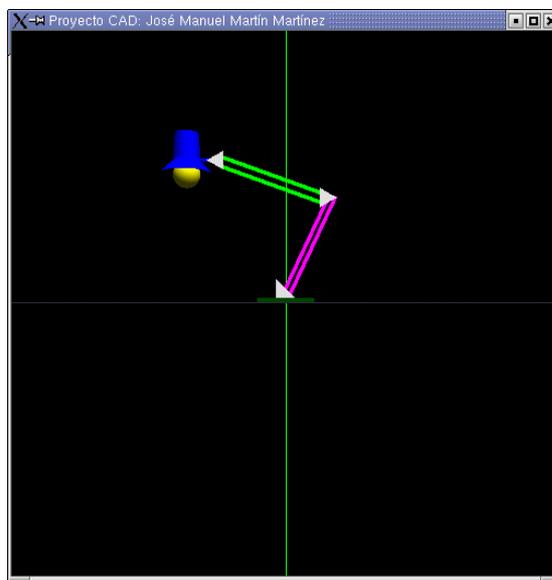


CAD: PROYECTO



José Manuel Martín Martínez

GRUPO: X 10-12

Profesor: Pedro Cano

PROYECTO DE CAD: FLEXO

ESTRUCTURAS DE DATOS Y DETALLES SOBRE LA IMPLEMENTACIÓN

Para el proyecto me he propuesto modelar un flexo tanto en su forma como en su funcionalidad. Para ello, he utilizado las siguientes estructuras de datos y funciones para manejarlas.

Todo lo referente al flexo en cuanto a dimensiones, puntos de movimiento, ángulos que puede girar, luz, posición, ... está en la clase *Flexo*.

Cada instancia de *Flexo* tendrá, entre otras, las siguientes variables:

Variable	Función
id	Numero que identifica unívocamente a una instancia concreta de <i>Flexo</i>
Luz	Luz de OpenGL asignada a ese flexo. Será <code>GL_LIGHT0 + i</code>
Color...	Diversos vectores de tamaño 3 para almacenar el color de cada parte
RotBase	Angulo de rotación con respecto la base del flexo.
rotBrazoAbajo	Angulo de rotación de el brazo articulado inferior
rotBrazoArriba	Idem del brazo superior
RotCabezaV rotCabezaH	La parte de la bombilla del flexo tiene dos grados de libertad. Puede rotar horizontalmente y verticalmente. En estas variables se almacenan los ángulos de rotación.

Para acceder a estas variables se han empleado métodos de instancia que sirven para consultarlas y para modificarlas. Por ejemplo: *SetrotBase(float)* que modifica la variable *rotBase*.

No todos los ángulos son permitidos en el flexo. Así por ejemplo, el brazo inferior tiene un límite entre 30 y -30 grados. Esto se controla en dichos métodos para todas las piezas.

Las dimensiones de cada pieza están definidas en *Flexo.cpp*. Como se puede observar, se declaran al principio del fichero como constantes y luego se utilizan las variables constantes para crear el modelo. Así si quisiéramos cambiar el tamaño del flexo únicamente tendríamos que modificar el valor de las constantes.

El modelo jerárquico que se ha utilizado se puede ver en la función *Dibujar()* de la clase *Flexo*. Lo más importante del modelo jerárquico es mantener la orientación de los elementos superiores del flexo cuando los de abajo rotan. Tenemos que tener en cuenta dos cosas:

- la posición de los elementos cambia en función del ángulo que rote la pieza sobre la que se apoya
- su orientación no ha de cambiar

Para resolver el problema se deshace el ángulo rotado por la pieza inferior antes de colocarlo en la posición en la que va la pieza. Es decir, primero se rota la pieza superior en sentido contrario a la que rota la inferior para luego colocarla a la distancia justa, rotada con respecto a la pieza inferior. Con esto conseguimos un funcionamiento correcto.

Cada parte se ha modelado independientemente y luego se ha unido a la estructura, para facilitar la comprensión e claridad del código.

La luz está integrada en el modelo jerárquico, colocada justo debajo de la esfera que simula la bombilla, por lo que se ve afectada por los movimientos y rotaciones del modelo. Para encender y apagar la luz únicamente hay que llamar a `glEnable()` / `glDisable()`.

Todo este modelo se ha montado sobre la práctica 4 con la que trabajamos en clase. Ha sido necesario cambiar algunas cosas, como por ejemplo la función `pick()` y añadir otras nuevas como es el caso de la función para gestionar el movimiento pasivo del ratón.

