HEVAH Project

Carina Soledad González González
Daniel J. Díaz González
F. Javier Medina Santos
Zebenzuí Pérez Ramos

October 8, 2004

Abstract

HEVA is a 3D animated character developed with Blender that evaluates the knowledge of a user in a particular subject using an adaptive test.

The avatar was built with the Game Engine of Blender 2.25 and Python.

Our adaptive test is based on XML as data manager and CLIPS as inference engine. Both are connected with a new library in Python that we had to develop for this purpose.

Some characteristic of our project include:

- Mesh deformation in real time (dynamic creation and edition of IpoCurves in Game Engine).
- Mesh deformation in real time (dynamic creation and edition of IpoCurves in Game Engine).
- Interaction with external applications to Blender (such as CLIPS) using pipes.
- Use of threads for concurrent execution of tasks.
- Use of Text to Speech.

Our system is formed by three fundamental modules:

- Question Manager (QM): responsible of select and evaluate questions for the adaptive test. It has been developed in Python.
- User Interface (UI): compose by the avatar developed in Blender.
- Inference Engine (IE): responsible of select the level of knowledge of user through the process of rules in Clips.
The IE and the QM determine the results of the test, that are stored in XML files. These data are processed by the UI using a module developed in Python. The interaction among these modules is carried out by the GameEngine of Blender 2.25, so our animated character has the total control of the system leading the interaction with the user.

Introduction

HEVAH is a character 3D that is born within a more extensive project. This degree project, call "PORTAD and HEVAH: Interface Web and Character 3D for the adaptive evaluation of test in XML", it was made for Superior Engineering in Computer science in the University of La Laguna.

As its name indicates, it is divided in two main modules: Interface Web and personage 3D. Both modules make the evaluation of adaptive tests. In this article we will focus on the personage 3D. HEVAH makes the evaluation the most friendly way for the user. HEVAH has humanoide shape and her function is formulating the questions to the user.

It has been decided to use adaptive tests. This type of test adapts to the knowledge demonstrated by the user at every moment. Once the test ends, an evaluation is made about the obtained answers to evaluate the student.

Finally, the object is to present in a friendly and amazing way an adaptive tests for the user. As well, this test must evaluate the knowledge of the student the best possible way.

HEVAH

HEVAH is the module developed in Blender in the degree project "PORTAD and HEVAH". The work developed in Blender is not limited the modeled one and the animation, it exists a complex development in Python.

Modeled and Animation

The part of modeled and animation was completely made in Blender. The personage is formed by a head and all those elements related to it.

At the time of approaching the eyes, we did not have too many problems. It was decided to create an eye Pixar style. This type of eye is characterized to give much personality to the character by means of the depth. This is easy to simulate, for it is created the eye from 4 parts: ocular globe, rainbow, cheat on and pupil.
Of course, the eyes have to be complemented with eyelashes. For its implementation it’s has been used a plane with alpha channel. Let us see in detail this technique.

Firstly, a plane is inserted, of the approximated thickness of an eyelash. An image with black lines is created (the eyelashes) and the rest is transparent (using a channel alpha). This image is mapped in the rectangle created previously and it is curved to obtain the effect of curly of the eyelashes.

The hair is one of the elements that more realism gives to an animated personage, but, simultaneously, it is one of the most complicated part to approach. An example we find in the cinematographic overproduction is Final Fantasy, whose budget goes up to around the 100 million dollars. Great part of the budget was for the creation of the hair of the protagonist. According to the own people in charge of the project, the most expensive and difficult was to reproduce the straight hair of the young girl, in order that the 60,000 hairs had independent movement. We would like to approach to the realism of this film but our budget is far away from the one of Fantasy End (about 100 million dollars).

The hair that has been modeled has been developed from a mesh to which the wished form has occurred it. Next, this mesh has been divided by heights and made slow parent of each one of the heights carried out weak. Finally an offset has been assigned to these heights obtaining a realistic movement.

All this is located in a stage formed by enough elements.

There is a table that serves as support for the head, which is within a room with metallic walls. All this structure has been illuminated with centers of type lamp (lamp). The illumination of the room has not been created with a predefined scheme, it has been proved until finding the wished effect.

The hole left in one of the walls has filled up with a window. Through this hole and the ceiling, the sky, that has been created using 2 planes, can be observed. A texture has been applied to these planes by means of mapping UV. In the sky it is possible to observe the way a cloud movement takes place. This was obtained by means of a plane with a applied texture.

In order to represent the answers given by the users, a picture has been designed to show the chosen option. After formulating the question, the user must introduce his election by means of keyboard. After this, the picture will reflect the answer until the moment at which a new answer is asked for to him (that is, when finishing of formulating the following question).

Through the window, an incessant traffic of air vehicles can be observed. This effect helps to give the impression that we are in a great futurist city.
Gestures

In order to give her a little more life and naturalness to the personage, we decided to create script that it executed of random way a set of gestures. Within these gestures we found the movement of head, eyes, mouth, etc.

Basically this is the algorithm that has been implemented:

In some especific times (regulated by timer) it is generated a number random that will determine the gesture. When the gesture finishes the character remains in delay until the following gesture is generated.

During the execution of the previous algorithm different types of gestures take place. The gestures can be:

IPOs from GameEngine: In order to activate animations supported by the module of games; in our case, animations of rotation and position.

IPOs from code: We invoked from code the IPOs that uses Relative Vertex Keys. We do not have left more alternative than to call to our method for the execution of this type of IPO from Python.

Motion: In some occasions we used this type of animation due to its simplicity.

