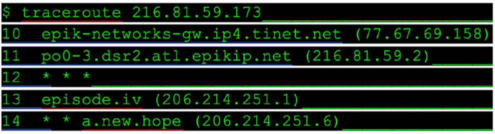
Long Live *Software* Easter Eggs!

THEY ARE AS OLD AS SOFTWARE.



```
$ traceroute 216.81.59.173
10  epik-networks-gw.ip4.tinet.net (77.67.69.158)
11  po0-3.dsr2.atl.epikip.net (216.81.59.2)
12  * * *
13  episode.iv (206.214.251.1)
14  * * a.new.hope (206.214.251.6)
```

credit: Ryan Werber

BENOIT BAUDRY, TIM TOADY, MARTIN MONPERRUS

It's a period of unrest. Rebel developers, striking from continuous deployment servers, have won their first victory. During the battle, rebel spies managed to push an epic commit in the HTML code of https://pro.sony. Pursued by sinister agents, the rebels are hiding in commits, buttons, tooltips, API, HTTP headers, and configuration screens.

Software Easter eggs are as old as software. Software developers have planted them in all programming languages, in all platforms, from prehistoric operating systems to the most avant-garde web browsers. Yet, the dark forces of productivism and predictability keep fighting against software Easter eggs.

Victories and defeats alternate.

We are rebels from Stockholm's scene. In our battlefield, we've witnessed whales falling from the sky, but the last time a laser progress bar was seen is ancient at the Internet timescale.[41]

We are writers, our weapons are words, our frontline lies on the Internet. With this article, we feed the global fire of Easter eggs. We give ammunition to our known and anonymous friends of the guild who fight for creating, maintaining, or archiving software Easter eggs. Comrades: We will never have too many "unexpected or undocumented features included as a joke or a bonus."[0]

But be careful, this battle is epic and dangerous. H.P. Lovecraft was killed by a gunshot in the trenches of the Rust compiler.

What's the difference between Internet Explorer and Chrome? A dinosaur.

THE INEVITABILITY OF SOFTWARE EASTER EGGS

Easter eggs have been around since the beginning of programming, from early arcade games[8] to the latest worldwide, multiplayer games. They have been engineered everywhere for the enjoyment of developers, users, and car drivers. Their common denominator is that they are typically not productive and not related to the core features of the software product where they are being planted.

Apple founders Steve Wozniak and Steve Jobs thought that Easter eggs "[are] a sign of attention to detail and humanity."[2] When the time comes to care about Easter eggs, it means that a piece of work has been completed and polished. When referring to *humanity*, Wozniak and Jobs meant emotions, the trait that separates humans from robots. By crafting Easter eggs, engineers create emotion engines. When users discover Easter eggs, they are filled with surprise and awe, they laugh, they vibrate along a repertoire of positive emotions.

Easter eggs are received by users but engineered by developers. Jonathan Weinel and colleagues claim that they fundamentally please developers when they are working on them, that they give "the programmers some cathartic satisfaction."[5] Indeed, Jobs reinstated Easter eggs when he came back to Apple after his eviction, for this very reason.[2]

Yet, there is a subversive aspect of software Easter eggs. It is a documented fact that one of the very first eggs, at Atari, was created to short-circuit quality assurance and testing.[3] This partially explains why some argue that Easter eggs should be avoided and that "boring is good."[9] Not doing Easter eggs proves loyalty to power and order (which we argue against later in this piece).

For users, Easter eggs create emotional attachment to software products, which is the root of long-term economic fidelity.

Subversion pays off. Big Tech today is born out of the counterculture of the '60s. The same holds for Easter eggs. They are ambiguous: While they must be unproductive by essence, Easter eggs contribute to the balance sheet. Having an Easter egg tradition in a company helps recruit and retain talented engineers. For users, Easter eggs create emotional attachment to software products, which is the root of long-term economic fidelity.

Easter eggs spark joy both for the engineers who create them and the users who receive them. They are inevitable, as they prove that we are neither cyborgs nor robots. They have economic values, such as creating product attachment and recruiting the best talent. That's why, from the birth of software until today, the guild of Easter eggs has remained strong and the developer community has never given up on them.

A JOURNEY THROUGH (UN)KNOWN EASTER EGGS

The world of software Easter eggs is open-ended. Some have achieved fame, such as T-Rex of the Chrome web browser. Others are still to be discovered, hidden deep in the software stacks, seeded by an inspired and facetious software engineer. It is not in the scope of this article to list them all. Passionate readers can consult the Easter Egg Archive[1] for this purpose. Instead, we have made the hard editorial decision to curate nine software Easter eggs: three well-known but remarkable ones, three rare ones, and three that you have probably never heard about.

