WiiLab: Bringing Together the Nintendo Wiimote and MATLAB

Jordan Brindza, Jessica Szweda, Qi Liao, Yingxin Jiang, Aaron Striegel
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556 USA
brindza@seas.upenn.edu, jessica.szweda@centre.edu
qliao@nd.edu, yjiang3@nd.edu, striegel@nd.edu

Abstract - **It is often a daunting task for engineering students to learn programming. Learning by playing has proven to be an effective way to engage students at nearly all levels of the curriculum. In this work, we will describe how we developed modules for teaching MATLAB to freshmen engineers through the use of the Nintendo Wiimote and motion capture. Through the work of a summer REU project, we developed and prototyped a unified set of C# and MATLAB libraries for Windows entitled WiiLab. In particular, the work focused on creating an easy-to-use API that largely abstracted the complexity of interacting with the Wiimote. In the work, we describe several laboratory modules developed for WiiLab and experiences from using the Wiimote in a four hundred student freshmen engineering course.**

Index Terms – Engineering Education, MATLAB, Nintendo Wii, Programming

INTRODUCTION

For many of the grand challenges facing society today, the solutions often encompass a significant degree of computational exploration and validation. Critically, many of the grand challenges of science and, in a larger sense, society require the ability to think in a computational manner. To that end, education with respect to computational thinking becomes a critical lynchpin for tackling said grand challenges.

As a result, the problem of how to teach programming and, in particular, introductory programming has become more important than ever. Approaches to improve the introductory programming experience vary significantly with considerable attention paid to both motivating the need for computational understanding and presenting programming in intuitive, immersive environments [1-13].

The recent introduction of inexpensive, robust motion capture devices, such as with the Nintendo Wii, affords exciting new opportunities for innovation in this particular space. We posit that a toolkit that allows the educator and student to fuse multiple senses (motion, vision, auditory, touch) could unleash tremendous creativity, both with respect to educational and student creativity. In this paper, we focus on efforts at the University of Notre Dame that bring a simple, effective interface to MATLAB for the Nintendo Wiimote. We present the motivations and technical aspects of Wiilab and relate how the toolkit was employed for an introductory freshmen engineering course (EG 10111).

COURSE BACKGROUND

At the University of Notre Dame, all engineering students take two introductory engineering courses consisting of four modules with one module focusing on computer programming. Other modules are left to the discretion of the module instructor with topics ranging from statistics to nanotechnology to numerical methods.

The course description for EG 10111, the first in the two course sequence, is as follows:

> *This is the first of a two-part sequence to introduce engineering to First Year intents and to establish a foundation for their studies in any of the engineering disciplines. Group-orientated design projects are used to provide a multidisciplinary view of engineering systems and to present the engineering method. Computing skills are developed for engineering analysis, synthesis, and technical communication.*

The three credit course has three weekly meetings consisting of two lectures and one learning center laboratory session. Each module in the course lasts eight weeks and culminates in a group project on the particular module focus.

MATLAB PROGRAMMING MODULE

For the freshmen engineering sequence, one module of the course is devoted to computer programming. Due to the fact that all engineering majors take the course, MATLAB was selected as it was perceived to have the broadest impact across all engineering disciplines. For nearly all disciplines outsides of Computer Science / Engineering and Electrical Engineering, significant course work involves future application of MATLAB. Key programming concepts for the module include: general MATLAB syntax, variables, arrays, control logic, looping, and the usage of functions.

In general, lecture material covers MATLAB concepts through detailed examples of each key topic. Lecture material is supplemented through homework as well as the smaller learning center sections where concepts are directly put into practice. Students are required to work individually for homework but are allowed to work as pairs for the learning center modules.

Despite its limited timeframe, the module is quite aggressive in order to culminate in a project covering the target concepts for the module. The typical final project involves a graphical simulation with semi-realistic movement in order to satisfy various success and failure criterion. Previous projects have ranged from control of a UAV across a flood plain (see Figure 1) to docking of the Orion module at the International Space Station to piloting a lunar lander on a randomly generated moonscape.

