Step-by-Step Guide to the V HACKATHON

In partnership with GAMEDEV.JS

This presentation is a summary of the workshop that was held on 15 April, 9 PM (GMT+8)on Discord

Why join the V Hackathon

Level up your skills!
WebGL, Shaders, RUST, WASM

Win generous prizes!
16 winners, prizes ranging from $500 to $1000

Turn your Shader into an NFT Collectible!
Upload your art to the blockchain to make it immutable

Learn more about blockchain!
V Systems is a team of blockchain experts since 2012

v systems GAMEDEVJS

How to join the V Hackathon


Step 1: Code your Shader


Step 2: Compile your Shader into a binary format


Step 3: Publish it on the V Systems chain


Step 4: Fill out the form with the Shader NFT ID


Wait for the Winner Announcement!

v systems GAMEDEVJS

STEP 1

CODE YOUR SHADER

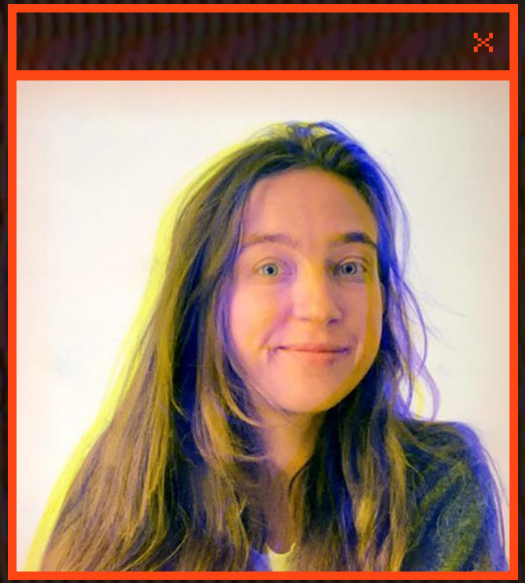
WEBGL SHADERS

Shader Expert & Judge

Ksenia Kondrashova

Web Developer &Interactive Graphics Artist

Ksenia is a freelance web developer specialized in computer graphics, motion design, interactive animation, and high-performance visualizations.



uuuuuulala ksenia-k ksenia-k.com ksenia-kondrashova

v systems GAMEDEVJS

Shaders in Web Development

3D Visualizations

- zygotebody.com – Human body visualization
- github.com – Globe visualizing GitHub activity
- nasa.gov – Asteroids map

Cool Interactions and Transitions

- designembraced.com – On-click content opener
- Image transition by Ash Thornton – On-scroll image transition
  Just

Pure Fun

- blobmixer.14islands.com – Iconic blob animation
- monopo.london – Shader background

WebGL also powers many tools you're used to: photo editors, maps, and more Even though it’s not new, shaders are having a moment in web de

Tech & Tools

A shader is a **piece of code** that runs directly **on the GPU**

When you build a webpage, you normally use languages like JavaScript, and your code runs on the CPU

→ **WebGL* builds a "bridge" between web apps and GPU**

By now, WebGPU is gradually replacing WebGL

Requirements to run a shader

- A bit of **web dev knowledge**
  HTML, CSS, JS, or frameworks like React
- **Shader knowledge**
  GLSL in WebGL, WGSL in WebGPU
- A general understanding of WebGL/WebGPU for passing data between JS and the GPU

2D and 3D Graphics

📄 Both 2D and 3D graphics are rendered on the <canvas> element. From HTML perspective, having <canvas> element on the page is all you need

🖌️ CSS also plays a very small role to set size & position of <canvas>

⚙️ Everything happens in

- JavaScript - or a framework like React
- GLSL (or WGSL) shaders
  This code is getting inserted in JS as a string and to be passed straight to GPU

WebGL for 3D

In 3D websites, you typically use both **vertex & fragment shaders**

(you can use only fragment shaders and make 3D shapes with ray marching but it's usually too heavy on performance)

To make a 3D website you need a library like Three.js (or react-three-fiber, the React version of it)

It provides a level of **abstraction** over WebGL, so you don't need to reinvent things like cameras, meshes, lights, geometries, materials, raycasting, etc.

WebGL for 3D (cont'd)

With all the primitives available, you don't really need to write custom shaders. But you can insert GLSL code in both vertex and fragment shaders

3D apple model with animated vertex shader

Loading a 3d apple model, rendering it on the webpage with Three.js and insert the noise to the vertex shader to animate the geometry

Torus Knot with custom fragment shader

Taking a Torus Knot geometry that's available in three.js and using a custom fragment shader for knot material

WebGL for 2D

The **vertex shader** is always on the web page that just has a fragment shader on it, but it only provides a simple plane that takes up the entire <canvas>

With JavaScript and WebGL, we build an interface between shader and <canvas> and pass uniforms to GLSL code:

- Canvas resolution - can be dynamic, like a full-screen animation
- Time value
- Cursor position (for basic interactivity)
- Colors, images, or any custom data

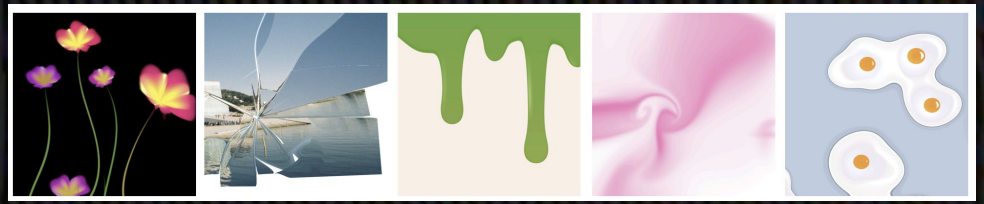
Advanced setups can include framebuffers, textures, multiple shader passes, etc etc.

