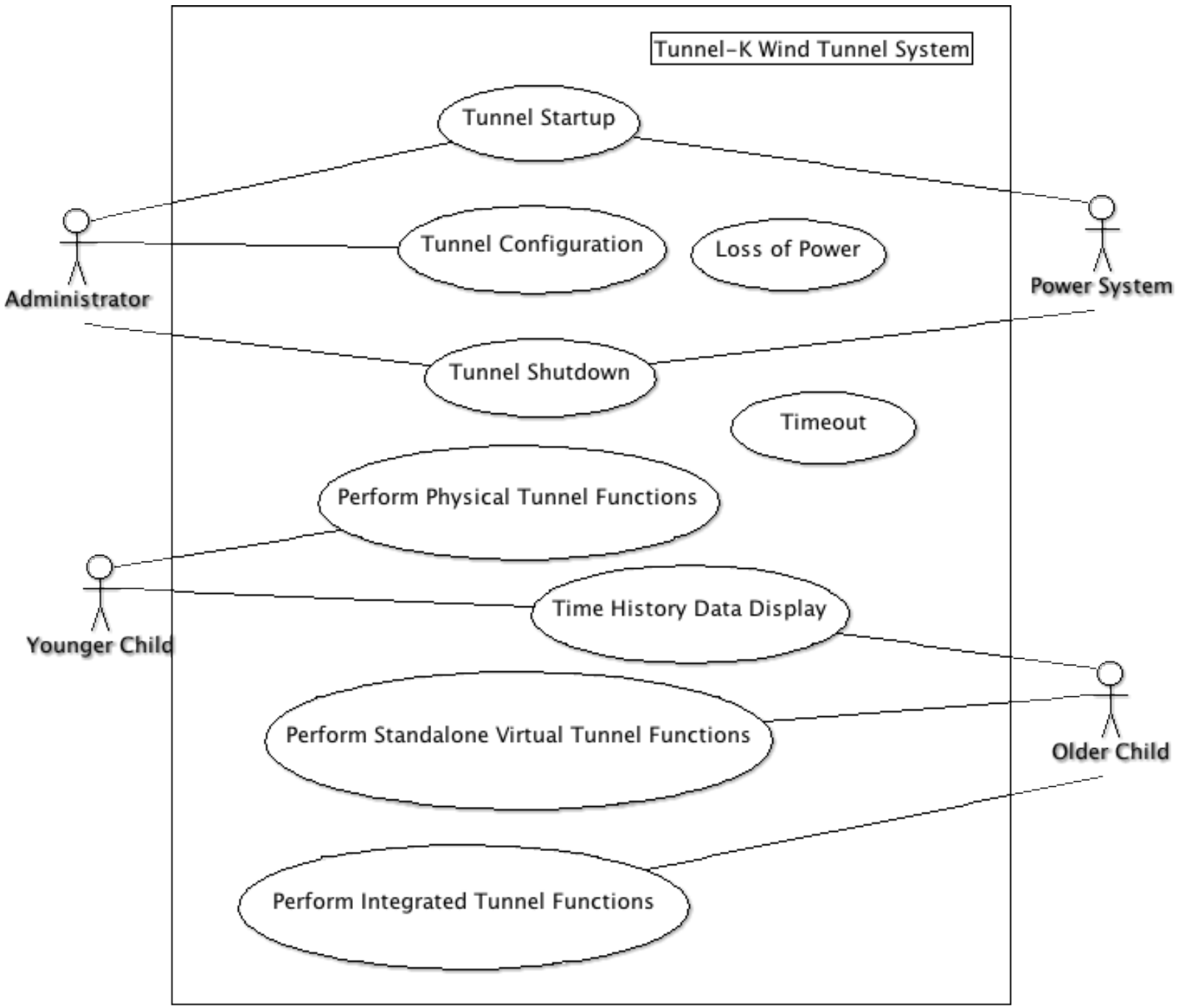
Tunnel-K Prototype Review

December 5, 2011

Agenda

- Tunnel-K Introduction
  - Use Cases
  - Exhibit Flow
- GUI Mockup
- Virtual Wind Tunnel
  - Environment
  - Whiteboard
  - Visualization
- Physical Wind Tunnel
  - Hall Effect Sensor
  - Fan Motor Control
  - Temperature/Humidity
  - Ethernet Shield
- Future Work

