



## matrix (web-based green code rain, made with love)

**News Update September 2022:** this project was featured in Vice Motherboard, along with insight into the effect from Lilly Wachowski.

### Quick Links

- Classic Matrix code
- Starting from a blank screen (`skipIntro=false`) (which some people really like, but isn't the default mode)
- 3D mode
- Mirror mode, with camera and without. (Click to make ripples.)
- Matrix Resurrections updated code
- Trinity mode
- Operator Matrix code (with ripple effects)
- Megacity Mode, as seen in Revolutions

### Variants

- Code of the "Nightmare Matrix"

- 
- (you know, this stuff).
  - Code of the “Paradise Matrix”
    - (AKA this stuff).
  - A custom variant I call “Palimpsest”
  - A custom variant I call “Twilight”
  - Morpheus mode
  - Bugs mode
  - Pride flag colors
  - Trans flag colors
  - Custom stripes (`effect=stripes&stripeColors=R,G,B,R,G,B,R,G,B, etc`)
  - Custom palette (`palette=R,G,B,%,R,G,B,%,R,G,B,%, etc`)
  - Custom image (`url=www.website.com/picture.jpg`)
  - Debug view (`effect=none`) (*epilepsy warning*: this once had lots of flickering)
  - Holographic version (requires a Looking Glass display; see it in action [here](#))

### Typography

- The free classic font (TrueType).
- The free *Resurrections* font (TrueType).
- The unofficial glyph database.

*Concept* -  $(t,i,x,y) \Rightarrow (1-(\sin(x*7)-(y/15)+(t+2))\%1)/3$  - Educational version in Scratch

### Contents

- About
- Goals
- Sidenote: other people’s Matrix effects
- Customization
- Troubleshooting
- Future directions
- Friends of the project
- Colophon
- Other details

---

## About

This project is a web implementation of the raining green code seen in the *Matrix* franchise. It's built right on top of the functional WebGL wrapper, REGL, with beta support for the upcoming graphics API WebGPU; its previous Three.js version is maintained in a separate branch.

This project runs right in the web browser; you can serve it with any HTTP/HTTPS server, with no additional setup. For example, on a Mac you can point a Terminal shell at a local copy of the project and run a simple HTTP server with Python: `cd /path/to/the/project ; python3 -m http.server`

## Goals

There are four kinds of Matrix effects people call “digital rain”: 1. The green symbols that “rain down” operators’ screens endlessly 2. Scenes from within the simulation that depict green symbols streaking across everything 3. The films’ opening title graphics, which dazzle viewers and then draw them into the world of the franchise 4. The “dialing” visualization at the opening of *The Matrix* and *Resurrections*

A motivated fan can attempt to portray any of these. However, this project focuses specifically on #1 and #3— an endless effect, visually stunning and mystifying, that feels right at home on any screen.

The following criteria guided the development process:

- **Get the right glyphs.** Like the actual ones. By now everyone’s heard how the Matrix glyphs are some treatment of katakana, but they also include a few characters from Susan Kare’s Chicago typeface. The Matrix glyphs in *this* project come from the source: cleaned up vectors from an old SWF from the promotional site for an official Matrix product, archived back in 2007. That’s how deep this rabbit hole goes, friends. (Please support the Internet Archive!)
- **Get the new glyphs.** When *Resurrections* hit theaters in December 2021, it debuted an expanded glyph set with a daunting *135 symbols*. Virtually all of them were recovered from the movie trailers for this project and uploaded before the film’s release! ...But they were of relatively poor quality. Fortunately, in this age of 720p reference material and tie-in marketing, a decent sized sample of new glyphs were eventually reverse-engineered from a sparkly watch ad, and the rest were lovingly synthesized from frames of a behind-the-scenes VFX video.
- **Make it look sweet in 2D.** The most versatile, recognizable and mesmerizing manifestation of the code rain is when it seems to pour right down your screen like rain on a windowpane. While depth effects are cool, they can obscure the details that make the difference between a goodtrix and a *greatrix*.