WebGL for 2D (cont'd)

Combining shaders with JS can produce cool, unique visuals:

- Pass click position to a shader
- Pass image and click as a texture
- Swap framebuffers to draw on previous frame
- Scroll-based shader
- Fluid simulation with multiple shader passes

Compared to other web graphic tools (SVG, Canvas API), WebGL is way more powerful but also harder to set up and debug



Resources

- 📚 thebookofshaders.com - Shader intro
- 🎓 threejs-journey.com - three.js course with good shaders chapter (paid, but 100%
- worth it)
- 🔧 webglfundamentals.org and webgl2fundamentals.org - everything to know about WebGL
- underhood
- 🎨 shadertoy.com - open-source shaders to browse
- 🧪 codepen.io - open-source web demos, search for WebGL/shaders experiments

STEP 2

COMPILE INTO A BINARY

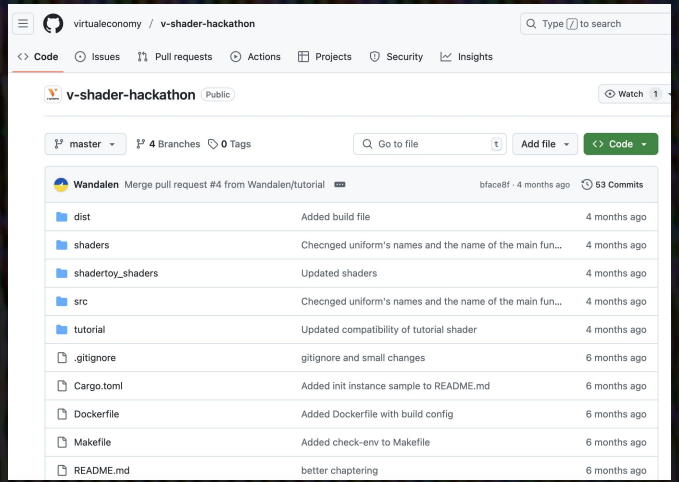
RUST WASM

Publish your Shader on the blockchain!

Step 1: Follow our Github repo instruction to compile the Shader into a binary format

github.com/virtualeconomy/v-shader-hackathon

Tutorial

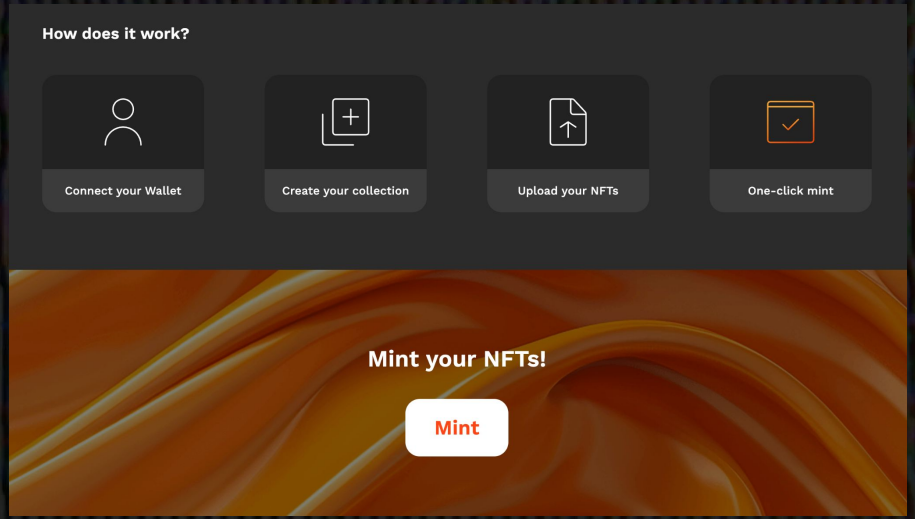


v systems GAMEDEVJS

STEP 3

PUBLISH IT ON CHAIN

Publish your art on the chain with our one-click Minting tool:https://marketplace.v.systems/mint



