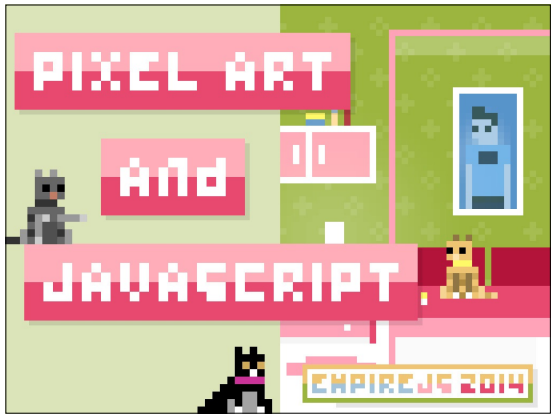
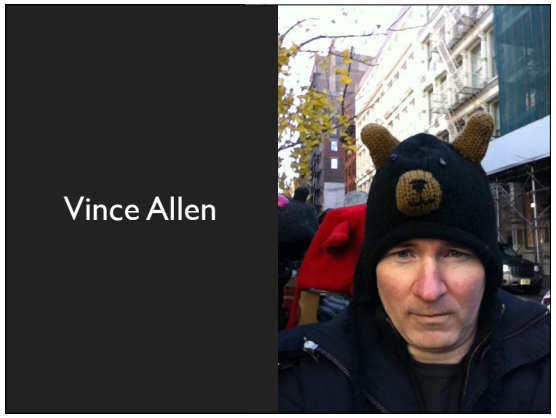
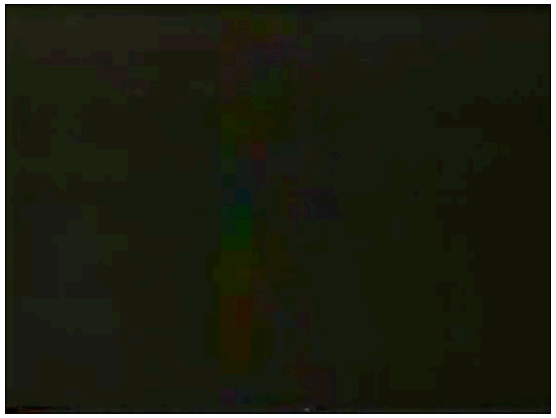


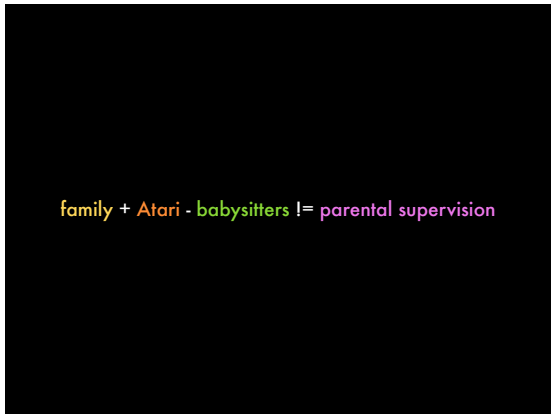
The title of this presentation is Pixel Art and JavaScript. We'll talk about pixel art. But we'll also talk about how to render pixel art in a web browser and output it for playback as video.



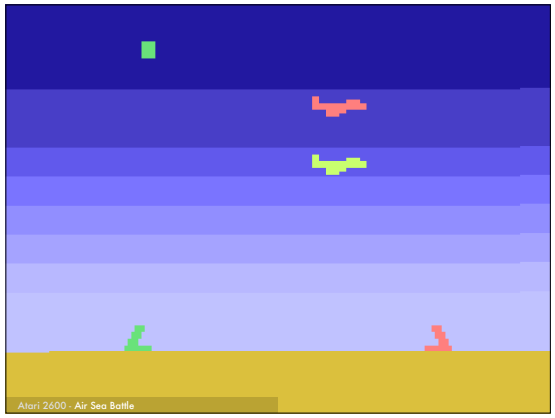
I'm Vince Allen, software engineering manager at Spotify. You can reach me @vinceallenvince on Twitter and find code for all the demos at github.com/vinceallenvince.



