GYMNASIUM

INFORMATION DESIGN AND VISUALIZATION FUNDAMENTALS

Lesson 3 Transcript

Interaction Design Principles

ABOUT THIS HANDOUT

This handout includes the following:

- A list of the core concepts covered in this lesson.
- The assignment(s) for this lesson.
- A list of readings and resources for this lesson including books, articles and websites mentioned in the videos by the instructor, plus bonus readings and resources hand-picked by the instructor.
- A transcript of the lecture videos for this lesson

CORE CONCEPTS

1. Because of changes in technology, both in software and hardware, and the way, places, and timeframes that we consume media, information graphics can do more, communicate more, and reach more people, than ever before. This leads to a need to formalize the way we approach interactive graphics.
2. The lack of support for Flash on most mobile devices has led to a move toward more web-native formats for interactive graphic creation: HTML5, CSS, JavaScript, and video. These formats can travel well between desktops, tablets, and mobile phones.
3. Interactive graphics involve a wide range of skills, more often than not they are developed in teams where people with complementary expertise can come together to create something great. In other words, it's rare that one person can do it all!
4. Instead of designing information onto a page, animated graphics have the advantage of revealing a topic over time. Instead of relying on arrows and symbols, animated graphics can combine motion, sound, and annotation to create powerful linear narratives.
5. Reducing the interface as much as possible to get out of the way of the story is the best approach. For example, in the New York Times project "Snowfall" the act of scrolling would trigger playback of animations and videos rather than adding extraneous buttons to do the same thing.
6. One thing to consider when designing interactive graphics is that information should be presented hierarchically. Ben Schneiderman defines the visual information seeking mantra as "Overview first, zoom in, filter, then details on demand."
7. It is important to consider all the forms of interaction that your users may be using, particularly those with touchscreen devices. Although the technologies are constantly evolving, consider all the digital platforms in which your graphic may be consumed from the outset, and let that guide how you design, whether that means one version, or several.
8. An increasingly popular technology framework for interactive graphics is D3.js, a JavaScript-based tool for loading data into a web page, and generating visuals from that data. D3 stands for data-driven documents. Essentially, you start with a set of data, and end up with a set of web-based documents, or any format that can be read by a web browser. D3 facilitates this connection between the data, and the documents.

ASSIGNMENTS

1. Quiz
2. Design an interactive

   Take the graphic you created in the previous lesson, and design an interactive, or animated version. You don’t have to actually create the interactive version unless you are feeling especially ambitious, but create a plan for how you would re-design the graphic.

   You can create a storyboard with notes that demonstrates what the graphic would look like in interactive form, how a user would interact with it, and how it would look in different states of user interaction.

INTRODUCTION

This is lesson three of Information Design and Visualization Fundamentals, an online course developed by Aquent. In the previous lessons, we defined information design and visualization, learned about some of the fundamental principles of communicating with a visual language, and became familiar with a large range of information graphics; covering everything from static and illustrated print graphics, to animated information graphics, to interactive data visualizations.

In this lesson, we're going to focus specifically on the interactive graphic form. There are many reasons to focus a chapter of this class entirely on the interactive graphic form. Because of changes in technology, both in software and hardware, and the way, places, and timeframes that we consume media, information graphics can do more, communicate more, and reach more people, than ever before. This is something I've absolutely seen evolve, even in the roughly 10-year span of my career, so far.

The whole concept of information graphics has gone from something I had to explain, to something that is a recognized part of the media culture. Media businesses, from journalism to advertising, have taken note of the engagement brought by these projects. This change can be attributed to how much more sophisticated and engaging information graphics have become, in tandem with the new technologies that have brought interactivity into the equation.

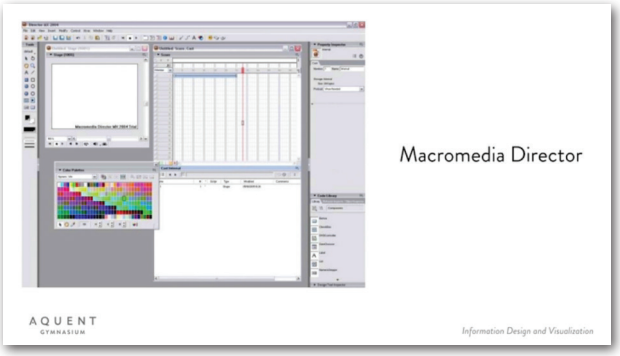
With a changing media landscape, workflows have changed as well. At the New York Times, my experience has been one of, in some ways, a reversing of priorities. Graphics are often created for both print and digital platforms on the same topic. It used to be that the print graphic always received the initial focus. And, once that was in a close-to-publishable shape, we would figure out what to do with it online.

