USE OF HYPERVIDEO FOR TEACHING C/C++

Luiz-Alberto Vieira-Dias
Andre Catoto-Dias

UNIVAP-Paraiba Valley University
School of Computer Science
São Jose dos Campos, SP, Brazil
vdias@univap.br

ABSTRACT

This paper presents the use of Hypervideo for teaching the computer language C/C++. This new technology allows the user to learn at its own pace, without following linearly a book chapter, DVD or VHS video. The user can at any time stop, return or skip a portion of the media. The most important feature is that he or she can branch for a more detailed explanation simply by pressing an icon inside the video frame and come back at any time. A demo Hypervideo (including videos + text + voice + music + graphics) to teach C/C++ was prepared in order to demonstrate the feasibility of the technology, using only off-the-shelf software.

1. INTRODUCTION

When a student starts to learn C/C++ [1] he or she has a certain difficulty, for it takes a long time until he or she is able to really write his first programs (not just copy examples), because of the new concepts he has to grasp. The structure of the program is fairly easy, but for a beginner it sounds cumbersome. A typical computer program in C/C++ has its main parts: a) Libraries; b) Definitions; c) Function prototypes; d) Global variables types and classes definitions; e) Main function; f) Functions. All the functions, including the main function [main( )], may have its own local variables and classes. The class definition is the main concept to grasp for Object Oriented Programming. A class has objects and methods, that are the operations allowed for the objects belonging to the class.

An Hypervideo was prepared, not intended to be a full course on C/C++, but just an example on how this new technology can be put to work as a platform for virtual education. The idea of using hypervideo is to allow the user to follow a tree, or a graph [2], instead of being forced to follow linearly a book chapter until its end, as in a traditional course in DVD or VHS video.

2. HYPERVIDEO TECHNOLOGY

Hypervideo is defined as a way of navigation in interactive digital videos, available on accessible media, like the internet and other digital interfaces (local net, your computer, CD-ROM's, DVD, etc). It is similar to hypertext, but the links could be videos or computer generated scenes in addition to text and audio [3].

Hypervideo consists of a base digital video, with icons on certain selected frames that could be clicked by the mouse during the presentation of the video. When an icon is clicked by the mouse, the control moves to another scene. It is shown in the demo that what is called "video" or scene may be a still picture a

computer screen or even a computer program listing or print-out. It is possible to add music, noises or voice. Each frame (base or branched) has controls that may be pressed (clicked) by the user, that decides its own navigation through the hypervideo. It is possible to go back at anytime to the base video, returning to the point of departure, through a go-back button. All videos, base and branched, may have a time line that permits them to stop, run, fast forward, rewind, fast backward, be presented frame-by-frame, and pause.

Figure 1 below presents a frame of an hypervideo. Note the control buttons on the top of the frame and the branch and back buttons inside the frame. Inside the scene there is a yellow button to branch to another scene (branched video), and a circle with a green arrow inside to go back to the main video (base video). Note the time line, on the top of the screen.

Figure 1- Example of a scene from a hypervideo

3. DESCRIPTION OF THE DEMO HYPERVIDEO

The hypervideo for teaching C/C++ was developed using only widely available off-the-shelf tools, like the Adobe Premiere 6.0 [4], and Macromedia Flash MX [5], on a personal computer running on Windows XP [6]. In the developed hypervideo a resolution of 320 x 240 pixels turned out to be adequate. On a commercial system a better resolution would be more appropriate.

Such hypervideo starts with a teacher in a nice environment explaining the main features of C/C++ (base video). As the course progresses the student may turn to another video or computer screen (branch video) in which he or she may visualize a program listing being formed, or a program running, its print-out, its output screen, or even another video explaining some aspects in depth, by just clicking the mouse over the branch icon. A music background was included. At anytime he could go back to the base video by clicking the mouse over the return button. In every branched video it is possible to branch to another sub-branch (if existing) or go back, depending on the way the structure of the course was made.

The authors do believe that the learning in this kind of environment is more effective, for instead of a dull presentation in a classroom by a static teacher, the student is stimulated by a dynamic teacher moving in an interesting setting, alternating computer listings with prize-winner trouts in a Japanese garden. The interested student may stop on the more complex listings, and study it in depth, or return to points in which he or she wants to know more, like for example, to recall a crucial concept, at anytime.