Use Case Perspective



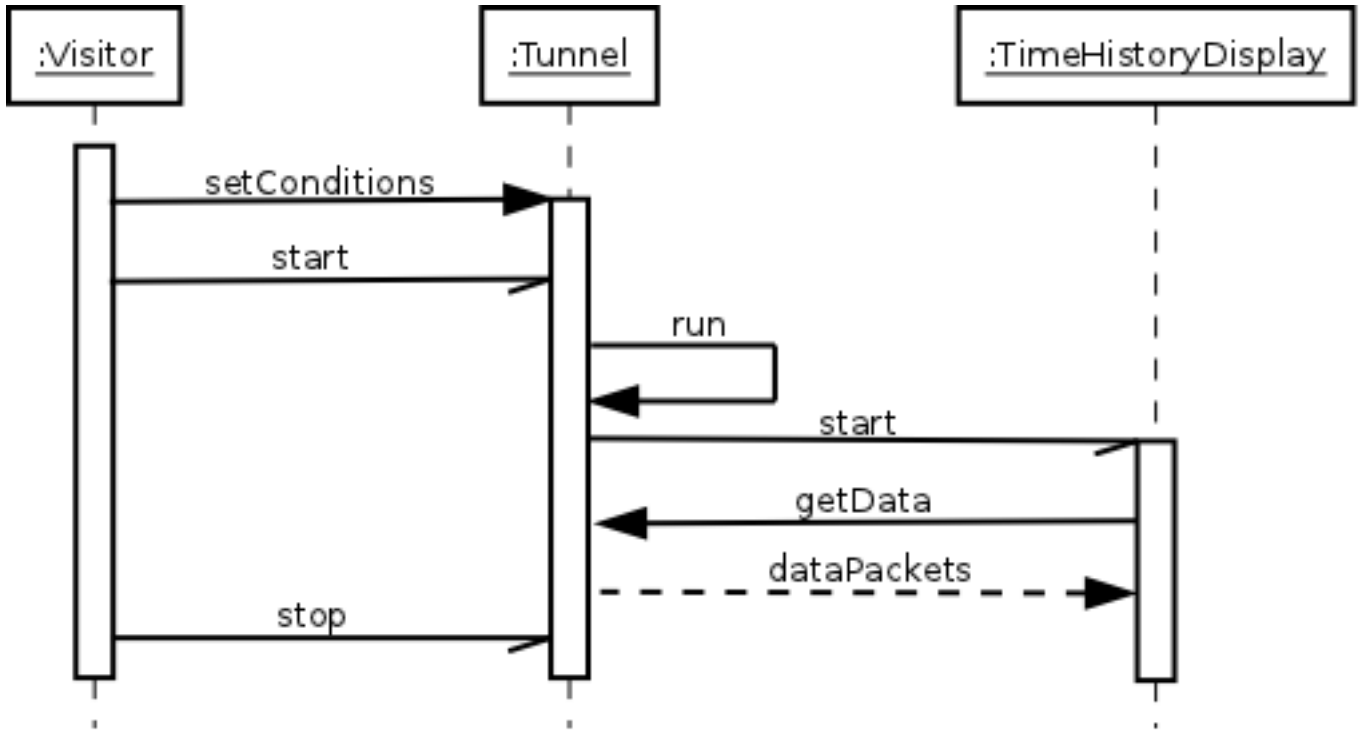


Exhibit Flow for a Younger Visitor

