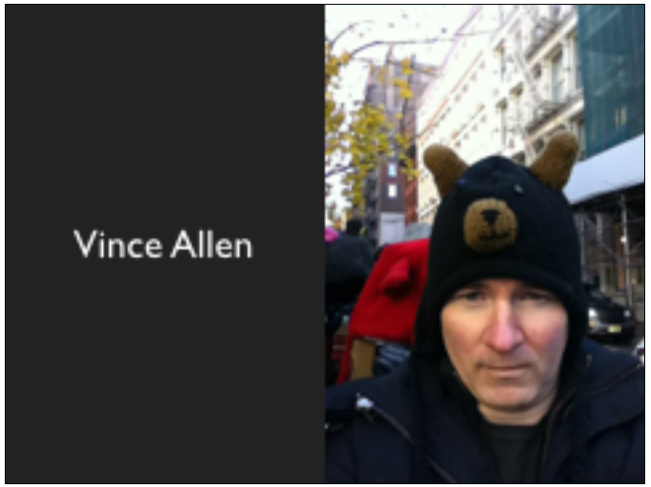
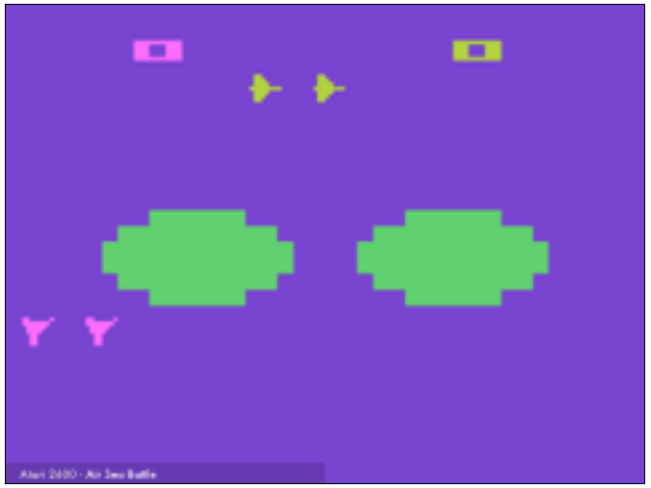
The title of this talk is “Pixel Art and Complex Systems”. We’re going to talk about pixel art and how to render and animate pixel art in a web browser. We’re also going to talk about abstraction and how it can help us better understand complex systems.



I’m Vince Allen, software engineering manager at Spotify. You can reach me @vinceallenvince on Twitter and find code for all the demos at github.com/vinceallenvince.

Cheers to Atari for addressing gender issues in gaming early on. I was 9 or 10 years-old when I got an Atari 2600. It was my first glimpse of pixel art.

http://www.youtube.com/watch?v=c_P4i9Etj3Q



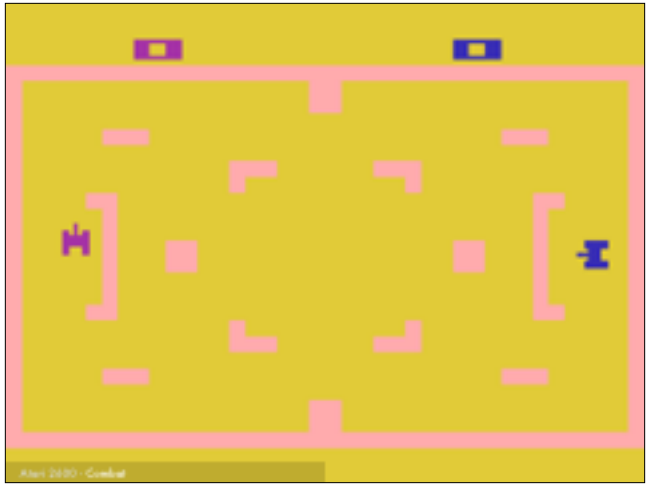
I easily spent hundreds if not thousands of hours on that machine.

These are the types of images I was interacting with.

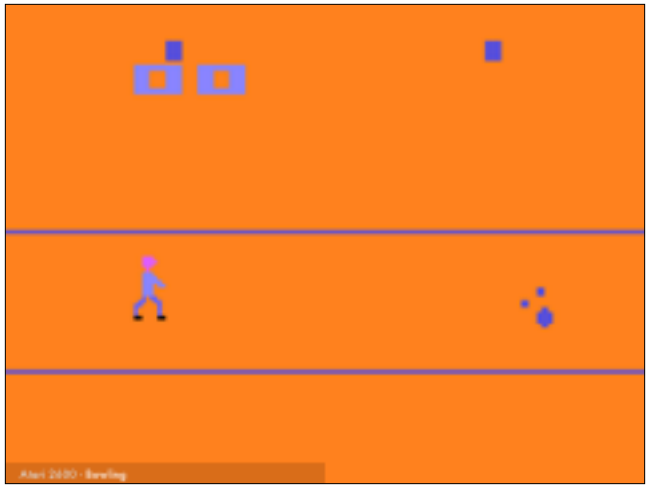
They were simple.

They were an abstraction of the real world and certainly the fantastic worlds depicted on the cartridge's packaging.

But given their simplicity, these images framed a world where I escaped and explored day after day.



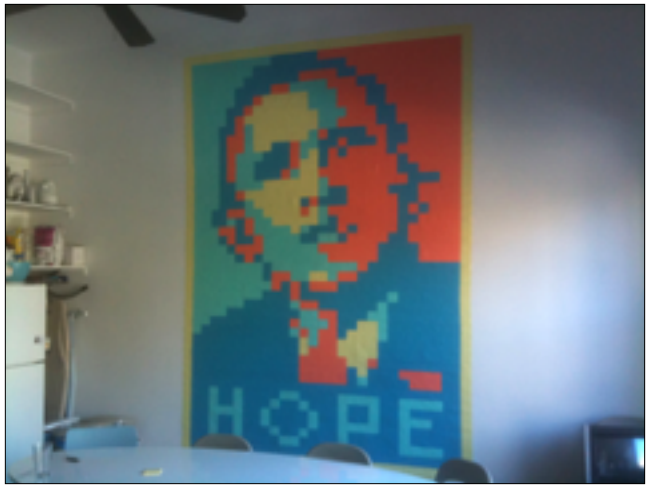
Compared to a modern console, the technical capabilities of the 2600 were limited. But I still remember the first time I discovered a glitch in Atari's Combat allowing me to drive my tank past the walls defining the game space. I ended up lost in an empty void.



Whether intended or not, this bug reinforced my impression that the game world was limitless. And like the tragic hero of Atari's "Bowling", I stood at the center of a vast universe with no audience to cheer my accomplishments.

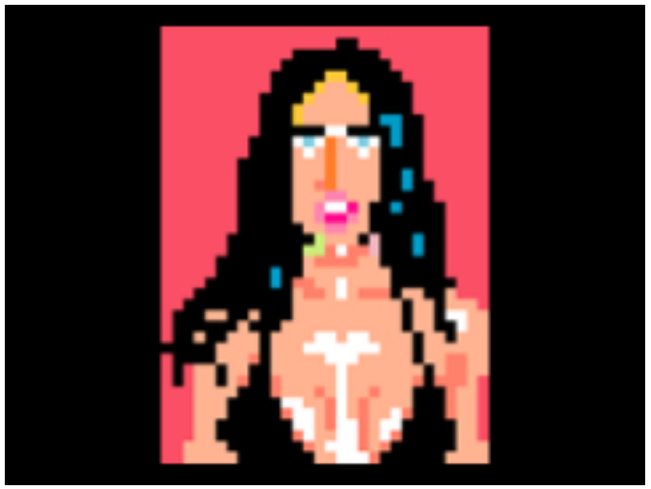
Fast forward 20 years… soon after President Obama's 2008 campaign kicked off, I moved into an open-plan Brooklyn apartment with high ceilings and big, blank walls. After seeing the Shepard Fairy poster that dominated the campaign visuals, I thought, "I want that on my wall."