Como se puede ver al ejecutar la práctica, cuando pasamos con el puntero del ratón sobre una parte movable del flexo ésta torna a un color rojo. Esto se ha conseguido con la función `ratonpasivo()` que es llamada por `glut` cada vez que se mueve el ratón si que esté pulsado ningún botón. En ella llamamos a `pick()` que nos indica si algún flexo está bajo el ratón y en éste nos indica también que parte de él. Así que se llama a los métodos correspondientes para cambiar el color. Esta característica del programa puede ser un poco costosa respecto a computación por lo que he introducido un flag que comprueba si ha cambiado la selección para no tener que volver a dibujar si no es necesario, aunque la función `pick()` si es necesario que se ejecute cada vez que se mueva el ratón, con los gastos que ello supone.

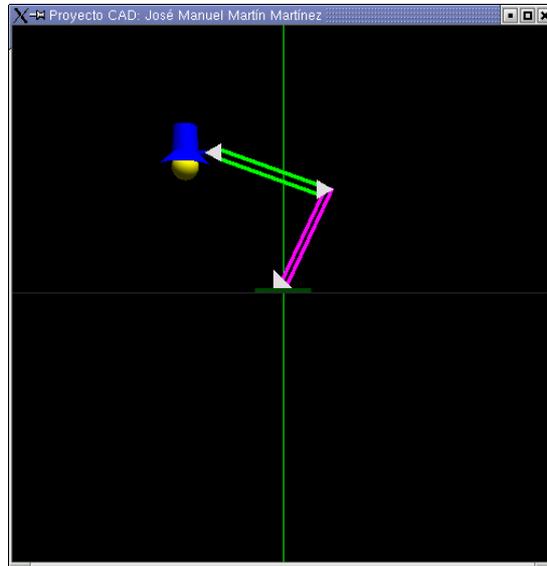
Podemos añadir flexos nuevos. Para ello se ha tenido en cuenta que el flexo no se salga del suelo y no se solape con ningún otro.

Con la función `colision()` podemos comprobar si se chocan las bases de los flexos, no permitiendo mover un flexo sobre otro.

Por último las luces. Se puede encender o apagar una luz de un flexo concreto si se pulsa el botón derecho sobre un flexo concreto o la luz del mundo si no hay flexo seleccionado.

MANUAL DE USUARIO

Instalación: Para compilar y ejecutar la practica, copiar los archivos fuentes y de cabecera a una carpeta al disco duro. Se ha adjuntado un Makefile para facilitar la tarea de compilar.



Al lanzar el programa aparecerá una pantalla como esta:

Inicialmente, aparece un flexo sobre una superficie blanca. Nos podemos mover por este mundo con ayuda del teclado. Así por ejemplo tenemos las siguientes posibilidades:

Teclas + y - : hacen zoom sobre la imagen, es decir, acerca y aleja la cámara.

Teclas movimiento: rotan con respecto al eje X y eje Y

Z y z: rotan con respecto al eje Z

Además de con el teclado, podemos movernos con la ayuda del ratón. Si hacemos click izquierdo sobre una zona vacío (en la que no haya ningún flexo) y mantenemos pulsado el botón podemos arrastrar el mundo.

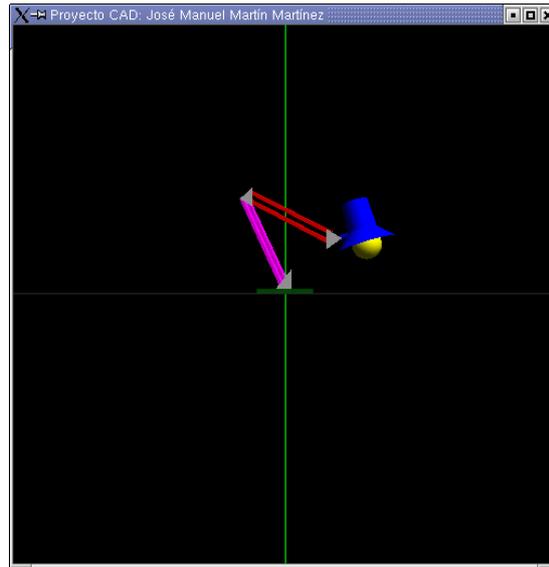
Si movemos el ratón sobre el flexo se puede ver como las partes por las que pasamos con el puntero adquieren un color rojo. Esto quiere indicar que dichas partes son susceptibles de ser movidas o articuladas. Para ello, basta pinchar y arrastrar. Las posibles articulaciones o movimientos son:

- Triangulo sobre la base: pinchando sobre él y moviendo el ratón horizontalmente conseguiremos que el flexo rote sobre su eje.

