AirScope

Envision the air around you.

Marc Campasano

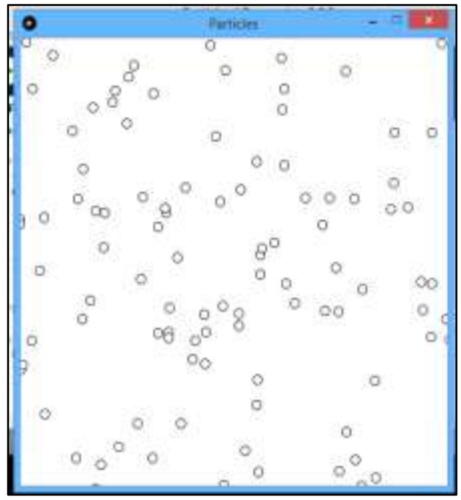
Beyond Bits and Atoms

Winter 2016

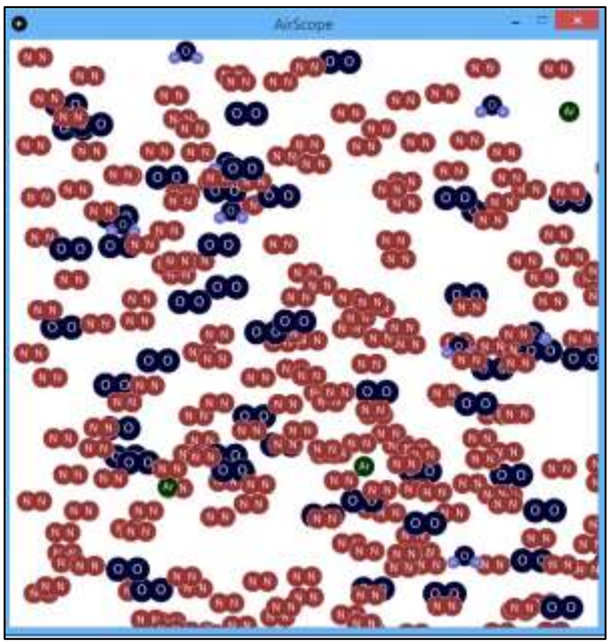
Final Project

Design Process Documentation

The earliest concept sketch of AirScope. The original idea was a "visual weather station" with far fewer features.

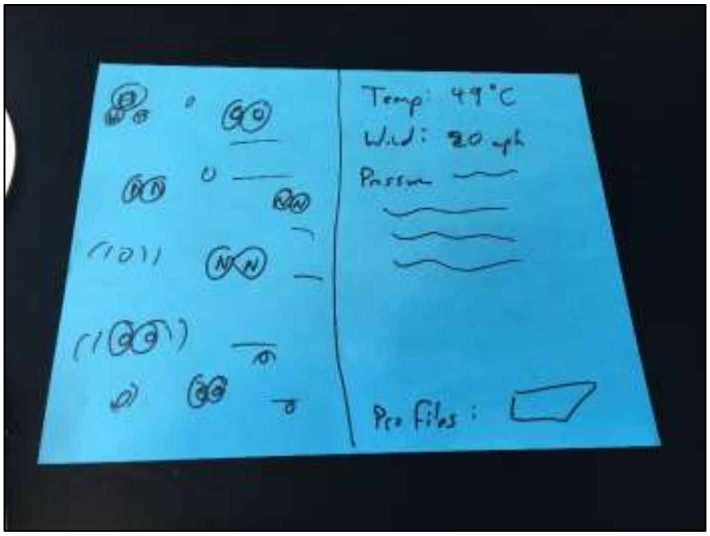


An early test to see if I could build a system of moving particles in Processing.

