

Tarea N°2

Prof: Daniel Calderón
Auxiliares: Pablo Pizarro, Diego Donoso

Instrucciones Generales

- La tarea es individual.
- Este enunciado presenta dos opciones, escoja e implemente solo una de ellas.
- El plazo de entrega se indica en tareas/u-cursos.
- Esta tarea debe ser trabajada y entregada en el repositorio privado git bitbucket utilizado en la primera tarea. Instrucciones detalladas en guía-git-bitbucket.pdf disponible en material docente.
- Guarde todo su trabajo para esta tarea en una carpeta de nombre **tarea2x**, con x indicando la opción que haya escogido.
- DEBE utilizar: Python 3.5 (o superior), OpenGL core profile, GLFW.

Entregables

- Reporte de documentación (1-2 planas). Formato pdf. Detalles disponibles en primera clase. (2 %)
- Código que implemente su solución. Incluya TODO lo que su código necesita, incluyendo archivos proporcionados en clases (Los cuales pueden ser modificados si ud. lo requiere). (9 %)
- Al menos 10 commits en su repositorio git remoto ilustrando progreso en su trabajo. (1 %)
- Video demostrativo de 20-30 segundos. (0.5 %)
- Todo lo anterior debe estar disponible en su repositorio git bitbucket remoto.
- Puntaje total: 12.5 % del curso.

Objetivo

- Ejercitar el uso de OpenGL core profile en una aplicación simple en 3 dimensiones.
- Aplicar conceptos modelado de objetos en 3D.
- Entender el uso de las texturas y de la cámara.
- Implementación de una aplicación sencilla.

Tarea 2.A - Barco Simulator

El objetivo de esta tarea es implementar un simulador de un barco, el cual debe recorrer un determinado camino alrededor de una paradisíaca isla en medio del mar. El camino es ingresado al simulador como argumento en la terminal de comandos (Ingresando un archivo de texto que contiene las coordenadas), en donde se conectan mediante una curva suave. La isla puede ser generada de cualquier manera, aunque debe cambiar de color en función de su altura, tal como se muestra en la Figura 1.

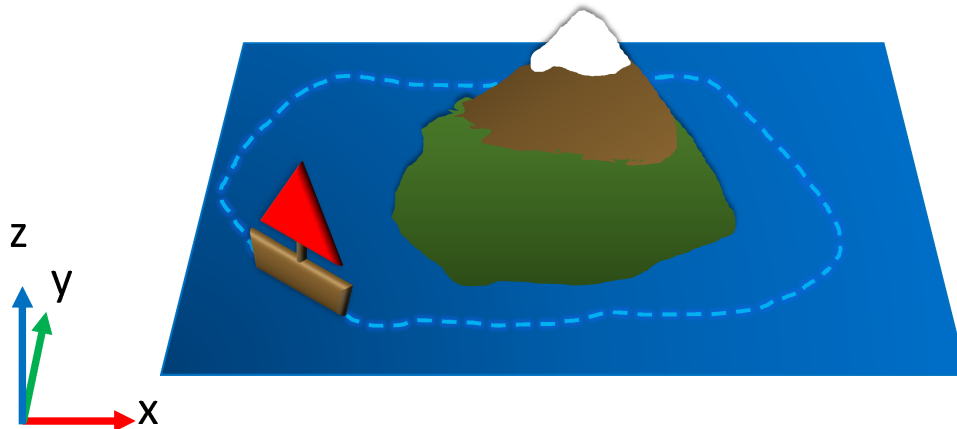


Figura 1: La idealización del simulador en todo su esplendor.

Requisitos gráficos

- El mar debe ser representado usando un plano en 2D.
- El barco debe ser construido utilizando al menos cuatro elementos distintos en 3D.
- El cerro puede ser modelado de cualquier manera, siempre y cuando el color del terreno varíe en función de su altura z ; puede usar una interpolación o bien una escala discreta como el siguiente ejemplo:

$$\text{Color terreno} = \begin{cases} \text{Verde} & z < 0.5 \\ \text{Café} & 0.5 \leq z \leq 0.8 \\ \text{Blanco} & 0.8 < z \leq 1 \end{cases}$$

- Se deben usar al menos tres texturas para colorear los modelos, elija ud. qué puede llevar textura: El barco, agua, cielo, nubes, detalles de la isla, etc.
- La trayectoria que seguirá el barco está definida a través de la interpolación dada por splines de Catmull-Rom sobre los puntos dados por el usuario.
- El modelo se debe representar en un mundo de tamaño $2x2$, en donde la isla está centrada y posee un tamaño máximo de $1x1$. El cerro debe tener una altura máxima de 1 unidad. El barco puede tener el tamaño que usted estime apropiado. La Figura 2 ilustra un ejemplo del modelo:

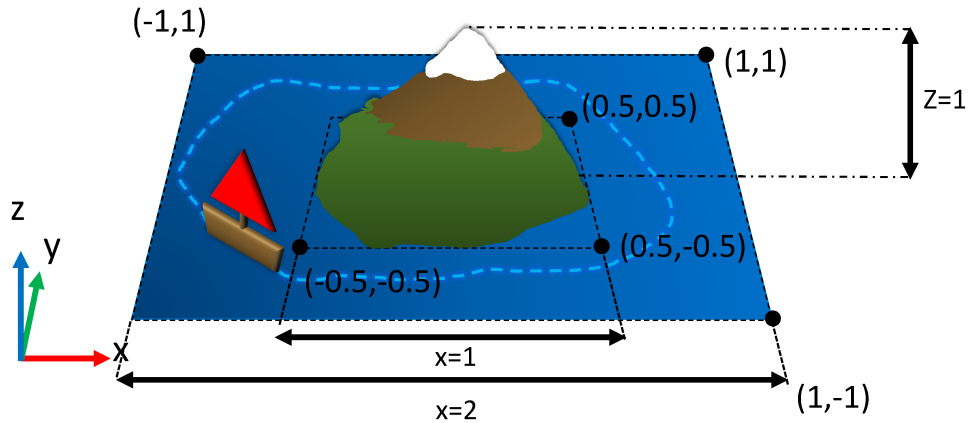


Figura 2: Propiedades gráficas del modelo.

Requisitos del programa

- El programa debe leer un archivo que contiene las coordenadas en las que navegará el barco. Este solo puede desplazarse a través del agua (Escoja puntos apropiados). La llamada al programa debe cumplir con el siguiente formato:

```
1 py -3 simulator.py trayectoria.txt
```

Código 1: Ejemplo de trayectoria.txt, puntos referidos al plano (x,y).

```
1 -0.9, -0.9
2 -0.7, 0.2
3 -0.1, 0.6
4 0.4, 0.8
5 0.9, 0.9
6 0.6, 0
7 0.4, -0.5
```

- Por simplicidad, no se preocupe por posibles colisiones entre el barco y la isla.
- El tiempo total de navegación debe ser de aproximadamente 2 segundos entre cada par de puntos.
- Configure estratégicamente 4 cámaras en torno a la escena. Las teclas 1, 2, 3 y 4 deben acceder a dichas cámaras.
- Con la tecla **ESPACIO** se puede alternar la visualización de la trayectoria.

Distribución de puntaje

- Modelo (3D): 5 puntos.
- Texturado: 1 punto.
- Interacción con el usuario (Curvas+Input): 2 puntos.
- Configuración de las cámaras: 1 punto.

Tarea 2.B - Visualizador de Edificios

El objetivo de esta tarea es implementar un visualizador de distintos edificios icónicos del mundo, con el objetivo de servir como un programa interactivo para la importante conferencia de arquitectura que se desarrollará en Chile. La escena posee un edificio en la mitad de la escena, un fondo y una cámara móvil alrededor del edificio.

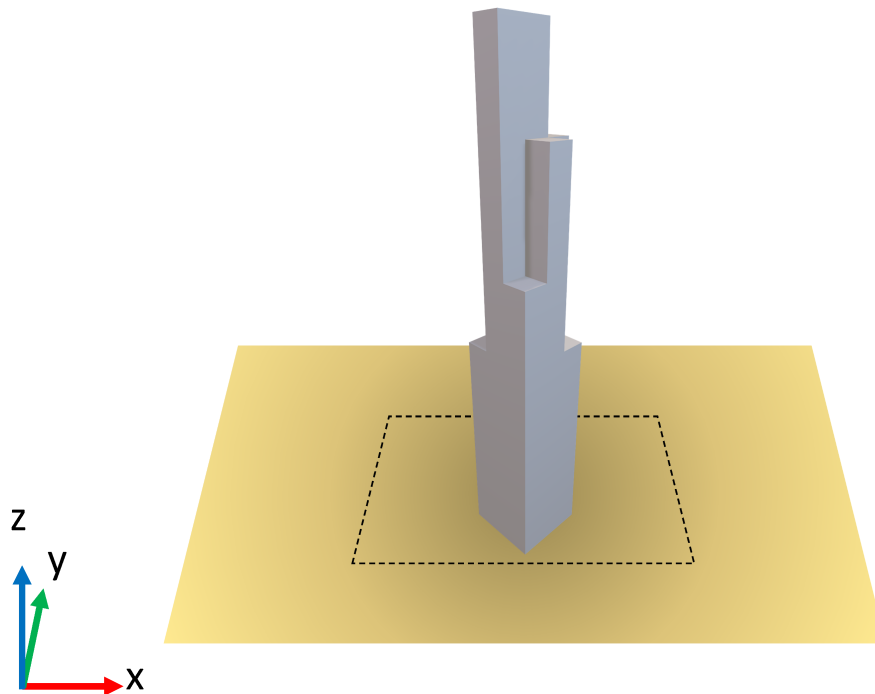


Figura 3: Visualizador de edificios.

