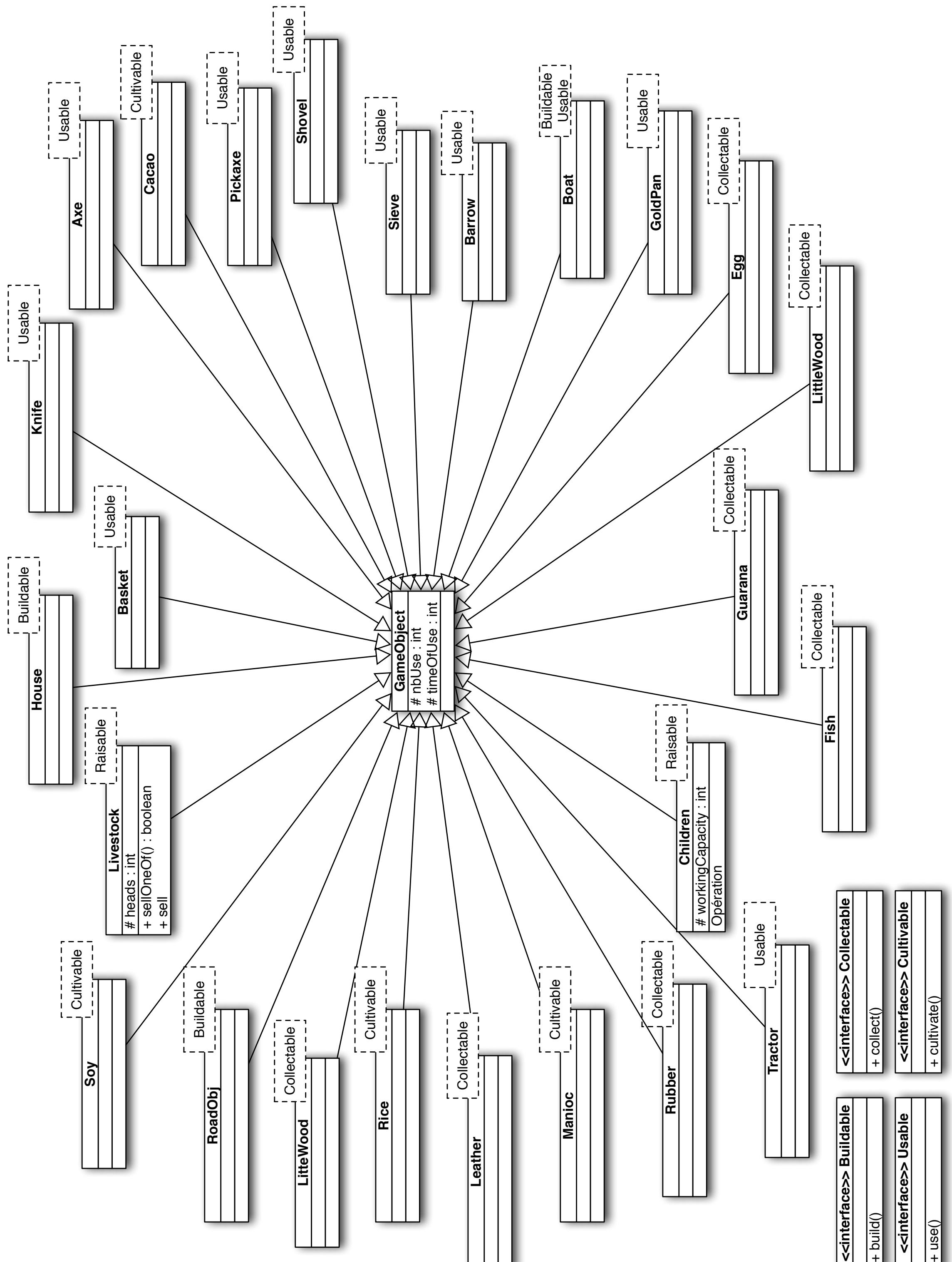

Figure B.4: UML diagram about Objects

Appendix C

Paper related to our work

In the context of the 14th International Conference on Artificial Intelligence in Education, we participated in writing a paper about Digital Natives' Learning and Teaching. It is presented in the following pages.