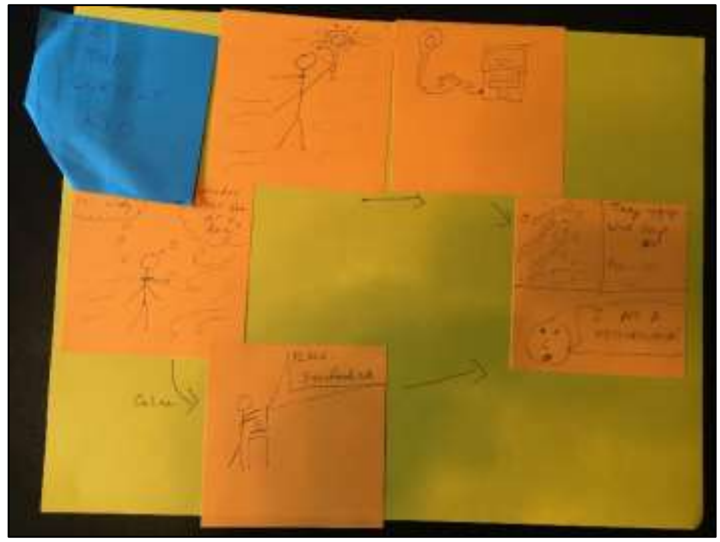


The visualization up and running. At this stage, the keyboard controlled the speed and vibration of the particles.

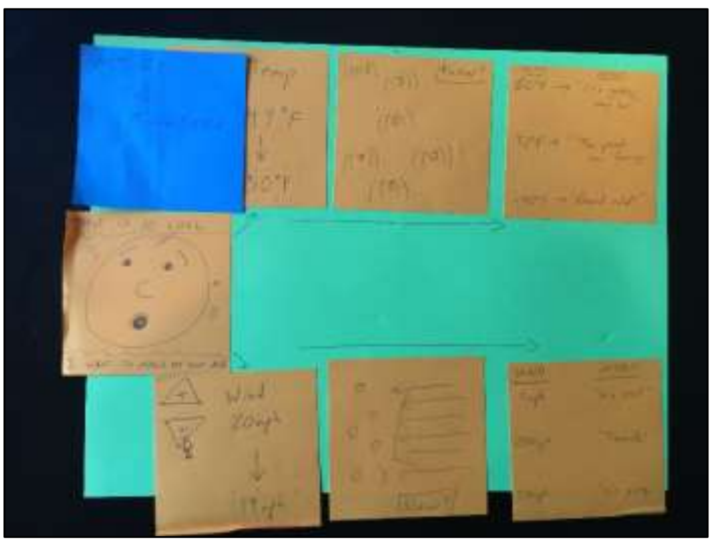
The first cardboard prototype of the lens. This didn't change much before the final wood version.



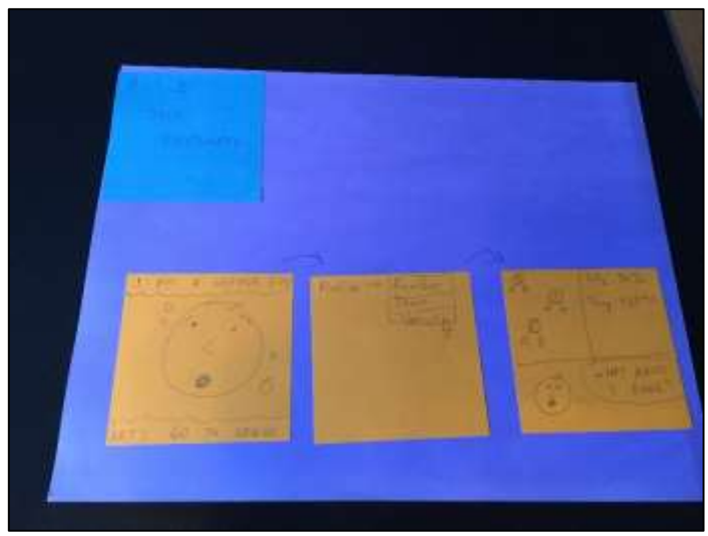
The interface sketch I showed in class.



Part 1 of the user storyboard I presented. This demos the Internet and lens data retrieval to explore local air.

