Shale

Test Plan

May 3, 2009

Paul Gerhardt

Amanda Orin

Tomas Ramirez

Jessa Rothenberg

Kaiti Trimble

CSCI 4308-4318. Software Engineering Project 1 & 2

Department of Computer Science

University of Colorado at Boulder

2008-2009 sponsored by

CU Craft Technology Group

PROJECT PROPOSAL

The purpose of the project (Shale) is to construct a system in which physical and virtual objects interact. A previous implementation of this idea, Laser Ball, combined virtual bouncing balls with physical pieces of cardboard.

The image to the left shows what the Laser Ball system looks like. A virtual screen is created by projecting light onto a wall. Balls are created by briefly shining a laser pointer onto the screen. The balls fall down, and when they collide with the black pieces cardboard shapes, they bounce like a real ball bouncing off of a hard surface.

A major requirement of this new project is that the physical components of the system be dynamic in some way, for instance, a box might flash a light when it is hit, or a seesaw moves when a ball falls on its raised end. In addition to having a system with dynamic physical components, the project sponsor hopes to have a system that is fun-to-use, as it may have entertainment and educational applications in the future.

TABLE OF CONTENTS

TABLE OF FIGURES

1 INTRODUCTION

CU Craft Technology Group is a part of the Center for Lifelong Learning and Design at the University of Colorado specializing in the integration of computation and craft materials to produce mathematical or educational toys and activities for children. The group consists of a small number of faculty, graduate, and undergraduate students working on the Boulder Campus of the University.

Previously, CU Craft Technology collaborated with a team of undergraduate software engineers to develop a project entitled Laser Ball. This program created a combination of virtual and physical elements through the use of digital image recognition. Project Shale is an expansion upon this project, and will include wireless communication with the physical objects, in order to enable more advanced interactions such as movements, lights, and sound.

A conceptual diagram of the overall system is presented in **Figure 1**. The figure shows the software to be implemented, Shale (Levers Shadows & Wheels). Shale receives input from the webcam and outputs data to the projector, which projects virtual objects (e.g. falling spheres) onto a whiteboard or blank wall. Affixed to the wall are mechanical/physical objects, and the beam of a green laser pointer, which are seen by the webcam, and interpreted by digital image processing software within Shale. Shale then combines the webcam's input and correlates the locations of the virtual objects to create an interactive environment. Shale will also interact with the physical objects through use of a wireless transmitter. It will send signals to these physical objects when they interact with virtual objects, triggering movement (through motors), lights (through LEDs), or sounds (through speakers).

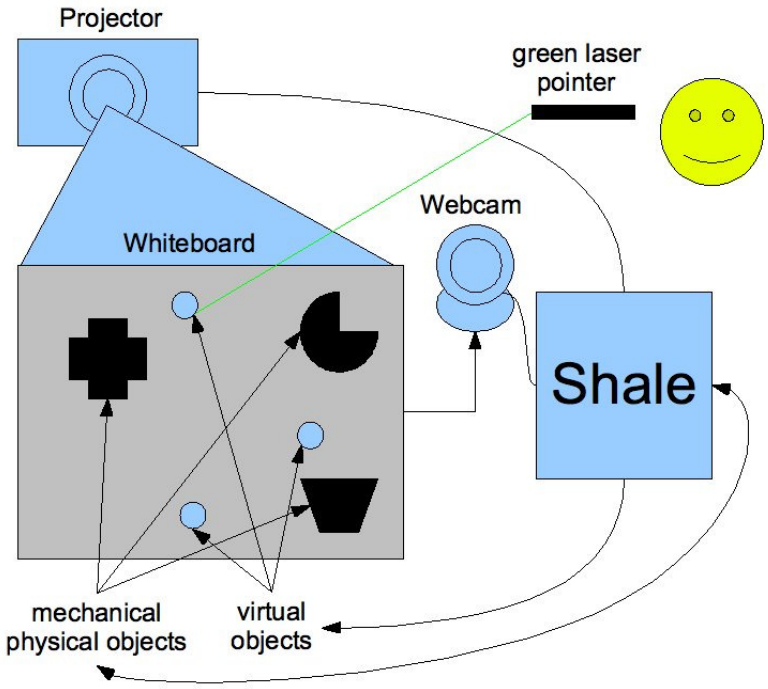

Figure 1: *Conceptual Overview of the Shale System*

This paper defines the current test plan for Shale. These tests will be updated as they evolve over the course of the project. A list of useful references is also provided at the end of this paper.

