Universidad de Chile
FCFM - DCC
CC3501

Felipe Morales
26/07/2021

Reporte Tarea 3:
Party Pool

Modelación y Computación Gráfica
CC3501-1

Alumno: Felipe Morales
Fecha: 26 de Julio de 2021

Introducción

Este documento presenta una revisión de los contenidos utilizados y el funcionamiento del programa para esta tercera entrega del curso CC3501-1 Modelación y Computación Gráfica. Se entregarán los debidos argumentos para el uso de las distintas unidades del curso y así entregar una perspectiva de la implementación. Algunos supuestos son:

- No se hizo uso de iluminación global pero se intenta simular esto añadiendo algunas figuras que representan las sombras.
- La escena trata de representar un bar donde existe una luz titilando.
- El palo de billar es representado por un palo de hockey de un archivo OBJ pues no se encuentra un archivo OBJ de un palo de billar.
- Usaremos 10 bolas de billar en esta entrega más la bola blanca.

Desarrollo e implementación

Se decide hacer uso de Modelación Jerárquica dado que se busca construir una escena la cual presenta subconjuntos que se comportan de distinta forma, donde presentamos la complejidad de rotar alguna figura, como el palo de billar, y luego moverlo a una posición cercana a la bola blanca. Es por eso que importa el orden en que aplicamos las distintas transformaciones. También tenemos el conjunto de las bolas que interactúan entre ellas y el conjunto de las sombras que deben seguir la posición de las bolas.

El propósito de tener estos distintos subconjuntos es que se comportan de distintas formas en el estilo del juego por lo que se aprovecha el uso de grafos y las transformaciones por niveles.

A continuación se presenta los distintos diagramas para cada escena con sus respectivas transformaciones donde T() corresponde a una traslación, R() es una rotación y S() es un escalado:

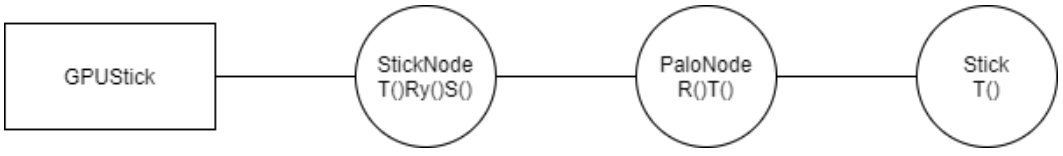

Imagen 1: Modelación jerárquica para palo de billar.

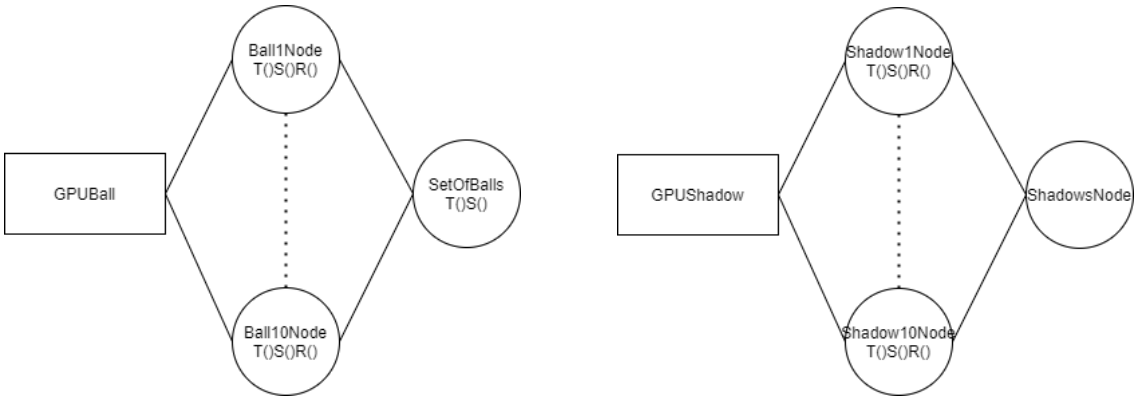

Imagen 2 y 3: MJ para escena de bolas de billar y sus sombras respectivamente.

En cuanto a las transformaciones están en gran parte de los modelos como puede ser visto en las imágenes anteriores por lo que se hizo uso de la mayoría de las transformaciones para la resolución de la tarea.

Para la sección de shaders contamos con 4 de estos pues tenemos 4 funcionalidades distintas. En primer lugar usamos 2 shaders básicos, sin modificaciones, de transformaciones los cuales se encargan de los modelos con y sin texturas. Los restantes son de autoría propia y sirven para los efectos de luz.

