Shale

Requirements

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

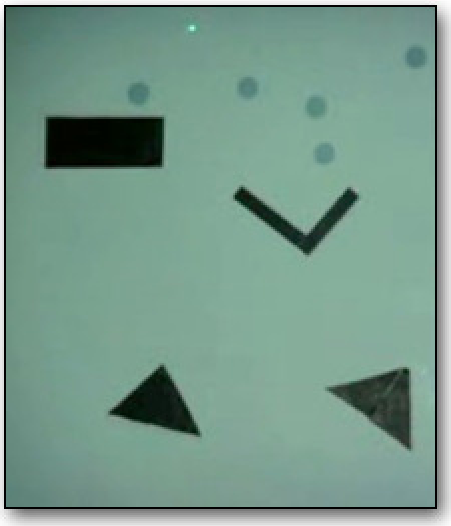
2008-2009

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).

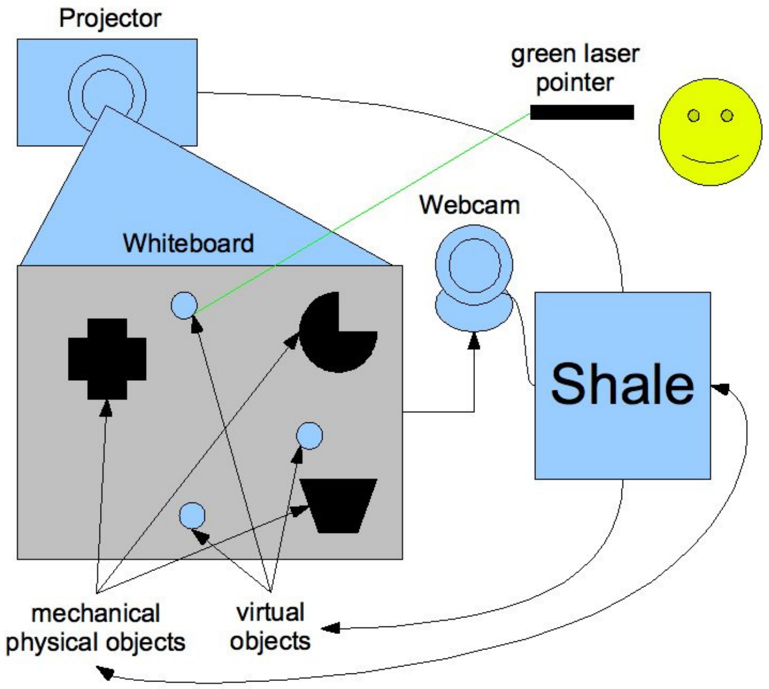


Figure 1: *Conceptual Overview of the Shale System*

This paper defines the current set of requirements for Shale. These requirements 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 REQUIREMENTS

To make requirements more manageable, they have been divided into several logical sections. These sections include the requirements of the Supporting Environment, Functional Requirements, Performance Requirements, Documentation Requirements and Release Requirements.

2.1 Supporting Environment

The Supporting Environment includes the development environment that must be used to create Shale, as well as the hardware and software runtime environments within which Shale must work.

2.1.1 Development Environment

Shale must be developed in the following environment:

- Windows, or Mac OSX operating system capable of running Processing PDE 135 Beta or Later
- Sun Java 6 JDK or later
- Webcam, preferably Logitech® Quickcam Pro 9000™
- Projector, preferably 1024 x 768 pixel 24-bit color resolution
- Wireless Arduino Diecimila boards with compatible motor (accessed via XBEE)
- Arduino Processing library
- Processing PDE 135 Beta or later
- SpringGUI Library

2.1.2 Software Runtime Environment

Shale must execute correctly on a platform having the following software runtime environment:

- Windows XP SP3, Windows Vista SP1, or Mac OS X 10.4.11 (PPC and Intel) performance may bet degraded
- Processing PDE 135 Beta or later
- Sun Java 6 JDK or later
- Functional webcam with appropriate drivers, preferably Logitech® Quickcam Pro 9000™

2.1.3 Hardware Runtime Environment

Shale must execute correctly on a platform having the following hardware runtime environment:

- 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-button pointing device with laser (green)
- Arduino Diecimila used to control mechanical objects
- Wireless communication device adaptor for Arduino Atmel (XBEE)
- A series of mechanical objects to be constructed by the team, including: 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)

2.2 Functional Requirements

Functional Requirements specify the functionality that Shale is required to provide to its users. This includes requirements related to the input Shale must be able to accept input from a webcam, allow for debugging, output serial communication, render objects via projector, control mechanical objects, and perform all of these tasks in a timely manner.