Digital Natives' Learning and Teaching: the Amazonia Serious Game Scenario

Ines DI LORETO[1 a], Abdelkader GOUAICH [a], Fabien HERVOUET[2], Guillaume DALICHOUX[2], Alexandre FOUCHER [2], Panupat PATRAMOL[2], Stefano Alessandro CERRI [a],
[a] *LIRMM: Laboratory of Informatics, Robotics, and Microelectronics of Montpellier*

Abstract. Prensky's talk about *Digital Natives and Digital Immigrants* is often used as a milestone for the field of study addressing this "new generation" of users. However, when talking about *Digital Natives* the general attitude is to think of them simply as students and not as perspective teachers. This is, in our opinion, a lost opportunity: their way of studying and using technologies at present will influence the way they will work in their future workplaces. When they will assume, for example, the role of teachers, what kind of tools and environments will they really need? We will try to address this problem analyzing a project we held with four master's students, for the creation of a serious game about Amazon deforestation, were they had an interesting role, at the crossroad between students and teachers.

Keywords. Co-adaptation, digital natives, serious games, user modeling, learning environments

Introduction

The intersection of Technologies and Education in general, and AI&ED in particular, have historically been marked by several profound debates - and by the corresponding research orientations- disputes, arguments, often rich of paradoxes.

In our opinion, these debates are still useful as reference points for the past but rather limited for the future, since the oppositions concern a teaching versus a learning centered approach, or a formal versus informal learning [1]. For the rest of the paper, our position will be as much as possible the one of observers of the "natural" phenomena occurring in learning (and in learners, *Digital Natives*[2]) having the goal to identify what kind of conception will be likely to be used by the same people once they will become teachers (in the deep sense of the term: at school as "real" teachers, as seniors in the workplace, as parents with their children) in the information society of the next years. Our conviction is that the real revolution within society, schooling included, will emerge not from new laws, curricula, instructional design methods, tools

1 Laboratory of Informatics, Robotics, and Microelectronics of Montpellier (LIRMM)
University Montpellier 2& National Center of Scientific Research (CNRS) 161 Rue Ada, 34392 Montpellier, France E-mail:diloreto@lirmm.fr
On leave from DICO – Dept. of Informatics and Communication, Università degli Studi di Milano, Via Comelico, 39, Italy
2 Students attending the master on informatics at the University of Montpellier 2

or professional training investments, but from the radically different approach in solving problems adopted "naturally" by *Digital Natives*, that will impact their teaching methods, at home as well as at school. In this position, we totally adhere to the authoritative views express by Eileen Scanlon and Tim O'Shea [3] and by Marc Eisenstadt [4].

As a matter of fact, nowadays *Digital Natives* are migrating into a workplace run and organized by the "previous generation" natives. Obviously, the first scenario we can think about is a workplace world dominated by *Digital Immigrants* (elders who often feel less at ease with new technologies), a place where *Digital Natives* adapt to *Digital Immigrants* technologies and working methods. However, at a certain moment *Digital Natives* will be the only generation present in this imaginary workplace. If we focus on the educational field, this means that *Digital Natives* will be the only "teachers" for the "next generation" of students. For this reason, it will be interesting, to answer to questions such as: How different are they from previous students and will be from previous teachers? Will they need different tools or environments from previous ones?

While an articulated answer to these question would need an ethnographic approach, we will try to sketch out some implications based on the evolution of a project that involved four *Digital Natives* with an interesting role, at the crossroad between student and teacher, as a preliminary source of reflections eventually to be further investigated.

In the rest of the paper, first of all we will talk about the subject of the project, a serious game for sensitizing players to the Amazon situation, that was, in our opinion, the first cause of motivation for the students' involvement as "teachers". In fact, the social impact of the project was a strong motivation for them, so that they took very seriously their educational role, as enablers for sensitization to Amazon problems. Next we will try to understand what kind of generation are our future teachers, the so called *Digital Natives*. We will then get into the core of the project, basing on students' description of the tools they used for the project management, and we will finally sketch some implication.