But now, it's usually the opposite. There are so many more options and considerations when designing for digital platforms that, out of necessity, we usually begin there when designing a graphic. Then, once that is largely figured out, we will design the print version and that is usually a careful, and considered reduction of what we've done in digital. Occasionally, this reduction ends up as an improvement, a better curation, but usually, it has to be a compromise.

As a matter of fact, one of the larger graphics projects that I can point to that seem to really exemplify this change in workflow was the Ground Zero rebuilding project that we have seen throughout this class. The animated graphics pertaining to this project were developed first. Then, the assets and ideas born from these were reimagined for the static print design.

It almost seems quaint, now, to imagine the dominance of print in our design thinking. But, in a few short years, thinking about information designs as moving, interactive, alive things has created a cultural shift in the discipline.

So, how did we get here? Interactive graphics began in simpler ways, along with the rise of the internet. Early tools, like Macromedia Director allowed programming, animation, and graphics to be combined, and published as applets in the Shockwave format that could be published on web pages. Macromedia was eventually purchased by Adobe, along with their Flash product, which came after Director as a purely web-focused interactive graphics, and animation tool.



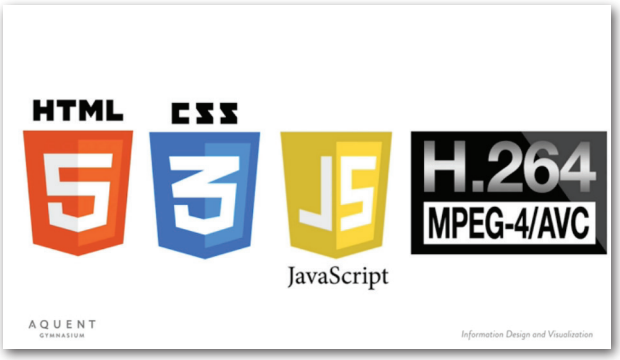
Flash, along with its ActionScript language, became the predominant platform for creating interactive graphics for the web. Sophisticated animations and interactives could be created with Flash. Here is an early Flash graphic we created at the Times called, “The Master of Clay Takes Aim at the Fast Courts.” The graphic compares Rafael Nadal’s performance on different tennis court surfaces, while at the same time explaining how these different court surfaces affect the way the game is played.

Flash is still widely used on the web, but at the Times, and many other media companies, it has all but been abandoned. This can be attributed to one big, but also very small, reason, the iPhone. The iPhone essentially invented mobile, and at the same time, completely shunned Flash. As you are likely aware, any Flash website visited on an iPhone simply displays a message that the content cannot be displayed.

This was a big deal, and made for a lot of outcry, much like many Apple decisions before it; abandoning the floppy drive, or abandoning the optical drive, et cetera. And, as more and more people began consuming media on the iPhone, and similar devices, it became apparent that using a platform that could not be accessed was no longer a viable option, despite the many design opportunities that Flash afforded.

Much like the abandonment of the floppy disk, Apple’s help in bringing about the beginning of the end for Flash was painful at first, but, ultimately, the jolt that interactive graphics needed. Flash is simply not native to the platform, and doesn’t take full advantage of the modern interconnectivity, and searchability of the internet.

So, with the departure from Flash came a move toward more web-native formats for interactive graphic creation: HTML5, CSS, JavaScript, and video. These formats can travel well between desktops, tablets, and mobile phones. This is not to say that different versions don’t need to be developed for these different platforms. But, the underlying technologies are transportable.



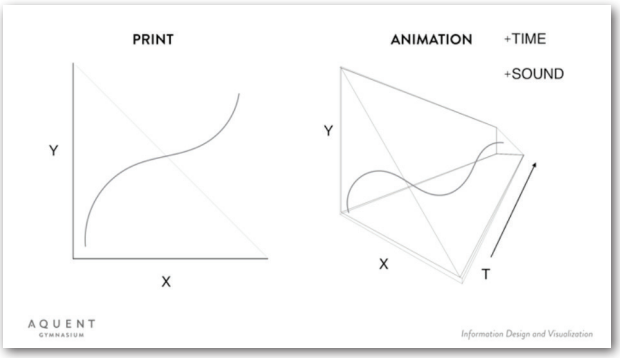
So, to recap, new technologies are guaranteed to keep changing, and being invented, but, the shift over the last decade from predominantly consuming media on print, to consuming media on screens, was truly a seismic change for information design. The opportunities for interaction and movement in graphics on these platforms is exponentially more diverse.