2 Test Environment

The Shale tests should be conducted in the following runtime environment, as specified in the *Shale Requirements*:

Software

- Windows XP SP3, Windows Vista SP1, or Mac OS X 10.4.11
- Processing PDE 135 Beta or later **[Processing]**
- Sun Java 6 JDK or later **[Sun Java 6 SDK]**

Hardware

- Hardware supporting the software runtime environment described above and with the following minimum configuration:
    - 1GHz processor
    - 1GB available disk space
    - 256 MB RAM
- Minimum 1024 x 768 pixel 24-bit color projection display
- One green laser pointer.
- Functional webcam with appropriate drivers, preferably Logitech® Quickcam Pro 9000™**[Logitech Quickcam Pro 9000]**
- Arduino Diecimila used to control mechanical objects **[Arduino Diecimila]**
- Wireless communication device adaptor for Arduino Atmel (XBEE)
- A series of mechanical objects constructed by the team, such as: an LED reactive block, a see-saw, a spring block, and a water-wheel to interact with virtual objects
    - LED reactive block will require a switch that can be switched on/off wirelessly
    - See-saw will require a servo that can be controlled wirelessly
    - Spring block will require a spring and a stepper motor
    - Water-wheel will require a stepper motor capable of rotating in both directions (clockwise and counter)

3 Tests

Tests are organized into groups that target specific areas of the program.

- Button Bar
- Collision Detection
- Mechanical Actuation
- Image Recognition
- Performance
- Documentation
- Release

Each test in the test plan has seven components:

Purpose	The reason for the test.
Procedure	The set of steps to follow to conduct the test.
Expected Result	The result necessary to pass the test.
Comments	This is for comments the tester might have.
Date	The date that the test was conducted on.
Tester	The name of the person who conducted the test.
Outcome	The outcome of the test (PASS or FAIL).

Windows XP	**Windows Vista**	**Mac OS X**
PASS	PASS	PASS

3.1 Button Bar

Shale has several buttons—Thunk, Restart, and Add Balls—that the user will use to control the interaction. These tests verify the correct behavior of the buttons.

3.1.1 Thunk Button

Purpose

This test verifies that the Thunk button causes the virtual balls to move appropriately.

Procedure

1) Shine the laser pointer on the Thunk button until the button changes color.
2) Release the laser pointer.
3) Observe the resulting behavior.

Expected Result

When the Thunk button is selected, regardless of how many times or which time it is selected, it should always attempt to jumble the virtual objects around the window.

The virtual objects should still collide correctly immediately after the Thunk button is activated.

Comments

Date

5/3/2009

Tester

Jessa Rothenberg

Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.1.2 Restart Button

Purpose

This test verifies the behavior of the Restart button.

Procedure

1) Shine the laser pointer on the Restart button until the button changes color.
2) Observe the resulting behavior.

Expected Result

When the Restart button is selected regardless of which time it is selected, all virtual objects should be cleared from the window and one new virtual ball should appear.

Comments

Date

5/3/2009

Tester

Jessa Rothenberg

Outcome	Windows XP	Windows Vista	Mac OS X
	PASS	PASS	PASS

3.1.3 Add Balls Button

Purpose	This test verifies the behavior of the Add Balls button.
Procedure	1) Shine the laser pointer on the Add Balls button until the button changes color. 2) Observe the resulting behavior. 3) Once again, shine the laser pointer on the Add Balls button until the button changes color.
Expected Result	When the Add Balls button is first selected via the laser pointer, a set of 5 virtual objects should appear at the top near said button and drop down in the window. When selected again, another set of 5 should appear, bringing the total in the window during this test to 11 virtual objects, including the original starting virtual object.
Comments	---
Date	5/3/2009
Tester	Jessa Rothenberg

Outcome	Windows XP	Windows Vista	Mac OS X
	PASS	PASS	PASS

3.2 Collision Detection

Shale's collision detection involves both collisions between a virtual object and a physical object, and collisions between two virtual objects. These tests will verify correct behavior in both cases.