1. The Amazon Situation

As we have said, the topic of the project was one of the motivations that caused the students' switch from the "project developers" role to the "enablers" one. While the Amazon's deforestation problem is a complex one, is worth to spend some lines describing it for better understanding why this switch happened.

In recent years the Amazon has experienced a high levels of deforestation, including the largest loss of forest on record between 2002 and 2004. Between May 2000 and August 2006, Brazil lost nearly 150,000 square kilometers of forest—an area larger than Greece—and since 1970, over 600,000 square kilometers (232,000 square miles) of Amazon rainforest have been destroyed[5].

Several studies tried to analyze the Amazon deforestation through models of deforestation patterns[6]. Results show that heterogeneous occupation patterns of the Amazon can only be explained when combining several factors related to the organization of the productive systems, such as favorable environmental conditions and

access to local and national markets, proximity to urban centers and roads, agrarian structure, and so on. While for years studying Amazon deforestation meant only study such deforestation patterns over the land, in recent years there was an important shift in the focus. Studies such as [7] state that containing the advance of deforestation in Brazilian Amazonia requires understanding *people actions*, i.e. roles and movements of the involved actors. For example, landless migrants have significant roles in clearing the land they occupy, colonists and other small farmers are also responsible for substantial amounts of clearing, but ranchers constitute the largest component of the region's clearing. Capitalized farmers, including agribusiness for soy production, have tremendous impact in certain areas, such as Mato Grosso. Landgrabbers, or grileiros, are important in entering public land and beginning the process of deforestation and transfer of land to subsequent groups of actors, and so on.

As we have seen, land changes in Amazon are the result of a complex web of interactions between human and biophysical factors, which act over a wide range of temporal and spatial scales. An important opportunity for us (the authors) to address such a complex scenario came when, in November 2008, two conditions were met:

- a conference was held in Manaus[8] where the key actors committed to the management of scientific knowledge by means of ICT met(among them one of the authors of this paper as invited speaker); and
- 4 master's students adopted a proposal for a "student's project": a "serious games" for knowing and learning the Amazonia biodiversity. We proposed them to exploit the simulations resulted from the GEOMA sub-projects [9] - in particular Dr. Tiago Garcia de Senna Carneiro projects[3] - in order to collaboratively design and prototypically implement a serious games about human decisions (such as deforestation) in Amazonia. We also proposed them a set of tools for asynchronous (such as Wikipedia) and synchronous communication (Agora, GSD and De Visu)[10][28].

2. Digital Natives : from Vital Statistics to Common Practices

Before addressing the way the project was managed by the students, is in our opinion interesting to understand the way our students' generation address learning and technologies.

From a vital statistic point of view the so called Digital natives generation include people born between 1981 and 2000. However, we can see them under a more technological perspective. Today's 24-year-old was born in 1985 – "10 years after the first consumer computers went on sale and the same year that the breakthrough "third generation" video game, Nintendo's "Super Mario Brothers," first went to market. When this 24-year-old was a child, the basic format of instant messaging was developed. And at the time he entered kindergarten in 1990, Tim Berners-Lee "wrote a computer program" called the World Wide Web. At the dawn of high school (for him in 1999), Sean Fanning created the Napster file-sharing service. When he graduated

3 Computer Science Department Federal University of Ouro Preto (UFOP)- Brazil; Head of TerraLAB - Laboratory for Modeling and Simulation of Earth Systems, National Institute of Space Research (INPE)

from high school four years later, his gifts might have included an iPod (patented in 2002) and a camera phone (first shipped in early 2003). Our 24-year-old college career saw the rise of blogs (already two-years-old in 2000), RSS feeds (coded in 2000), Wikipedia (2001), social network sites (Friendster was launched in 2002), tagging (Del.icio.us was created in 2003), free online phone calling (Skype software was made available in 2003), podcasts (term coined in 2004), and the video explosion that has occurred as broadband internet connections become the norm in households (YouTube went live in 2005)." [11]

