



dcc

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

ANIMACIÓN DEL MOVIMIENTO DE UNA SUPERFICIE FLUIDA

TAREA 3 – SHADERS

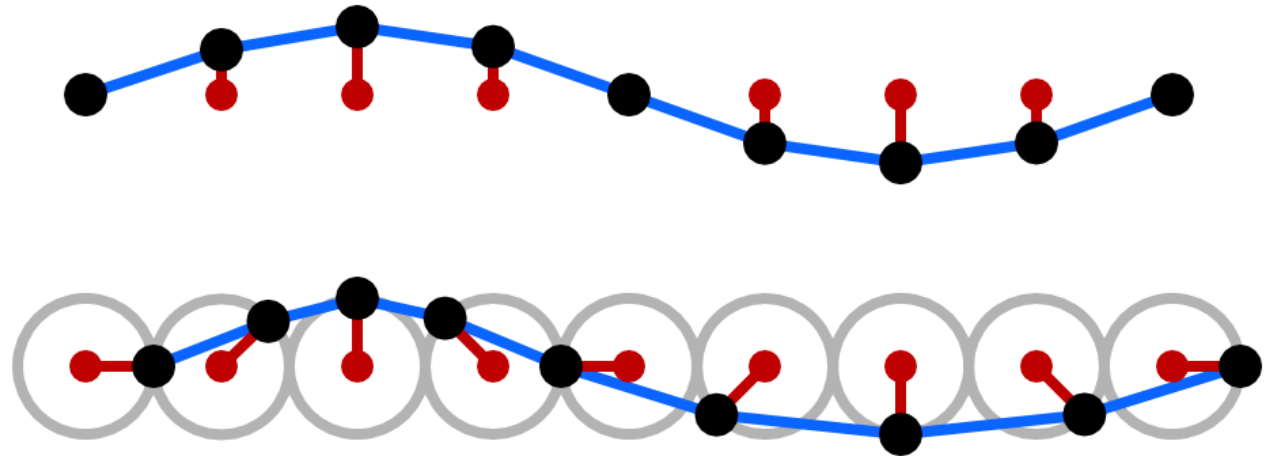
CC7515-I – COMPUTACIÓN EN GPU

PROFESORA: NANCY HITSCHFELD K

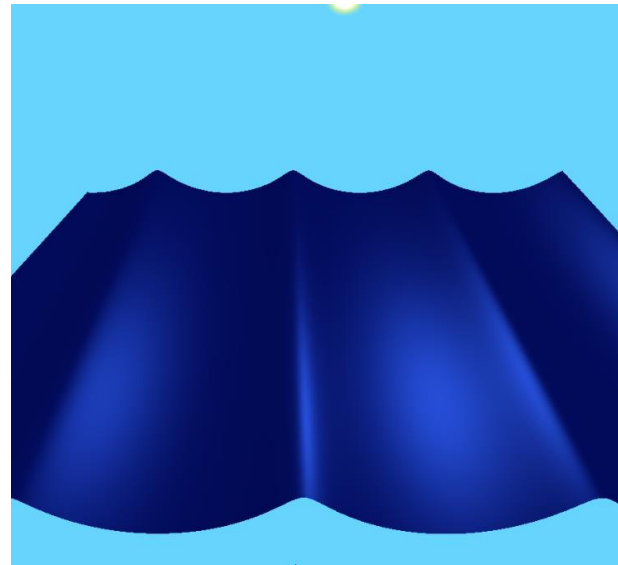
ALUMNO: SEBASTIÁN OLMOS H

GERSTNER WAVES

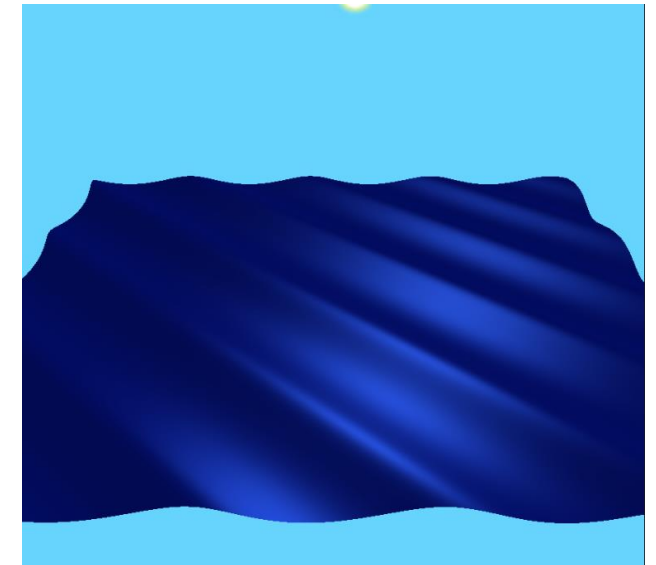
- Funciones seno y coseno en una dimensión son muy simples para simular una superficie fluida
- Se usa el método de Gerstner Waves para calcular oleaje en tiempo real sin ir a modelos más complejos como el modelo de Stokes
- Con este método no solo varía las alturas de los vértices, sino que también varían sus demás coordenadas en el espacio
- Se crea una malla simulando un plano para mover cada punto con el método escogido en tiempo real
- Se hace el cálculo de cada vértice en el Vertex Shader



Sine vs. Gerstner wave.



Simple wave en una dirección



Simple wave en ambas direcciones

ECUACIONES EN CADA VÉRTICE

Donde:

- S es el steepness que es una medida para controlar la amplitud del oleaje
- W es wavelength, medida de distancia entre cada ola
- G es la gravedad, de la cual depende la velocidad del oleaje
- P es el punto del espacio que tendrá el vértice a partir de sus coordenadas iniciales en la malla x, y
- D es un vector 2D con la dirección del oleaje
- t es el tiempo en el instante del calculo
- T y B son los vectores tangente y binormal, que sirven para calcular el vector normal N y entregárselo al fragment shader para los cálculos de iluminación