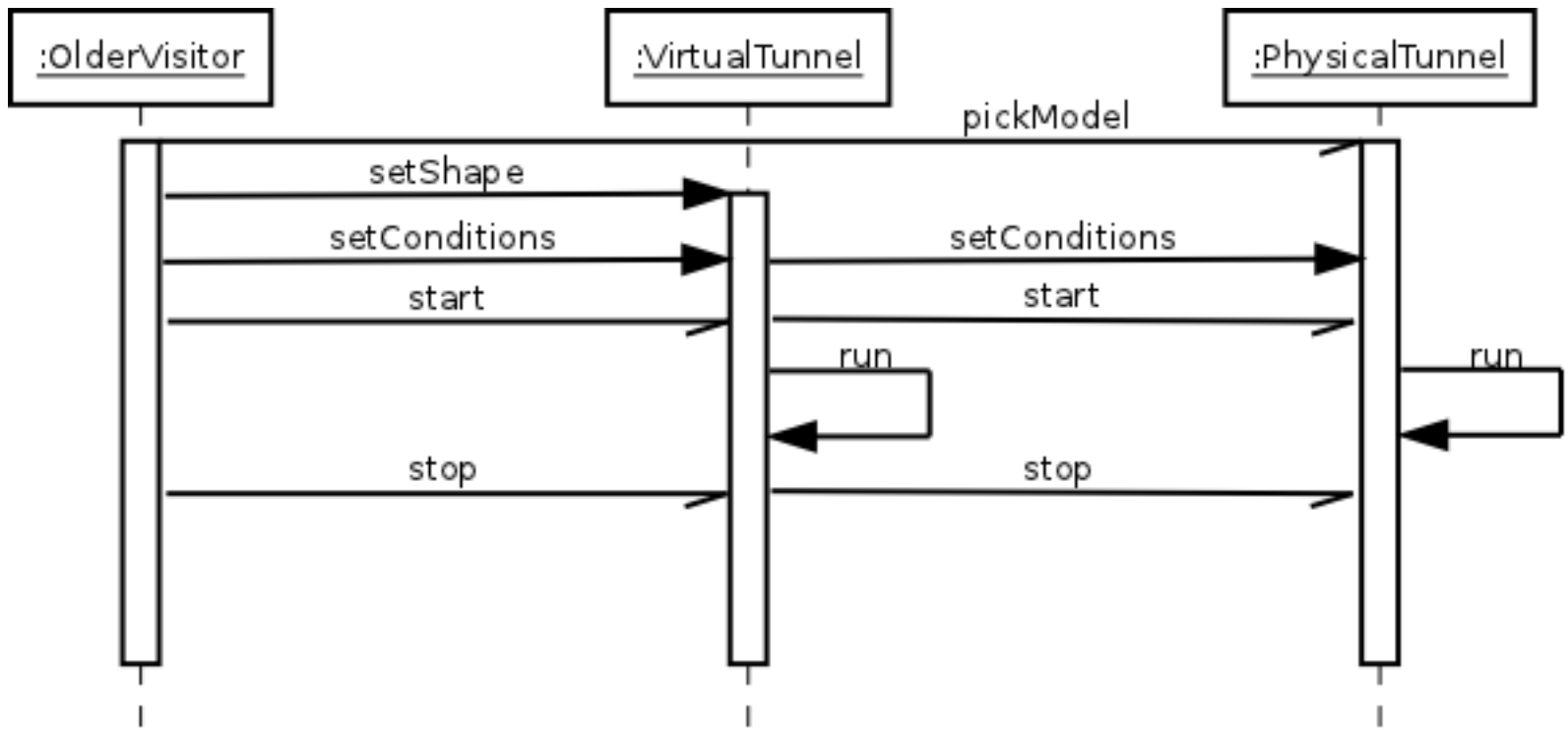
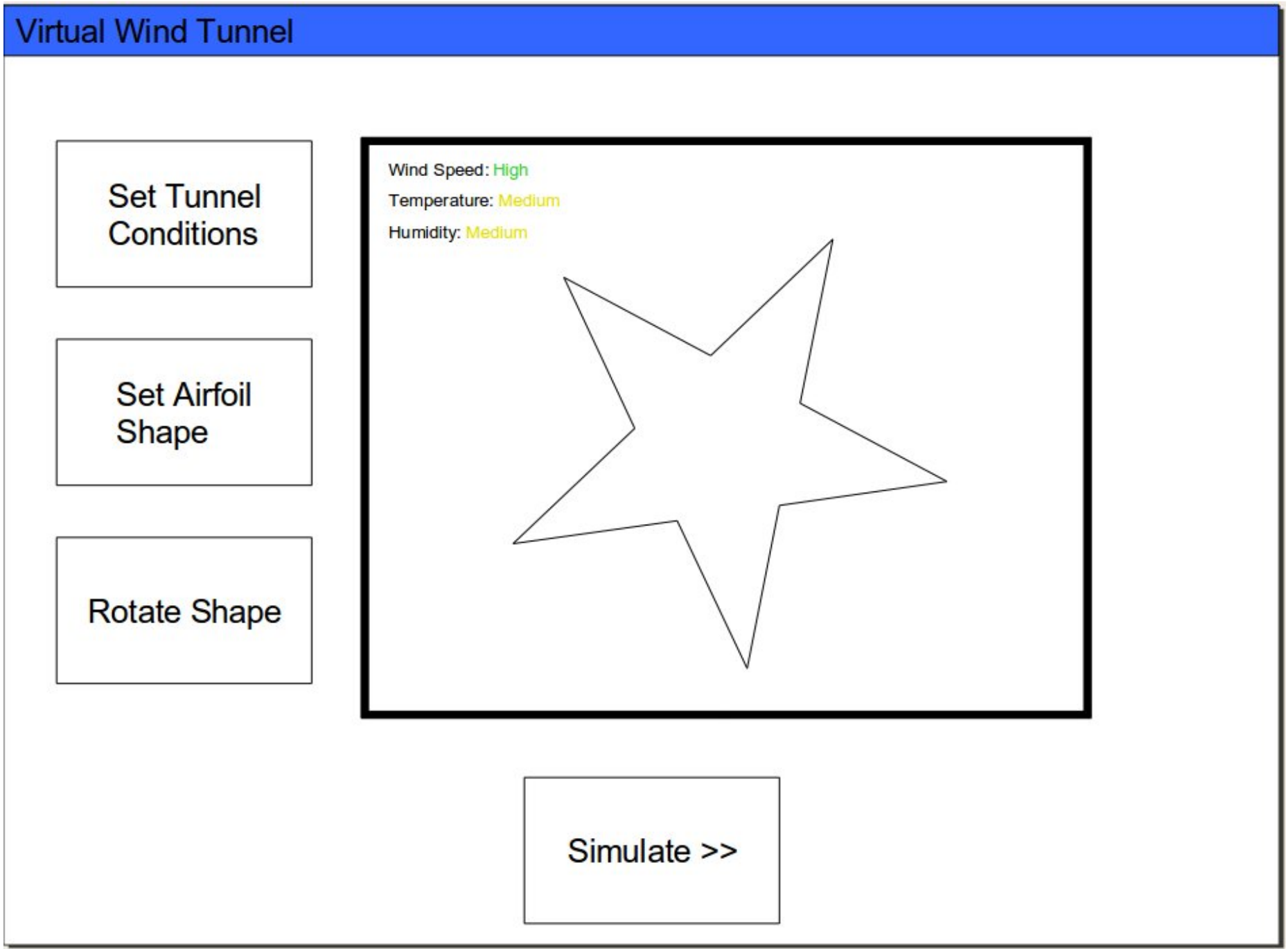
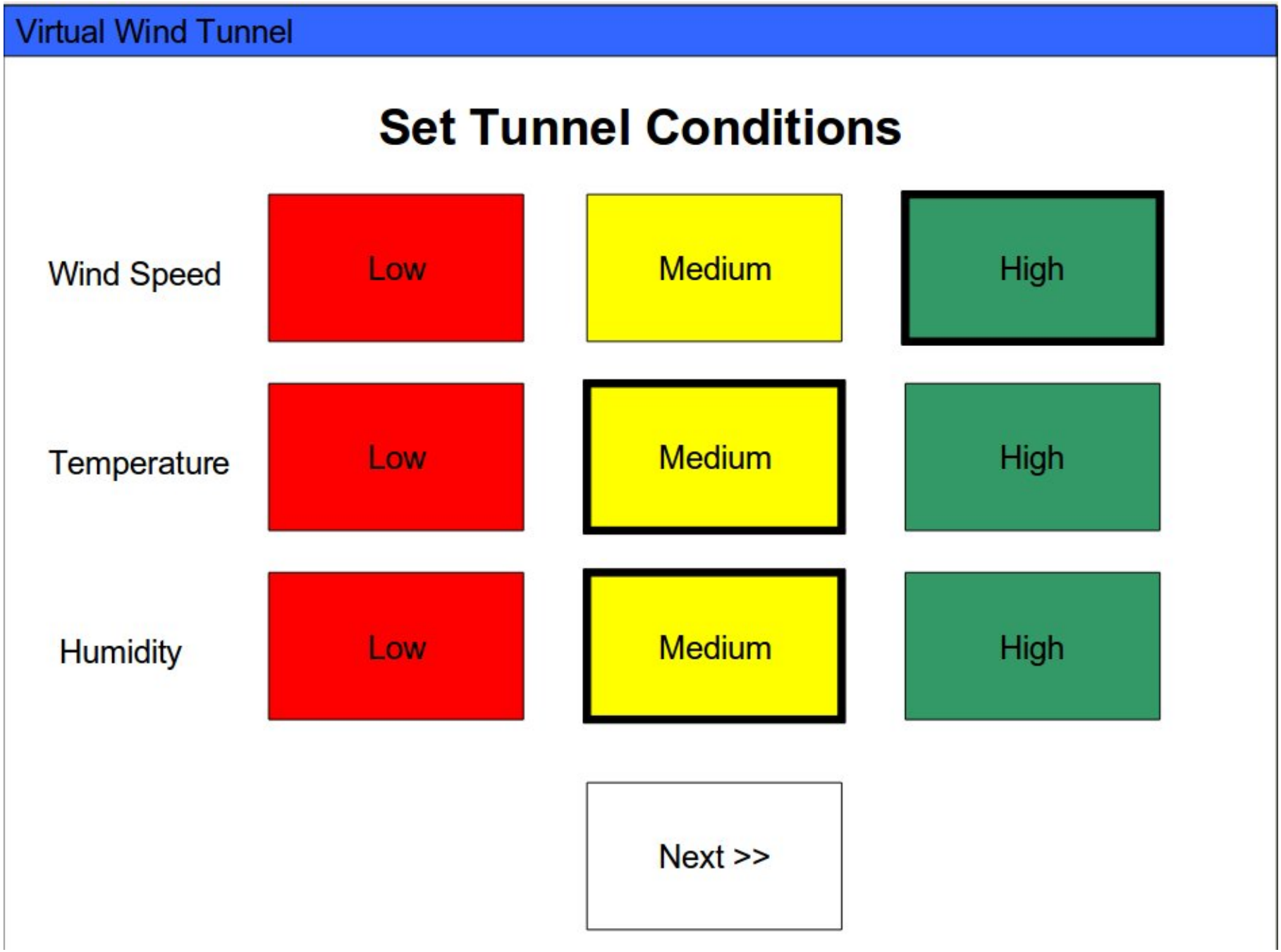