La decoración de la escena se trabaja con el uso de texturas donde existen cuadros donde utilizamos técnicas de GL_REPEAT para jugar con los diseños y, al mismo modo, se utilizan técnicas de antialiasing para suavizar ciertas texturas. Otras figuras donde se aplicaron las texturas son las bolas de billar.

La iluminación global no se alcanza a implementar en la tarea por lo que se considera no logrado y queda propuesto. Lo mismo para la visualización científica.

Respecto a las diferencias finitas se considera parcialmente logrado pues se utilizan para la resolución de una EDO con condiciones iniciales utilizando la estrategia de Euler. Donde la siguiente EDO se resuelve de la forma:

$$x'' = roce * gravedad$$
$$velocidad \mathrel{+}= deltaTime * (gravedad * roce)$$
$$posicion \mathrel{+}= velocidad * deltaTime$$

Las colisiones y físicas de la tarea se cumplen parcialmente pues se logra la colisión en el borde la mesa pero falta desarrollar las colisiones entre las bolas de billar. Dicho esto, la bola blanca logra moverse dentro de la mesa considerando el roce y el coeficiente de restitución.

La lógica del juego cumple con un mínimo de requisitos por lo que no se puede considerar que el objetivo base del enunciado fue logrado. En este sentido solo se logra realizar el modelo de la tarea pero falta implementar más funcionalidades.

Todas las entradas del usuario funcionan donde este puede controlar donde quiere golpear la bola blanca, puede visualizar una vista superior de la mesa de pool y puede activar una cámara automática que recorre la mesa. La visualización del estado del programa es completa y correcta para el contexto del problema

Instrucciones de ejecución

Este programa recibe 1 argumento como archivo config.json el cual considera: factor de fricción cinética como “friccion” y un coeficiente de restitución como “restitucion”. Estos parámetros se pueden cambiar en el archivo mencionado. Así, el programa se ejecuta en la terminal de la siguiente forma:

```
pool_party.py config.json
```

El control del juego consiste en que el usuario puede mover el palo de billar a la izquierda con la tecla [←], a la derecha con la tecla [→] y golpeando la bola con la tecla [z]. Por otro lado, el usuario puede activar la vista superior de la escena con la tecla [1] y , además, puede activar la cámara automática con la tecla [2]. Con la tecla barra espaciadora se puede visualizar los polígonos sin rellenar y con la tecla [TAB] se tiene iluminación Phong.

Resultados

El programa consiste en jugar pool golpeando la bola blanca para mover las otras bolas hasta los hoyos sin embargo no se alcanzó a desarrollar las colision entre las bolas por lo que de momento solo se puede mover la bola blanca dentro de la mesa.

Se usan distintas técnicas aprendidas durante el curso para decorar la escena y realizar los distintos movimientos de los objetos en escena.

A continuación se visualizan distintas instancias del juego con las distintas funcionalidades.

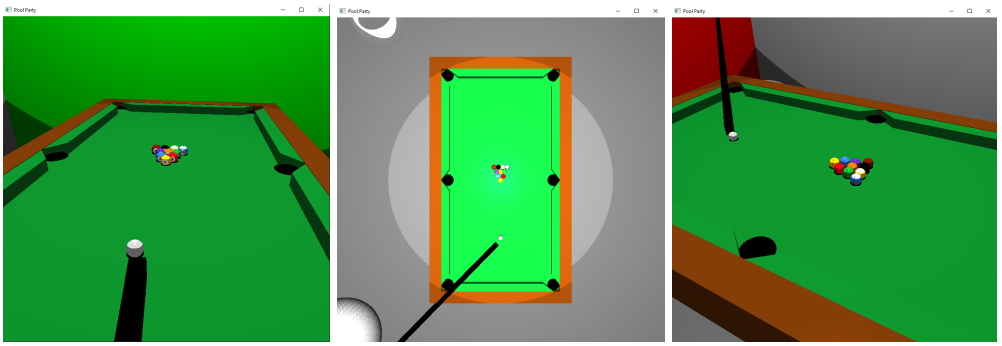

Imagen 4: Instancias del juego

Universidad de Chile
FCFM - DCC
CC3501

Reporte Tarea3b

Felipe Morales
26/07/2021

Autoevaluación

Criterio-Puntaje	0	1	2	3
Shaders				x
Transformaciones				x
Texturas			x	
Modelación Jerárquica				x
Texturas			x	
Iluminación Global	x			x
Visualización Científica	x			
Funcionalidades mecánicas o lógica de juego		x		
Entradas o control de usuario				x
Visualización de estado del programa			x	