$$k = \frac{2\pi}{W} \quad c = \sqrt{\frac{G}{k}}$$

$$a = \frac{S}{k}$$

$$f = k \left(D \cdot \begin{bmatrix} x \\ y \end{bmatrix} - ct \right)$$

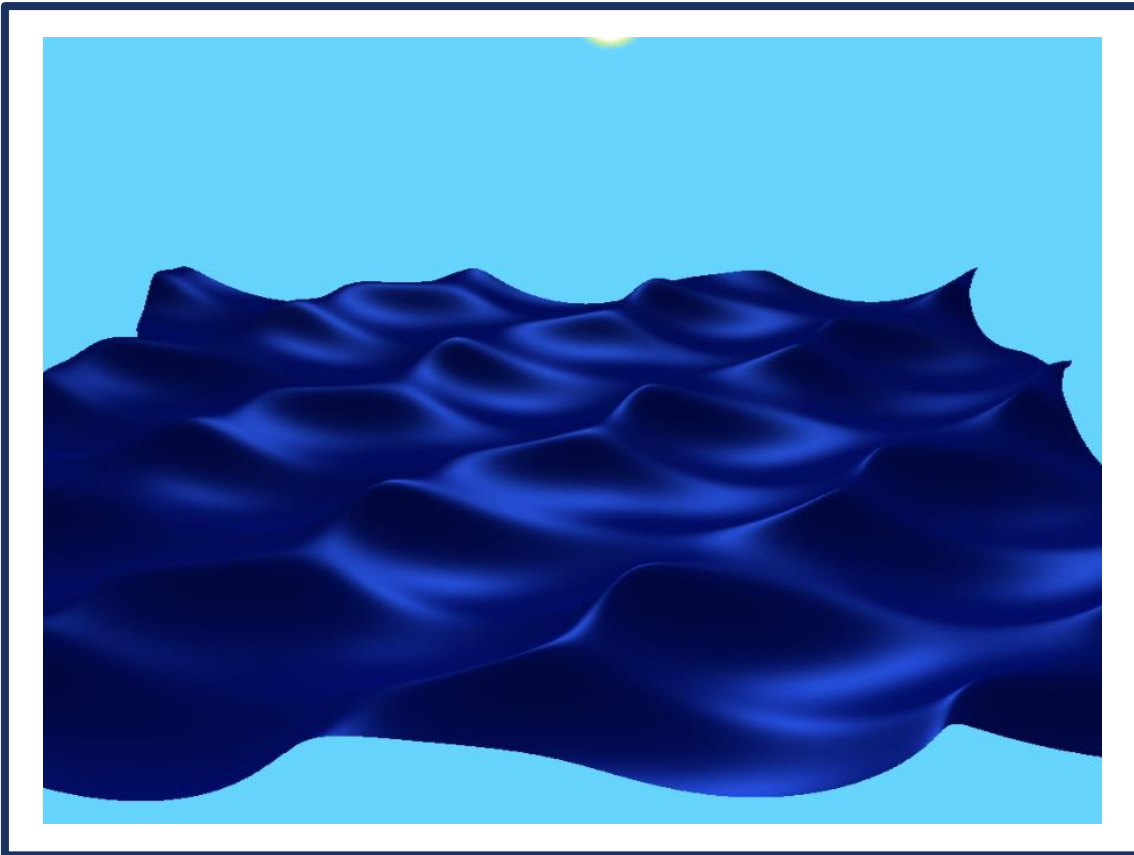
$$P = \begin{bmatrix} x + D_x a \cos f \\ y + D_y a \cos f \\ a \sin f \end{bmatrix}$$

$$T = \begin{bmatrix} 1 - (D_x)^2 S \sin f \\ -D_x D_y S \sin f \\ D_x S \cos f \end{bmatrix}$$

$$B = \begin{bmatrix} -D_x D_y S \sin f \\ 1 - (D_y)^2 S \sin f \\ D_y S \cos f \end{bmatrix}$$

$$N = \frac{T \times B}{\|T \times B\|}$$

MULTIPLE WAVES

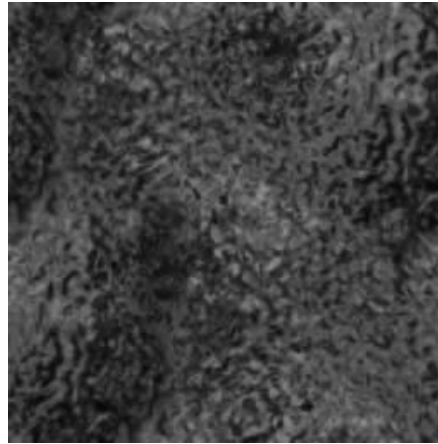


Suma de tres waves

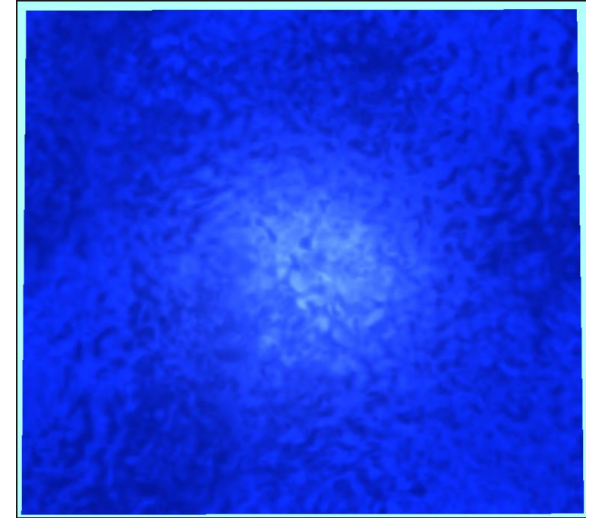
- El cálculo anterior simula el oleaje en una sola dirección
- Simulaciones realistas consideran la suma de varios oleajes en múltiples direcciones y con diferentes V y S .
- En esta aplicación se realizarán la suma de tres oleajes para permitir la total modificación de cada parámetro por parte del usuario