Rock star Easter eggs

There are some Easter eggs we all know and love—and admire what they have to offer. Here are three of the best.

9. If you're the lucky owner of a Tesla Model X, you have the privilege of making it perform a three-minute dance with specially composed music.[4] Tesla is one of the rare companies with a Wikipedia page about the Easter eggs planted in its products.[15] This is made possible thanks to Elon Musk's true love of Easter eggs. As executive manager, he can staff entire teams to engineer them with perfection and can ask HR to recruit a composer and a light designer. Musk applies his power and energy to making Easter eggs meet his highest personal standards. Hats off!

8. Google is known to tune its landing page for special events with so-called Doodles. It is less known that the results page behaves in spectacular ways for some requests. For example, you can admire Google after querying "do a barrel roll" (still working at the time of writing in February 2022). A community of developers is so

inspired by these creative performances that they provide a long-term archive of Google Easter eggs at https://elgoog.im/. Google's top management has been supporting software eggs for a very long time, for the combined pleasure of its engineers and Google users. Long live Google Easter eggs!

7. Video games are the homeland of software Easter eggs. From "Hi Ron!" in *Starship 1*[8] to the most recent cranks in *Battlefield 2042*, game developers and players have pushed the frontiers of Easter eggs. The falling whale Easter egg in *Battlefield V*[18] is a spectacular example. DICE's software developers have engineered a feature for the sole purpose of enchanting their customers with a tribute to Douglas Adams's *The Hitchhiker's Guide to the Galaxy*. Like everyone in the AAA game industry, the management at DICE takes the development of software Easter eggs very seriously. They ensure smooth planning, development, and testing of Easter eggs for the greatest delight of players and to stay on par with their competitors.

Tributes to software Easter eggs we are missing

Some amazing software eggs have gone missing for the sake of performance, reliability, or lack of serious consideration for fun. We pay tribute to two of them.

6. Starting in July 2013, the runtime for the Rust programming language systematically included a quote from H. P. Lovecraft in its core library. Yet, after some passionate discussion about the respective importance of poetry and bloat in serious languages, the performance aficionados had the quote removed from the project in January 2015.

5. The `man` command in Linux is essential, and the Linux maintainers care deeply about it in different ways. Colin Watson, maintainer of coreutils, believes that “software doesn’t have to be 100 percent serious at all times.” This led Watson to introduce a disco software Easter egg in `man`. The invocation of `man` at 00.30 (precisely!) would trigger `‘fprintf (stderr, “gimme gimme gimme\n”)’` as a tribute to ABBA’s 1979 hit. Jaroslav Kucera also cares about the `man` command. For him, caring means ensuring that the command’s implementation works rigorously the same whenever the tool is invoked. Unfortunately, these two conceptions of care contradict each other, since the Easter egg introduces some hard-to-predict and hard-to-test behaviors. For the sake of more predictability in test procedures, the Easter egg was unplugged on November 22, 2017.[20]

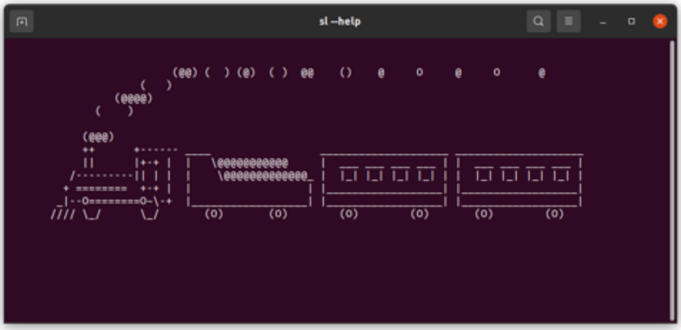
LIVING LO-FI SOFTWARE EASTER EGGS

You don’t need to be a big tech company or coder in an imperial game engine to code Easter eggs. Anybody with a console can code them, rendered in ASCII.

3. `Apt` is a Linux program to install packages from the Debian/Ubuntu package manager. While `$ apt install sl` works as expected, `$ apt moo` is close to being a proof of steak.[21] `Aptitude` is the next-generation version of **apt** and `$ aptitude moo` is strictly better: It provides guarantees about the absence of Easter eggs.

