Project 1 Documentation

Name: Matthew Camera

Project Name: Dustbox

Project Link: <https://people.rit.edu/mjc9536/330/Project%201/index.html>

A. Overall Theme/Impact

1. I think that my project has a coherent theme, is engaging enough to capture attention for a bare minimum of two minutes (hopefully more), and it goes significantly beyond the topics we have learned in class.
2. Topics I taught myself over the course of this project:
   1. Variable & object access times
      1. While my project is certainly nowhere near as optimized as it could be, I made a conscious effort to research and optimize the containers I use and in what scope.
      2. Draws to the canvas via manipulation of canvas imageData. This was one of the hardest tasks to figure out on the project, but it was vital to draw the particles as pixels.
      3. Simulated physics. This one is never easy, so I’m going to mention it.
3. I definitely would like to include this in my portfolio down the road. This is one of the few times I’ve been very satisfied with the end product of a school project.

B. User experience

1. Anyone who is familiar with falling sand games will immediately recognize how to use it, and anyone who isn’t can quickly understand. The sandbox nature of the app means that much of the learning is done through experimentation, and this extends soundly to the controls.
2. It does have text content, though only because they are required. I’m confident that the app is intuitive to pick up, use, and understand without textual instructions.
3. As far as I’m aware, it catches and handles all errors that may occur within the .js scripts. Knock me for this one if you find one--I’ll eat my words.
4. I can’t speak for anyone else, but I think the visual design is acceptable at worst and pleasing at best. I focused on sticking to colors and color combinations that are easier on the eyes.

C. Media

1. CSS and HTML both passed validation (based on W3C’s validation tools).
2. Built entirely on canvas.
3. Uses canvas.save/canvas.restore when appropriate (drawing the background and sub-border)
4. Uses an embedded font (Poppins)
5. Uses semantic HTML elements

D. Code

1. Naming conventions and coding standards are up to snuff.
2. Event handlers are not inline.
3. Uses ES6 modules.
4. I tried to be very thorough with my commenting (though not overbearingly so). I hope this proves helpful.

What went right:

-Hey, it works! And not too bad, either!

-I learned a whole lot in doing this project, and I even enjoyed it too. Many of the topics mentioned previously are things I don’t think I would have learned until much later in my web studies, if ever at all.

What went wrong:

-I bit off more than I could chew, I think. I overestimated how much we needed to do for this project.

Things I would’ve liked to include:

-More elements. I feel like that’s low-hanging fruit.

-More reactions. They are cumbersome to program, but they are the soul of the game.

-Mobile functionality. I started on this but didn’t get back to it as it is much more complicated than I anticipated.

Known bugs:

-Occasionally, when glass is fused from lightning, one or more glass particles are rendered in the wrong location. Number is usually very small.

Non-course resources:

-Element buttons were based on the buttons from the W3Schools CSS button examples.

-Poppins web font

I feel like I earned around a 94 on this project. There are a few minor issues, but overall I definitely feel that this project exceeded my expectations going in. It took a lot of work to get it to where it is now and I do think that I pushed myself above and beyond the scope of the project requirements to produce the final product. It works almost perfectly, it’s pleasant to look at, fun to fiddle with, and a great way to burn 10-15 minutes.