# Learn how to count with MAO

DALLO Anthony, EDDASSER Omar, CANZERINI Mathieu

April 1, 2012

L3 MIAGE

# Contents

# Part I
# Introduction

The aim of our subject was to create software for children. With this software children should learn how to count.

Before starting the subject, we asked one teacher the method she uses to teach children counting. She told us about using hands, fingers and the dice constellation. So, we decided to be original and focus on hands and fingers as they are less used than the dice constellation. We also decided to turn the software in a game because its the best way to teach children something.

In this report we will explain how we analyzed the subject firstly, then we will describe the interface and finally we will briefly talk about technologies we used.

# Part II
# Analyze

Before programming our software we made an in-depth analyze of the subject. The software is for five years children so at their age, they only starting to read and count. Thats why we had to adapt the software by using voices instead of text.

Our software is made MAO, which is an acronym for our firsts letters of our surnames (Mathieu  Anthony  Omar). We chose this name because its easier to remember it.

As the software is for children they need heroes to help them during the game. Then we chose different heroes because everyone has different feelings. We started by Sponge Bob, Cars and Madagascars penguin. But there was some problem; there werent anything for girls, as all of them are liked to be for boys. Then we did some changes and we chose Sponge Bob (which is for boys), Bloom from Winx (which is for girls) and the Puss in Boots (which is kind of neutral).

After the heroes choice, we had to decide what sort of game we should program. By thinking of this we had to see:

- how many levels we should put,
- how many times the child could do wrong things,
- how we should help him to improve his knowledge,
- how we should manage the levels progress.

So, we decided to program three levels ordered by difficulty order. In the first level we try to teach him how to count with his fingers. In the second level we teach him how numbers are represented and in the last level we teach him how to add numbers.

According the fact the user is a child, we had to give him a reward after each good choice he does. Thats why we decided to show him an episode of the hero at the end of the level. Plus, we show him that he is right by a sound after each choice. We also show him that he is wrong with the same process. If the child does several wrong choices he will be led to the previous level. If he goes wrong, we also help him at the level two and tree by showing hands and fingers from the first level next to numbers. So, we have different types of users. The level one is for beginners, the second is for novices, and the third is for competent users.

On the level selections screen, we also chose to add the hero. By doing this the hero will show the child which level he can do or not. We think that it makes the software more alive and less impersonal. Plus, the child is guided during all the game.

We spoke about sounds and voices. In fact weve added several sounds for different cases. By doing this weve tried to make the game not repetitive. So, if the child makes a mistake he will have one sound, which will be different for two repetitive mistakes, then the same for three. After that well delete three false answers associated with different sounds.

Furthermore, we chose to integrate some lessons to help the child studying numbers. Lessons are of course vocal lessons since we keep thinking that at five years old the child is not able to read everything.

Finally our software can be turned into English and then if the child is enough strong with numbers in French, he could study numbers in English. In this version, we only had to replace French sounds by English sounds. It makes our software more efficient.

Now we are going to present the interface in detail.

# Part III
# Interface

The mainly point in our subject is the fact that users of the application are five years old. Thats why we have made different choices such as:

- do not use any text in our application
- use also sound effects
- try to explain why the user have made a mistake and not only notice it
- use several meaningful colors except the red which can be assimilated to the danger

Then, we have made some templates of our application.

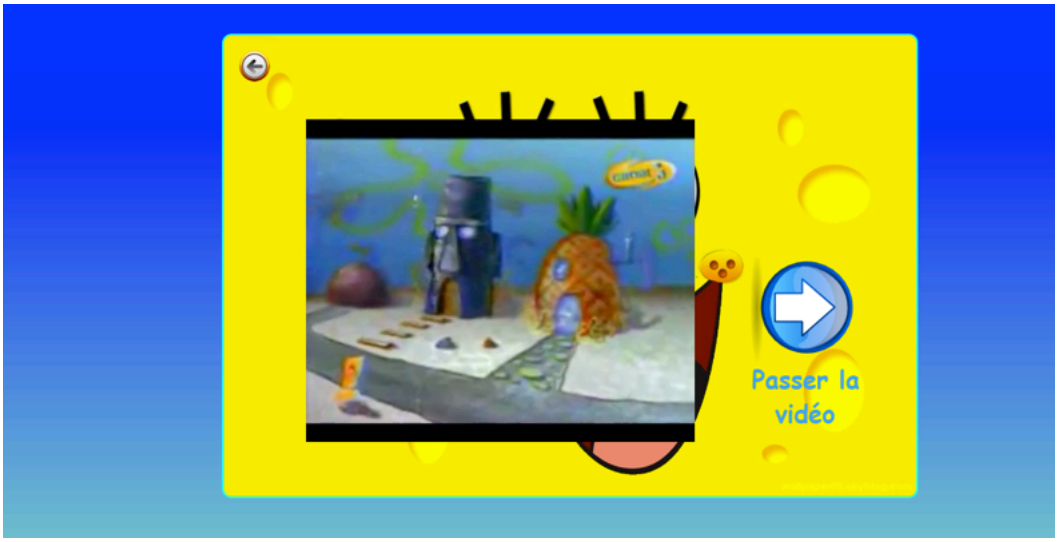
First the Homepage: We have chosen three different characters, which define three environments in our application:

- Sponge Bob
- Bloom (Winx)
- The Puss in Boots

The user can choose with which character he wants to improve his mental arithmetic.

When the user hovers on one of the characters, the background changes to a background with colors about the character:

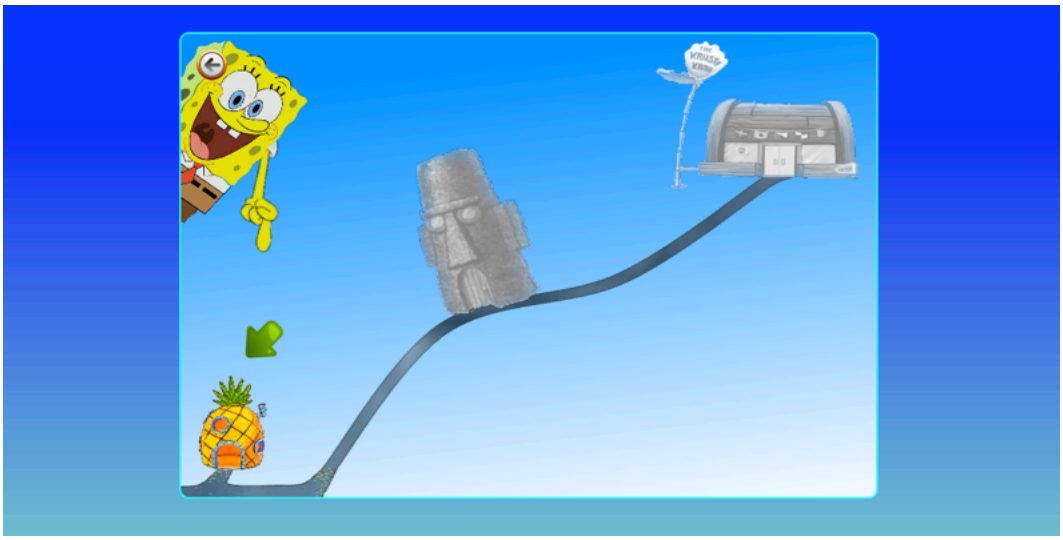

After, when the user clicks on the character, we arrive in another screen with a video. The user can jump to the next screen by clicking on the big arrow or can watch the video:



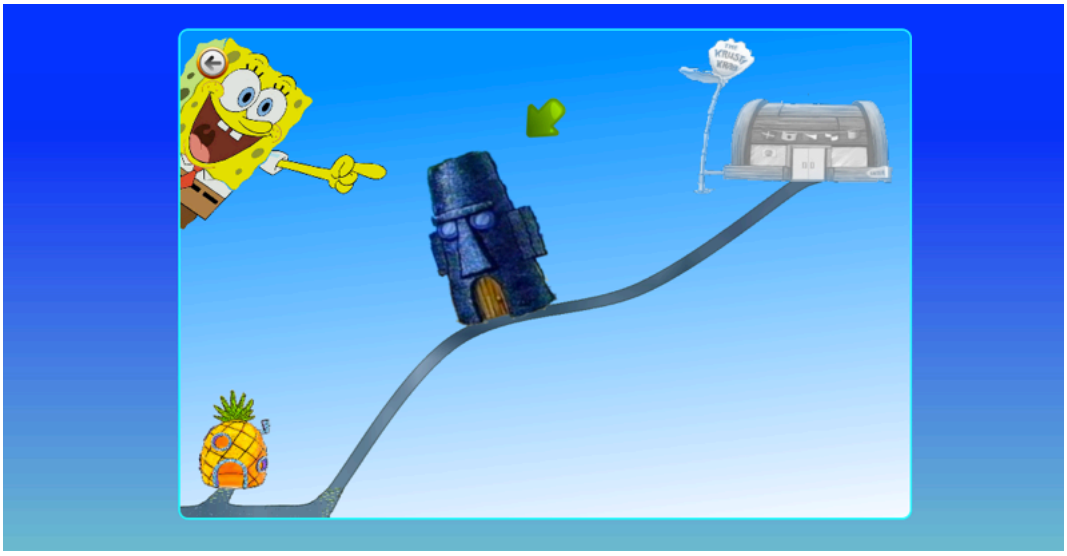

We can see that our screens are very simple. Here we have only two buttons: the small arrow to go back and the big arrow to jump to the next screen. Finally there is a frame where the video is displayed.

After the screen we arrive in the screen of the levels.

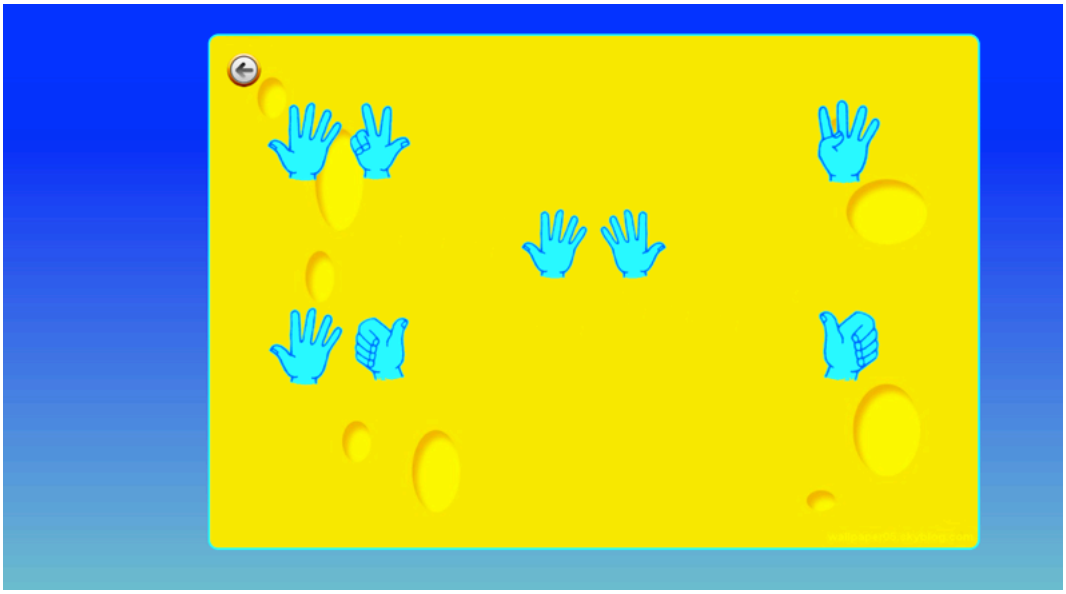
There are three levels per character. The user can play the level two only if he has passed the level one and in the same way for the level three.



We have disabled levels that the user cant play with (here the level 2 and the level 3). Furthermore, the character (Bob) notices where the user has to click to continue with his arm and his eyes. When the user has passed the first level, the character notices the second level:



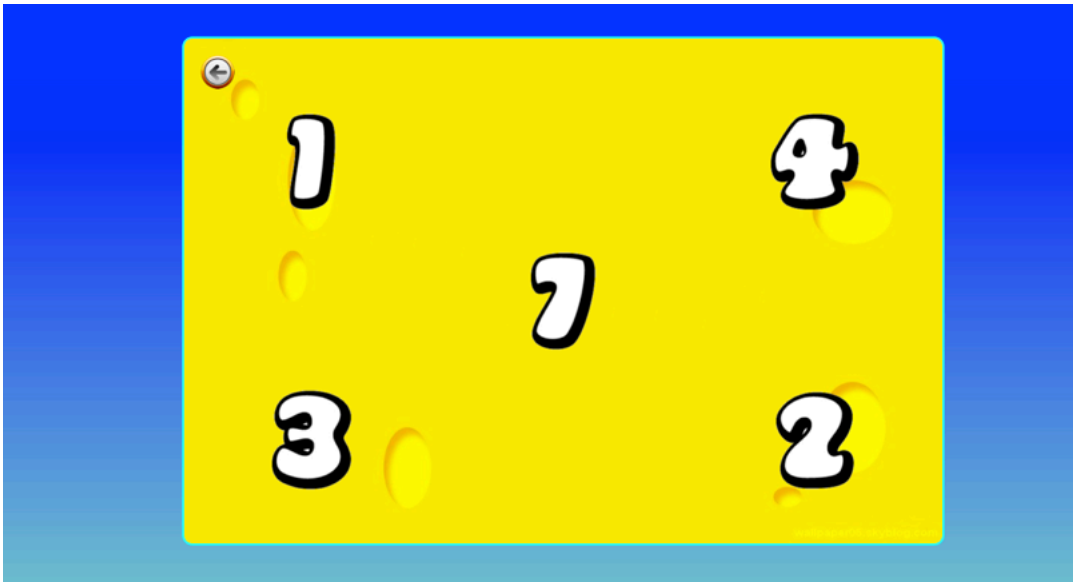
Concerning the game itself. We have decided to improve the user knowledge about numbers. Thats why in the first level we use hand pictures:



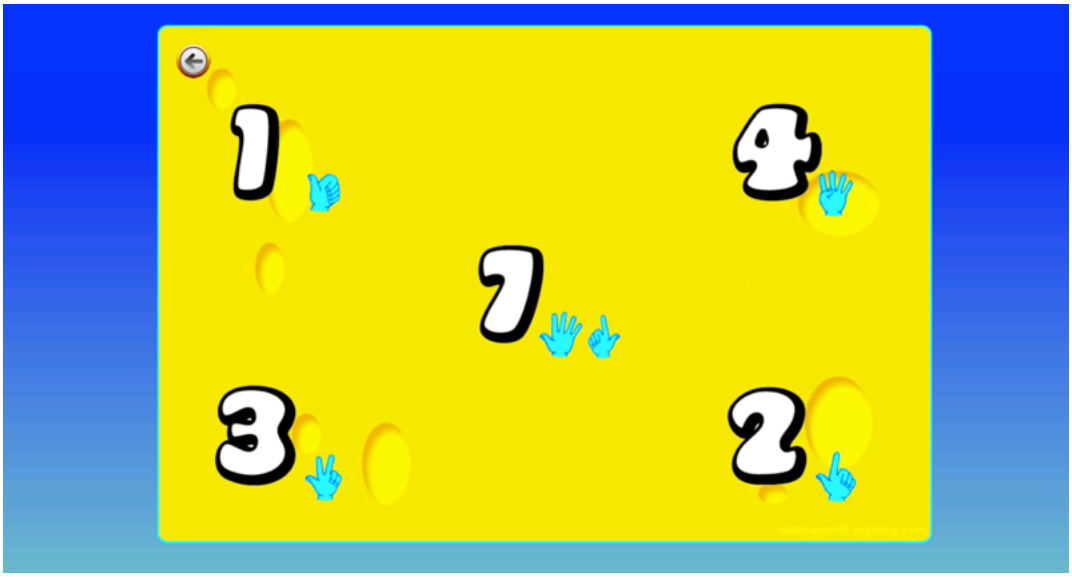
If the user makes three mistakes, we remove two propositions:

When the user has passed all questions, he can watch the continuation of the video and can play the second level.