Tutorial

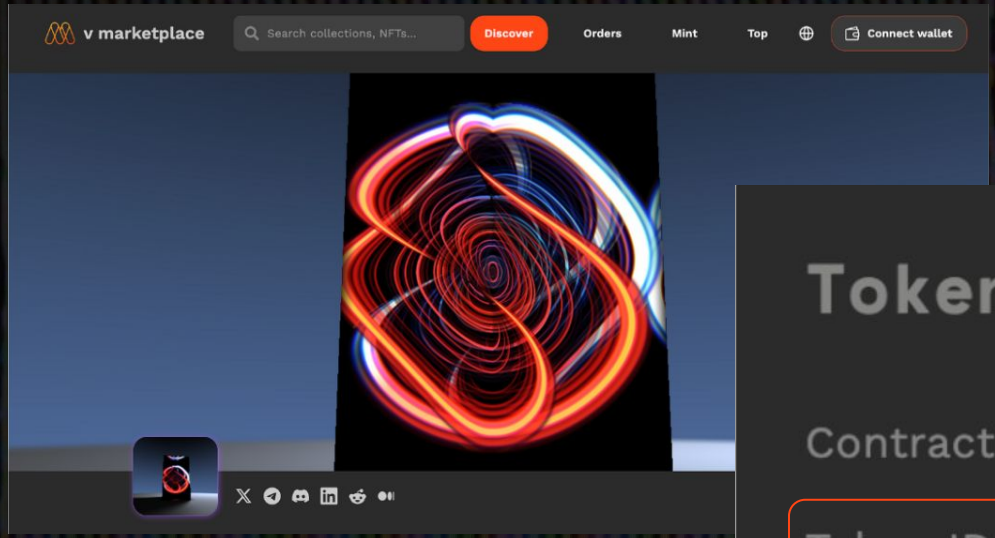
STEP 4

SUBMIT FOR REVIEW

Fill out the form with your Shader NFT ID:
https://hackathon.v.systems/submit

Tutorial

v systems GAMEDEVJS



Token Details

Contract Address	CC4TbdZ...9H568aR
Token ID	TWZ8Tma...jt6WkwX
Blockchain	VSYS
Token Standards	NonFungibleContract

Example

RESOURCES

Resources

About shaders	https://thebookofshaders.com/

Interactive Shader editors	http://editor.thebookofshaders.com/
	https://www.shadertoy.com/howto

WebGL tutorials	
WebGL step-by-step tutorial	https://learnwebgl.brown37.net/https://learnwebgl.brown37.net/
WebGL step-by-step tutorial	https://webglfundamentals.org/
WebGL step-by-step tutorial	https://webgl2fundamentals.org/
WebGL documentation	https://developer.mozilla.org/en-US/docs/Learn/WebGL

About V Systems	
Official website	https://v.systems/
V Systems Marketplace	https://marketplace.v.systems/
Upload your Shader to the V Systems' blockchain	https://marketplace.v.systems/mint
About the V Systems Marketplace	https://medium.com/vsystems/v-marketplace-decentralized-nft-platform-9f813342007d

Use the V Marketplace	
Open a Titan Wallet	https://medium.com/vsystems/how-to-create-a-new-account-on-titan-wallet-5556985da07c
How to upload your project to the V Systems chain	https://medium.com/vsystems/mint-list-your-nfts-on-v-systems-8f85d839cd18

https://github.com/virtualeconomy/v-shader-hackathon
https://hackathon.v.systems/techresources
https://medium.com/@vsystems

Questions

- **How can I use a shader to mix two images of different resolution?**
  - For example, I want to use a perlin noise texture, apply it to an image (or a sprite in a game) to dynamically dissolve the sprite.
- What should our submissions look like?
  - Is it ok to submit a zip file containing a html page running a WebGL program when opened in a browser?

Schedule

UNTIL 26 APRIL	27 APRIL - 15 MAY	16 MAY 2025
Hackathon Submissions	Project review	Winner announcement

ABOUT V SYSTEMS

About V Systems

V Systems (VSYS) is an open-source network that supports the efficient and agile development of decentralized applications.

Public Blockchain for Developers

www.v.systems

4 seconds
Average block size

<1 sec Average transfer completion time

15 nodes Worldwide

0.1 VSYS Transaction fee

V Systems' community comprises over **30,000 devs and VSYS enthusiasts**

Enterprise Blockchain Solutions

codedsolution.xyz

Cyberport 數碼港

GREEN FINANCE DEVELOPMENT

OUTSTANDING ONE STOP LOYALTY & VOUCHER MANAGEMENT PLATFORM
ViO

BEST CONSUMER PAYMENTS INITIATIVE - OUTSTANDING BLOCKCHAIN IN PAYMENT SOLUTION
Co+Pay

1ST PRIZE - BLOCKCHAIN TECHNOLOGY POWERING THE DIGITAL VOUCHER AND LOYALTY POINTS ECOSYSTEM
ViO

OUTSTANDING SME AWARD IN DIGITAL TRANSFORMATION / TECHNOLOGY

3RD PRIZE - HONG KONG 2023 WEB 3.0 CREATION

HONG KONG GREEN FINTECH MAP 2024

BASIC TECHNOLOGY – PAYMENT AND REMITTANCE TECHNOLOGIES
Co+Pay

v systems

GAMEDEVJS

Follow us on social media

hackathon.v.systems/
panel

VSYSCoin

V Systems

t.me/vhackathon

discord.gg/2N9cWJ42JX

Medium

V Systems

v systems

GAMEDEVJS