Exhibit Flow for an Older Visitor

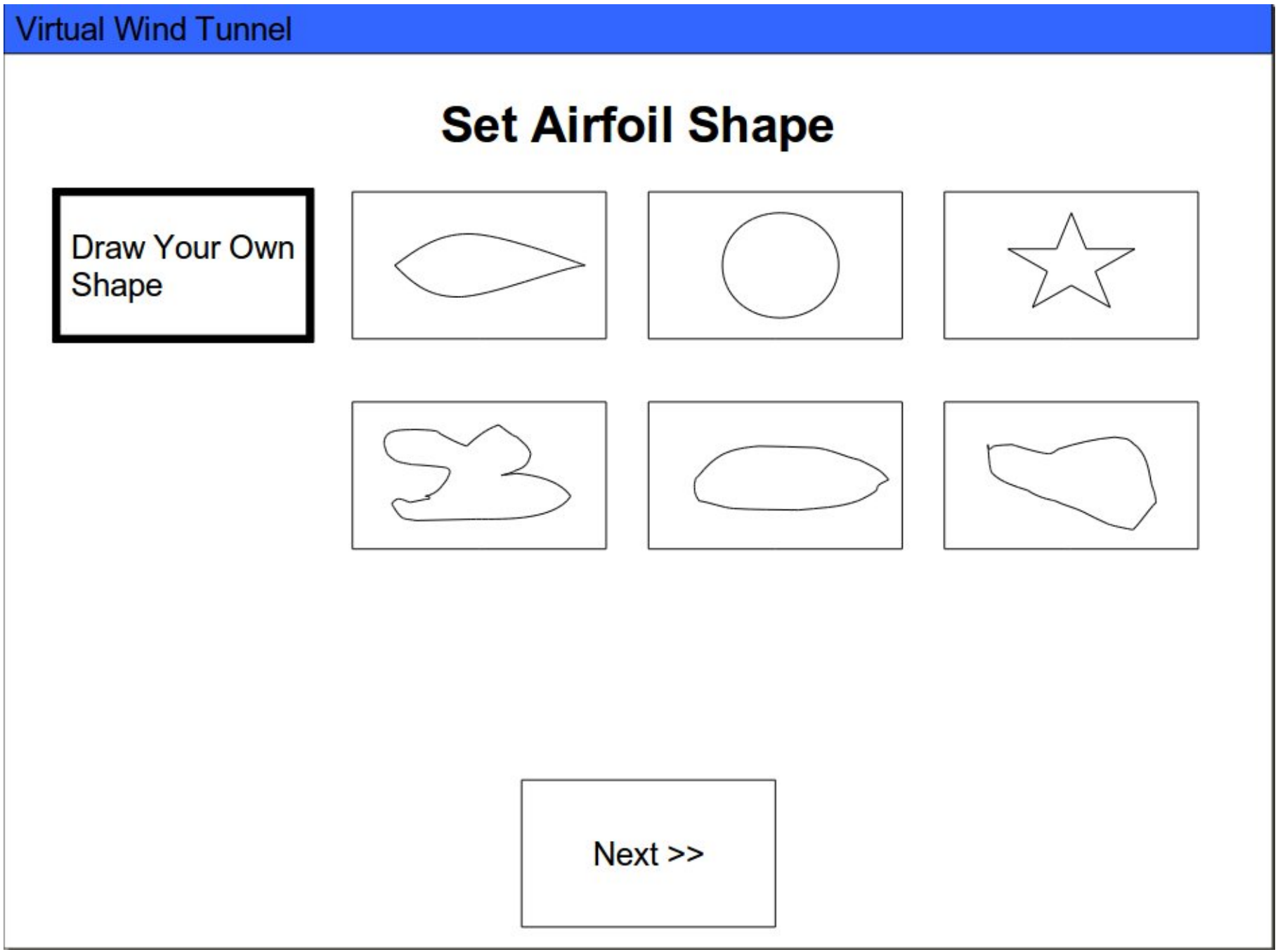
GUI Mockup


Virtual Wind Tunnel
Set Tunnel Conditions
Set Airfoil Shape
Rotate Shape
Wind Speed: High
Temperature: Medium
Humidity: Medium
Simulate >>