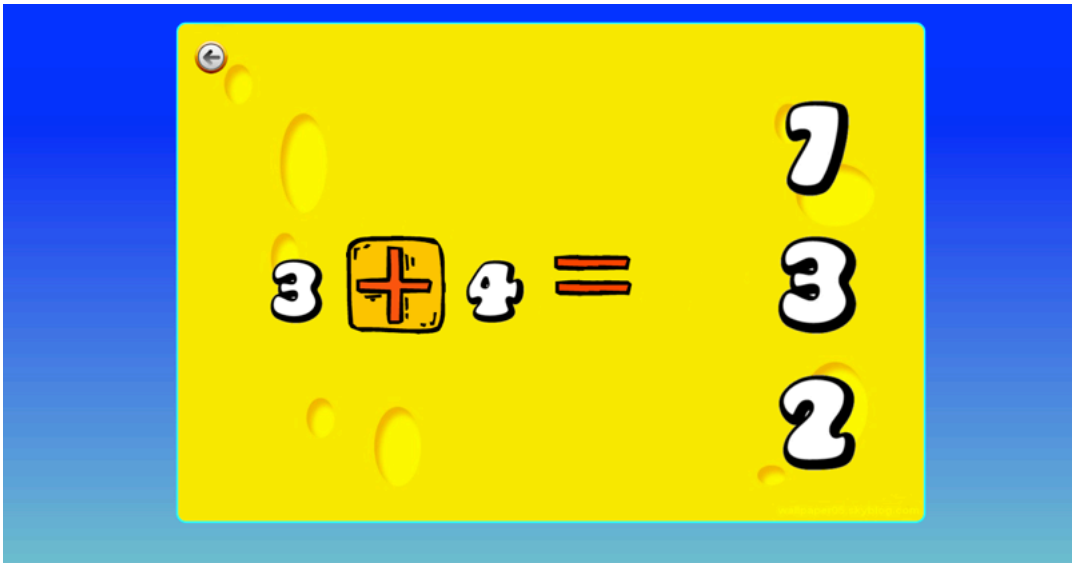
In the second level, we advise the user to recognize the number.




If the user makes three mistakes, we add the hand picture near the number to help the child to recognize the number:

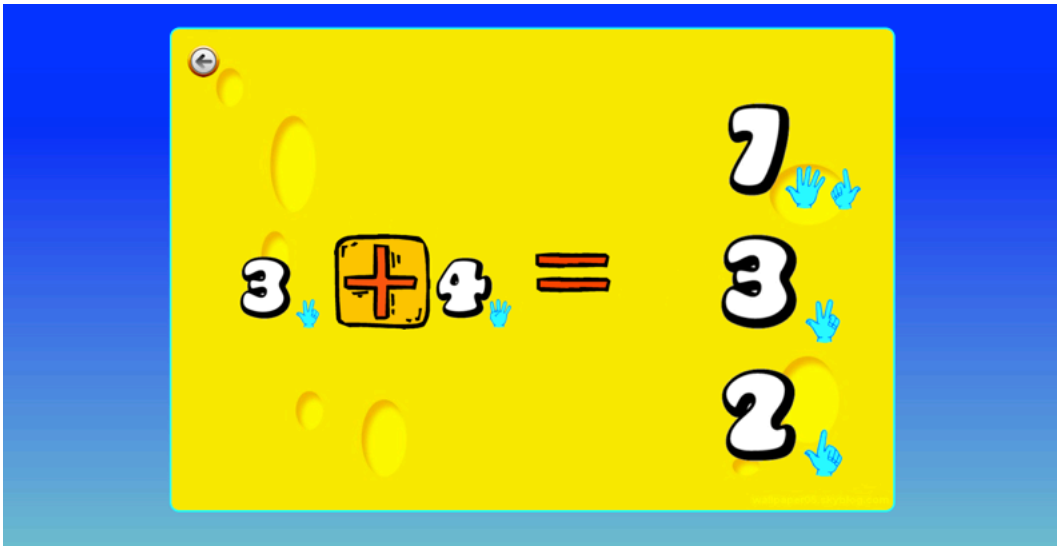



If the child makes too many mistakes, he is automatically led to the previous level. Concerning the final level, the user has to make some simple additions:




In the same way as the previous level, if the user makes three mistakes, we add the hand picture near the number to help the child to recognize the

number and count with his fingers:




If the child makes too many mistakes, he is automatically led to the previous level. When the child has passed all the levels, he can look the entire video.

Now we are going to present tools we used.

# Part IV
# Tools used

The programming languages we used to program our software are: Hypertext Preprocessor (PHP), Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript and Asynchronous JavaScript (Ajax).


PHP is used in the server part, it generates the user view which is in HTML/CSS/JavaScript. We have chosen them because they are adapted to our project and we have lot of experience in each of them. These languages make our application work in many sort of device (for example: computer, tab or smartphone).

We also chose these languages because our software is then a layer. In fact, we can add more heroes simply or we can add more languages just by changing sounds.

# Part V
# Conclusion

This project has been very interesting in several ways. In fact, it has permitted us to reuse notions we studied in class.

We also worked on a subject we liked: teach children how to count, and with technologies we liked: web technologies.

Then we had to work in team of three in real conditions and we saw all steps of a project, which are: searching information about the domain, draw a layout of the project, program the software and make tests with users.