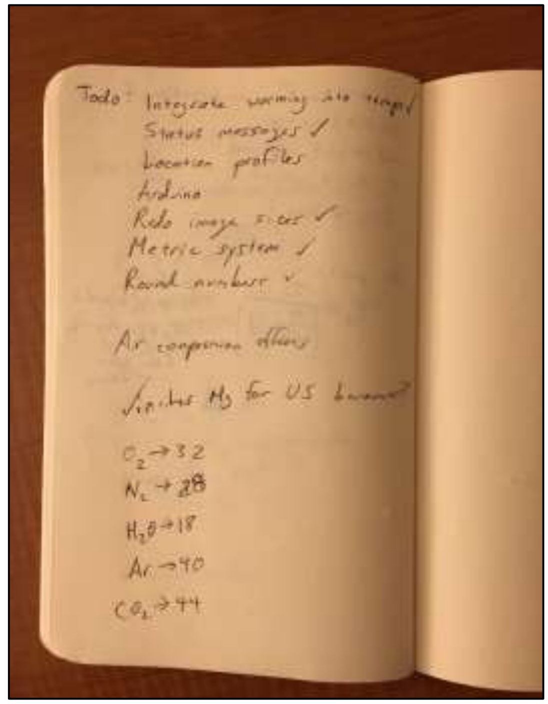


Part 2 of the user storyboard, which shows off the tinkering functions and macro-level descriptions.

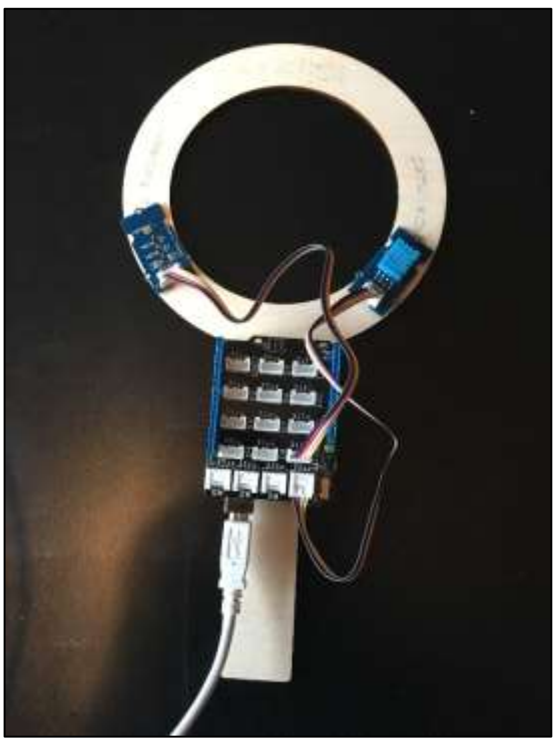


Part 3 of the storyboard, which shows the biome “air profiles” built into AirScope.

Notes on calculating the proportion of water molecules in the air from temperature, pressure, and relative humidity. On the right, plans for the macro-feedback descriptors.



A to-do list from a few days before expo. On the bottom, molecular weights for accurately scaling the size of molecules.



The final lens, with an Arduino, a barometer/thermometer sensor, and humidity sensor glued on to a laser-cut wooden piece.

AirScope in its final form: the program running using data fetched from the lens.

Knowledge Domain, Learning Goal and Audience

AirScope is a product for learning about the molecular composition and behavior of air and how they relate to macro-level conditions. Air is all around us and we, children included, observe and interact with it all the time. However, schools typically do not teach what air really is until later in children's education. The current science education track makes it difficult for young students to study air as a fluid because the language of molecules and particles comes late in their schooling. The behavior and composition of air are often taught in terms of chemical elements and ideal gas laws, which aren't studied until high school chemistry and physics.

Perhaps the reason students don't study air in-depth until late in childhood is because it is invisible. Since it can't be picked apart or directly seen by students, the advanced concepts and mathematics of high school science are the only popular tools for studying air. If air could be made visible, students could study it earlier in their education.

AirScope is an eye into the air around the student. It is a tool for late-elementary to middle-school students to explore the scientific roots of air and weather phenomena by presenting air data in a visual form that simulates the molecular level. AirScope can provide learners in the years before high school with an intuition of what air, temperature, humidity, and other air phenomena are without requiring them to know advanced heat, chemistry, and gas mechanics. The visual representation enables them to understand a general idea of what is happening to the air on hot, cold, windy, calm, damp, or dry days, and across climates. The tools for tweaking parameters and exploring the world contained within AirScope enable them to discover new facts about air and the planet.

