Multimodal Tamagotchi

Spring semester project

Professor : Denis Lalanne
Students : Simone Raimondi, Marcel Grosjean

1 Introduction

Tamagotchi, this names brings up a lot of memories to the people who was born in the '90s. Who does not remember the little egged-shaped computer with three buttons that sold over 76 million units from his launch in 1996?

It was exactly this small device that brought us to the choice of our project for the course of Future User Interface: recreate the Tamagotchi game but instead of just providing three buttons to interact with it, try to make use of newer, more modern and more immersive user interfaces. The idea for this project also came from the suggestion of the professor to try to work on an application that uses multimodal interfaces and also provides a feeling of empathy with it.

Figure 1 : First tamagotchi marketed

2 Project description

At the beginning the idea was to try to recreate a sort of animal companion with which you could interact by using voice, gestures and it was able to understand your emotion and react accordingly. The idea looked very nice to us, but we soon found out that we had to readjust what we wanted to do due to different reasons.

Providing a software that is able to learn how you express emotions and react to them can be very challenging. It requires to implement some machine learning algorithms that must be trained in order to give the best possible and most natural feedback to the user. This would have taken too much time and so we relied on a JS library that already tries to do that for us. The library we decide to use can be found here: https://github.com/auduno/clmtrackr. It already provides some basic emotion recognition with a pretty good accuracy. Sometime it does not work perfectly but we thought that it would be enough for our needs.

The biggest difficulties arose when we thought about adding the gestures recognition to our system. By doing a bit of researches, we found out that things get really tricky when we speak about gesture recognition. The biggest problem was that, since we wanted to provide an interface that was as natural as possible, we did not want to rely on a custom glow with sensors, which would have greatly improved the accuracy of our gestures recognition but also it would had make the system much less natural to the user. After evaluating the pros and cons of adding the gestures

recognition we decided to skip them because they would have taken way too much time to implement.

The last thing we wanted in our project was to add voice recognition. Voice is one of the most natural way of interacting with a companion. Who does not speak to his, even when you know that he's probably understanding just a few of the words that you are saying? Probably most of the owner of a pet do it anyway. As for gestures, voice recognition is not something trivial to implement, especially for all the external source of noise that we can have, the accent of the person or difficulties in pronouncing some words. But as explained before, voice was something that we absolutely wanted in our project, so we tried to find a library that does that for us. The final decision was to use annayang https://www.talater.com/annyang. Even if the first sentence on the website is "SpeechRecognition that just works", this is not completely true. In fact, we had to try some way around to make the voice recognition work as good as we could and still it's not perfect. But in the end we were still pretty happy with the result.

3 Gameplay

The properties of the original Tamagotchi are the following:

- Age: 1 year older every day
- Name: Indicated at birth and never change, 5 letter max
- Gender: Indicated at birth
- Weight: Must make sure that he doesn't get overweight

It has several states during its lifetime. At the begin it's an egg, and then it will turn into a baby and if we keep him healthy, it will grow up into a child, adolescent and in the end as an adult. Moreover, playing and feeding him are essential to keep it in good health, otherwise it might get sick and die. The Tamagotchi usually at sleep and you can wake him up turning the lights on

It also has several levels which represents different condition of it and the higher they are, the better it is:

- Appetite: can give FOOD or SNACK. The user must control the level of appetite before feeding otherwise it will gain weight and get sick
- Happiness: Make it happy by DANCING or JUMPING with it
- Education: Can PUNISH or PRAISE it
- Sleep: You can turn ON/OFF lights to make it fall asleep or wake it up

Everyday it becomes one year older and you must try yo keep it in good health and to not feed it too much otherwise it will die.

Tamagotchi can also get sick if you don't take proper care of it, but you can heal it with some medicine in that case.

4 Multimodal interaction with the Tamagotchi

In this section we explain how we implemented the different modalities in the gameplay of the Tamagotchi.

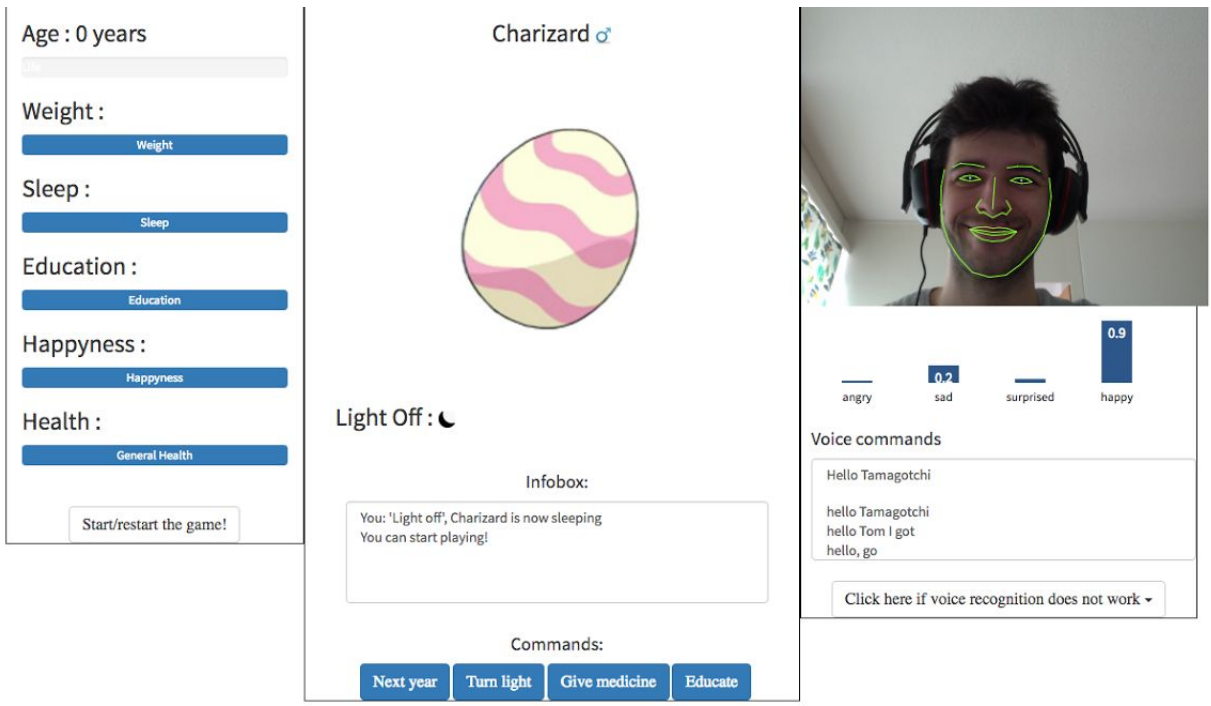
Due to the difficulties that arose in having a precise interaction by only using voice and emotion recognition, we had to make some simplifications to the system and rely on the mouse input to make sure that we are able to give all the commands that we want.