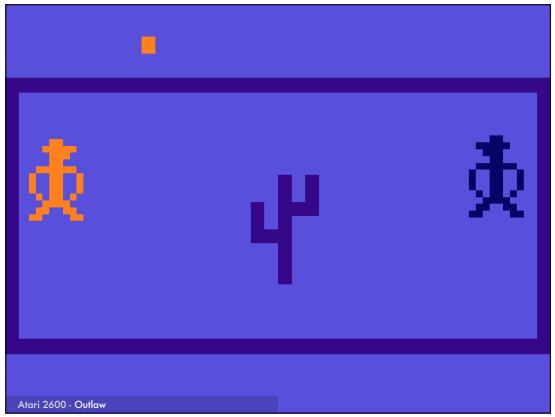
[Announcer] "This commercial is based on a true story."
[Mom] "Hello? Tracy, no, we don't need a babysitter tonight. Thanks anyway."
[Announcer] "After a family bought an Atari video game, they had no problem getting babysitters."



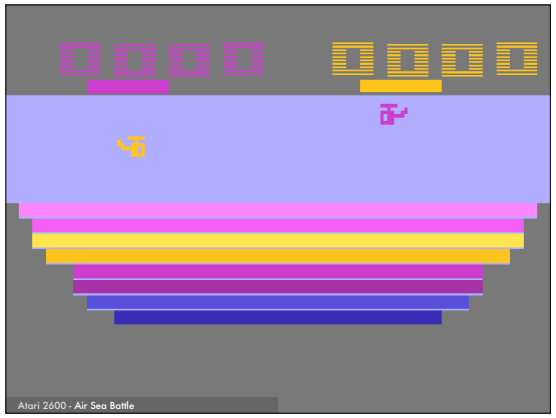
Despite what this TV commercial implies, playing Atari did not lead to me spending more time with my parents. But it did provide my first access to pixel art.



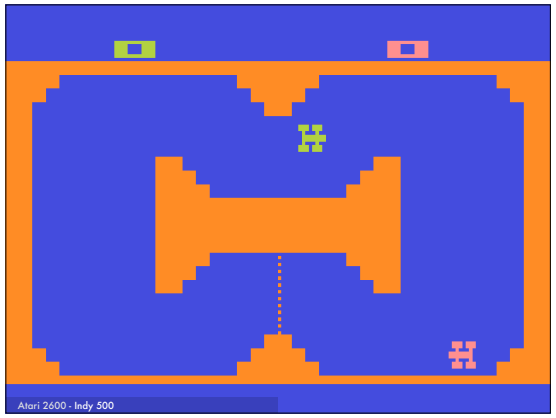
I easily spent hundreds if not thousands of hours playing the Atari 2600.



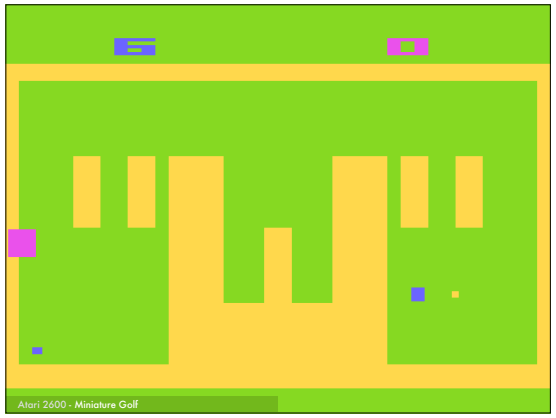
These are the images I was interacting with.



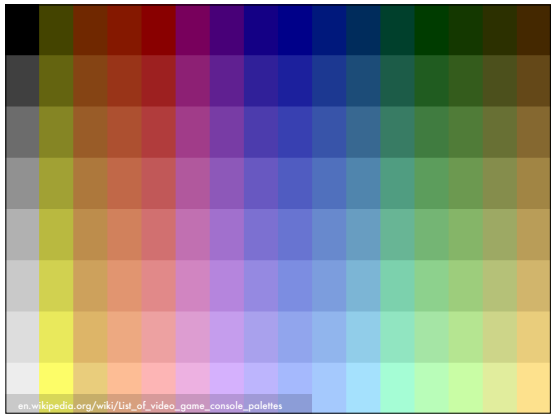
They were simple.



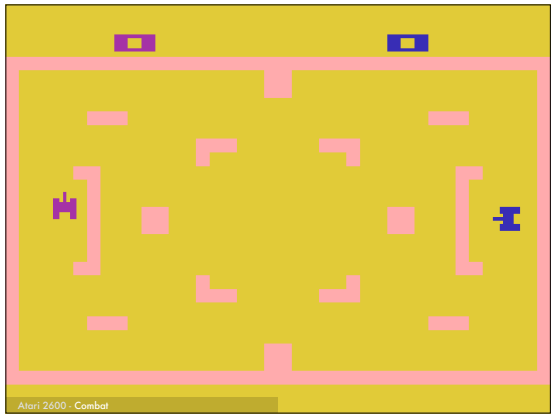
They were an abstraction of the real world and the fantastic worlds depicted on the cartridge's packaging.