3.2.1 Physical-Virtual Object Collision

Purpose	This test determines if virtual objects collide with physical objects correctly.		
Procedure	1) Place a physical object on the Stage. 2) Thunk until the ball has touched the physical object at least 10 times. 3) Check if ball bounces off of the physical object each time.		
Expected Result	When the ball touches the physical object, it should bounce off of the physical object.		
Comments	---		
Date	5/3/2009		
Tester	Tomas Ramirez		
Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.2.2 Virtual-Virtual Object Collision

Purpose	This test determines if virtual objects collide correctly.		
Procedure	1) Start Shale. 2) Add balls. 3) Check if balls bounce off of each other.		
Expected Result	When any two balls touch, they should bounce off of each other.		
Comments	---		
Date	5/3/2009		
Tester	Tomas Ramirez		
Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.3 Mechanical Actuation

These tests verify the basic communication and commands from Shale to an Arduino-controlled physical object with a motor. They test wireless communication by way of starting the motor and stopping the motor.

3.3.1 Start

Purpose	This test verifies that the physical devices receives and correctly interprets a signal to start its motor.
Procedure	1) Ensure the physical object is appropriately powered. 2) Ensure Xbee device is connected to the appropriate pins. 3) Ensure that the motor shield has power. 4) Connect the second Xbee device to the computer. 5) Send a test signal over the computer's COM port to start the device. 6) Observe the resulting behavior.
Expected Result	When the computer sends a signal to the physical device, the physical device is expected to start its motor; in the case of a DC motor, the motor spins. This test verifies that both a communication link has been established and that the physical object is responding to commands to start its motor.
Comments	---
Date	5/3/2009
Tester	Paul Gerhardt

Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.3.2 Stop

Purpose	This test verifies that the physical devices receives and correctly interprets a signal to stop its motor.

Procedure	1) Ensure the physical object is appropriately powered. 2) Ensure Xbee device is connected to the appropriate pins. 3) Ensure that the motor shield has power. 4) Connect the second Xbee to the computer. 5) Send a test signal over the computer's COM port to start the device. 6) Send a test signal over the computer's COM port to stop the device. 7) Observe the resulting behavior.		
Expected Result	When the computer sends a signal to the physical device, the physical device is expected to stop its motor; in the case of a DC motor, the motor stops spinning. This test verifies that both a communication link has been established and that the physical object is responding to commands to stop its motor.		
Comments	---		
Date	5/3/2009		
Tester	Paul Gerhardt		
Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.4 Image Recognition

Shale's image recognition requires that objects be recognized, all desired objects be recognized correctly, and that no non-objects be detected erroneously at all.

3.4.1 Object Recognition

Purpose

This test verifies that Shale recognizes physical objects that have been placed on the stage.

Procedure

1) Prior to running, place a physical object on the stage.
2) Run Shale.
3) Ensure that all virtual objects interact appropriately with the object.
4) Pause Shale and replace the physical object with a different physical object.
5) Resume Shale.
6) Ensure that at least one virtual ball interacts appropriately with the object.

Expected Result

When a virtual ball interacts with a physical object, the direction and velocity of the virtual object's motion should change appropriately (i.e. it should bounce off the object). This should happen for both objects used in the test.

Comments

Date

5/3/2009

Tester

Kaiti Trimble

Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.4.2 False Negatives/Positives

Purpose

This test verifies that Shale does not identify objects which do not exist (noise), and does not miss any objects that should be identified.

Procedure

1) Set Shale to print out a trace of the coordinates of all objects identified in the image recognition step by way of setting a debugging variable to enable a printout of these coordinates.
2) Run Shale.
3) Check the number and location of physical objects and laser points identified.

Expected Result	The position and quantity of objects and laser points identified will match their actual physical representations.		
Comments	---		
Date	5/3/2009		
Tester	Jessa Rothenberg		
Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.5 Performance

The Performance tests verify that performance-related requirements have been met.

