Luke Miller

Rich Media 330.02

Web Experience

**Media:** The experience uses the basic media needed. There are only three different images used (differently sized circles) to keep loading times down while still providing decent aliasing on the edges of the circles. The web font Lato was used throughout. The experience runs through canvas using PIXI.JS. All other requirements met.

**Interaction:** The user is able to change the strength of gravity, the speed of the simulation, the zoom of the camera, the location of the camera, whether collisions or gravity should be calculated, and finally how many particles should be created on the next restart. Moving is controlled through clicking and dragging on the canvas, all other controls are in the options bar on the side.

**Usability:** The experience is automated, and the controls are extremely easy to grasp and understand, as well as being intuitive. As the experience is an observation, it’s easy to tell whether it’s just started, collapsing into a chaotic nature, or spreading out and ending. There are no keyboard shortcuts for the user to worry about.

**Experience:** The experience is, simply put, a visualization of an N-Body simulation. However, by visualizing a complicated task, it lead to a pleasing result overall. With tests that I’ve done through showing others, the experience seemed to please them and they enjoyed watching just how chaotic things become. The controls are simple and easy to use, making the entire experience much less ‘broken’ through pauses trying to change things. Even though everything is just a circle, watching things grow and change color and collide with each other is oddly satisfying. Overall, even though it seems basic, I believe the experience went beyond what was required.

**Coding:** This project uses one library: PIXI.JS. This is used in order to speed up drawing and offload it to the GPU. For a 2,000 particle system, drawing went from 5-6ms per frame down to 1ms per frame (draw also updates the positions). The experience also goes far beyond the coding taught so far by including a spatial hashing data structure, allowing for an increase in optimization of roughly 50-500 times faster than without. Prior to its inclusion, the calculations for a 2,000 particle system ran at roughly 3-4 million calls of collision checking and gravitation per frame. With the spatial hashing system, it runs at roughly 110,000 call per frame, an improvement of 3600%. This is the section that I went far beyond the requirements on.

**Final Grade:** I believe this project is at least a 95 or above.