- 
- **Make it look sweet in 3D, too.** To facilitate future support of stereoscopic and holographic displays, it made sense to nail down a 3D variation, but it looks pretty on any kind of display.
  - **The 2D glyphs are in a fixed grid and don't move.** The “raindrops” we see in the effect are simply waves of illumination of stationary symbols that occupy a column. To get a better look at this, try setting the `fallSpeed` to a number close to 0.
  - **Get the glow and color right.** Matrix symbols aren't just some shade of phosphorous green; they're first given a bloom effect, and then get tone-mapped to the green color palette.
  - **Capture the proper rhythm of raindrops falling.** Multiple raindrops often occupy a column at the same time, and they may have different speeds, but they can never collide. This project achieves this with a sawtooth wave, modulating the width of the teeth to keep things interesting. The tips of those teeth— the cells in the grid where the sawtooth dips— are where we put the “cursors” (or “tracers”) at the bottom of each raindrop.
  - **Capture the glyph cycling sequence.** The symbols in *Reloaded* and *Revolutions*' opening titles, which were at one point the highest fidelity versions of the 2D effect, change according to a repeating sequence (see the unofficial glyph database). This is only a technical detail, and no longer drives the glyph cycle in this project, but it can be used to analyze promotional material.
  - **Whip up some artistic license and imagine the “previous” Matrix versions.** The sequels describe a paradisiacal predecessor to the Matrix that was too idyllic, and another earlier, nightmarish Hobbesian version that proved too campy. They depict some programs running older, differently colored code. So, this project dares to speculate how these old Matrix versions looked and acted.
  - **Support a broad range of customization options, and use them to produce other non-canonical variants.** See the list of links above for the full set of available versions, and see the list below to see all the ways you can personalize the effect for yourself.
  - **Make it free, open source and web based.** The MIT License permits distribution and modification of this project. Both are highly encouraged!
  - **Support as many browsers and devices as possible.** For all the flack it receives, the web is the most ubiquitous and accessible platform for sharing graphics, or anything really. This project is built on the web stack so it can reach wherever the web goes.
  - **Promote a progressive interpretation of the film franchise.** *The Matrix* is an action film you can enjoy without critical analysis, but if you do read into it, you'll be rewarded. And let's be clear: **The Matrix is a story about transitioning, directed by two siblings who transitioned.** This is undeniable. Its franchise has plenty more themes, and plenty of room for interpretation, but the communities of misogynists and bigots who claim this imagery for their movements cannot be tolerated in any form. This is a chance to open minds, not shut them.

---

## Sidenote: other people's Matrix effects

The number of implementations out there of this effect is a testament to the size of the film's impact on popular culture. For decades, I've enjoyed searching for and comparing them from time to time. That's probably how you arrived here— it's *fun* to see what kinds of solutions different people come up with to a problem, when the process is purely recreational and its success is subjective. I myself tried and failed to make the effect many times over.

Some of the earliest, roughest versions were made after the film hit theaters in March, but before it was released on home media in October— people were recreating the effect purely from memory. Others probably used the official screensaver as a reference, which was made by the time-strapped developers of the (excellent, defunct) official site from the images and multimedia tools they had available.

## Customization

You can customize the digital rain quite a bit by stapling “URL variables” to its URL— by putting a ‘?’ at the end of the link above, and then chaining together words, like this:

<https://rezmason.github.io/matrix/?width=100&fallSpeed=-0.1&effect=none>

Now you know link fu. Here's a list of customization options:

- **version** - the version of the Matrix to simulate. Default is “classic”.
  - “classic” is the Matrix code everyone knows and loves, mostly based on the sequels’ opening title graphics.
  - “3d” is the classic code in 3D mode.
  - “megacity” is a variation of the classic code that includes the Megacity as a glyph, as is seen in the opening titles of *Revolutions*.
  - “operator” is more reminiscent of the matrix code as it appears in the first movie’s opening titles, and on operators’ screens: flatter, crowded, without a gradient, and with occasional effects (such as a square ripple).
  - “nightmare” is how the Matrix may have appeared in the Merovingian’s heyday: flashy, foreboding, relentless.
  - “paradise” is how the Matrix’s idyllic predecessor may have appeared: warm, simplistic, encompassing.
  - “resurrections” is the updated Matrix code
  - “palimpsest” is a custom version inspired by the art and sound of Rob Dougan’s *Furious Angels*.
- **skipIntro** - whether or not to start from a blank screen. Can be “true” or “false”, default is *true*.

