

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

Water Shader

EDAF80: Computer Graphics

Rikard Olajos



Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

- Texture coordinates
- Cube mapping and skybox
- Phong shading
- Any questions?

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

1 Introduction

2 Waves

3 Shading

4 Assignment 4

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

Introduction

- Finch, M.: [Effective Water Simulation from Physical Models](#)
- Excerpt from *GPU Gems*
- Note: uses $(\mathbf{B}, \mathbf{T}, \mathbf{N})$ instead of $(\mathbf{t}, \mathbf{b}, \mathbf{n})$
- 2nd note: there is a mix-up of terminology (phase/frequency/wave number)



Introduction

Article

Overview

Waves

One wave

Sum of waves

▲ Exercise 4-1

GLSL functions

Shading

Water colour

Reflection

Animated normal mapping

Tangent space

Fresnel reflection

Fresnel refraction

Final result

Assignment 4

Demo

▲ Exercise 4-2

Introduction

Article

Overview

Waves

One wave

Sum of waves

▲ Exercise 4-1

GLSL functions

Shading

Water colour

Reflection

Animated normal
mapping

Tangent space

Fresnel reflection

Fresnel refraction

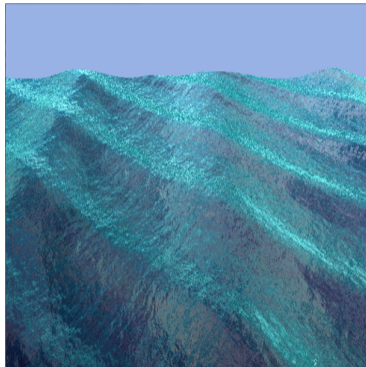
Final result

Assignment 4

Demo

▲ Exercise 4-2

- Waves
- Shallow & deep colour
- Fresnel reflection
- Fresnel refraction
- Animated normal mapping



Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

Waves

Introduction

Article
Overview

Waves

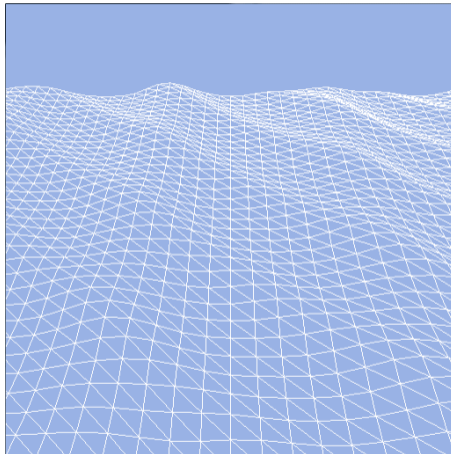
One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2



WAVES

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

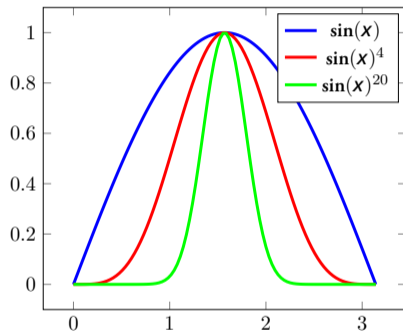
Assignment 4

Demo
▲ Exercise 4-2

- Sum of sines
- Form sharper crest by raising to an exponent k (*sharpness*)

$$\sin(x) \rightarrow \sin(x)^k$$

- Similar to *shininess* in the Phong model



ONE WAVE

- A = amplitude
- $D = (D_x, D_z)$ = direction of travel
- f = frequency
- φ = phase constant
- k = sharpness
- t = time
- (x, z) = position on plane

$$y = G(x, z, t) = A(\sin((D_x \cdot x + D_z \cdot z) \cdot f + \varphi t) \cdot 0.5 + 0.5)^k$$

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

- A = amplitude
- $D = (D_x, D_z)$ = direction of travel
- f = frequency
- φ = phase constant
- k = sharpness
- t = time
- (x, z) = position on plane

$$y = G(x, z, t) = A(\sin((D_x \cdot x + D_z \cdot z) \cdot f + \varphi t) \cdot 0.5 + 0.5)^k$$

$$\frac{\partial G}{\partial x} = 0.5kfA(\sin((D_x \cdot x + D_z \cdot z) \cdot f + \varphi t) \cdot 0.5 + 0.5)^{k-1} \cdot \cos((D_x \cdot x + D_z \cdot z) \cdot f + \varphi t) \cdot D_x$$

$$\frac{\partial G}{\partial z} = 0.5kfA(\sin((D_x \cdot x + D_z \cdot z) \cdot f + \varphi t) \cdot 0.5 + 0.5)^{k-1} \cdot \cos((D_x \cdot x + D_z \cdot z) \cdot f + \varphi t) \cdot D_z$$