A note about the limits of AirScope: I decided during the design process to focus on the chemical composition of air in a single place. Related topics I considered including, but rejected, included planet-wide atmospheric phenomena, larger particles in the air like dust and pollen, and a greater focus on weather. I decided to keep the scope narrow for reasons of complexity and accuracy. Studying and programming dynamics of the entire atmosphere would have expanded the academic content and scale of this project well beyond the "lens into your local air" original concept. A greater focus on weather, incorporating things like temperature fronts and precipitation, would have entailed similar system-wide thinking beyond the core concept (rain, for example, is more a result of regional pressure systems and sky-high water vapor than of very local air). I decided to cut pollen from the visualization after I realized that pollen particles are trillions of times larger than molecules. Ultimately, I think these exclusions led to a more focused, coherent product that is better equipped to teach its specific content area.

Literature Review and Theory

AirScope is first and foremost a *restructuration* of topics related to air composition, gas behaviors, and air chemistry. Wilensky (2010) defines the *structuration* of a set of knowledge as "the encoding of the knowledge in a domain as a function of the representational infrastructure used to express the knowledge" (p. 2). Traditionally, air-related topics have been taught via heavily symbolic structurations of math and science, such as math expressions and chemical symbols. Students must first understand algebra, the periodic table of the elements, the particle model of physics, and topics like kinetic energy before they can fully engage with the content when it is taught this way.

Airscope restructures air as a visual system, with the symbolic notation and mathematics backgrounded into the code. In chemistry class, students mainly engage with pressure as a variable to plug into equations, but AirScope displays pressure for what it *is*—the weight of the amount of gas over a spot. Speed appears in physics equations as distance over time; in AirScope, speed manifests as the speed of the molecules onscreen. Water molecules in AirScope appear as two tiny hydrogen atoms attached to an oxygen atom, a visual translation of the chemical formula H_2O. This restructuration makes AirScope's presentation of air more intuitive and accessible than the symbolic presentation offered in schools.

AirScope's visualization is particularly appropriate for engendering the "thinking in levels" described by Wilensky and Resnick (1999). AirScope is very similar to the "Gas in a Box" case of levels thinking in that paper (pp. 12-17), in that important scientific phenomena only manifest when many independent gas molecules are observed together. As in the real world, there is no such thing as a "molecule of air" in AirScope—there are only molecules of nitrogen gas, oxygen gas, water vapor, argon, and carbon dioxide. Air only emerges when these molecules come together as a mixture, and when the user observes them together onscreen, "one level up" from the individual particles. One molecule moving across the screen is just a molecule, but a thousand moving together become wind, a new "level" of phenomena that automatically emerges from their relationship. No single carbon dioxide molecule can teach about the greenhouse effect, and no single water molecule can teach about humidity. Only at the level of the whole air, of which these compounds are a small part, can the user appreciate the effects of varying the proportions of water vapor and carbon dioxide.

One of AirScope's major features, the "lens" that reads data from nearby air, leverages the *cultural form* of a magnifying glass to clarify its function and emphasize the core metaphor of AirScope. Horn (2013) describes cultural forms as "social constructions or conventions that are linked to recurrent patterns of activities" (p. 1). For tangible interaction learning products, items that invoke cultural forms can suggest how the product is used and what it is teaching. Magnifying glasses enlarge materials under study, and a magnifying glass is a cultural symbol of inquiry and investigation. AirScope leverages this function and symbolism to communicate that the lens is an investigatory tool for examining small things. When the user sends data from the lens to AirScope, the process simulates enlarging the air in the lens as if the lens really were a powerful magnifying glass. (When I demonstrated the lens at expo, a few users seemed to believe that the molecules onscreen were actually representations of specific molecules in the air!)

Product Review and Innovativeness

AirScope's focus on the molecular dynamics of local air, and their relationship to macro-level phenomena, sets it apart from similar products. There are plenty of kid-oriented weather station products, but AirScope is the only product I know of that simulates the molecular behavior of nearby air in response to weather variables, rather than merely reporting the data like a TV weather report. Weather- and air-oriented science simulations tend to focus on entire-atmosphere level dynamics, or narrow in on one item such as temperature or humidity. See, for example, the simulations at https://cimss.ssec.wisc.edu/wxfest/ or http://scied.ucar.edu/games-sims-weather-climate-atmosphere, or the "Weather Maker" at http://teacher.scholastic.com/activities/wwatch/sim/game.htm. Products that do focus on gas particles tend to separate their science from weather and other macro-level phenomena to focus on more traditional chemistry-class lessons. At http://interactives.ck12.org/plix/chemistry/, for example (which some cK-12 employees told be about at expo), there are many simulations of individual chemistry and physics topics that appear in AirScope, but none that present them all together in one environment.

