



Final Demonstration

Shale

*(**SH**adows **LE**vers & **W**heels)*

Paul Gerhardt
Amanda Orin
Tomas Ramirez
Jessa Rothenberg
Kaiti Trimble



The Shale Team



. . . Paul Gerhardt



. . . Amanda Orin



. . . Tomas Ramirez



. . . Jessa Rothenberg

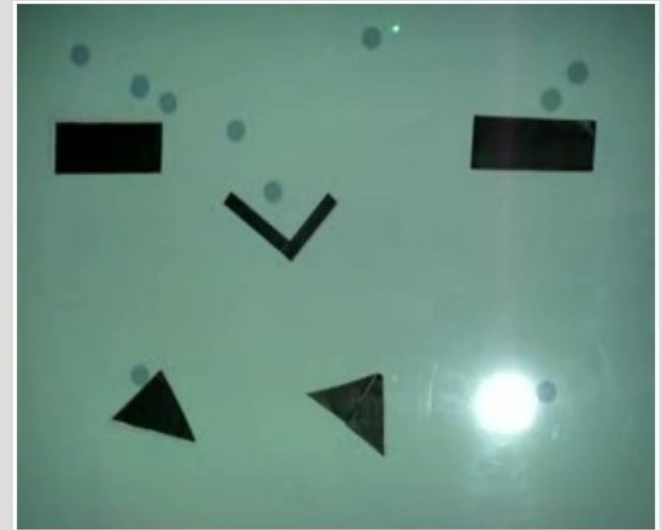


. . . Kaiti Trimble



Presentation Focus

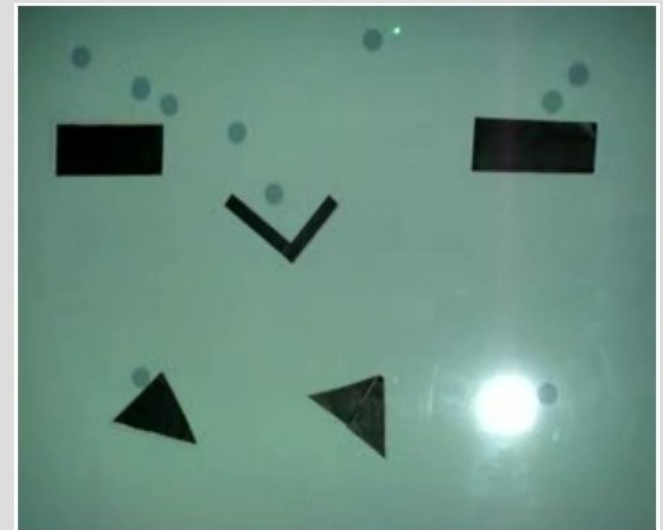
- Overview of the Project
- Software Demonstration
- Overview of the Design & Implementation





Presentation Focus

- Overview of the Project
 - The Class
 - The Problem
 - The Solution
- Software Demonstration
- Overview of the Design & Implementation
- Known Issues & Future Work





The Problem

- Interactions between virtual and physical objects
 - Virtual: balls, balloons, bubbles
 - Physical: bumper, see saw, waterwheel
- User input
 - move physical objects
 - create virtual objects
- Expanding the idea



The Class

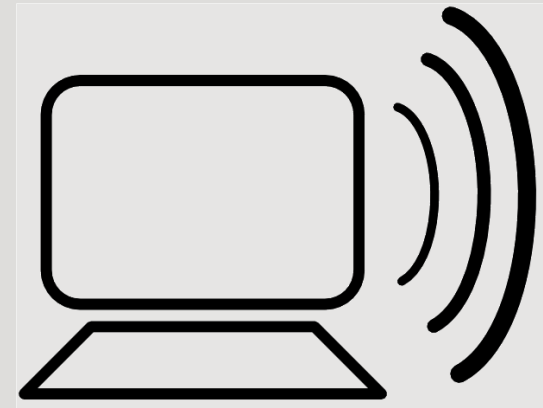
- Computer Science Capstone
- 49 Students... 10 Teams
- Industry Projects
 - Game Creation Game for the Nintendo Wii
 - AgentSheets, Inc. (Boulder, CO)
 - Auto-Categorization of Content Publishers
 - Lijit Networks, Inc. (Boulder, CO)
 - MyJinji – Social Networking Portal
 - MyJinji (San Francisco, CA)





The Problem

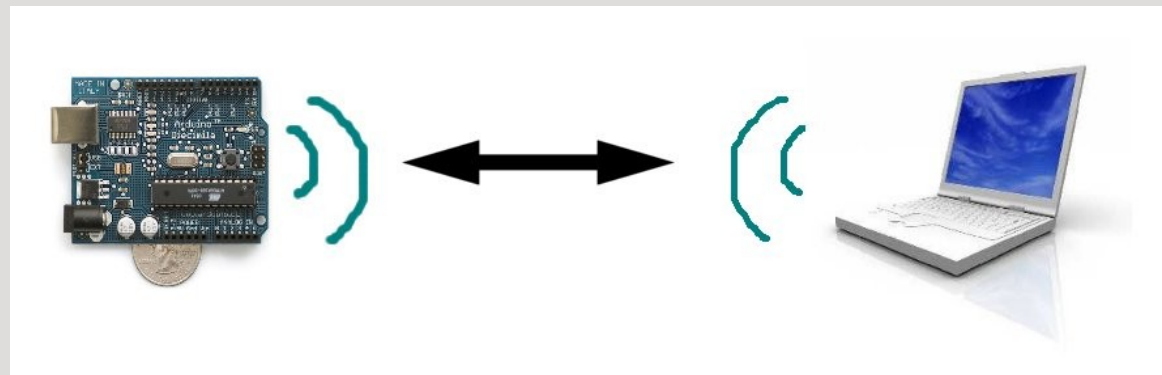
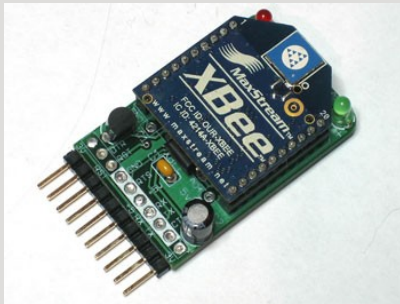
- Laserball...
 - Virtual objects
 - “Dumb” physical objects
- Needed:
 - Incorporate reactive physical objects
 - Wireless communication





Solution: Shale

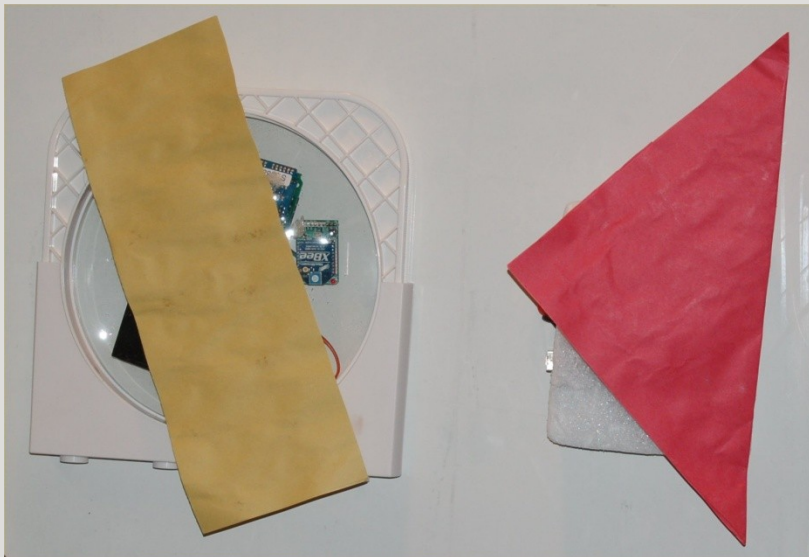
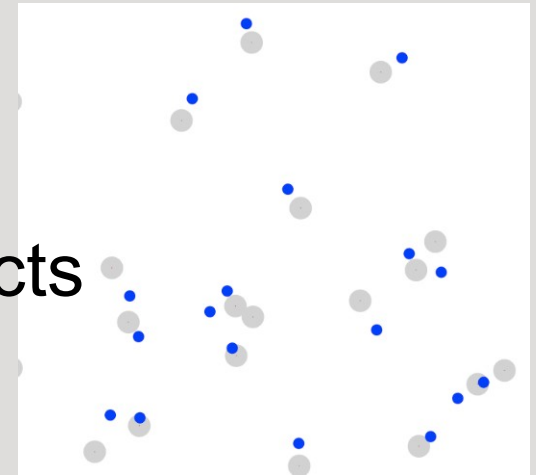
- Extend the LaserBall project with reactive physical components
 - See-saw, Bumper, Sound Emitter
- Wireless communication with components
 - Xbee (Zigbee 802.15.4 mesh networking)