As we can see from this little history, *Digital natives* cannot be defined only through vital statistics, but through a set of characteristics and common experiences. As Palfrey and Gasser [12] claim, digital natives "are connected to one another by a common culture. Major aspects of their lives- social interaction, friendships, civic activities – are mediated by digital technologies. And they've never known any other way of life.". We can then say that digital natives are characterized by a set of common practices, including the amount of time they spend using digital technologies, rather than by their age.

From a "learning" point of view: "*Digital Natives* are used to receive information really fast. They like to parallel process and multi-task. They prefer their graphics *before* their text rather than the opposite. They prefer random access (like hypertext). They function best when networked. They thrive on instant gratification and frequent rewards." [2]. Moreover, they like to "learn by doing" (doing is more important than knowing, and learning is accomplished through trial and error) and they consider human interaction an essential part in the learning process[13]. Furthermore, Studies at the Pew Internet & American Life Project [14] show that virtually all college students play video, computer or Internet games and 73% of teens do so. As a result, for example, they become accustomed with a style of learning that takes place informally[15].

Finally, if we look at their views on technology:

The definition of **technology is not confined to computers or the Internet**. Technology is viewed as any electronically based application or piece of equipment that meets a need for access to information or communication. For *Digital Natives*, technologies that are still considered transformative by their parents' and grandparents' standards (for example, instant messaging) are a basic part of their everyday lives. For them, technology is simply "what's new".

Customization is central to their definition of technology. Technology is something that adapts to their needs, not something that requires them to change. For them customization is the ability to adapt technology to meet individual needs, rather than vice versa [16]

The *use of* available technologies is precisely what we found very interesting in the way our students approached a project that, paradoxically, is clearly centered on the *production of* new technologies. In the next paragraphs we will address this aspect in order to find what will be the "future generation" of teachers' approach to technologies.

3. The Amazonia Serious Game Project: the Story so Far

As we have said, the complexity of the Amazon situation is based on land changes and at the same time on people behavior. In order to address such a complexity we (as teachers) decided for the creation of a serious game. In the purposes of the project, such a mean will allows us to achieve two major goals: sensitize people (perspective users of the game) to Amazon problems, and model users' behaviors and believes through their own taken actions (the first may be considered a pedagogical goal, the second one a kind of experiment for group modeling). In the first part of this section we will detail our motivation for the choice of the serious game as "pedagogical" mean, in a general way. A description of the generated serious game would be an interesting element of analysis for understanding better why the students take very seriously their educational role, i.e. their role as enablers for sensitization (in fact, students also developed the history and worked on the game design). However, for our purposes - understanding *Digital Natives* approach to technologies in order to identify what they will need when they will become teachers – is more interesting to analyze the way they approached the others technologies they used. In the second part of this section, we will show what "really happened" when students started to work for the project.

2.1 Serious Games and Learning (or, old generation, very academic reasons for the approach)

The serious games movement started with the U.S. Army's release of the video game *America's Army* in 2002 [17][18]. The same year the Woodrow Wilson Center for International Scholar in Washington, D.C. founded *the Serious Games Initiative*, and the term "serious games" became widespread[19].

The term itself is nowadays established, but there is no current single definition of the concept. Serious games usually refer to games used for training, advertising, simulation, or education, that are designed to run on personal computers or video game consoles.

Among the various (more or less) formal definitions for serious games we like Zyda's [20, p.26] words:

"Serious game: a mental contest, played with a computer in accordance with specific rules, that uses entertainment to further government or corporate training, education, health, public policy, and strategic communication objectives."