In the next lesson, we'll take a look at the diverse variety of interactions that graphics can employ to educate, entertain, and inform readers.

DESIGNING FOR DIGITAL ANIMATION AND INTERACTIVITY

As mentioned in the previous chapter, the proliferation of screens as one of our main access points for media, has created many opportunities for information design beyond the static information graphic. Technologies like high-quality web video, and the advancing web native formats of HTML, CSS, and JavaScript, along with many additional libraries and extensions of each, gives designers many new ways to communicate to an audience. As we look at a number of examples in this chapter, it's important to keep in mind that interactive graphics involve a wide range of skills, and so, are more often than not, developed in teams where people with complementary expertise can come together to create something great.

So, let's take a look at how we design with these technologies in mind. The most basic interaction is simply to click play. Animated information graphics are often presented in a video format.



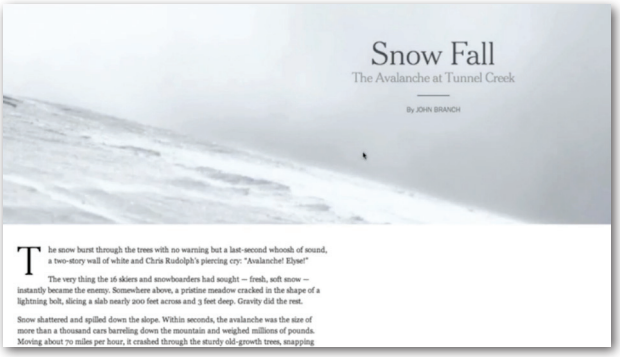
Instead of designing information onto a page, animated graphics have the advantage of revealing a topic over time. Instead of relying on arrows and symbols, animated graphics can combine motion, sound, and annotation to create powerful linear narratives. Take this graphic I worked on at the Times that's called, "Connecting Music and Gesture".

In this project, we used motion capture to record the movements of famed conductor Alan Gilbert, as he conducted a small ensemble, to better understand how conducting works. This motion was annotated with motion trails to help readers see the kind of motions he was carving out in space. A project like this simply does not translate well to print, as it requires motion and audio to truly grasp the meaning.

The page was designed with the video at the top. And below, we pull out key quotes for the reader to pay special attention to. Presented in a video format, this graphic can be viewed just as easily on desktops, tablets, and mobile phones. Teamwise, this project included a specialized outside group to help record the motion capture, a 3D modeler and a developer for the page.

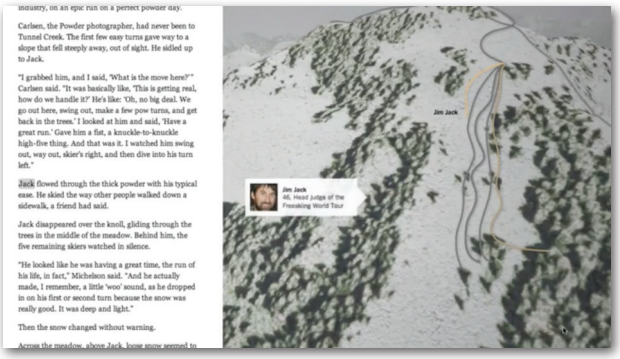
Sometimes animated graphics in the video format don't require pressing play at all. A device we have used in a number of recent projects is to combine scrolling, the most natural interaction with a web page, with autoplay video. This essentially removes all cumbersome interface elements, while still allowing a reader to interact with a graphic.

One of the first big ways we used this was in an integrated media story called, "Snowfall", about a deadly avalanche. With this piece as you scroll through the story, you come upon animated graphic elements that automatically play as you reach them. The first you encounter is a flyover of the terrain the story is describing.



It creates a highly-engaging experience of multimedia and story. To continue on, once you've consumed as much of the graphic as you want, you simply keep scrolling. The same device was used in this story for animated graphic elements that don't take over the screen fully.