In summary, I believe AirScope makes two major innovations beyond the competition. First, it marries the molecular science of air with macro-level phenomena the user can relate to. It is an intersection of "weather station" and "gas simulation" that better relates those topics to each other and the observable world to science class than a product that only focuses on weather or one that only teaches about gas. Second, AirScope is a constructivist environment that lets users explore many interrelated concepts, rather than a smaller simulation that deeply explores one aspect of chemistry, climate, or physics. Students make discoveries in AirScope by tinkering and observing, not because it is designed to lead them to a specific lesson couched in traditional science class structurations.

Design Process

My initial idea was a "visual weather station": a monitor that would show temperature, wind, and humidity data both as numbers and as a visual representation of particles in the air. As I explored the idea further and began planning it out, I realized that it would be better to think of the project as a visualization of air rather than of weather. The sensors I planned to use (initially GogoBoard sensors, but Richard advised that Grove sensors with Arduino would be better) would be best for describing local air, not the conditions of an entire region, and regional dynamics would be very complicated to simulate.

My first step toward making the product was to see if I could make a working system for a lot of moving particles in Processing. I am not a computer science student, though I have taken a few basic computer science classes and have dabbled in small programming projects. For the first prototype day in class, I had a demo that displayed bouncing circles whose speed I could change with the keyboard.

When I met with the teaching team, by then I had replaced the circles with a mixture of molecules and implemented keyboard-controlled temperature, wind speed, and wind direction. The teaching team liked my idea and responded with suggestions as to how I could add more exploration to the product. Professor Blikstein first suggested the idea that there should be updates reporting what was happening in the world on the macro-level, which ended up being a major feature of the final project. The teaching team also helped flesh out the “explore biomes” feature that loaded pre-constructed “air profiles” simulating faraway places. They also helped me figure out how to implement the carbon dioxide features. It was important to me that AirScope include a greenhouse effect simulation, but the visualization in AirScope was not suited to this because CO_2, important as it is, is only a tiny proportion of the gas in our air. I worried that visually displaying so few CO_2 molecules might mislead students to believe that CO_2 is not very important. I decided to report CO_2 concentration explicitly in the interface, with a foregrounded greenhouse effect to illustrate its influence on temperature.

My original proposal for AirScope said that the lens would be the only way to get local data, and that, if it ever became a real product, I’d implement fetching data from the Internet. After talking with the teaching team, I realized that grabbing Internet data was not as hard as I thought, and I actually had that feature up and running in a few hours of studying how to do it. I am using the Weather Underground API to grab ZIP-code based weather data. Implementing the Arduino sensors actually turned out to be much harder.

In the class storyboard feedback session, my critiquing group generally liked the idea, but communicated that my design was not intuitive enough. They only saw a mock-up of the interface, but they felt that they didn’t understand what it did without my explanation. In the final interface, I tried to ensure that everything was clearly labeled and that the button descriptions were as intuitive as possible. I was generally pleased with the reception that the interface got at the expo in that people generally seemed to know what to do once the core concept was explained to them.

The particle visualization was done fairly early in the process. The last week of work was mostly about implementing the control interface and working out the back-end math for controlling the text descriptors and emergent statistics like wind chill and heat index. These features, and the biome air profiles, entailed a bit of Internet research. (In particular, figuring out how to translate relative humidity into the proportion of water molecules in the air was more complicated than expected.)

The final feature implemented was the lens, which took some struggling with Arduino code before it would work. Thanks to Richard for all his help on this. The Arduino and sensors are hot-clued to a laser-cut “magnifying glass,” which takes a snapshot of the data at the time the “use lens” button is clicked.

