2002

Game Engines for Use in Context Aware Research

McDowell, Perry

UBI Comp 2002, Adjunct Proceedings, Fourth International Conference on Ubiquitous Computing, September 29-October 1, 2002, Sweden

UBICOMP2002

ADJUNCT PROCEEDINGS

PETER LJUNGSTRAND AND LARS ERIK HOLMQUIST (EDS.)

FOURTH INTERNATIONAL CONFERENCE ON UBIQUITOUS COMPUTING
SEPTEMBER 29 - OCTOBER 1, 2002
DRAKEN CINEMA AND CONFERENCE CENTER
GÖTEBORG, SWEDEN

PREFACE

This volume is the adjunct proceedings of the UbiComp 2002 international conference on ubiquitous computing, collecting extended abstracts for posters, doctoral consortium, and video submissions. At this point in time, we already know that this year's conference will be by far the biggest UbiComp yet, and the number of submissions in the different categories reflected this.

Posters are a vital part of an academic conference, since they provide an opening for late-breaking and controversial work. They also provide an opportunity for researchers to discuss and test new ideas in an informal setting. We accepted a total of 27 posters for UbiComp 2002, spanning every imaginable area of UbiComp research and development, from telematics to user studies, from protocols for music sharing to ubiquitous computing in education. The poster abstracts are the first and by far the most substantial part of this volume.

Doctoral students are important for the continued growth and development of the UbiComp research field. This year, we arranged the first UbiComp Doctoral Consortium, chaired by Anind Dey of Intel Research Berkley. The abstracts submitted by the doctoral consortium students comprise the second part of the adjunct proceedings.

The video program was an innovative new part of this year's conference, where we invited researchers and practitioners to submit videos that represented UbiComp research, new and old. We received a very high number of submissions and the video jury spent many hours viewing and commenting the videos. While the video program is mostly a "live" event, in the form of a screening of all selected videos at the UbiComp conference, several authors also chose to supply written abstracts. These comprise the final part of this volume.

Göteborg, Sweden, September 2002

Peter Ljungstrand, Posters and Video program chair

Lars Erik Holmquist, General Chair

This volume is published by the Viktoria Institute, Göteborg, Sweden, September 2002.
Cover design by Sus Lundgren, PLAY Studio, Interactive Institute.
Printed by TeknologTryck, Elektroteknologsektionen Chalmers, Göteborg.

TABLE OF CONTENTS

EXTENDED ABSTRACTS FROM THE UBICOMP 2002 POSTER PROGRAM

EXTENDED ABSTRACTS FROM THE UBICOMP 2002 DOCTORAL CONSORTIUM

EXTENDED ABSTRACTS FROM THE UBICOMP 2002 VIDEO PROGRAM

Game Engines for Use in Context Aware Research

Perry McDowell, Andrzej Kapolka, Michael Capps, Mike Zyda
MOVES Institute, Naval Postgraduate School
833 Dyer Road, Rm Sp-517
Monterey, CA 93943-5118 USA
+1 831 656 7591
{mcdowell, akapolk, capps, zyda} @nps.navy.mil

ABSTRACT

One of the biggest difficulties to overcome in creating and testing context aware applications is the interface with the real world. This includes both inputting data from the real world into a symbolic format and outputting the data to the user in a useful fashion. In this poster, we describe how we used a commercial game engine to overcome these difficulties and take the place of the real world.

Keywords

Context-aware, game engines, augmented cognition

INTRODUCTION

Ubiquitous computing is at a similar point to that which home computing was twenty-five years ago. Currently, while the hardware is cheap enough to permit market saturation, the applications have not yet reached the future that Weiser envisioned nearly a decade ago[1]. The main problem is that interfaces are lacking, and the only people using such systems are small groups of enthusiasts who are willing to endure a cumbersome interface. For ubiquitous computing to gain widespread use, the equivalent of the GUI must be created so novice users can use it easily.

We feel that creating context-aware applications are this equivalent of the desktop GUI for ubiquitous computing. By correctly determining the context of a situation, the computer can provide much better and more personalized assistance to the user. Building a system where the device can intelligently use all that information without the user having to input it will make it available to novice users.

Simple Context Aware Architecture

As part of our research, we have determined that, in order to be effective, context-aware ubiquitous applications must be able to perform three major actions, which are illustrated in Figure 1:

1. on the left hand side, the application converts the real world into a symbolic representation;
2. in the center, the application uses that representation to determine the context of a situation and a course of action for the user;
3. on the right side, the application informs the user of the recommended course of action and/or supply him with information he requires.

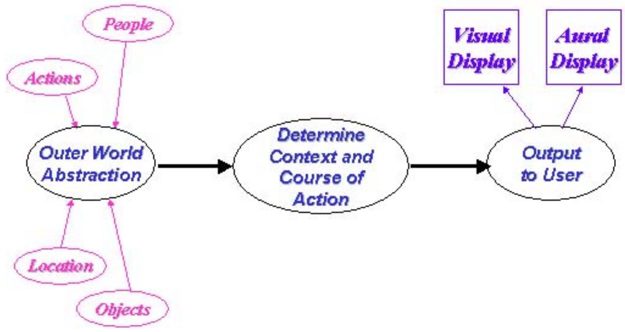

Figure 1 - Diagram of simple context-aware application

Notice that, of the three actions in Figure 1, only the center one is actually concerned with determining context. The left and right portions are concerned with issues secondary to the core issue of context. Therefore, all the time spent on those issues is time not spent on determining context.