In this section describing the different paths skiers take in the story, running alongside the story is an animated graphic that draws out each ski path being described. It's a particularly hard thing to follow with just description. So, the graphic is a great aid in understanding this element, which is in turn key to understanding the way that events play out.



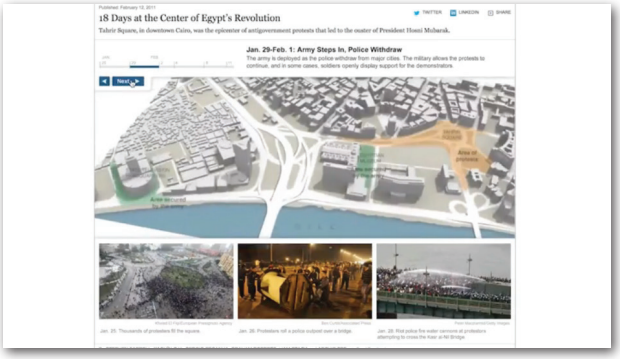
Again, notice that the only actual requirement on the reader is to scroll. Interaction and graphics need not mean many buttons to be effective. In fact, in many cases, reducing interface as much as possible to get out of the way of the story is the best approach.

Another example in this story is this little side element that shows how a ski airbag works. This is a difficult thing to understand from only the description in the story. So, having this little autoplay video-based animated graphic to the side is very helpful in understanding how this works.

"Snowfall" was a multi-chapter project requiring skills across the gamut, with a team of approximately 12 people. We used the scroll autoplay interaction again for the 2014 Sochi Olympics. Here, by scrolling as you would in a written story, you seamlessly can move between photography, video, annotation segments, and animations. By using the familiar (and native to the web) scrolling interaction, the user already intuitively knows how to interact with this graphic even though they are not scrolling type or images in the usual fashion.

This is a good time to recommend some advice. If you find that you need to add a section to your interactive graphic called "how to use this graphic," you should probably rethink your design. Interactives should not need instructions. And, as soon as you add a section like that, you're going to lose readers. Design intuitively and you won't need instructions.

Another kind of interaction often used in interactive graphics is the stepper. This is another linear form of interaction, like the scroll, or play base interactions we have seen so far. The difference here is that it can give the user a bit more freedom to jump around, especially if each step is given its own button. Steppers are a great way to develop a story, and benefit from allowing the reader to linger on each frame. This way more type, and more dense information can be presented at each stop rather than in a video, say, where the designer might need to balance between information on screen, and having too much static imagery.



At the Times, we have used steppers often for breaking news stories. For instance, the format was used extensively in our coverage of the Fukushima nuclear reactor disaster in Japan. We created several graphics in this format to explain what these reactors were like inside, and what was happening inside to cause such devastation.

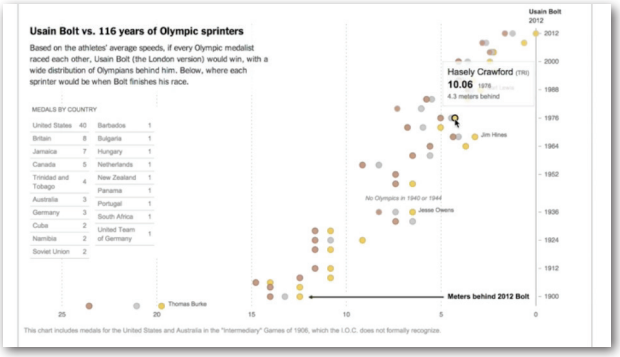
Another example is a stepper created to explain how hydrof-racking to collect natural gas works. Each step has relevant illustration and annotation to spend time with. But, the animation connecting each step is what gives the reader the sense of how the process works.

Often, interactive graphics will need to have a button that, when clicked, gives certain options, and creates some kind of feedback. Buttons should be designed to obviously be buttons, first of all. They should also be prominent, so that a reader doesn't have to search to find them, or to realize that they're there.

In this graphic about how the Metropolitan Opera stage works, large type leads the eye directly to the button. The button also gives feedback when the mouse hovers over the area of the button graphic. When clicked, the button causes an action. In this case, it triggers the animation of the model of the Met stage, to demonstrate how this particular element of the stage works.