When comparing serious games with just computer games, Zyda argues that serious games have more than just story, art, and software. It is the addition of *pedagogy* activities that makes games serious. Now, we want to take his *pedagogy* term in the Greek original sense: paidagōgeō; from παιδ *paid*: child and άγω *ágō*: lead; literally, "to lead the child". So, in our vision, serious games are not simply a way to "instruct " about something (a skill or a competence) but a way to convey knowledge within a motivationally rich context (enabling the self construction of knowledge but not excluding the acquisition of scientifically correct information), and, why not, sensitize people about potential consequences of behavioral choices performed while playing.

If we can think serious games in this way, the question of interest concerns the claimed positive effects of such games, or of their applications from related and

sometimes overlapping areas such as e-learning, edutainment, game-based learning, and digital game-based learning.

Corti [21], considers that Game-based learning (GBL) and serious games have the potential of improving training activities and initiatives by virtue of, e.g., their engagement, motivation, role playing, and repeatability: failed strategies etc. can be modified and tried again. Digital game-based learning (DGBL) is closely related to GBL, with the additional restriction that it concerns digital games. Analyses have been conducted over the years, consistently showing that games promote learning [22][23].

DGBL is, following Prensky [24] ideas, based on two key premises; firstly, the thinking patterns of learners today have changed, that is, today's students are 'native speakers' in the language of digital media. Secondly, this generation has experienced a radically new form of computer and video game play, and "this new form of entertainment has shaped their preferences and abilities and offers an enormous potential for their learning, both as children and as adults" (ibid., p. 6).

So, we posed ourselves the same question as Squire et al.[25]: how educational technologists will respond to the "digital native speakers", i.e., "a generation of students who, raised on interactive games, expect the same kinds of interactive experiences from their educational media?" (p.34).

As we have mentioned, our answer was linked to our idea of pedagogy: we will convey knowledge through a serious game, and the game will not be an educational game in the classical sense of edutainment: it will talk the *Digital Natives* language.

3.2 Bring the project to light (or, what really happened when digital natives become enablers/ pedagogues)

Towards the end of the project we asked our students to describe what kind of tools they used for managing their work (as we have said in par.1 some were proposed by the teachers, but, as a matter of fact, most of them were chosen by the students). They produced a report describing the tools but also why they find them useful (or not). In the rest of this section we present a synthesis of their report, that will be analyzed in the last section in order to identify a few properties of our experiment supporting or refusing convictions about the needs for tools in the years to come.

Flash Meeting

The first software they were asked to interact with was Flash meeting (FM)[26], an Open University's[27] project that implements a set of collaborative tools to manage real time video meetings with participants from all over the world.

From a technological point of view, students found very interesting the simple way used to access the meeting (a web browser with Adobe Flash and an internet connection). They also liked the easy way to manage "speaking turns" through queues, a functionality that is not available in most Instant Messaging software.

From an organizational point of view, they found compelling that the meeting has a limited duration (nearly one hour and a half in each case), because this forced them to effectively prepare it, thing that they never did for their "informal meetings". Finally, they considered the possibility to record the meeting the most important feature of this tool "because when you make a meeting speaking English with participants from

different nationalities, it is very important to listen a few times more the conference, to make sure you well understood everything". [see Fig.1]

Their final comment on FM was that "It is full of interesting features to help the collaboration and to forget that the meeting is virtual".

Figure 1. A screenshot from FlashMeeting.

Face to face meetings

The students also participated in (more or less formal) face to face meetings. The "formal" meetings were held in the LIRMM laboratory -with the teacher presence- while the "informal" ones were held at one student's home. They considered the first kind of meeting as an opportunity to summarize and discuss the ideas they had since the last meeting (for example they reviewed the needs and the roadmap of the project). In their words "It allows to give everyone a global vision of the project, and especially to motivate the troops".

Meetings at home were their informal way to manage the project. It's interesting to note that nobody told them to do this kind of meeting (nor to add others tools). Simply, where they felt a gap in communication or in project management, they instinctively filled it. They liked this kind of meetings because "there are more ideas that spring out, and you are allowed to think in a more easy way". Moreover, the process of stay together and eat together after the meeting contributed to improve, as they said, the cohesion of the work group, because it allowed them to know each other better.