Solution: Shale

- Recognize Physical Objects
 - Use of color to distinguish objects

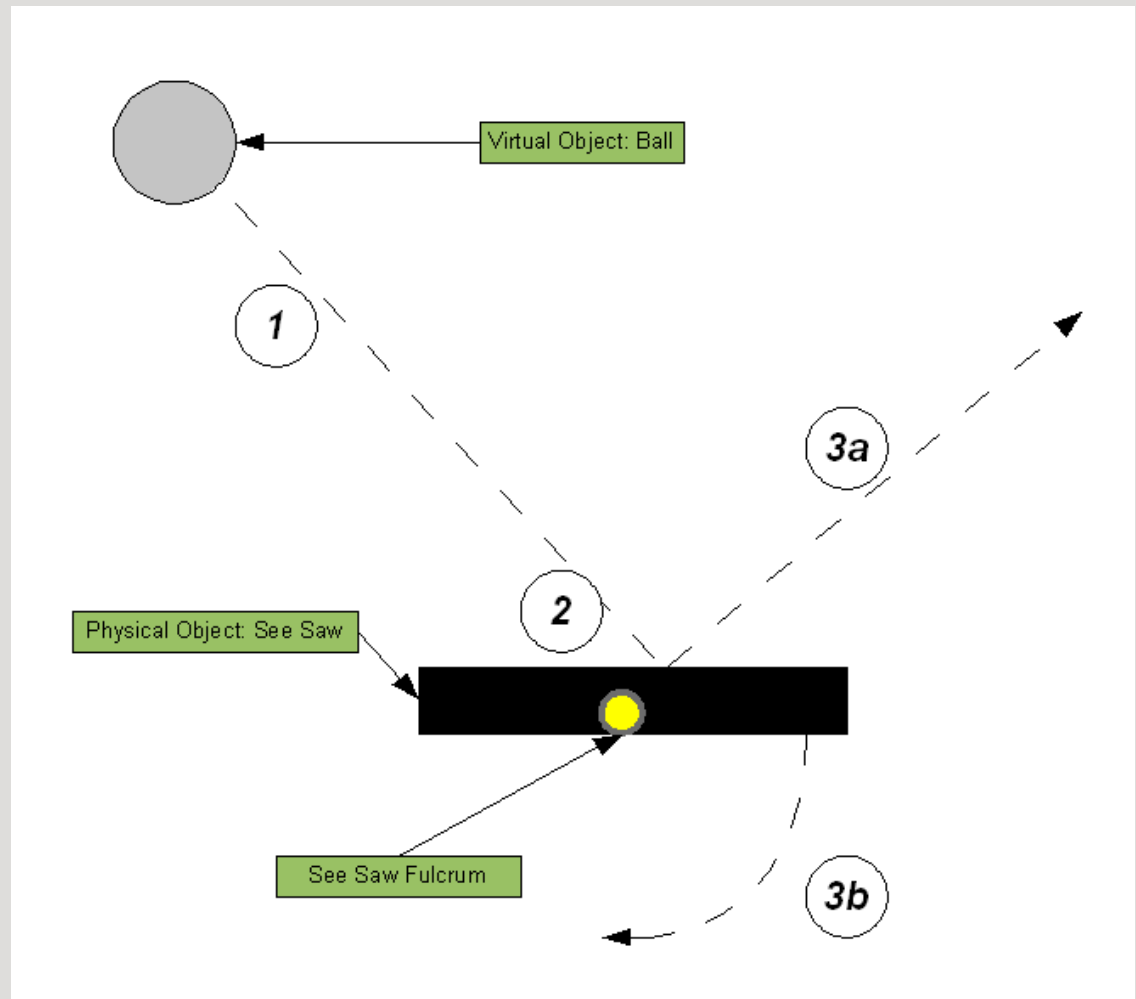




Shale: An Example

Steps:

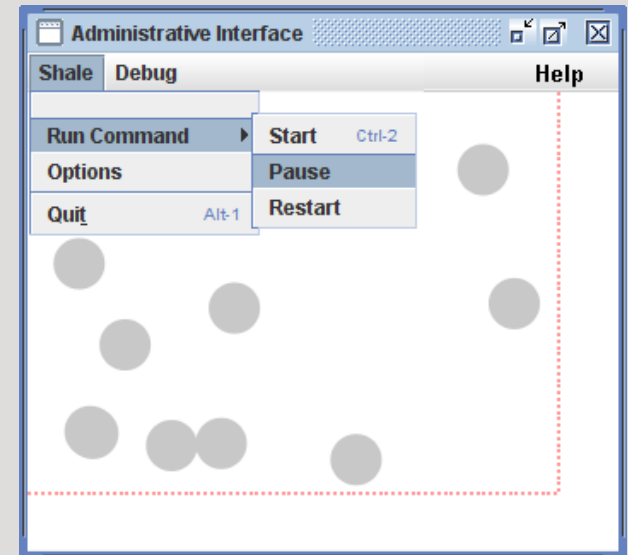
1. Virtual Object (the ball) moves towards the Physical Object (the see saw)
2. Virtual Object and Physical Object collide
3. Virtual Object (ball) ricochets off in a different direction AND Physical Object (see saw) turns in reaction to the collision





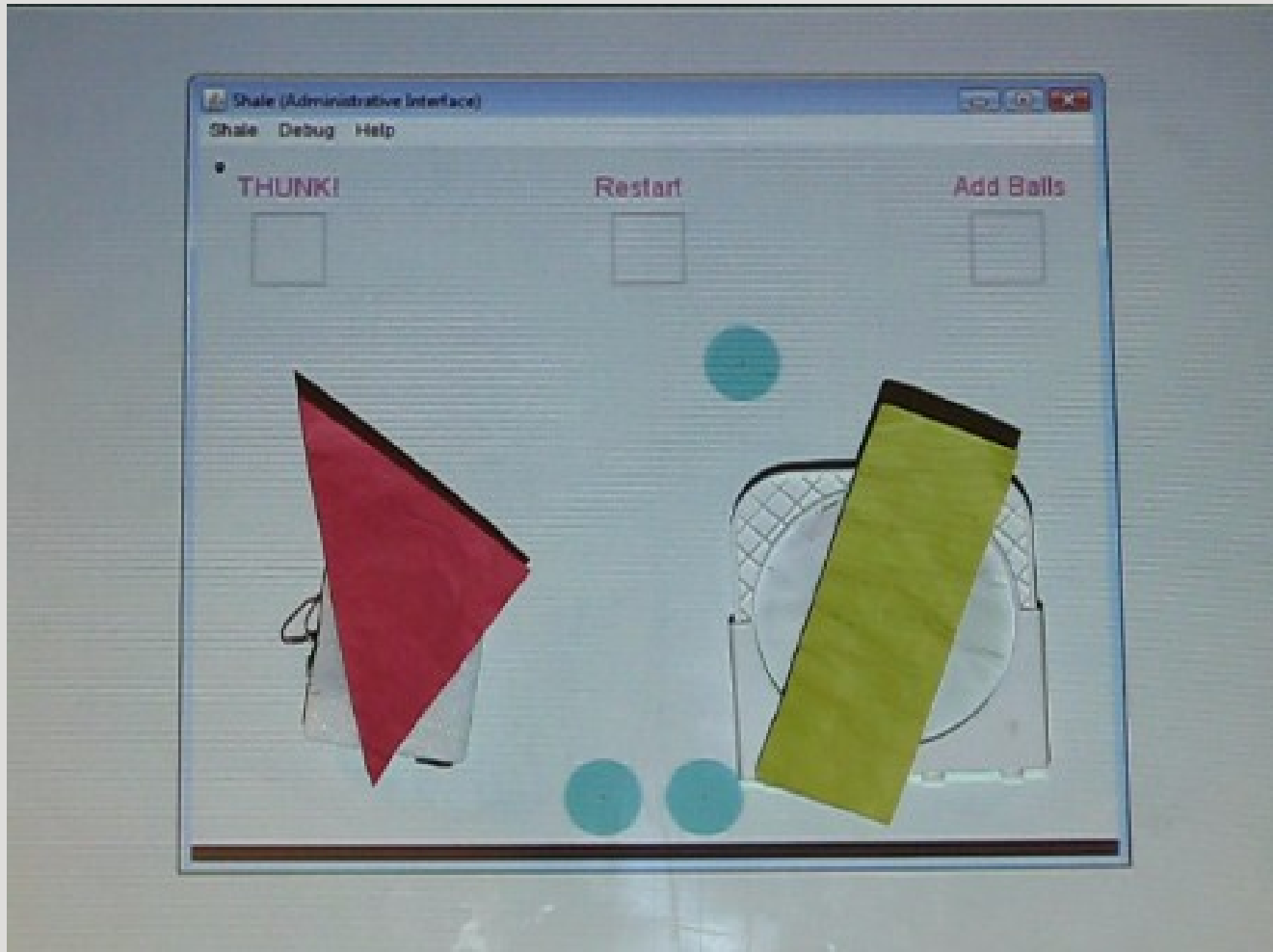
Presentation Focus

- Overview of the Project
- **Software Demonstration**
- Overview of the Design & Implementation
- Known Issues & Future Work





