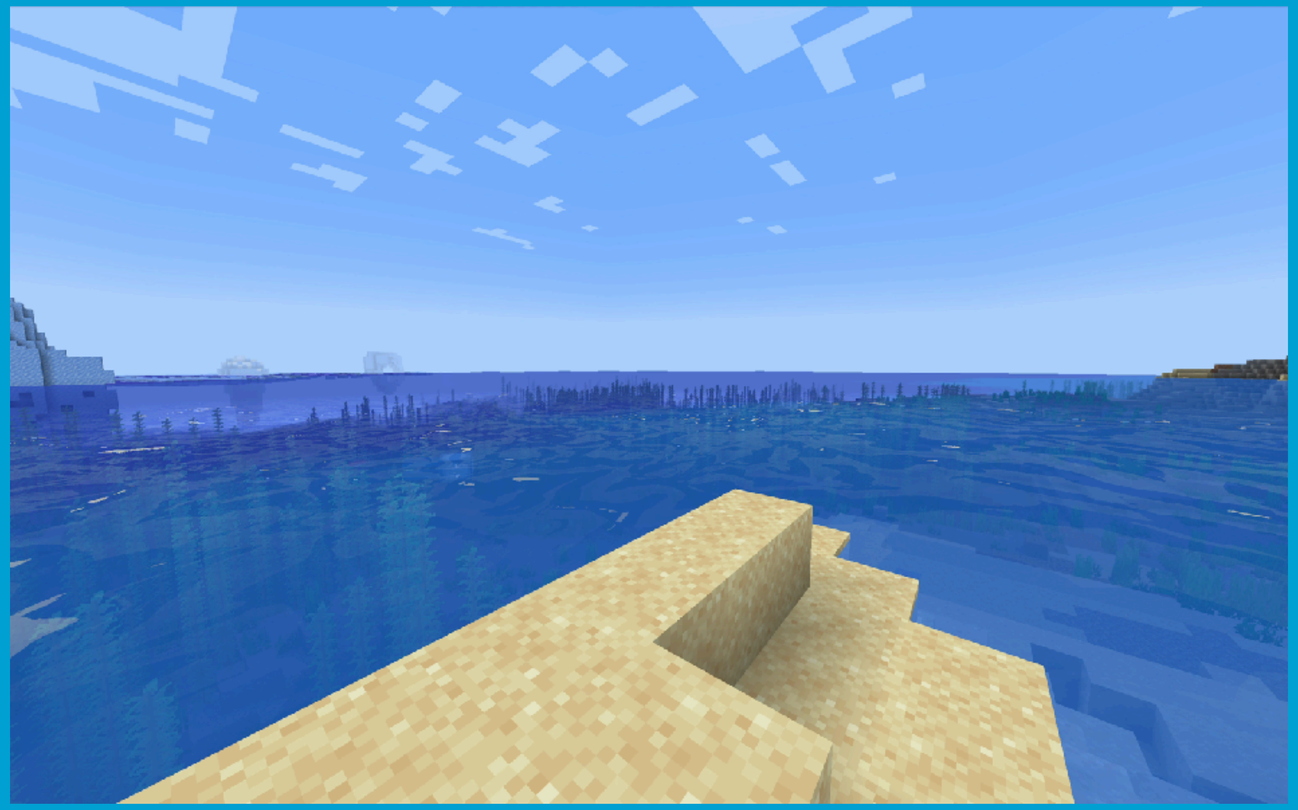
SHADER WORKSHOP

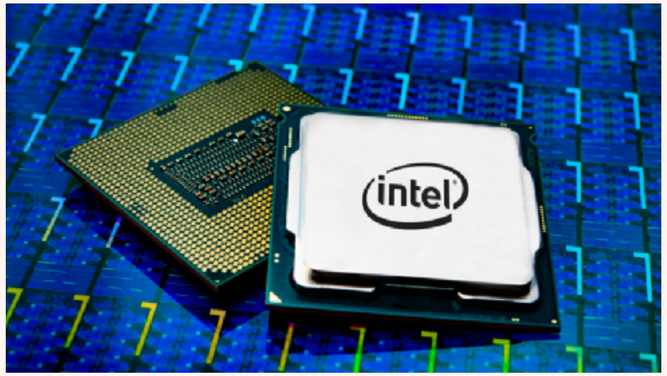
THE GOAL:



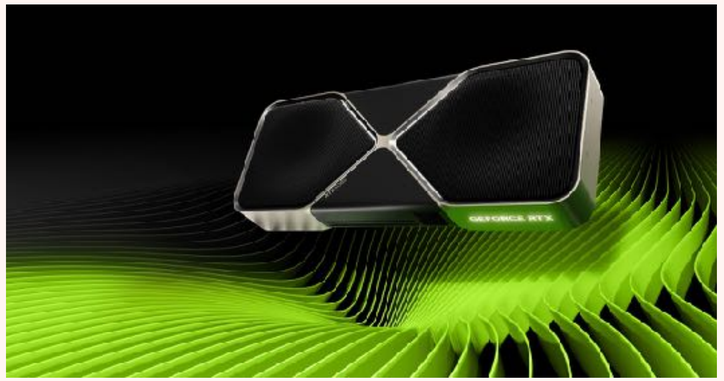
WHAT IS A SHADER

SHADERS ARE PROGRAMS FOR GRAPHICS PROCESSORS!

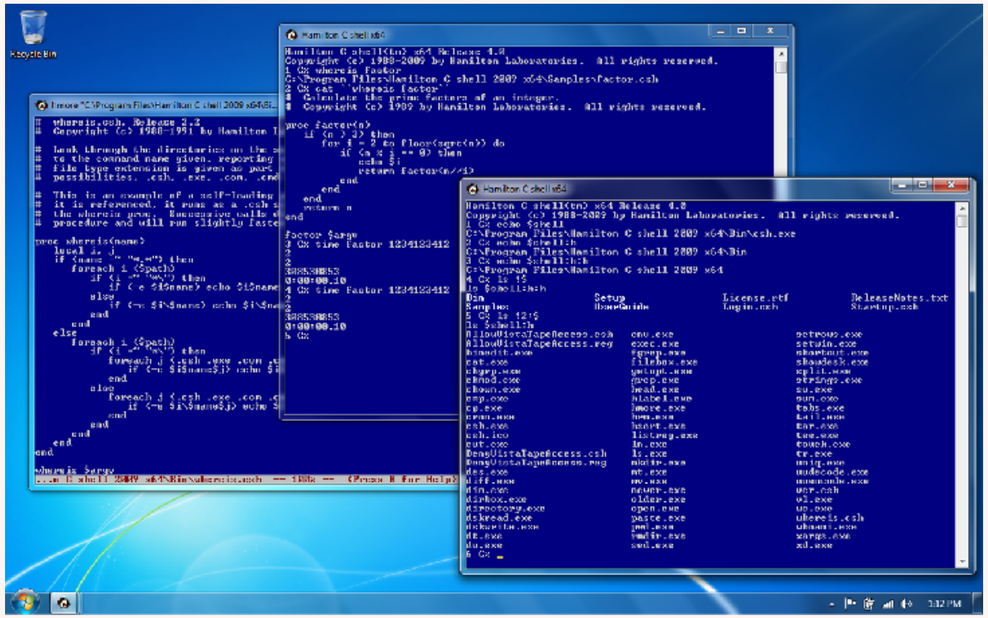
Regular programs run on the CPU



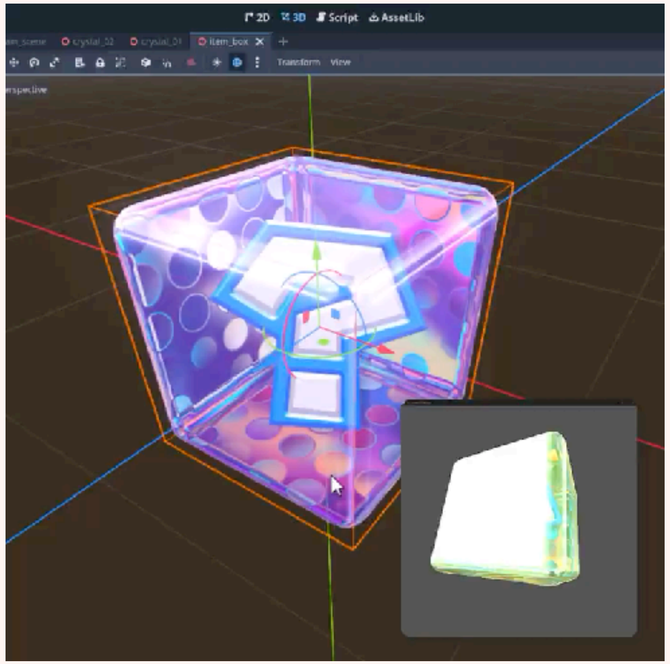
Shaders run on the GPU



Regular Program

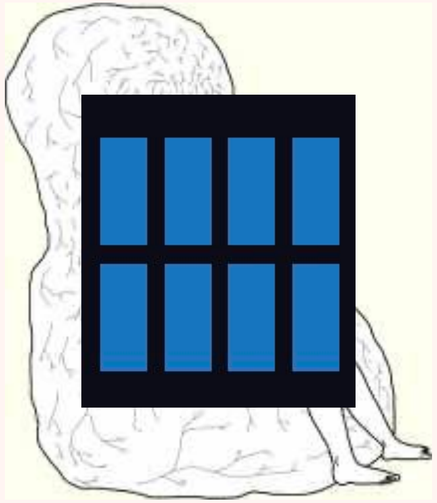


