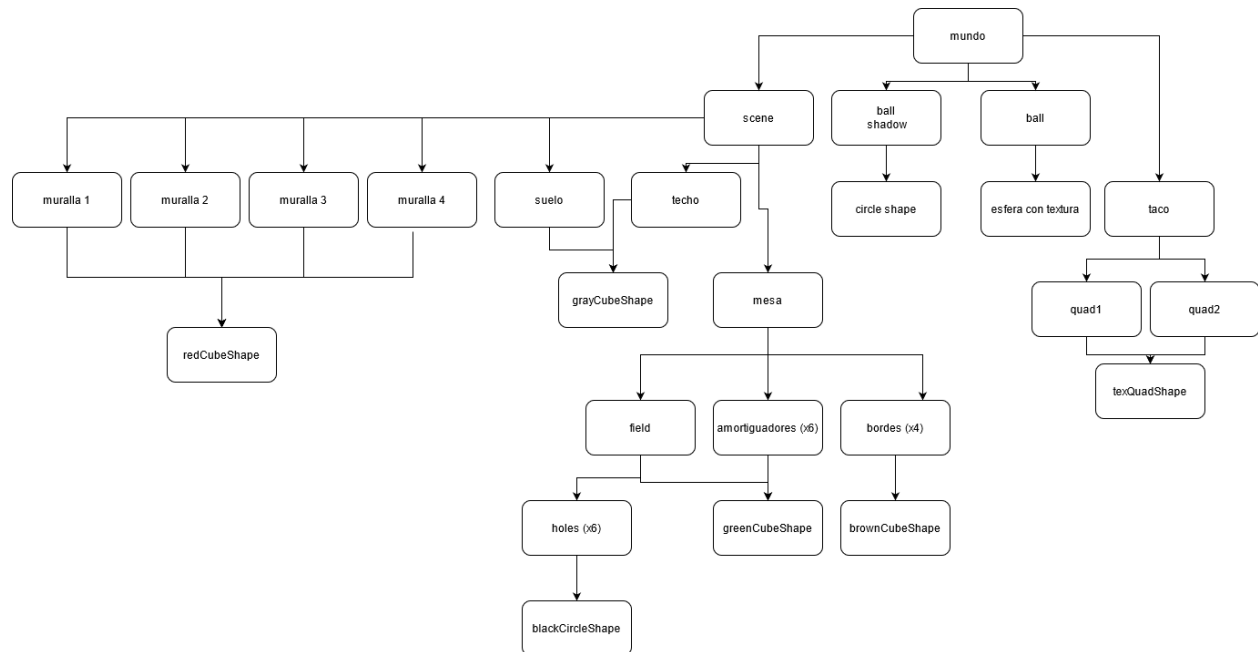


## Solución Propuesta

La solución propuesta cumple con todos los requisitos presentes en el enunciado. Lo logrado fue:

- El modelo de la mesa, con los amortiguadores y bolsillos, está hecho con modelación jerárquica con figuras 3d simples, y 2d para el caso de los bolsillos. La escena tiene de hijos a la mesa, las paredes, techo y el suelo.
- Las 16 bolas son clases Ball con un modelo en 3d con textura para la bola, y un modelo 2d para la sombra. La clase Ball tiene el método action para modelar la física del movimiento de cada bola, implementando la integración de Euler. También, tiene el método update para actualizar las transformaciones de posición, escala, y rotación para el modelo de la bola, mientras que para la sombra se aplican sólo transformaciones de escala y posición. Además, tiene un método delete para eliminar la bola de la escena en el caso de caer en un bolsillo de la mesa.
- El taco se modela en 2d mediante dos cuadrados con transformación de shearing para simular un taco de pool. Debido al movimiento de la cámara polar, la posición del taco no concuerda siempre con la posición que debería tener idealmente.
- Una luz en la parte de arriba de la escena, es la que ilumina toda la aplicación.
- La elección de bola objetivo mediante las teclas Q y W, permite seleccionar la bola que se desea golpear.
- Se hace uso de una cámara polar para moverse en la escena.
- El golpe del taco actúa con la bola seleccionada, modificándole la velocidad en la dirección que apunta la cámara. Además, la cantidad de velocidad modificada dependerá de qué tan cerca está la cámara de la bola, permitiendo hacer un golpe más fuerte mientras más lejos de la bola está la cámara.
- Con la tecla 1 se activa la cámara desde arriba. No permite hacer ni una acción con esa cámara activada.
- En el archivo config.json está el coeficiente de fricción seteado en 0.4, y el de restitución en 0.9.
- Toda la interacción con la aplicación depende de la clase Controller, la cual tiene los métodos para manejar la cámara y actualizar los estados de las bolas.
- Las colisiones entre bolas son en 2d y son del tipo esfera vs esfera, mientras que para la colisión con los amortiguadores y bolsillos de la mesa son esfera vs plano.
- Una vez hecho un golpe a una bola, se debe esperar a que todas las bolas se queden quietas para volver a golpear.
- La rotación de las bolas en el eje Y no siempre funciona correcto, pero el del eje X sí.

