iPad: A Mobile Surgical Console

Computer Integrated Surgery II, Spring, 2011

Hanlin Wan and Jonathan Satria

Mentors: Balazs Vagvolgyi and Russell Taylor

Introduction

- Developed an application that allows the iPad to act as a configurable interface for multiple components of the EyeRobot.
- Eliminates inconvenience and clutter caused by multiple computers and associated peripherals.

Problem

- The current system for the EyeRobot requires several distinct components to control various settings such as visual magnification, brightness, and contrast. Each component is controlled through individual component windows and the entire system often requires multiple computers.
- This creates several problems:
  - Inconvenience due to the necessity of multiple computers and programs running at the same time.
  - Cluttered space in an already small workspace caused by multiple computers, monitors, keyboards, mice, etc. (figure 1)
  - Sterilization concerns over bloody gloves coming into contact with keyboards and mice, which are hard to clean afterwards.

Solution

- Compiled the necessary components (cisst, ICE) to be compatible with the iPad architecture (armv7).
- ICE makefiles were modified to remove unused components.
- cisst cmake files were modified to include compiler settings for the arm processor.
- Created an automated script that builds ICE, its dependencies, and cisst, installing the needed libraries and header files to a single location.
- Created an iPad application that interfaces with the Scenario Manager. (figure 2)
  - Reduces the number of computers needed to a single one - the visualization console.
  - Controls various settings of the Visualization Console and OCT Console such as brightness and annotations.
  - Two panel design, finger swipes, and a favorites bar allow for easy navigation between components.
- Detailed tutorial for users to implement additional features in the future.
- Setting up XCode with the necessary libraries.
- Linking actions to commands.

Figure 1: Multiple computers and connected peripheries take up a large portion of an already small workspace.

Figure 2: Screenshot of the Visualization Console module which can adjust settings and turn on/off features.

Outcomes and Results

- We succeeded in creating an easy-to-use iPad application that can easily interface with the Surgical Assistant Workstation.
- Improved ease and efficiency during animal testing.
- "Yesterday was a great success in terms of the iPad integration into the surgical system. It made a huge difference during the experiment." – Balazs

Future Work

- We will not have an active role in furthering the project, but will be happy to answer any questions.
- Our tutorials will be a good framework for further development.
- We leave the project in the hands of Balazs and Marcin, whom have successfully created applications that interfaces with cisst and ICE.

Lessons Learned

- Compiling of cross-platform software can be very difficult to get working properly.
- Learned how to create and debug iPad applications.

Credits

- Hanlin Wan – Did much of the work getting cisst, ICE, and iOS to compile and link properly; Created the automated script for easy compilation and wrote tutorial for manual compilation. Wrote primary GUI design.
- Jonathan Satria – Worked on GUI design and programming; Wrote the tutorials for developing iOS application that integrates cisst and ICE.

Support by and Acknowledgements

- Core NSF CISST/ERC
- Thank you to our mentors Balazs Vagvolgyi and Dr. Russell Taylor for helping us through with the project.
- Thank you to Dr. Russell Taylor for providing the funds needed to acquire the iPads.

Engineering Research Center for Computer Integrated Surgical Systems and Technology