

The background is a solid blue color with abstract, wavy, organic shapes in a lighter shade of blue, creating a sense of movement and depth. Scattered throughout the background are numerous small, white, circular bubbles of varying sizes, some with thin outlines and others as simple dots, giving the impression of an underwater scene.

Lil Koi Pond

By Me (Nick C)

The Goal: Make a lil koi pond.

01

Make a basic scene setup

Pretty self explanatory

02

Make basic world texture

It's called pondbase.png,
it's the base of the pond

03

Make water

It's a box, and a plane,
they're both blue

04

Make fish

Make some lil guys to
swim around and be fish

04

Make ripples and fish food

Give us a way to play with
our lil fishes

The current state:



So you want to make realistic water?

its literally just shadders

My First Shaders

Final-Project > shaders > vertexShader.glsl

```
1 uniform float time;
2 uniform vec3 clickPosition;
3 varying vec3 vColor;
4 varying vec2 vUv;
5 varying float wave;
6
7 void main() {
8     vec3 pos = position;
9
10    // Calculate distance from the click position
11    float distance = distance(pos.xy, clickPosition.xy) * 0.5; // Scale distance to make ripples larger
12
13    // Wave parameters
14    float waveSpeed = 8.0; // Reduced speed for smoother ripples
15    float waveNumber = 1.0; // Lower frequency for larger waves
16    float decayRate = 1.5; // Reduced decay to allow waves to travel farther
17
18    // Generate ripples
19    float ripple1 = 0.1 * sin(distance * waveSpeed - (time - 0.0) * waveNumber);
20    float ripple2 = 0.05 * sin(distance * waveSpeed - (time - 0.4) * waveNumber);
21    float ripple3 = 0.05 * sin(distance * waveSpeed - (time - 0.8) * waveNumber);
22
23    // Apply decay factor to fade ripples with distance
24    ripple1 *= exp(-distance * decayRate);
25    ripple2 *= exp(-distance * decayRate);
26    ripple3 *= exp(-distance * decayRate);
27
28    // Combine ripples and apply to the y-coordinate
29    float totalWave = (ripple1 + ripple2 + ripple3) / (distance * 0.5 + 1.0); // Slightly reduce the divisor for higher wave amplitudes
30    pos.y += totalWave;
31    wave = totalWave;
32
33    // Calculate reflection vector for the environment map
34    vec3 cameraOverVertex = normalize(cameraPosition - pos);
35    vec3 reflect = reflect(cameraOverVertex, normalize(normal));
36
37    vUv = uv;
38    gl_Position = projectionMatrix * modelViewMatrix * vec4(pos, 1.0);
39
40 }
```

Final-Project > shaders > fragmentShader.glsl

```
1 uniform vec3 baseColor; // Base water color
2 uniform float opacity; // Transparency level
3 uniform sampler2D backgroundTexture; // Background texture (pond base)
4 varying vec2 vUv; // Texture coordinates
5 varying float vWave; // Ripple height
6 varying vec3 vNormal; // Perturbed normal for lighting
7
8 void main() {
9    // Distort UVs based on ripple height for subtle refraction effect
10   vec2 distortedUv = vUv + vWave * 0.05; // Increased distortion for ripples
11
12   // Sample the background texture (pond base)
13   vec3 background = texture2D(backgroundTexture, distortedUv).rgb;
14
15   // Add Fresnel effect for light interaction
16   vec3 viewDir = normalize(vec3(0.0, 0.0, 1.0));
17   float fresnel = pow(1.0 - dot(viewDir, normalize(vNormal)), 3.0);
18   vec3 highlightColor = vec3(1.0, 1.0, 1.0) * fresnel;
19
20   // Enhance ripple edges
21   float rippleEdge = smoothstep(0.1, 0.3, abs(vWave)) * 2.0; // Stronger edges
22   vec3 rippleOutline = vec3(1.0) * rippleEdge; // White outline for ripple peaks
23
24   // Add shadows for ripples (darkened regions beneath waves)
25   float shadow = smoothstep(0.1, 0.3, -vWave) * 0.4; // Darker shadows
26   vec3 rippleShadow = vec3(0.0, 0.0, 0.0) * shadow; // Black shadows
27
28   // Blend base color, background, shadows, and highlights
29   vec3 waterColor = mix(baseColor, background, 0.6); // Base water with background texture
30   waterColor += highlightColor * rippleEdge; // Add Fresnel highlights to edges
31   waterColor += rippleOutline; // Add white outline
32   waterColor += rippleShadow; // Add ripple shadows
33
34   // Final color with transparency
35   gl_FragColor = vec4(waterColor, opacity);
36 }
```

[illegible]

The background is a solid blue color. It features several white, wavy, organic shapes that resemble liquid or smoke, primarily located along the top and bottom edges. Scattered throughout the blue area are numerous small, white, semi-transparent circles of varying sizes, giving the impression of bubbles or dust particles.

Thanks