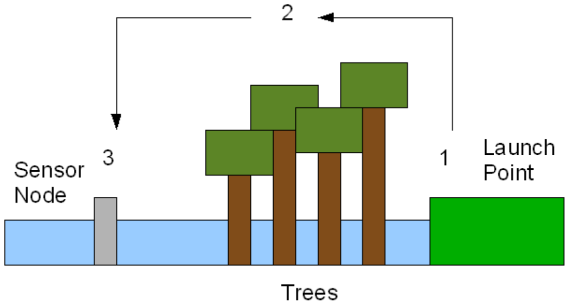

Figure 1 – Fall 2008 MATLAB module final project – UAV piloting

While the module is successful at delivering its core content, the large lecture and limited timeframe can be quite difficult for students. Moreover, this difficulty is only compounded by the fact that the students are still largely adjusting to life at college. Recent changes to the course to improve course dynamics include:

- *Pair-wise final project*: In previous offerings, the final project was done individually. In fall 2008, students were allowed to work as pairs for the final project with significant positive student reaction to the change.
- *Enhanced campus license:* A recent campus license agreement has made the full set of relevant MATLAB functionality available to all students and faculty on the Windows and Mac platforms.

Moreover, as the course encompasses all engineering majors, there is significant variance in the level of experience that students have with computing. The vast majority of the students taking the course have little to no programming experience making this module the first exposure to programming. When this fact is coupled with a general aversion of non-computer science students to programming, we felt there was a need to move beyond the traditional focus on "syntactic sugar" of MATLAB but still preserve the utility of MATLAB for later years of the students' education.

To that end, we identified recent work with the Nintendo Wiimote libraries as a potential new mechanism for teaching MATLAB content. The core idea would be that the Wiimote could both enhance lecture as well as act as an extended project outlet for those wishing to pursue it. Beyond having a general familiarity to many students, the Wiimote itself offered significant potential for creativity in terms of module design and execution.

BACKGROUND – NINTENDO WIIMOTE

The Nintendo Wii Remote (aka the Wiimote - see Figure 2) was introduced with considerable fanfare with the Nintendo Wii in the fall of 2006 to North America. While the controls of the remote itself were largely unremarkable, the intuitive nature of the motion capture control was unparalleled in consumer-level electronics. The free inclusion of the Wii Sports game was an instant hit, notably due to the mass appeal afforded by the unique interface.

Video games that in the past had required esoteric key combinations could now be played by mimicking the actions of the actual activity, whether that activity was golf, baseball, tennis, or bowling. In short, one's life experience in those particular activities were the only knowledge required (also available in game through on-screen instructions) with feedback through visual cues being instantaneous and often immensely gratifying. The inclusion of motion capture in large part bridged a tremendous generational gap no longer requiring extensive manuals nor training to use the device. The cross-generational appeal was not lost on the manufacturer (Nintendo) which frequently employed ads showing senior citizens or entire families enjoying video games that were once the near exclusive domain of younger, more technology savvy individuals.

The Wiimote itself inspired a large amount of external third party development to adapt the Wiimote to various other activities due in part to the relatively open Bluetooth interface of the Wiimote and its exceptionally low cost ($40 each). In contrast to significantly more expensive devices employed in model motion capture and beyond (Flock of Birds [14], etc.), the capabilities of the Wiimote were cheaper by multiple orders of magnitude, yet obtains significant robustness albeit with limited accuracy. For the cost of the Wiimote itself, a Bluetooth adapter (often included with new computers), and open source tools, one can easily explore motion capture. Hence, a wide variety of applications and toolkits emerged including notable examples such as the GlovePIE toolkit [15], low cost smart board