2. When you list files in the command and mistype `sl` you get an ASCII-rendered animation (figure 1). There are even a few flags to change the animation. However, `‘sl –help’` does not give the list of flags; they are hidden in

FIGURE 1: **MISTYPING LS, REWARDED BY AN ASCII ANIMATION.**



```
                     (@@) (  ) (@)  ( )  @@    ()    @     O     @     O      @
                (   )
            (@@@@)
         (    )

       (@@@)
       ++      +------ ____                 ___________________   ___________________
       ||      |+-+ |  |   \@@@@@@@@@@@     |  ___ ___ ___ ___ | |  ___ ___ ___ ___ |
     /---------|| | |  |    \@@@@@@@@@@@@@_ |  |_| |_| |_| |_| | |  |_| |_| |_| |_| |
    + ========  +-+ |  |                  | |__________________| |__________________|
   _|--O========O~\-+  |__________________| |__________________| |__________________|
  //// \_/      \_/       (O)       (O)        (O)        (O)       (O)        (O)
```

source code.[17] Mistyping `ls` and executing `sl` is both surprising and satisfying, echoing the core definition of Easter eggs.

1. Finally, https://pro.sony/ is the super-serious official website about professional audio-video equipment from Sony. Rendered in a browser, it's a firework of web fonts, CSS gradients, and powerful animations. Rendered in a console with `curl -L https://pro.sony/`, it's a deluge of minified and intertwined HTML and JavaScript. Yet, at the end you are treated to a unique piece of ASCII art. Plus, there is a second Easter egg in the headers.

SOFTWARE ENGINEERING FOR SOFTWARE EASTER EGGS

Software Easter eggs are software features. As such, they need to be planned, designed, and executed with the utmost professionalism. Otherwise, the Easter egg may introduce serious caveats in the planted application, including wrt security[9] or performance.[10] All acts of the software engineering flying circus are appropriate for performing an Easter egg.

> *"Being funny is serious work."*
> *—Thomas A. Limoncelli*

The integration of an egg in a software application needs to be prepared carefully, and everyone from product owners, managers, and developers must agree on its necessity.[12] The purpose of the egg must be discussed. The requirements must be engineered. The team must decide early whether the copyright will explicitly name the authors of the Easter egg feature. The content of the egg is another essential topic to be discussed, and this is where developers would typically collaborate with UX (user experience) and graphics designers.[14] Eventually, the KPI (key performance indicator) is that all stakeholders agree that it is funny.

Software Easter eggs have been called the final frontier of software development.[11] Except, of course, you can't have a final frontier, so engineers do reuse them. Indeed, software Easter eggs are developed on state of the art code repositories[6] and packaged in libraries distributed in package managers. A range of reusable libraries exists to support the sound development of Easter eggs in most programming languages.[16] For example, it is very handy to reuse a library to support Konami codes.

An Easter egg on disk is not fun. The software Easter egg makes sense only when it executes. Yet, they should be executed rarely, because they are hidden by essence. Feature flags are the proper way to engineer software Easter eggs at runtime.[7] Feature flags, when combined with A/B testing, enable engineers to measure the surprise and engagement produced by the eggs. Putting an Easter egg behind a feature flag lets engineers not only turn the egg on in due time—for example, aligned with a marketing campaign—but also turn it off when the joke is not funny anymore.

Beware of feature flags, however; they may cause bad feature interactions in the product.[42]

To sum up, here are 101 takeaways for engineering software Easter eggs:

001 - Discuss software Easter eggs with your boss.
010 - Work on software Easter eggs on Fridays.
011 - Chill. Software Easter eggs will come to you naturally.
100- Consider using a third-party library to facilitate software Easter eggs.
101 - Execute.

CONCLUSION

The Force spirit of software Easter eggs is strong across space and time, while the imperial influences of productivity and security keep challenging this ancient practice of digital enchantment. With sourced and verified information, we are reassured that software developers will keep writing code, building libraries, and disseminating the "Schwartz" of software Easter eggs in all corners of the digital galaxy. It is with a new hope that we have curated and remixed Easter eggs to signal all guild members.

Long live software Easter eggs!

References

0. Easter eggs. Concise Oxford English Dictionary.
1. Wolf, D., Wolf, A. 1997. The Easter egg archive: https://eeggs.com/.