- 
- **font** - the set of glyphs to draw. Current options are “matrixcode”, “resurrections”, “gothic”, “coptic”, “huberfishA”, and “huberfishD”.
  - **width** - the number of columns (and rows) to draw. Default is 80.
  - **volumetric** - when set to “true”, this renders the glyphs with depth, slowly approaching the eye. Default is “false”.
  - **density** - the number of 3D raindrops to draw, proportional to the default. Default is 1.0.
  - **forwardSpeed** - the rate that the 3D raindrops approach. Default is 1.0.
  - **slant** - the angle that the 2D raindrops fall, in degrees. Default is 0.
  - **bloomSize** - the glow quality, from 0 to 1. Default is 0.4. Lowering this value may help the digital rain run smoother on your device.
  - **bloomStrength** - the glow intensity, from 0 to 1. Default is 0.7.
  - **ditherMagnitude** - the amount to randomly darken pixels, to conceal banding. Default is 0.05.
  - **resolution** - the image size, relative to the window size. Default is 1. Lowering this value may improve your performance, especially on high pixel density displays.
  - **raindropLength** - the vertical scale of “raindrops” in the columns. Can be any number.
  - **animationSpeed** - the overall speed of the animation. Can be any number.
  - **fallSpeed** - the speed of the rain’s descent. Can be any number.
  - **cycleSpeed** - the speed that the glyphs change their symbol. Can be any number.
  - **effect** - alternatives to the default post-processing effect. Can be “plain”, “pride”, “stripes”, “none”, “image” or “mirror”.
    - (“none” displays the ‘debug view’, a behind-the-scenes look at the anatomy of the effect.)
  - **camera** - some effects, ie. the mirror effect, optionally support webcam input. Can be “true” or “false”. Default is false.
  - **stripeColors** - if you set the effect to “stripes”, you can specify the colors of vertical stripes as alternating *R,G,B* numeric values, like so: <https://rezmason.github.io/matrix/?effect=stripes&stripeColors=1,0,0,1,1,0,0,1,0>
  - **palette** — with the normal “palette” effect, you can specify the colors and placement of the colors along the color grade as alternating *R,G,B,%* numeric values, like so: <https://rezmason.github.io/matrix/?palette=0.1,0,0.2,0,0.2,0.5,0,0.5,1,0.7,0,1>
  - **backgroundColor**, **cursorColor**, **glintColor** — other *R,G,B* values that apply to the corresponding parts of the effect.
  - **paletteHSL**, **stripeHSL**, **backgroundHSL**, **cursorHSL**, and **glintHSL** — the same as the above, except they use *H,S,L* (hue, saturation, lightness) instead of *R,G,B*.
  - **cursorIntensity**, — the brightness of cursors’ glow. Can be any number greater than zero. Default is 2.0.
  - **glintIntensity** — the brightness of glint glow, if there is any glint. Can be any number
-

---

greater than zero. Default is 1.0.

- **url** - if you set the effect to “image”, this is how you specify which image to load. It doesn’t work with any URL; I suggest grabbing them from Wikipedia: <https://rezmason.github.io/matrix/?effect=image&url=https://upload.wikimedia.org/wikipedia/commons/f/f5/EagleRock.jpg>
- **loops** - (WIP) if set to “true”, this causes the effect to loop, so that it can be converted into a looping video.
- **fps** — the framerate of the effect. Can be any number between 0 and 60. Default is 60.
- **suppressWarnings** - if set to “true”, this suppresses any warnings that would otherwise appear— when viewing the project on a device with no GPU, for example.

## Troubleshooting

There haven’t been many reported issues yet that weren’t quick fixes, but one has stood out: many visitors have previously *disabled hardware acceleration* in their Chrome browsers, at the advice of well-meaning Internet websites.

What this does is cause Chrome to fall back to **SwiftShader**, a software renderer that runs projects like this one at a much slower rate. Because of this, if you are seeing serious performance issues on Chrome, it’s recommended that you ensure hardware acceleration is enabled in your browser settings.

## Future directions

This project is still in active development, but some upcoming features are worth mentioning.

- **An audio element.** Things make sounds, don’t they? Yes, they do, especially in movies. And while silence is precious, there are plans to provide a setting that introduces some kind of pleasant audio treatment to the effect.
- **A user interface that isn’t a URL.** This project supports a lot of configurable options under the hood, and it would be wise to add a fun looking UI that exposes them all to visitors in an intuitive way.

## Friends of the project

- Vesuveus was gracious to spend time discussing this project and the effect that inspires it on his long-running podcast, The Matrix Online Revisited with Vesuveus. Fandom is the interwoven story of people interacting with a piece of media, and Vesuveus keeps ours alive and gives them perspective.

- 
- Alexi García’s stunning in-depth comparison of the many home video releases of *The Matrix* is a must-see for any fan of the franchise, and those curious about how a movie can subtly change over time. Alexi’s diligence and familiarity with the material are to thank for the high-fidelity references and high-fidelity feedback that have helped shape this project. Visit his main site for more information.
  - GitHub user 57r31 produced a proof of concept that led to the interactive mirror effect.

## Colophon

The Coptic glyphs in the “Paradise Matrix” version are derived from George Douros’s font “Symbola”, due to their similarity to the script in CG II of Nag Hammadi. If a 4th century Gnostic scribe trolled Athanasius over IRC, it might look like this.

The Gothic glyphs in the “Nightmare Matrix” version are derived from Dr. jur. Robert Pfeffer’s font “Silubur”, which are inspired by the uncial script found in the Codex Argenteus. If a werewolf emailed a vampire in the 6th century, it might look like this.

The glyphs used in the “Palimpsest” and “Twilight” versions are derived from Teague Chrystie’s font “Huberfish”, a fictitious alphabet that comes in several styles. If a spacedock technician bought a soda from a vending machine in an cool utopian future that will never happen, it might look like this.

## Other details

The glyphs are formatted as a multi-channel distance field (or MSDF) via Victor Chlumsky’s msdf-gen. This format preserves the crisp edges and corners of vector graphics when rendered as textures. Chlumsky’s thesis paper, which is in English and is also easy to read, is available to download [here](#).

The raindrops themselves are particles computed on the GPU and stored in textures, much smaller than the final render. The data sent from the CPU to the GPU every frame is negligible.