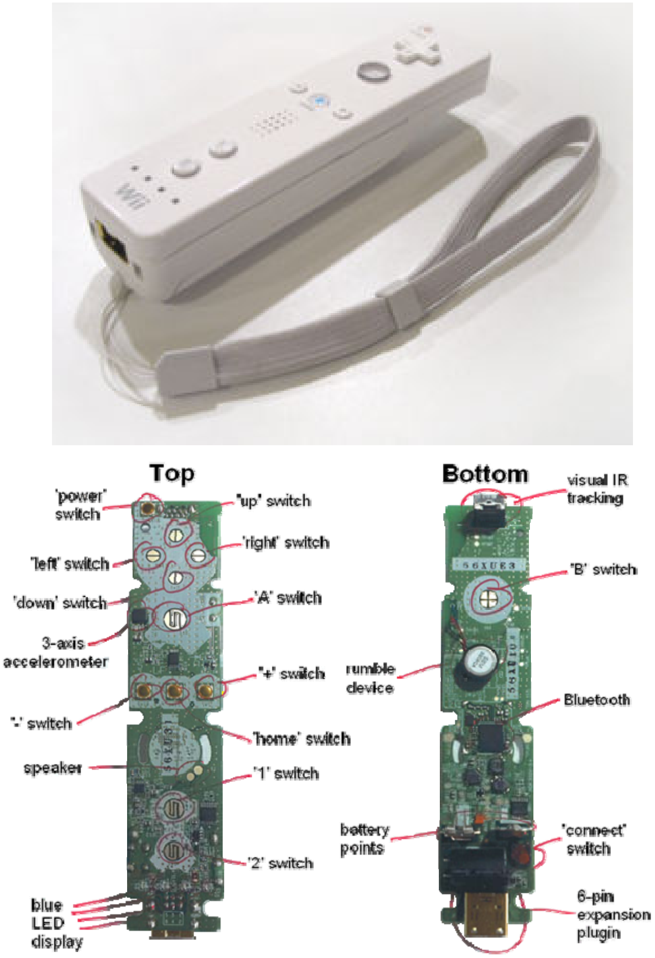

Figure 2 – Nintendo WiiMote

approaches (undergraduate project at CMU) [16], and adaptations of other devices for motion input to the Wii (acceleration controller on IBM Thinkpad notebooks, etc.). Most recently, an extension to the Wiimote will be released later in the year that will significantly improve the accuracy of motion capture at roughly the equivalent cost of the Wiimote itself.

Capabilities of the Wiimote include sensing acceleration in the general X, Y, and Z directions, sensing positions if an IR sensor bar is used, rumble feedback via vibration, and finally the ability to play limited audio clips through the speaker of the Wiimote. An upcoming addition to the Wiimote in the summer of 2009 should also dramatically improve the acceleration precision of the Wiimote allowing for significantly improved motion detection.

While the majority of such functionality was available through the various open source tools, the ease of interaction with MATLAB for an experienced programmer, let alone a freshmen starting out in programming was quite limited. To that end, two summer REU (Research Experience for Undergraduates) students were tasked with creating an intuitive, easy to use mechanism for using the Wiimote in MATLAB. The students were supervised by two graduate students as well as the REU co-PI, Dr. Striegel. The eventual result of the summer projects was WiiLab, an open-source suite of tools freely available for download at [17].

WIILAB: WII REMOTE AND MATLAB

The WiiLab project had several functional requirements, namely that it needed to run on the existing equipment of the Engineering Learning Center (Windows XP, MATLAB R2008a), to run with extremely simplified functions for gathering / reporting data from the Wiimote, and to run with the capacity for visual, audio, and haptic (rumble) feedback.

Over the course of the summer, the students used a partially working set of open source C# code by Brian Peek [18] to adapt the code into a Windows COM service with bridges to MATLAB. A full suite of helper functions allowing one to poll, discover, and interact with the Wiimote for various acceleration values were written together with numerous blocks of example code including Pong, the Inverted Pendulum problem (see Figure 3), a bouncing ball, and a spinning wheel.

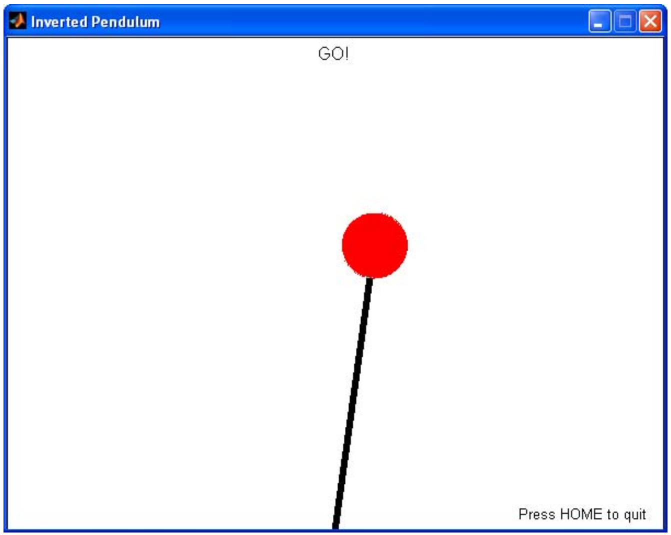

Figure 3 – Inverted Pendulum Application

The key focal point of the project was to enable simple data collection that could be plugged into control loops in examples or code by the student. For instance, once a workstation was validated through the external test application, the MATLAB code for connectivity was largely hidden from the user, requiring only a simple initialization function:

```
% connect the Wiimote
initializeWiimote();
```