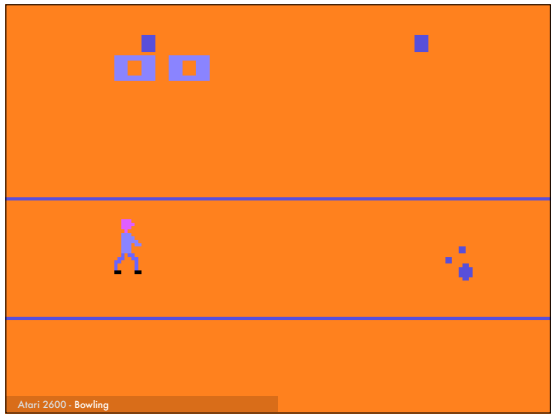
But given their simplicity, these images framed a world where I escaped and explored day after day.



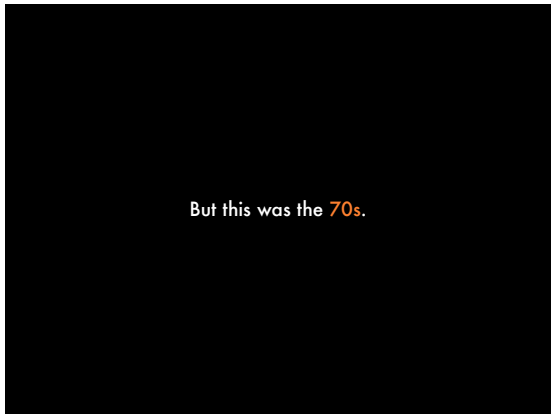
This was the Atari 2600's palette of 128 colors. Quite beautiful.



Compared to a modern console, the technical capabilities 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.

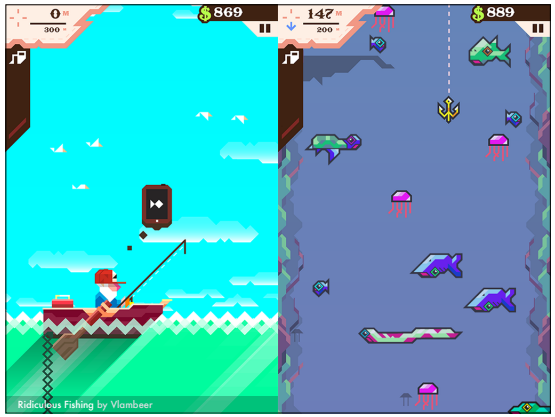


Whether intended or not, this bug reinforced my impression that this game world was limitless. And like the tragic hero of Atari's "Bowling", I stood at the center of a vast universe with no audience to cheer my accomplishments.



There's a lot about the 70s that we left behind. Before we look at creating pixel art today, it's worth asking, is pixel art just a cultural artifact that we reference for solely nostalgic reasons? To find out, let's look at current references to the 8bit, pixel art style.

Sword and Sworcery was released in 2011… by 2013 it had sold over 1.5 million copies. If you're looking for an example of cinematic pixel art, this is it. It's one of my all-time favorite games.



Ridiculous Fishing won iPhone Game of the Year for 2013.

When I first saw Minecraft, I remember thinking, that's 3D pixel art! Minecraft's creator recently announced the game has over 100 million registered users. People play a lot of pixel art games. But what about the future of pixel art games?

I recently backed a game off IndieGoGo called RIOT that is a simulator. As a player you adopt the role of rioters or policemen and let the riot play itself out.

You can see the pixel art style is really interesting. Leonard Menchiari is both lead artist and the game's director.

The team has not announced an official release date. But I can't wait to play this.

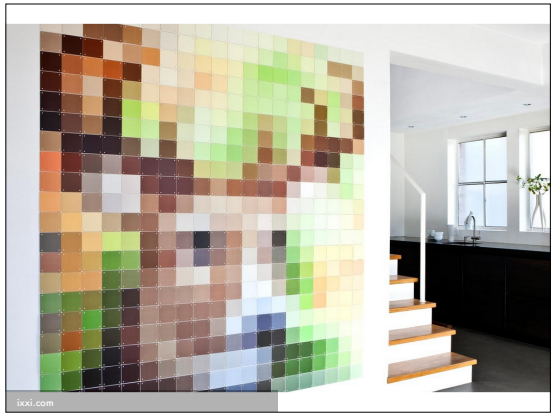
This is another game I'm looking forward to playing called Spirit by Holden Boyles.