3.5.1 Startup Time

Purpose	This test verifies that Shale starts within the required time constraints.		
Procedure	1) Run Shale on all platforms. 2) Observe the elapsed time from invocation until the window is usable in each case.		
Expected Result	The elapsed time from invocation to usable interface should always be less than 5 seconds.		
Comments	---		
Date	5/3/2009		
Tester	Amanda Orin		
Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.5.2 Capture Rate

Purpose	This test verifies that Shale can handle the required minimum of ~10 frames per second for image capturing.		
Procedure	1) Run Shale normally with at least one physical object in place. 2) Observe the resulting behavior.		
Expected Result	Shale should behave normally and should draw the virtual representation of the physical object in the appropriate place. Determining ~10 frames per second is by observing the resulting behavior does not produce a lag that the naked eye can spot in the capture rate.		
Comments	---		
Date	5/3/2009		
Tester	Amanda Orin		
Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.5.3 Physical Object Support

Purpose	This test verifies that Shale can handle the required minimum of 1+ physical objects during interaction time.
Procedure	1) Run Shale with at least 1-2 physical objects placed within webcam recognition range. 2) Observe the resulting behavior.
Expected Result	Shale should behave normally and keep up with the collision detection between virtual and physical objects. The tester should not find unusual behavior such as unresponsive reaction to commands being sent by Shale, throwing exceptions or errors, or a delay in a collision response by the physical objects.
Comments	Tested on Mac OS X via serial interface.
Date	5/3/2009
Tester	Amanda Orin

Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.6 Documentation

The *Shale Requirements* document specified the documentation, both papers and presentations, that was to be provided and how that documentation was to be created. The Documentation tests verify that appropriate documentation is provided.

3.6.1 Document Content

Purpose This test verifies that the appropriate set of documents was provided.

Procedure Review the set of documents provided in the release.

Expected Result The following documents should be provided:

- Development Documents:
  - Requirements
  - Design
  - Test Plan
  - Release Notes
- User Documents:
  - Project Website
- Presentations:
  - Overview
  - State of the Project
  - Final Demo

Comments ---

Date 5/3/2009

Tester Amanda Orin

Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.6.2 Document Tools

Purpose This test verifies that the documents were developed using the appropriate tools.

Procedure Open each paper in Microsoft Word 2004 (or later) and each presentation in Microsoft PowerPoint 2004 (or later).

Expected Result	Each document should be compatible with the corresponding Word or PowerPoint version as appropriate.		
Comments	---		
Date	5/3/2009		
Tester	Amanda Orin		
Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.6.3 Document PDFs

Purpose	This test verifies that PDF versions of all documents were provided.		
Procedure	Locate the PDF version of each paper and presentation and open it.		
Expected Result	Each document should be a PDF version of the current paper or presentation.		
Comments	---		
Date	5/3/2009		
Tester	Amanda Orin		
Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.7 Release

Release tests verify that the release has the appropriate format as well as content.

3.7.1 Release Format

Purpose	This test verifies that the release has the appropriate format.		
Procedure	Determine the format of the release.		
Expected Result	The release should be a compressed archive file, created using WinZip or **gzip**.		
Comments	---		
Date	5/3/2009		
Tester	Amanda Orin		
Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

3.7.2 Release Content

Purpose	This test verifies that the release has the appropriate content.		
Procedure	Determine the content of the release.		
Expected Result	The release should contain the following: ▪ entire source tree of software ▪ source to all documentation ▪ documentation in hardcopy form		
Comments	---		
Date	5/3/2009		
Tester	Amanda Orin		
Outcome	**Windows XP**	**Windows Vista**	**Mac OS X**
	PASS	PASS	PASS

4 SUMMARY

This document is intended to give a detailed test plan for Shale. It provided test plans for each important section of the project: user interface buttons, collision detection, mechanical actuation, image recognition, performance, documentation, and release. This should provide sufficient testing to determine if Shale is working correctly or not.

5 REFERENCES

There are a number of documents related to this paper that are useful for further reading.

[Processing]

"Learning" *Learning\Processing 1.0 (BETA)*. Sept 2001. Processing (September 12, 2008) <http://processing.org/learning/index.html>

[Arduino Diecimila]

"Arduino Diecimila" *Arduino - ArduinoBoardDiecimila*. October 2008. Aduino (October 30, 2008) <http://arduino.cc/en/Main/ArduinoBoardDiecimila>

[Sun Java 6 SDK]

"Java SE Downloads" *Java SE Downloads*. Continuous. Sun Microsystems. (September 12, 2008) <http://java.sun.com/javase/downloads/index.jsp>

[Logitech Quickcam Pro 9000]

"Quickcam® Pro 9000" *QuickCam® Pro 9000*. Continuous. Logitech. (September 12, 2008) <http://www.logitech.com/index.cfm/webcam_communications/webcams/devices/3056&cl=us,en>