Start: 2:00 P.M.

Tutorial from last log no longer works with PIXI v3.0

However, the release of v3.0 also included a “bunnymark” demo, which is essentially a stress test of the system via many bouncing bunnies. Linked here: <http://www.goodboydigital.com/pixijs/bunnymark_v3/>

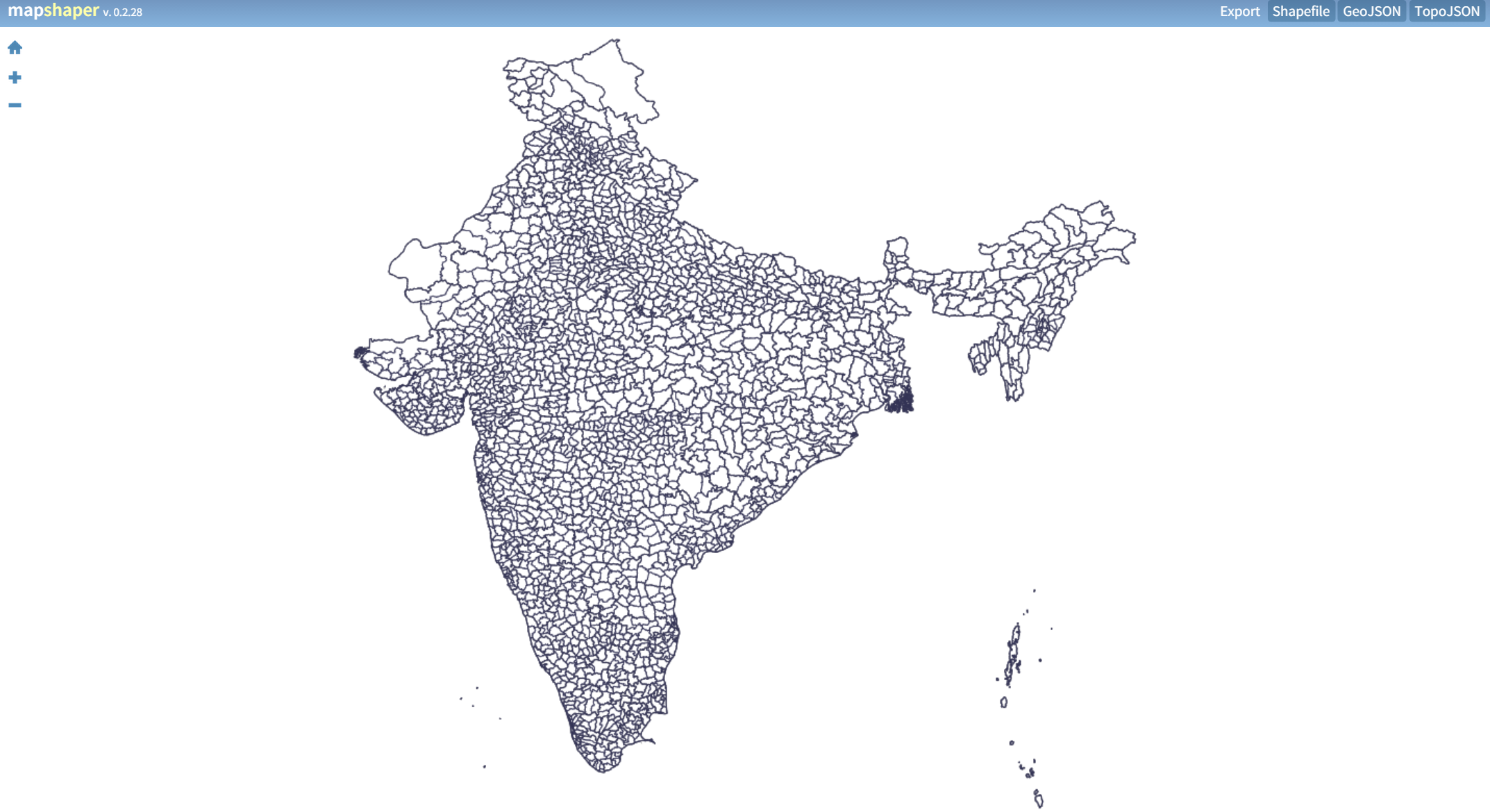
On my system, I can safely reach 36,000 individual bunnies before the framerate falls below 50 fps. Also of note: “**Every single one of those bunnies has its own position/scale/rotation and alpha!”**

