Universidad de Chile
FCFM - DCC
CC3501 **Reporte Tarea 2c** 20 de junio de 2021

Solución Propuesta

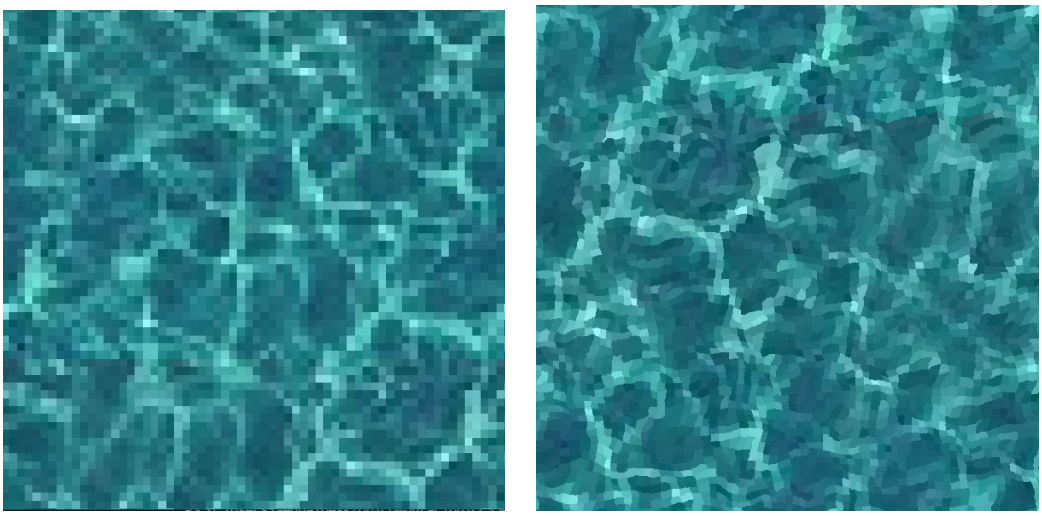


Figura 1: Textura original a la izquierda, y aplicando el displacement map a la derecha

Los archivos displacement_view.py, water_slide.py, models.py, shapes.py, custom_curves.py y custom_shapes.py fueron creados desde cero para facilitar el trabajo, exceptuando algunas funciones que se indican en los propios archivos. Los archivos newLightShaders.py y custom_GPUshape.py son modificaciones de archivos entregados por el profesor o los auxiliares.

Para la primera parte de la tarea, se crea un shader personalizado que recibe dos texturas para utilizar una como displacement map, y la otra para ser visualizada, permite activar o desactivar el displacement map. Estas modificaciones se realizan en el fragment shader, donde el color de cada pixel del displacement map afecta a la posición de un pixel de la textura del agua. Las coordenadas usadas en el fragment shader además se trasladan en diagonal dos floats entregados en uniform para simular el movimiento de la corriente.

La segunda parte de la tarea utiliza en primer lugar un skybox dentro del cual se encuentran dos toboganes. En custom_curves se encuentran funciones que, entre otras cosas, crean una curva de Catmull-Rom de una cantidad de puntos arbitraria. A partir de esto se crean dos curvas y luego se utilizan como base para darle la forma a los toboganes con la función createTube de shapes. Ambos toboganes forman parte de un grafo de escena.

El tobogán 1 está formado con la función createHalfTubeMesh de custom_shapes, por una malla de polígonos consistente en múltiples vértices con forma de semicírculo alrededor de cada punto de la curva usada, dichos semicírculos se encuentran en planos paralelos, por lo que no admiten giros mayores a 90°. El tobogán 2 está formado similarmente con la función createHalfTubeMeshRot de custom_shapes, que crea todos los semicírculos en el origen, para luego rotarlos según la tangente de la curva y trasladarlos con transformaciones en CPU, generando una malla que si admite giros cerrados. La tangente de cada punto es calculada con la pendiente del punto anterior y siguiente de la curva, $m = \frac{x_{i+1}-x_{i-1}}{y_{i+1}-y_{i-1}}$, y el ángulo

adecuado es la inversa de la tangente.