Voice recognition is mostly used to give the orders. For example to turn the lights on and off or to speak with the Tamagotchi.

Emotion recognition is not used directly as an input for different commands. It is request by the program when it receives an input to decide how to react to it. As explained above, due to implementation difficulties and time constraints, it was not possible to offer a stochastic system that reacts after learning how the user expresses his emotions, but we decided to stick with a deterministic system that reacts in a predefined way according to the command and to the emotion that in reads at that moment from the webcam.

The only inputs fusion that we perform is to get the current command given by the user, either through clicking the buttons or by voice, and the current emotion recognized. Once this two informations are received by the system, the output is simply computed in a deterministic way according to the different combinations.

The next image shows an example of the interface of the game when playing



You can see on the left the different levels explained before of the Tamagotchi. In the middle you have the name on top, a graphical representation of the current state of it, the status of the lights and below we provide an infobox with all the last events that happened when playing. On bottom are some input buttons that you can use to interact. On the right, you can see what the webcam is capturing, the levels of the different emotions that are currently being recognized. Below, the captured voice commands with a utility list of commands that you can use in case voice recognition

is not working properly. This last thing was necessary due to all the problems we had while trying to make voice recognition working. Probably using some more advanced libraries would have lead to much better results, but unlucky we could not find anything better for Javascript, so we worked out this solution to have another interface that you can use in case voice is not working.

5 Case / Care analysis

In this section we briefly explain the analysis of our application according to the CASE and CARE models.

5.1 Case

- Alternate: Turning lights on and off requires to click on the “Turn light” button and tell using the voice if you want the lights to be turned on or off
- Synergistic: When you ask to the Tamagotchi if he wants to play, the system will use both the voice recognition and the emotion recognition to provide the reaction
- Exclusive: You can also use the button to provide the commands instead of the voice, and input one command using the mouse and another one after it using the voice

5.2 Care

- Complementarity: You need to press a button to specify which kind of input you are going to give after using voice
- Assignment: If you want to make the Tamagotchi one year older, you have to press the “Next year” button
- Redundancy: It is possible to given the same input by using the list of recognized command voice or by using the voice recognition system
- Equivalence: It’s possible to input the commands using the predefined buttons for voice command instead of speaking to the application

6 User evaluation

The hypothesis that we want to confirme using our project and evaluating it with different users is that providing natural user interface like voice and emotion recognition provides a better usage of the application and also provides a better empathy of the user with the application.
The problem is that this variables are not easy to measure such as time for example.

One possible solution to evaluate the user experience and confirm our hypothesis is to ask directly to the different users at the end of the test some precise questions about the experience with the application. For example how they felt when at the age of 10 the Tamagotchi died when they did not know in advance that this would happen.

The user should be divided into two groups, one using all the possible interfaces and the other one using only the mouse and button input. The problem with this is that the learning curve for the users would not be the same but we need to split them into two groups otherwise the surprise effect would not be present anymore when the Tamagotchi dies.

Another possibility would be to use the emotion recognition system to capture how the user reacts to the death of the Tamagotchi according to the interface he’s using. This would allow us to see if

there is actually a correlation between the empathy the user creates with the application when using the natural interfaces compared to the simple mouse and buttons one.

Another thing that we could measure is how much the user enjoys using the application. This again could also be done by asking some specific questions at the end of the experience or by using the emotion recognition to see how much the user is happy or surprised when using the application.

6.1 Independent variables

The independent variables that we could change for the different user tests are the interface mouse and buttons or the voice and emotion recognition.

6.2 Dependant variables

The dependent variables that we need to measure in order to confirm our hypothesis are how related to the Tamagotchi the user felt at the end, how natural and easy it was to use the system according to the interface and the enjoyability of the experience.

One thing that we must take into account but varies a lot from user to user is the learning curve according to the different interfaces provided.

7 Conclusions

Our initial goal at the beginning of the project was to create an immersive experience using the Tamagotchi concept of having a virtual companion that reacts to your voice, gestures and emotions. Unfortunately, we soon found out that it was way too hard and it would have taken way too much time to create such a system. So we opted for a more small one, with some specific usage of the different interfaces in a multimodal way that still allowed us to create an application that we can use to test how the different users reacts, behave and feel related to when using it.

The main thing we understood is that capturing the natural input of a person is a very challenging task, especially when it comes to processing and merging them and provide a reasonable response from the program. Moreover, when it comes to testing, it’s hard to measure the feeling of a person with respect to a virtual program. We personally felt that interacting with the Tamagotchi using the natural interfaces was much more enjoyable then clicking on buttons, but still it was hard to behave in a natural way with it.