It's an adventure game that tells the story of an old man who moves to a village and uncovers its secrets. The style reminds me of the classic adventure games I played when I was a kid.

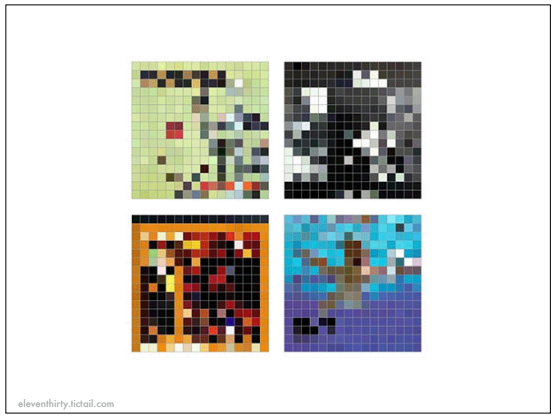
You can see pixel art games reach a massive, global audience and represent some of the most popular games that we play today… and the games we'll soon be playing.

You can go outside and see pixel art. This photo was taken on April 18th, 2008 in New York's East Village.

I remember the first time I saw Antony Gormley's work. Many of his sculptures abstract the human form in a way that reminds me of pixel art.



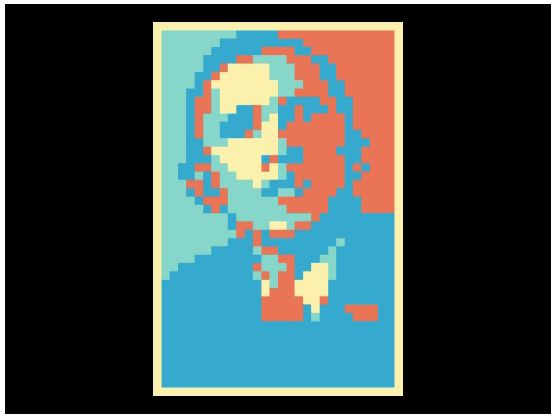
If you're looking to hang something on your wall, upload your art to ixxi.com they'll send you a giant pixel art version…



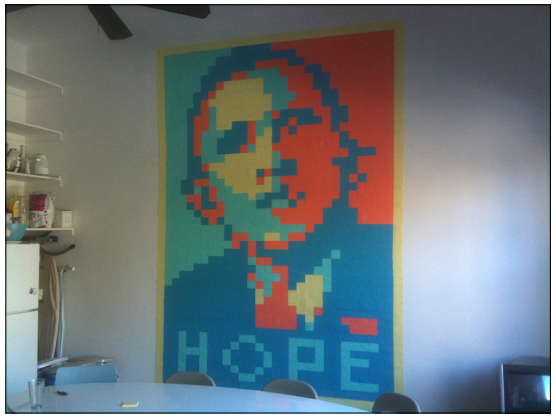
… or a company called ElevenThirty will sell you framed pixel art versions of your favorite album covers.

Whether pixel art is game art, street art, commercial or high art, I believe the style is more popular today than it's ever been.

Soon after the kickoff of President Obama's 2008 campaign, 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."

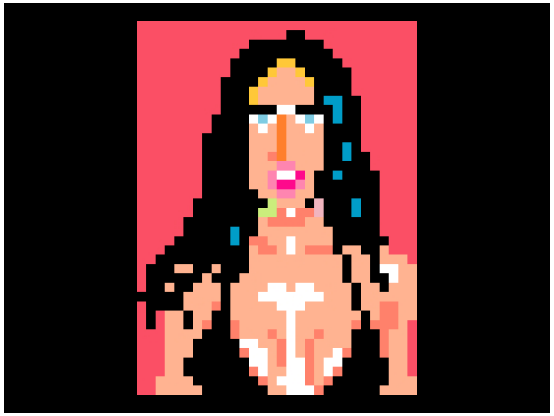


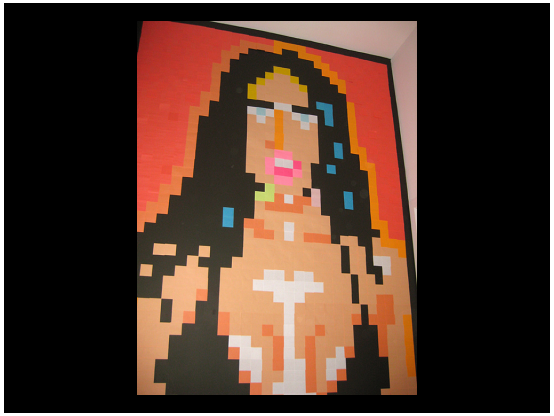
I had seen people create Post-it mosaics and thought, that's it! So I wrote a script that converted the colors in the poster the available colors in the Post-it color palette.