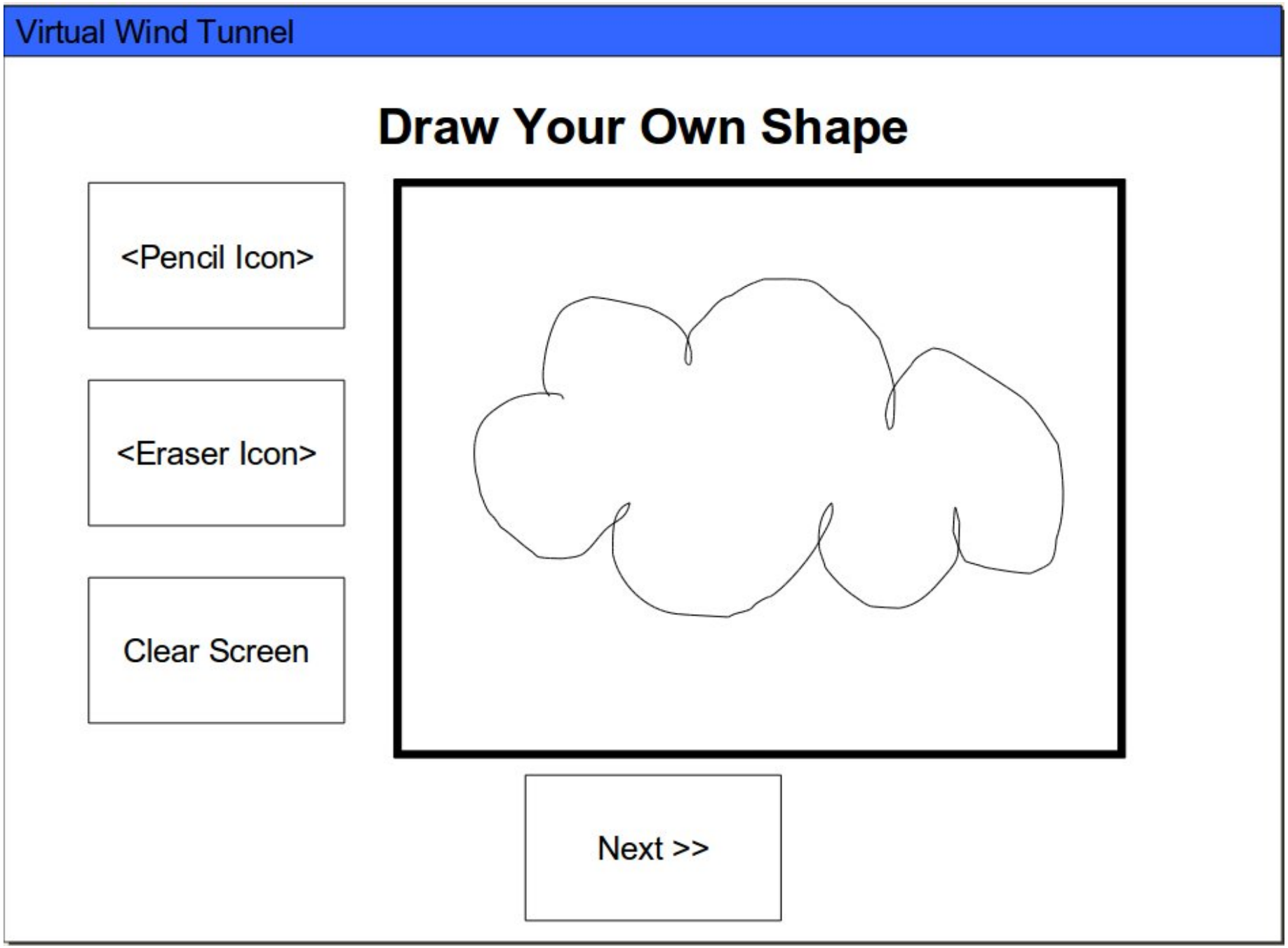
GUI Mockup


Virtual Wind Tunnel
Set Tunnel Conditions
Wind Speed
Low
Medium
High
Temperature
Low
Medium
High
Humidity
Low
Medium
High
Next >>