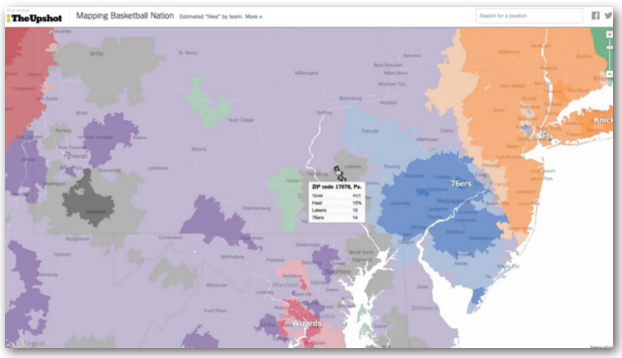
This leads us to another form of interactivity to consider: hover. When bringing the mouse over a certain area of a graphic, an action can occur without any clicking. This is often used in data visualizations, and in mapping applications.

Take this data visualization we created for the 2012 London Olympics. In the middle section of the graphic, we have a chart displaying data that represents the finish times of all the medalists throughout the modern Olympics; showing where each athlete would have ended up in relation to the current record holder. The chart itself does a good job at showing the general picture of improvement over time. But, the hover

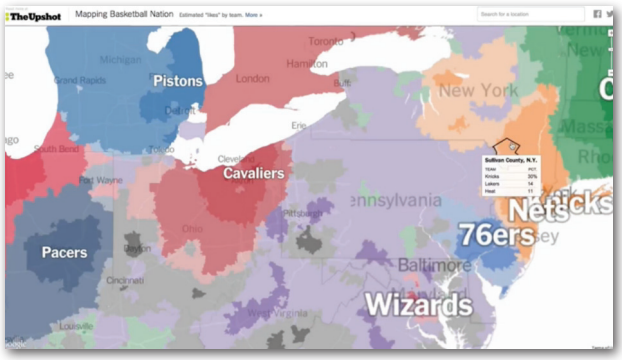


interaction with the mouse allows the reader to get more specific data about each data point; the name of the athlete, the nationality, the year, the actual finish time, and how many meters exactly that would put that athlete behind the current record holder.

Trying to display all this information in the chart would have made for a fairly cluttered ugly graphic, and would in many ways obscure the way the chart shows the general progress, over time. Another example of hover is in the interactive maps we've created. Take for example, this map showing basketball allegiances across the country collected by analyzing Facebook data. As we hover the mouse over the map, we access more specific data about how each county cheers.



One thing to consider when designing interactives generally is that information should be presented hierarchically. Ben Schneiderman defines the visual information seeking mantra as overview first, zoom in filter, then details on demand. This map demonstrates this well.



First, we are given a general impression of the data across the country. But then, we can zoom in and filter this map as it pertains to our particular interests and perspective. We can use this map to discuss a number of other interactions as well, for example, double click to zoom. Each time we double click the mouse, we get closer in to the map, revealing more detail. Click and drag, this allows us to move the map around so that we don't need to zoom out to explore more areas.

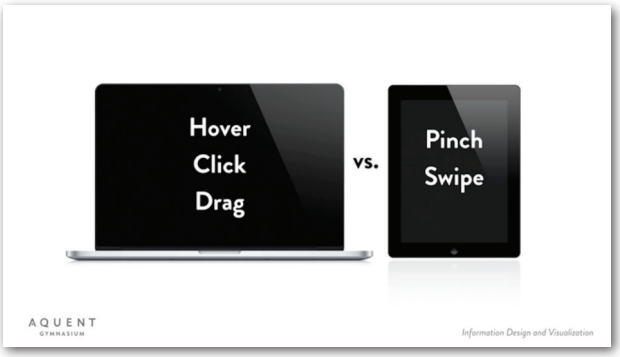
Now, notice this interface element in the upper right. This is called a slider. Sliders are another commonly used interaction element in graphics. In this case, the slider presents another way to move in, and out of the map. In other interactives, like this Times buy versus rent calculator, sliders allow the reader to easily change a variety of parameters to assess the best living decision, based on their finances.

Another interesting interaction that is used less often, but can be useful in the right circumstances, is drag and drop. In this Times graphic from the 2012 election, states represented as scale bubbles can be dragged into different piles, so that the reader can play with different election scenarios. Take note that many of these interactions have involved using a mouse; something that may in the near future be an endangered species.

Certainly the mouse is still a dominant way of interacting with digital media. But, the rise of touch screens has forced designers to rethink many of these interactions. Clash was abandoned by many interactions designers because of its inability to port to mobile environments.