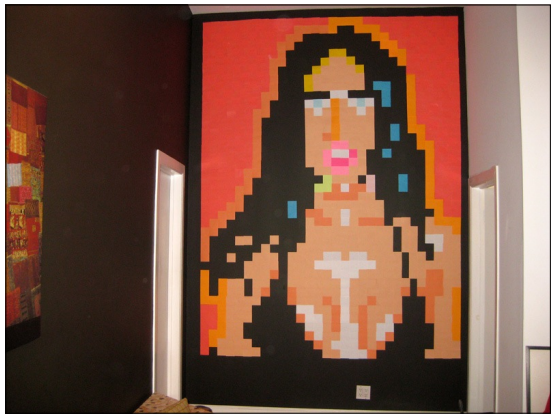
And here's the completed work in my kitchen. It's 30 Post-its wide X 46 Post-its high which is 7.5 feet by 10 feet. That's a total of 1380 Post-its. I think it took 6 - 8 hours to complete.

Do you remember this GQ cover?

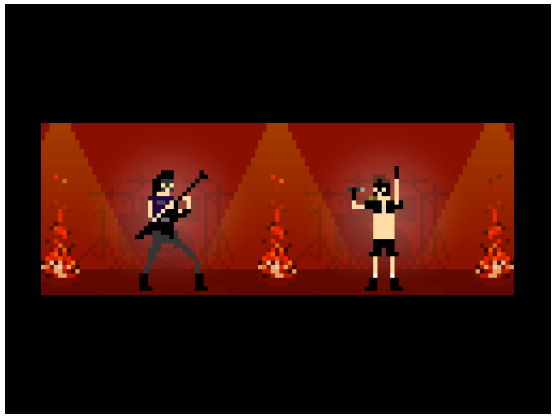




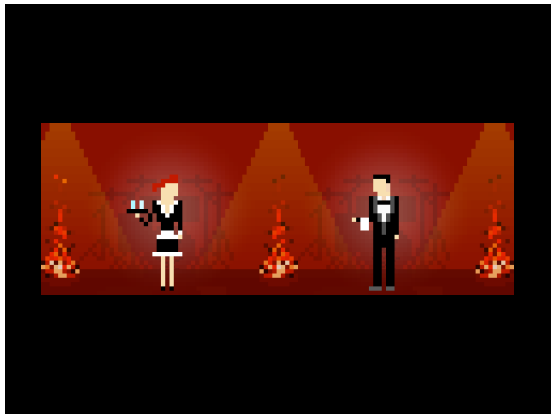
This is a 10-foot high Megan Fox made entirely out of Post-its.



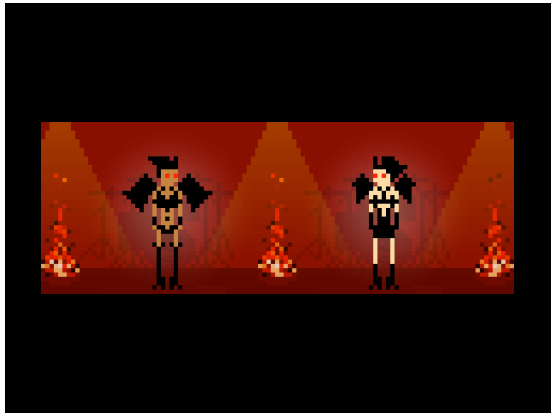
It's 7.5 feet wide and used 12 different colors from 6 different stacks of Post-its for a total of 828 sheets.



But Post-it art is hard.

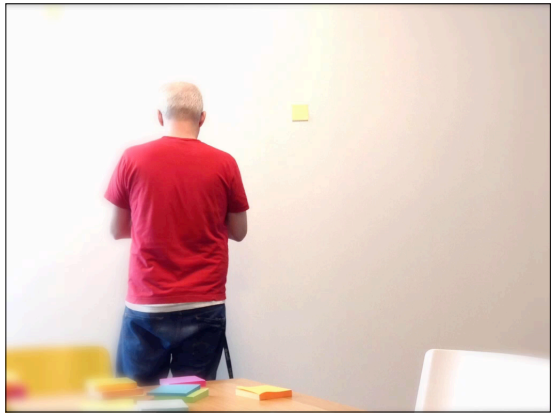


You're up and down ladders. Post-its eventually fall off. So I started creating digital pixel art.

