

To: Manager

From: Suzanne Becker, Zach Walsh

Subject: eval('Javascript'): A Reflection

In the development of Santorini, we found Javascript to be an exceptionally flexible tool. At all stages of writing a particular component, the lack of types made it very easy to prototype, stub out, and test the new functionality. This made it easy to design properly, writing a skeleton and unit tests before actually writing any code. For example, after writing the Remote Proxy Player, we were at an intermediate stage in writing the server-side code. We wanted to verify its behavior as an independent component before moving forward with writing the infrastructure that would eventually use it. So, we wrote a quick script that demonstrated its functionality, and Javascript made this very easy for us.

On the other hand, this flexibility required a disciplined focus on good design. Javascript's untyped nature made it very easy to be fluid with data definitions and signatures. In particular, any change to a data definition was incredibly easy to do locally, but very difficult to guarantee globally. It was tempting to add an additional value to an enumeration, change the contents of an array, or make any number of other changes without fully understanding the consequences. Tracking down every line of code affected by such changes quickly became a nightmare, because execution could fail silently in any number of ways.

Another issue was the limitation of our environment. Namely, we were stuck using Node 6. This prevented us from using the latest language features, like `async/await` and threading. Not being able to use `async/await` affected code readability, because we were tied to Javascript's event-based paradigm. The lack of threads was also a serious issue, and prevented us from writing proper sandboxes for foreign code running locally.