Once properly initialized, the programmer can then poll the Wiimote for its acceleration or drive its visual output (LEDs) or haptic output (rumble) to interact. The API itself defaults to connecting to the first Wiimote connected rather than forcing the user to choose from a list of Wiimotes already connected. Advanced internal functionality allows for multiple Wiimotes to be connected but the base API does not require such a specification.

```
% pressing the Wiimote's HOME button will terminate the program
 while (~isButtonPressed('HOME'))
    % want to rotate the inverse pendulum based on the Wiimote's
    % x-acceleration
    theta = -2*getWiimoteAccel();

    % obtain the mass's current x-coordinate
    XCMass = getCenter(mass);
    % account for gravity by increasing the angle of rotation as the
    % pendulum gets further from its equilibrium point
    theta = theta - 0.08*(XCMass - center);

    % use a basic trigonometric function to calculate the pendulum's
    % current angle of elevation
    beta = acos((XCMass-center)/(length + 20));
    % determine what the new angle of elevation will be upon rotation
    alpha = beta - deg2rad(-1*theta);
    % use the new angle to determine what the mass's height will be
    newYCMass = (length + 20)*sin(alpha);
```

Figure 4 – Example Code – Inverted Pendulum

The connection API for the Wiimote is as follows:

- `initializeWiimote():` Attempt to connect to a default Wiimote.
- `isWiimoteConnected():` Check to make sure that the Wiimote is still connected.
- `disconnectWiimote():` Stop connecting to the Wiimote.

The Wiimote itself can be sampled for its state through the following API:

- `isButtonPressed(Button):` Determine if one of the buttons (A, B, direction, home, 1, 2) is pressed.
- `getWiimoteAccel():` Retrieve the acceleration in the x, y, and z directions.
- `getWiimoteIR():` Retrieve the current IR location information.

Finally, feedback can be directed at the Wiimote through the feedback API:

- `setWiimoteLEDs(led1,led2,led3,led4):` Light or turn off any of the four Wiimote LEDs
- `setWiimoteRumble(On):` Turn the Wiimote rumble on or off

Although the Wiilab toolkit does include the ability to send audio through the Wiimote, the limited ability of MATLAB for threading and the general difficulty level associated with playing audio relegated it to the advanced API listed in the documentation for Wiilab.

APPLICATION – EG 10111 – FALL 2008

The fall semester of 2008 was the first application of the Wiilab code into the classroom. Although the software was quite robust, deployment and logistics issues in extending the learning center to support four hundred students prevented deployment for the fall after development only finished in August of 2008.

However, the code itself was used successfully for the lecture component of the course. In the course, several lectures were complimented with the use of the Wiimote to demonstrate concepts including control logic, looping, and basic graphics. The highlight of the usage as noted by several students on their course feedback was the lecture bringing in the Showcase Showdown from the Price is Right also known as the giant spinning wheel.

The beginning of the lecture featured the task at hand for the learning center which involved using only the keyboard. For the lecture, however, a special version was developed using the Wiilab software along with additional logic to have students compete. For the demonstration, three students were selected to come down and spin the wheel in front of the 40 foot projection screen. The demonstration was complete with introduction music, the horns for exceeding the target amount of $1.00, and logic to determine the winner. Students were encouraged to examine the results and explore how the code operated of which many elected to pursue.

CONCLUSION

In conclusion, the Wiilab software offers tremendous potential for bringing Wiimote programming to MATLAB. The platform offers a low-cost, intuitive approach to experiment with motion capture throughout the curriculum. It is our hope that others will expand upon this work and contribute further examples to our on-line code repository. Moreover, we strongly believe that the use of intuitive actions such as motion can offer significant appeal to broaden the interest of students into computer programming.