We can run into a slew of issues if we don't consider the differences between mouse and touch interfaces when designing our graphics. We don't always want to have to design multiple versions of a graphic when we want them to work on all screens. Sometimes this is unavoidable. But, it's something you should always consider from the outset.

For instance, what will hover mean in a touchscreen environment? On most touch screens, there's no such interaction. So, if hover is going to play a major part in your interactive, and you want it to work on touch screens as well, it's good practice to consider an alternative interaction that won't require you to rethink your whole graphic structure. On most touch screens, when presenting an interactive that contains hover instructions, clicking once will cause the hover response, and clicking again will act as the desktop's click response. In some cases, this will suffice, but not always.



At the same time, touch screens add interactions that are not typical on desktops, like pinch to zoom or swipe. These interactions can sometimes be much more elegant than the counterpart interaction solutions on desktops. This, as mentioned, is a very new area for interaction design.

So, new solutions, and best practices are being developed all the time. But for now, the best advice I can give is to consider all the digital platforms in which your graphic may be consumed from the outset, and let that guide how you design, whether that means one version, or several. The current workflow at the Times is to treat each platform as its own branch of the graphic production process, and, when possible, to multi-thread the process, and design for each, simultaneously.

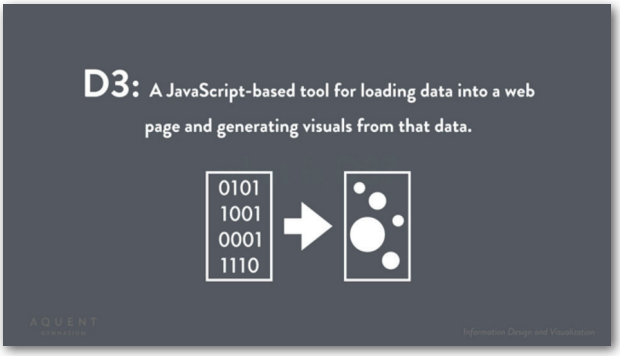
So to recap, the universe of interactive, and animated information graphics springs many opportunities and considerations for how we approach design. We now need to consider the many different kinds of interactions at our disposal. And, the growing variety of devices from desktops, to tablets, to mobile requires consideration for how our graphics will port from one to the other. In the next lesson, we'll take a look at d3.js, a JavaScript library that has become an essential tool for interactive information designers on the web.

INTERACTIVE DATA VISUALIZATION WITH D3.JS

As we've learned so far, technologies can come and go. Even huge game-changing ones like Flash can disappear in, pardon the pun, a flash. And so, thus far, I've tried to teach concepts and approaches over any particular technology. That said, it is at the moment of the creation of this class, a disservice to discuss the world of interactive information graphics without giving some awareness of D3.js.

We've learned that JavaScript is one of the important technologies driving modern interactive information graphics. D3 is a JavaScript library that accesses the power of the language without requiring a full knowledge of JavaScript, or everything to be written from scratch.

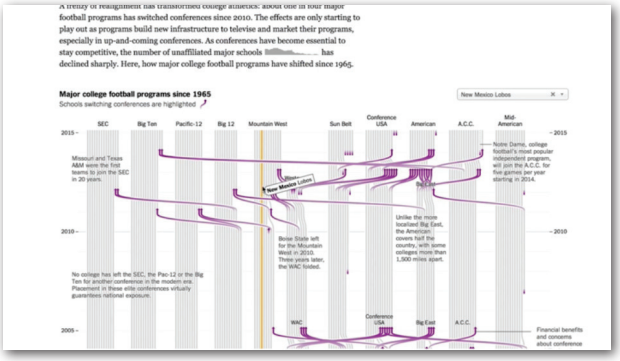
So, what is D3 exactly? D3 is a JavaScript-based tool for loading data into a web page, and generating visuals from that data. D3 stands for data-driven documents. Essentially, you start with a set of data, and end up with a set of web-based documents, or any format that can be read by a web browser. D3 facilitates this connection between the data, and the documents.



D3 was primarily written by a colleague of mine, Mike Bostock. I recommend perusing his website, address noted on this slide, to see examples and updates about D3.

Many of the more sophisticated online interactive graphics you've come across have likely been created with the help of D3. From the Times, the buy-versus-rent calculator, and the basketball fan maps we saw previously, were both created using D3.