Skype

In students analysis, Skype's main drawback, compared to tools like Flash Meeting, is linked to the installation and registration process. If you want to communicate with someone, he/she has to download and install the software, and finally register a Skype account just to be able to receive your call or message (as we have seen FM uses a light web client and only the booker needs to register). They also found that the impossibility to record the meeting was a problem for their "professional" use of the tool.

However, they found also important that everyone in Skype was able speak at the same time: "you can interact more easily and quickly with your interlocutor, and this is

a way to be more efficient". Their main interest in Skype was that they were able to use it in an instant way: if they were connected and they wanted to know what the others think about an idea, they just had to launch the Skype call (it's very interesting to note that this implies that they were practically always connected to the application).

Agora GSD

Agora[28] is a platform which aims to deploy collaborative services among virtual communities on the internet. The Agora platform includes two collaborative services: The Grid Shared Desktop (GSD), a service for sharing any application , and De Visu, a video-conference service. For the project, the students experienced only the Shared Desktop.

Students had different opinions over this tool. From one hand they found it interesting because it allowed them to communicate and to collaborate thorough the shared desktop. In an occasion, when there was the need to show in a shared way the result of their work and they weren't able to use Agora, one of them claimed that he would have really appreciated to be able to use it, exactly for the shared desktop idea. However, some of them found that "in face of this tool, everybody is not equal. The simple fact that we needed an introduction was not a good thing. What will happen to non-informatics people?" (they are referring to a demo that was held to show them the shared desktop functioning). We are pretty sure that, if the project really needed a way to show in a guided, shared way the results, they will have found a tool adequate to manage it.

Wiki

In students' opinion, for the project the wiki[30] was not used as a communication tool, but as a tool for future collaboration. In their words "Maybe this project will go on after we will be forced to stop it, with other students next years. If so, the wiki is a huge amount of knowledge, a trace of what we thought, discussed and decided about it." They really liked the idea to let a persistent trace of their work, and the direct evidence of this statement is that they spent a significant amount of time not only "filling" the wiki's pages, but also "personalizing" it, through images and little "hackings" in order to adapt the wiki to their needs (this match with Roberts' founding about *Digital Natives* approach to customization, see par.2) .

Summarizing, for them the wiki provided a well structured and centralized outcome, an organized persistent trace of their knowledge and their decision processes.

Emails

When they analyzed this communication tool they found the already well known advantages and disadvantages of asynchronous communication. You don't need to set an appointment to send a mail, and you can send information in a message whenever you want. Messages are almost sent to everyone in the group, so there is a persistence of these messages. However, the way we used mails (through simple clients and not based on a repository) generated some versioning problem with the attached files.
This is why they searched, found, and used alternative tools for managing the problem.

SVN (GoogleCode)

Google Code Subversion (SVN) [29] is a mean to manage concurrent versions of files on large projects. Students decided to use it because it allowed them to work at the same time without fearing data loss (you can easily restore any old version of a file

uploaded before). An interesting remark over this tool was “we are getting used to it, because it is a very common tool in development, and widely used”.

DropBox

DropBox[31] is a software that allows you to create a shared directory which could be easily synchronized between collaborators, just using an Internet connection. The students adopted it spontaneously (as they did for Skype, SVN, and the informal face to face meetings), because they liked it’s “collaboration oriented” space.

In fact, everyone can access the files they put in the shared directory at anytime. However they found a lack in the potentiality of the tool. In fact, every time they added a file they had to sent an e-mail to announce it. “It wasn’t a communication tool because each time we shared a new file with DropBox, we have to send a mail to everybody to inform them of its existence, but it really dealt with knowledge sharing because everyone of us may decide to create a new file and to put some knowledge into it...”

4. Lessons Learned

On the basis of the above described student report, in the rest of the paper we will try to outline some interesting aspects that emerged when we tried to abstract from it:

The fading of real/virtual opposition: First of all, when we asked them to list all the tools they used for communication, they naturally added face to face meetings. While their comment on Flash Meeting let think otherwise (“It is full of interesting features to help the collaboration and to forget that the meeting is virtual”) it is our opinion that this is a concrete sign of the fading of the traditional real/virtual opposition (meeting with pizza and meeting with Skype are considered at the same level).

