



ERC | CISST



LABORATORY FOR  
Computational  
Sensing + Robotics  
THE JOHNS HOPKINS UNIVERSITY

# iPad Mobile Surgical Console

Hanlin Wan

Jonathan Satria

Mentors: Balázs P Vágvolgyi,  
Dr. Russell Taylor

# Background & Motivation

- Multiple computers in surgical OR for configuration
  - Video, lighting, overlays, etc.
- Cluttered space
- Inconvenient operation
  - Mouse & keyboard input
  - Decentralized
- Sterilization concerns



# Project Goals

- Application for centralized control
- GUI for easy systems configuration
- Touchscreen ability - easy to use, easy to clean
- Ultimately, iPad application to control multiple consoles from a mobile unit

# Timeline

- Project on schedule

	February		March			April				May			
Task	1	2	3	4	5	6	7	8	9	10	11	12	
Install Cmake	Done				Spring Break								
Compile CISST		Done											
Install ICE			Done										
iOS Interface Documentation			Done										
Familiarize with iOS Development			Done										
Build GUI for Components							In Progress						
GUI/Components Interface							In Progress						
GUI Documentation													
GUI Revisions													
Mock OR Testing													
Project Documentation													

# Compiling cisst for iOS

- Integrated with cmake
- Compiled with ICE flag
- Easily imported into XCode
- Well documented guide for compilation

# Using Cmake

- CMakeList.txt includes iphone.cmake
- iphone.cmake adds/modifies compiler settings specific for iOS
- Cmake file generated and compiled as usual