- Brazo de abajo: pinchando y moviendo horizontalmente conseguimos que avance y retroceda.

- Brazo superior: de igual forma que el anterior, pero en este caso el movimiento del ratón ha de hacerse de arriba a abajo.

- Cabeza del flexo: tiene dos grados de libertad uno vertical y otro horizontal, de acuerdo como se arrastre con el ratón.



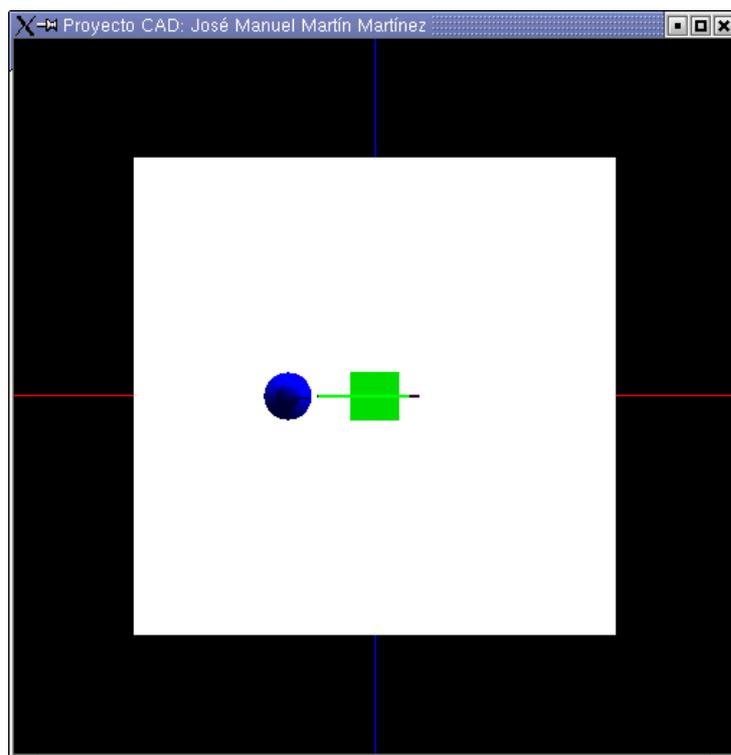
Flexo después de haber sido articulado

Si se pulsa el botón derecho aparecerá el siguiente menú:

Vista Normal
Colocar Flexo
Encender
Apagar
Exit

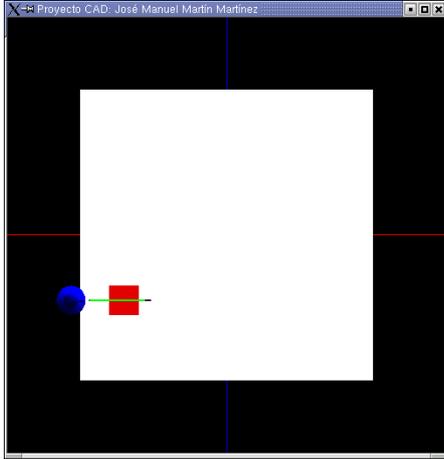
Pasemos a comentar que función tiene cada entrada del menú.

- ↪ Vista Normal: es con la que arranca el programa. En ella podemos articular el flexo y movernos por el mundo gracias al ratón y teclado, como ya se ha explicado antes.
- ↪ Colocar flexo: se cambia a proyección ortogonal como se ve en la figura

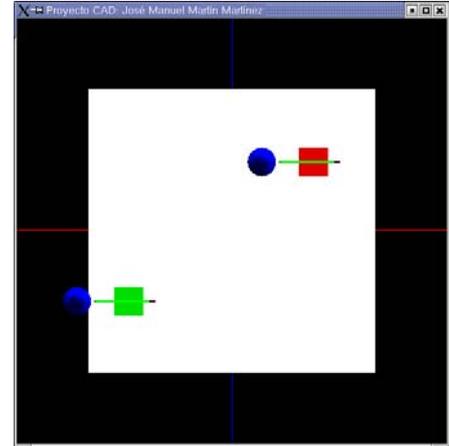


aquí podemos, o bien mover un flexo existente seleccionándolo y arrastrándolo, o bien, añadir nuevos flexos pinchando en una zona vacía del suelo.

Notar que solo se puede arrastrar dentro de la zona blanca así como no se pueden añadir flexos fuera de dicha zona.