Technical Construction and Functionality

How it works. AirScope is computer program written with Processing that visualizes the movement, composition, and behavior of air. The left half of the screen is a visualization of molecules moving through space. The right half is the controls. Data input via the controls determines the behavior of the molecules in the visualization. The data affects the visualization in the following ways:

1. Wind speed and direction from the incoming data determine the speed and direction of the molecules’ movement. Every molecule moves according to a single vector whose angle and magnitude are determined by the data.
2. Temperature data determines the strength of the vibration in the molecules’ movement. This is a visualization of the principle that temperature is related to the energy in a substance. Every step of the visualization, each particle is offset from the universal wind vector by a random amount. At hotter temperatures, this random offset can be larger, which makes the vibrations more severe.

3. Barometric pressure data determines the number molecules in the visualization. This represents the fact that barometric pressure is a function of the weight of air in the atmosphere over a point.
4. The humidity data determines the prevalence of water molecules in the visualization.
5. A data set determines the chemical composition of the air. This data is a set of probabilities that AirScope uses whenever it refreshes a particle to determine what molecule it will be. This is used in the carbon dioxide function as well as the planet Venus location. Most of the time, AirScope is set to average Earth air composition.

AirScope starts out using a default data set to generate the visualization. The user can change the data being used in four ways:

1. The user can use "+," "-," and "set" buttons to manually tweak the wind speed, wind direction, temperature, humidity, pressure, and CO_2 concentration. The "set" buttons let the user type in their own numbers.
2. The user can click on preset "air profiles" to see what the air would be like in exotic locations, using pre-populated data linked to those buttons. This prototype included the Sahara Desert, the Amazon rainforest, the South Pole, and planet Venus.
3. The user can click a button to enter a ZIP code to see what the air is like at that location. AirScope fetches temperature, wind, pressure, and humidity data from the Weather Underground API for this function.
4. The user can click a button to retrieve humidity, temperature, and pressure data from the AirScope "magnifying lens." The lens is a wand shaped like a magnifying glass to suggest that the user is magnifying the air around them, as if the lens were a powerful microscope.

The lens has sensors on its rim that connect to an Arduino, which outputs the readings to AirScope on the user's computer.

The control area of the screen displays all this data. It also displays statistics derived from this data including the heat index, wind chill, dew point, and greenhouse effect, plus text descriptions of what the weather is like based on the data. For example, high winds will trigger messages telling the user of the destruction that might happen, and the right conditions will trigger messages that there is frost, dew, and/or fog. There is also a toggle to switch between the U.S. and metric systems of units. I made the control panel with the ControlP5 library for Processing.

How to use it. While individual learners could study AirScope themselves, the product would be best in a science classroom. A teacher could direct their students to AirScope with a few guiding questions, such as "What is in the air?", "What happens when more CO_2 enters the atmosphere?", "What is humidity?", or "When does dew form?" Students can discover the answers to all of these questions by collecting data and tweaking values in AirScope. Their explorations would lead them to new discoveries and inspire them to invent experiments to do by tweaking the values or examining the air in different places.

For example, imagine that students are assigned the question "When does dew form?" They load up AirScope and decide out of curiosity to check the air in their ZIP code. AirScope tells them that it's a little windy, not very humid, and below freezing, but there's no dew indicator. One of the students knows that humidity is water, so they decide to see what happens when they raise the humidity. At 100% relative humidity, AirScope tells them that they've created frost. Another student knows that frost is ice, so they raise the temperature to see if the ice becomes water. At 33 degrees Fahrenheit, voila! AirScope tells them there's dew. Using a little prior knowledge and

AirScope's feedback, they've discovered themselves that dew forms at 100% relative humidity when the temperature is above freezing.

There are other potential lessons with AirScope. A science class could observe what the weather does to the air over the course of months and draw inferences about how weather phenomena manifest on the molecular level. Combined with observations from outside AirScope, they might see if there are ways to predict from the air when it is likely to rain, or try to determine what causes wind. Or they could see what the air is like in different local places—they would find, for example, that the air in a bathroom after a shower and the air in a refrigerator are quite different.

Works Cited

Horn, M. S. (2013). The role of cultural forms in tangible interaction design. *TEI 2013*. Barcelona.

Wilensky, U. (2010). Restructurations: Reformulating knowledge disciplines through new representational forms. *Constructionism 2010*. Paris.

Wilensky, U. & Resnick, M. (1999). Thinking in levels: A dynamic systems approach to making sense of the world. *Journal of Science Education and Technology*, *8*(1).