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

- One wave

$$y = \mathbf{G}(\mathbf{x}, \mathbf{z}, t) = A(\sin((\mathbf{D}_x \cdot \mathbf{x} + \mathbf{D}_z \cdot \mathbf{z}) \cdot f + \varphi t) \cdot 0.5 + 0.5)^k$$

- Sum of waves

$$H(\mathbf{x}, \mathbf{z}, t) = \sum \mathbf{G}_i$$

$$\frac{\partial H}{\partial \mathbf{x}} = \sum \frac{\partial \mathbf{G}_i}{\partial \mathbf{x}}$$

$$\frac{\partial H}{\partial \mathbf{z}} = \sum \frac{\partial \mathbf{G}_i}{\partial \mathbf{z}}$$

Seminar Exercise 4-1: Procedural Vertex Animation

Introduction

Article

Overview

Waves

One wave

Sum of waves

▲ Exercise 4-1

GLSL functions

Shading

Water colour

Reflection

Animated normal
mapping

Tangent space

Fresnel reflection

Fresnel refraction

Final result

Assignment 4

Demo

▲ Exercise 4-2

- 1 Implement a wave function using `sin()`.
 - Have the wave depend on the *p.x* variable and *time*.
 - Reduce the amplitude of the wave to 0.05.
 - Set the spatial frequency to 25.0.
 - Change the phase constant (temporal frequency) to 5.0.
 - Introduce a dependency on the *p.y* variable as well.
 - A good direction for this wave is (0.4, 0.2).

- Parameter qualifiers
 - **in**: copied into the function (default)
 - **out**: copied out of the function
 - **inout**: copied into and out of the function

```
void my_function(in float a, out float b, inout float c) {  
    b = a + c;  
    c = 7.7;  
}
```

```
float a = 2.2; float b = 3.3; float c = 4.4;  
my_function(a, b, c); // b = 6.6, c = 7.7
```

- Can not be recursive
- More info in the [OpenGL wiki](#)

GLSL WAVE FUNCTION

```
layout (location = 0) in vec3 vertex;  
layout (location = 1) in vec3 normal;
```

```
uniform mat4 vertex_model_to_world;  
uniform mat4 normal_model_to_world;  
uniform mat4 vertex_world_to_clip;  
/* Add time uniform */
```

```
out VS_OUT {  
    vec3 vertex;  
    vec3 normal;  
} vs_out;
```

```
float wave(vec2 position, vec2 direction, float amplitude, float frequency,  
          float phase, float sharpness, float time)  
{  
    return amplitude * pow(sin((position.x * direction.x + position.y * direction.y)  
                            * frequency + phase * time) * 0.5 + 0.5, sharpness);  
}
```

```
void main()  
{  
    vec3 displaced_vertex = vertex;  
    displaced_vertex.y += wave(vertex.xz, vec2(-1.0, 0.0), /* fill in... */ );  
  
    vs_out.vertex = vec3(vertex_model_to_world * vec4(displaced_vertex, 1.0));  
    vs_out.normal = vec3(normal_model_to_world * vec4(normal, 0.0));  
  
    gl_Position = vertex_world_to_clip * vertex_model_to_world * vec4(vertex, 1.0);  
}
```

Introduction

Article

Overview

Waves

One wave

Sum of waves

▲ Exercise 4-1

GLSL functions

Shading

Water colour

Reflection

Animated normal
mapping

Tangent space

Fresnel reflection

Fresnel refraction

Final result

Assignment 4

Demo

▲ Exercise 4-2

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

- Keep track of elapsed time

```
float elapsed_time_s = 0.0f;
```

- Update each frame in render loop

```
elapsed_time_s += std::chrono::duration<float>(deltaTimeUs).count();
```

- Send time to shader program

```
glUniform1f(glGetUniformLocation(program, "t"), elapsed_time_s);
```


Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

Shading

WATER COLOUR

Introduction

- Article
- Overview

Waves

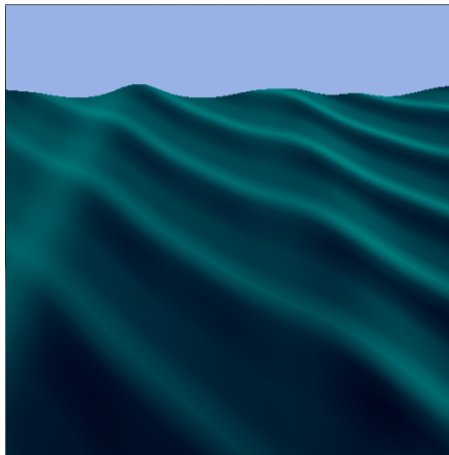
- One wave
- Sum of waves
- ▲ Exercise 4-1
- GLSL functions