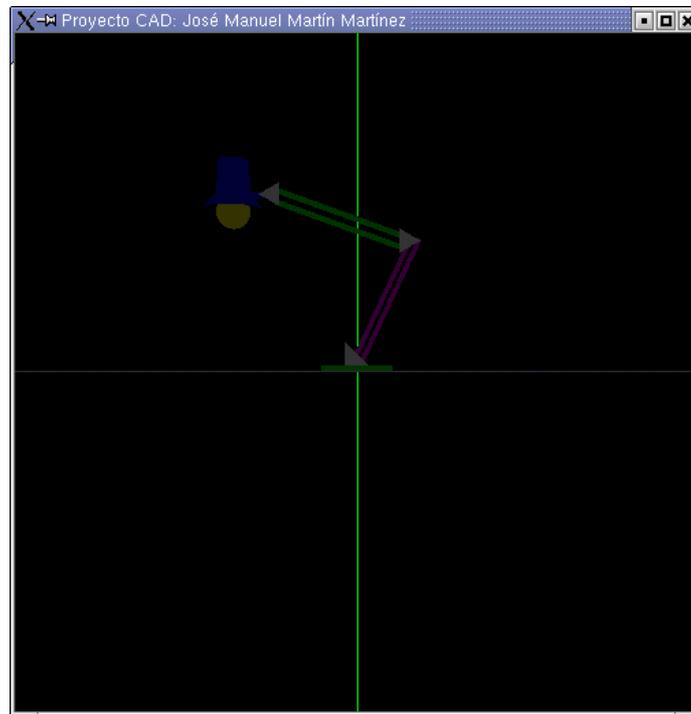


Flexo tras haber sido cambiado de sitio

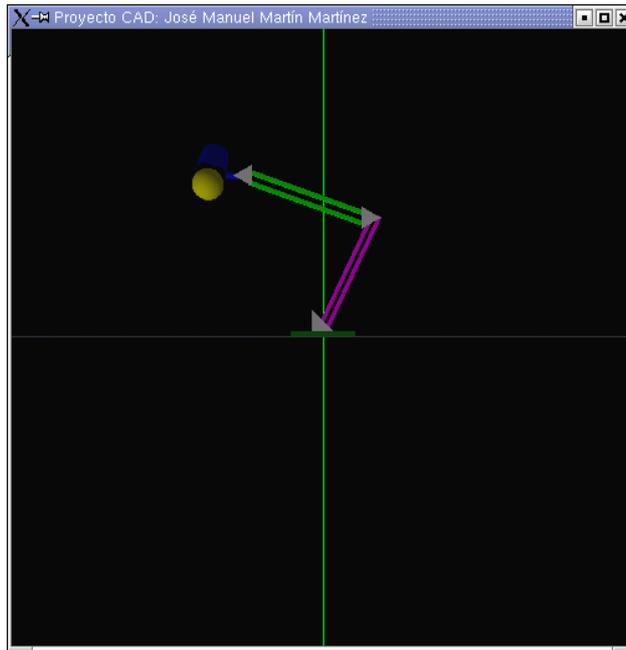


Nuevo flexo añadido

☞ Encender / Apagar: enciende o apaga el flexo sobre el que esté el ratón. Si el puntero no está sobre ningún flexo entonces enciende o apaga la luz ambiental



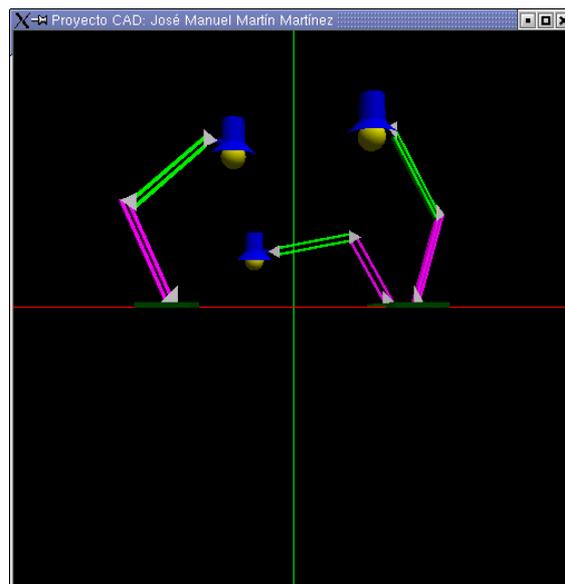
Escena tras apagar la luz ambiental



Escena con luz ambiental apagada y luz del flexo encendida

- Salir: abandona el programa.

El número máximo de flexos que se pueden insertar viene impuesto por el número de luces que soporte la biblioteca gráfica con la que estemos trabajando. Cada flexo es independiente de los demás, pudiéndose mover por separado cada uno de ellos y encender y apagar su luz.



Varios flexos, cada uno independiente del resto