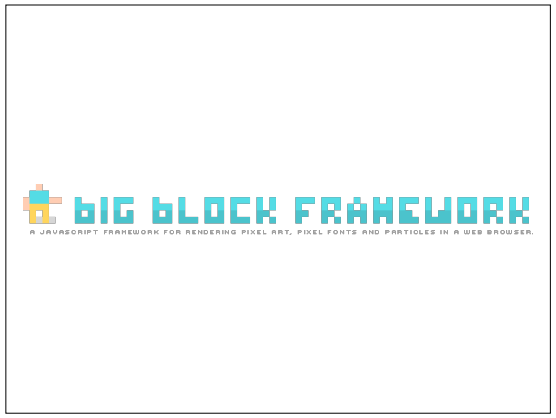


These are from a game I've been working on.

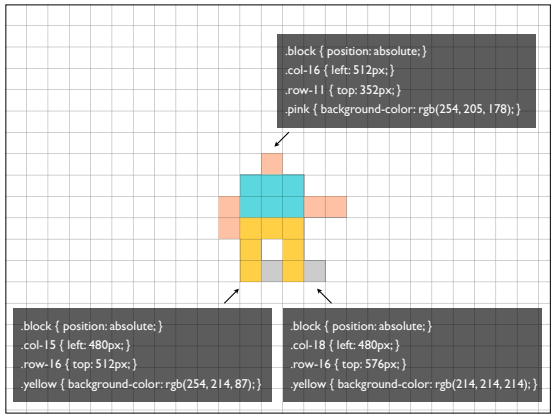
This is me recreating a comic book cover.



These were all static pieces. But the more I created, the more I wanted them to move. And being a JS developer, of course I was thinking about how to animate them in a web browser. My first solution built on what I learned while sticking Post-its on the wall. These Post-its were just big pixels with an x, y location, a color and a scale. 4 properties. Simple. Surely a web browser

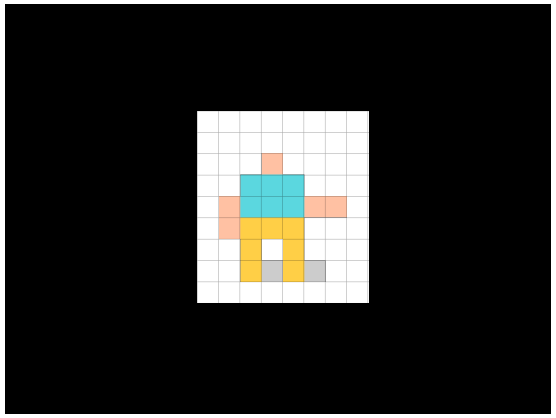


My first approach called Big Block Framework tackled the problem literally. The browser window was the wall and HTML div elements were the Post-its.

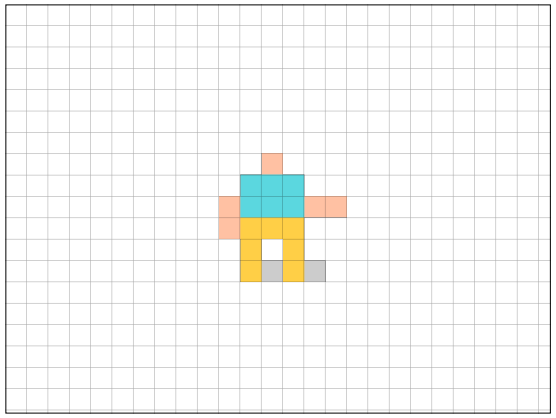


I divided the screen into a grid of 8 pixel x 8 pixel squares. One option was to create an animation loop and update each div's style properties as needed. Turns out that wasn't the fastest solution. Instead, I pre-calculated every possible CSS property and updated each div's class name to drive animation. It's not a very conventional approach. But it worked.

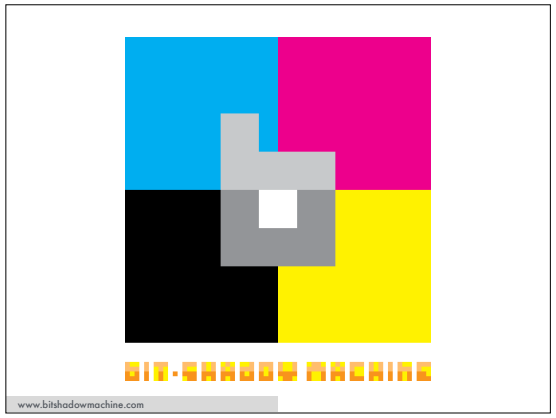
For examples, visit:
http://vinceallenvince.github.io/empirejs2014/