Shading

- Water colour**
- Reflection
- Animated normal mapping
- Tangent space
- Fresnel reflection
- Fresnel refraction
- Final result

Assignment 4

- Demo
- ▲ Exercise 4-2



WATER COLOUR

Introduction

Article
Overview

Waves



One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

- `colordeep` = (0.0, 0.0, 0.1, 1.0) 
- `colorshallow` = (0.0, 0.5, 0.5, 1.0) 
- $\mathbf{n} = (-\partial H / \partial x, 1, -\partial H / \partial z)$
- `facing` = $1 - \max(\mathbf{V} \cdot \mathbf{n}, 0)$
- `colorwater` = `mix(colordeep, colorshallow, facing)`

Introduction

Article
Overview

Waves

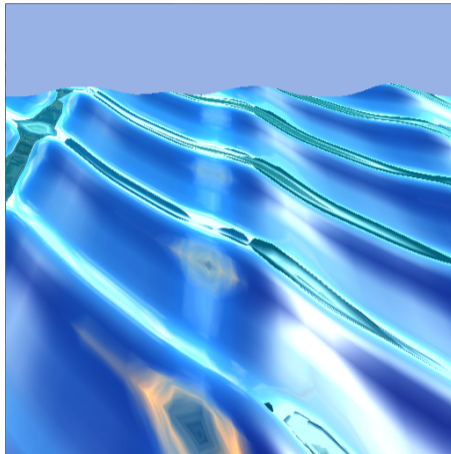
One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2



Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

- Cube mapping, as in assignment 3
- $\mathbf{n} = (-\partial H / \partial \mathbf{x}, 1, -\partial H / \partial \mathbf{z})$
- $\mathbf{R} = \text{reflect}(-\mathbf{V}, \mathbf{n})$
- `color = colorwater + reflection`
- Reflection is looked up in the cube map

ANIMATED NORMAL MAPPING

Introduction

Article
Overview

Waves

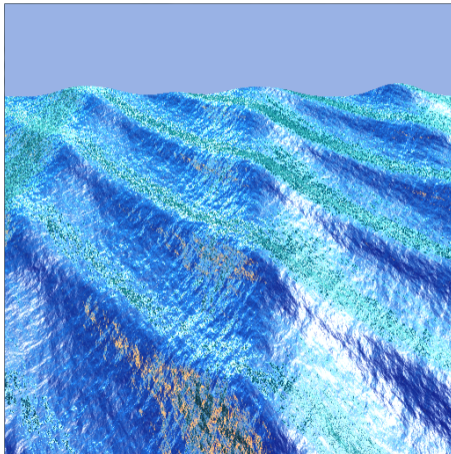
One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2



ANIMATED NORMAL MAPPING

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

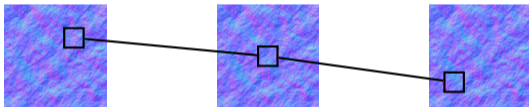
Shading

Water colour
Reflection
Animated normal mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

- “Sliding windows” — superposition from multiple, time-dependent normal map coordinates



- Vertex shader: calculate coordinate pairs
- Fragment shader: read and do superposition

ANIMATED NORMAL MAPPING: COORDINATES

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

```
vec2 texScale = vec2(8, 4);  
float normalTime = mod(time, 100.0);  
vec2 normalSpeed = vec2(-0.05, 0.0);  
  
normalCoord0.xy =  
    texCoord.xy * texScale + normalTime * normalSpeed;  
normalCoord1.xy =  
    texCoord.xy * texScale * 2 + normalTime * normalSpeed * 4;  
normalCoord2.xy =  
    texCoord.xy * texScale * 4 + normalTime * normalSpeed * 8;
```


ANIMATE NORMAL MAPPING: READ, REMAP, SUPERPOSITION

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

- $\mathbf{n}_i = \text{texture}(\text{normalTexture}, \text{normalCoord}_i) * 2 - 1$
- $\mathbf{n}_{\text{bump}} = \text{normalize}(\sum \mathbf{n}_i)$ (this is in *tangent space*)

- Equation for 3-D point

$$\mathbf{P}(\mathbf{x}, \mathbf{z}, t) = (\mathbf{x}, H(\mathbf{x}, \mathbf{z}, t), \mathbf{z})$$

