Hungry Chrome

Hacking The Browser by Cory Forsyth

ITP, Spring 2018

Student: Assel Dmitriyeva

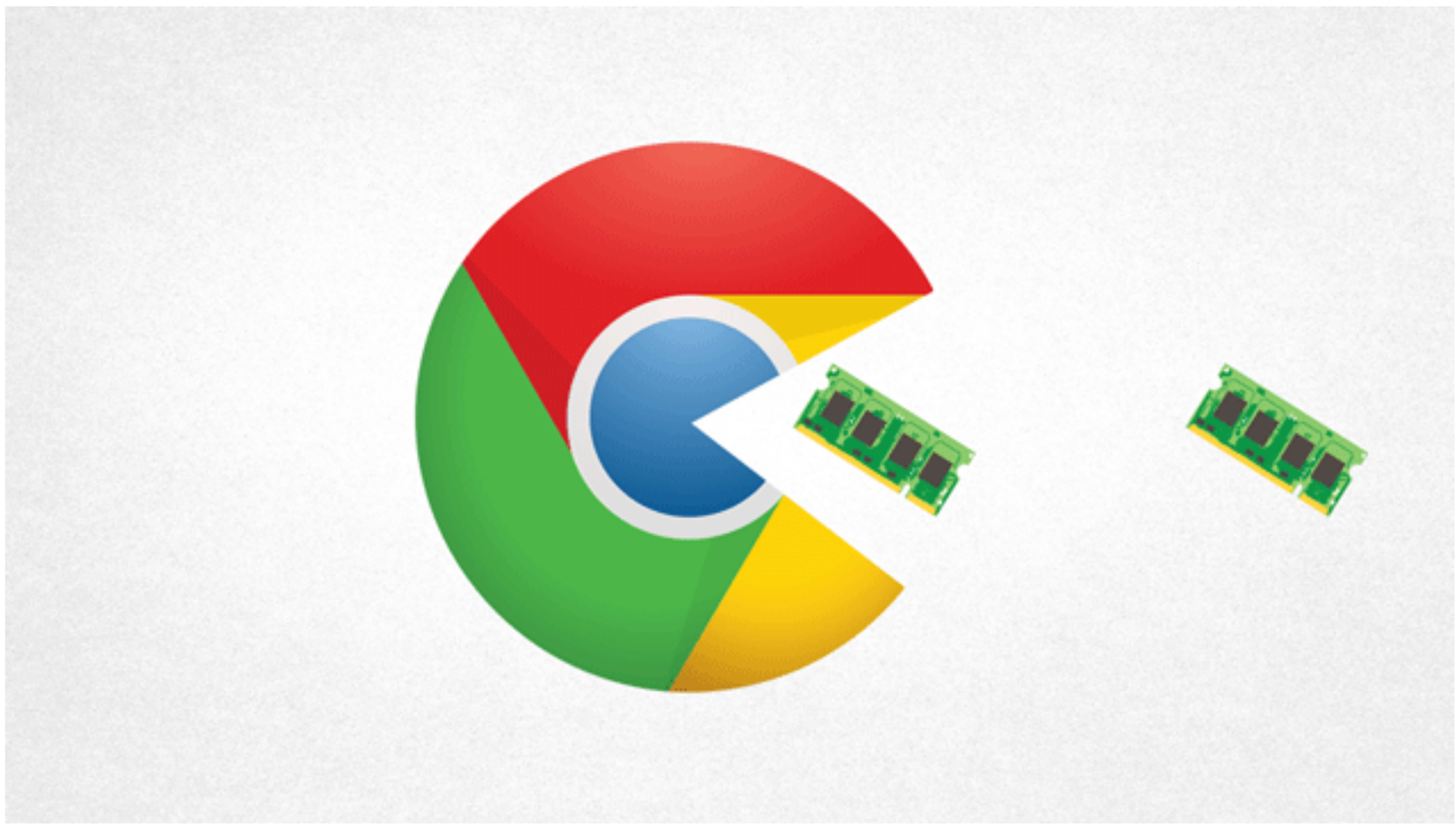
Why Does Chrome Use So Much RAM?
Is Chrome's RAM-hungry reputation is well deserved?

We can refer to **Google Chrome Task Manager:**

The Google Chrome browser has a built-in Task Manager for users to see how much memory and CPU web pages, extensions, and Google processes are using while Chrome is running.
This feature is very useful if:
- *Chrome has become slow*
- *is using too much memory*
- *or freezes a lot*

Could we get the system resources consumed per browser tab?
Is there programmatic access to Task Manager data / chrome.processes API?

Wondering why your browsing experience got interrupted? Let’s make action!

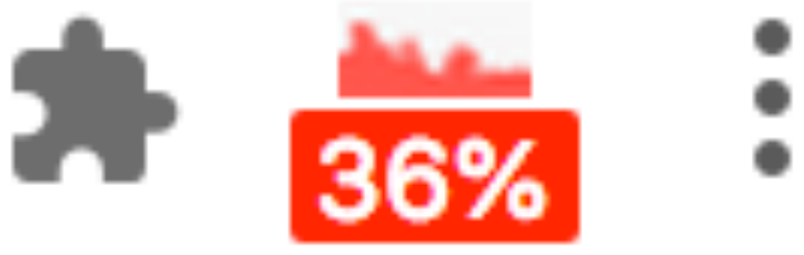


Now you can monitor Chrome's CPU and memory usage, identify and kill expensive processes.

Hungry Chrome

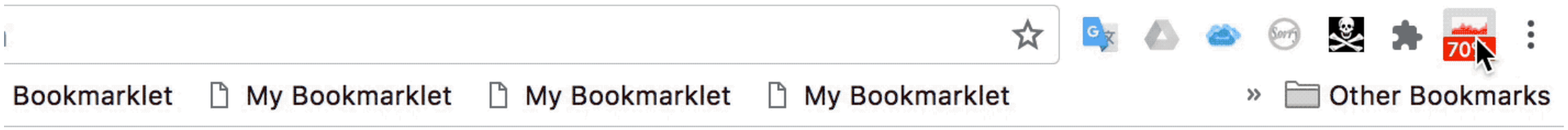
extension makes it easy to identify which tabs or processes are misbehaving, and terminate them to make your life bright again!

Deliverables



Current CPU usage and animated graph are in a always available in a small icon in your address bar. You can easily detect when something goes wrong with your memory.

A detailed popup with list of all processes sorted by CPU or memory usage, historic CPU graphs, and actions to reload or shut down misbehaving processes:



of text that is layered over the icon. Badges make it easy to update the

n about the state of the extension.

haracters or less.

n.setBadgeText and **browserAction.setBadgeBackgroundColor**,

en the user clicks the icon. The popup can contain any HTML contents that

. file with the popup's contents. Specify the HTML file in the **default_popup**

wserAction.setPopup method.

Installation

- Go to chrome://flags and enable Experimental Extension APIs
- Relaunch Chrome
- Download the repo and unzip.
- Go to chrome://extensions
- Ensure the "Developer mode" is ticked, and click "Load unpacked”
- Select the folder you've just unzipped
- And that’s it! Now you can detect hungry tabs and extensions in your Chrome!

Further steps:

- Javascript memory allocated
- Javascript memory used
- Add actions: reload, close the tab, i.e chrome.tabs.reload
- Once chrome.processes API will make it into the Release channel, upload to Chrome Web store