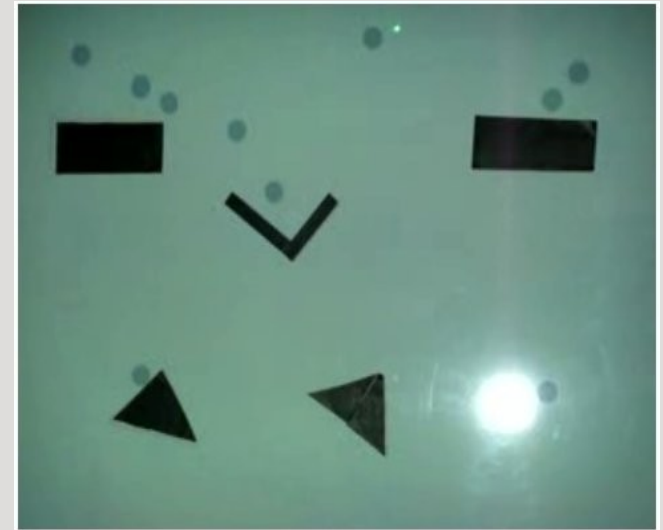
Software Demonstration





Presentation Focus

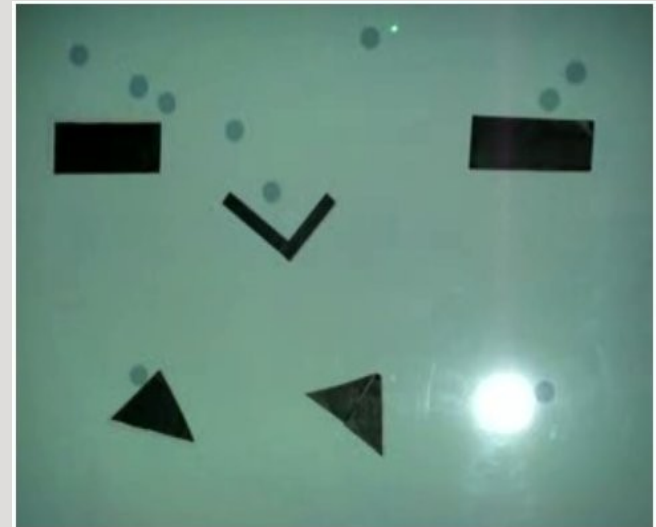
- Overview of the Project
- Software Demonstration
- Overview of the Design & Implementation
 - Requirements
 - User Interface Design
 - Architecture
- Known Issues & Future Work





Presentation Focus

- Overview of the Project
- Software Demonstration
- Overview of the Design & Implementation
 - **Requirements**
 - Development Requirements
 - Environment Requirements
 - Hardware Requirements
 - Functional Requirements
 - User Interface Design
 - Architecture
- Known Issues & Future Work





Development Requirements

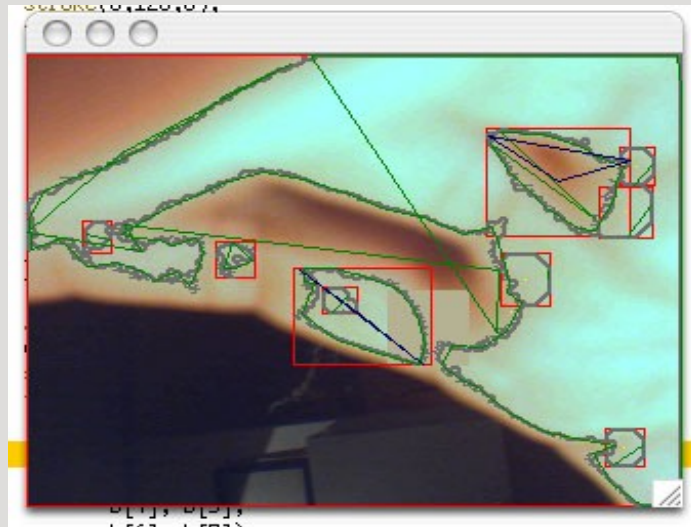
- Processing PDE 1.0
 - LaserBall
 - Easy graphical content
 - Cross-platform
 - Compatible with hardware





Development Requirements

- JMyron Image Processing Library
 - Compatible with Processing
 - Easy to use





Environment Requirements

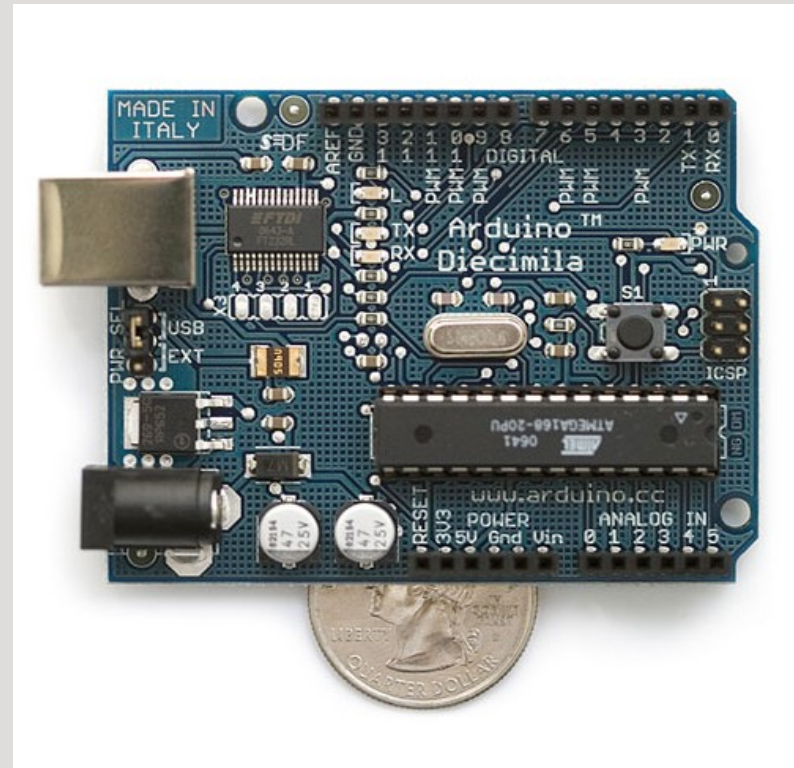
- Windows XP or Vista
- Mac OS 10.5 Intel Chip
- JRE 1.5 or better





Hardware Requirements

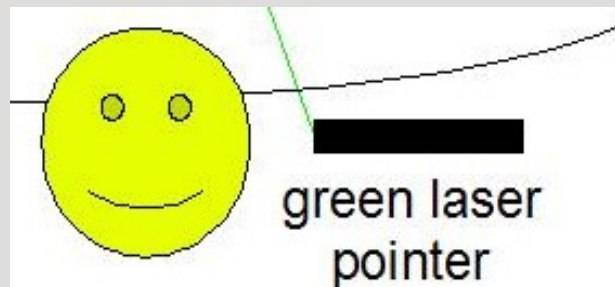
- Projector
- Web camera
- Projection Surface
- Laser Pointer
- Arduino micro-controllers





