The web as craft or assebly line (part 1): Everything easy is hard again

Doing web development used to be simple. All we needed was a text editor and either good knowledge of HTML, CSS and Javascript or bookmarks on your browser pointing to Eric Meyer's CSS resources on the CWRU server and links to the best tutorials you could find about CSS and Javascript with the differences for each browser and how to branch the code to acommodate each browser's idiosyncracies.

Those days are long gone. Reading David Rupert's The tangled webs we weave reflected some of my own struggles when working with existing projects.

But even when working on our projects, starting a new project forces us to ask many questions before we even get started.

Are we using Typescript? (and do we need to?) Are we using a build system? (if so which one?) If we're not using Typescript what version of the language are we using? Are we transpilling it so that it'll work with ES5?

If you build web content on a regular basis you may have internalized this and have a set of scripts ready to run and generate the skeleton of a website project ready to go with all the tools you know you will use.

But what happens to people new to frontend, backend, or full stack development or to people who have left the field for a while and now return to a drastically changed development landscape?

If you're starting your own project, research all the alternatives and pick the ones you think may work best (and accept that the level of complexity will increase as the project grows) or learn the new tools the project has implemented (if you're joining an existing project), whether you agree with them or not.

- Everything Easy is Hard Again — Frank Chimero
- What Screens Want — Frank Chimero
- The Web's Grain — Frank Chimero
- The web is too damn complex — Robin Rendle

- The arc of design — Robin Rendle

The web as craft or assebly line (part 2): The soul of the web

The other side of the argument is that we've let the web become boring. We've lost the whimsical designs from the early days.

Examples like the Space Jam website, from 1996, show what earlier websites looked like:

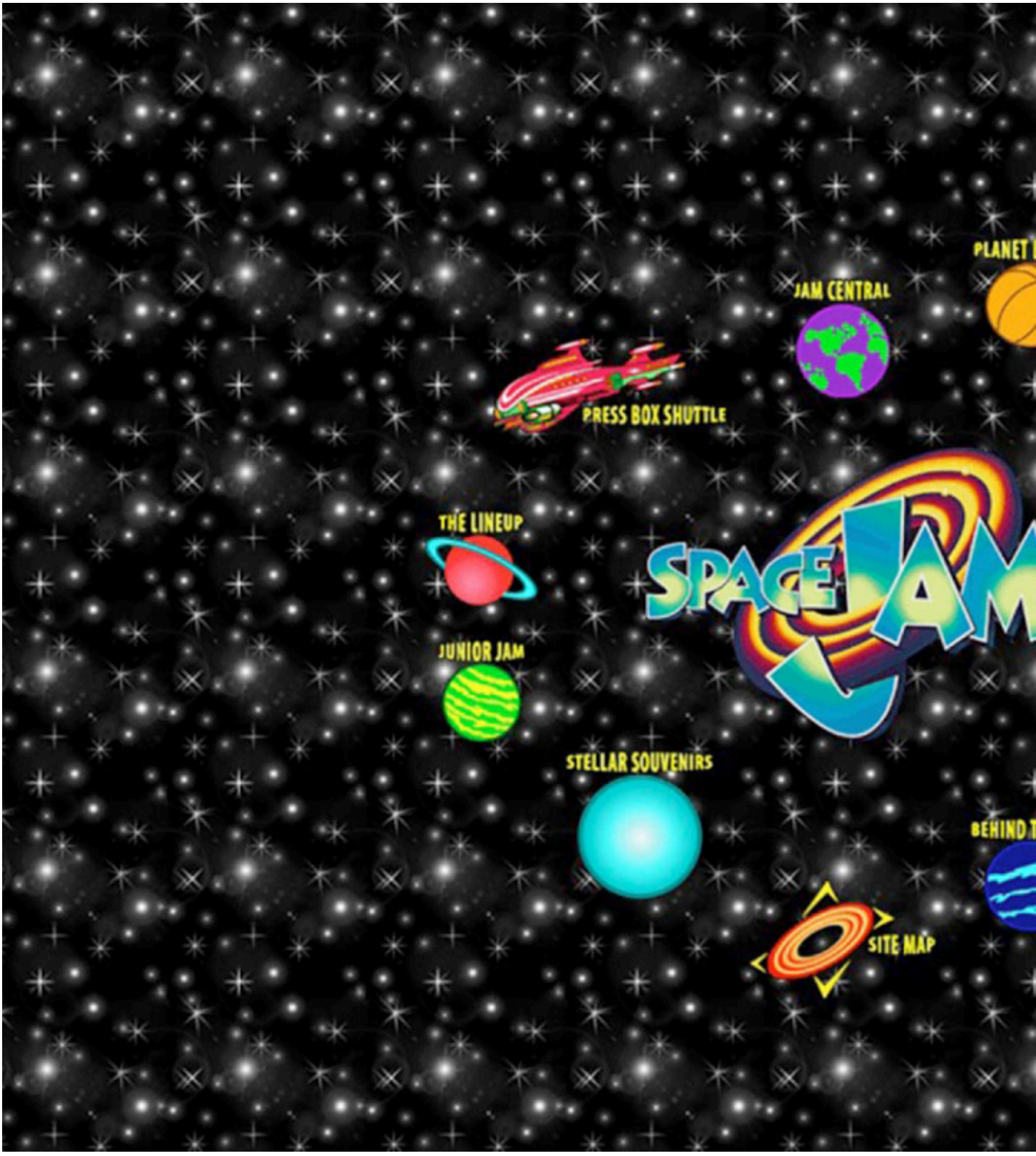

Figure 1: The Space Jam website shows early design concepts for websites

As much as some of us hated it, tables for layout had their use :)

Dave Ellis' All Websites Look The Same and Sarah Drasner's In Defense of a Fussy Website ask two different but related topics: the sameness of our websites and the need for some level of whimsy on the websites we build.

It doesn’t have to be a big thing. It could be just as simple as the top right corner of the screen doing something different when youinteract with that portion of the screen (and nothing at all when you don’t).

It could also be a design that moves away from a box/column layout and lets itself be more engaging and attractive to users without the need for a lot of extra code.

Why can’t the web be like print?

Ever since we’ve had a commercial web the talk has been “the web is not print” or a variation thereof. The web is not print, true, but that doesn’t mean that we should limit our uses of the technologies that make up the web.

In her presentation about the nature of web layouts, Jen Simmons walk us throught the history of layouts on the web and how they have evolved and we have not taken advantage of them.

She goes on to show how newer CSS technologies make it possible to break out of the current layout ruts we’re in.

Andy Clarke’s Inspired by design series presents interesting takes designing web content and how to apply design techniques to web content.

Likewise, Jen Simmons shows how can we make accessible magazine style layout on the web and how to make them responsive to keep them working in smaller form factors.

What I found particular interesting from Andy's presentations is the concepts of compound grids, the idea of combining 2 grids of different sizes and using the resulting grid in our designs.

CSS has opened a lot of creative opportunities for us to use. Most of the time we don't even have to code workarounds, the way things work without the enchancements still provides a good user experience.

It’s up to us to explore these new opportunities and leverage them to create better user experiences.