The demo hypervideo developed for this paper has the structure shown in Figure 2. It starts with an introduction about the computer language C/C++, then it goes to the structure of C/C++ programs. At this

point it may proceed to variables, objects, methods and classes, or branch for a more detailed example of the structure of a program, as seen in Figure 3.

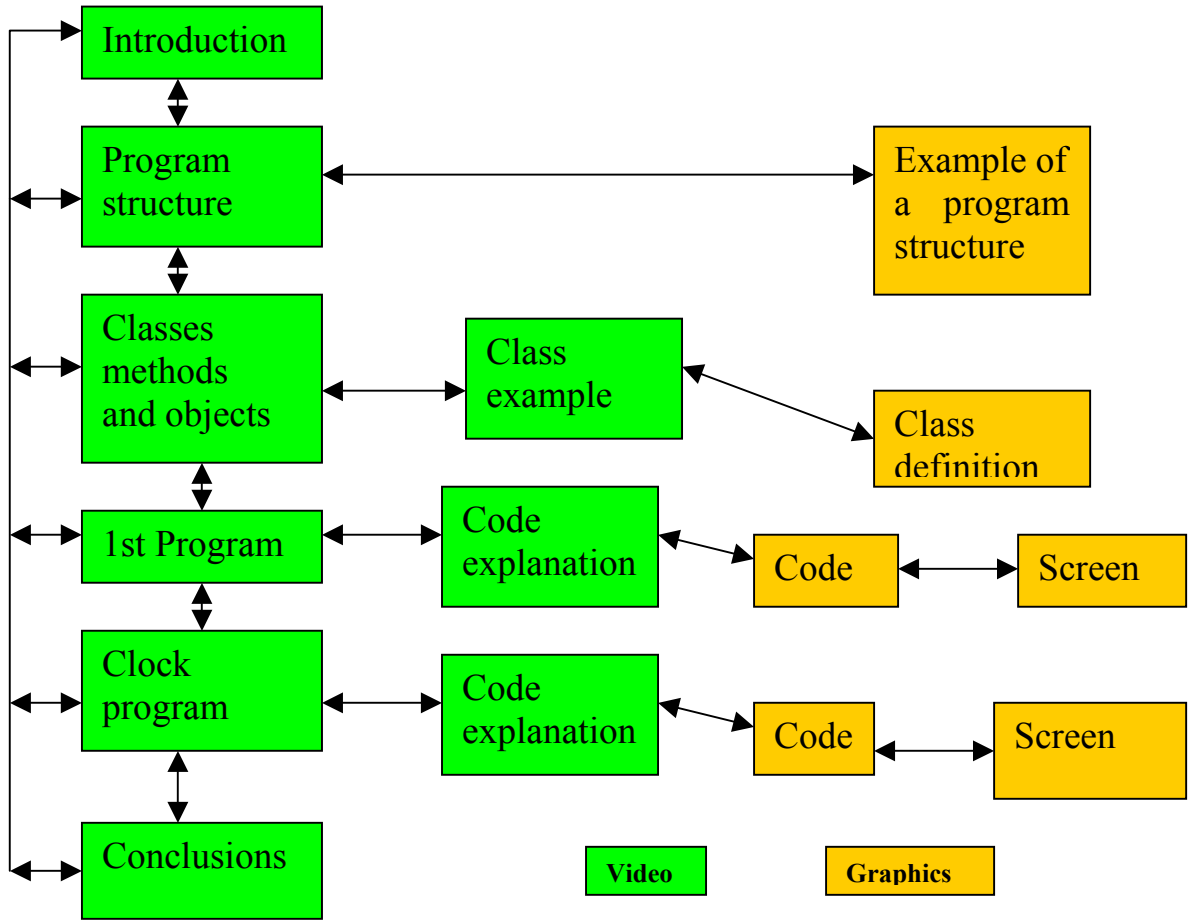

Figure 2 – Demo Hypervideo structure

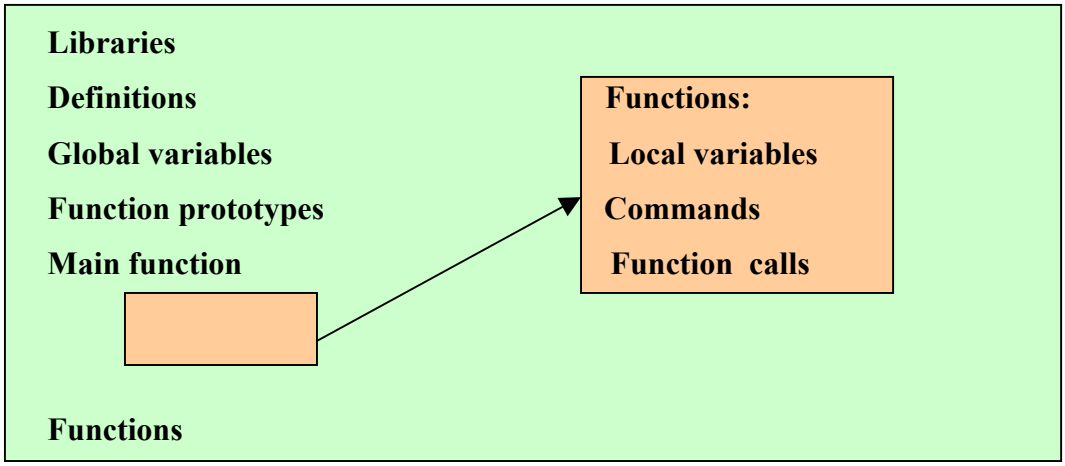

Figure 3 – C/C++ Program structure example

If the user wants to know more about classes it can go to a branch in which an example of a class is presented (class fishes), and if the student is not satisfied he or she may go even deeper and see a precise definition of class.

