

TRABAJO #2

Se implemento todos los items asignados en la tarea.

Para almacenar los puntos basto con definir memoria dinamica suficiente para tales puntos.

Cuando la esfera cambiaba de ruta, los puntos tenian un comportamiento raro, para esto se uso una matriz M que almacenaba el historial de las rotaciones.

Se calcularon los vectores de ruta, luego se multiplico por $(0,1,0)$ (productos cruz) para obtener el eje de rotacion, para cada ruta.

Cuando se implemento los primeros items se definio 2 tipos de shader, uno que recibia posiciones y colores(por cada vertice) y otro que solamente recibia posiciones,y el color se le asignaba mediante una uniform variable. Esto se hizo por que tanto el plano como la esfera tenian un solo color, en cambio los axis tenian colores diferentes para cada componente, luego me di cuenta que con tan solo un flag se podria reutilizar el mismo shader.

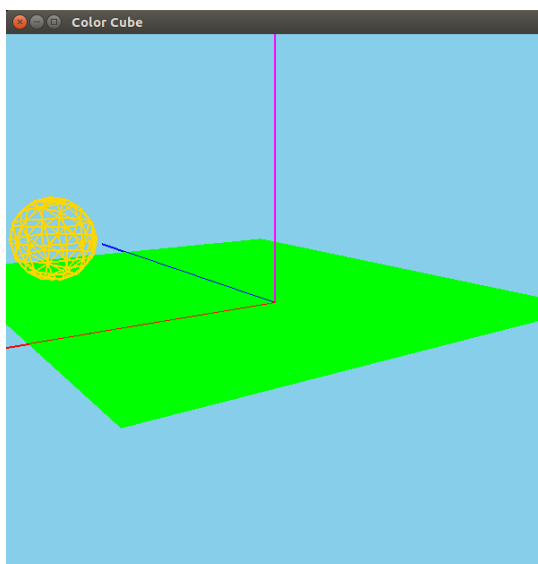
Posteriormente se definio un nuevo shader, el cual era para cuando la iluminacion estuviese habilitada. Aquí teniamos flat o smooth para el shading, ademas podemos elegir el tipo de de fuente de luz (spot or point).

En cuanto a la copia de los atributos ya sea de la esfera o el plano, primero se envio todo los atributos(posiciones, normal para el flat y normal para el smooth), luego definiamos VAOs, los cuales nos permitian recordar configuraiones de los atributos asignados, esto ayudo mucho en cuanto al codigo(era mas legible ya que no habia que estar constantemente asignando los atributos de los VBOs).

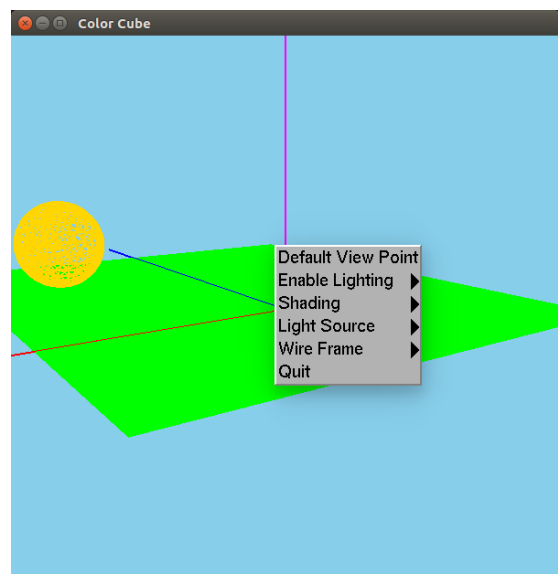
Para calcular el color fianal, todo se trabajo en sistema de la camara (viewer), previamente transformabamos y luego enviamos al shader para su respectivo calculo.

Nota: En las imagenes parece como si los ejes estarian cortados, en realidad no es asi, sino que al reducirlos para que entren. Da esa impresion de entrecortado.