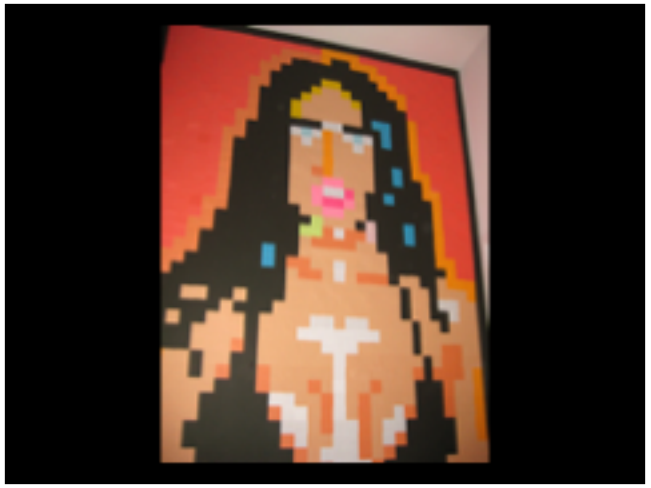
Fairey was sold out of them. And I wanted a larger format anyway. So I wrote a script that converted the colors in the poster the available colors in the Post-it Note color palette.



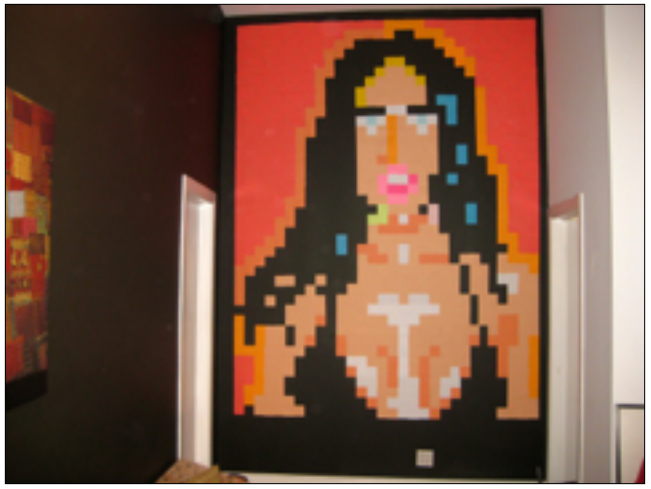
And here's the completed work in my kitchen. It's 30 Post-its wide X 46 Post-its high which is 7.5 feet by 10 feet. That's a total of 1380 Post-its. I think it took 6 - 8 hours to complete.

Do you remember this GQ cover?



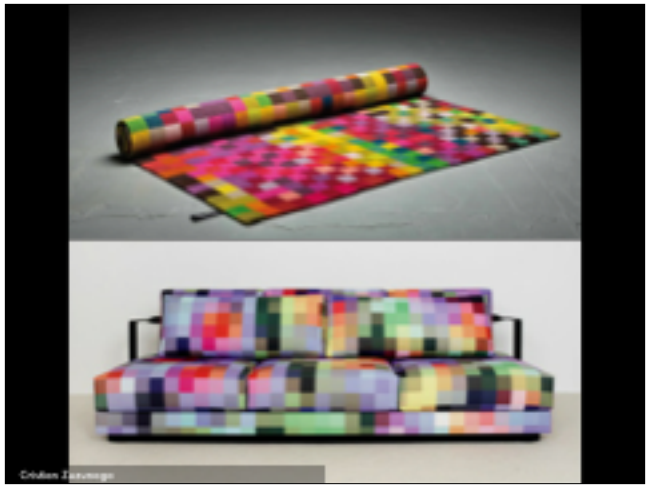


This is a 10-foot high Megan Fox made entirely out of Post-its.



My roommates were getting worried about the pattern that was starting to emerge. But I was having fun. This was the first time I thought about the **art** in pixel art. My experience with the style was limited to virtual world. Meanwhile, off-screen pixel artists had been creating work for years for…

… the street…



… our interiors…

… and as fine art. I went to the Dumbo Arts Festival last month and saw a lot of great artwork. The Jumbo DUMBO Puppy was one of my favorite pieces.

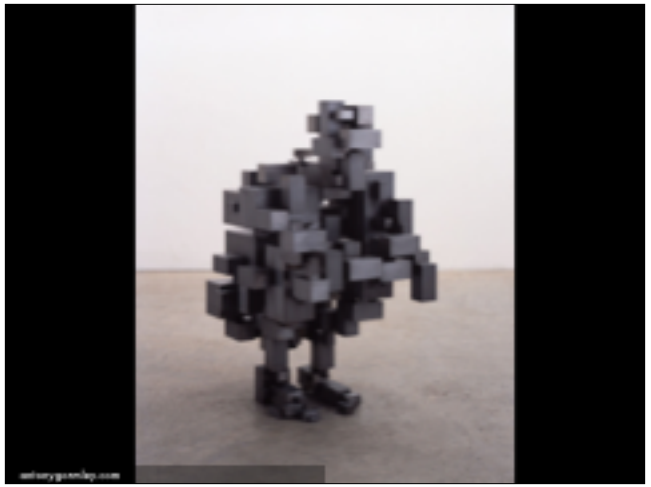
This is another puppy further down the street. I didn't find an artist statement. But something interesting happens when you manipulate scale like this and apply a physical abstraction we normally associate with pixel art.

I remember when I first saw Antony Gormley's work. He's a British sculptor who has installed work all over the world.

Walking through the Flatiron district, I spied one of his figures from 'Event Horizon'.

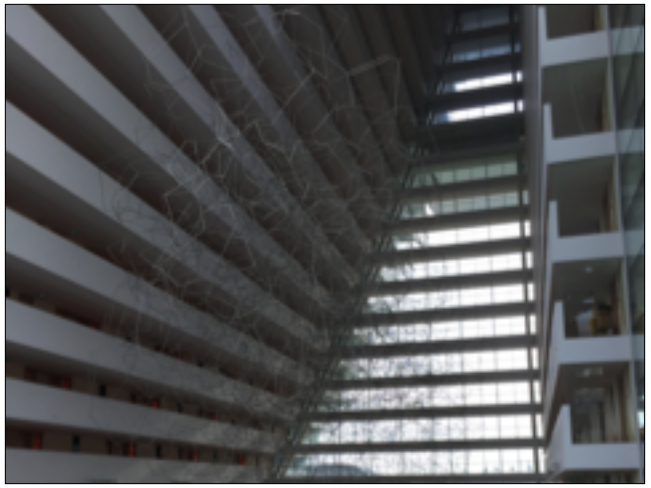
He placed 31 life-size human figures on top of buildings and in parks around Manhattan.