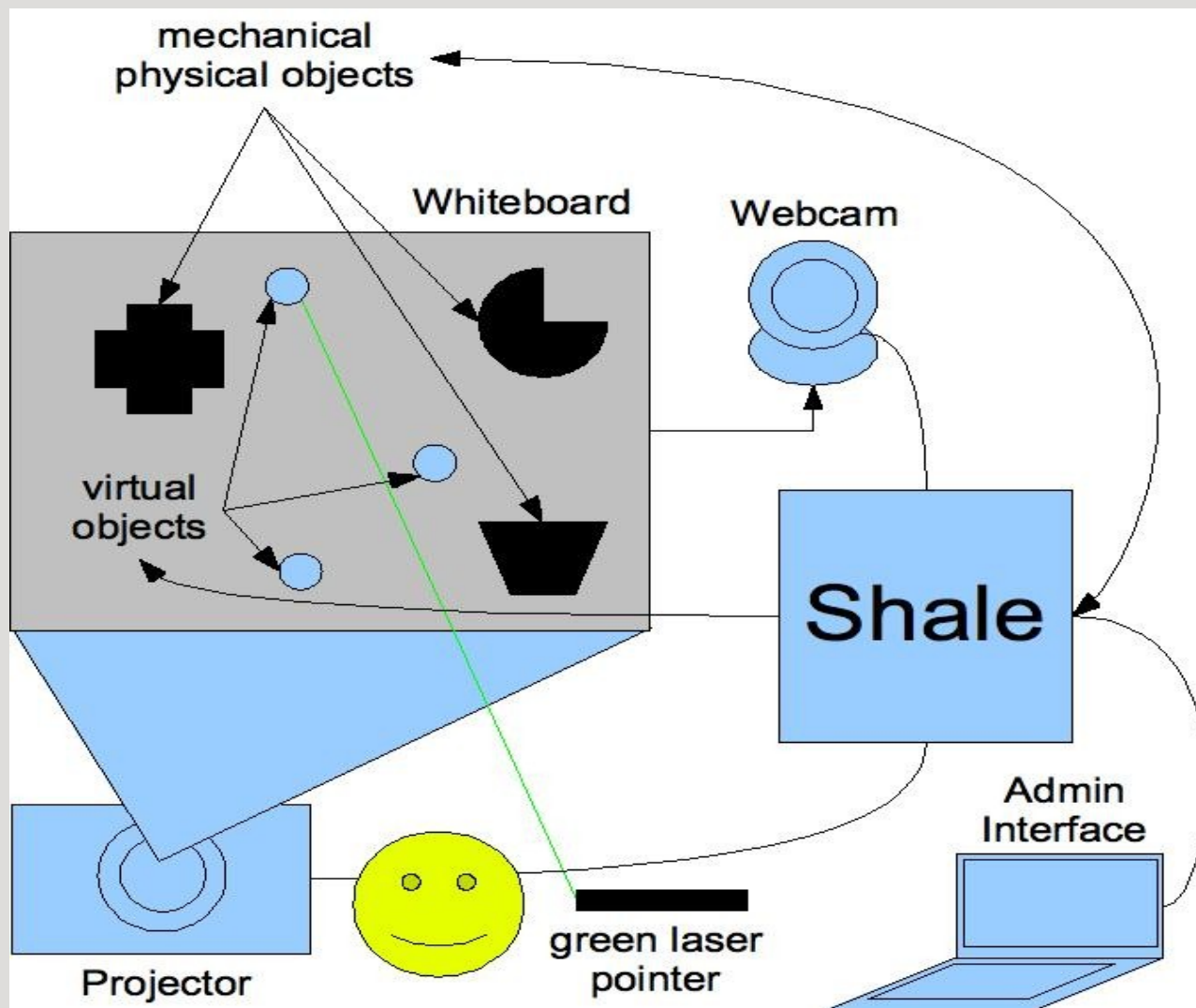
Functional Requirements

- Interactions between physical and virtual objects
 - Project virtual objects
 - Screen Update Interval
 - Detect and control physical objects
 - Respond to user actions





Conceptual Overview





Presentation Focus

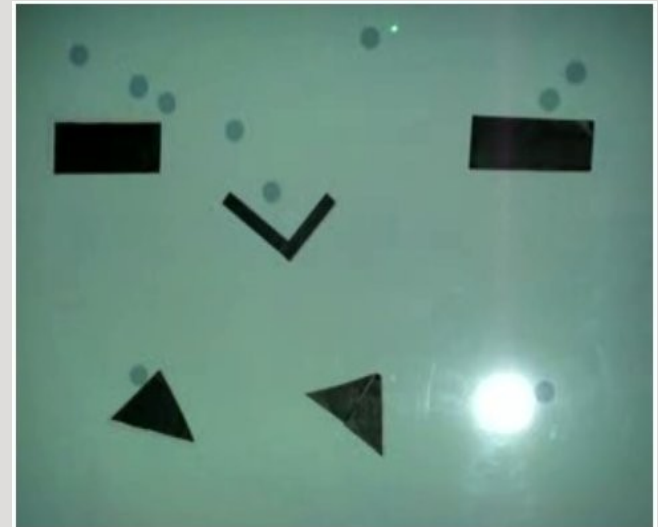
- Overview of the Project
- Software Demonstration
- Overview of the Design & Implementation
 - Requirements
 - **User Interface Design**
 - Architecture
- Known Issues & Future Work





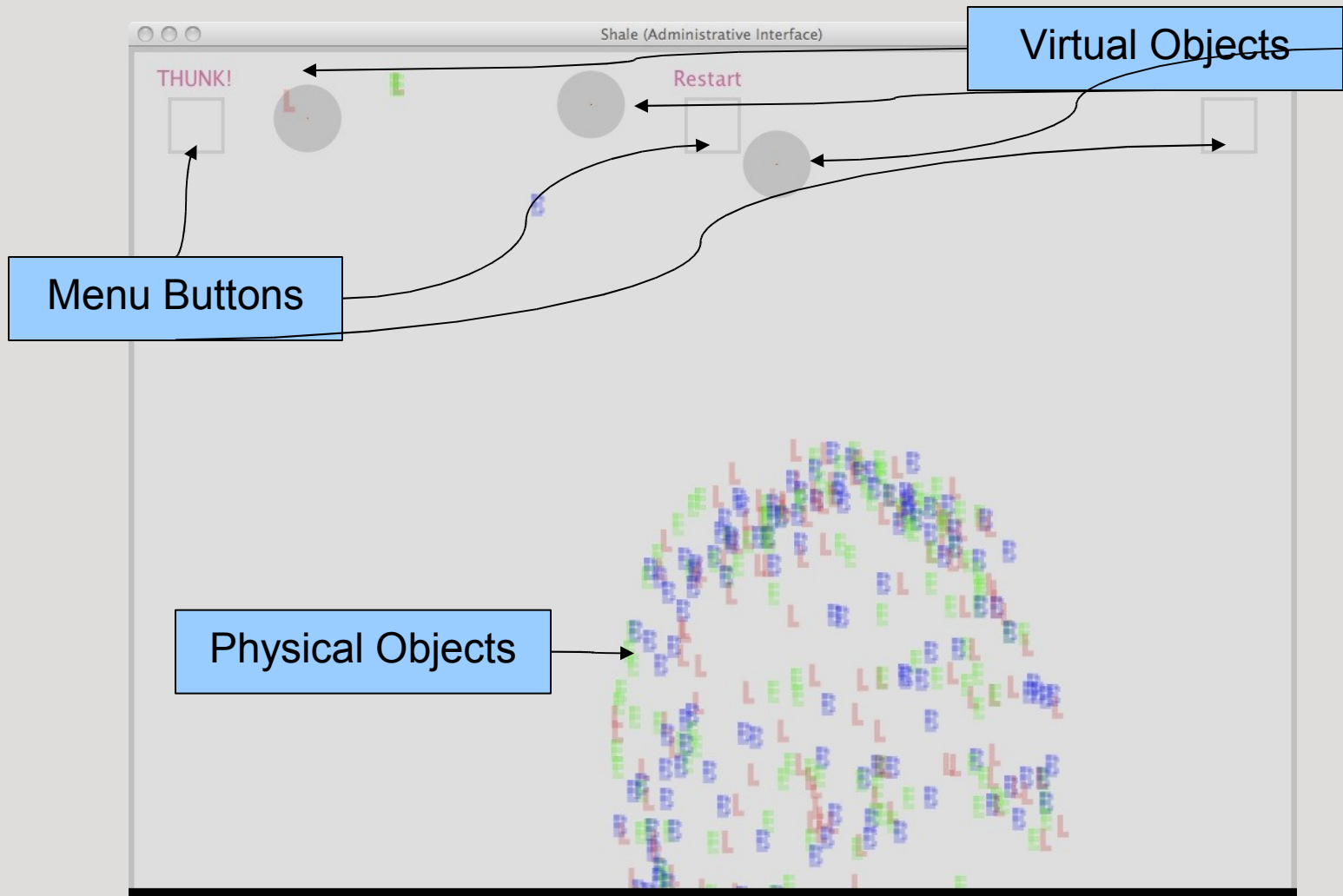
Presentation Focus

- Overview of the Project
- Software Demonstration
- Overview of the Design & Implementation
 - Requirements
 - **User Interface Design**
 - Standard User Interface
 - Administrative Interface
 - Architecture
- Known Issues & Future Work