DESPLAZAMIENTO DE TEXTURAS

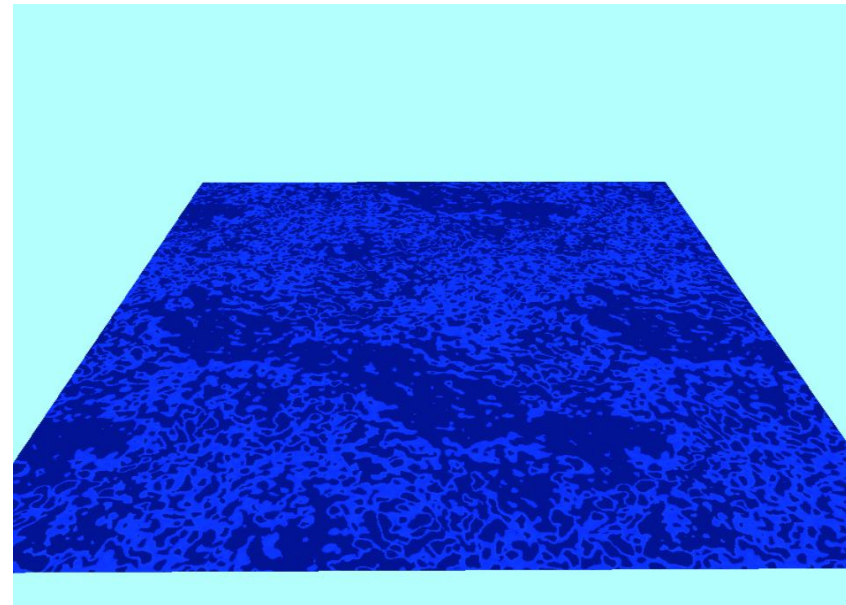
- Se usa una textura de ruido para añadir zonas del fluido de distinto color o simular la espuma generada en el mar.
- Se implementa en el Fragment Shader
- Se escala para controlar el nivel de detalle
- Se aplica un valor para la saturación y otro valor para desplazar la textura a una cierta velocidad constante por la malla.



