



WAVESworld

a testbed for constructing 3D
semi-autonomous animated characters

Michael Boyle Johnson

Bachelor of Science
Department of Computer Science
University of Illinois at Urbana-Champaign, Illinois, 1988

Master of Science in Visual Studies
Media Arts and Sciences Section
School of Architecture and Planning
Massachusetts Institute of Technology, 1991

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree
of Doctor of Philosophy at the
Massachusetts Institute of Technology

June 1995

author:

Michael Boyle Johnson
Program in Media Arts and Sciences
May 5, 1995

certified by:

David L. Zeltzer
Principal Research Scientist
Research Lab for Electronics
Massachusetts Institute of Technology

accepted by:

Stephen A. Benton
Chair, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

WAVESworld

a testbed for constructing 3D
semi-autonomous animated characters

abstract

by Michael Boyle Johnson

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning, on May 5, 1995
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

This dissertation describes a testbed for experimenting with the issues surrounding designing, developing, debugging, and delivering three-dimensional, semi-autonomous animated characters.

This testbed sits atop an object-oriented framework for constructing and animating models. This framework facilitates iterative construction of the parts and props that comprise a character. It also provides facilities for writing and wiring together agents, which are processes that measure and manipulate models over time to produce a character's behavior. This framework encourages and facilitates encapsulation and reuse at many levels, which benefits collaborative character construction.

This testbed can be used to compose three-dimensional, photorealistic animatable characters, where characters are composed of variously interconnected agents and a model, where a model is a set of objects encapsulating shape, shading and state information over time. It's possible to quickly build reusable, composable, blendable behaviors, where a behavior is the result of some set of processes operating on a model over time.

One especially useful result of using this framework to develop media is its facility in acting as a very rich and compact storage medium for photorealistic scenes. This storage representation builds directly atop the RenderMan Interface, an industry standard for describing photorealistic scenes. In our object-oriented representation, though, since we maintain some level of continuity in our scenes over time, such scenes can have 3D models that change over time, where different parts of the models in the scene can be changing at different rates. Especially interesting is that these scenes need only a very modest run-time system for playback at arbitrary frame rates with respect to the scene time.

Assuming the underlying components of the scene were sampled appropriately, the scene can be played back at arbitrary spatial and temporal frequency. In other words, the scene can be treated as continuous media. With appropriate sampling, the representation is not lossy. For a large class of scenes, this allows orders of magnitude of compression of the amount of data which need be stored or transmitted. This framework is extensible, so compound components of a scene can be encapsulated and efficiently stored and transmitted.

This work was supported in part by ARPA/Rome Laboratories, NHK (Japan Broadcasting Co.), the Office of Naval Research, equipment gifts from Apple Computer, Hewlett-Packard and Silicon Graphics, and software gifts from Altsys, NeXT, Pixar, GESTEL Italia srl, Stone Design, and Larry Gritz.

Ph.D. Thesis Committee members



David L. Zeltzer, Ph.D.
Principal Research Scientist
Research Lab for Electronics
Massachusetts Institute of Technology

Pattie Maes, Ph.D.
Assistant Professor, Autonomous Agents Group
Media Laboratory
Massachusetts Institute of Technology

Ed Catmull, Ph.D.
President
Pixar

- 8 ▪ Acknowledgements
- 10 ▪ Colophon

1

Introduction and Overview

- 11 ▪ Introduction
- 12 _Preface
- 13 _Background
- 14 The Thesis
- 16 ▪ The Problem Posed as Questions
- 16 ▪ My Approach: An Overview
- 12 _A Quick WavesWorld Glossary
- 19 ▪ Contributions of this Work
- 20 ▪ Map of the Dissertation

2

Related Work

- 21 ▪ Introduction
- 22 ▪ Character Construction in the Media Arts
- 22 _An Actor Prepares
- 22 Mirror, Mirror...
- 23 Demeanor: A Character's Potential for Action
- 23 Action!
- 24 _Building a Character
- 24 Let's Take it Again, From the Top...
- 26 _Summary
- 26 ▪ What Have We Learned?
- 27 ▪ Computer Graphics
- 27 _Task Level Animation
- 28 _Some Interesting Actors and Characters
- 29 _Animation Systems
- 30 _Materials & Scene Description Languages
- 31 ▪ Artificial Intelligence