This in our opinion means that future “teachers” will easily use “virtual” and “real” means for their purposes, so they will need environments that will allow them to switch seamlessly between the two.

A love for persistence: It’s interesting to notice that the students searched tools able to offer persistence (e.g., the files repository and the wiki) to their work. Moreover, they liked the permanence of video meetings - where virtual meeting became real also trough its permanence and repetition, and therefore through its location in the whole history of the project.

This in our opinion means that future “teachers” will need environments able to support persistence (and obviously the associated retrieval).

MashUps as a natural instinct: This is in our opinion the most interesting aspect. As we have noticed, if students found a gap between the tools they were required to use and their needs, they filled it instinctively (as most young people do in internet trough mash-up[4]).

In our opinion this implies that this generation of perspective “teachers” - that is already used to mash-ups - will not need pre-packed tools but, more and more, environments were the current and future tools will be accessed and made available for use.

The new Media Grammar for informal learning: Some of the tools our students listed addressed very particular computer scientist problems (e.g. SVN and DropBox addressed the problem of managing file versioning). However, if we try to abstract, we can note that for them creating a project was all about communication, knowledge creation and knowledge management/sharing. In their report, they never mentioned the tools for "creating" the game. This may seems at first a little bit strange, because nobody instructed them on how to build a game. However, they made instinctively interesting suggestion about how to create it, because, for them, games are experienced tools. We can note the same concerning Skype: most of the tools they used were not "learned" at school, but in an informal way. Moreover their use of communication tools (e.g. Skype for a quick call about an idea) let really think about peer supported learning.

This in our opinion means that future "teachers", will be literate on many different "new" media, unlike the old generations , and will give much more importance to collaboration, social and informal learning (learning from the others, learning from the context, learning by doing) than to the "tools" they are using (learning at school from books, following curricula).

5. Conclusions

As we have already said, we didn't use an ethnographic approach, so this paper has no pretenses to give any definitive response to the formulated questions . However we think that in the development of the project some interesting elements emerged that will be worth of further in-depth examination.

Sensitization about Amazon deforestation was, for the students-enablers, the primary goal of this project. The social impact of the project was for them a strong motivation so that they took very seriously their educational role, as enablers for sensitization to Amazon problems. We think that the creation of a (serious)game -a well known medium for them - had the advantage to let them focus on management and development issue, rather than on the medium grammar (as it often happens with other kind of applications that are given to teachers for their use). We know that our students were a very particular kind of students (high level of abilities in informatics, high motivation, and so on) but it is our opinion that future teachers will be the same: considering tools (including perhaps classical concepts in curricula) as commodities and not as the scope of what they are doing, and, in the best of possible worlds, highly motivated. Moreover, we may assume that as teachers they will focus on abstracting and generalizing the instance of the solving problem process itself to new contexts. rather than in the art of constructing new tools and applications for solving new problems, because they will assume their students will need the same as what they needed themselves when they were students.

References

[1] J.Sefton-Green, Literature review in informal learning with technology outside school, *NESTA Futurelab Research Report 7*, 2004. Retrieved on April 7, 2009 from http://www.nestafuturelab.org/research/reviews/07_01.htm