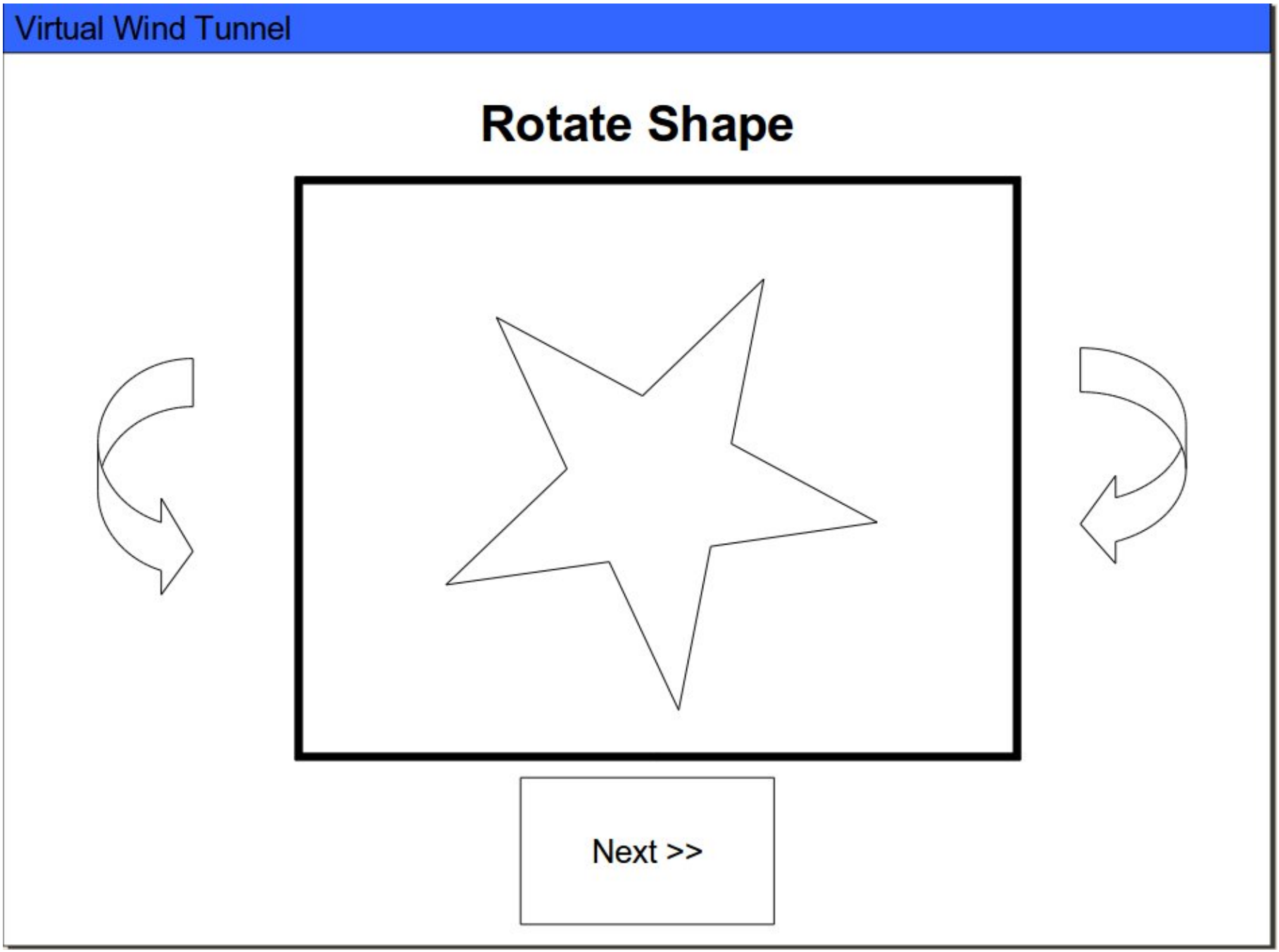
GUI Mockup


Virtual Wind Tunnel
Set Airfoil Shape
Draw Your Own Shape
Next >>

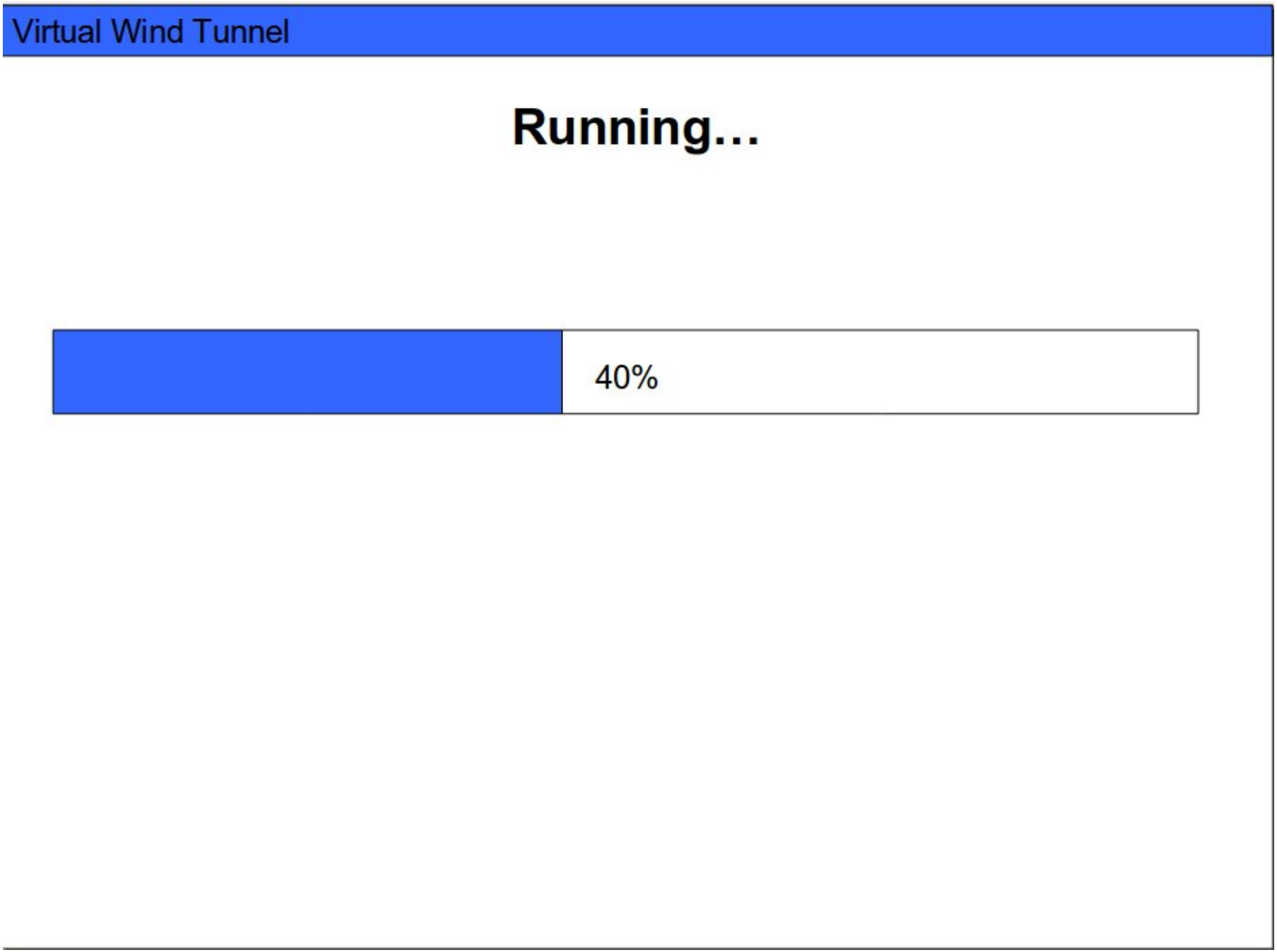
GUI Mockup


Virtual Wind Tunnel
Draw Your Own Shape
<Pencil Icon>
<Eraser Icon>
Clear Screen
Next >>

