

PROYECTO III – EFECTO DE OLA

El proyecto consiste en programar la simulación de dos olas, siguiendo la ecuación 2, del siguiente link:

http://http.developer.nvidia.com/GPUGems/gpugems_ch01.html

Tengan en cuenta que una ola depende de las siguientes variables, que deberán ser proporcionadas por el usuario:

- a. L = distancia entre cada ola.
- b. A = altura de la ola.
- c. S = velocidad de la ola.
- d. D = vector de dos coordenadas que determina la dirección de la ola.

Además en la ecuación de la ola, existen variables que se crean a partir de los datos de entrada.

Por ejemplo, $W = 2\pi / L$, así como otras variables. Por lo tanto vean la página web con cuidado para ver cómo emplear cada variable.

En la ecuación 2, verán un símbolo de sumatoria, esta fórmula está diseñada, para una **n** cantidad de olas. **Solamente harán 2.**

Es decir, debe haber un L1, L2, A1, A2, S1, S2, D1 y D2.

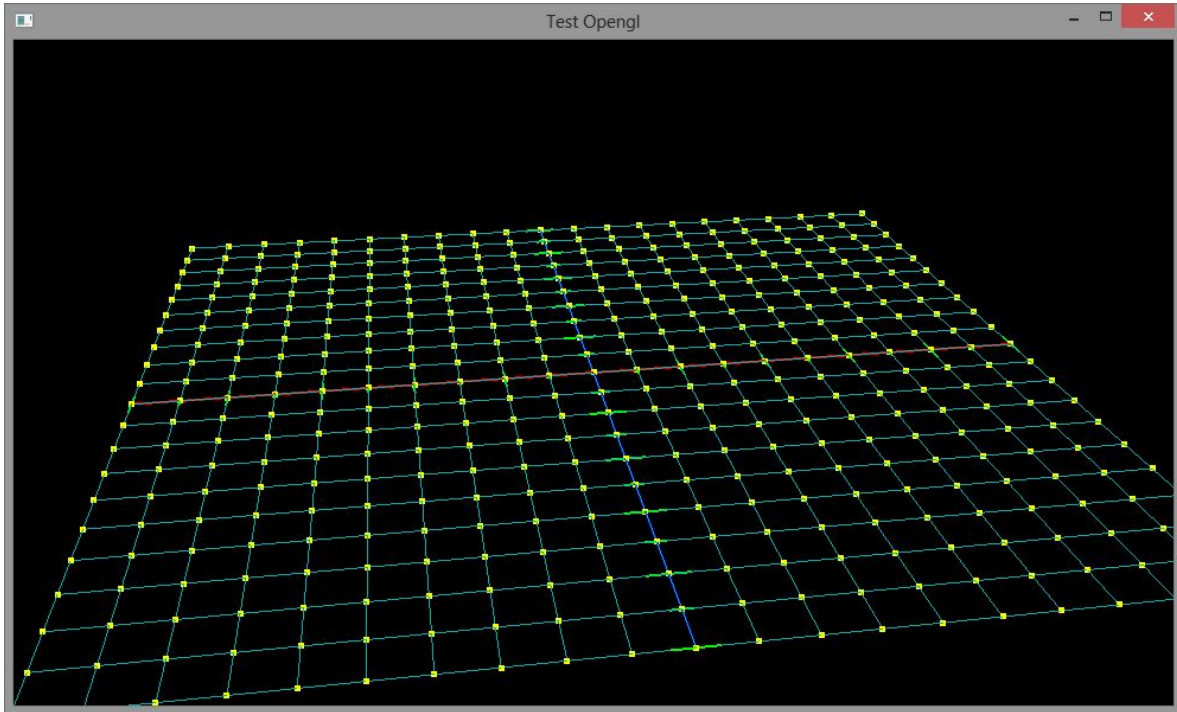
Equation 2

$$H(x, y, t) = \sum \left(A_i \times \sin \left(\mathbf{D}_i \cdot (x, y) \times w_i + t \times \varphi_i \right) \right),$$