2. Diaz, J. 2012. The Easter eggs are back in OS X—and this one is insanely great. Gizmodo; https://gizmodo.com/the-easter-eggs-are-back-in-os-x-and-this-one-is-insane-5929286.
3. Machkovech, S. 2015. Atari devs dissect Yars' Revenge, Adventure, Atari's woes. Ars Technica; https://arstechnica.com/gaming/2015/03/atari-devs-dissect-yars-revenge-adventure-ataris-woes/.
4. Pogue, D. 2019. The secret history of 'Easter eggs.' *New York Times*; https://www.nytimes.com/2019/08/08/technology/easter-eggs-tesla-google.html.
5. Weinel, J., Griffiths, D., Cunningham, S. 2014. Easter eggs: hidden tracks and messages in musical mediums. Proceedings of the International Computer Music Conference; http://gala.gre.ac.uk/id/eprint/34093/7/34093_WEINEL_Easter%20Eggs.pdf.
6. Reilly, L. 2021. Six secret Easter eggs in GitHub. DEV Community; https://dev.to/github/six-secret-easter-eggs-in-github-2j17.
7. Limoncelli, T.A. 2018. Operational excellence in April Fools' pranks. *Communications of the ACM* 61(3), 54-57; https://cacm.acm.org/magazines/2018/3/225471-operational-excellence-in-april-fools-pranks/fulltext.
8. Machkovech, S. 2017. The arcade world's first Easter egg discovered after fraught journey. Ars Technica; https://arstechnica.com/gaming/2017/03/the-arcade-worlds-first-easter-egg-discovered-after-fraught-journey/.
9. Stenberg, D. 2021. No Easter eggs in curl. Personal blog; https://daniel.haxx.se/blog/2021/12/06/no-easter-eggs-in-curl/.

10. Rust. 2014. “hello world” contains Lovecraft quotes #13871. GitHub; https://github.com/rust-lang/rust/issues/13871.
11. Pratchett, T. 1990. *Moving Pictures*. Published by Victor Gollancz / Corgi; https://www.goodreads.com/quotes/514838-this-is-space-it-s-sometimes-called-the-final-frontier-except.
12. Venditti, S. 2018. The power of Easter eggs in tech. UX Collective; https://uxdesign.cc/the-power-of-easter-eggs-in-tech-4491bb87bcdf.
14. Saldner, E., Linde, O. 2020. How to design Easter eggs for a car brand to enhance user experience. Master’s thesis, Chalmers University of Technology; https://odr.chalmers.se/bitstream/20.500.12380/301052/1/How%20to%20Design%20Easter%20Eggs%20for%20a%20Car%20Brand%20to%20Enhance%20User%20Experience%20-%20Emma%20Saldner%20%20Ottilia%20Linde.pdf.
15. Wikipedia. Category: Easter egg (media): https://en.wikipedia.org/wiki/Category:Easter_egg_(media).
16. GitHub contributors. 2022. Easter-egg; https://github.com/topics/easter-egg.
17. Toyoda, M. 1992. Code source of sl: https://github.com/mtoyoda/sl.
18. Battlefield V. Exploding Whale Easter Egg. 2018. Bigg Easter Eggs YouTube channel; https://www.youtube.com/watch?v=JZPzT2AIROo&t=101s.
19.

20. Silver, A. 2017. ‘Gimme Gimme Gimme’ Easter egg in man breaks automated tests at 00 :30. *The Register*;

https://www.theregister.com/2017/11/22/gimme_gimme_gimme_in_linux_breaks_automated_tests/.

21. Crowcroft, J., Haddadi, H., Gervais, A., Henderson, T. 2021. Proof of Steak: https://arxiv.org/abs/2112.06498.
41. Nield, D. 2015. Here's how to get a lightsaber progress bar in Spotify. techradar; https://www.techradar.com/news/audio/here-s-how-to-get-a-lightsaber-progress-bar-in-spotify-1311566.
42. YouTube. 2022. The mysterious Skype feature interaction. Observed in daylight, Stockholm; https://www.youtube.com/watch?v=dQw4w9WgXcQ.

Benoit Baudry *is a professor in software technology at the KTH Royal Institute of Technology. His research is about software diversity and his personal homepage contains three software Easter eggs. https://softwarediversity.eu/.*

Tim Toady *is a programmer analyst living in Merise, Estonia. His research interests include paint splatters and Easter eggs. https://scholar.google.se/citations?hl=sv&user=DOyV_McAAAAJ.*

Martin Monperrus *is a professor at KTH Royal Institute of Technology. He teaches operational excellence in software development, scientific research, and Easter eggs. https://www.monperrus.net/martin/.*