Now the user is ready for his or her first program. It is the classic program "Hello Miami". The student can see the code and the output, in a branch. Figure 4 presents the code of this program, and Figure 5 its output.

```
#include<stdio.h>
#include<conio.h>

void main( )
{
clrscr( );
      printf("\n\n *********** Hello Miami! ******************");
getch( );
}
```

Figure 4 – Program "Hello Miami"

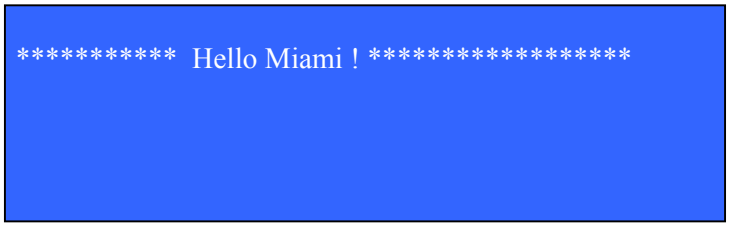

Figure 5 – Output of Program "Hello Miami"

Next the user is ready to go to a more complex program that draws an analog clock on the computer screen and presents the actual time. The time is taken from the computer digital clock and presented as in a dial of an analog clock. If the user chooses to branch he or she can visualize the code of the second program and see the clock running. Then the instructor presents the conclusions and the demo finishes.