Shader

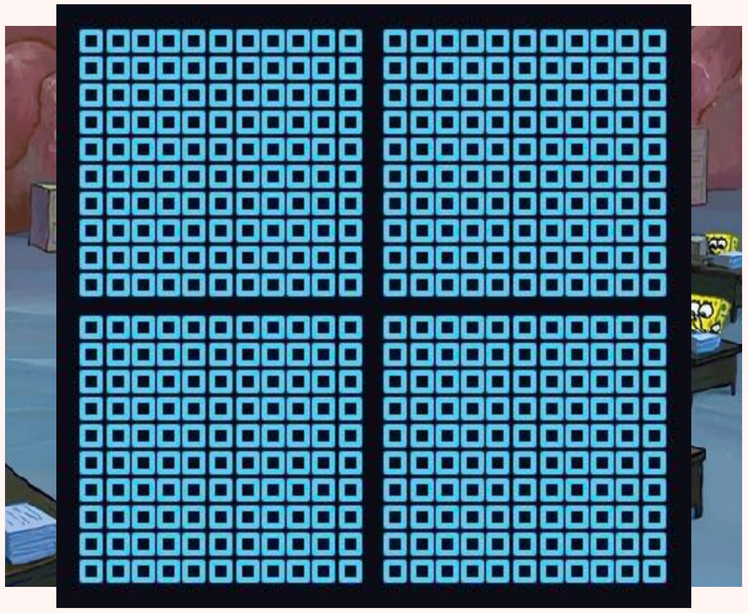


CPUS ARE COMPLETELY DIFFERENT FROM GPUS

CPU

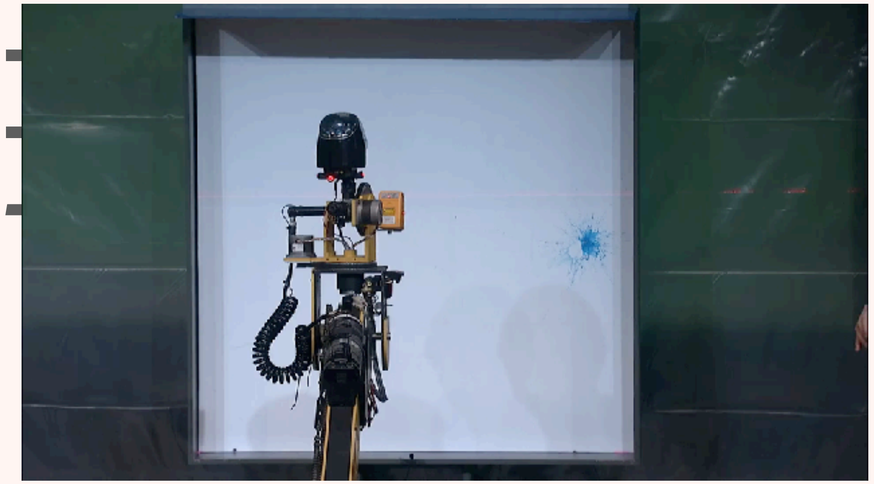


GPU

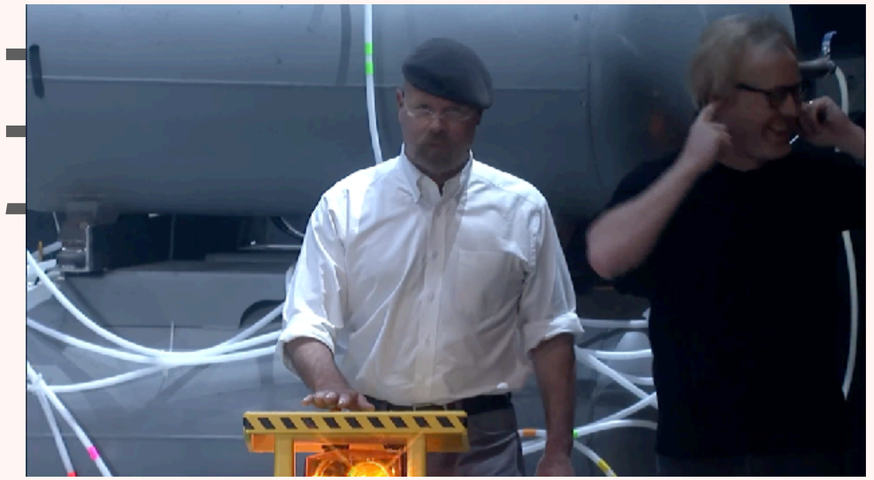


TL;DR

CPU



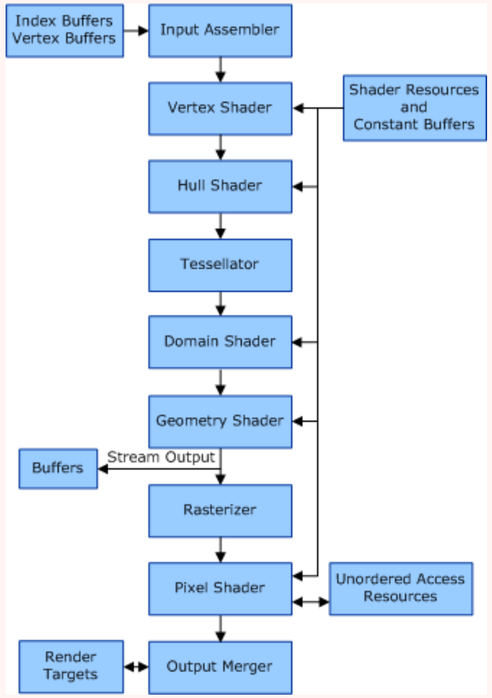
GPU



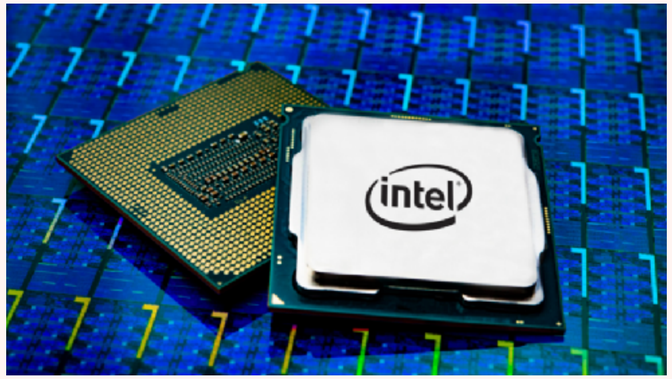
WHY DO WE NEED SPECIAL PROGRAMMING LANGUAGES JUST FOR THE GPU??

Graphics processors have a very different pipeline than CPUs!

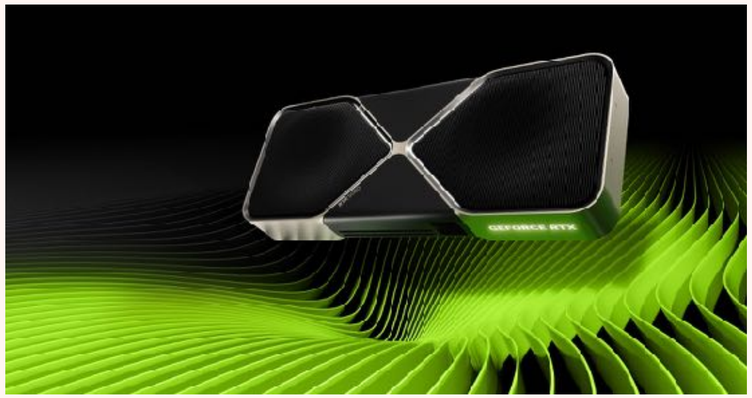
Specialized shader languages make it waaaaay easier to program for a GPU compared to traditional languages.