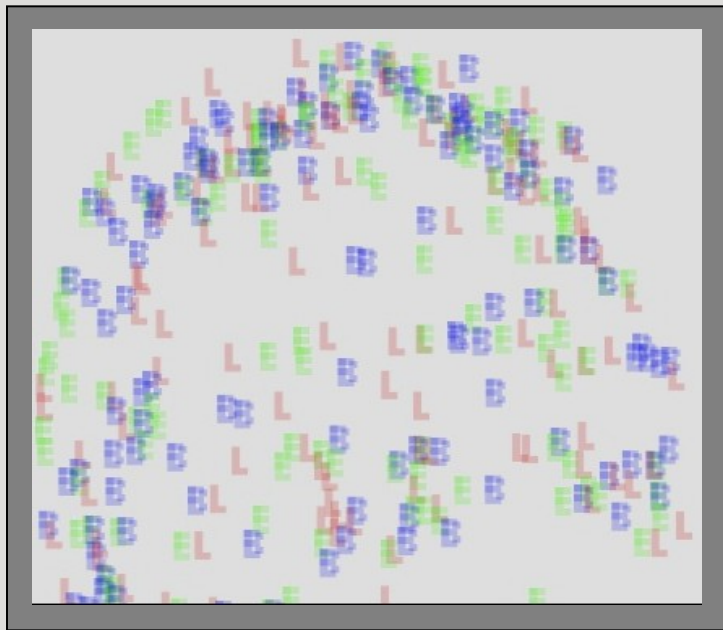
Shale: The Standard Interface





Current View

- Simple display onto the Stage
 - Physical Objects (letters)
 - Virtual Objects (gray)



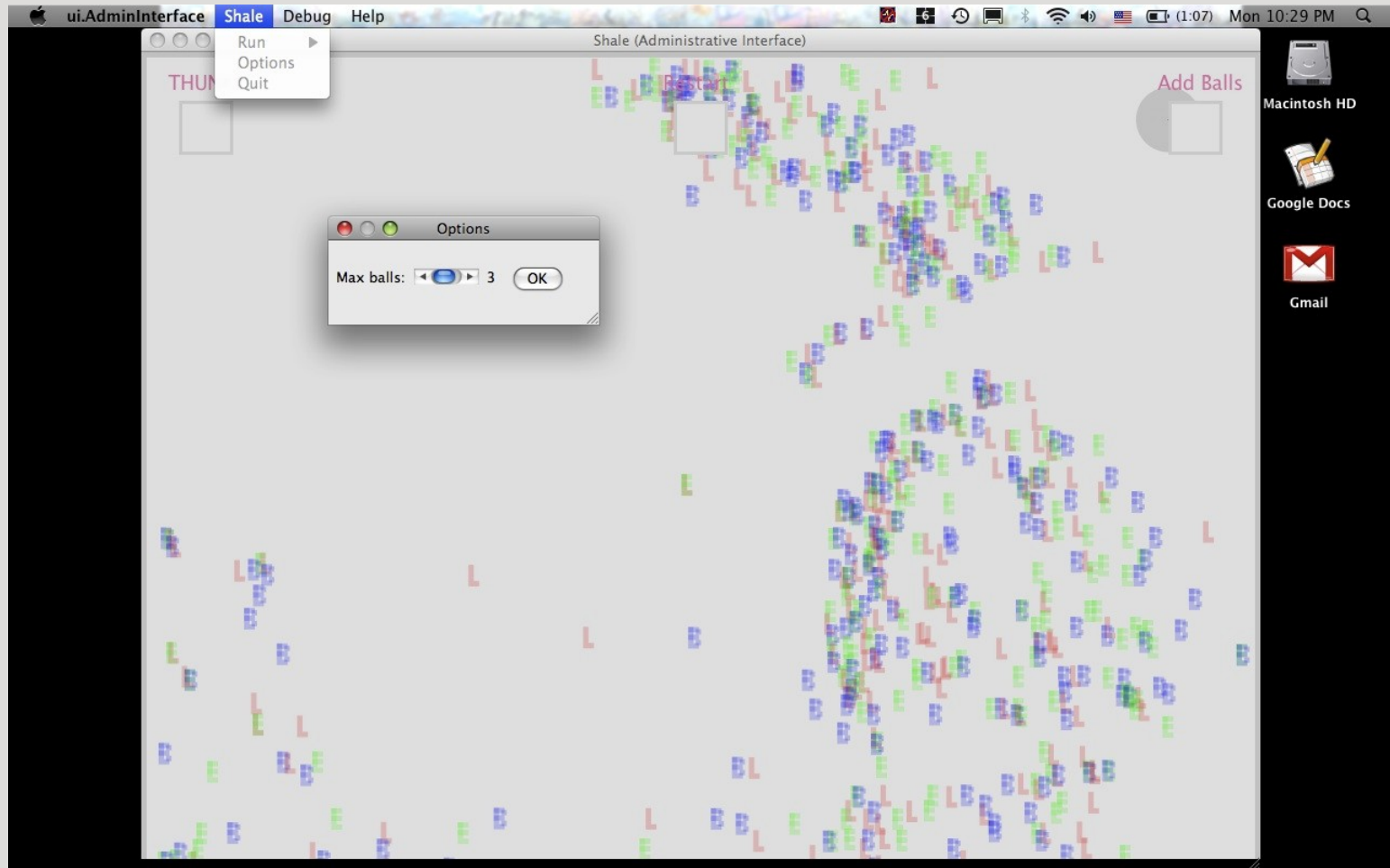


Menu Buttons





Administrative Interface





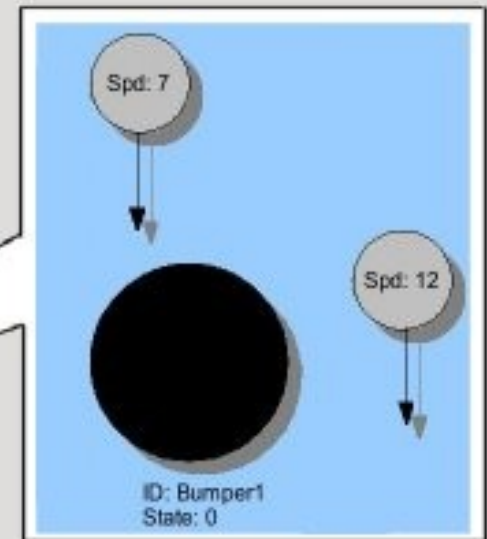
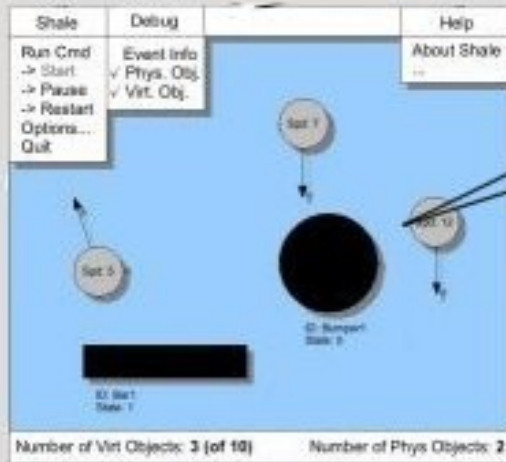
Current View

- **Physical Objects**

- Can toggle debugging features

- **Virtual Objects**

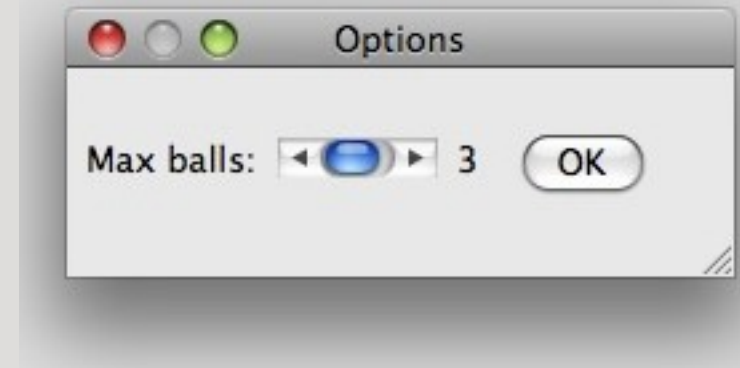
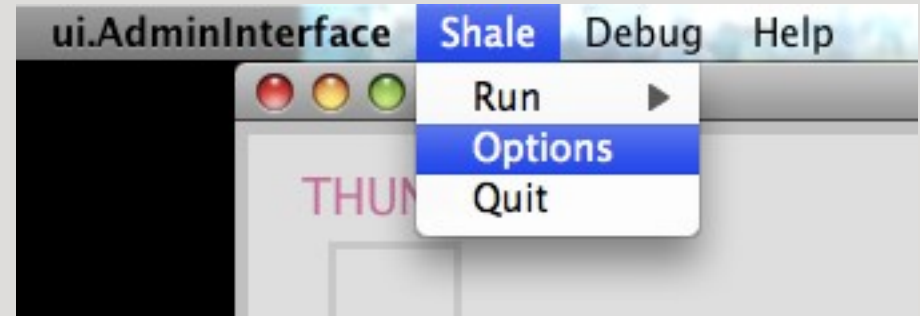
- Can toggle debugging features





Menu Bar: Shale – Options

- Options...
 - Opens dialog box to modify user preferences:
 - Maximum virtual object quantity
 - Objects are removed (oldest first) when this value is exceeded