# Cmake Settings

```
Terminal — bash — 113x53
Page 1 of 4
BUILD_DOC OFF
BUILD_EXAMPLES OFF
BUILD_LIBS_cisst3DUserInterfac OFF
BUILD_LIBS_cisstCommon ON
BUILD_LIBS_cisstDevices OFF
BUILD_LIBS_cisstInteractive OFF
BUILD_LIBS_cisstMultiTask ON
BUILD_LIBS_cisstNumerical OFF
BUILD_LIBS_cisstOSAbstraction ON
BUILD_LIBS_cisstParameterTypes OFF
BUILD_LIBS_cisstRobot OFF
BUILD_LIBS_cisstStereoVision OFF
BUILD_LIBS_cisstVector ON
BUILD_TESTS OFF
BUILD_UTILITIES OFF
CISST_BUILD_SHARED_LIBS OFF
CISST_CLEAR_INTERNALS OFF
CISST_HAS_CMAKE_DEBUG OFF
CISST_HAS_FLTK OFF
CISST_HAS_OPENGL OFF
CISST_HAS_QT OFF
CISST_HAS_SWIG_PYTHON OFF
CISST_HAS_XML OFF
CISST_MTS_HAS_ICE ON
CMAKE_AR /Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/ar
CMAKE_BUILD_TYPE
CMAKE_COLOR_MAKEFILE ON
CMAKE_CXX_COMPILER /Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/g++-4.2
CMAKE_CXX_FLAGS -x objective-c++
CMAKE_CXX_FLAGS_DEBUG -x objective-c++;-DDEBUG=1;-ggdb
CMAKE_CXX_FLAGS_MINISIZEREL -Os -DNDEBUG
CMAKE_CXX_FLAGS_RELEASE -x objective-c++;-DNDEBUG=1
CMAKE_CXX_FLAGS_RELWITHDEBINFO -x objective-c++;-DNDEBUG=1;-ggdb
CMAKE_C_COMPILER /Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/gcc-4.2
CMAKE_C_FLAGS -std=c99;-x objective-c
CMAKE_C_FLAGS_DEBUG -std=c99;-x objective-c;-DDEBUG=1;-ggdb
CMAKE_C_FLAGS_MINISIZEREL -Os -DNDEBUG
CMAKE_C_FLAGS_RELEASE -std=c99;-x objective-c;-DNDEBUG=1
CMAKE_C_FLAGS_RELWITHDEBINFO -std=c99;-x objective-c;-DNDEBUG=1;-ggdb
CMAKE_EXE_LINKER_FLAGS
CMAKE_EXE_LINKER_FLAGS_DEBUG
CMAKE_EXE_LINKER_FLAGS_MINISIZE
CMAKE_EXE_LINKER_FLAGS_RELEASE
CMAKE_EXE_LINKER_FLAGS_RELWITH
CMAKE_FIND_ROOT_PATH /Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS4.2.sdk
CMAKE_INSTALL_NAME_TOOL /Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/install_name_tool

BUILD_DOC: Build the documentation.
Press [enter] to edit option
Press [c] to configure
Press [h] for help
Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently On)
CMake Version 2.8.4
```

```
Terminal — bash — 113x53
Page 2 of 4
CMAKE_INSTALL_PREFIX ./ipadLib
CMAKE_LINKER /Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/ld
CMAKE_MAKE_PROGRAM /opt/local/bin/gmake
CMAKE_MODULE_LINKER_FLAGS
CMAKE_MODULE_LINKER_FLAGS_DEBU
CMAKE_MODULE_LINKER_FLAGS_MINS
CMAKE_MODULE_LINKER_FLAGS_RELE
CMAKE_MODULE_LINKER_FLAGS_RELW
CMAKE_NM /Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/nm
CMAKE_OBJCOPY CMAKE_OBJCOPY-NOTFOUND
CMAKE_OBJDUMP CMAKE_OBJDUMP-NOTFOUND
CMAKE_OSX_ARCHITECTURES armv7
CMAKE_OSX_DEPLOYMENT_TARGET
CMAKE_OSX_SYSROOT /Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS4.2.sdk
CMAKE_RANLIB /Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/ranlib
CMAKE_SHARED_LINKER_FLAGS
CMAKE_SHARED_LINKER_FLAGS_DEBU
CMAKE_SHARED_LINKER_FLAGS_MINS
CMAKE_SHARED_LINKER_FLAGS_RELE
CMAKE_SHARED_LINKER_FLAGS_RELW
CMAKE_SKIP_RPATH OFF
CMAKE_STRIP /Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/strip
CMAKE_SYSTEM_PROCESSOR arm
CMAKE_USE_RELATIVE_PATHS OFF
CMAKE_VERBOSE_MAKEFILE OFF
CMN_ASSERT_DISABLED OFF
CMN_ASSERT_THROWS_EXCEPTION OFF
CMN_THROW_DOES_ABORT OFF
DEVROOT /Developer/Platforms/iPhoneOS.platform/Developer
FLTK_BASE_LIBRARY FLTK_BASE_LIBRARY-NOTFOUND
FLTK_CONFIG_SCRIPT FLTK_CONFIG_SCRIPT-NOTFOUND
FLTK_DIR FLTK_DIR-NOTFOUND
FLTK_FLUID_EXECUTABLE FLTK_FLUID_EXECUTABLE-NOTFOUND
FLTK_FORMS_LIBRARY FLTK_FORMS_LIBRARY-NOTFOUND
FLTK_GL_LIBRARY FLTK_GL_LIBRARY-NOTFOUND
FLTK_IMAGES_LIBRARY FLTK_IMAGES_LIBRARY-NOTFOUND
FLTK_INCLUDE_DIR FLTK_INCLUDE_DIR-NOTFOUND
FLTK_MATH_LIBRARY /usr/lib/libm.dylib
ICE_FOUND ON
ICE_HOME /opt/local
ICE_ICE_H_INCLUDE_DIR /opt/local/include
ICE_LIBRARY_NAME_ZEROC_ICE /opt/local/lib/libZeroCice.dylib
ICE_SLICE2CPP /opt/local/bin/slice2cpp
LIBXML2_INCLUDE_DIR /opt/local/include/libxml2
LIBXML2_LIBRARIES /opt/local/lib/libxml2.dylib
LIBXML2_XMLINT_EXECUTABLE /opt/local/bin/xmlint

CMAKE_INSTALL_PREFIX:
Press [enter] to edit option
Press [c] to configure
Press [h] for help
Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently On)
CMake Version 2.8.4
```