Looking at more of Gormley's work, I found sculptures that presented the human form in way that immediately reminded me of pixel art.



These are extreme abstractions…

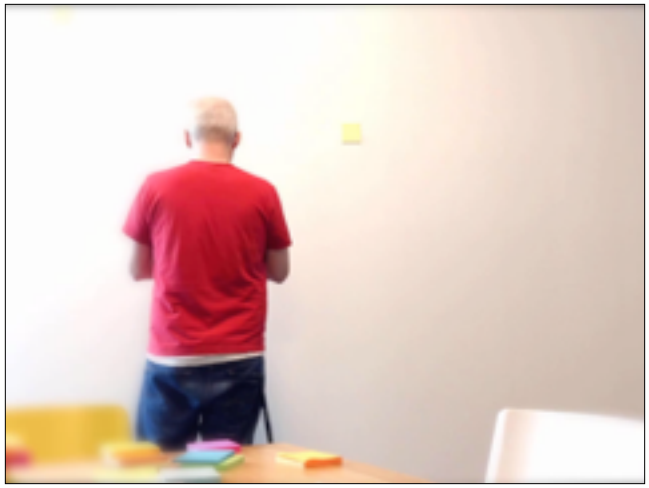
… that use the cubic form as a metaphor for the body.



You can find Gormley's sculptures in Singapore. This is a piece called DRIFT hanging in Tower 1 of the Marina Bay Sands.

A series that holds particular resonance for me is Allotment. It was first installed in Malmo, Sweden where Gormley asked 300 volunteers to offer their bodily measurements. He then cast their form as two-piece, hollow, concrete cases with holes representing eyes, ears, nose, etc.

Gormely's work focuses on the juxtaposition of the inner and outer self. Like these pieces, the inside of his sculptures are often hollow suggesting an infinity within. It's up to the observer to fill in what's missing. This exchange commits the viewer to empathize with the artist and invest in "completing" the work.

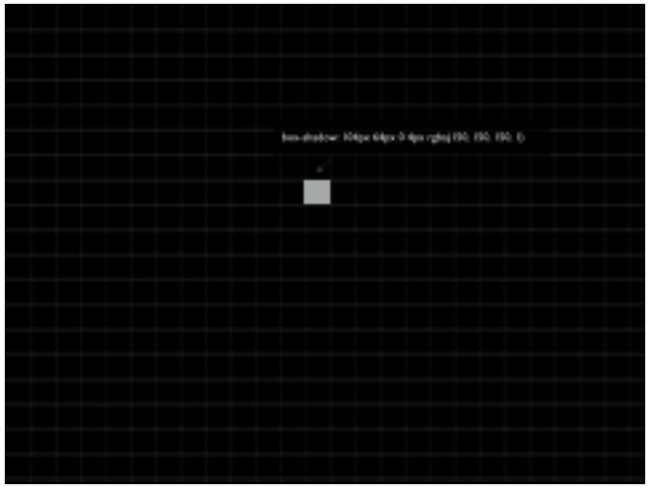


Considering pixel art in a broader sense, these artists are highly influential and inspiring. As a JS developer, I've been thinking a lot about how to create pixel art and abstract systems in a web browser. Sticking Post-its on a wall is a great lesson. Post-its are just big pixels with an x, y location, a color and a scale. 4 properties. Simple. Surely a web browser can handle it.

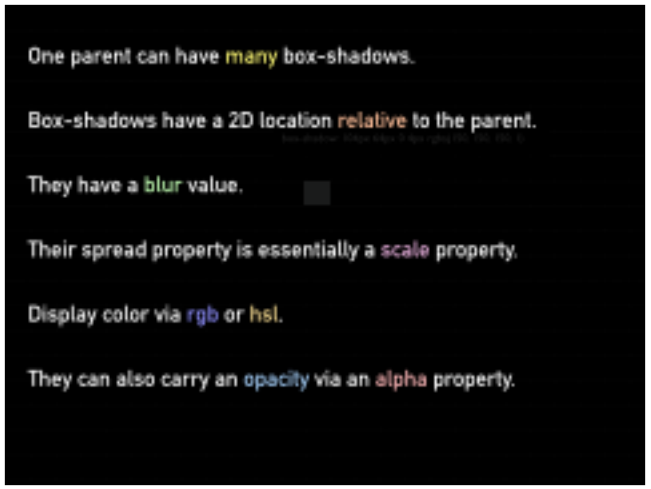
I've created a few approaches but landed on one I call the Bit-Shadow Machine. It uses the web browser's true Post-it Note equivalent, the CSS box-shadow. Let me explain.

We divide the visible browser into a grid and place a single div in the top left corner.

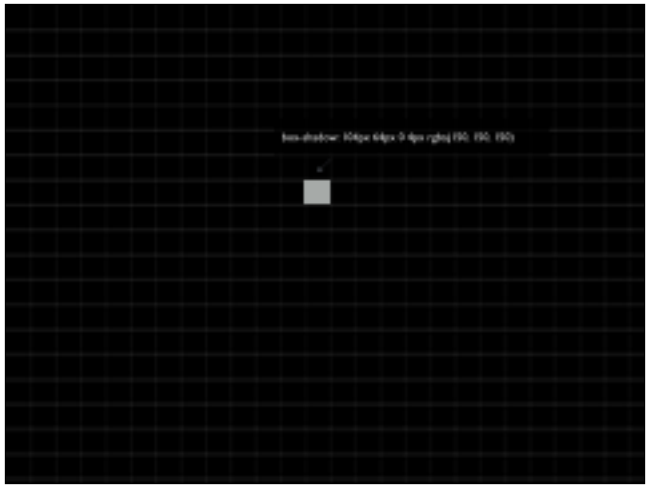
Next, we hide the div by assigning a height and width of 0.



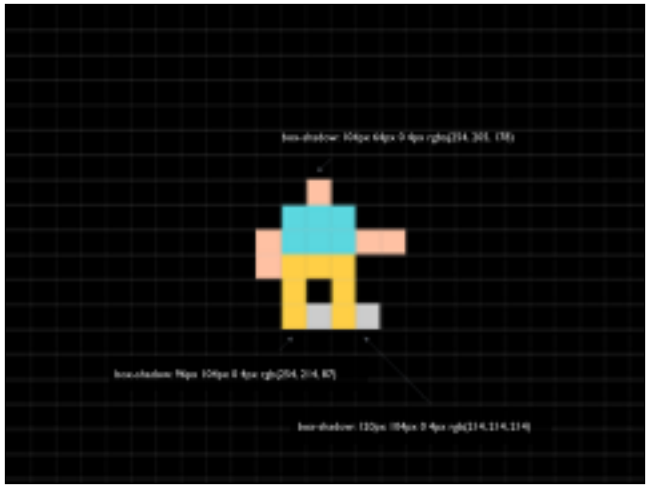
An interesting thing about box-shadows, if we hide the parent like this, we can still see its box-shadows.



Other qualities.. one parent can have many box-shadows. Box-shadows have a 2D location relative to the parent. They have a blur value. Their spread property is essentially a scale property. They display color via rgb or hsl. They can also carry an opacity via an alpha property.



Remember we only need 4 properties to render pixel art. Box-shadows give us those properties and more.



This means it's only a matter of adding more box-shadows and adjusting their properties to get what we want.