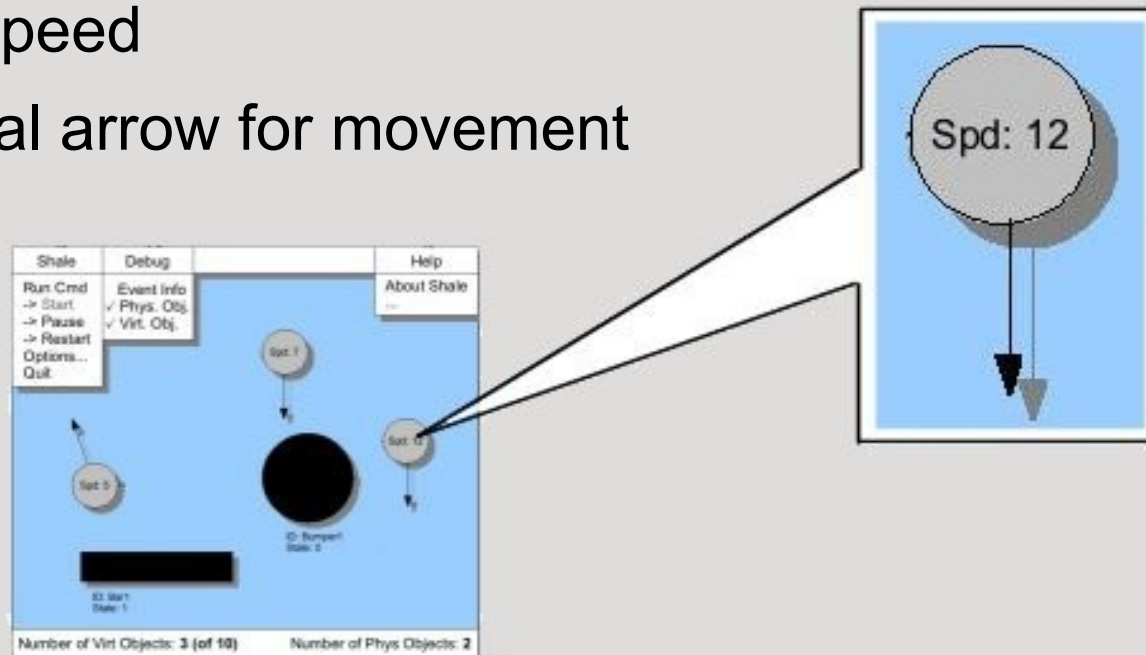
Menu Bar: Shale - Quit





Debug: Virtual Objects

- **Virtual Objects**
 - current speed
 - directional arrow for movement

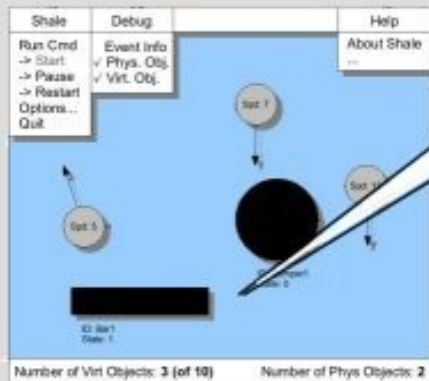




Debug: Physical Objects

- **Physical Objects**

- Display the ID of each object
- Display the current state of each object

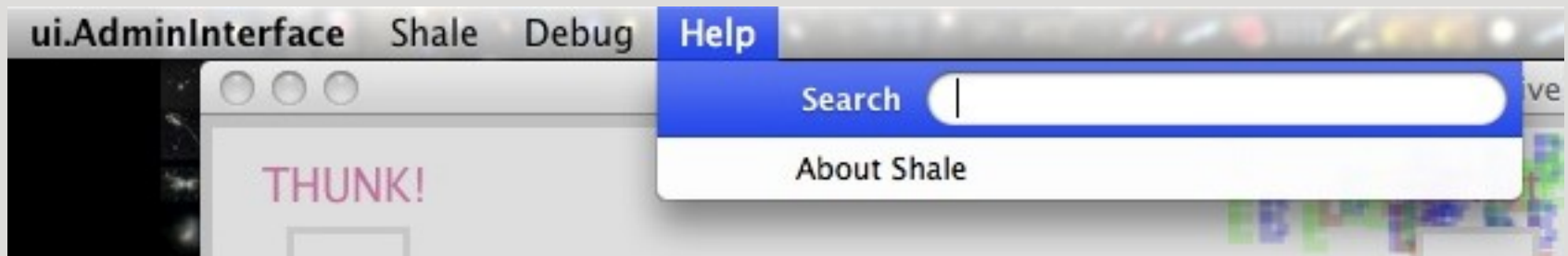


ID: Bar1
State: 1



Menu Bar: Help

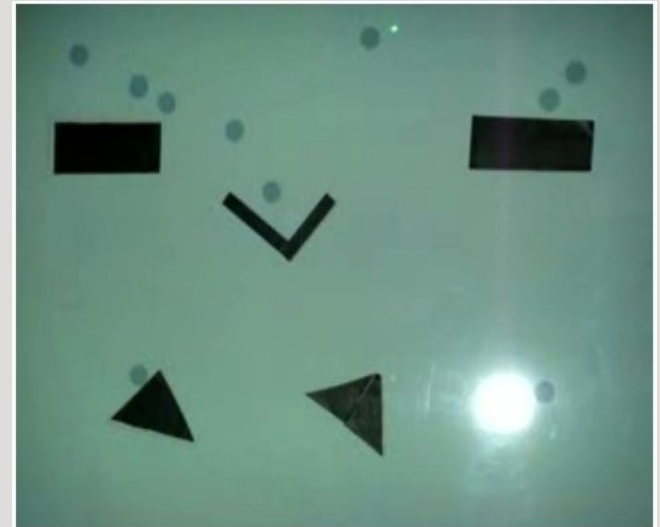
- **About Shale**
 - information about the Shale software
 - brief background
 - link to the project website
- ...
 - Other useful topics





Presentation Focus

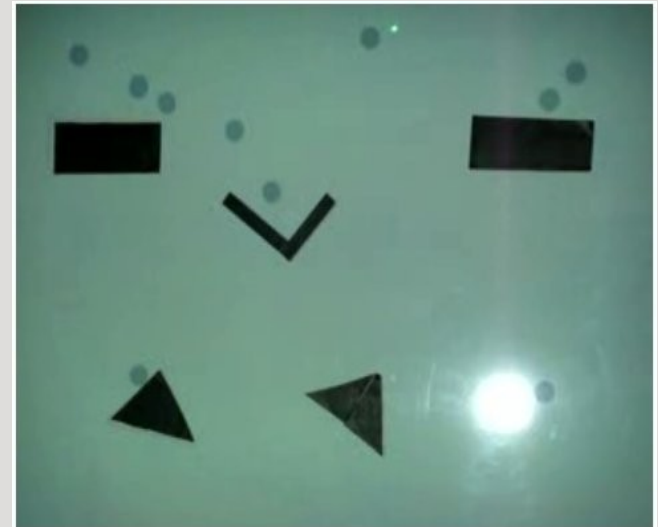
- Overview of the Project
- Software Demonstration
- Overview of the Design & Implementation
 - Requirements
 - User Interface Design
 - **Architecture**
- Known Issues & Future Work