[2] M. Prensky, Digital Natives, Digital Immigrants, October, 2001. Retrieved on April 7, 2009 from http://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf
[3] E. Scanlon, T. O'Shea, New educational technology models for social and personal computing, Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007), icalt, pp.11-14, 2007
[4] M. Eisenstadt, Does Elearning Have To Be So Awful? (Time to Mashup or Shutup). Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007), icalt, pp.6-10, 2007
[5] Mongabay website: http://www.mongabay.com/brazil.html
[6] P. D. AGUIAR, G. CÂMARA, M. I. S. ESCADA, Spatial statistical analysis of land-use determinants in the Brazilian Amazonia: Exploring intra-regional heterogeneity. *Ecological Modelling*, v. 209, p. 169–188, 2007.
[7] P.M. Fearnside, The Roles and Movement of Actors in the Deforestation of Brazilian Amazonia in *Ecology and Society* 13(1): 23. (2008)
[8] Amazonia em perspectiva Conference website: http://lba.inpa.gov.br/IW_SKM_and_ICT/
[9] GEOMA Projects website: http://www.geoma.lncc.br/
[10] P. Dugenie, C. Jonquet, and S. A. Cerri, "The Principle of Immanence in GRID-Multiagent Integrated Systems," in 4th International Workshop On Agents and Web Services Merging in Distributed Environments, AWeSOMe'08, OTM 2008 Workshops. vol. 5333, R. Meersman, Z. Tari, and P. Herrero, Eds. Monterey, Mexico: Springer-Verlag, 2008, pp. 98-107.
[11] L. Rainie, Digital Natives Invade the Workplace. Pew Internet & American Life Project: Washington, DC, 2006. Retrieved on April 7, 2009 from: http://pewresearch.org/pubs/70/digital-natives-invade-the-workplace
[12] J.Palfrey, U. Gasser, Born Digital: Understanding the First General of Digital Natives. New York: Basic Books, 2008
[13] B. McNeely, Using technology as a learning tool, not just a cool new thing. In D. Oblinger & J. Oblinger (Eds.), Educating the Net Generation, pp. 4.1-4.10, Boulder, CO: EDUCAUSE, 2005. Retrieved on April 7, 2009 from http://www.educause.edu/educatingthenetgen
[14] A. Lenhart, et al.: Teens, Video Games, and Civics, Pew Internet & American Life Project (2008). Retrieved on April 7, 2009 from: http://www.pewinternet.org/Reports/2008/Teens-Video-Games-and-Civics.aspx
[15] J. Kirriemuir, A. McFarlane, Literature review in games and learning. Report 8. Bristol. Nesta Futurelab , 2004
[16] G. R. Roberts, Technology and learning expectations of the Net generation. In D. Oblinger and J. Oblinger, eds. ducating the Net generation, 2005. Retrieved on April 7, 2009 from: http://www.educause.edu/TechnologyandLearningExpectationsoftheNetGeneration/6056
[17] The American's Army game website: http://www.americasarmy.com
[18] J. Gudmundsen, Movement aims to get serious about games. *USA Today*, 5/19/2006 Retrieved on April 7, 2009 from: www.usatoday.com/tech/gaming/2006-05-19-serious-games_x.htm
[19] The Serious Games Initiative website: http://www.seriousgames.org/index2.html
[20] M. Zyda, From visual simulation to virtual reality to games. *Computer*, 38(9), 25-32, 2005
[21] K. Corti, Games-based Learning; a serious business application. PIXELearning Limited, 2006. [www.pixelearning.com/docs/games_basedlearning_pixelearning.pdf]
[22] M. Szczurek, Meta-analysis of simulation games effectiveness for cognitive learning, 1982. Dissertation Abstracts International, 43(04), 1031A.
[23] R. Van Eck, Digital game-based learning: It's not just the digital natives who are restless. *EDUCAUSE Review,* 41(2),16–30, 2006
[24] M. Prensky, *Don't bother me mom, I'm learning!* Paragon House Publishers, 2006 www.marcprensky.com/writing/Prensky%20-%20Ch1-Digital%20Game-Based%20Learning.pdf
[25] K. Squire, & C. Steinkuehler, Meet the gamers. *Library Journal*, 130(7), 38–42, 2005
[26] Flash meeting website: http://flashmeeting.open.ac.uk/
[27] Open university website: http://www.open.ac.uk/
[28] Agora website: http://agora.lirmm.fr
[29] Google code website : http://code.google.com/
[30] Amazonia serious game wiki : http://agora.lirmm.fr/asg09/
[31] Drop box website: http://www.getdropbox.com/