Profiling Long Paint Times with DevTools' Continuous Painting Mode

By Paul Irish

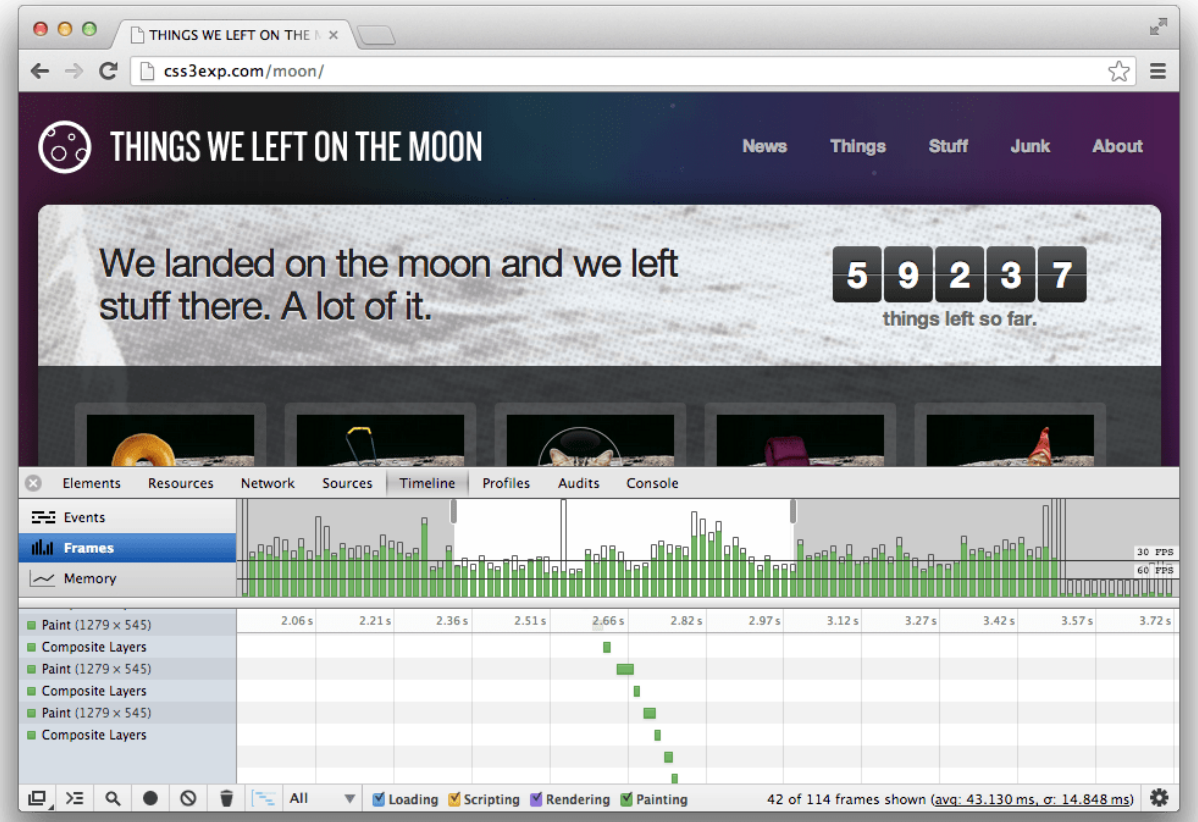
Paul is a contributor to Web**Fundamentals**

Continuous painting mode for paint profiling is now available in Chrome Canary. This article explains how you identify a problem in page painting time and how you can use this new tool to detect bottlenecks in painting performance.

