Computer Engineering Portfolio Overview

Suraj N. Kurapati

March 17, 2006

Abstract

This brief overview summarizes three undergraduate Computer Engineering projects which greatly influenced my present skills and abilities, strengthened my resilience in failure, and enhanced my team-work skills.

1 Battle against the Death Star (CMPE-118, Spring 2005)

The droid shown in Figure 1 was my final project for the very first *Introduction to Mechatronics* class taught at UCSC. The goal was to design and build "an autonomous machine that will destroy the most recently completed Death Star, using the Rebellion's deadliest weapon: small foam balls" [1].

(a) View from above.

(b) Facing view.

Figure 1: Photographs of the completed droid.

I contributed in the areas of conceptual design, physical construction (foam-core, hot glue, nails, wood, and duct-tape), mechanical actuation (motor driving

and solenoid firing), control logic (embedded software), and the signal conditioning. Undeniably, this project emphasized adaptability of skill in various disciplines.

2 Ultimate Death Match game (CMPS-115, Winter 2005)

This project gave me new insight of industrial project management practices. It involved lots of paper-work, including stringent specification of requirements, design reviews, life-cycle planning, end-user documentation, acceptance tests, and more. The goal was to build a real-time, multi-player game using the Java programming language (see Figure 2).

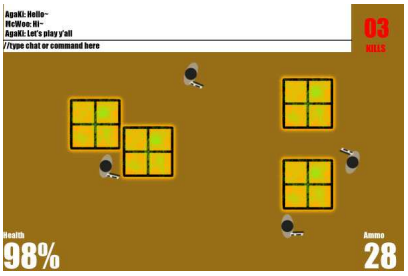

Figure 2: Conceptual illustration of our game.

We used Java 1.5 with RMI[1] for inter-process communication and BeanShell[2] for scripting. Furthermore, our project was generously hosted by the SOE DForge website, where we used the on-line Wiki to collaboratively author documents, the Subversion[3] repository to manage source code, and the WebDAV[4] to store files.

In addition to being the leader of this project, I contributed to the overall concept, modular decomposition, framework design, and authored nearly all of the code. The multi-disciplinary aspects of this project were team and project management, quality assurance, and technical writing.

3 64-bit port of SESC simulator (CMPE-202, Fall 2005)

This project aimed to port the Super ESCalar (SESC) architectural simulator [2] so that it compiles and runs natively on both 32- and 64-bit architectures.

[1] `http://java.sun.com/products/jdk/rmi`
[2] `http://www.beanshell.org`
[3] `http://subversion.tigris.org`
[4] `http://www.webdav.org`

I used Doxygen[5] to generate documentation, GDB for source-level debugging, and C99 constructs to improve portability (see Reference [3]), in terms of platform-independent storage types, and source code readability.

This project focused mainly on the discipline of software engineering. Through it, I experienced first-hand the importance of writing portable code and maintaining knowledge of the latest portable development practices.

References

[1] G.H. Elkaim, "The Death Star (Episode III: The Revenge of the Slugs)," [Online document], 2005 Spring, [cited 2006 March 14], Available HTTP: `http://www.soe.ucsc.edu/classes/cmpe118/Spring05/LectureNotes/StarWars.pdf`

[2] J. Renau, "SESC: cycle accurate architectural simulator," [Online document], 2005 December 28, [cited 2006 March 14], Available HTTP: `http://sesc.sourceforge.net`

[3] A. Jaeger, "Porting to 64-bit GNU/Linux Systems," in Proceedings of *GCC Developers Summit*, 2003, pp. 107–120. Ottawa, Canada.

[5] `http://www.stack.nl/~dimitri/doxygen/`