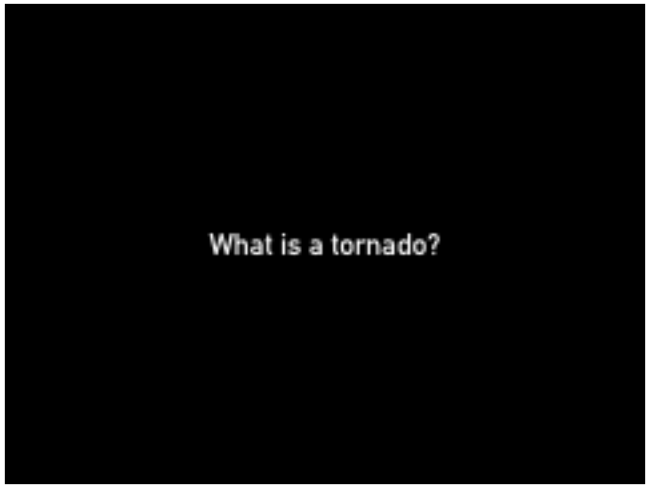
For examples visit:
http://vinceallenvince.github.io/JSAsia2014/

We can render simple simulations and frame by frame animations. But what about rendering something more complex… like a tornado.

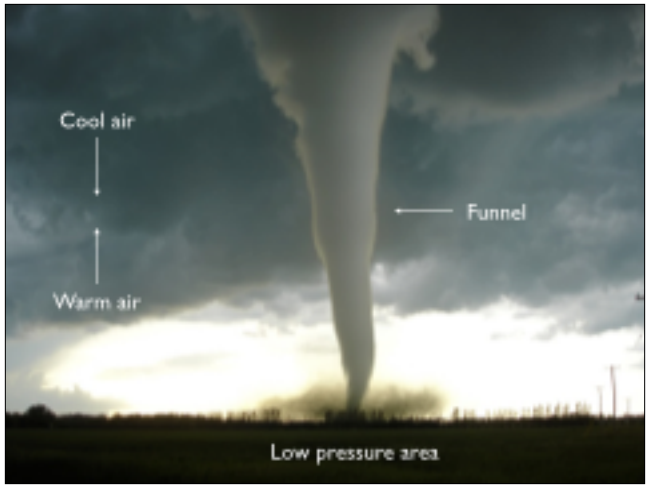
I've always been fascinated with tornadoes and how they work. Maybe it's because I grew up in an area that gets them frequently.

They are big and scary. But miniaturization has a strange effect. When we take something enormous and violent and shrink it down, it becomes cute. So that's what we're going to do, now… use JavaScript, CSS and the DOM and create tiny tornadoes in a web browser.

Here's a real miniature tornado, a dust devil which is a vertical column of rotating air.

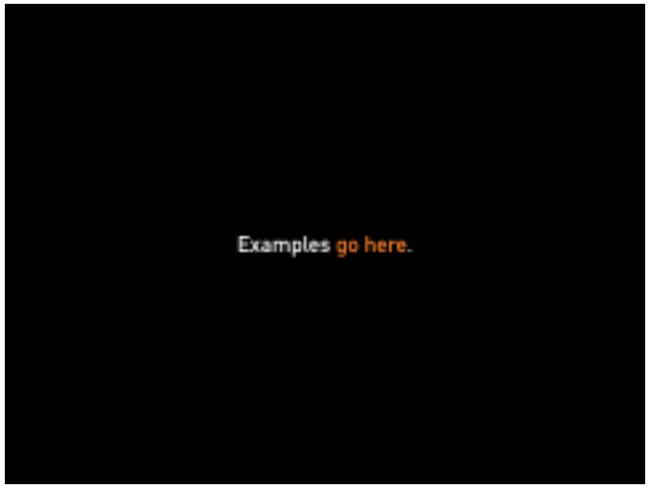


Similar to the Dust Devil, a tornado is a column of spinning air. However, the tornado's distinctive feature is that it's attached to a supercell which is a massive rotating thunderstorm.



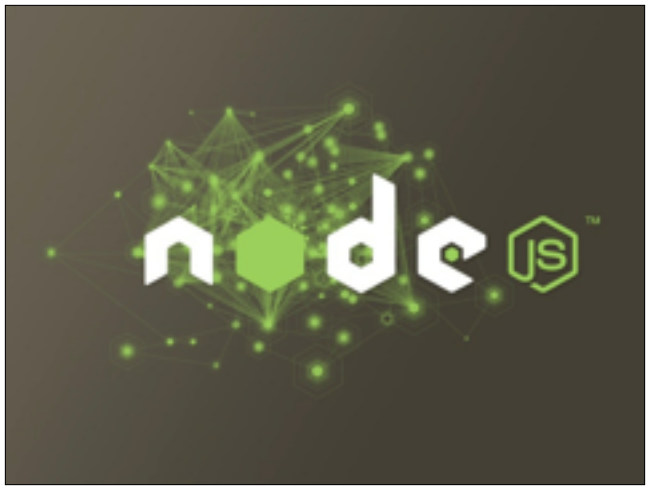
Tornadoes form when cold air from the Supercell accelerates down and meets warmer air. It creates an updraft and an area of low pressure which forms a condensation funnel as warmer air rushes upward. When the funnel reaches the ground, we have a tornado.

To render it in a browser, we need to identify its core visual components. The base is the point where the tornado touches the ground. The debris cloud is a cloud of dirt, cars and stuff that the tornado is kicking up. The spine is the tornado's armature. It defines the shape of the funnel. The shell is the wrap of clouds and condensation rotating around the spine.

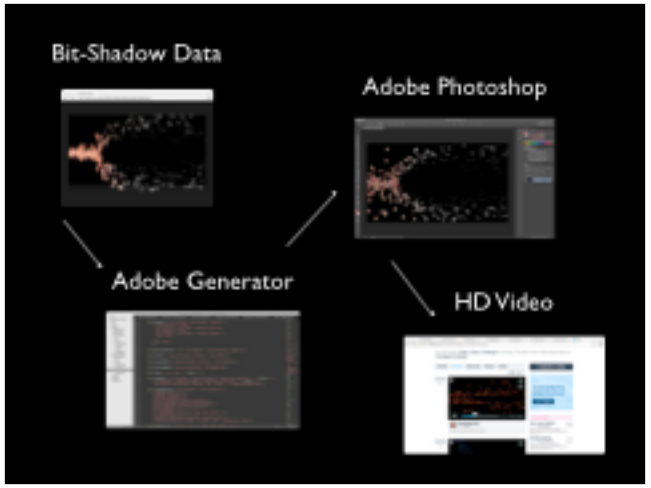


For examples visit:
http://vinceallenvince.github.io/JSAsia2014/

As I created more of these simulations, I wanted to move out of the browser and render them as video. It turns out there's a JavaScript based solution using Photoshop…



… and NodeJS.