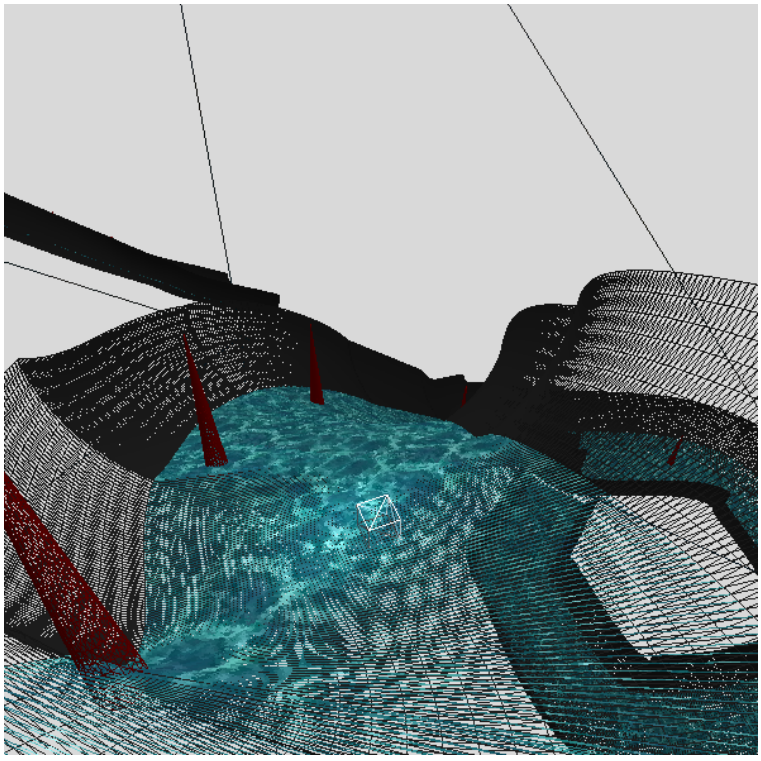
El agua del tobogán es una malla de polígonos que usa vértices a alturas similares de la malla del tobogán, con una textura y un displacement map aplicados, se usa también otro shader personalizado para crear el efecto en un entorno 3D con iluminación local.

Los obstáculos son mallas de polígonos que describen conos de cierta altura, estos se posicionan a lo largo de la curva con separaciones similares entre si, para evitar escenarios imposibles de ganar. Sus posiciones en la sección perpendicular son aleatorias dada una interpolación lineal. Todos los obstáculos se encuentran en un grafo de escena como hijos de un nodo.

Las normales de todas las mallas de polígonos fueron calculadas con una función de openmesh, request_vertex_normals().

El jugador sigue en general a la curva del respectivo tobogán, pero para los movimientos de la sección transversal del tobogán se mueve mediante una interpolación lineal entre los límites de la izquierda y la derecha. Si el jugador colisiona con algún obstáculo, su posición en la curva se resetea a la inicial. Todos los movimientos se realizan mediante transformaciones en GPU.

La cámara sigue la misma curva que el jugador, pero con un poco de retraso, y con cierta elevación.



Instrucciones de Ejecución

- Librerias adicionales a OpenGL utilizadas: sys, os.path, math, random, numpy, PIL, openmesh
- Métodos de ejecución:
  - python displacement_map.py
  - python water_slide.py N V
- Modo de uso y/o teclas de control: Up, Left, Right, 1, 2, Q, Espacio, Escape

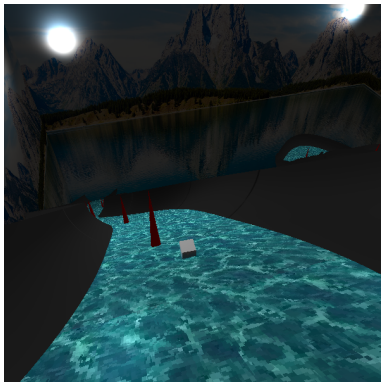
En la primera parte de la tarea, la tecla Q activa y desactiva el diplacement map.

En la segunda parte, N es la cantidad de obstáculos a posicionar a lo largo de cada tobogán, y V es la velocidad a la que avanza el jugador por el tobogán. Se recomiendan valores cercanos a 10 para N, y 15 para V.

El jugador comienza a moverse con la tecla arriba, y se mueve de lado a lado con las teclas izquierda y derecha. Las teclas 1 y 2 cambian la posición de la cámara entre ambos jugadores (hay un jugador en cada tobogán). La tecla Q hace visible la curva que genera a cada tobogán, y la barra espaciadora alterna entre rellenar los polígonos o dibujar solo sus aristas. El juego se cierra con el botón “Escape”

Resultados

En la segunda parte de la tarea, se crean dos curvas, y a partir de estas, dos toboganes hechos con mallas de polígonos. Se posicionan dos jugadores en cada tobogán, y la cantidad de obstáculos especificada a lo largo de cada malla. Se genera un río que fluye por el tobogán, el efecto resultó distinto al de la primera parte. Al presionar la tecla arriba, el jugador comienza a moverse hasta llegar al final de la curva, donde se queda detenido hasta terminar el programa.



Autoevaluación

Criterio-Puntaje	0	1	2	3
OpenGL			x	
Shaders			x	
Modelos geométricos			x	
Transformaciones			x	
Texturas			x	
Modelación jerárquica		x		
Curvas			x	
Vistas y Proyecciones			x	
Iluminación Local		x		
Mallas de Polígonos			x	
Funcionalidades mecánicas o lógica de juego				x
Entradas o Control de usuario			x	
Visualización de estado del programa			x	