When to Use Which?

Meet the loveable Canvas and learn when to use it over the DOM.

WE'LL COVER THE FOLLOWING

- The DOM
- The Canvas

Whenever a section heading promises a clear answer, you know that the answer is never clear. This time is no exception. Choosing between the immediate mode-ness of the `canvas` and the retained mode-ness of your DOM is not an exclusive decision. It can actually be quite scandalous! You can choose just one, the other, or even both.

In this section, let's build on the overview you saw in the previous section and look at the advantages and disadvantages of each of the approaches.

The DOM

Since we are talking about the DOM here, you will spend a majority of the time, possibly even all of your time, in this retained mode world. Despite the comforts it provides, it isn't perfect.

Let's look at its perfections and imperfections in more detail:

1. **Easy to use.** The DOM abstracts away a lot of the details that would otherwise get you bogged down. Examples of details that can bog down even the best of us include layout, event handling, clean up, selection/highlighting, accessibility, being DPI friendly, and so on.
2. **Redrawing is handled for you.** You only specify what you want to display on the screen. The details of how to do that and how often to refresh are all left to the Graphics API to handle.
3. **CSS! CSS! CSS!** You can easily modify the visuals of your DOM elements

using CSS.

4. **Animations are easy to define and modify.** Because of the CSS support, you can easily define animations or transitions, specify an easing function, make a few other tweaks, and you are good to go. This applies to JavaScript-based animations as well. If you are using JavaScript to animate an element's properties, you just have to get your `requestAnimationFrame` loop setup to update the property values. Everything else is taken care off...such as when to redraw or how to maintain a smooth frame rate.
5. **Memory intensive.** You know all of the details that get taken care of for you when using a DOM element? Well, that care doesn't come cheap. Your DOM elements are very complex little things, and all of this complexity takes up space in your browser's memory. The more elements you are dealing with, the more resource hungry it all gets.
6. **Less control over how things get drawn.** For certain graphics-related tasks, the default rendering may be a bit limiting. Browsers optimize for their particular needs, and those optimizations may go counter to what you want to do.

The Canvas

If this were a popularity contest, I would feel pretty bad for immediate mode and the `canvas` element that uses it. Fortunately, it isn't! Immediate mode systems certainly carry their own weight - even in the more limited cases they are used in.

Let's look at some of their cool (and less cool) features in more detail:

1. **Fast. Really fast.** Because an immediate mode system doesn't maintain its own model, your code is all that stands between you and the browser redrawing. The many layers of abstraction that slow operations down simply do not exist in the immediate mode world.
2. **You have a lot of flexibility.** Since your code controls all aspects of when and how something is drawn to the screen, you can tweak and customize the output any way you would like.
3. **Great for dealing with many elements.** Compared to a retained mode system where every little addition to your scene takes up extra memory,

immediate mode systems don’t have that problem. Generally, an immediate mode system will always use less memory than a retained mode system - something that becomes more noticeable as you add more and more elements into the mix.

4. **It can be slow when drawing to large areas.** How quickly a redraw completes is proportional to the number of pixels you are re-drawing. If your addressing a really large area, things could get slower if you are not careful and do not optimize appropriately.
5. **It is complex.** Because you are handling more of what it takes to get something to display on the screen, there are a lot more details for you to keep track of. Getting up to speed with the various draw commands and how they are used is no picnic either.