- Tangent space

$$\mathbf{t} = \left(\frac{\partial \mathbf{x}}{\partial \mathbf{x}}, \frac{\partial H}{\partial \mathbf{x}}, \frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) = \left(1, \frac{\partial H}{\partial \mathbf{x}}, 0 \right)$$

$$\mathbf{b} = \left(\frac{\partial \mathbf{x}}{\partial \mathbf{z}}, \frac{\partial H}{\partial \mathbf{z}}, \frac{\partial \mathbf{z}}{\partial \mathbf{z}} \right) = \left(0, \frac{\partial H}{\partial \mathbf{z}}, 1 \right)$$

$$\mathbf{n} = \mathbf{t} \times \mathbf{b} = \left(-\frac{\partial H}{\partial \mathbf{x}}, 1, -\frac{\partial H}{\partial \mathbf{z}} \right)$$

- tangent space \rightarrow world space

$$\mathbf{TBN} * \mathbf{n}$$

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

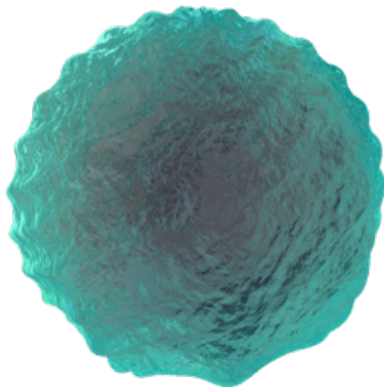
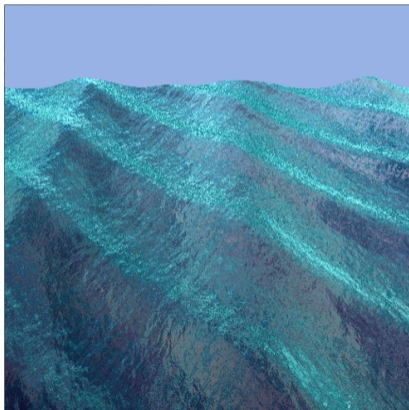
Assignment 4

Demo
▲ Exercise 4-2

MORE COMPLEX SURFACES

- In general: wave \rightarrow surface \rightarrow model \rightarrow world

$$\mathbf{WORLD} * \mathbf{TBN}_{\text{surface}} * \mathbf{TBN}_{\text{water}} * \mathbf{n}$$



Introduction

Article

Overview

Waves

One wave

Sum of waves

▲ Exercise 4-1

GLSL functions

Shading

Water colour

Reflection

Animated normal
mapping

Tangent space

Fresnel reflection

Fresnel refraction

Final result

Assignment 4

Demo

▲ Exercise 4-2

FRESNEL REFLECTION

Introduction

Article
Overview

Waves

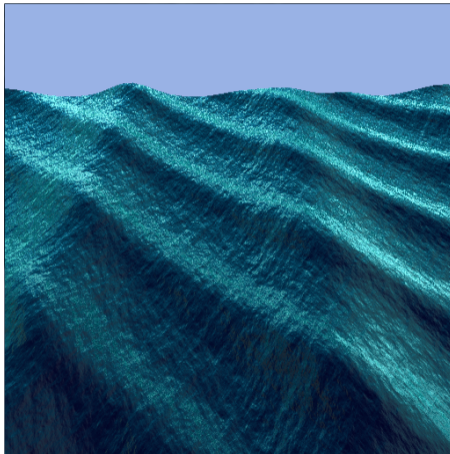
One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2



Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

- How much light reflects at a glancing angle
- (and how much refracts)

$$\text{fresnel} = R_0 + (1 - R_0) * (1 - \mathbf{V} \cdot \mathbf{n})^5$$

$$R_0 = \left(\frac{n_1 - n_2}{n_1 + n_2} \right)^2$$

- Air to water: $R_0 = 0.02037$
- `color = colorwater + reflection * fresnel`

FRESNEL REFRACTION

Introduction

Article
Overview

Waves

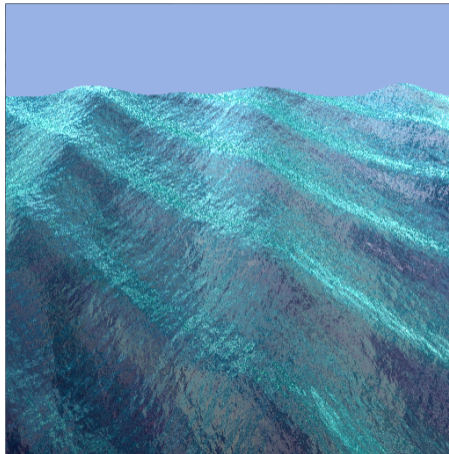
One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2



