



CSM4

Rawan Bardini

Zach Fager

Jeff Neet

Client: John Jackson

Advisor: Cyndi Rader

CS Field Session 2009

I. Abstract

Female enrollment in the field of computer science is declining. Demonstrations of cutting-edge technology can be used to promote interest and increase enrollment. The incorporation of Wii Remote technology targets these demonstrations toward high school students. This technology allows for the creation of visually-oriented, interactive demonstrations that showcase the relationships between hardware, software, and the physical world. Students can experience a hands-on example of computer science and software engineering in action.

II. Introduction

Female enrollment in the field of computer science is declining both at the Colorado School of Mines and nationwide. CSM is running technology camps and outreach clubs to attract middle and high school students to the field of computer science. John Jackson, the active leader of these activities, has been awarded a grant to promote enrollment by developing visually stimulating and interactive computer science based activities. This demonstration should encourage young students to pursue interests in the computer science field, and potentially enroll at CSM.

III. Requirements

The "Wii Love Programming" software has been designed to incorporate Wii Remote technology into an educational setting, where students will not only enjoy working with the Wii Remote, but will begin to understand programming. The students will use a Wii Remote to control a video game on the computer where the controls are scrambled. Students are then instructed to use the program to help redesign the controls on the Wii Remote to become more intuitive. Students will have the ability to explore and modify the underlying programming code. The program is visually appealing and easy to use, allowing the students to easily understand how to redesign the controls.

i. Functional Requirements

1. Create GlovePIE scripts through Java
2. Launch video game and GlovePIE from Java
3. Provide options to program and play different games
4. Display programmable buttons/swing features of the Wii Remote
5. Display all possible game controls for each button/swing feature
 1. Display script code line(s) for chosen control
6. Ability to preserve controls and create script
7. Functionality to launch the game with current script
8. View existing GlovePIE script as it is being created
9. Edit script manually via interface
10. Provide 'Cheater Mode' option
11. Start over feature that sets controls back to default
12. Save and Load features to preserve script controls

ii. Non-Functional requirements

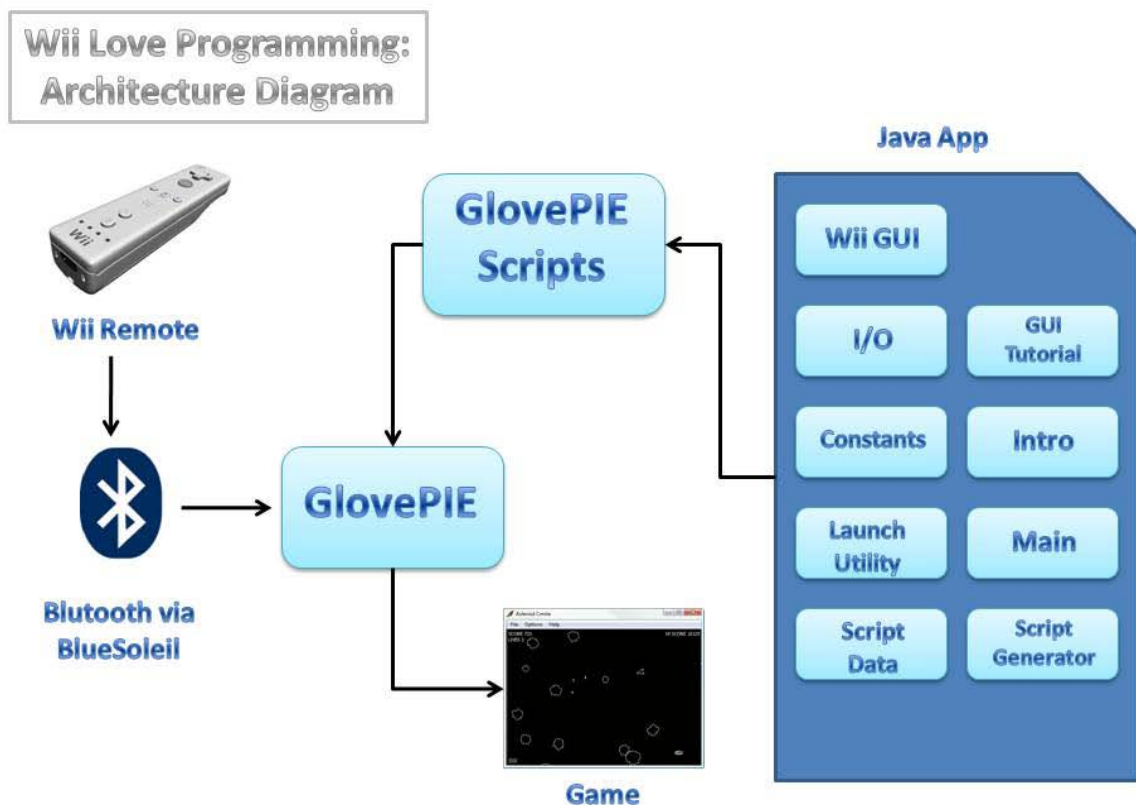
1. Use Wii Remote technology for input source
2. Provide visually oriented user interface
3. Complete by June 17th 2009
4. Use only open-source software
5. Allow for various activities
 1. Video games
 2. Internet Browser
 3. Flash games
6. Promote interest for Computer Science