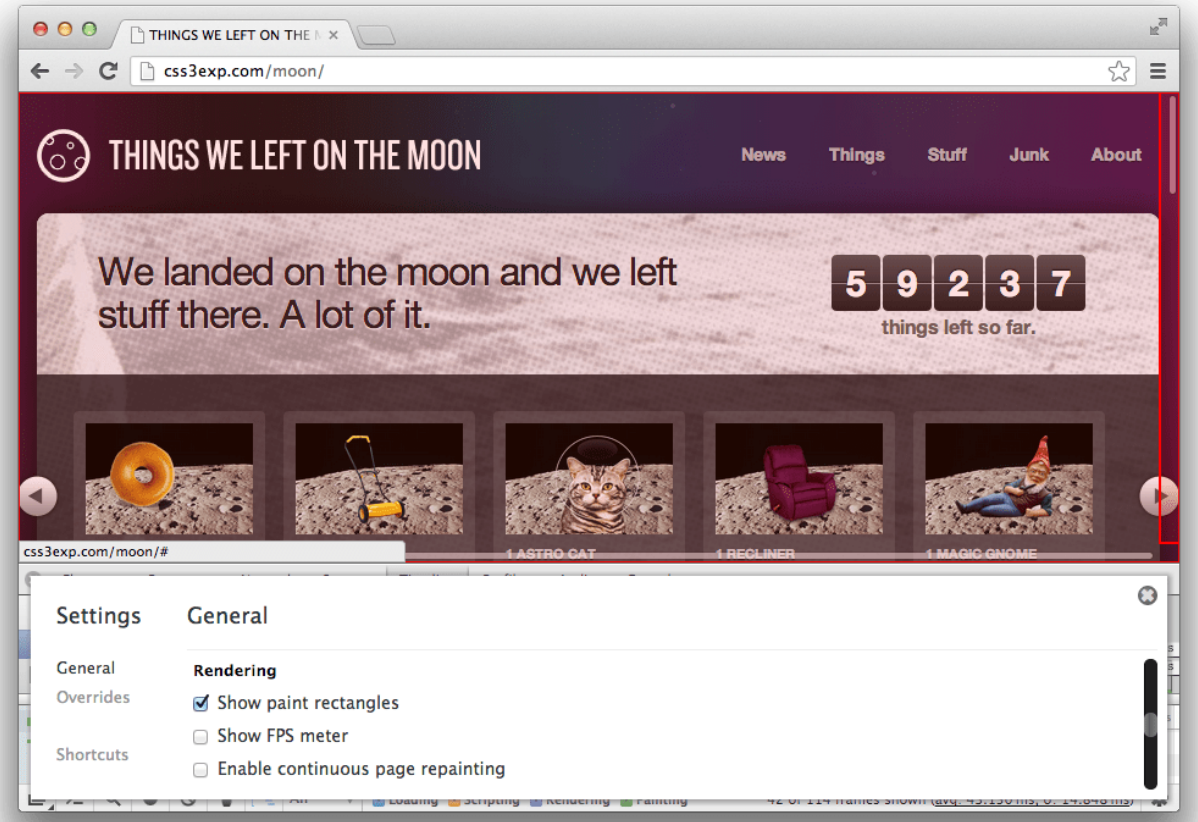
Investigating painting time on your page

So you noticed that your page doesn't scroll smoothly. This is how you would start tackling the problem. For our example, we'll use the demo page Things We Left On The Moon by Dan Cederholm as our example.

You open the Web Inspector, start a Timeline recording and scroll your page up and down. Then you look at the vertical timelines, that show you what happened in each frame.



If you see that most time is spent painting (big green bars above 60fps), you need to take a closer look at why this is happening. To investigate your paints, use the **Show paint rectangles** setting of the Web Inspector (cog icon in the bottom right corner of the Web Inspector). This will show you the regions where Chrome paints.



There are different reasons for Chrome to repaint areas of the page:

- DOM nodes get changed in JavaScript, which causes Chrome to recalculate the layout of the page.
- Animations are playing that get updated in a frame-based cycle.
- User interaction, like hovering, causes style changes on certain elements.
- Any other operation that causes the page layout to change.

As a developer you need to be aware of the repaints happening on your page. Looking at the paint rectangles is a great way of doing that. In the example screenshot above you can see that the whole screen is covered in a big paint rectangle. This means the whole screen is repainted as you scroll, which is not good. In this specific case this is caused by the CSS style `background-attachment:fixed` which causes the background image of the page to stay at the same position while the content of the page moves on top of it as you scroll.

If you identify that the repaints cover a big area and/or take a long time, you have two options:

1. You can try to change the page layout to reduce the amount of painting. If possible Chrome paints the visible page only once and adds parts that have not been visible as

you scroll down. However, there are cases when Chrome needs to repaint certain areas. For example the CSS rule `position:fixed`, which is often used for navigation elements that stay in the same position, can cause these repaints.

2. If you want to keep your page layout, you can try to reduce the painting cost of the areas that get repainted. Not every CSS style has the same painting cost, some have little impact, others a lot. Figuring out the painting costs of certain styles can be a lot of work. You can do this by toggling styles in the Elements panel and looking at the difference in the Timeline recording, which means switching between panels and doing lots of recordings. This is where **continuous painting mode** comes into play.

Continuous painting mode

Continuous painting mode is a tool that helps you identify which elements are costly on the page. It puts the page into an always repainting state, showing a counter of how much painting work is happening. Then, you can hide elements and mutate styles, watching the counter, in order to figure out what is slow.