El grafo de escena:



Hay que recalcar que, para no extender tanto el grafo de escena, en el caso de los amortiguadores y los bordes de la mesa se dejó con (x6) y (x4) respectivamente, para indicar que hay 6 y 4 nodos prácticamente iguales, que se diferencian por las transformaciones de posición y escala.

## Instrucciones de Ejecución

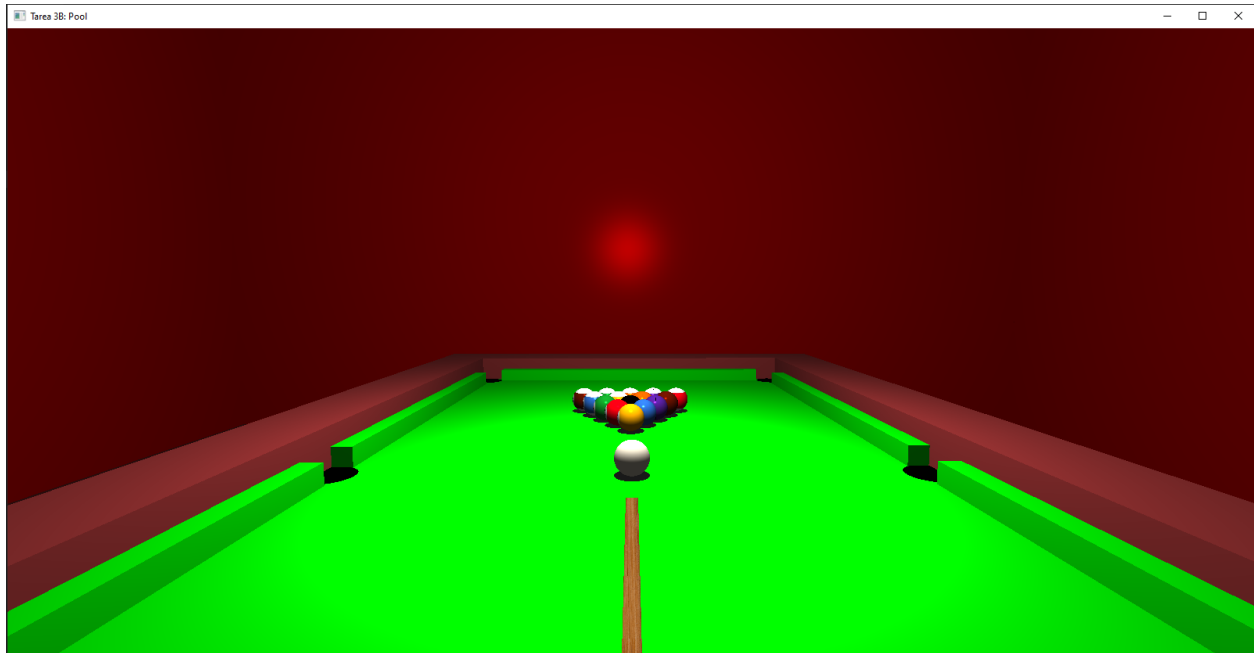
Teniendo las librerías del curso instaladas y estando en la carpeta donde se encuentra `pool_party.py` y sus archivos relacionados, se ejecuta el archivo mediante `python pool_party.py config.json`.