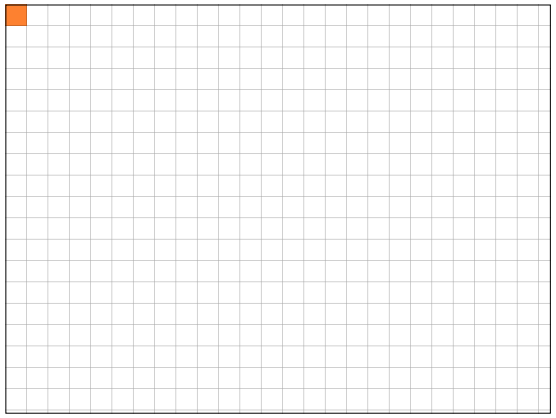
The problem with this approach is pretty apparent. Pre-calculating all possible positions and colors meant I was generating thousands of styles for a relatively small drawing area.



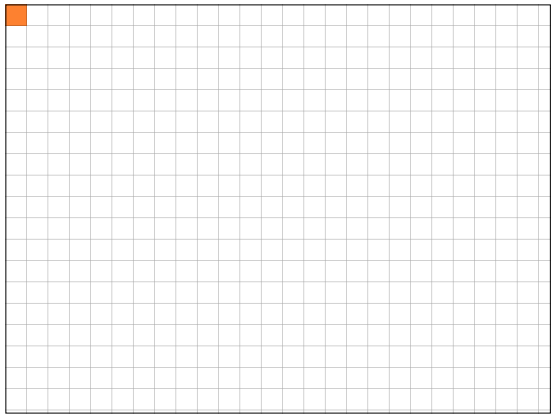
Larger screens were not feasible. Using divs was the real problem. To render the compositions I had in mind, I needed a different way to represent big pixels in a web browser that did not rely on DOM elements.



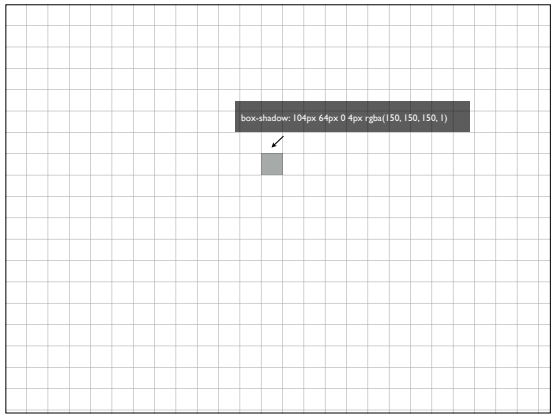
My current approach is called Bit-Shadow Machine which uses the web browser's true equivalent of a Post-it note, the CSS box-shadow. Let me explain.



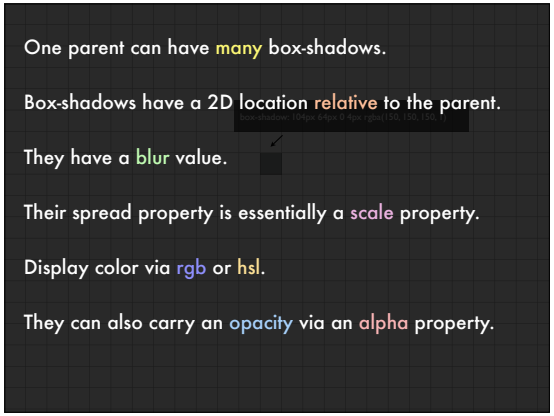
We still divide the visible browser into a grid and place a single div in the top left corner.



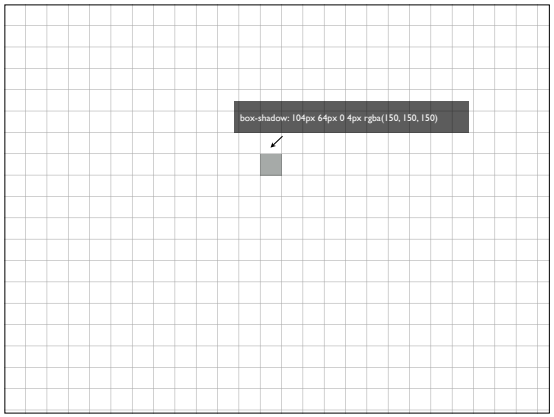
Next, we hide the div by assigning a height and width of 0.