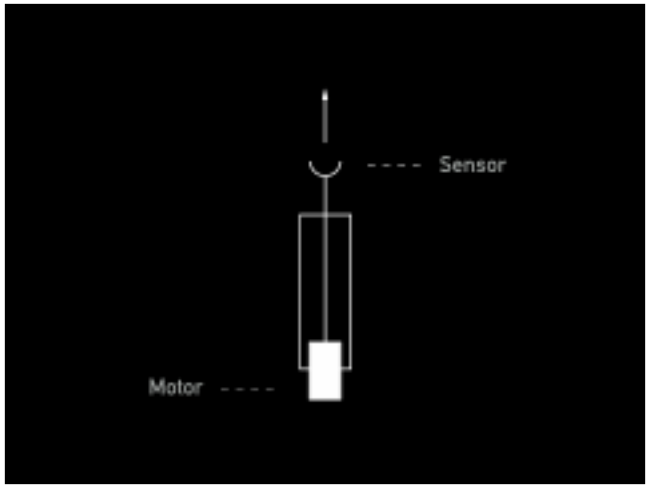


With the 14.1 release of Photoshop, you get Adobe Generator which allows you to automate tasks in Photoshop using Node. Generator allows you to create Photoshop plugins which means we can write Photoshop plug-ins using JavaScript. I created a plug-in that renders each frame of these simulations and outputs still frames I eventually compile into video.

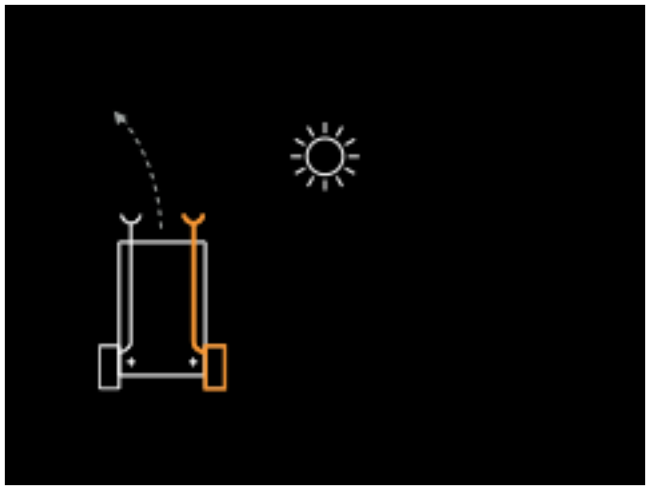
While the high resolution version is pretty, when we pixelate it, we see the key aspect of a tornado system that defines its iconic look. The conservation of angular momentum means mass closer to the rotational axis will rotate faster. As the air warms, it loses energy, moves away from the axis and dissipates. That's how we get the classic, upside-down pear shape. We're going to be using this video technique with the rest of our simulations.

Valentino Braitenberg… we're going to use a method he developed to investigate some more complex systems. He was an Italian neuroscientist interested in studying how human beings developed reasoning and intelligence.

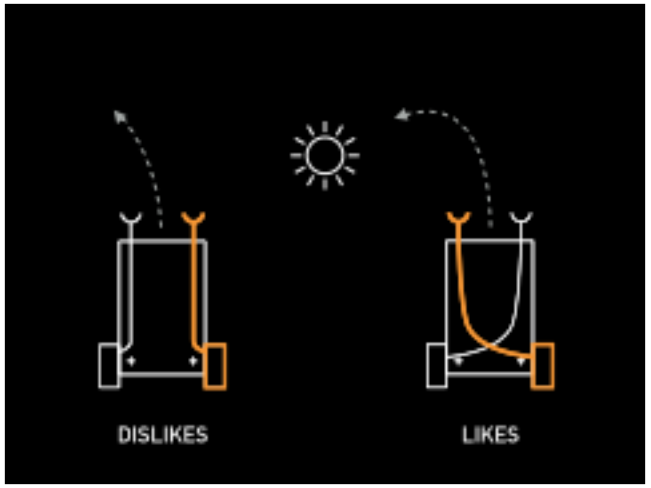
Instead of looking at humans, he started with insects. By examining simple brains, he hoped he would find the building blocks to our cognitive evolution. He also carefully considered his methodology. He focused on the mechanics and inner workings of simple systems as a path toward answering larger questions. By understanding how our brains evolved, he hoped to better understand how our brains work.



Along the way he invented Braitenberg Vehicles. Each vehicle is equipped with sensors tuned to specific input. For example, the first vehicle has one sensor wired to a motor. The sensor is tuned to temperature and activates the motor in proportion to the temperature it senses. The effect… the vehicle slows down around cold things and speeds up around hot things.



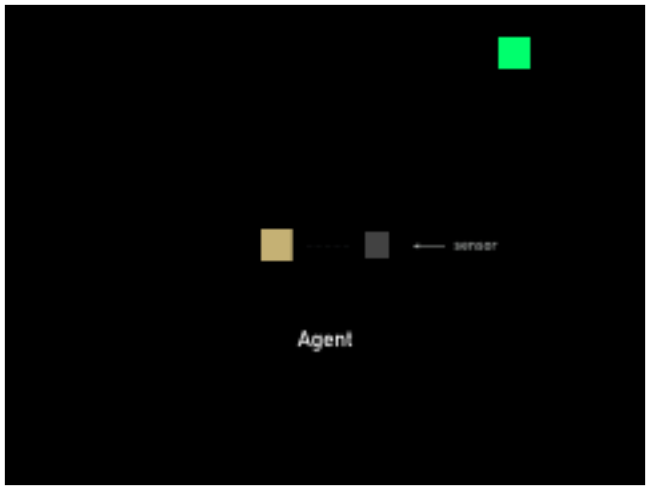
The vehicles progress with increasingly sophisticated sensors and wiring. For example, Vehicle 2a has two sensors wired to two motors. The sensors are activated by light. The right-side sensor is closer to the light source and delivers a stronger input to the right-side motor. The result, the vehicle steers away from light.



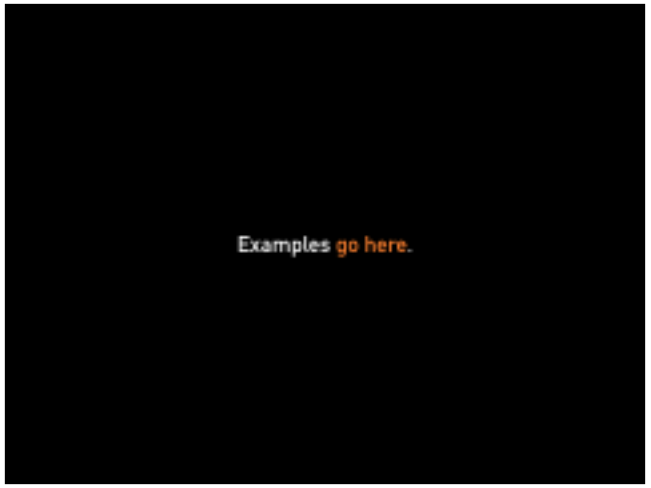
Vehicle 2b has two sensors wired to two motors on opposite sides. When exposed to light, the left-side sensor drives the right-side motor more than the left-side motor. The result, the vehicle steers toward the light. As observers, we would say vehicle 2a DISLIKES light while vehicle 2b LIKES it

---

As personal robotics becomes more accessible to makers, more people have been experimenting with these types of vehicles. I visited the Art Science Museum a few days ago and saw these robots at the end of the da Vinci exhibit. They operate under the same basic principles Braitenberg describes.



We’re going to build some systems using simple autonomous vehicles equipped with sensors wired to seek out specific stimuli.

