Alice: Easy to Use Interactive 3D Graphics

Jeffrey S. Pierce
Steve Audia, Tommy Burnette, Kevin Christiansen, Dennis Cosgrove,
Matt Conway, Ken Hinckley, Kristen Monkaitis,
James Patten, Joe Shochet, David Staack, Brian Stearns,
Chris Sturgill, George Williams, Randy Pausch

Computer Science Department, Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213
{jpierce, pausch} @cs.cmu.edu

ABSTRACT

Alice is a rapid prototyping system used to create three dimensional graphics simulations like those seen in virtual reality applications. Alice uses an interpreted language called Python as its scripting language to implement user actions. This interactive development environment allows users to explore many more design options than is possible in a compiled language environment. The alpha version of Alice for Windows 95 is available for free over the internet, with the beta release scheduled for August.

KEYWORDS

Virtual reality, 3D graphics, rapid prototyping, usability engineering

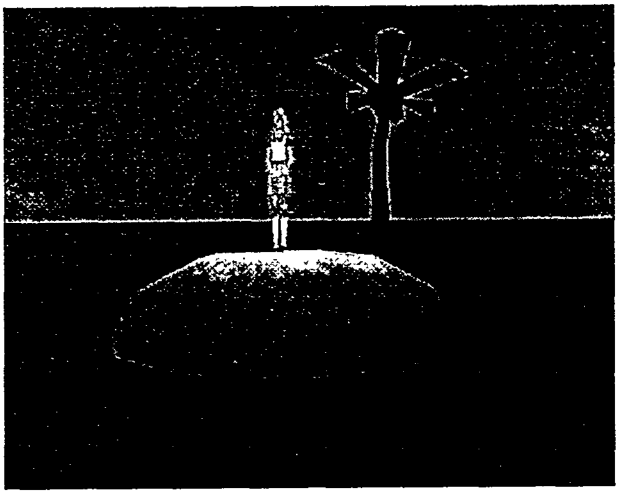
INTRODUCTION

Current Intel and Macintosh platforms now have sufficient power to run interactive 3D graphics programs without hardware acceleration. Yet the development of our ability to author these applications has not kept pace with the development of hardware to run them. Specifically, current authoring systems are generally too *hard to learn*, requiring several months of specialized training, and impose a long turnaround time for developers to observe the results of changes in code, making them *too hard to change.* Our goal now is to provide ease of authoring: to build development environments for interactive 3D worlds so that end-users can create and experiment without specialized 3D graphics training.

Alice, a 3D scripting environment started by the User Interface Group at the University of Virginia and now under development at Carnegie Mellon University, is a first step toward addressing the 3D authoring problem. The goal of the Alice project is to lower the training and education barriers and allow everyone to experiment with the new medium of interactive 3D graphics by attacking this problem on three fronts: the programming language, the primitives and abstractions used to describe the behavior and animation of 3D objects, and the development environment that supports the author.

A BRIEF DESCRIPTION OF PYTHON

Alice uses Python, an interpreted language written by Guido van Rossum [1], for its scripting language. Python is a high level, object oriented language with a clean syntax that is relatively easy to learn. Some observers have compared Python to "pseudocode that runs." Because Python is



Alice runs on Windows 95 computers, and does not require hardware acceleration for acceptable performance. This illustration is a screengrab of one of the worlds that has been built using Alice. These "opening shots" are the default worlds we provide with Alice. Users can also easily create their own worlds if they don't want to use one of the defaults. The author of this particular world has brought our Alice model into the world and placed her on the island. UIST attendees will have the opportunity to experiment with Alice and create worlds of their own.

UIST 97 Banff, Alberta, Canada
Copyright 1997 ACM 0-89791-881-9/97/10..$3.50

interpreted, we make it extremely easy for authors to ask *what if.* Changes can be made and their effects can be observed in seconds because interpreted languages allow new code to be evaluated while a program is running. This constrasts with C or C++, where authors are forced to endure a turnaround time of minutes due to the necessary compilation step. Python is also extensible in C and C++, so that more experienced programmers can extend Alice as needed.

PRIMITIVES AND ABSTRACTIONS

Current APIs for creating interactive 3D graphics expose the author to the underlying implementation of the rendering engine. The scale function available in most APIs provides a good example. This function not only scales the geometry of the object, but affects subsequent translation operations as well. Our user tests have shown this to be highly non-intuitive behavior. We believe that there is no inherent need for users to understand 4x4 homogenous matrix transformations in order to build 3D worlds. We have carefully structured the Alice API so that users are not exposed to the underlying implementation. The word matrix does not appear anywhere in the Alice API. We have also removed references to the X, Y, and Z directions from the API, because our user tests have demonstrated that users have difficulty determining what the X, Y, and Z directions are for a given object. Instead, Alice users move objects Forward/Back, Left/Right, and Up/Down, making Alice feel a great deal like a 3D version of LOGO [2].

Another problem with current APIs is that they only provide primitive operations. Creating high level behaviors for objects using these APIs has been compared to baking an apple pie given the periodic table. We have tried to provide a number of high level commands for specifying object behaviors. The PointAt command (that orients one object toward another) and the Place command (positioning one object relative to another) are prime examples. In addition, commands in Alice can be animated simply by specifying an optional Duration parameter, and animations can be composed to create more complex animations using DoInOrder and DoTogether.

ALICE IS EASY

Much work has gone into creating a system that lowers the cognitive burden of creating 3D interactive programs. To do this, we have used findings in the spatial understanding literature [3] [4] to help structure the way programmers think about problems in three dimensional object placement and dynamic behavior. These findings are reflected in the design of behavioral abstractions provided in the Alice API.

Alice has also undergone a constant user testing process throughout its development. We have performed more than 80 formal user tests on the beta version of Alice, and more than 5000 people have downloaded the alpha version of Alice from our home page. We are working closely with high school teachers in Lynchburg, Virginia who are using Alice in their classrooms. We are also working with a number of people who are providing us detailed feedback about their use of Alice, including a mother and her daughters (9 and 11) who are using Alice in their home schooling.

ALICE IS FREE

The beta version of Alice is now available for free from our website. The initial beta version will be available from our website in May of 1997, with a subsequent release to follow in August. Alice runs on Windows 95 platforms, and does not require hardware acceleration for acceptable performance.

WHAT WE WILL BE PRESENTING

The alpha version of Alice was presented at UIST 96. Since then, Alice has undergone a great deal of development and user testing in preparation for the beta release. Alice beta uses Direct 3D and DirectSound from DirectX version 3, and uses VisualBasic for the GUI. We will have a number of PCs running Alice that people can use to experiment with Alice themselves. These PCs will also be used to demonstrate the new Alice API and GUI, the tools that have been created to simplify interacting with Alice, and the changes in functionality, like the redesigned animation capabilities.

AUDIO / VIDEO NEEDS

We will need three Pentium PCs with monitors and CD-ROM drives that we can install Alice on.

WHERE TO GO FOR MORE INFORMATION

The Alice web page is

`http://alice.virginia.edu`

REFERENCES

[1] Guido van Rossum, Interactively Testing Remote Servers Using the Python Programming Language. This paper and the Python programming language is available from http://www.python.org

[2] Seymour Papert, *MindStorms: Children, Computers, and Powerful Ideas*, Basic Books, New York, 1980.

[3] Stephen Kosslyn, *Image and Mind*, Harvard University Press, Cambridge, Mass, 1980.

[4] M. C. Corballis *The Nature of the Left-Right Coding*. Paper presented at the annual meeting of the American Psychological Association, Los Angeles, 1981.