3

Modeling in WavesWorld

35	• Introduction
37	• Representation Requirements
37	_Comprehensive CG Vocabulary
37	_Encouraging Encapsulation and Abstraction for Materials
38	_Continuous over Time
39	_Composable
39	_"The nice things about standards, there's so many to choose from"
40	_Building on the RenderMan Interface
41	_Why not use OpenInventor (or OpenGL)?
43	• Three Simple Examples
43	_Static Example: A Cylinder
45	_Dynamic Example: Rotatable Cylinder
50	_Abstraction Example: Squishy Sphere
52	• All about Eve
52	_Making a Model "Animatable"
53	Animatable == has the potential for change
54	_Articulated Variables: A Model's Namespace
54	_The Model == Renderable Objects (some of which are Animatable)
55	_The Eve Commands
59	_Composing New Classes
59	_Building up the Scene
61	_Shooting the Scene
63	_Tcl Additions
63	Types
64	Attaching to and Detaching from Variables
65	Efficient Math on Arrays
65	More Math Functions
67	• Building and Debugging Models
67	_Dealing with a Model as a Database
67	_Manipulating and Visualizing Model Parameters
69	_How Does it Work?
73	Drag and Drop Debugging
75	_The WWTCLKit:UI Classes for Manipulation and Visualization
75	Useful GUI Classes
75	GUI Classes for Manipulation
76	GUI Classes for Manipulation and Visualization
77	• Reusability: The Packaging Problem
77	_A Quandary in 2D Illustration
79	_Thinking of Collaborating on A Film Noir
80	_Packaging up a Robot: Collaboratively Building a Character in WavesWorld
81	Making Choices Map to Other Choices
83	• More (Slightly Complex) Models
83	_Venetian Blinds
88	_Room
90	_Dimitri's Ring from "Dizzy Horse"
93	_SphereHead
94	_Audreyell
95	_Rocky da Humanoid
98	_Robbit
101	• Conclusion

4

An Agent-Based Approach to Animation

102	• Introduction
103	• What's an Agent?
103	• Background
103	_Build-aDude
104	_Planning Algorithms
104	_Early Worlds
104	raveWorld
105	sanderWorld
105	Lessons Learned
106	_Revisiting Planning
107	_The Need for Building Blendable Behavior
108	_An Agent-Based Approach to Building Behavior
109	• A Simple Example: Point's Point of View
114	• Agents in WavesWorld
115	_Sensor Agents
115	Receptors
117	_Goal Agents
118	_Skill Agents
118	The Anatomy of a Skill Agent
118	Reflex, Oscillator, or Servo-Mechanism
118	Writing a Skill Agent
120	The Importance of Conventions
121	_Different Agent Configurations
122	• Trading Complexity: Model vs. Behavior/Space vs. Time
124	_"Behavior" Shaders: A Useful Analogy
126	One-Offs/Special Purpose
126	Reusable/General Purpose
128	• Building Behaviors
129	_A Simple Orient Behavior
138	_Building a Reusable Behavior
138	Initial->Min->Max->Initial: Predetermined Percentages
139	Initial->Min->Max->Initial: Constant Velocity
143	Reuse on SphereHead
145	Reuse on Rocky
145	Reuse on Robbit
146	_Building Behaviors by Composition
149	Audreyell in the Room
151	• Conclusion

5

Modeling in WavesWorld

152
153

- Conclusions
- Future Directions

A

154 Reactive Planning in WavesWorld

R

164 References

WAVESworld

acknowledgements

to my advisor, David Zeltzer, for sticking with me as I seemingly meandered all over the place, who shielded me from politics I didn't want to know about or deal with, for good discussions and great advice.

to my committee, Ed Catmull and Pattie Maes, for their helpful comments and useful advice along the way.

to Lew Tucker, when he was at Thinking Machines, who funded my summers, lent a sympathetic ear, and always made sure I had the resources I needed.