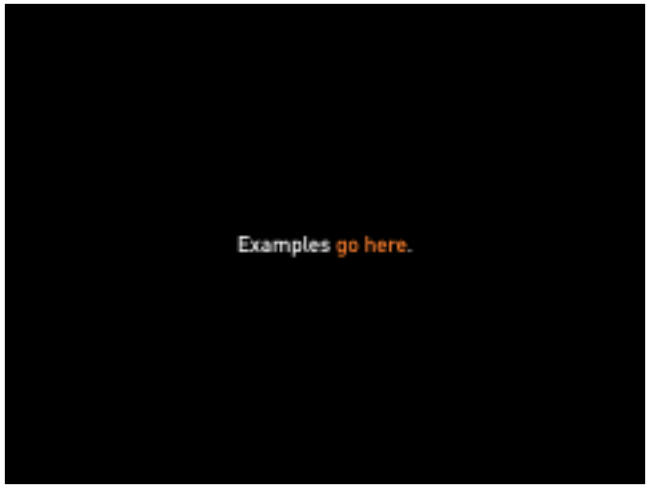


For examples visit:

http://vinceallenvince.github.io/JSAsia2014/

It's gets more interesting when agents are tuned to seek out each other. We're going to observe a simulation called Sheep vs. Wolves. Both have sensors. But the wolves pursue sheep while sheep avoid wolves.

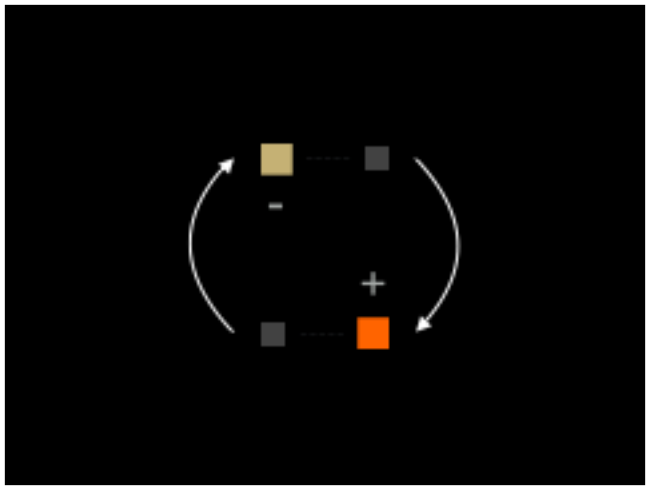
If a wolf happens to catch a sheep, it turns the sheep into a wolf.



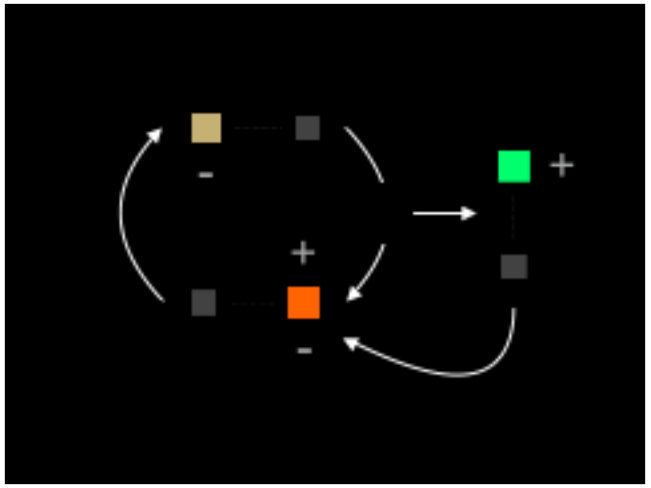
For examples visit:

http://vinceallenvince.github.io/JSAsia2014/

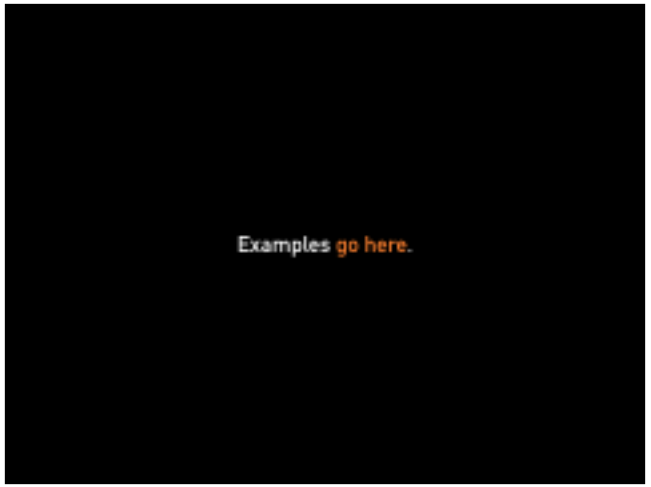
http://vimeo.com/groups/221221



This is an example of a positive feedback loop. Think about a herd of cows and a stampede. A change in the system causes an exponential increase in the same type of change.

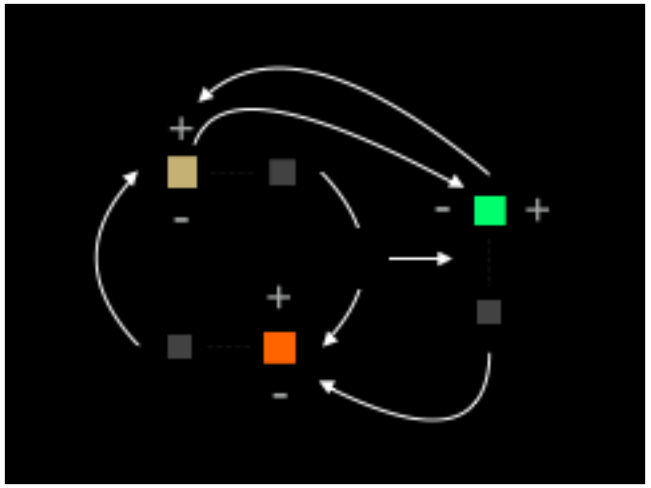


To work toward a more balanced system, we need to introduce some negative feedback. In the Sheep vs. Wolves simulation, we're going to add zombies. When a wolf catches a sheep, there's a small chance the wolf will create a zombie instead of a sheep. Zombies chase wolves and turn them into sheep.

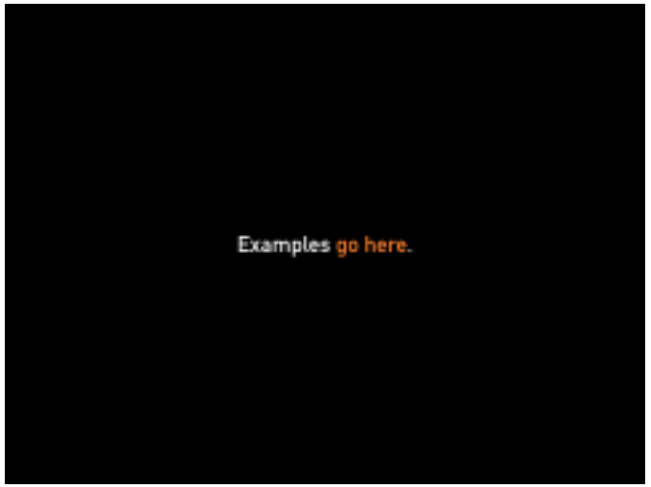


For examples visit:
http://vinceallenvince.github.io/JSAsia2014/

http://vimeo.com/groups/221221



To reach an equilibrium, we need more negative feedback for the zombies. We'll add a sensor to the sheep so they pursue zombies. If they catch one, they turn it into a sheep.