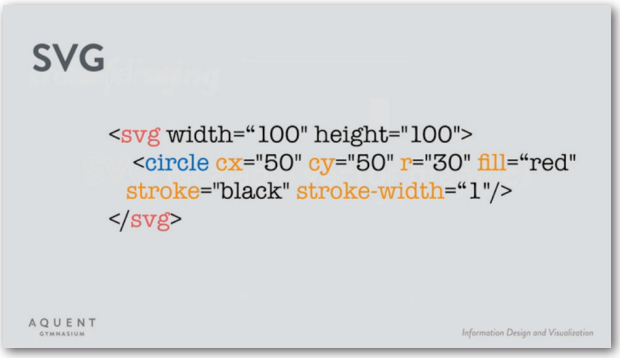
Another example, this graphic showing the history of NCAA conferences, and this graphic showing the path to victory in the America's Cup finale, and this graphic showing the variability of tax rates for US companies.



D3 ultimately allows for the generation and manipulation of web documents with data. This is accomplished by loading data into the browser's memory, binding data to elements within the document, creating new elements as needed, transforming elements, and setting their visual properties, and transitioning between states in response to some input.

D3 is most useful and elegant when creating and manipulating a type of an on-screen element called an SVG, or scalable vector graphic. If you've used Adobe Illustrator, then you already understand what vector graphics are. And, Illustrator can be used to export elements in the SVG format.

But designers working in D3 also define SVG elements using code, alone. So, for example, to draw a circle, the syntax might be something like this. This creates an SVG element, which can be thought of as the canvas you're drawing into, and tells the circle where on that canvas to be drawn, giving it a radius, a fill, and a stroke, just as you might do with the drawing tools and pallets in Illustrator.



The creation of SVG artwork, and its integration with HTML, CSS, and JavaScript is beyond the scope of this course. But luckily, there are some excellent resources out there. Here are two we can recommend.

Todd Parker of the Filament Group has made the slides from his presentation, Leaving Pixels Behind, available online. This is a great introduction to SVG creation and workflow from the designer's point of view. The link is below, but we will also provide it within the resources document for this lesson, available on the classroom site. Additionally, we could recommend this presentation on SVG by Sara Soueidan named, Styling and Animating Scalable Vector Graphics, available on YouTube.

The power of D3 comes in its ability to load in data, attach it to an SVG element, and then scale, or move, or otherwise transition the element based on that assigned data value. And, D3 can select, create, manipulate and bind data to any web page element, be it a paragraph, or a div, or an SVG element.

D3 can create axes for charts, and quickly scale, and adjust these axes dynamically as the data changes. And, it has event listeners that make possible the many interactions we discussed in the previous chapter, so that your graphic knows when a mouse is hovering over an object, when a click has occurred, when a slider is being moved, et cetera.

A good tool to see D3 in action, as well as its relationship with HTML and CSS, is CodePen. Mike Bostock created a CodePen that lets you play around with the syntax and parameters, and see how the results change in real time.

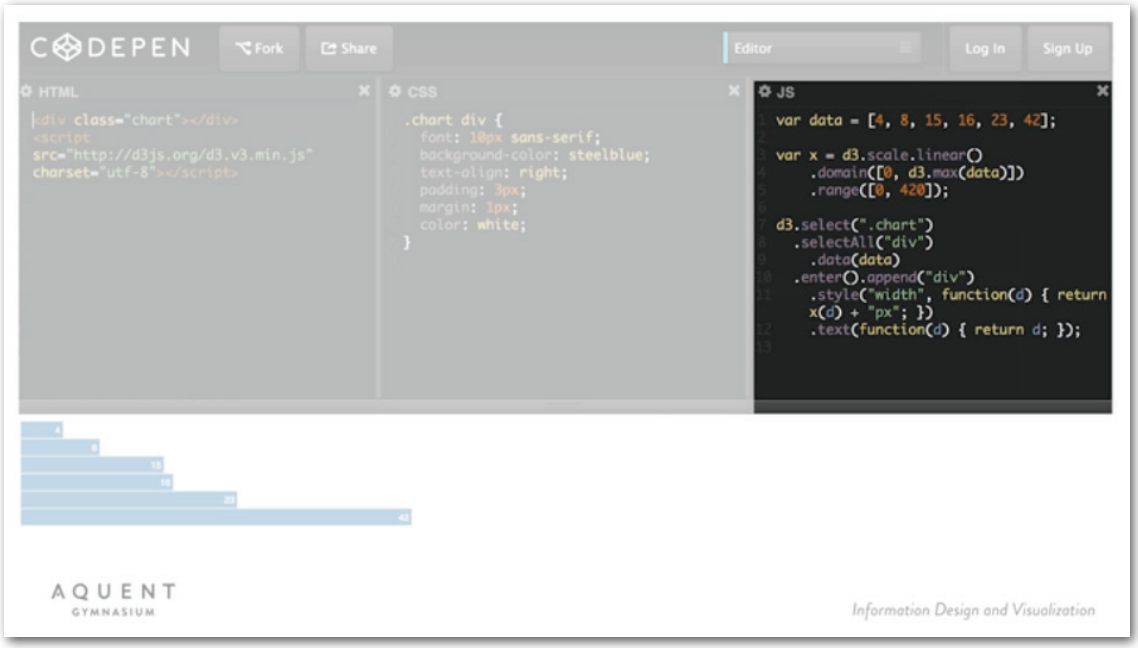
Here, we have a bar chart created from the HTML div element. In the HTML code, a div with the class type, chart is created and, very importantly, our D3 library is loaded. This needs to be included in any page that plans to use D3.