Textura de ruido para desplazamiento

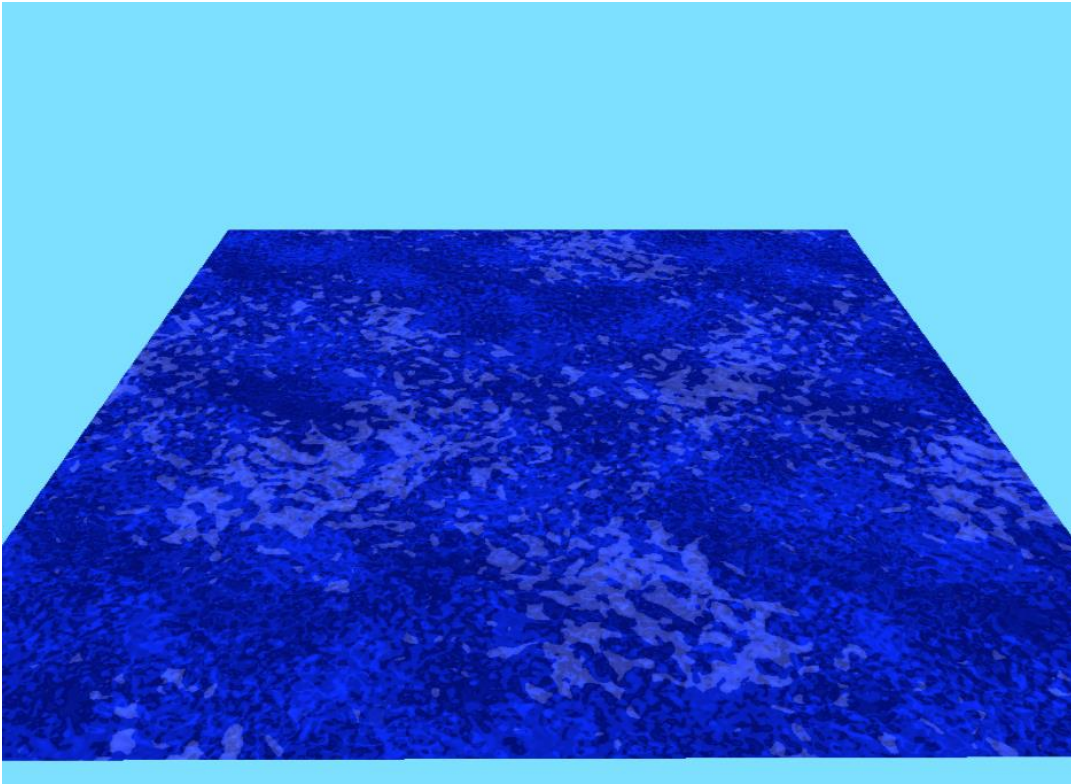


Ruido simple implementado



Ruido escalado, limitado y saturado

MÚLTIPLE DESPLAZAMIENTOS

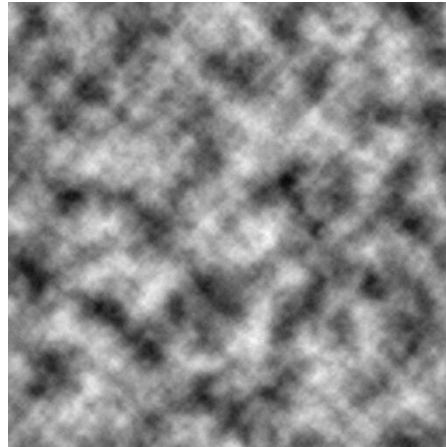


Tres capas de desplazamiento con diferentes configuraciones

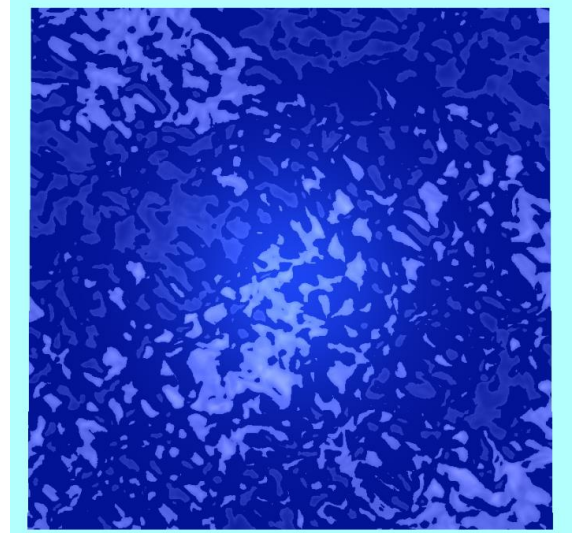
- Se usan tres capas de texturas con desplazamiento para simular distintos escenarios
- De la misma manera que con el oleaje cada capa con sus configuraciones queda al control del usuario