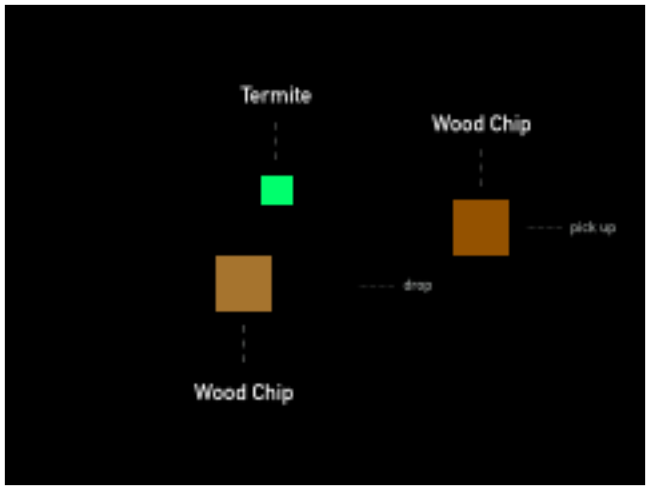


For examples visit:
http://vinceallenvince.github.io/JSAsia2014/

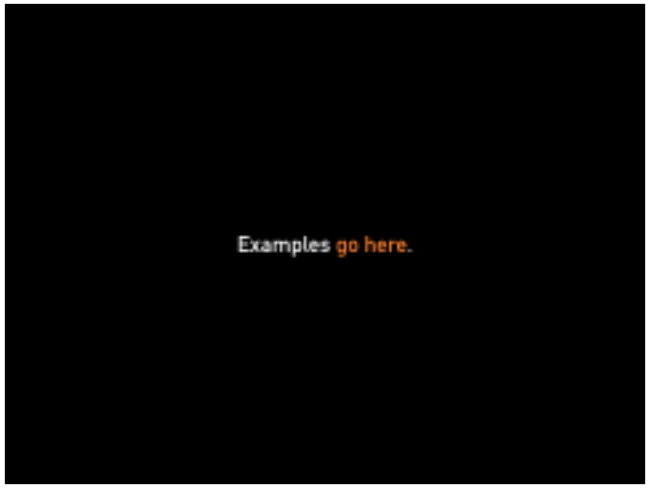
http://vimeo.com/groups/221221

Termites are incredible builders.

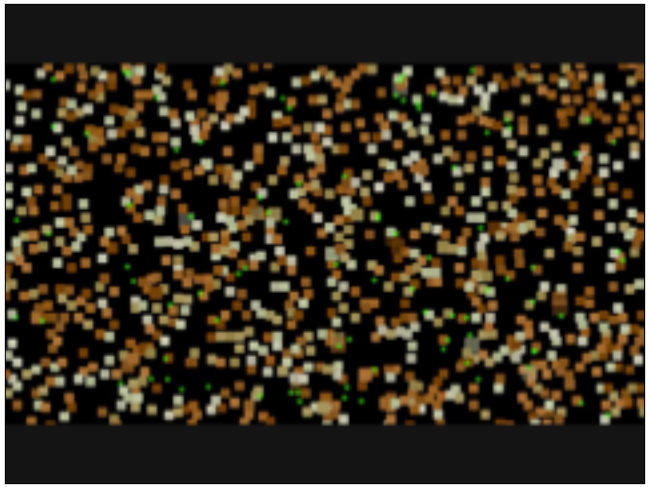
These are some termite mounds in Australia that suggest termites have a natural sense of architecture. We know they have very simple brains only capable of carrying out a simple set of instructions. How are they able to create highly complex structures.



Michael Resnick explores this paradox in his book “Turtles, Termites, and Traffic Jams”. He created a simulation involving simulated termites and wood chips. The termites randomly walk around. When the encounter a wood chip, they pick it up. When the hit another chip, they drop what they are carrying and continue to wander.

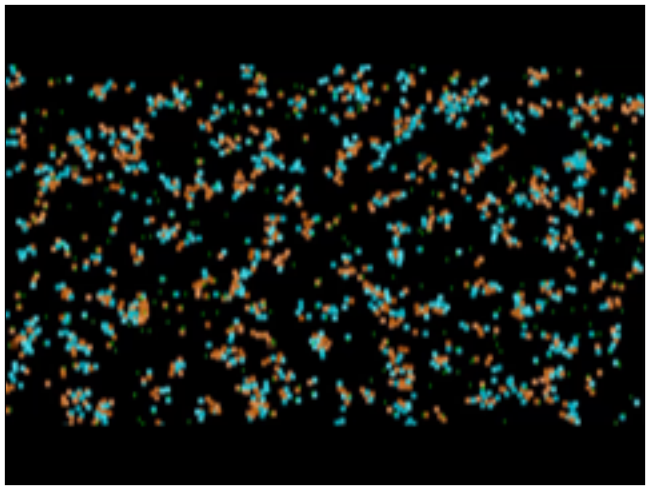


For examples visit:
http://vinceallenvince.github.io/JSAsia2014/

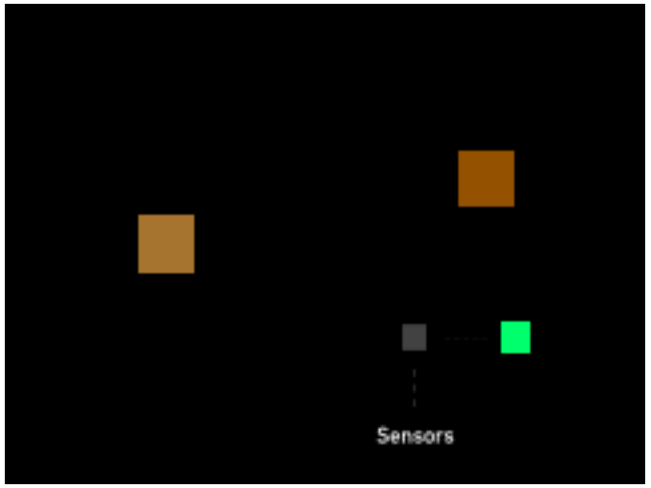


It's not hard to understand what's happening. The larger a pile becomes, the greater probability it will acquire more wood chips. It's another example of a positive feedback loop.