FRESNEL REFRACTION

- Use function `refract(I, N, eta)`
- Takes *incident* vector, I, normal N, and ratio of refraction indices, eta
- Air → water

$$\text{eta} = n_1/n_2 = 1.0/1.33$$

- Water → air

$$\text{eta} = n_2/n_1 = 1.33/1.0$$

- `color = colorwater + reflection * fresnel + refraction * (1 - fresnel)`
- Reflection and refraction are looked up in the cube map

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

Introduction

Article
Overview

Waves

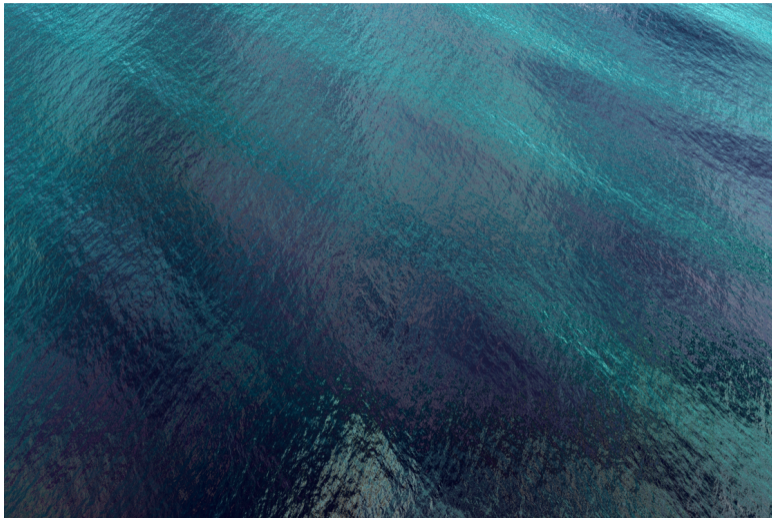
One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2



Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

Assignment 4

Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

- Expand `createQuad()`
 - Increase tessellation
 - Add texture coordinates
- Water shader
 - Use `diffuse.vert` and `diffuse.frag` as guidance
- Resources (`res/`):
 - `textures/waves.png`, normal map
 - `cubemaps/NissiBeach2/`, cube map set
- Files you have to modify (or add)
 - `src/EDAF80/assignment4.cpp`
 - `src/EDAF80/parametric_shapes.cpp`
 - `shaders/EDAF80/water.vert` (add)
 - `shaders/EDAF80/water.frag` (add)

Introduction

Article
Overview

Waves

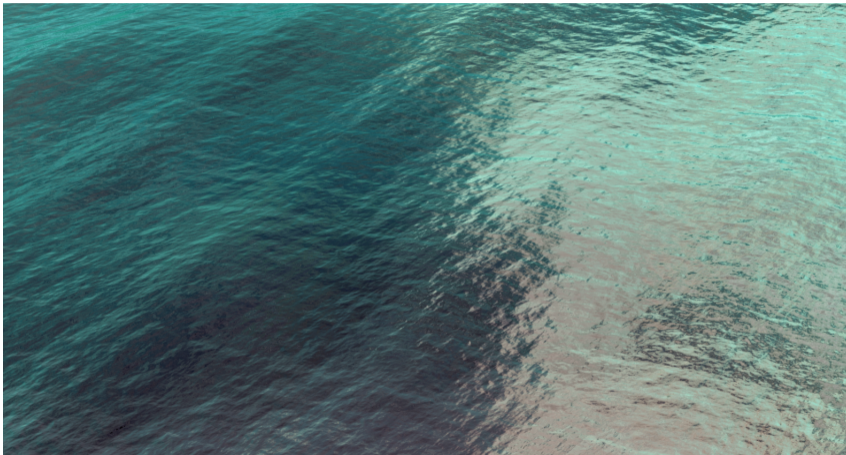
One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2



Introduction

Article
Overview

Waves

One wave
Sum of waves
▲ Exercise 4-1
GLSL functions

Shading

Water colour
Reflection
Animated normal
mapping
Tangent space
Fresnel reflection
Fresnel refraction
Final result

Assignment 4

Demo
▲ Exercise 4-2

Seminar Exercise 4-2: Procedural Vertex Animation

- 1 Replace the `checkboard()` function call in `main()` with `france()`. Then implement the Flag of France as a procedural texture.
- 2 Implement the function `sweden()` and have it generate the Flag of Sweden.
- 3 Come up with your own crazy procedural texture!