

Learning computer science by Francis Héту

Computer science and mathematics

Design

Junior Age 14

Polyvalente de l'Érablière

Outaouais

Presented at the Science Fair in 2001

Project Summary

Using Visual Basic, I developed a software program that makes it possible to learn the basics of computer science. It is divided into three different categories: Beginner, Intermediate and Expert. My software program is very user-friendly. It is designed as an introduction to computer science. It deals with such topics as internal and external peripherals, binary language and networking.

Project report

Introduction

I developed a computer science tutorial program for the 2001 Science Fair using Visual Basic programming software. My software program is divided into three categories: Beginner, Intermediate and Expert. I deal with different topics in each category. My primary objective was to create a user-friendly tutorial program that would enable beginners to learn the basics of computer science. My software program also enables users to determine whether they have a penchant for this field of the future.

Finding an idea

At first, when I was looking for a topic, it took several hours to finally come up with an idea. Since I had a strong interest in computer science, I decided to develop an interactive tutorial program. I then had to decide how to present it. Once again, I thought long and hard about the different possibilities. Each one had both positive and negative aspects. (See table on p. 3.) I finally decided to develop a software program. I integrated a Web site and a multimedia presentation into my project. I use the Web site to present

my project to the public. It includes explanations about the purpose of my project, as well as how it works and its possible applications. It can also be used to download the program or order it free of charge. The multimedia presentation at the Science Fair will present important information about my software program, such as its aim, target audience and advantages over other tutorial programs.

Prior to development

I now had an idea with potential, but first I had to learn certain concepts in order to create the software program. I already had a basic knowledge of Visual Basic, but this wasn't enough for what I wanted to do. I had to borrow some books from the library and visit a few Web sites that discussed the Visual Basic programming language. I spent nearly 30 hours reading, doing practical exercises and querying before being able to start the real version of my software program.

Analysis chart of the different methods of presenting my project

Possible methods	Positive aspects	Negative aspects
Multimedia presentation (Slide show)	Tremendous instructional possibilities Good graphics interface	Final file too big Very difficult to create questionnaires
Software program	Unlimited possibilities Takes up very little disk space	A lot of code to write More difficult to make the software interactive
Web site	Worldwide access Easier to market	More difficult to make the interface highly interactive Need to have an Internet connection

During development

I had to find an attractive graphic interface for my software program. I tried a dozen possibilities before deciding on one that was attractive and could be taken seriously. I made sure that all my objects, images and commands were simple and user-friendly.

I once again spent several hours looking for all the necessary documentation. I searched the Internet and the library for information before starting to write the text for my learning material. As I mentioned in the introduction, my software program is divided up into three categories (Beginner, Intermediate and Expert). Each level is then divided into subcategories.

Explanatory table of subcategories

SUBCATEGORY	EXPLANATION
Introduction	A short summary of topics covered at this level
Learning	The material to be learned to complete this level
Exercises	Practical exercises in which the user must complete sentences
Games	A fun method of reviewing important concepts
Pretest	A glimpse at the material covered in the final exam
Final exam	The final stage of learning that helps determine whether users have understood the material or experienced any difficulties

I had now reached the longest and most difficult stage: software programming. To simplify my task, I began by writing a pseudo-code (coding preparation on paper) and then I completed the software programming, which took about 60 hours.

Problems encountered

Despite my planning, I encountered problems throughout the programming stage. The biggest problem involved the registration of users' names, which required the use of several variables (memory storage space for data) and was very difficult to coordinate. I also had difficulty programming the background music. I referred to a Web site, where I

found the necessary coding. It seems useless to list all the problems I encountered, since I made all the necessary efforts to overcome them.

Optimization

After finally completing my software program, I changed a few images to make them more interesting. I changed the way I presented my texts in the “Learning” subcategory, to make them more attractive. I improved the coding by eliminating as much of it as possible. I once again tried to make my software program more user-friendly by generating small text boxes when the cursor is placed on a command. The greatest and most interesting improvement involved the addition of an assistant to help users throughout the learning process.

Why choose my software program?

There are numerous software programs available to learn about computer science, most of which are excellent. Some, however, are very expensive, others are difficult to use and still others are boring because the graphic interface is unattractive. In some cases, the concepts are poorly structured or irrelevant. Unlike some of these tutorials, mine is very user-friendly, the graphic interface is pleasant, and the information is relevant and can be used by anyone wanting to learn the basic principles of computer science. (See the table below.)

Table of various concepts taught in my software program

Categories	Concepts taught
<u>Beginner</u>	External and internal computer components Computer maintenance Basic computer language (binary code) Operating systems Screen resolution
<u>Intermediate</u>	The origin of the computer Ports Adapting work space (office) Continuation of binary code (Doing binary calculations)
<u>Expert</u>	Networking Security Continuation of binary code (hexadecimal and octal) Explanations of different words used in computer science

The Science Fair

I made a few copies of my software program on CD in order to be able to distribute it to some of the teachers at the Science Fair. I will give visitors the chance to try my software program in order to encourage them to use it. I will also organize a small contest in which participants will have to listen to my explanations and then answer a question. The names of those who answered correctly will go into a draw for a chance to win one of five copies of my software program.

Conclusion

My family and friends tested my software program and everyone thought it was satisfactory. In fact, I have not received any negative feedback. This year, I plan to present my software program to a government organization or a publishing house. I hope that, as part of the education reform, it can be used to introduce all young people to the world of computer science. My tutorial would definitely satisfy all teachers' expectations.