Una vez entra en ejecución el programa aparece la mesa de pool con las 15 bolas de colores ordenadas, la bola blanca al otro lado de la mesa, y el taco listo para golpear la bola blanca. Con las teclas Q y W se puede seleccionar la bola a golpear. Con las flechas izquierda y derecha del teclado se puede rotar la cámara en torno a la bola a golpear. Con la tecla Z se puede golpear la bola seleccionada, y con las flechas arriba y abajo se puede modificar la fuerza con la que se golpea la bola seleccionada. Si una bola toca uno de los hoyos de la mesa, entonces desaparece de la escena. Con la tecla 1 se puede cambiar la cámara para dejarla mirando “desde arriba”; y mientras se mantenga esa cámara, no se puede realizar ni una acción.

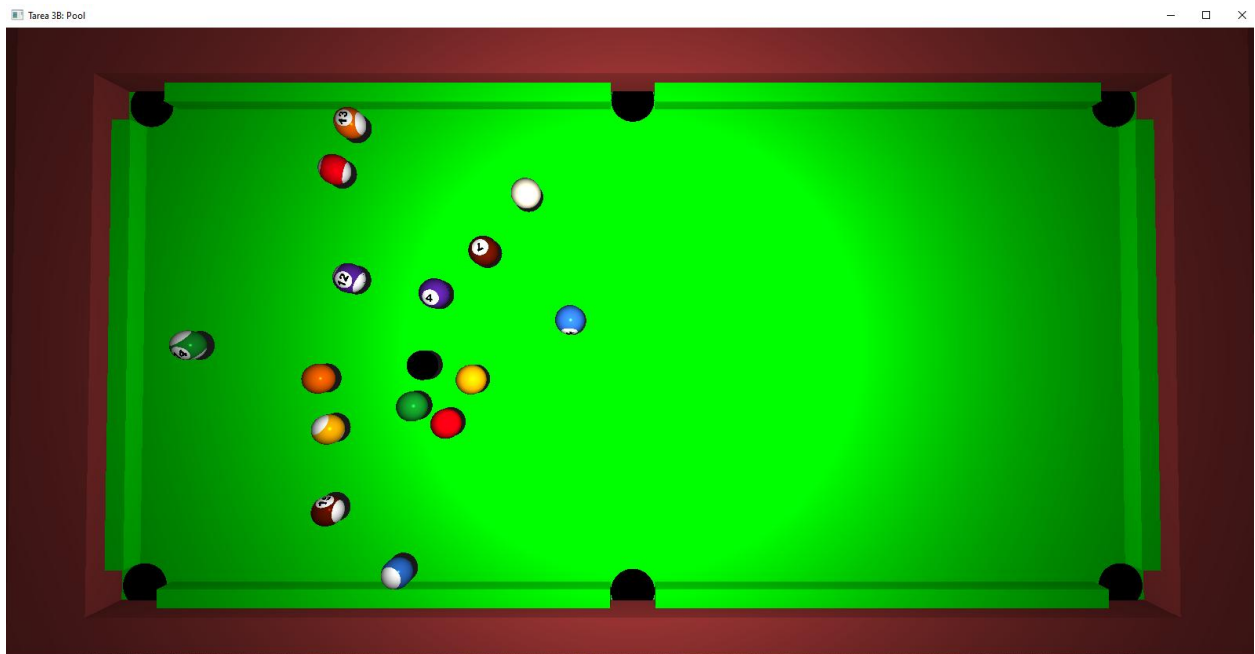
El programa no tiene aplicada ni una lógica de partida de juego, por lo tanto, una vez que desaparecen todas las bolas de la escena, la aplicación se cierra.

## Resultados

Vista inicial:



Vista desde arriba luego de haber iniciado partida:



## Autoevaluación

Criterio-Puntaje	0	1	2	3
Categorías elección (shaders, modelos geométricos, transformaciones)		x		
Texturas		x		
Iluminación global	x			
Visualización científica	x			
Diferencias finitas			x	
Colisiones y física			x	
Lógica de la aplicación				x
Control de usuario			x	
Visualización de estado			x	-----