Animation

Due to the versatility that allows, they were used RVKs. Next it is detailed the different relative keys used (related or not to the movement of the mouth).

Keys used for Key blinking 1: Closed straight eye. Key 2: Closed left eye.

Keys used for Key speech Key 3: Inferior lip slightly lowered. Key 4: Movement of right comisura. Key 5: Movement of left comisura. Key 6: Superior lip slightly raised. Key 7: Tongue down. Key 8: Tongue arrives Key 9: Key inferior lip raised Key 10: Lowered superior lip

Other key Key 11: Smile

By using all these keys, it is possible obtaine that the mouth adopts the wished form. Since the module of games of Blender 2.25 does not support the RVKs natively, it has been used a method in Python that is in charge of it.

The values of the keys are stored in a configuration file that reads HEVAH when being sent. In this file the buccal groups and the letters belonging to each of them are identified themselves, with the optimal values for their representation. This allows an extense range of action: from being able to sharp the simulation of the movement of each buccal group, to the generalization of these movements creating the corresponding file to different languages.
TextToSpeech (TTS)

The TextToSpeech (TTS from now on) is a tool that allows to transform text written in spoken text. In our case, the TTS was implemented by means of Speech API of Microsoft. This API allows to make speech functions from the code in Python.

The first problem when trying to communicate Blender with the TTS is the sequential, that is to say, when it calls to the TTS, the execution in Blender is left blocked until the execution of the sintetyzer of voice ends. This is totally unacceptable since the voice and the movement of the mouth would be totally out of phase. In order to resolve this problem it has been necessary to use thread-demon; with this one obtains that the execution of the TTS and the one of the Blender are independent. Once established this independence we were with another problem, the possibility that a new thread is executed, or rather demon, without has finished the previous one. In order to solve this, it simply places a semaphore that it indicates if the execution of the thread has finished, so that it allows only a new call to the TTS when the previous one has finished.

Implementation of the illumination

In general, the illumination was created with standard lamps of Blender distributed in an arbitrary form. There are many elements without illumination, because has not been considered necessary. These elements are:

The hair: when having a very dark colour, the illumination would not be appreciated. The eyelashes and the eyebrows: by the same reason that the hair. The bellboys and the ships. In the case of the eyes, two lamps were located to give something of light.

The stage was illuminated adding lamps as the surroundings were created.

To create the illumination of the most important part, the head, we used a predefined scheme.

For the illumination of the head it was used a classic scheme in the cinema. This system uses three light sources. Key light: it is the light located in front of the head. It is of Spot type (directed light) and is strongest of the scene. Light of stuffed: it is the central light that has circular form. It is of Lamp type, to be able to fill up better the hollows lazy by the key light. Back light: it is the light located behind the head. In the Blender it is a light of Spot type of smaller force than the other two.
Interaction with the character

Module of games Blender (GameEngine) has allowed to conduct very complex operations being supported in scripts in Python. Next, it is explained the basic structure of the logic of games used in the project.

Control System

This file of Python makes the main workings of the project, it is the kernel of the logic of games. The nucleus of HEVAH control is being executed continuously from the beginning of the program.

The control kernel makes the following functions: To make the mouse appear To initialize the CLIPS, load the profile of the user To load all the necessary files To pass to the following question To send the blinking

Speech functions

The implementation of the speech in the GameEngine, became by means of two modules that develop the following workings:

hablar.py. One is in charge to send the Speech API, that is as well in charge to initiate the sound of the voice. This bookstore has many parameters, which it allows to make changes on the voice of the character in a quite simple way.

moverboca.py. Its function is to cause that the mouth of the character moves while speaking. The way in which the mouth appears in the configuration file.

Gesticulation

It exists another script that is executed on continuous way:

gestos.py. This file is in charge to cause that the character makes random gestures eventually.

Its main function is to modify the value of the variable ”k”. This value is obtained from random way and establishes that type of gesture is due to execute. In the section of gestures it is explained that gestures were implemented.

Each gesture activates when the variable ”k” is in certain interval.
Answer choice

The system allows to respond with all the letters of the alphabet. For it the user simply must press it in the keyboard. Script is in charge to read this value from keyboard. Change of question

This is one of the fundamental interactions. This part is very complex, since the motor of HEVAH inference is implemented in CLIPS and it was needed to communicate with it from Blender.

For it it had to implement a tunnel between the process of Blender and the process of CLIPS, using the bookstores of management of processes of python. The working of this tunnel is quite simple, but is quite complicated to implement; the idea is to have a process of CLIPS sent in background, and that HEVAH is sending to it the entrances and gathering the exits; that is to say, that HEVAH is the user who is introducing the adecuated commands in the CLIPS and reading the corresponding exits.

There exists a problem with a multiplatform, since these bookstores are different depending on the used operating system. Whereas for operating systems Windows it was used win32api and win32sys, for Linux systems "pexpect" it was decided to use a called bookstore; that it allows to interact with processes sent in background.

Conclusion

After having been almost a year using the module of games Blender, we are convinced of its power and versatility. Nevertheless, there are still several things to improve: texturization, illumination, RVK’s…. Given the power of Blender like modeled program, it is a pity that the models lose quality when incorporating them to the module of games.

For that reason we considered that it would be very interesting to work in the development and evolution of the GameEngine, so that Blender can get to be a professional alternative in the field of the creation of games; we know that this it is not its objective, but with so well laid bases, it would be an authentic pity not to continue progressing in this aspect.

Finally, to say that we consider Blender like a very serious alternative to the programs of modeled closed, being in addition an ideal tool in diverse educative surroundings.