In the CSS code, the style of our div element is defined, which is, in this scenario, the visual element that will make up the bars of our chart.

In the JavaScript code, our data is represented in a variable called, data. Below that, D3 scales, which helps our chart change dynamically as the data changes, sets a domain and range for the chart. Below that, is the very common D3 select operation, which essentially creates new div elements on the page, and binds the data to these, creating our bar chart.

So, let's see what happens when we start changing some things in the code. Right now, our bars are blue. But if we change this element in our CSS to, say, red, we will see our chart colors change in turn. Over in the JavaScript panel, if we change this eight in the data to a 30, we will see our chart change to reflect the changing data.

But, one of the great things about D3 is its handling of changing ranges of data. So, if we change the number to a 300 instead, we'll see that the entire chart rescales to make room for this data point, rather than allow it to run off the end of the web page.



So now, you've seen how data, and D3, and style all interact to create the visualization on the screen. I recommend visiting this site, and playing around with this some more on your own.

You might also want to check out Mike's tutorials, if you know that D3 is going to be important in the kind of graphics you would like to produce. These should get you started creating right away.

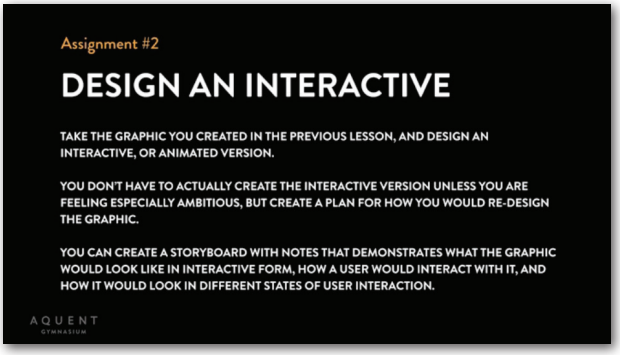
Learning D3 is a course unto itself. So, for this class, it's enough that you understand that it's one of the main technologies driving many of the interactive graphics you see today, and have a general understanding of the role it plays in making these graphics possible.

Connecting visual elements to data in a flexible way, and having these elements react to input from the user, are the fundamental building blocks of interactive graphics, and are why it's important to be aware of powerful tools like D3.

In the next lesson, we'll take a look at the many tools of the trade related to information design, as well as some reading resources that will help you continue to learn beyond this course.

Homework. The first assignment is a quiz. Each lesson in the course has one and it's designed to help you reinforce the concepts covered in the lesson. Quizzes are available on the classroom site, after this video is done.

For assignment two, take the graphic you created in the previous lesson, and design an interactive version. You don't have to actually create the interactive version, unless you are feeling especially ambitious. But, create a plan for how you would redesign the graphic. You can create a storyboard with notes that demonstrates what the graphic would look like in an interactive form, how a user would interact with it, and how it would look in different states of user interaction.



After you've finished your assignment, post a link to it in the classroom, and then find one of your classmates' assignments, and take some time to look it over, and offer feedback.

That's it for now. I'll see you in the next lesson.