SHADER LANGUAGES ARE TIED TO THE GRAPHICS API



Graphics API



Since 99% of programs run on a CPU, graphics APIs allow the CPU and GPU to talk to each other.

Some of the names of these APIs include: OpenGL, DirectX, Vulkan, and Metal

~SHADER LANGUAGES~

GLSL

The common language for OpenGL and Vulkan

We're using this ^^

HLSL

The DirectX shader language

CUDA

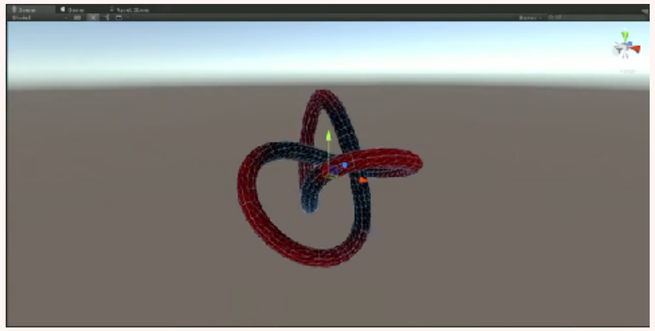
Nvidia's language for non-graphics shaders.
(ML, AI, Simulations, Crypto...)

Other

Game consoles and other special snowflakes (Pixar, Apple, etc..) can have custom shader languages

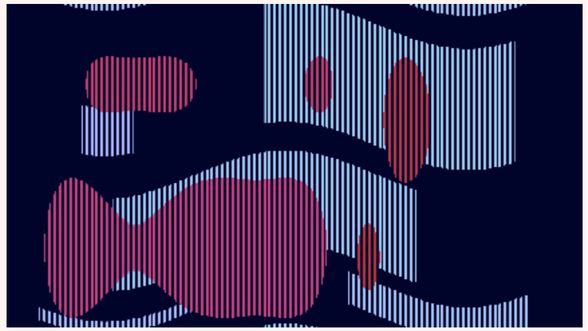
HOW DO WE MAKE A MINECRAFT SHADER

THE DIFFERENT CORE TYPES OF SHADER



Vertex Shader

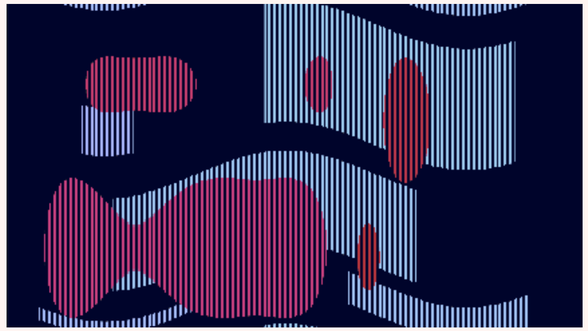
Allows you to manipulate the different vertices (points) in a scene