IV. Design

i. High-level Design

The Wii Love Programming software features a Java-based user interface that interacts with the Wii Remote, Bluetooth wireless technology, GlovePIE scripting technology, and external video games. An architecture diagram has been included (Figure 1).

Figure 1: Architecture Diagram



Wii Remote

The Wii Remote is a handheld Bluetooth device that allows users to interact with software via an accelerometer.

BlueSoleil

Bluetooth is a very common form of short range wireless technology. To facilitate the interface between the Wii Remote and the computer system, software called BlueSoleil is used.

GlovePIE

GlovePIE stands for Glove Programmable Input Emulator. The software uses GlovePIE to capture data from the Wii Remote and store this data in a file. GlovePIE runs user defined scripts, and a batch file can be utilized to launch a predefined script.

External Video Game Software

The Wii Love Programming software interacts with external video games.

ii. Detail Design

1. Modules

1.1. Wii Remote Module

In conjunction with the software, a Wii Remote (Figure 2) operates the game being played. The Wii Remote connects to the computer via Bluetooth, allowing the student to interact with the game through movement via the accelerometer and buttons. Eleven out of the twelve buttons on the Wii Remote can be reprogrammed through the software. The 'Power' button is not programmable due to the risk of turning off the entire Wii Remote.

Additionally, the students can use the accelerometer controls in the game. They have the option of using six different swing motions. These motions are up, down, left, right, rotate left, and rotate right. The swing motions also have a property that the students will set that determines the amount of force needed when swinging the Wii Remote to activate the accelerometer control.

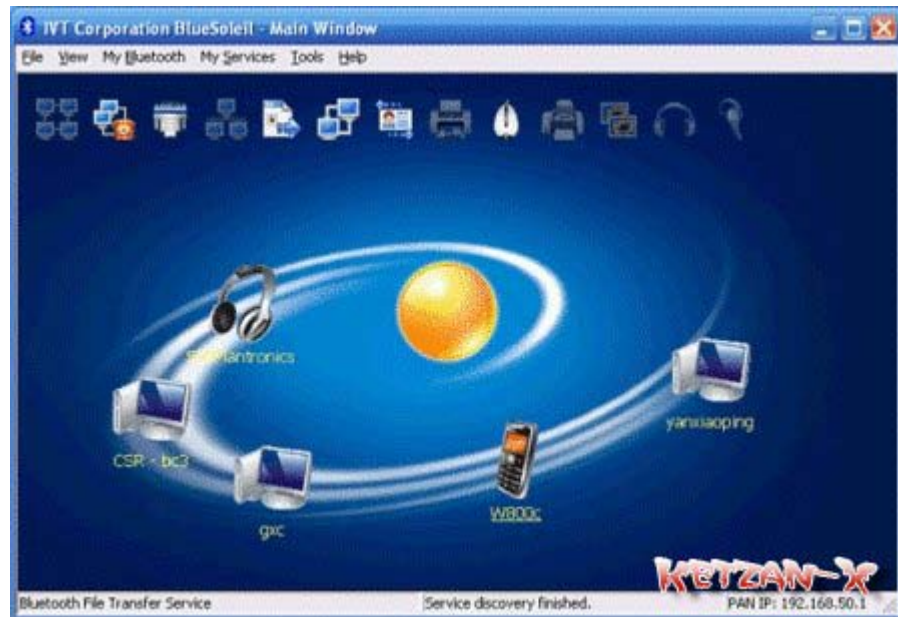
Figure 2: Wii Remote



1.2. BlueSoleil Module

The Wii Remote connects to the computer via Bluetooth technology, which allows for wireless exchange of data over short distances. By using Bluetooth dongles that connect to the computer via a USB 2.0 port, any computer can become part of a network of wireless devices. External Bluetooth connection software called BlueSoleil is necessary to connect the Wii Remote to the computer. BlueSoleil allows the Wii Remote to be detected by the computer, establishes a wireless connection between them (Figure 3), and allows for the detection of movement or button presses on the Wii Remote by the computer.