# XCode Project

The screenshot displays the Xcode IDE interface for a project named "cisstIPad". The left sidebar shows the "Groups & Files" view, with folders for "ipadlib-simulator", "Classes", "Other Sources", and "Resources". The main editor area shows the "File Name" list, with files like "AppDelegate.mm", "Foundation.framework", and "libcisstCommon.a" (highlighted with a red box). The "Target Info" window is open, showing the "Build" tab with search paths for headers and libraries. The "Header Search Paths" and "Library Search Paths" are both set to "/Users/cisstmacbook/Code/cisst/cisstIPad/./ipadlib-simulator/include" and "/Users/cisstmacbook/Code/cisst/cisstIPad/./ipadlib-simulator/lib" respectively, both highlighted with red boxes. The "main.mm" file is open in the editor, showing the start of the main function.

Target "cisstIPad\_Simulator" Info

Configuration: Active (Debug) Search in Build Settings

Show: All Settings

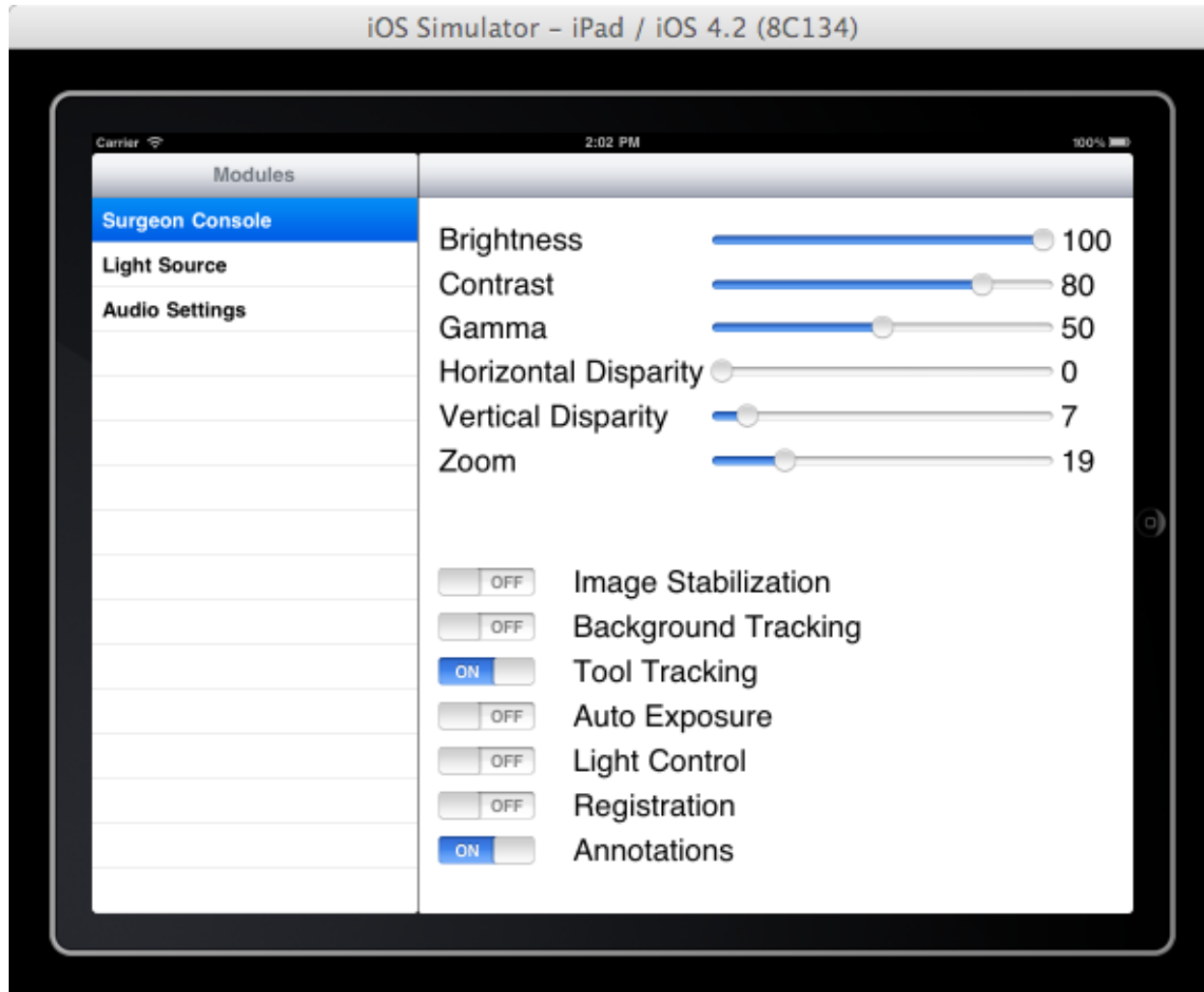
Setting	Value
Framework Search Paths	
Header Search Paths	"/Users/cisstmacbook/Code/cisst/cisstIPad/./ipadlib-simulator/include"
Library Search Paths	"/Users/cisstmacbook/Code/cisst/cisstIPad/./ipadlib-simulator/lib"
Rez Search Paths	
Sub-Directories to Exclude in Recursive Searches	*.nib *.lproj *.framework *.gch (*) CVS .svn .git *.xcodproj *.xcode *.pbpro...
Sub-Directories to Include in Recursive Searches	
User Header Search Paths	
Unit Testing	
Other Test Flags	
Test Host	
Test Rig	
Versioning	
Current Project Version	
Generated Versioning Source Filename	cisstIPad_vers.c
Generated Versioning Variables	
Versioning Name Prefix	
Versioning Name Suffix	
Versioning System	None
Versioning Username	cisstmacbook
GCC 4.2 - Code Generation	

```
#import <UIKit/UIKit.h>

int main(int argc, char *argv[]) {
    int retVal = UIApplicationMain(argc, argv, nil, nil);
    return retVal;
}
```