Fragment Shader

“Colours” in the spaces between the vertices

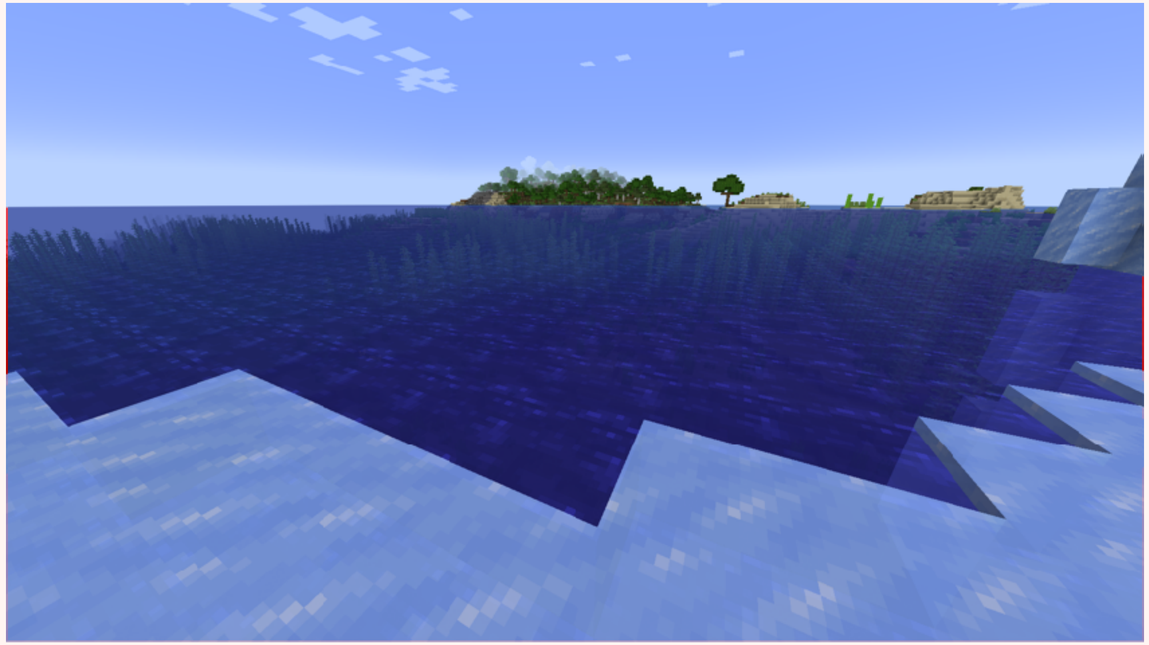
WE'LL JUST FOCUS ON USING FRAGMENT SHADERS HERE



Fragment Shader

"Colours" in the spaces between the vertices

LET’S PAINT THE WATER RED!

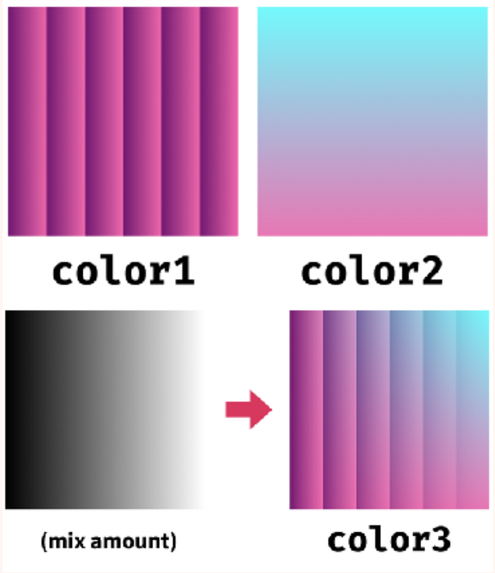


HOW DO WE MAKE A WATER SHADER

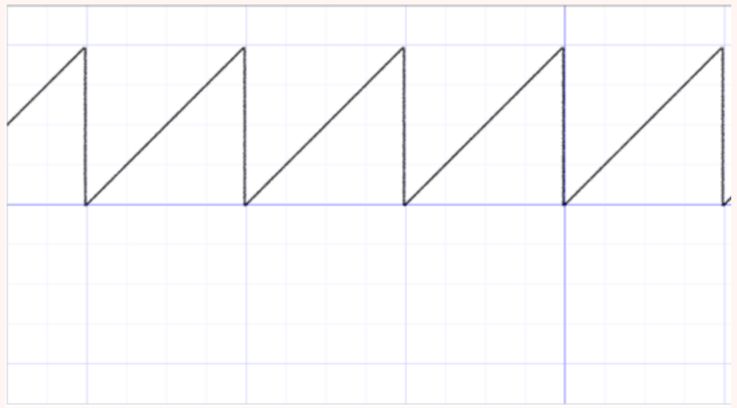
GLSL Math Utilities

mix();

