The following is an email explaining the billboard loading optimization and noscript handling that was implemented on 6/8/2014.

Okay, all set. Yours again.

I should explain what I did...

Issues I had noticed:

* If scripts were blocked by a browser extension or a network device (like in many workplaces), the noscript flag was not set in the browser and thus #slides repeated the imaged down the page, as seen in closed Git issue [#37](https://github.com/modalexii/Rocketeria/issues/37).
* Fetching jquery.slides.js (fucking huuuge) and js.js in head was seriously dampering visual page load
* Enabling Google's [pagespeed](https://developers.google.com/speed/pagespeed/) js optimizations broke most of my attempts at more dynamic JS loading (e.g., via $('head').append() ), but I really wanted to make use of these optimizations

Initially I tried using getScript() to fetch jquery.slides.js and js.js on $(document).ready(), but this created a visual load weirdness wherein the repeating billboard image appeared for a few moments, until $(document).ready() fired. This was tolerable on broadband but very awkward on slower/mobile network connections.

Additional observations that guided the eventual solution:

* Defining event handlers for billboard arrows (previously in js.js) can be done at any point.
* Initializing #slides config must be done after jquery.slides.js has loaded
* Pagespeed optimizations, if so configured, can combine same-domain src'd js files and dump any sufficiently small js files in to head.
* Each additional GET request (e.g., for src'd js files) kills speed, partly because of browser's same-domain parallel download limits, and partly because most people still have mediocre internet.

So here is the solution that is now in place:

* The unreliable noscript block was removed (you may want to remove references to \*-no-js in rocketeria.css)
* div #slides now has a single billboard image that is identical to the others (no special class or anything). This functions as a noscript block should, only it also works when arbitrary scripts are blocked or even broken
* Further billboard images start off in a script block (#slides-content). If scripting is disabled or otherwise prevented, they never get shown or impact the page appearance in any way.
* The contents of js.js now live with the #slides initialization in head, thereby eliminating a GET request and doing some of pagespeed's work for it
* On $(document).ready(), jquery.slides.js is fetched with getScript(), then the contents of #slides-content are appended to #slides, then the billboard arrow event handlers are defined and #slides is initialized.

So basically the page loads in it's entirety with a single billboard image, then if it can run scripts, it downloads the stuff to run the billboard and kicks it off.

Clear as mud?

KS