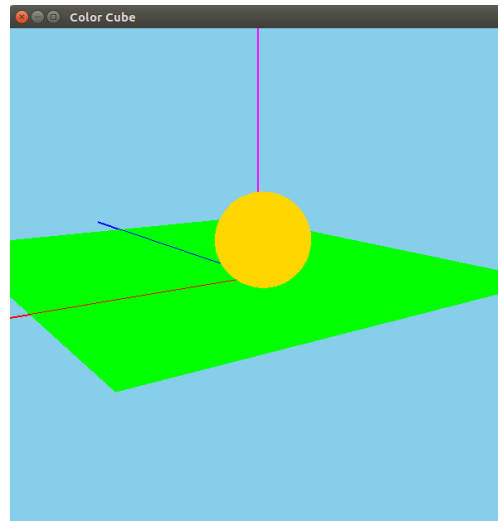
Esfera con wire-frame (ya rotaba)



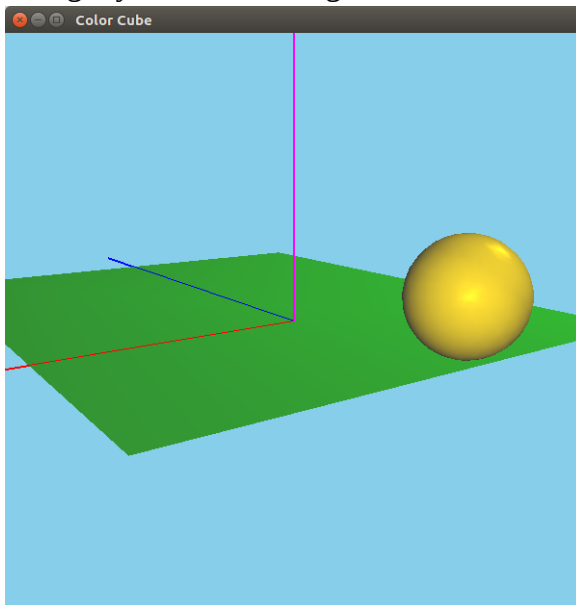
implementando menus



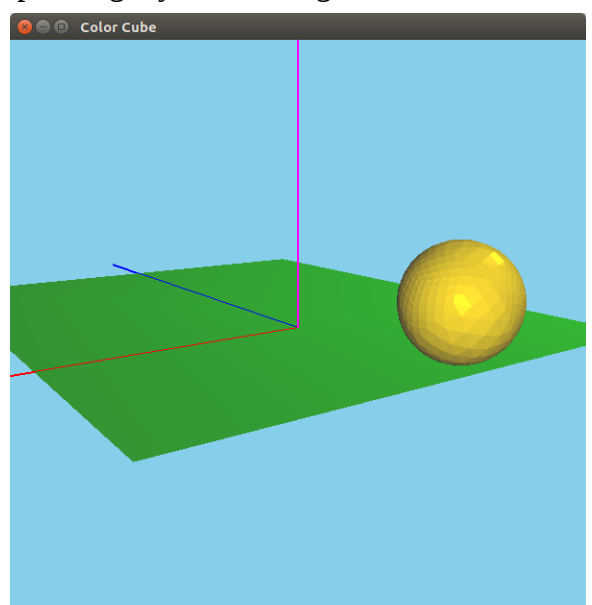
esfera solida(pausa y avanza con clic derecho)



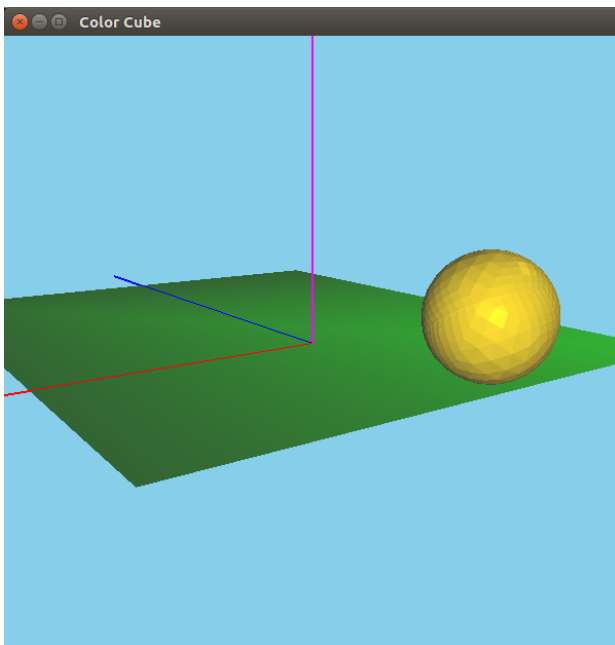
point light y smooth shading



point light y flat shading



spot light and flat shading



spot light and smooth shading