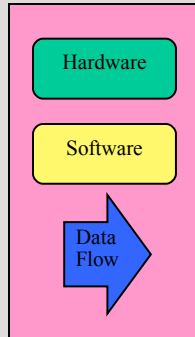
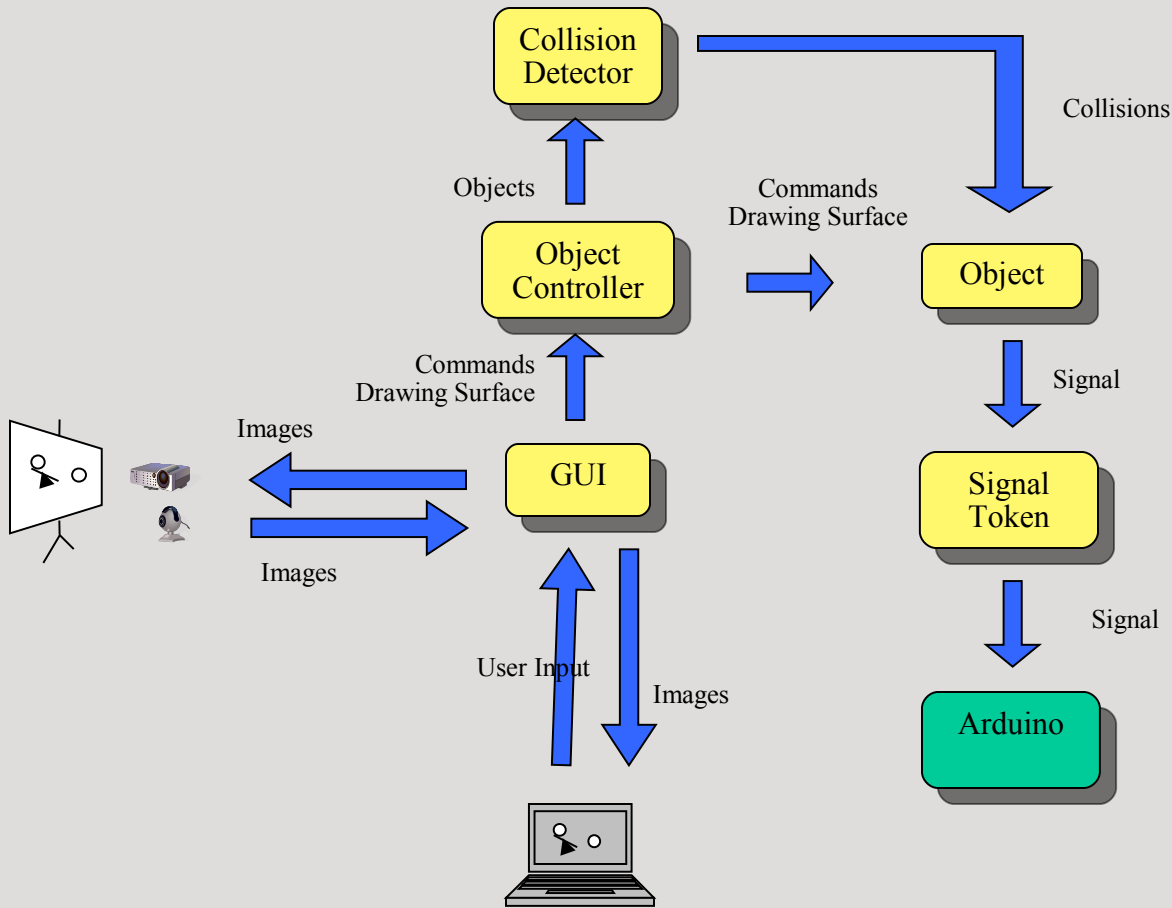
Presentation Focus

- Overview of the Project
- Software Demonstration
- Overview of the Design & Implementation
 - Requirements
 - User Interface Design
 - **Architecture**
 - Overview
 - Shale Classes
- Known Issues & Future Work