To the fine folks at Pixar for the summer of '93, especially Dana, Annabella, Bill, Tony, Dan, Tom, Kay, Mark, Mike, John and all the rest who made sure I enjoyed myself as I was working there, and an extra thanks to Ed, who found the money to bring me out in the first place.

to Nicholas Negroponte, for the gift and challenge of the Media Lab, for fomenting quirkiness, and for always answering his e-mail promptly.

to Muriel Cooper, who understood what the hell I was talking about, which is much rarer than I realized at the time.

to Dave Springer and those at NeXT and Pixar who worked on NeXT's 3DKit and qrman, which gave me a great leg up in implementing the WW3DKit.

to all those who worked on the RenderMan Interface and prman.

to my buddy Larry Gritz and his amazing Blue Moon Rendering Tools (and shaders), without whom I would have been unable to make my points so graphically. You're a prince!

to John Ousterhout and the whole tcl community; the finest example of distributed collaboration I've ever seen.

to Jean-Marie Hullot and the other folks at NeXT that worked on InterfaceBuilder, and the rest of NEXTSTEP, for that matter.

to the past denizens of the SnakePit and the present inhabitants of the Bomb Shelter, especially Josh, Joe, Tom and Neil, for many, many fruitful late night conversations.

to Ahn, who's selfless HP hardware swap enabled me actually get done in 1995.

to Linda, Lena, James, and Santana, who kept the paperwork working (and were way more cheerful about it than I had any right to expect).

to all my NEXTSTEP buddies, especially the ones who think WavesWorld is cool **and** have some inkling what it's good for. Andrew Stone has always been there, and Bruce Blumberg was invaluable in the early days of my getting up to speed on NeXT hacking. Peter, Ian, Terry, Ray, Joe, Leo, Guido, Rohit, Keith, Alex, Jeff, and all the rest of the folks from comp.sys.next.*.

to Warren Sacks, for not believing me until I convinced him, and then having faith in me once I had.

to Abbe Don, for late night kvetching and reminding me of the fun and power of collaboration.

to Steve Drucker, for always enlightening discussions, in the beginning and especially near the end.

to Henry Jenkins and Marc Davis, who taught me the value of literary theory and criticism.

and to all the rest at the Media Lab that have shaped my twisted thinking these past seven (gulp!) years. No need to name names; you know who you are...

to our cats, Kato and Psycho Kitty (and the dearly departed Barton), who have helped me through countless debugging sessions—every hacker should have cats.

Finally, to my family, whose emotional and financial support have always helped sustain and move me forward in this process: my parents Robert and Genevieve, my sisters Jean, Marirose, Helen, and of course, Elizabeth: my wife, my love, my best collaborator, my better half that makes me whole. Hey kiddo; we're outta here!

WAVESworld

colophon

This document was completely produced on my faithful NeXTStation Color, clones (formerly filmNoir), running NEXTSTEP 3.2.

It was laid out in FrameMaker 3, with 2D illustrations done in Altsys Virtuoso 2.0, LightHouse Design's Diagram2, Appsoft Image, Stone Design's Create, with Adobe System's Formata font for the main body font and Emigre Graphics' Triplex for title accents.

On the 3D side, in addition to WavesWorld itself, GESTEL's solidThinking modeler, Alias' PowerAnimator, Pixar's Typestry 2 and ShowPlace 2.1, Pixar's PhotoRealistic renderer, and Larry Gritz's Blue Moon Rendering Tools were used to model and render the illustrations in this dissertation.

The layout was designed by Elizabeth Glenewinkel. Using mockups she produced in FreeHand on the Mac, I transcribed the design into Virtuoso and FrameMaker templates. The original dissertation was all written directly in HTML using Emacs.app 4.0 (which incorporates GNU's emacs 19.xx), visualized with Lighthouse Design's OmniWeb, and cut and pasted into FrameMaker.

The author gratefully acknowledges the educational pricing provided by NeXT, LightHouse Design, Appsoft, Altsys, Stone Design and software donations by Altsys, Pixar, GESTEL Italia srl, Larry Gritz, and Stone Design.