GUI Mockup



GUI Mockup



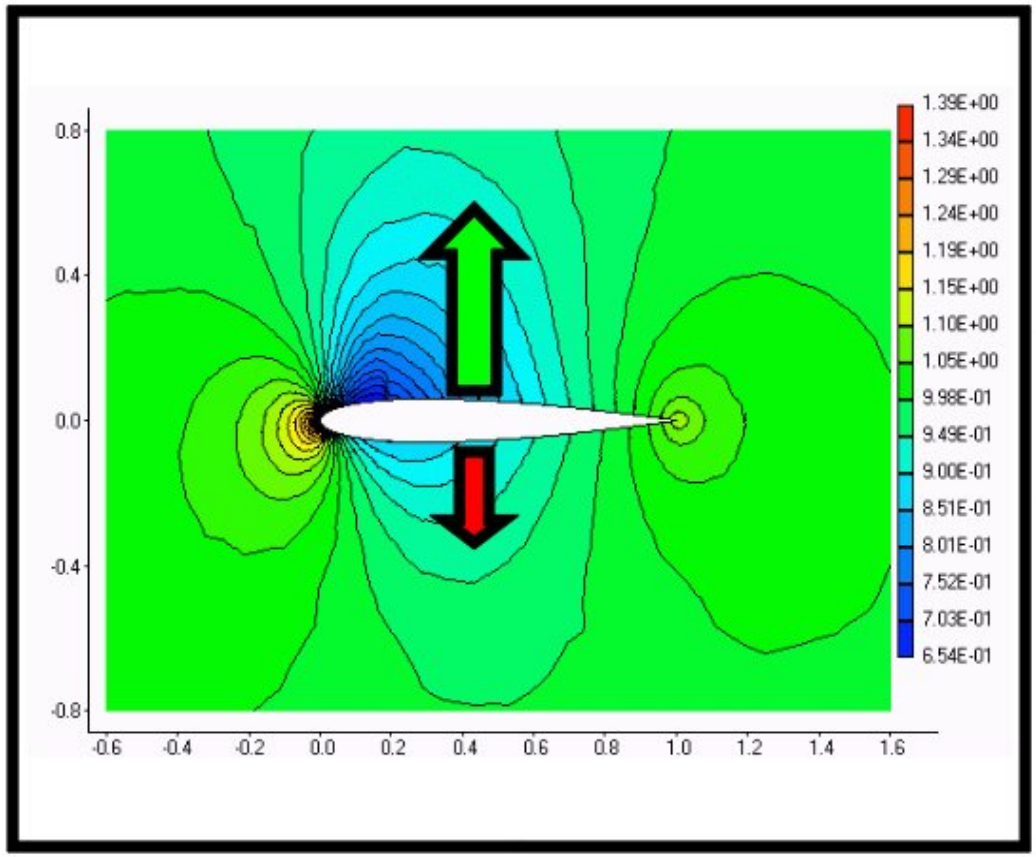
GUI Mockup

Virtual Wind Tunnel

Results

Tweet Results

Email Results



Lift: 1.895

Drag: 0.643

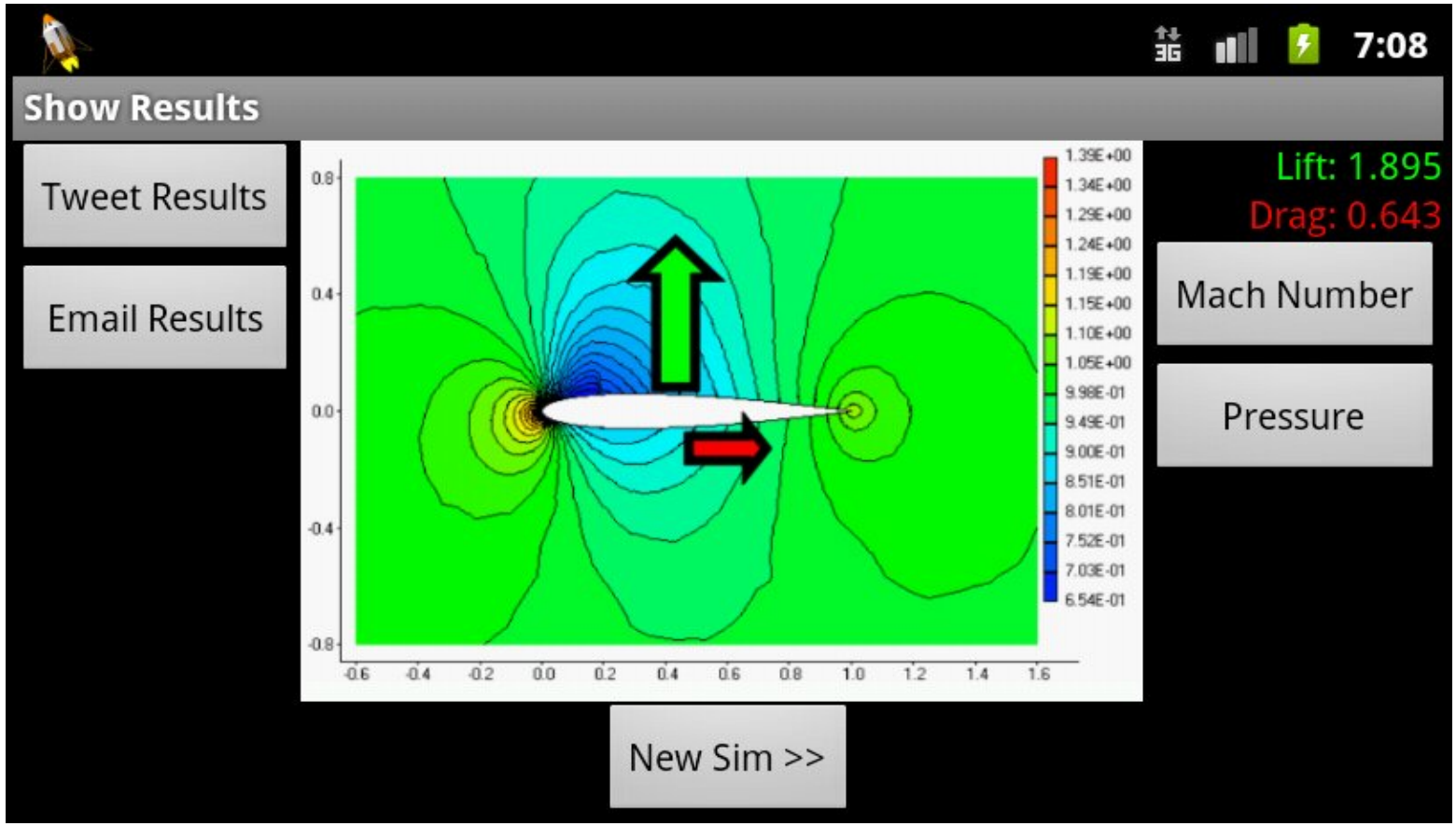
Mach Number

Pressure

New Sim >>

Virtual Wind Tunnel

- Environment
- Whiteboard
- Visualization



Virtual Wind Tunnel: Environment

- Android 2.3.3 target
  - To support x-large screen devices
- NDK
  - To utilize existing 2dflowsolver code
  - Tried to utilize unchanged cpp code as much as possible
- Development Environment/workflows
  - Eclipse
    - GUI, refactoring, layout, etc
  - Ant
    - Command line driven, vi, etc
- Testing
  - TunnelkTest project in Eclipse
  - Android JUnit support

Virtual Wind Tunnel: Whiteboard

- Currently based on FingerPaint.java example code
- Problems
  - Some custom shapes won't work
  - Open shapes
  - Concavities
  - Kids draw the darnedest things
- Possible different approaches
  - Bezier curves
  - Limit to only pre-existing shape set