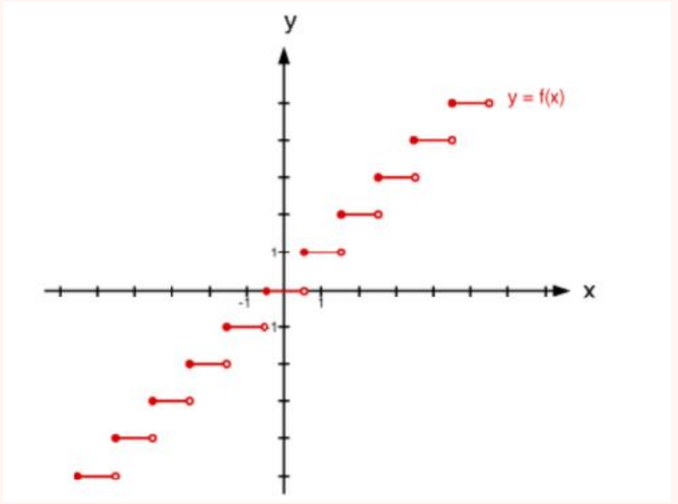
(aka lerp())



frac();



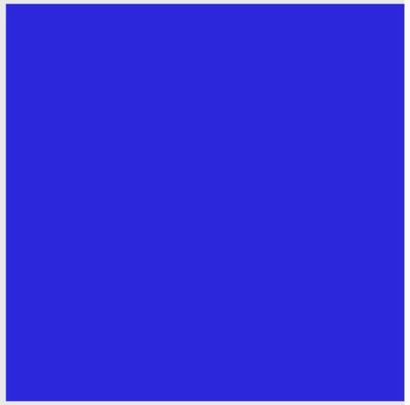
step();



mix();

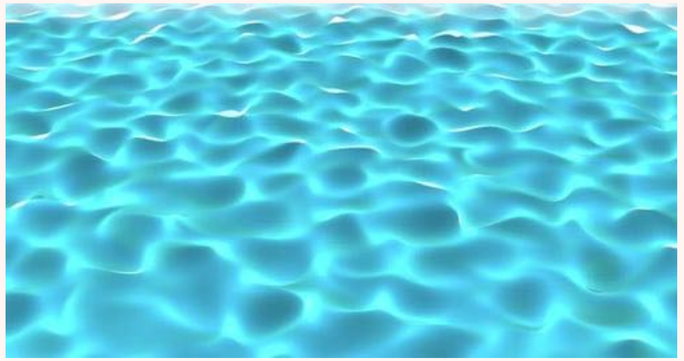
mix(); does linear interpolation between values x and y using the function a

```
#ifdef GL_ES
precision mediump float;
#endif

uniform vec2 u_resolution;
uniform float u_time;

vec3 colorA = vec3(0.149,0.141,0.912);
vec3 colorB = vec3(1.000,0.833,0.224);

void main() {
    vec3 color = vec3(0.0);

    float waveValue = abs(sin(u_time));

    // Mix uses waveValue (a value from 0-1) to
    // mix the two colors
    // Closer to colorA when waveValue = 0
    // Closer to colorB when waveValue = 1
    color = mix(colorA, colorB, waveValue);

    gl_FragColor = vec4(color,1.0);
}

```

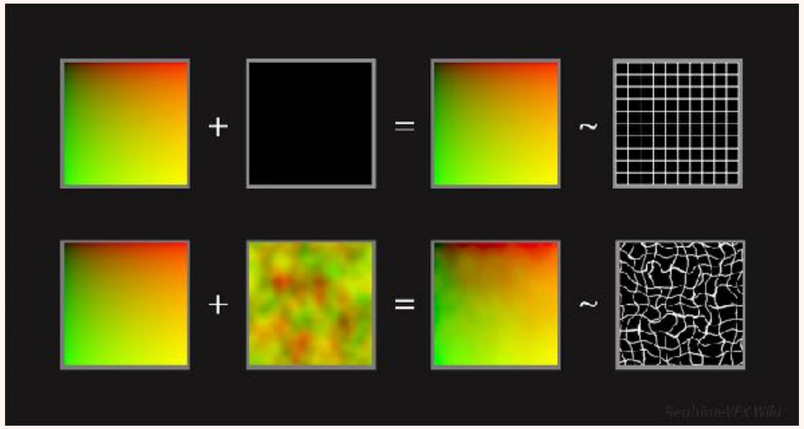


OUR WATER SHADER IS SPLIT INTO THREE STEPS

1. Water ripples



2. UV Distortion



3. Distance Fade

Water Ripples

We have textures of waves

If we overlap and move two of them that gives complex motion

Lastly, if we cut out the middle values that gives us this sparkly effect!

