SYNERGY: THE POWER OF CODEBASE "CROSS-POLLINATION"

Why Coding from Scratch can be Counterproductive

by Jeremy Fonte

INTRODUCTION

Innovation and creativity do not arise out of nothingness; rather, the creative process is a synthesis of prior art, code, culture, or knowledge, which together produce a new work. This synthesis can be abstract and facilitated by vague memories and learned standards, such as when a musician writes and records a new song; or, it can be concrete, such as when a coder combines various JavaScript libraries together to form a new product or tool. It is this second form of creative synthesis that I will cover in this article, with the hope of inspiring others to think twice before writing new code libraries from scratch.

The main code library I'll be referring to throughout this article is my JavaScript 3D library, 3DFOE. This library is a mix of Three.js, Google's Explorer Canvas, Modernizr, and some "duct tape" code I wrote to stitch it all together. You can examine this library further, and view a demo, by visiting http://3dfoe.com.

THE CODING FRONTIER

The very nature of computer programming is that of discovering the unknown, synthesizing a new solution through creative exploration; and while the sheer number of existing programming projects may seem intimidating or incomprehensible, they are the source of great power to any new project. Be they open-source or closed-source, free or commercial, the countless programs, libraries, and web APIs that are available to the public are mind-boggling.

There is, however, order in the chaos of the coding frontier: and furthermore, there is a great deal of dormant potential for any coder starting a new project. Need to write a robust database-backed app in Python? Look no further than SQLAlchemy. How about 3D graphics on the web, without plugins? Try Three.js, SpiderGL, or even my library, 3DFOE. Want your web app to have superb charting abilities? Try Highcharts, Processing.js, or Raphael.js. What I'm getting at is, there's no need to reinvent the wheel; there are a plethora of open-source or low-cost commercial libraries and apps out there, waiting to be used.

What's even more exciting, to me, is the fact that by simply combining existing libraries, the opportunity exists to create something new, something better - something innovative.

A CASE STUDY: 3DFOE, OR 3D FOR OLD IE

To illustrate my point, let's examine my JavaScript 3D graphics library for old versions of Internet Explorer – 3DFOE, or 3D For Old IE. I recognized a need for in-browser 3D graphics that support the vast market share of Microsoft's Internet Explorer, along with the general disdain end-users have for installing browser plugins just to view a web page. I was already familiar with Three.js (an excellent JavaScript 3D graphics library), in particular its 2D Canvas renderer, which works in most modern browsers, including IE9. I had used Google's Explorer Canvas in the past, when working with treemap visualizations in the JavaScript InfoVis Toolkit; Google's relatively small Explorer Canvas library simulates the 2D Canvas tag in versions of Internet Explorer 8 and below.

The idea then came to me: why not combine Three.js and Explorer Canvas? Surely with a bit of "duct tape" code, I could get pure JavaScript-based 3D graphics to render in old versions of IE. So, off I went writing the "duct tape" code, with a bit of help from Modernizr to see if the end-user's browser natively supported the Canvas tag; if it did, Google's Explorer Canvas was bypassed – otherwise, Explorer Canvas was utilized to simulate the Canvas element, allowing Three.js to render 3D graphics on an artificial Canvas tag.

Creativity As Synergistic Synthesis

I believe the above example illustrates how two or more independent projects, when "duct taped" together with some binding code, can produce a product that rises above its components in utility and functionality – what I'm calling synergistic synthesis. There are such rich resources available on the Internet that few coders need to architect and program a large project from scratch; in fact, to do so is often wasteful. We all use frameworks and platforms such as .NET, Java, jQuery, Dojo, and Cocoa on a day-to-day basis. I encourage you, when next tasked with solving a vexing problem, to recall the myriad libraries and tools you've come across. Then, put your keyboard aside, grab a tablet, and really research what's readily available in the open-source world already. Imagine how the various components you come across could be integrated to form a solution, being sure to note any limitations the open-source licenses might carry.

If there are few projects, or combinations of projects, which sufficiently address the programming problem you're tasked with solving, then and only then should you pick up your keyboard and start programming away.

Regularly researching new libraries and frameworks will eventually create an index of technological components in your mind – this will speed the process of imagining what's possible through creative combination. Finally, support the open-source world – because it already supports you! Making some or all of your derivative works open-source will help other programmers who need to solve a problem similar to yours.