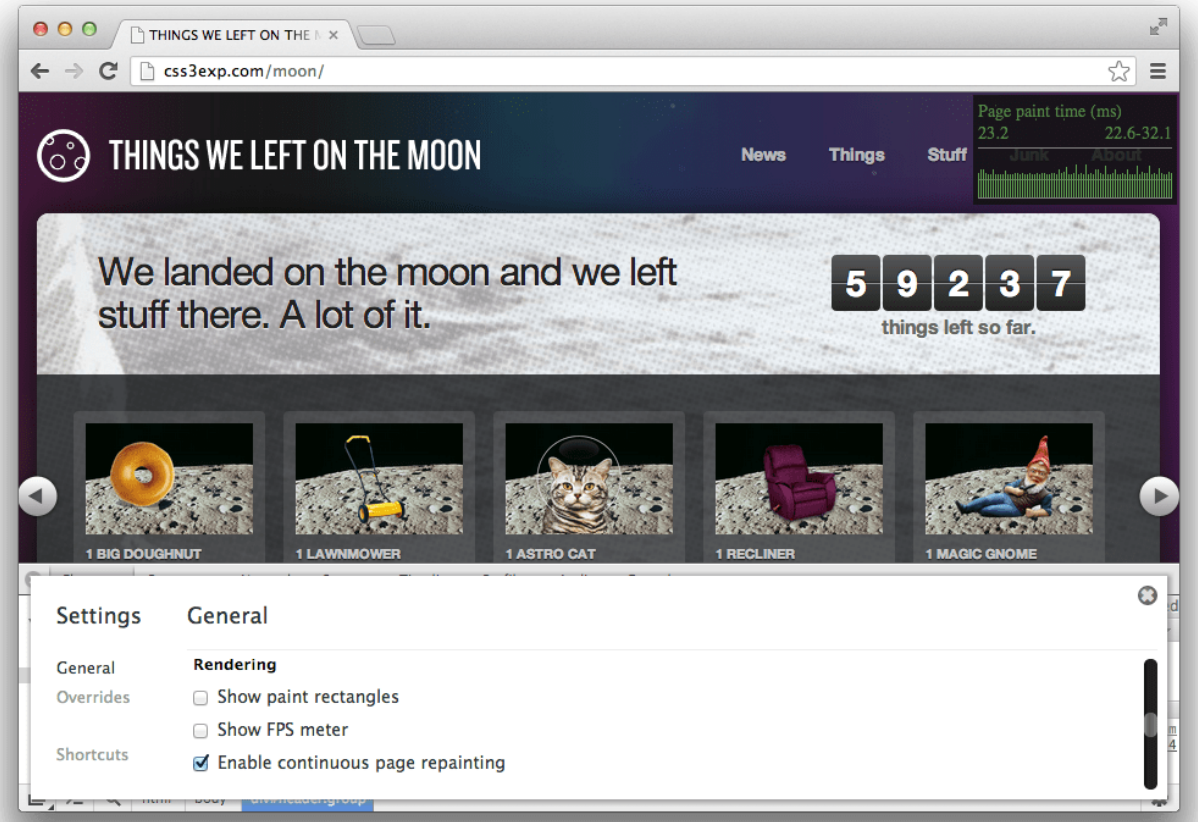
Setup

In order to use **continuous painting mode** you need to use Chrome Canary.

On **Linux** systems (and some Macs) you need to make sure that Chrome runs in compositing mode. This can be permanently enabled using the **GPU compositing on all pages** setting in `about:flags`.

How To Begin

Continuous painting mode can be enabled via the checkbox **Enable continuous page repainting** in the Web Inspector's settings (cog icon in the bottom right corner of the Web Inspector).



The small display in the top right corner shows you the measured paint times in milliseconds. More specifically it shows:

- The last measured paint time on the left.
- The minimum and maximum of the current graph on the right.
- A bar chart displaying the history of the last 80 frames on the bottom (the line in the chart indicates 16ms as a reference point).

The paint time measurements are dependent on screen resolution, window size and the hardware Chrome is running on. Be aware that these things are likely to be different for your users.