DISTORSIÓN DE TEXTURAS

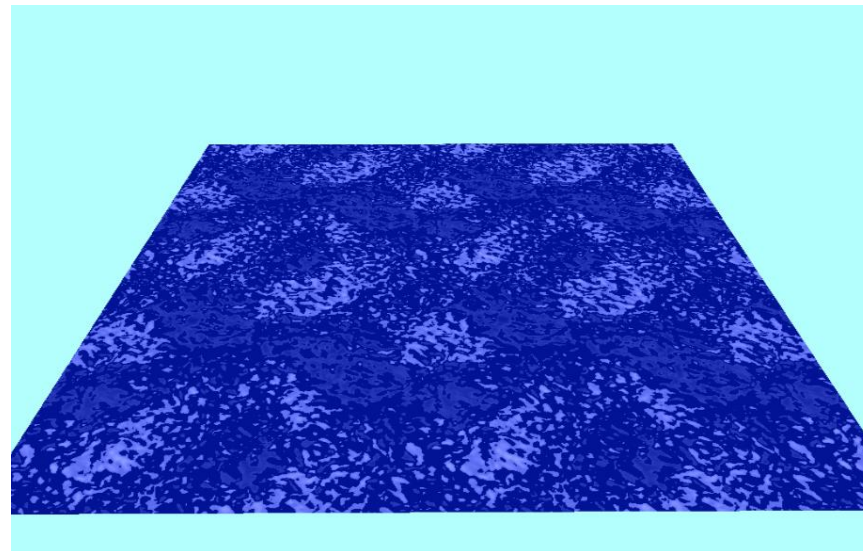
- Se usa una segunda textura para simular una distorsión a las capas desplazadas anteriormente
- Simula la naturaleza ondulatoria de los fluidos
- De la textura de ruido aplica una desplazamiento a la coordenada UV que se usa para leer la primera textura
- También se aplica un desplazamiento a esta textura, generando una distorsión diferente en cada instante de tiempo



