

An Adaptive 3D Virtual Environment for Learning the X3D Language

Luca Chittaro

HCI Lab, Dept. of Math and Computer Science
Via delle Scienze, 206 – 33100 Udine, Italy
chittaro@dimi.uniud.it

Roberto Ranon

HCI Lab, Dept. of Math and Computer Science
Via delle Scienze, 206 – 33100 Udine, Italy
ranon@dimi.uniud.it

ABSTRACT

This demo will illustrate the adaptive interface of an application (whose architecture and implementation have been presented in detail in [2]) for learning 3D Graphics programming (in the X3D ISO standard). The application situates users in a 3D Virtual Environment where they can learn by selecting and inspecting actual examples.

Author Keywords

Adaptive Interfaces, Personalization, X3D, Web-based learning.

ACM Classification Keywords

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – *Artificial, augmented, and virtual realities*. I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – *Virtual reality*. K.3.2 [Computers and Education]: Computer and Information Science Education – *Computer science education*.

INTRODUCTION

3D Virtual Environments (VEs) can help people in learning through direct experience by visualizing concepts and performing tasks in a reproduction of the real world or in completely fictional worlds that are suited to the learning task [2]. For these applications, adaptivity can play an important role in increasing the effectiveness of the learning process and usability of the interface. In [2], we have presented a system that is able to deliver personalized learning content in Educational VEs by using the well-known AHA! system [3] as its adaptive engine. Moreover, we also proposed a case study in which our system was used to build an adaptive 3D Web application for learning how to build interactive 3D graphics content using the eXtensible 3D (X3D) language [4], an ISO standard for

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IUT'08, January 13-16, 2008, Maspalomas, Gran Canaria, Spain.
Copyright 2008 ACM 978-1-59593-987-6/08/0001 \$5.00

building 3D content and VEs that can also be experienced through the Web. In this demo, we first briefly motivate the application and then show how a user interacts with it.

MOTIVATION

Traditional learning of 3D graphics programming is accomplished through textual explanations (e.g., textbooks or Web pages explaining how the chosen language works, the syntax and semantics of commands or language primitives), screenshots and code examples that demonstrate language features and usage. Our demo proposes an alternative, more interactive learning style for 3D graphics programming based on the following main ideas:

- the learning experience is situated in a VE, so that the user can: (i) learn from actual examples, and (ii) use them as a motivation to learn (e.g., "I'd like to learn how to obtain the particular graphic result I'm seeing").
- since code examples are fundamental to fully understand a language, but difficult to understand when the user has no sufficient prerequisite knowledge, the system exploits adaptivity to (i) focus the user on examples that are more suited to her knowledge level, and (ii) present them in a personalized way.

INTERACTING WITH THE APPLICATION

In the demo, the user enters a VE that reproduces a real-world computer science classroom (Fig. 1) to situate the learning experience in a well-known environment (the application is indeed targeted at computer science students).

The user is able to walk in the classroom and click on any object to inspect, in a separate window, the X3D code that implements that object, thus learning from actual examples. Code visualization is adaptive, i.e. it depends on the user's level of knowledge about X3D concepts (which is represented into the adaptive engine). In the code visualization (see the example in Fig. 2), the user can click on any name of language primitive (e.g., Shape, Cylinder, ...) to obtain a separate Web page explaining its syntax and semantics.

Our code visualization takes well-known text presentation techniques and applies them in the new context of program



Figure 1. The virtual computer science classroom where learning takes place.

```
<X3D version='3.0' profile='Immersive'>
  <head>
    <meta name='created' content='10 June 2007' />
  </head>
  <Scene>
    <Viewpoint description='front view' position='0 0 5' />
    <Shape>
      <Appearance> ... </Appearance>
      <Cylinder radius='1' height='4' />
    </Shape>
  </Scene>
</X3D>
```

Figure 2. Personalized code visualization. This X3D document defines a cylinder-shaped object. Black arrowheads close to primitives can be clicked to hide/show code fragments.

code visualization. In particular, we use:

- *color-coded syntax-highlighting* for code the user already understands or is suitable to learn (e.g., the Cylinder primitive in Fig. 2). Code not yet suitable to learn is displayed in light grey (e.g., the Viewpoint and Appearance primitives in Fig. 2); typically, a primitive is suitable to learn when the user has a sufficient degree of prerequisite knowledge;
- *scaling fragments* [1], i.e., adaptively using text size to emphasize content based on relevancy, to encode the suitability of X3D concepts: the greater the size, the more a primitive is suitable for learning, e.g., the Cylinder primitive in Fig. 2 is the most suitable primitive to learn;
- *stretchtext* [1], i.e., using placeholders to highlight the presence of and allow access to secondary information, for hiding code fragments that the user cannot currently understand and are longer than one line (e.g., in Fig. 2, the content of the Appearance primitive is hidden). This makes it easier for the user to focus on the more relevant parts of code, while not preventing access to full code.

Based on the user's understanding of the X3D language, some 3D objects in the VE may be more suitable to examine than others. Our application thus focuses user's attention by visually highlighting (as in Fig. 3) the 3D objects that demonstrate language features the user is ready to learn. Moreover, the visual appearance of suitable objects is also adapted to: (i) eliminate details that might distract or hinder learning or (ii) add details that might be useful for learning (e.g., indicating the position of light sources which is normally not visible in VEs).

Each time the user inspects the code of an object and reads about primitives syntax and semantics, her user model will be updated to reflect increases in her knowledge of the language. The demo shows the changes in the user model as the user interacts with the VE, reads about new primitives, and inspects code.

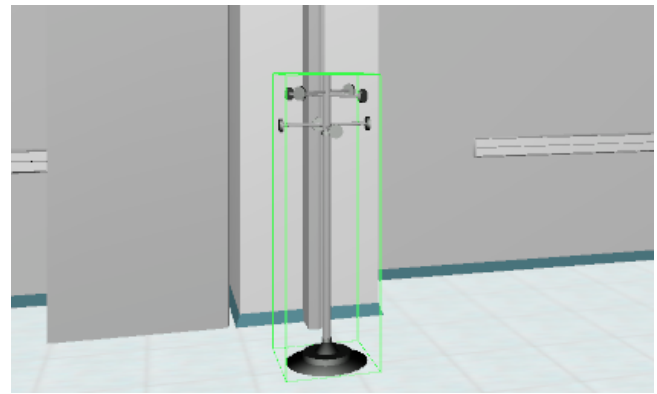


Figure 3. A green wireframe box visually highlights an object suitable for code exploration.

ACKNOWLEDGMENTS

Our research has been partially supported by the Italian Ministry of Education, University and Research (MIUR) under a PRIN 2005 grant.

REFERENCES

1. Brusilovsky, P. Adaptive hypermedia. *User Modeling and User Adapted Interaction* (Ten Year Anniversary Issue), 11, 1/2 (2001), 87-110.
2. Chittaro, L., Ranon, R. Adaptive Hypermedia Techniques for 3D Educational Virtual Environments, *IEEE Intelligent Systems* (Special Issue on Intelligent Educational Systems), 22, 4 (2007), 31-37.
3. De Bra P., Aerts A., Berden B., De Lange B., Rousseau B., Santic T., Smits D., Stash N., AHA! The Adaptive Hypermedia Architecture. *Proceedings of the ACM Hypertext Conference*, Nottingham, UK. ACM Press (2003) 81-84.
4. X3D International Standard (2004). X3D framework & SAI. ISO/IEC 19775:200x. Available at www.web3d.org/x3d/specifications (last access on November 2007).