Producto escalar

Multipliación
normal

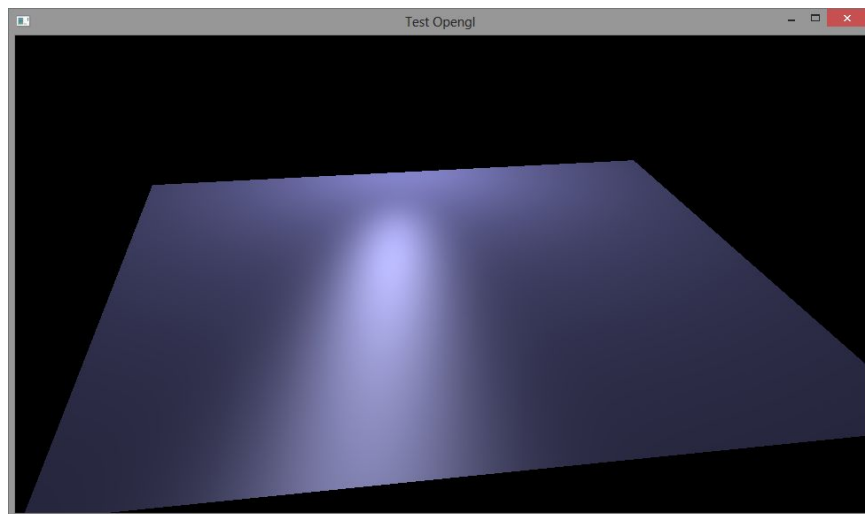
Las olas se deberán crear en un plano Nurbs de 21x21 puntos de control, estos puntos de control deben ir desde -10 al 10 del eje X y Z respectivamente.



CONSEJO: antes de intentar de dibujar la superficie NURBS asegúrense que los puntos estén ubicados donde deben.

Programa.

1. Al iniciar el programa se deberá ver el plano sin ningún tipo de altura.



Traten de no cambiar la cámara y la luz, la cámara fue colocada de tal manera de ver bien el efecto de las olas y la luz fue colocada convenientemente para crear un brillo en las olas.

2. Al presionar la tecla “r” empezara la animación de las olas.
3. Al presionar la tecla “p” se hace pausa de la animación y el plano mantiene la forma que tenía antes de presionar la tecla, para volver a correr la animación se presiona “r”.
4. El usuario podrá cambiar los valores de la ola de la siguiente manera:
 - a. Si el usuario presiona la tecla 1, modificara las variables L, A, S, D de la siguiente manera:
 - i. Tecla (a) $\rightarrow L = L - 0.1$
 - ii. Tecla (z) $\rightarrow L = L + 0.1$
 - iii. Tecla (s) $\rightarrow A = A - 0.1$
 - iv. Tecla (x) $\rightarrow A = A + 0.1$
 - v. Tecla (d) $\rightarrow S = S - 0.1$
 - vi. Tecla (c) $\rightarrow S = S + 0.1$
 - vii. Tecla (f) $\rightarrow D.x = D.x - 0.1$
 - viii. Tecla (v) $\rightarrow D.x = D.x + 0.1$
 - ix. Tecla (g) $\rightarrow D.y = D.y - 0.1$
 - x. Tecla (b) $\rightarrow D.y = D.y + 0.1$

Coloque 0.1, sin embargo pueden colocar algún otro valor, según su performance.

Si el usuario presiona 2, las mismas teclas alterarían las variables correspondientes a la segunda ola.