Figure 3: BlueSoleil



1.3. GlovePIE Module

GlovePIE, or Glove Programmable Input Emulator, is a scripting language that allows the input from the Wii Remote to be translated into key strokes, which control the game being played. The GlovePIE scripts dictate the controls, and are written through the system. Having students visually manipulate the Wii Remote controls through the GUI will in turn translate to lines of scripting code in GlovePIE. The GlovePIE script can then be used to play the game with the new controls.

1.4. External Video Game Software Module

Students will be able to play a variety of out-of-copyright and open source games using the Wii Remote. After a script is created through GlovePIE, the game starts and is played using controls specified in the script. Currently, these games include Asteroids (Figure 4) and Maryo (Figure 5). Each game has a set of controls to interact with and progress through the game. These set controls are options for the Wii Remote controls in the GUI, allowing students to indicate what Wii Remote input dictates which game control.

Figure 4: Asteroids



Figure 5: Maryo



1.5. Java Application Module

The Java application provides a variety of support services, including a graphical user interface, tutorials, script generation, input/output support, and launching utilities. These sub-modules are explained below.

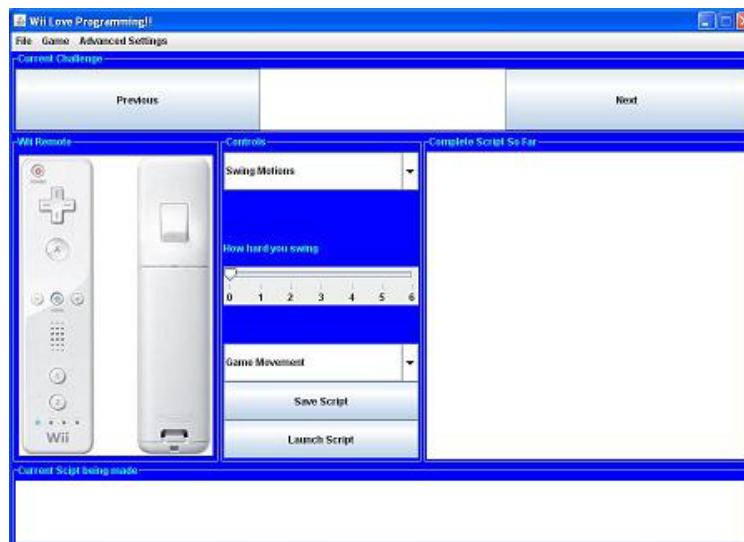
1.5.1. GUI

The GUI allows the user to interact with the internal workings of the software (Figure 6). The GUI includes a display of the Wii Remote as a visual aid, as well as options to select which game to play. To select a button on the Wii Remote, the user needs only to press the actual button on the Wii Remote itself. The GUI indicates which button has been selected for

modification by highlighting it on the onscreen Wii Remote image and also by saying the name of the button aloud. Various drop-down menus list the game controls that can be assigned to the selected button. Users have the option of setting the level of sensitivity of the built-in accelerometer that determines the amount of force needed to activate the control.

A display indicates the current line of code being generated from the user input. The student is able to see the line of code associated with a given control and action. Once students are happy with the control they have programmed, it is committed to an overall script view of all the controls already programmed. The game can be launched directly from the GUI, using the finalized GlovePIE script.

Figure 6: GUI Screen Shot



1.5.2. GUI Tutorial

The GUI tutorial will give a general explanation of how to use the interface.

1.5.3. Script Generator

The script generator takes the current button mapping and a condition specified by the user in the GUI and generates the corresponding line of code.

1.5.4. Input/Output (IO)

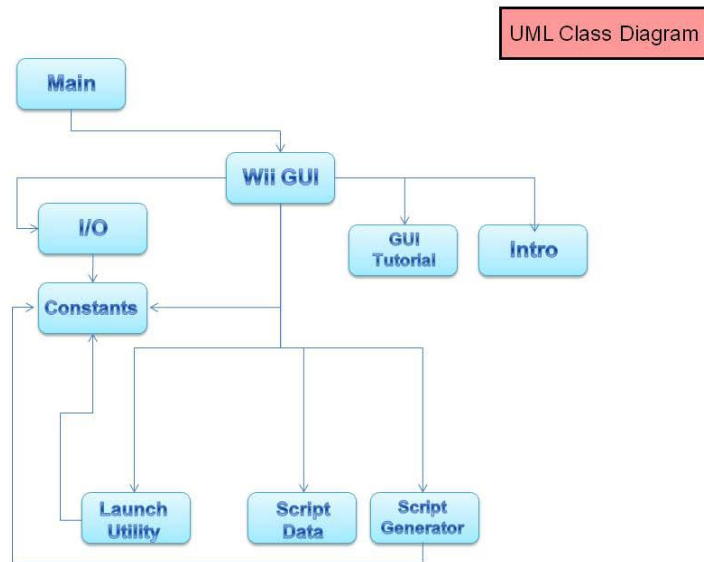
IO takes in the current game and line of script code created by the script generator, and creates a file containing the lines of code. These files are stored in a directory corresponding to the game. When the game is launched from the GUI, IO gathers the individual files and combines them into a single file. This file is then sent to GlovePIE and the new controls can now