Textura de ruido para la distorsión

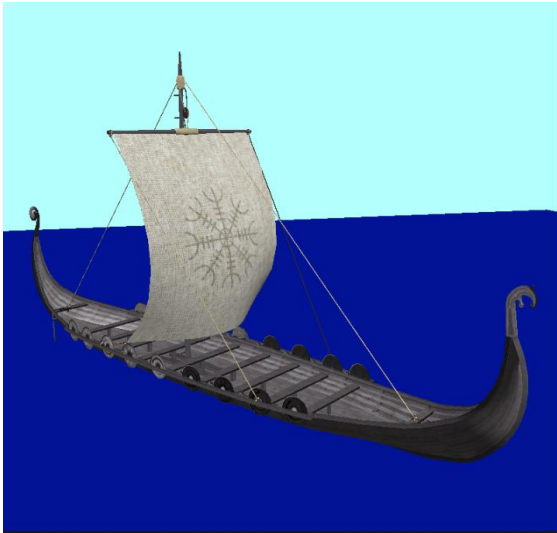


Distorsión sobre una capa de textura desplazada

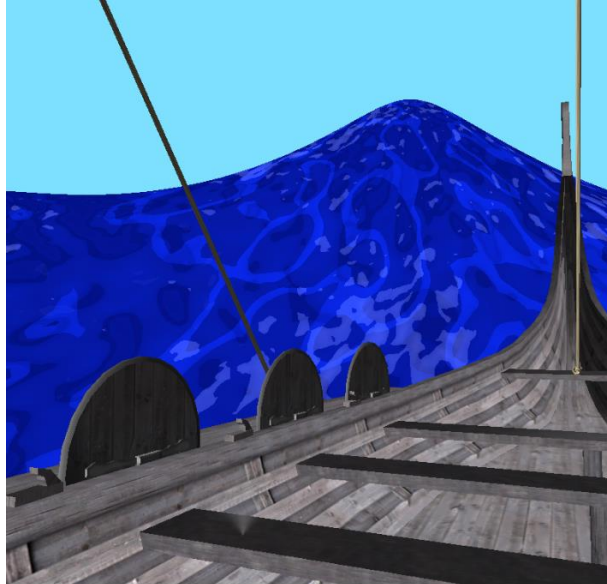


Distorsión escalada y desplazada sobre una textura

BARCO



Modelo de barco

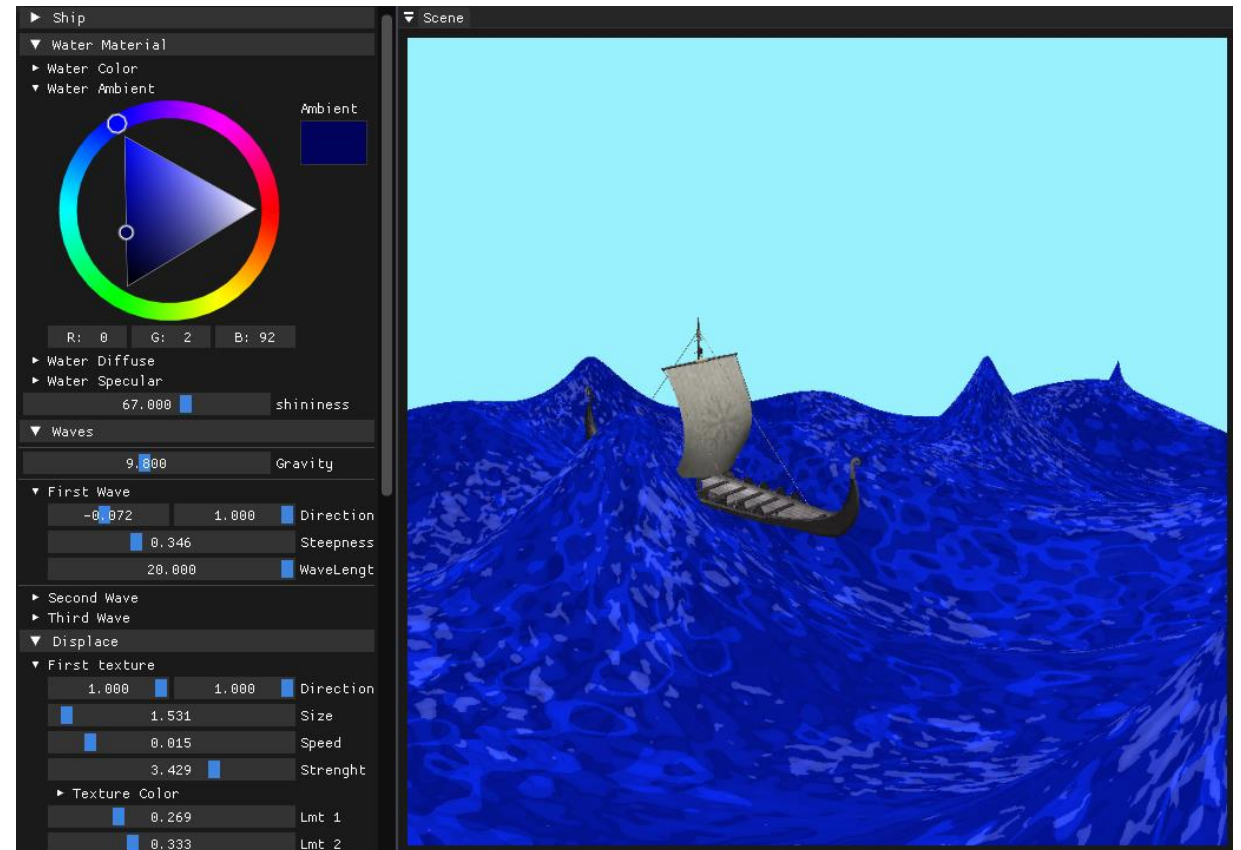


Vista desde barco

- Se carga el modelo de un barco vikingo con la librería Assimp
- Se consideran textura para aplicar color difuso y una mapa de normales para aplicar bump map
- Se realiza el cálculo del oleaje en CPU en la posición del barco para simular el efecto del oleaje sobre el.
- Su posición, escala y rotación queda a control del usuario
- También se incluye un botón para poder visualizar la escena desde dentro del barco.