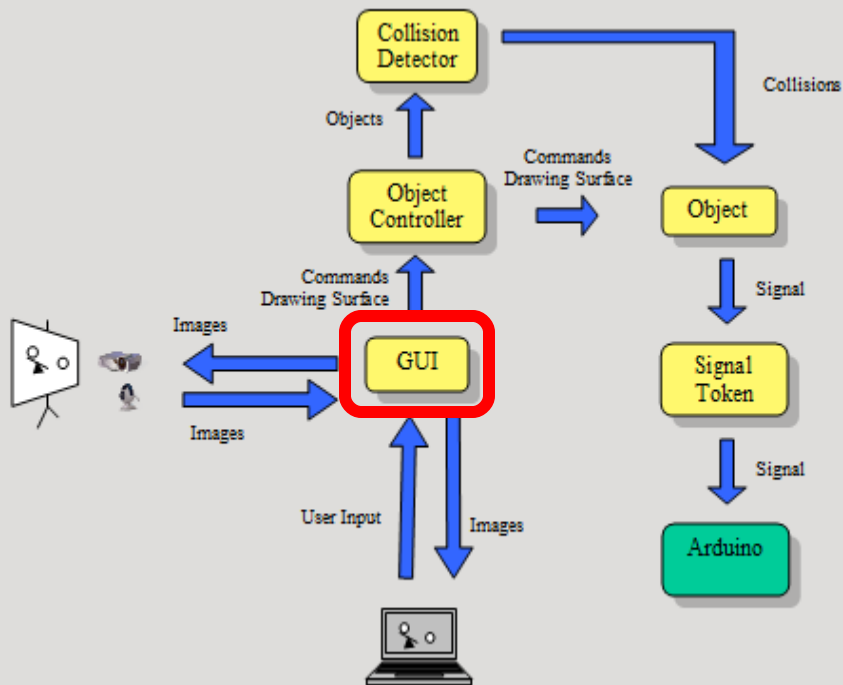


Shale Architecture





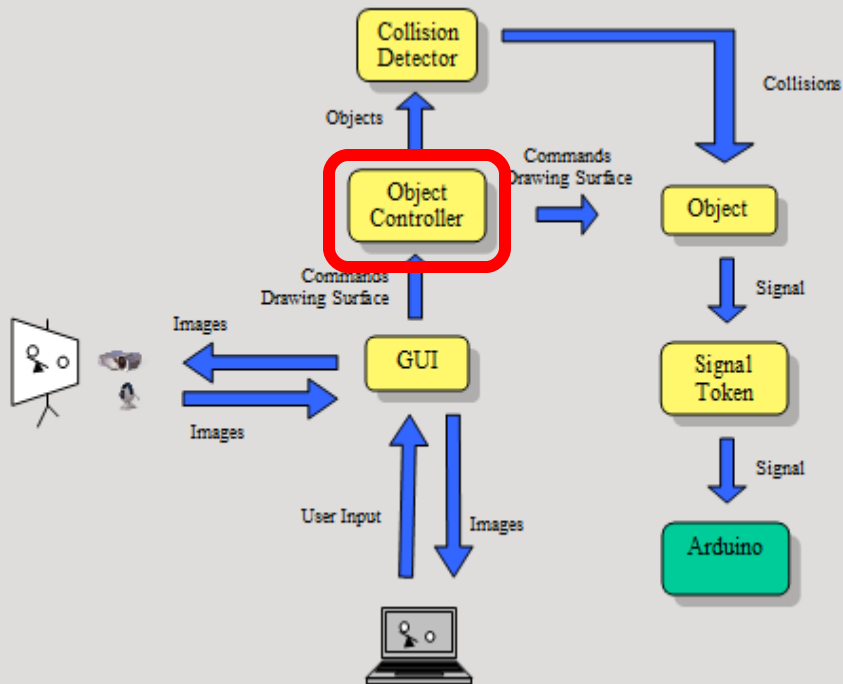
The Graphical User Interface



- Handle user input
- Process images
- Direct Object Controller



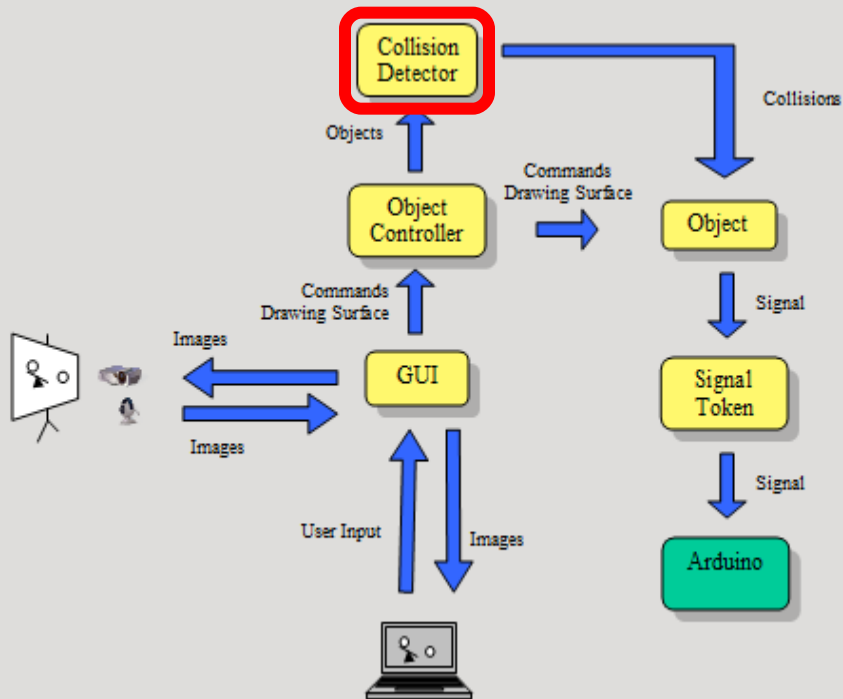
The Object Controller



- Maintain Objects
- Move Objects
- Get Signal Tokens



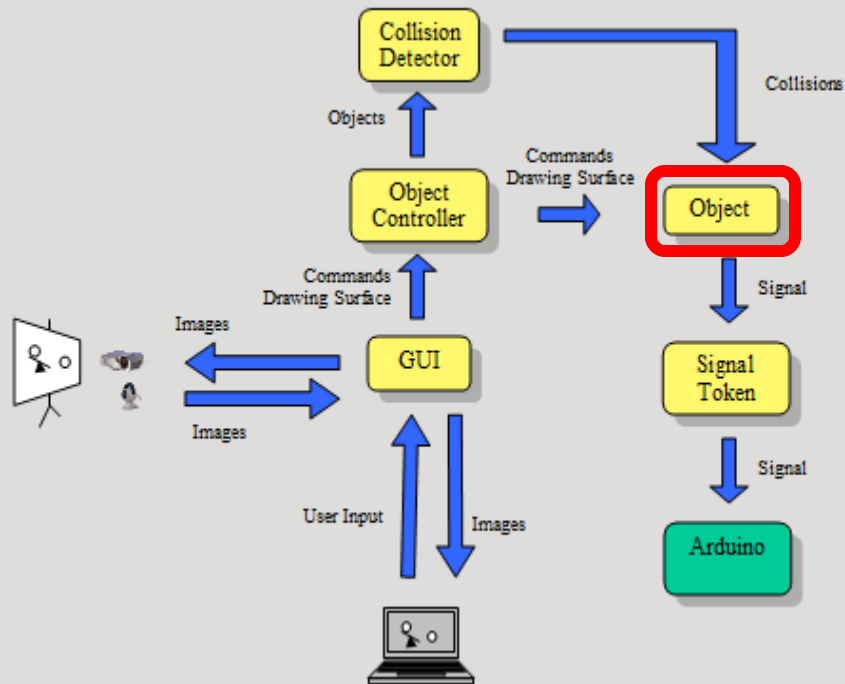
The Collision Detector



- Identify collisions
- Calculate velocities
- Activate Objects



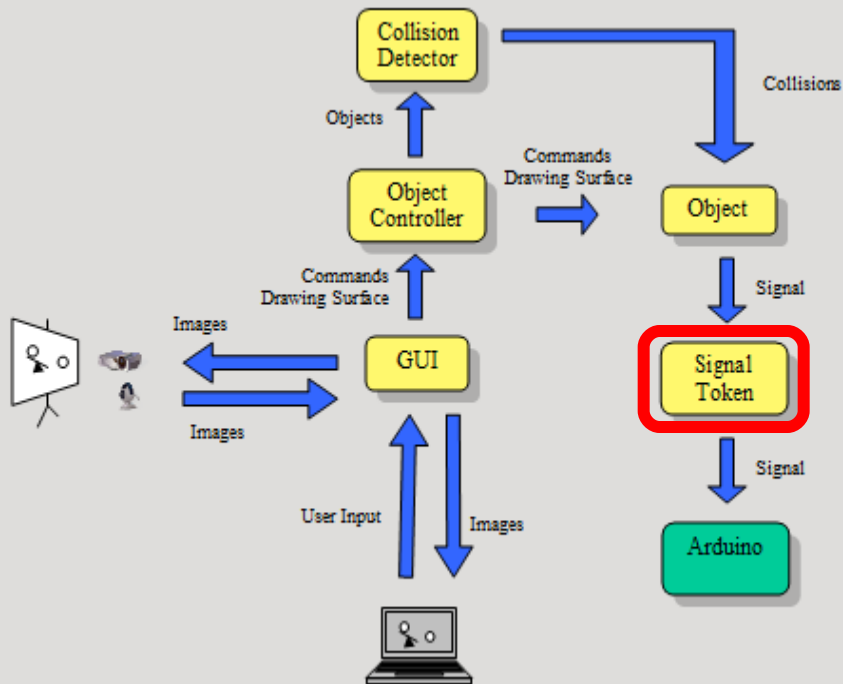
The Objects



- Position, speed, size
- Implement collision



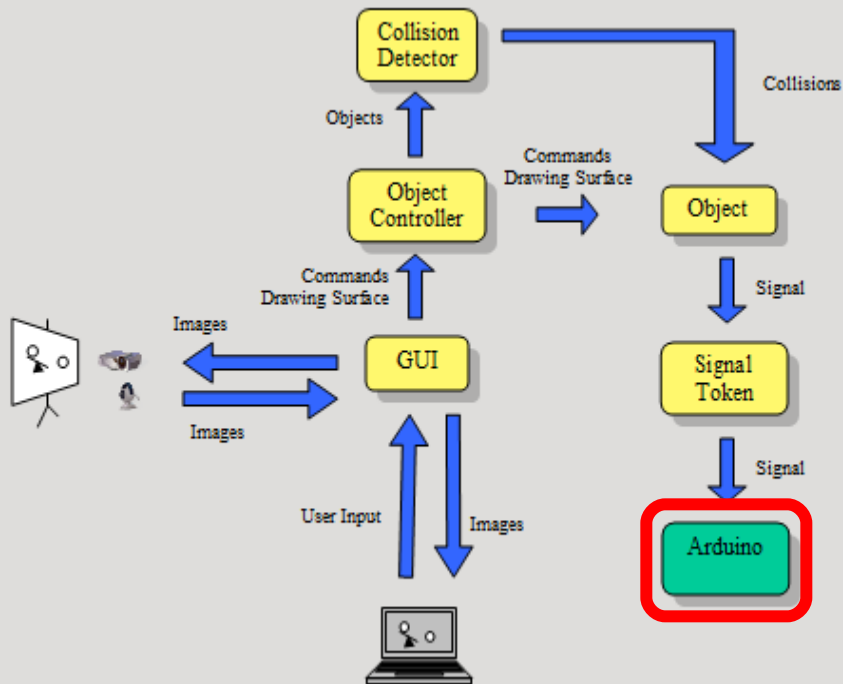
The Signal Tokens



- Unique Arduino signal
- Send signals



The Arduinos

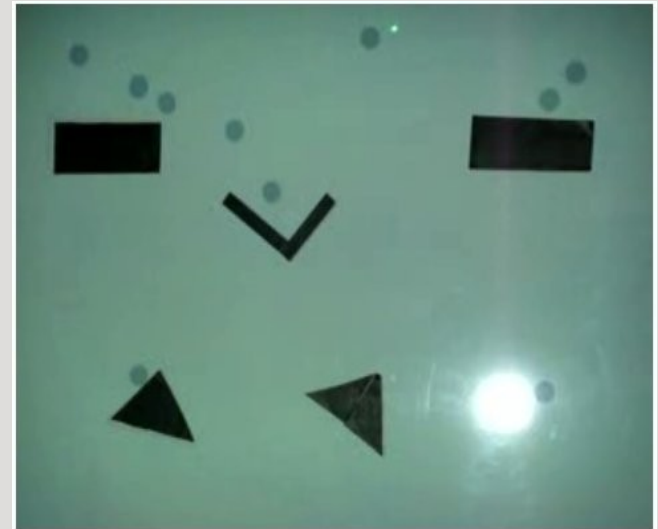


- Receive signals
- Do collision behavior



Presentation Focus

- Overview of the Project
- Software Demonstration
- Overview of the Design & Implementation
 - Requirements
 - User Interface Design
 - Architecture
- **Known Issues & Future Work**





Known Issues

- Calibration is a time-consuming process
- Collision detection
 - Balls stick together
- Macintosh webcam
 - Unable to detect objects
- Casing/Weight issues
 - Styrofoam flimsy, but lighter
 - Cubisto heavier but sturdy





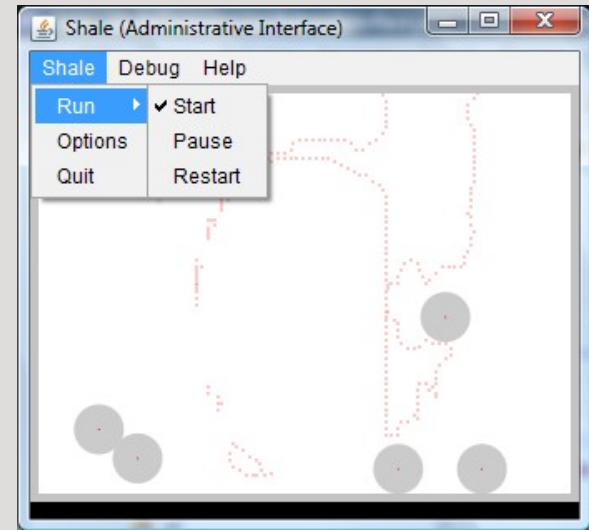
Future Work

- Improve image detection library
 - Each object is detected as many objects
- Improve casing
- Improvements upon the collision detection
 - Prevent objects from falling through other objects



Summary

- Overview of the Project
 - The Class
 - The Problem
 - The Solution
- Software Demonstration
- Overview of the Design & Implementation
 - Requirements
 - User Interface Design
 - Architecture
- Known Issues & Future Work





Q & A

The presentation is over...
But before we go, are there any
QUESTIONS?