2.2.1 Webcam Data

Shale's primary means of input will be through a real time webcam feed, though duplex interaction with mechanical objects may be added at a later stage. For the webcam feed Shale must be able to:

- Process the webcam stream in real time for pattern recognition of fixed objects.
- Provide interaction between virtual elements and physical elements in the form of collision detection and physics.
- Provide mechanical response with physical mechanical objects (see-saw) when collision occurs with virtual elements.
- Recognize input in the form of a green laser pointer, and place virtual balls at that location with a single-click per virtual ball.

2.2.2 Debugging Interface

A debugging interface will be implemented for developers for means of flow control through keyboard and mouse, and as a demonstration of the mechanics of the object recognition technology.

- When debugging is turned on, outlines of recognized objects will be displayed on the screen.

- Console or inline terminal output of events shall be output. Events may include collisions between real and virtual objects (or perceived collisions), signals to motors to actuate, or for virtual balls to be generated.
- A marginal performance drop (to be determined) is acceptable when debugging is turned on, though the program should still be able to operate in real time.

2.2.3 Serial Input Configuration

Since events as provided by a keyboard are simply ASCII encoded characters, it must be possible to interpret certain strokes as means for controlling program operation. Fortunately many libraries already exist for interpreting standard input. Through means of operating a keyboard, the user should be able to:

- Pause program operation.
- Reset generated virtual balls.
- Enable or Disable debugging mode through implementation of the "toggle pattern" by pressing the same key for Enable as well as Disable.

2.2.4 Event Display

Processing data computed by Shale must be displayed to the user:

- Processing output data should be displayed in a visual form.
- Due to the nature of the interaction between virtual and physical mediums, Shale will need to project onto a chromatically uniform white surface, such as a whiteboard.
- The medium being projected on should be capable of supporting interactive objects, not only supporting the load of tapped or magnetized black objects, but also for the more complex mechanical objects like the seesaw.

2.2.5 Mechanical Objects

One of the primary goals for Shale is the ability to interact across the virtual physical medium. This will be accomplished through the in physical response of mechanical objects when virtual objects "hit" them. The mechanical objects should be able to:

- Accept commands from Shale, including actuation instructions.
- Be movable throughout the projection environment.
- Preferably have a wireless means of communicating with Shale.

2.2.6 Mechanical Actuation Control

As stated the mechanical objects must have actuators capable of receiving instructions for which direction to manipulate their motors. Motors should be able to actuate continuously (greater than 360 degrees for servos):

- Forward – start actuation in a preconfigured direction

- Reverse – start actuation in the reverse direction
- Stop – stop actuation.
- These motors must be able to respond to virtual interaction as if they are simulating collision detection, compression for the spring block, and rotation interaction with gravity for the water-wheel.

2.3 Performance Requirements

This section specifies requirements related to speed of the software, given the minimum hardware configuration:

- Startup time less than 5 seconds.
- Process object recognition and interaction with a display of ~10 frames per second.
- Number of simultaneous physical objects being recognized ~10 elements per second.
- Number of simultaneous physical objects responding to interaction ~3 objects per second on the board at any given time.
- Maximum physical board size of 3x4 ft.

2.4 Documentation Requirements

These requirements specify the documentation, both papers and presentations, that is to be provided and how it will be created.

- Documents to be provided:
    - Development Documents:
        - Requirements
        - Design
        - Test Plan
        - Release Notes
    - User Documents:
        - User Reference
        - User Tutorial
    - Presentations:
        - Overview
        - State of the Project Address
        - Final Demo
- Specific content of each document will be determined later.
- User files and README written in plain-text format.

- All papers written using Microsoft Word 2004 or later in DOC format.
- All presentations created with Microsoft PowerPoint 2004 or later in PPT format
- Provide PDF versions of all documentation.

2.5 Release Requirements

Release Requirements deal with issues related to the release and delivery of the final product to the customer.

- Delivered as compressed archive file, created using zip.
- Archive file should include entire source tree of software.
- Archive file should include source to all documentation.
- Source and documentation available for download from project website – to be designed.

3 SUMMARY

This document describes the set of requirements for the Shale software package. This included requirements related to the environment used to develop Shale and the software and hardware environment in which Shale will run, functionality that Shale must provide, performance standards that the program will meet, supporting documents that will be written and the manner in which the entire package will be delivered. The document will continue to evolve as the project progresses.

4 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>