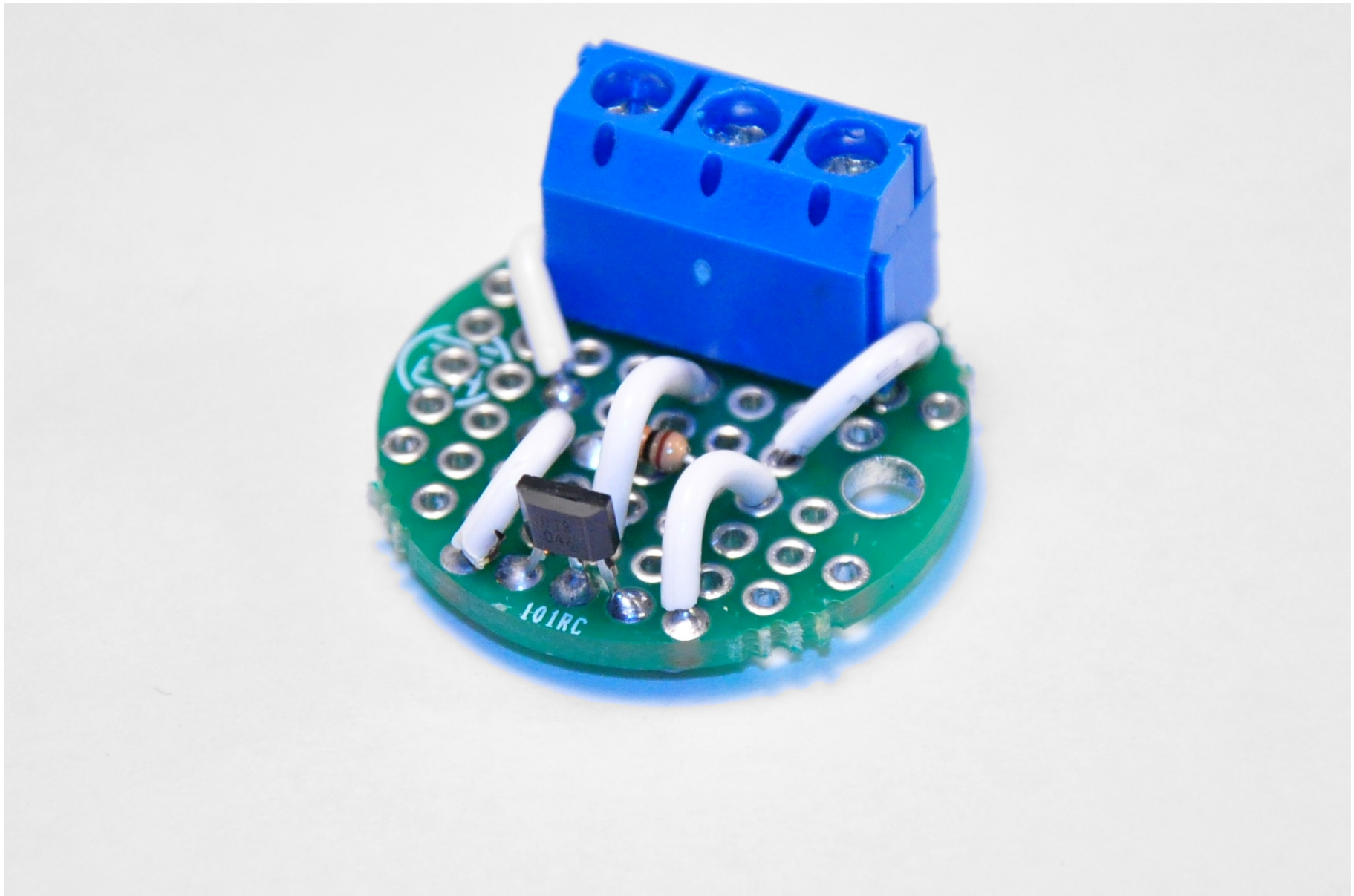
Virtual Wind Tunnel: Visualization

- Only mocked up in prototype
- Possible approaches
  - OpenGL ES
    - Low level graphics library
  - Processing for Android
    - Java-based graphics IDE
  - Pre-render using other apps
    - Only an option if limited to pre-existing shape set

Physical Wind Tunnel

- Hall Effect Sensor
- Fan Motor Control
- Temperature/humidity
  - Ethernet shield
  - Tweeting / logging capabilities

Physical Wind Tunnel: Hall Effect Sensor



Physical Wind Tunnel: Hall Effect Sensor

- Sensor to be mounted within 0.5" of fan blade mounted magnets
  - Any further distance causes invalid readings
- Arduino runs fast enough to support an readings of over 6700 RPM, far faster than the fan being used will run
- Prototype displays both hall effect sensor measurement as well as PC fan tachometer measurement for validation

Physical Wind Tunnel: Fan Motor Control

- Remote control light dimmer switch
  - Simple: Attach stepper motor to existing light dimmer switch currently controlling motor.
  - Advanced: Use X10 or SmartHome network connected light dimmer switch to control motor.
- Issues:
  - X10/SmartHome light dimmer with enough electrical rating to handle motor
  - Stepper motor with Audrino connections and enough granularity to provide control loop feedback response

Physical Wind Tunnel: Temperature and Humidity

- Currently using IC based temperature sensors
- Currently using uC based humidity sensors
- Mounting to proto-board in appropriate places on tunnel surfaces

- May explore different humidity measurement technique to achieve better stability in readings.

- Currently using Audrino Ethernet shield to tweet temperature and humidity, and send to data logging service for long term storage
- Plans to implement internal web page for virtual wind tunnel look up of current tunnel conditions.

Future Plans

Physical Wind Tunnel

- More surfaces made from plexiglass / acrylic for strength and student / visitor visibility
- A "measurement" tunnel exhibit flow for older students / visitors
- Debug Tweets when system has issues or when executing normal startup/shutdown procedures
- Interfacing with the Virtual Wind Tunnel

Future Plans

Virtual Wind Tunnel

- Full integration of application inputs to 2D Flow Solver
  - Setting Tunnel Conditions within Application
  - Whiteboarding an airfoil shape
    - Image Checking to prevent malformed airfoils
- Visualization Work
  - Better examination of results from simulation
- Interfacing with the Physical Wind Tunnel

Future Plans

Non-Functional Requirements Coverage

- Resiliency to hardware/power failure
- Ease of use, Safety, Resiliency to use/misuse
  - Chris and Josh have graciously volunteered their children for testing purposes

Suggestions for Change?

Acknowledgments

- Dr. Jeffery Kulick

- Sean Smith

- Misty Marshall

- Jim Masters

Thank You!!