be used to play the game.

1.5.5. LaunchUtil

LaunchUtil launches the GlovePIE and external game software. The class also handles killing GlovePIE and external game processes.

Figure 7: UML Diagram



2. Class Outline

iii. Use Cases

A. Administrator performs system set-up activities

Pre-condition: BlueSoleil is installed and configured correctly and "Wii Love Programming" software CD is installed and launched.

Post-condition: The Wii Remote is connected to the computer and the software is ready for student use.

1. Administrator follows user manual and connects Wii Remote to computer using BlueSoleil
2. Administrator launches software

B. Student engages in "Wii Love Programming"

1. Student reads introduction and selects to start game
2. Student plays the game with the default control scheme
3. Student views GUI Tutorial

4. Student reads instructions for the challenge
5. Student uses GUI to select and remap control of a Wii Remote input
 - a. Student selects Wii Remote button input from GUI
 - b. Student selects game action
6. Program displays corresponding generated script code
7. Student saves the current generated script code
8. Student repeats steps 5 - 7 until satisfied with controls
9. Student selects to start game
10. Program launches GlovePIE (hidden) and game
11. Student tests controls by playing game
12. Student returns to GUI and selects to resume programming
13. Student selects to progress to the next challenge (start step 4 with new challenge)

Alternate Flow:

4. a. Student selects Wii Remote accelerometer input from GUI
4. b. Student selects game action
4. c. Student sets condition

Alternate Flow:

13. a. Student is not satisfied with controls and returns to GUI

- C. Student engages the 'Advanced Configuration' mode
 1. Student selects "Advanced Options" menu, then 'Advanced Configuration'
 2. Warning message appears
 - a. No error handling in this mode so changes may break the software
 3. Student reads instructions in GUI
 4. Student manually edits script code
 5. Continue at Use Case B. 9
- D. Student engages the 'Enable Cheater Mode' option
 1. Student selects 'Advanced Options' menu, then 'Enable Cheater Mode'
 2. Checkbox appears already checked indicating cheater mode is on
 3. Corresponding game actions with cheats are generated
 4. Continue at Use Case B. 5

Alternate Flow:

3. a. Student un-checks checkbox to turn cheater mode off
3. b. Corresponding game actions no longer generate cheats

5. Student selects 'Advanced Options' menu, then 'Cheater Mode'
6. Checkbox disappears and cheats are no longer generated

- E. Student selects to save current script file
 1. Student selects option to save
 2. Program prompts for save path

3. Student enters path and saves file

Alternate Flow:

3. a. Path is invalid
3. b. Repeat step 2

- F. Student selects to load previous script file

1. Student selects option to load
2. Program prompts for load path
3. Student enters path
4. Program saves loaded file into the batch file

Alternate Flow:

3. a. Path is invalid
3. b. Repeat step 2

V. Implementation Details

i. Language

The software was designed using the Java language. The team chose to use Java due to its simple and powerful GUI design and implementation abilities

ii. Writing vs. Using Code

The functionality required for the Wii Love Programming software was very specific, so all implemented Java classes were written by the team. Other components of the system design used a variety of external software including BlueSoleil, GlovePIE, and video games.

BlueSoleil was highly recommended by a professor at Carnegie Mellon University as the easiest and most efficient software to manage Bluetooth connections [References, 1]. BlueSoleil was gracious enough to provide a free license for their product.

GlovePIE was chosen because it was open source, easy to use, and did not require any installation, making it very easy to incorporate onto the final CD.

The video games used are entirely open source, and were chosen because they would be fun and intuitive to use with the Wii Remote. The software also allows for open ended use in playing online flash games.

iii. Issues

One of the main issues that we had to deal with was the use of batch files. We were planning on using an external batch file to launch all of the necessary components used when running the software. That included GlovePIE scripts and the different video games. Unfortunately this was not possible due to the problems we had trying to map path names from a windows system into Java. We decided it was more trouble than it was worth and instead chose functionality with batch files, and then duplicated those features with the LaunchUtil class.