5. Recuerden que deben evaluar la función $H(X,Y,t)$ en los puntos de control originales, el valor que se obtiene se debe almacenar en la altura del punto de control que es el eje Y, la variable t es una variable que deben incrementar durante la animación, ejemplo:
 - a. $t = t + 0.1$
6. Esto es una función $Y = f(x,z)$, es decir, los puntos x e z no se alteran, solo la altura del punto de control.
7. El vector director D1 y D2 de cada ola, debe ser normalizado antes de usarlo en la ecuación, de lo contrario cuando muevan la dirección tendrán una animación errada.
8. Dependiendo de algunos valores extremos, el sistema puede comportarse en forma extraña, esto se debe a que no existe la suficiente resolución en el plano

para trabajar ciertos parámetros, por ejemplo, un valor de L muy pequeño causaría un movimiento raro.

A tomar en cuenta:

1. En la página web, el eje Z es la altura, por esa razón las fórmulas son del tipo $f(x,y) = Z$. Tengan en cuenta que en OpenGL la altura es el Y, por lo que la formula seria $f(x,z) = Y$.
2. Recuerden que los puntos de control que son 21x21, deben ir desde $X=-10..10$ hasta $Z=-10..10$.
3. Cada punto tiene 3 coordenadas de ahí que tienen que hacer una matriz tridimensional 21x21x3, donde cada capa de la profundidad de la matriz corresponde el X,Y,Z del punto en cuestión.

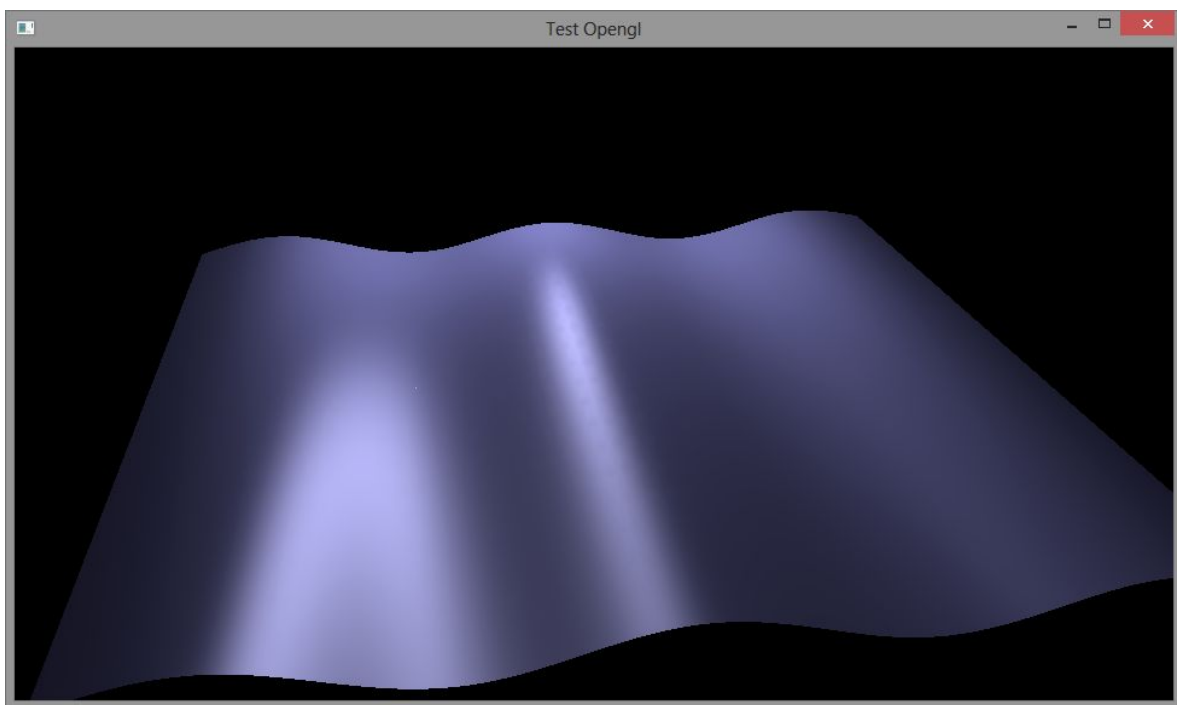
Especificaciones del proyecto:

1. El proyecto es en pareja..
2. El proyecto tiene un 12% de nota.
3. La fecha de entrega es hasta el Domingo 29/05, 11:59pm.
4. Deben enviarme por correo (comprimido en NombreEstudianteYNombreEstudiante.rar) el código fuente (solo los archivos .cpp, no es necesario comprimir todo el proyecto), el ejecutable.
5. **IMPORTANTE:** el asunto del correo debe estar escrito con el siguiente formato: "Graficas I 2016 - Proyecto III - OpenGL - NombreEstudianteYNombreEstudiante".

Valores de prueba:

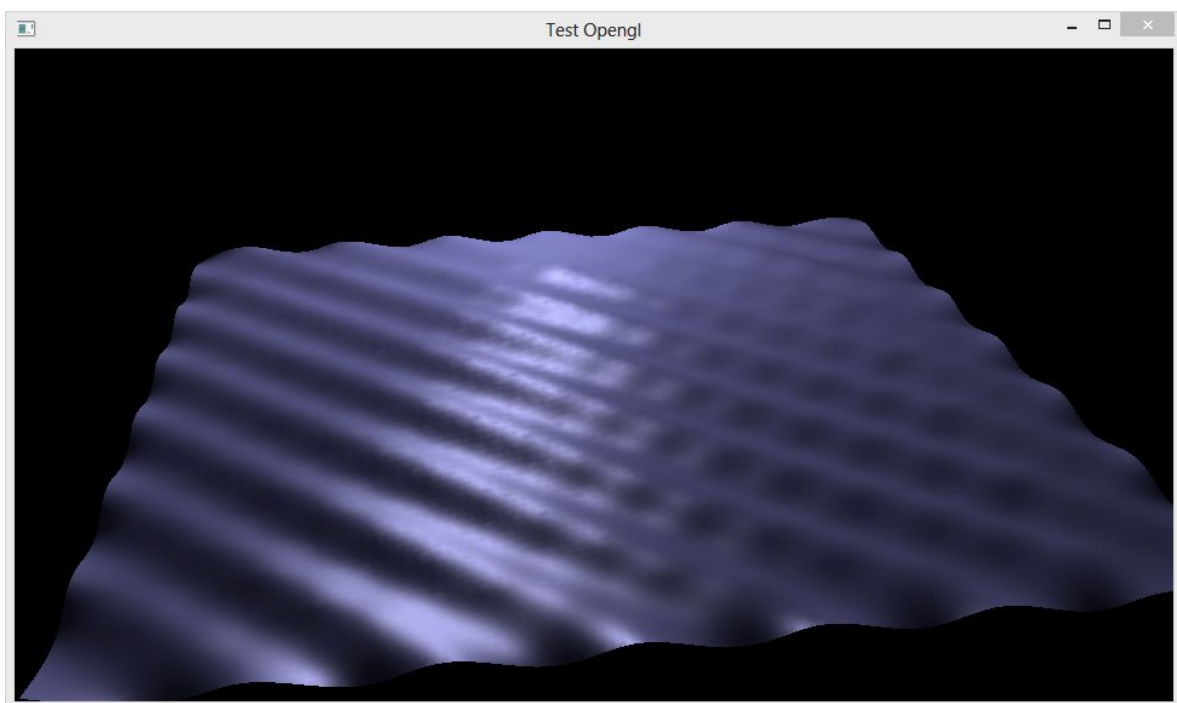
```
wL_01 = 8.0;  
aP_01 = 0.4;  
sP_01 = 2.0;  
dirX_01 = 0.0;  
dirY_01 = -1.0;
```

```
wL_02 = 4.0;  
aP_02 = 0.0;  
sP_02 = 0.0;  
dirX_02 = 1.0;  
dirY_02 = 1.0;
```