ACKNOWLEDGMENT

This work in part was supported in part by the National Science Foundation through the grants CNS03-47392 and CNS07-54933.

REFERENCES

[1] Stephen Cooper, Wanda Dann, Randy Pausch, “Teaching Objects-first In Introductory Computer Science,” SIGCSE 2003.

[2] C. Egert, K. Bierre, A. Phelps, P. Ventura, “Hello, M.U.P.P.E.T.S.: using a 3D collaborative virtual environment to motivate fundamental object-oriented learning,” In Companion To the 21st ACM SIGPLAN Symposium on Object-Oriented Programming Systems, Languages, and Applications (Portland, Oregon, USA, October 22 - 26, 2006). OOPSLA '06. ACM, New York, NY, 881-886.

[3] M. Guzdial, and E. Soloway, “Teaching the Nintendo generation to program,” Commun. ACM 45, 4 (Apr. 2002), 17-21, 2002.

[4] D. Hagan, S. Markham, “Does it help to have some programming experience before beginning a computing degree program?” SIGCSE Bull. 32, 3 (Sep. 2000), 25-28.

[5] J. Hamer, “An approach to teaching design patterns using musical composition,” in Proceedings of the 9th Annual SIGCSE Conference on innovation and Technology in Computer Science Education (Leeds, United Kingdom, June 28 - 30, 2004). ITiCSE '04. ACM, New York, NY, 156-160

[6] F. Klassner, S.D. Anderson, “LEGO Mindstorms: Not Just for K-12 Anymore,” IEEE Robotics and Automation Magazine, vol. 10, no. 2, pp. 12-18, 2003.

[7] M Kölling, J Rosenberg , “BlueJ–The Hitch Hiker’s Guide to Object Orientation,” Journal of Object-Oriented Programming, June, 2001.

[8] Barbara Moskal, Deborah Lurie, Stephen Cooper, “Evaluating the effectiveness of a new instructional approach”, in Proc. of SIGCSE, 2004.

[9] N. J. Mourtos, B. J. Furman, “Assessing the Effectiveness of an Introduction to Engineering Course for Freshmen,” in Proc. Of Frontiers of Education (FIE), 2003.

[10] Randy Pausch, Tommy Burnette, A.C. Capeheart, Matthew Conway, Dennis Cosgrove, Rob DeLine, Jim Durbin, Rich Gossweiler, Shuichi Koga, Jeff White “Alice: Rapid Prototyping System for Virtual Reality,” IEEE Computer Graphics and Applications, May 1995.

[11] M. Pedroni, B. Meyer, ”The inverted curriculum in practice”. SIGCSE Bull. 38, 1 (Mar. 2006), 481-485.

[12] M. Resnick, A. Bruckman, F. Martin, “Pianos not stereos: creating computational construction kits”, Interactions 3, 5 (Sep. 1996), 40-50.

[13] S. Wiedenbeck, “Factors affecting the success of non-majors in learning to program,” in Proceedings of the First international Workshop on Computing Education Research (Seattle, WA, USA, October 01 - 02, 2005). ICER '05. ACM, New York, NY, 13-24.

[14] Ascension Technology, Flock of Birds, http://www.ascension-tech.com/realtime/RTflockofBIRDS.php

[15] GlovePIE, http://carl.kenner.googlepages.com/glovepie_download

[16] Johnny Chung Lee, Wii Projects, http://johnnylee.net/projects/wii/

[17] WiiLab, http://netscale.cse.nd.edu/Wiilab

[18] Peek, Brain. (2007, March). Managed Library for Nintendo's Wiimote. Coding4Fun. Retrieved June 2008, from http://blogs.msdn.com/coding4fun/archive/2007/03/14/1879033.aspx.

AUTHOR INFORMATION

Jordan Brindza Student / NSF REU Site Participant, University of Pennsylvania, brindza@seas.upenn.edu

Jessica Szweda Student / NSF REU Site Participant, Centre College, jessica.szweda@centre.edu

Qi Liao Graduate Student, University of Notre Dame, qliao@nd.edu

Yingxin Jiang Graduate Student, University of Notre Dame, yjiang3@nd.edu

Aaron Striegel Associate Professor, University of Notre Dame, striegel@nd.edu