**Individually animating the assemblies/constituencies should definitely be possible with PIXI.js.**

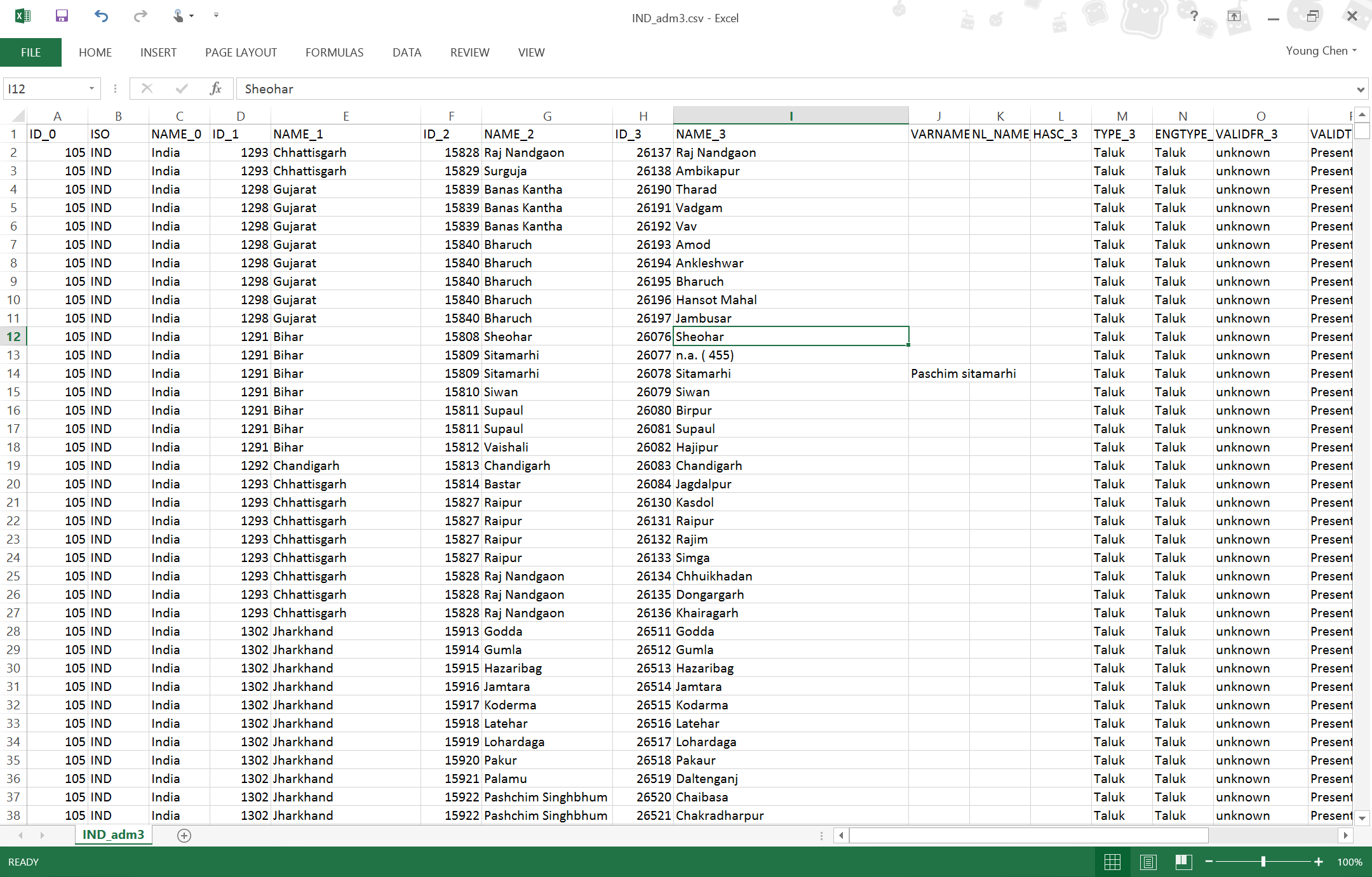
**The majority of the difficulties would be in setting up the resources for PIXI.js to use. They would have to be in bitmap format so that it will be possible to modify individual brightness independently.**

**Looking for a good way to parse the GIS shapefile found from here:** <http://www.diva-gis.org/gdata>

Shapefile viewer here: <http://mapshaper.org/> can be used to view and convert it to GeoJSON, which makes it somewhat more human readable.



DBF converter here: <http://dbfconv.com/>



The best course of action now would be to write some custom shapefile parsing javascript and draw all the constituencies as primitives with the PIXI.js framework. The data file would ideally correspond to the way the shapefile is organized. Each row would correspond to a date, so we won’t have to load the entire file into memory. Each column would correspond to a location, in the same order that the shapefile declared them, so they would have the same corresponding indices (PIXI graphics objects can be nested, but can only be retrieved via index [getChildAt(index)].

Finished: 3:30 P.M.