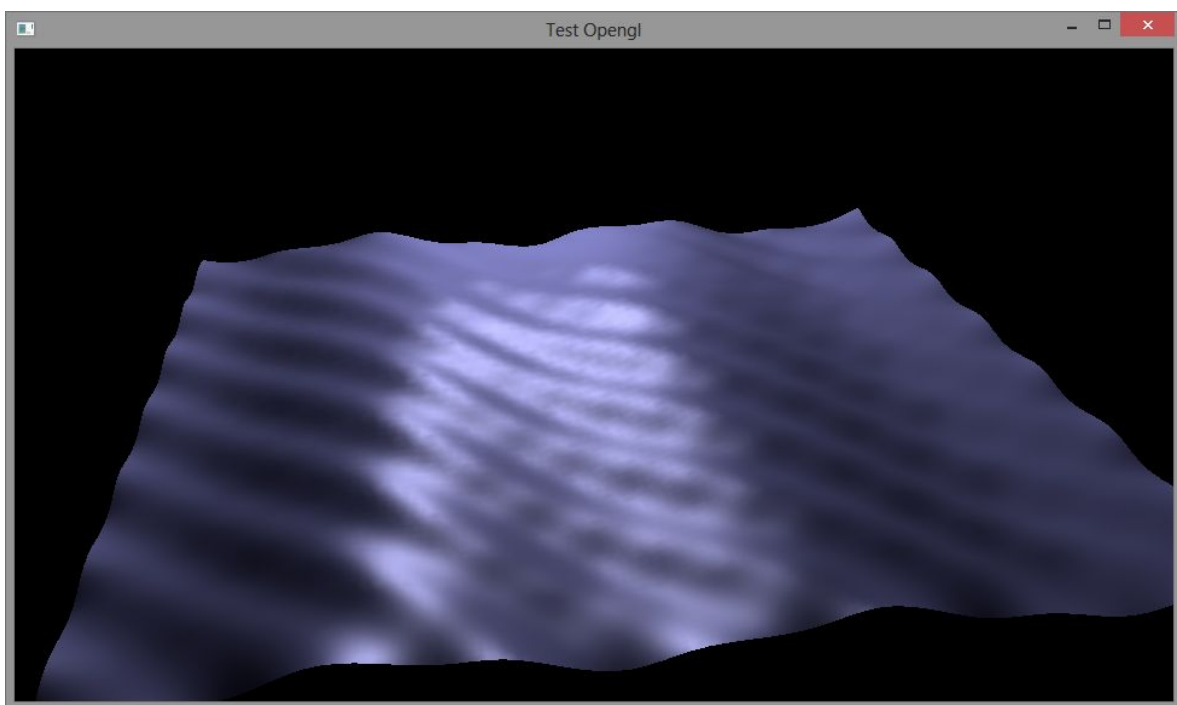
```
wL_01 = 8.0;  
aP_01 = 0.0;  
sP_01 = 2.0;  
dirX_01 = 0.0;  
dirY_01 = -1.0;
```

```
wL_02 = 3.0;  
aP_02 = 0.3;  
sP_02 = 1.0;  
dirX_02 = 1.0;  
dirY_02 = 1.0;
```



```
wL_01 = 8.0;  
aP_01 = 0.3;  
sP_01 = 0.5;  
dirX_01 = 0.0;  
dirY_01 = -1.0;
```

```
wL_02 = 3.0;  
aP_02 = 0.2;  
sP_02 = 1.0;  
dirX_02 = 1.0;  
dirY_02 = 1.0;
```



```
wL_01 = 20.0;  
aP_01 = 0.75;  
sP_01 = 3.0;  
dirX_01 = 0.5;  
dirY_01 = 1.0;
```

```
wL_02 = 20.0;  
aP_02 = 0.4;  
sP_02 = 3.0;  
dirX_02 = 1.0;  
dirY_02 = 0.4;
```