Difficult Ancillary Problems

The left side of Figure 1 is a difficult problem because truly context-aware systems must use a significant number of real world attributes, some obvious such as location, time, user's schedule, or computing resources available, and many not so obvious, such as the physiological status of the user or environmental data. The difficulty lies in gleaning this information from the real world. The sensors involved in determining this information can be extremely difficult to create and use, often requiring frequent and/or extensive calibration.

The right side of Figure 1, outputting information so that the user can exploit it, is equally troublesome. While the data can be output using a wide range of techniques, most of these techniques involve emerging technologies. These include text to speech, augmented reality, and mobile/ wearable computers. Like all emerging technology, each of these can be difficult to use, and again much of the researcher's time is spent fixing secondary equipment.

Researchers who wish to limit their research strictly to context find themselves spending a great deal of their time on building, maintaining and using what should be only supplemental equipment to their actual investigation into context-aware computing. Additionally, much of this equipment is quite expensive, which forms another barrier to investigation.

Our Solution – A Commercial Game Engine

Dey approached the difficulty of dealing with several different types of these inputs and outputs by creating the Context Toolkit[2]. The Context Toolkit had widgets which allowed designers to abstract the method of determining information important for determining context. Our method to perform context-aware research without having to build the entire required infrastructure to support them was to use a high end game engine to simulate the real world for our context-aware application. The game engine performs many tasks important, yet secondary, to the actual building of the game itself. A game engine proved an exceptional way to simulate both the left and right sides of Figure 1, since its myriad of functions serves the same purposes in the game world.

We decided to use the UNREAL Game Engine, by Epic Games, because of its superb rendering, sophisticated artificial intelligence, advanced physics, and ease of use. This allowed us to test our context-aware applications while removing all the distractions of building systems which had to operate in the real world.

Many other researchers are using game engines to perform research in a wide variety of fields. Starner *et al* used them to create Augmented Reality applications[3], while Manninen also used the UNREAL engine to power a game which demonstrated contextual issues of a mobile application.[4] Our work is very similar to Bylund and Espinoza's, who used the Quake III engine to drive their context aware GeoNotes system via Dey's Context Toolkit.[5, 6]

Proof of Concept

In order to test our hypothesis as to the suitability of a game engine as a vehicle for building context-aware applications, we built a simple simulation of a Special Forces mission which was demonstrated at the Augmented Cognition Workshop in December 2001. While the mission simulated is significantly simpler than an actual mission would be, and the context-awareness of the simulation does not approach the final expectations, it serves as a proof of concept. Additionally, it effectively demonstrates several of the features of our application.

In this simple scenario, as the soldier carries out his mission, the context-sensing device helps him perform tasks by giving him directional guidance in the form of augmented reality arrows overlaid on the screen, pointing out important information that is either hidden or is difficult to notice, displaying the device's "degree of certainty" as to the identification of an object, and automatically recording important information. This allows the user to apply all brainpower to the task at hand.

Results

We were able to build an application with rudimentary context sensing simulating a soldier on a mission in six weeks using two programmers. The situations set up in the simulation adequately tested the context-sensing ability of the application. The information used to determine the context of the soldier's situation included his location and that of his enemies, ammunition status, mission status, and orders from higher authority.

We found that using a game engine has several of the advantages we had anticipated. Because the game engine keeps track of all information about every entity in the world, the application only has to query the engine to get any required information. Additionally, there were no delays due to malfunctioning equipment, which would have certainly been the case in a real world scenario.

However, there were some difficulties encountered while building the system. We discovered it was necessary to "dumb down" the information provided by the engine. Since the engine can provide perfect information about any object in the world, we found it necessary to limit the scope and accuracy of the information the engine delivered to more accurately reflect the information a soldier in the field would have. For example, the game engine provided the location of a hidden enemy, but that information was not provided to the context module.

Additionally, using a game engine is akin to learning a new programming language, and there was a learning curve involved for the programmers.

Finally, game engines are continually in flux, with the company often providing updates to their code. These updates are not always fully backwards compliant, and it is possible than an update will break existing code. When this happens, the programmers are required to debug the code, and many times these errors are not readily apparent.

REFERENCES

1. Weiser, M., Some Computer Science Issues in Ubiquitous Computing, *CACM, Vol 36*, Jul 1993.
2. Dey, A., *Providing Architectural Support for Building Context-Aware Applications*, PhD Dissertation, Georgia Institute of Technology, December 2000.
3. Starner, T. *et al*, Towards Augmented Reality Gaming, *Proceedings of IMAGINA 2000*, Monaco.
4. Manninen, T., Towards Communicative, Collaborative and Construtive Multi-Player Games, In *Proceedings of Computer Games and Digital Cultures Conference.* Mayra, F. (ed.), June 7-8, Tampere, Finland, 2002.
5. Espinoza, F., *et al*, GeoNotes: Social and Navigational Aspects of Location-Based Information Systems, *Proceedings of Ubicomp 2001*, Atlanta, Ga., 2001.
6. Bylund, M., & Espinoza, F., Testing and Demonstrating Context-Aware Services with Quake III Arena, *CACM, Vol. 45*, Jan 2002.