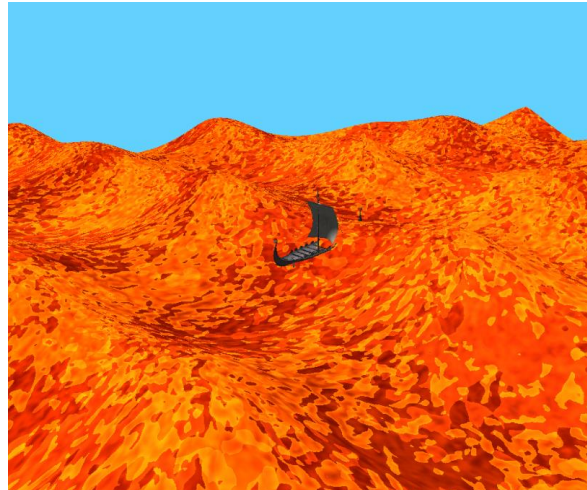
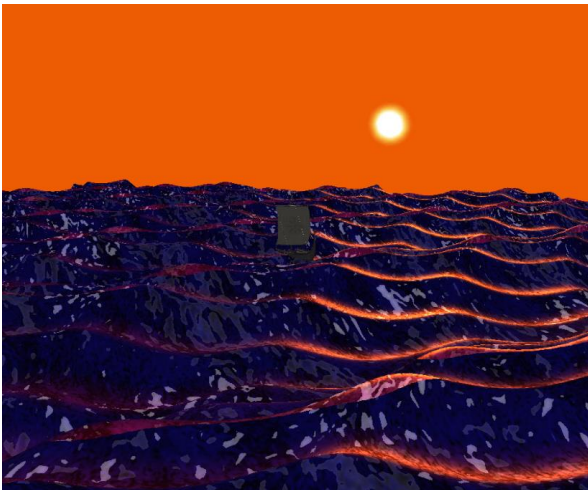
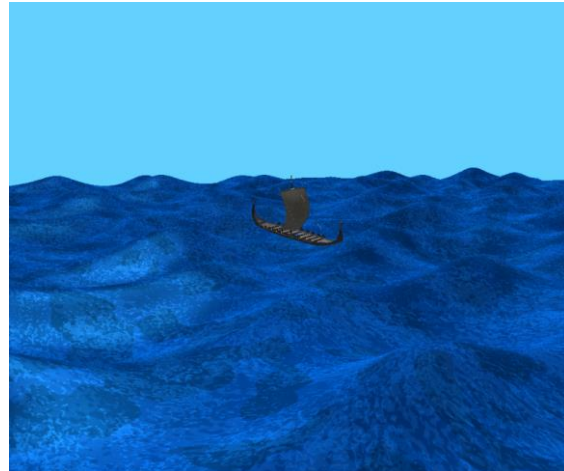
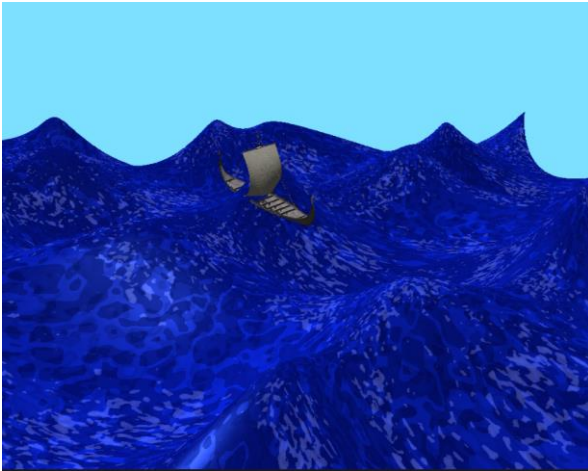
IMGUI COMO INPUT

- Se usa la librería ImGui para poder controlar todos los parámetros anteriormente descritos
- Se encuentra en el lado izquierdo de la aplicación
- También se implementa técnicas de docking para poder mover la ventana de la escena y poder cambiar su tamaño



Aplicación con menú desplegado

PERSONALIZACIÓN DE ESCENAS



- Con la gran cantidad de parámetros que controlan la escena se pueden generar distintos tipos de visualizaciones.
- Dentro del menú se dejan 4 escenas predeterminadas que corresponden a las señaladas.