# GUI Development Started



# Review What's Done

- Milestone 1 Complete
  - Cmake integration of iOS done
  - ICE compiled
  - XCode project created
  - Well documented installation guide
- Milestone 2 Started
  - Familiar with iOS GUI development
  - Sample GUI interface built

# What's Next?

- Continued GUI development
- Interface with module control

	February		March			April				May			
Task	1	2	3	4	5	6	7	8	9	10	11	12	
Install Cmake	Done				Spring Break								
Compile CISST		Done											
Install ICE			Done										
iOS Interface Documentation			Done										
Familiarize with iOS Development			Done										
Build GUI for Components							In Progress						
GUI/Components Interface							In Progress						
GUI Documentation													
GUI Revisions													
Mock OR Testing													
Project Documentation													

# Plans for EyeRobot Control

<b>Task</b>	<b>Target Date</b>
Actual control of Video Brightness setting for EyeRobot	4/8/11
GUI development for remaining components	4/8/11
Tutorial for integration of GUI and component controls on EyeRobot	4/11/11
Implementation of all other components for EyeRobot control	4/22/11
Add error checking to all components	4/29/11
Tutorial for handling error checking	5/2/11
Implement GUI changes based on feedback	5/6/11

# Deliverables

- **Minimum: (No longer applicable)**
  - Use iPad as a dummy console to VNC into a computer.
  - Create a GUI system to control the multiple components through the Scenario Manager.
- **Expected:**
  - Compile and build the cisst library on the iPad using CMake. **(completed)**
  - Build a GUI application for the iPad to control the various components. **(in progress)**
  - Perform clinical tests of the iPad control system in a mock OR setting. **(previously maximum)**
  - Perform revisions to the GUI based on user feedback. **(previously maximum)**
  - Detailed tutorial for iOS-cisst interface **(new expected)**
- **Maximum: (new maximum)**
  - Implement additional features

# New Maximum Deliverables

- Explore other uses of iPad
  - **Video Streaming** of live robot feed
  - **Accelerometer Use** to move robotic arms
  - **Microphone Use** to interface with voice control
- Other suggestions?

# Thank You

## Questions?