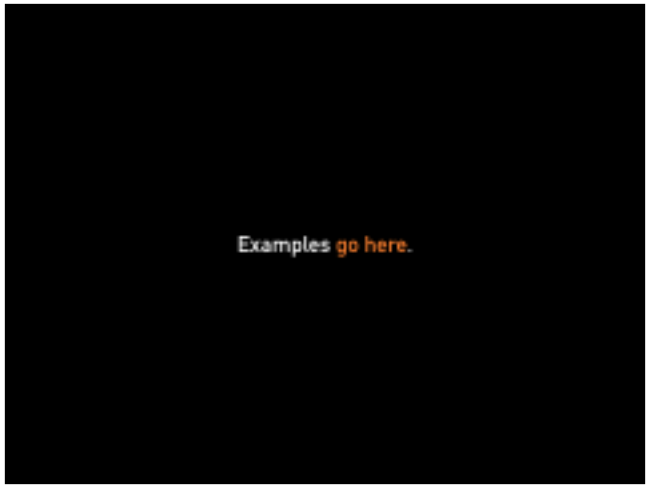
http://vimeo.com/groups/221221



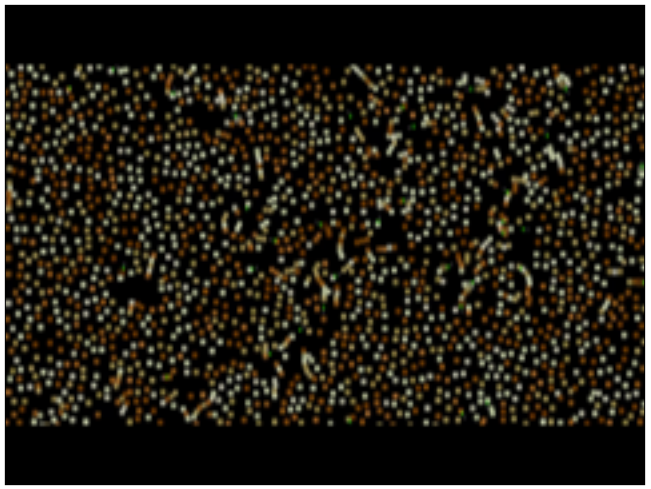
I decided to lower the termite to wood chip ratio, increase the total number of wood chips as well as the size of the environment. This simulation uses 3500 wood chips. It took the termites four days to organize them into one pile.



Four days is a long time to wait for a termite pile. What can we do to make the termites more efficient. Like the sheep and wolves, we can add sensors. In this next simulation, I gave the termites a sensor that activates if a wood chip is within a five-pixel radius.

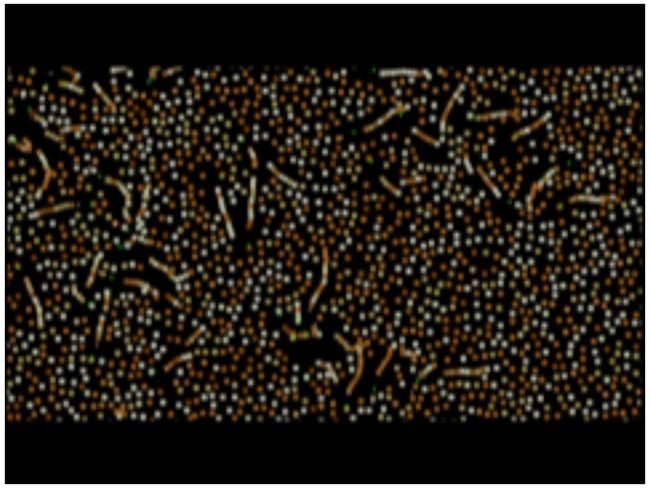


For examples visit:
http://vinceallenvince.github.io/JSAsia2014/



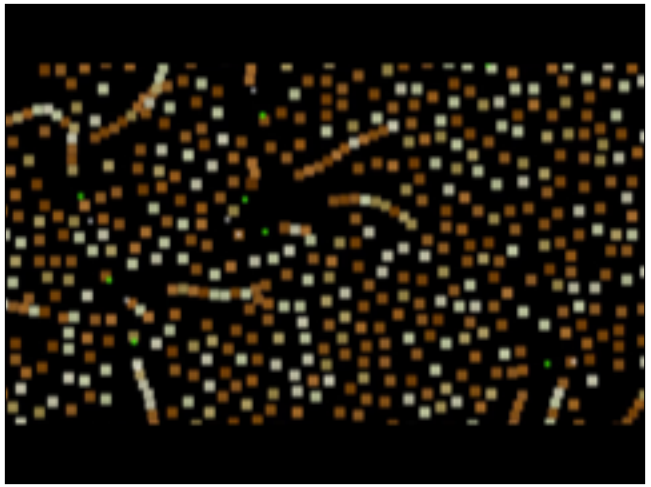
By slightly enhancing the termite's ability to navigate its environment, we dramatically change the structures they create and the level of design they exhibit.

http://vimeo.com/groups/221221

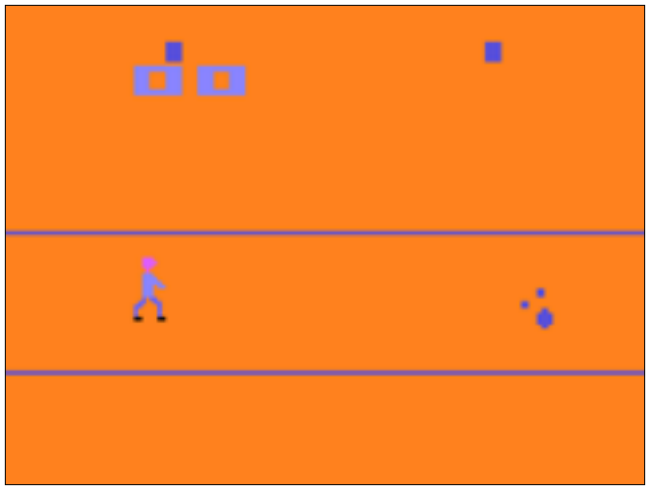


In this simulation, we increase the sensor sensitivity to 100 pixels and get a pattern that looks like a system of highways. It's not hard to understand what's happening. As soon as they turn around, their sensor is tuned to the nearest wood chip… which is often the one directly behind them.

While I'm using simple shapes in these simulations, I'm not suggesting this simulation is pixel art.



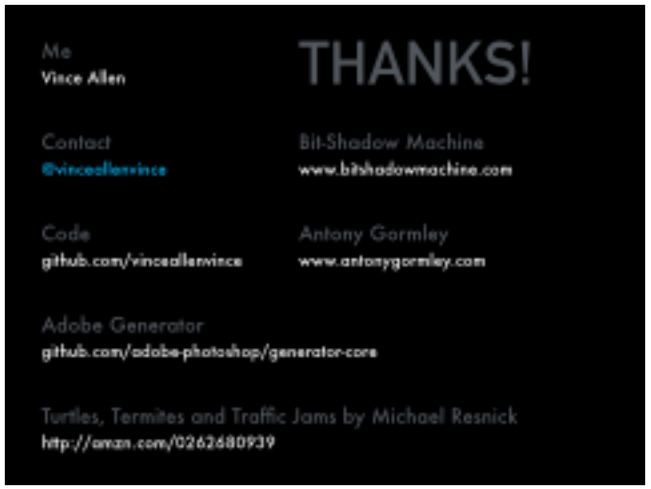
The big pixels symbolize the role abstraction plays in our investigation into complexity. In a real termite system there are thousands of variables in play. In our simulation, we reduced the system to its essential components. Only then can we understand how positive feedback and massive parallelism can create global order in a system governed only by local rules.



As observers, we're a lot like the heroic bowler trapped inside a simple but limitless world.

In the Sheep vs. Wolves simulations, the agents could represent anything. It was a conscious action on my part to call them sheep or call them wolves. If I had not labeled them, you would have created your own interpretation.

When we look at Antony Gormley's work, the artist's intent hinges on us filling in the gaps with what we carry inside us.



That's what I love about pixel art. It offers us an opportunity to complete the other side of an artist's creative impulse. It's a gift we receive when crossing the boundary from a high resolution exterior world into the boundless, interpretive freedom of our imagination.