Requisitos gráficos

- Se debe modelar sólo uno de los siguientes edificios:
 1. Empire State Building, New York, Estados Unidos.
 2. Willis Tower, Chicago, Estados Unidos.
 3. Titanium La Portada, Las Condes, Chile.
 4. Burj Al Arab, Dubái, Emiratos Árabes Unidos.
 5. Burj Khalifa, Dubái, Emiratos Árabes Unidos.
 6. Torres Petronas, Kuala Lumpur, Malasia.
 7. Taipei 101, Taipéi, Taiwan.
 8. Bank of China Tower, Hong Kong, China.
 9. Torre Eiffel, Paris, Francia.
 10. Estrella de la Muerte, En algún lugar de la galaxia, Star Wars.

- El suelo o superficie de apoyo del edificio debe ser un plano 2D con alguna textura de su agrado.
- El modelo se debe representar en un mundo de tamaño $2x2$, en donde el edificio está centrado y posee una dimensión en planta (Plano xy) máximo de $1x1$, la altura de la estructura puede ser cualquiera, la que estime conveniente. La siguiente figura ilustra una parametrización del modelo:

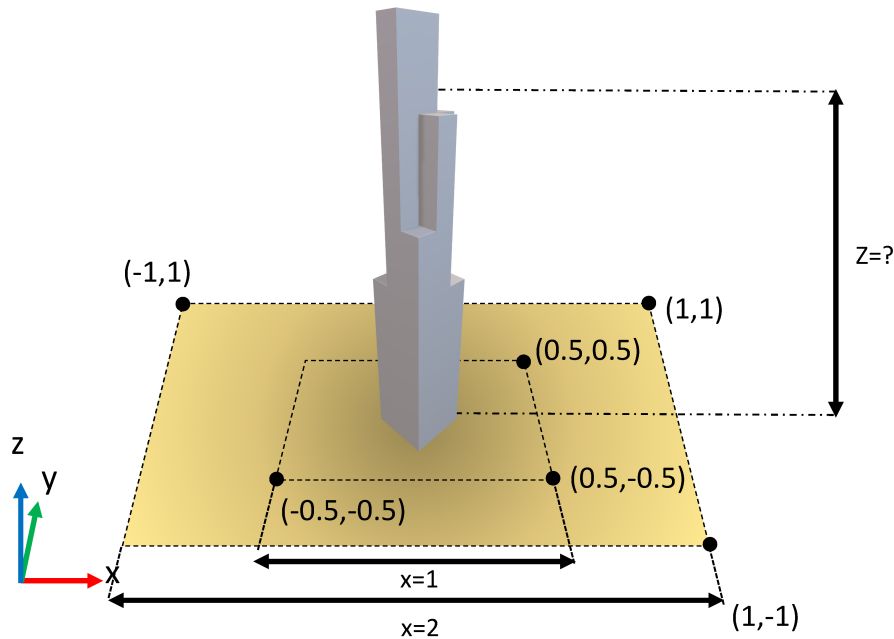


Figura 4: Propiedades gráficas del modelo.

- Los edificios deben ser modelados utilizando las metodologías vistas de OpenGL 3D, utilizando modelos simples y transformaciones o descripciones punto a punto. Debe utilizar además al menos cinco texturas y al menos una curva de las vistas en clases.
- Decore el fondo de la escena con al menos dos texturas.

Requisitos del programa

- Configure estratégicamente 4 cámaras estáticas para visualizar su estructura. Las teclas 1, 2, 3 y 4 deben acceder a dichas cámaras.
- Implemente una cámara móvil, la cual debe moverse alrededor del edificio simulado la vista de un helicóptero. La cámara siempre debe apuntar hacia el edificio, defina usted la forma de control (Teclado o mouse). Indique dichos controles en su reporte.

Distribución de puntaje

- Modelo (3D): 4 puntos.
- Texturado: 2 puntos.
- Interacción con el usuario (Input+Cámaras): 3 puntos.