In order to program an analog clock it is necessary the following steps: a) Header with the program and programmers names; b) Include the libraries; c) Define the constant π; d) Define individual function prototypes to draw the hour, minute and second pointers (of type **void** for they do not **return** values to other functions); e) Define the type of the global variable t (seen by all functions); f) Start the main function (of type **void**, for it does not **return** any value to other functions); g) Define the types of the local variables (seen only inside the main function); h) Begin graphic mode; i) Start an loop; j) get the time from computer clock; k) Draw the clock dial; l) Call functions to draw the pointers (at the proper position); m) clear screen; n) repeat steps i to m until the user presses the keys CONTROL + BREAK simultaneously; o) Put the code of the functions to draw the pointers (hour, minute and second). Note that the hour pointer is the shortest (65% the length of the seconds pointer), the minutes pointers is 85% the length of the seconds pointer. The complete code is presented in Figure 6.

```
************************************************

//  ************  PROGRAM CLOCK   ***********
//  ******************   by   ***************
//  ************  ANDRE CATOTO Dias  ********
//  ******************   and   ***************
//  *******   Luiz Alberto VIEIRA DIAS  *****

// **************   Libraries   *************
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<time.h>
#include<math.h>
#include<dos.h>
#include<stdlib.h>

//**********   Constant  PI  ***************
#define PI 3.1415926

// *******   Function prototypes  **********
void hourx(float,float,float,float);
void minutesx(float,float,float,float);
void secondsx(float,float,float,float);

// ********   Global variables   ***********
struct time t;

// **************  Main function  **********
void main( )
{
// ***********   Local Variables  **********
float x,y,xc,yc,radius,args,argm,argh;
int gdriver=DETECT, gmode;

// **********   Begin graphic mode   *******
initgraph(&gdriver, &gmode," ");
do
{
gettime(&t);
settime(&t);
// ***********   Draw dial   ***************
setcolor(MAGENTA);
xc=300;
yc=200;
radius=150;
circle (xc,yc,radius);

// ***********   Draw pointers   ***********

// *********  Hour pointer   ************
argh=30*t.ti_hour*PI/180.;
x= radius*(+sin(argh))*.65;
y= radius*(-cos(argh))*.65;
hourx(xc,yc,x,y);
// *******   Minutes pointer   **********
argm=6*t.ti_min*PI/180.;
x= radius*(+sin(argm))*.85;
y= radius*(-cos(argm))*.85;
minutesx(xc,yc,x,y);
// *******   Seconds pointer   **********
args=6*t.ti_sec*PI/180.;
x= radius*(+sin(args));
y= radius*(-cos(args));
secondsx(xc,yc,x,y);
delay(1000);

// **************   Clear Screen   ******
cleardevice( );
}  while(1);
getch( );
closegraph( );
}

//   **********   Functions   **************

void hourx(float xc,float yc,float x,float y)
{
setcolor(YELLOW);
line(xc,yc,xc+x,yc+y);
}

void minutesx(float xc,float yc,float x,float y)
{
setcolor(CYAN);
line(xc,yc,xc+x,yc+y);
}

void secondsx(float xc,float yc,float x,float y)
{
setcolor(RED);
line(xc,yc,xc+x,yc+y);
}
```

Figure 6 – Complete code for the Clock program

Figure 7 shows the program output (an analog clock running). Note the different colors on the dial, and on the pointers. With a minimal change in the code the appearance of the clock may be changed, for instance, the seconds pointer may be drawn inside a smaller dial, as seen in Figure 8.

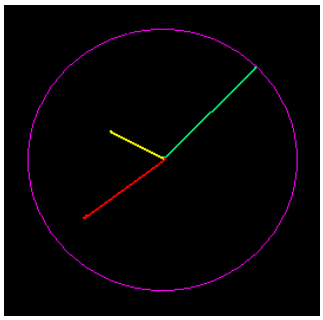

Figure 7 – Output of clock

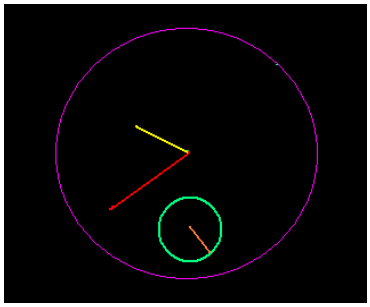

Figure 8 – Another possible output

4- CONCLUSIONS

Hypervideo technology is a powerful tool to do distance learning, for it provides an environment in which, contrary to e-books that uses computers with book technology, it uses computers with computer technology (videos + text + audio + graphics) technology. The user can take advantage of all the features of his or her computer.

The main idea behind the use of hypervideo in teaching the computer language C/C++ is to make the learning process as natural and pleasant as possible (the student learns as if he was being taught by a friend – a much faster process then going through a book or manual), and at its own pace. This feature makes hypervideo a very convenient tool for distance learning.

References

[1] HOLZNER, S. *C++ Black Book*. São Paulo, SP, Brazil: Makron Books, Thomson, 2002

[2] DROZDEK, A. *Estrutura de dados e algoritmos em C++* (*Data Structures and Algoritms in C++*).São Paulo, SP, Brazil: Thomson, 2002.

[3] BALCOM, D. *Hypervideo: Notes Toward a Rhetoric*. www.lcc.gatech.edu/ gallery/hypercafe, 2002. Accessed on December 2002.

[4] ADOBE *Adobe Premiere 5.0.* Harper Collins Publishers, New York, NY, 1998.

[5] REINHARDT, R. ; DOWD, S. *Flash MX, The Bible.* J. Wiley, New York, NY, 2002.

[6] RATHBONE, A. *Windows XP for Dummies.* IDG Books, New York, NY, 2002.