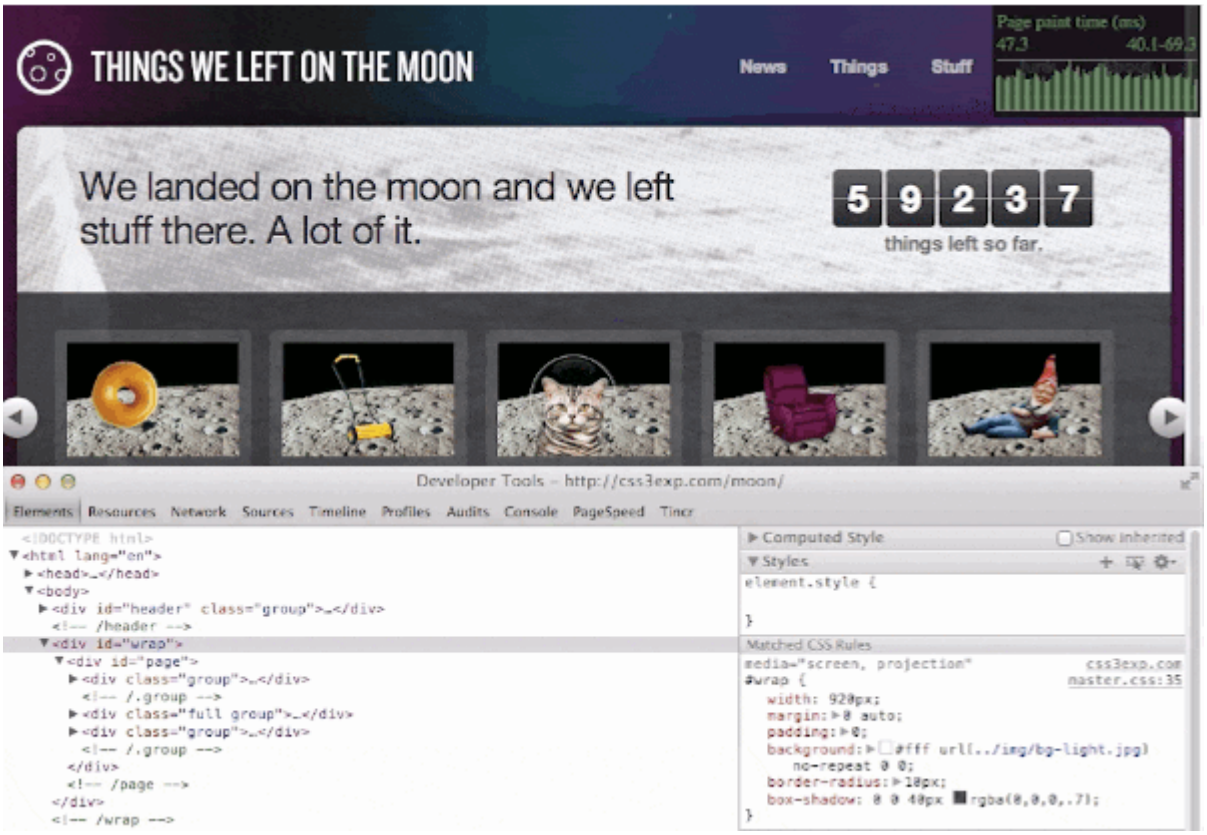
Workflow

This is how you can use **continuous painting mode** to track down elements and styles that add a lot of painting cost:

1. Open the Web Inspector's settings and check **Enable continuous page repainting**.

2. Go to the Elements panel and traverse the DOM tree with the arrow keys or by picking elements on the page.
3. Use the **H keyboard shortcut**, a newly introduced helper, to toggle visibility on an element.
4. Look at the paint time graph and try to spot an element that adds a lot of painting time.
5. Go through the CSS styles of that element, toggling them on and off while looking at the graph, to find the style that causes the slow down.
6. Change this style and do another Timeline recording to check if this made your page perform better.

The animation below shows toggling styles and its affect on paint time:



This example demonstrates how turning either one of the CSS styles `box-shadow` or `border-radius` off, reduces the painting time by a big amount. Using both `box-shadow` and`border-radius` on an element leads to very expensive painting operations, because Chrome can't optimize for this. So if you have an element that gets a lot of repaints, like in the example, you should avoid this combination.

Notes

Continuous painting mode repaints the whole visible page. This is usually not the case when browsing a web page. Scrolling usually only paints the parts that haven't been visible before. And for other changes on the page, only the smallest possible area is repainted. So check with another Timeline recording if your style improvements actually had an impact on the paint times of your page.

When using **continuous painting mode** you might discover that e.g. the CSS styles `border-radius` and `box-shadow` add a lot of painting time. It is not discouraged to use those features in general, they are awesome and we are happy they are finally here. But it's important to know when and where to use them. Avoid using them in areas with lots of repaints and avoid overusing them in general.

Learn more about painting and related topics on jankfree.com

Eberhard Gräther is student of MultiMediaTechnology at Salzburg University of Applied Sciences. He interned in the Chrome GPU team from 10/2012 to 03/2013 where he worked on rendering profiling tools. Follow him at @egraether or visit his site to find graphics demos and web games he has built.

Live Demo

Click below for a demo where Paul Irish uses continuous painting to identify a uniquely expensive paint operation.

Last updated July 2, 2018.