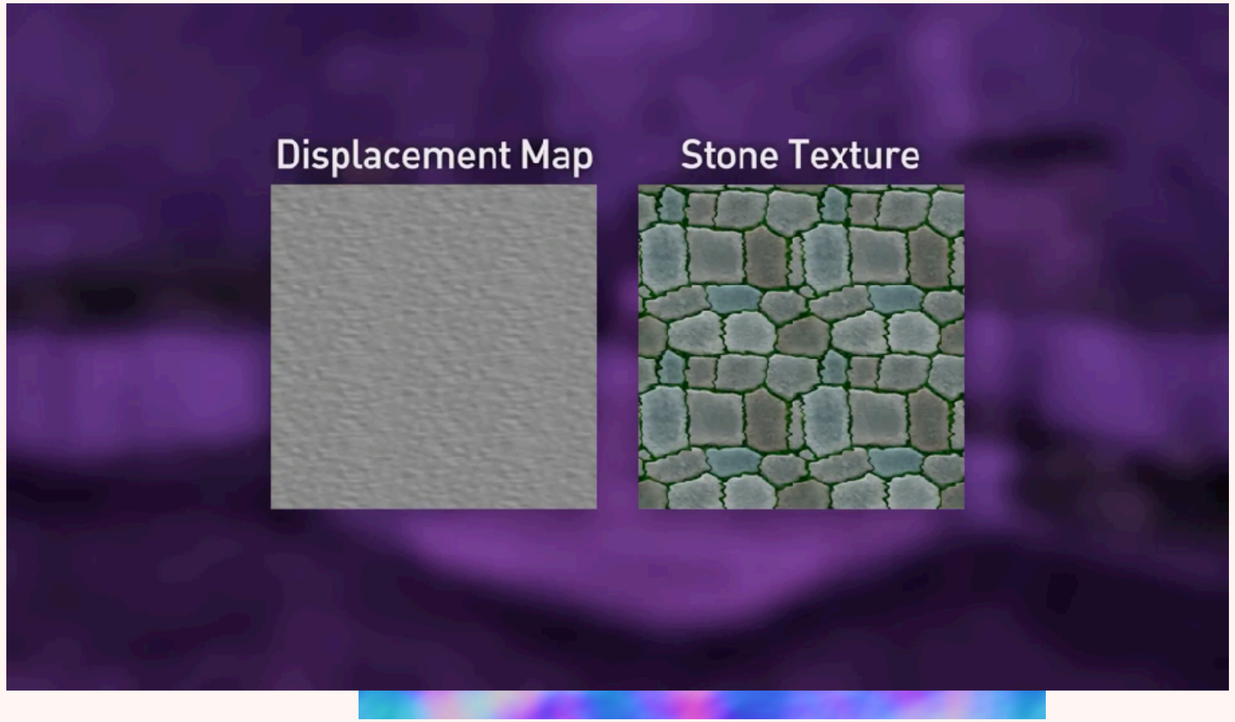
Let’s Try It!

UV Distortion

We have another different wave texture

By using this we can distort the waves we made before so that they look more “3D”

The distortion sets pixels to towards the top left or towards bottom right depending on how bright that area of the noise texture is



Let’s Try It!

3. Distance Fade

Ideal for reducing the repetition of the waves

Implemented by finding shader's distance from camera

Add a colour that makes the water noise invisible when the distance is before or after a certain range

Let’s Try It!

REFERENCES:

UNITY SHADER BIBLE - MARIO KART SHADER: HTTPS://X.COM/USHADERSBIBLE/STATUS/1888666752193646815
MYTHBUSTERS CPU VS. GPU: HTTPS://YOUTU.BE/-P28LKWTZRI?SI=QOBNTIJDXAIWXFQO
UNITY VERTEX SHADER VIDEO: HTTPS://YOUTU.BE/CW0MYDNEVJM?SI=-6A3XXOSKQCOQRUR
HOW SCROLLING TEXTURES GAVE SUPER MARIO GALAXY 2 ITS CHARM: HTTPS://YOUTU.BE/8RCRSOLIO7K?SI=FYXUJ27PLUNZ286W

HTTPS://NOCLIP.WEBSITE/

THANK YOU FOR COMING!