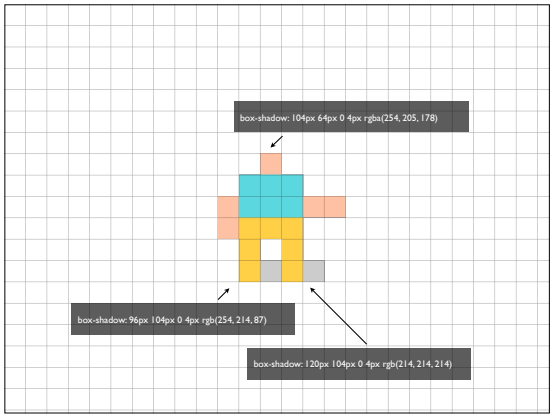
An interesting thing about box-shadows, if we hide the parent like this, we can still see its box-shadows.



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.

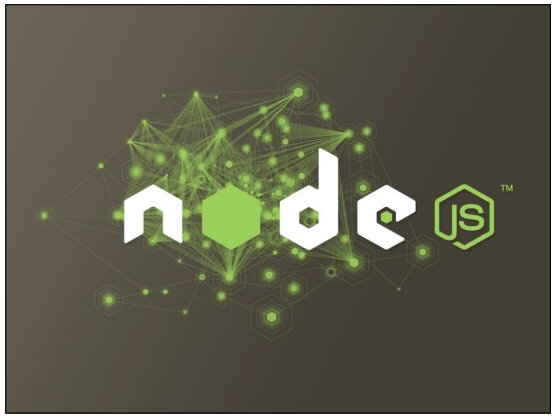


This means it's only a matter of adding more box-shadows and adjusting their properties to get what we want.

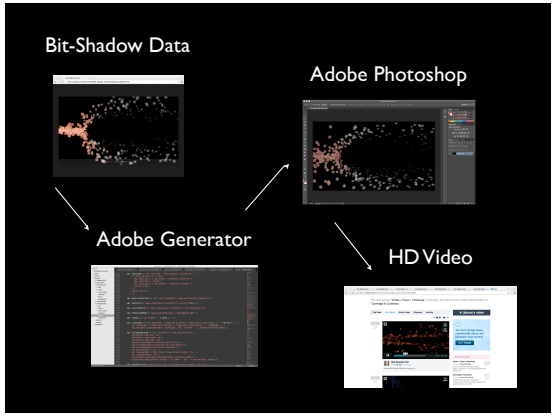
For examples visit:

http://vinceallenvince.github.io/empirejs2014/

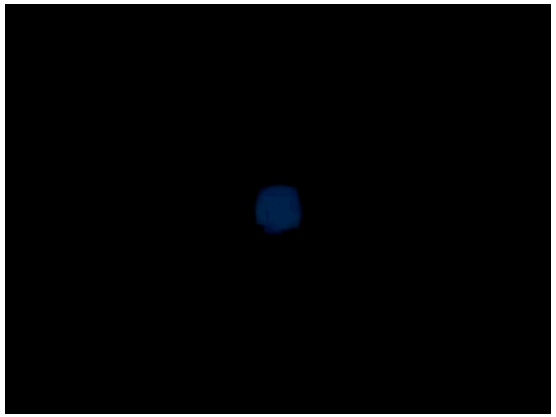
How could we output these renderings to video? We'll use two things… Photoshop.



… 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 compiled into video.



http://www.bitshadowmachine.com/video/blueagents001/

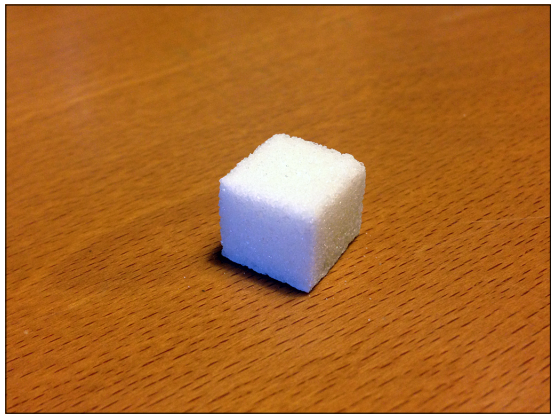
http://www.bitshadowmachine.com/video/redrepeller002/

For examples visit:

http://vinceallenvince.github.io/empirejs2014/

http://www.bitshadowmachine.com/video/orbit002/

Artist Brendan Jamison had a Kickstarter campaign to create a sculpture garden made out of sugar cubes in Harlem, NY. When I first saw this, I thought it's like an edible Minecraft.



It also had me wondering, what is a pixel? Pixels are the building blocks of our imagination. As a single unit, a pixel represents creative possibility. When I see one, I have the irresistible urge to add more. I hope you feel it too.