Another issue we faced that eventually turned out in our favor involved BlueSoleil. After agreeing to use BlueSoleil we realized a license would need to be purchased to use the software for more than 15 days. After contacting BlueSoleil and explaining our project in detail, they were gracious enough to give us a free license.

Another obstacle faced was a redefinition of the project 1 ½ weeks into the term. The team determined that the project goals could be better met with an entirely different project. It was up to us as a team to design a project that we felt was educational and involved Wii Remote technology, which also promoted Computer Science. Wii Love Programming was our idea, and after client and advisor approval, we started back at square one.

VI. Results

The software delivered met all of the functional and non-functional requirements initially requested by the client. Some of the additional features included in the final product include having the computer say the name of the Wii Remote button that is being pushed when programmed in the GUI. We also incorporated an extensive Users Guide, Take Home Instructions if students were interested in using the technology at home, and a PowerPoint that could be displayed by an instructor to students while using the software. We also added more features in the software as explained in detail below.

i. Project Progression

The final implementation has a few design and scope goals that were initially optional. As a team we decided to expand our scope to meet our client's needs. These optional capabilities include the ability to expand beyond the initial two video games. In addition to playing Asteroids and Maryo, users can program the Wii Remote to interact with a web browser, and interact with any online flash game.

As a result of this additional functionality, we introduced an Advanced Configuration mode, which allows the students to create their own scripts within the GUI. They are given a general commented outline of how to code the script, and are then free to implement the controls. The major drawback of this feature is the current lack of error handling. Should the student program something incorrectly, the program will not catch the error, and when the final script is launched in GlovePIE, it may or may not cause the software to crash.

Another expansion to the scope was 'Cheater Mode'. This mode demonstrates the capability of the Wii Remote when working with GlovePIE. GlovePIE maps different key strokes to any given input from the Wii Remote. Multiple key strokes can be mapped to one button press or action. Cheater mode currently enables the Asteroids ship to shoot four times instead of once, and also allows Maryo to jump twice instead of once.

VII. Conclusions

The Wii Love Programming software has been completed successfully, and should make a positive impact on high school students and bolster interest in computer

science. The incorporation of Wii Remote technology and general computer science concepts should help increase students interest and awareness of the computer science field, as well as the endless possibilities of technology. This increased interest and awareness will promote enrollment in the computer science department at universities.

i. Future Directions

The Wii Love Programming software can be expanded in numerous ways to provide an even more interactive and exciting design. Some of these options include error handling, additional games and challenges, multiple controllers, a more applicable cheater mode, and the incorporation of graphs and physics.

1. Error Handling

When in Advanced Configuration mode, the students have the ability to write and edit their own scripts to interact with internet browsers and online flash games. The only downside to this feature is that there is no error handling. If a student were to enter something incorrectly, the software may ultimately crash when the script was run. Some form of error handling would be extremely helpful and educational for the students to have while manually editing scripts. This may be very difficult to implement because the script is meant to be very open ended, and the students could be mapping a variety of key strokes.

2. Other Games and Challenges

Currently only two games are available, Asteroids and Maryo, but more games would provide a more diverse learning environment. Some games may be more intuitive when used with a Wii Remote as opposed to a keyboard. One option is Frogger. Swing motions from the Wii Remote would be a great way to maneuver the frog. More games could provide challenges for the students to complete. There are endless possibilities to the types of challenges that could be incorporated.

3. Multiple Controllers

Another possibility would be incorporating multiple Wii Remotes or Nunchuck controllers. This could be especially useful for games that may have a lot of game actions, or ones that would be intuitive to use with a component in each hand. This functionality would depend on the specific game. Multiple controllers could also allow multiple users to play the same game.

4. Cheater Mode

The basic cheater mode currently supported could expand to support other cheat features. Currently, the cheater mode is very limited due to the simplicity of the games provided. It could make combination moves that involve multiple key strokes very easy to do with just one Wii Remote control.

5. Graphs and Physics

The incorporation of graphs and physics into the software would be very beneficial to students. The Wii Remote uses an accelerometer to track its swing actions. Some

future possibilities include dynamically graphing motions based on preset conditions, displaying the relationship between acceleration and velocity, and related activities.

VIII. References

[1] Johnny Lee, Carnegie Mellon University Professor: [://johnnylee.net/projects/wii/](http://johnnylee.net/projects/wii/)

[2] BlueSoleil Website: [://www.bluesoleil.com](http://www.bluesoleil.com)