

# Tarea: The Puppet

## Gráficas por Computadora

Fecha de entrega: lunes 12 de octubre, 2020

### 1 Temas

- Modelos jerárquicos y estructuras de datos.
- Pilas de matrices, pilas de acciones (undo/redo).
- Lectura de un modelo jerárquico desde un archivo.
- Z-buffer y eliminación de superficies ocultas.

### 2 Enunciado

Esta tarea se trata de crear una “marioneta” de manera jerárquica a partir de un número de instancias de una primitiva (esfera). Construirás y manejarás una estructura jerárquica para representar la marioneta, que será dibujada (*rendered*) e iluminada (*lighted*) utilizando OpenGL. Deberás construir una interfaz de usuario para manipular los ángulos de las articulaciones de la marioneta y para rotar y trasladar a la marioneta entera. Mantendrás también una pila de transformaciones de tipo *undo* y *redo*.

Para crear el modelo de la marioneta deberás instanciar y transformar apropiadamente la esfera para crear el torso (con su centro como el origen del sistema coordenado de la marioneta). Los hombros y la cadera serán instancias más pequeñas de la esfera y estarán posicionadas relativamente al sistema coordenado del torso. Sus centros formarán el origen de sistemas coordenados secundarios. El centro de la esfera del cuello y la parte superior de los brazos serán el tercer sistema coordenado subordinado al sistema coordenado de los hombros. La esfera de la cabeza se posicionará relativamente al centro del cuello y cada esfera del antebrazo posicionada relativamente al centro de la parte superior del brazo. Las manos serán esferas pequeñas posicionadas relativamente al antebrazo. Las piernas, con muslo, pantorrilla y pie se construirán de manera similar respecto al sistema coordenado de la cadera.

El esqueleto general será como el mostrado en la Figura 1 donde la figura de la izquierda muestra el traslape de las esferas escaladas y en la figura a la derecha se han eliminado las superficies no visibles. El círculo en el centro de la cara es la nariz. Deberás tener al menos un elemento dentro de la cara para que podamos determinar si las rotaciones de la cabeza son correctas, no tiene que ser una nariz, puedes agregar ojos, orejas, cabello, boca, antenas, etc. a tu marioneta.

Puedes ser creativo con tu modelo siempre y cuando tenga al menos el mismo número de grados de libertad (número de articulaciones) que este. Puedes pensar por ejemplo en hacer un gorila, perro, oso, dinosaurio, extraterrestre, etc. pero cambiar el modelo no es requisito para esta tarea. De la misma forma puedes usar otras primitivas de modelo además de la esfera, pero tampoco es un requisito.

El sistema coordenado de cada parte del cuerpo, excepto el torso, se define relativamente al sistema coordenado de su padre. Las dependencias se muestran en la Figura 2. La nariz se dejó fuera de la jerarquía puesto que es parte del modelo de la cabeza.

Puedes utilizar un lenguaje de scripting, por ejemplo LUA para especificar los parámetros y jerarquía de los objetos que compondrán tu marioneta, pero también puedes poner la jerarquía explícitamente

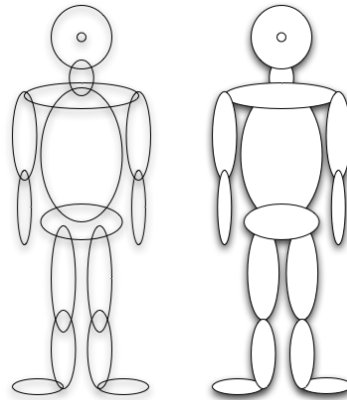


Figure 1: Modelo de la marioneta

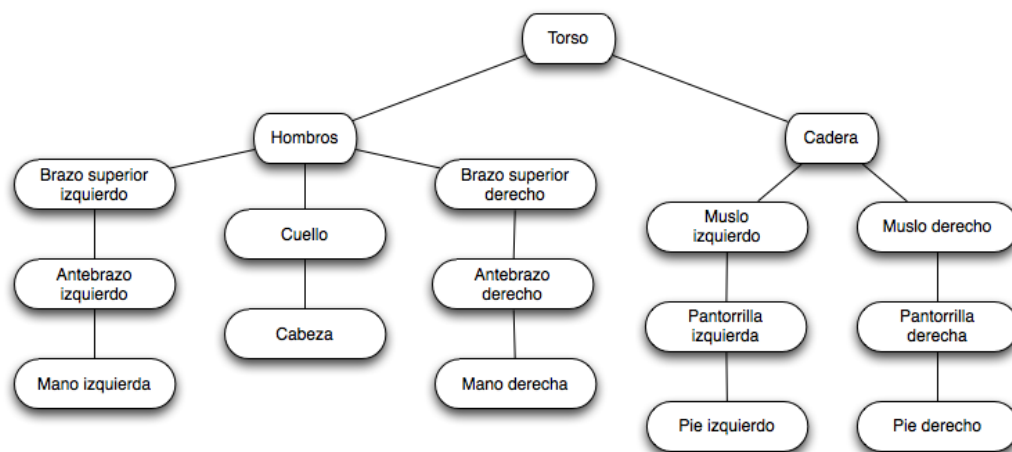


Figure 2: Modelo jerárquico de los objetos que forman la marioneta

en el código. Nota que la estructura mostrada en la Figura 2 es una representación física de las partes del cuerpo que están conectadas y no la estructura de datos. Para la estructura de datos necesitarás *nodos de transformación* intermedios para cada parte del cuerpo para colocar de manera apropiada las esferas en su lugar y para las articulaciones. Te recomiendo no poner hijos bajo los nodos que no tengan escalas uniformes, esto es, pon las primitivas solamente en los nodos hoja.

La definición de la marioneta deberá incluir 15 ángulos libres, 3 para cada brazo y pierna y 3 para la cabeza y cuello. Estos ángulos se usarán para mover a la marioneta. Los tres ángulos del brazo especificarán la rotación del brazo superior en la articulación del hombro, la rotación del antebrazo alrededor de la articulación del codo y la rotación de la mano alrededor de la articulación de la muñeca, las tres en el mismo plano. Los tres ángulos de las piernas especificarán la rotación del muslo alrededor de la articulación a la cadera, la rotación de la pantorrilla alrededor de la articulación de la rodilla y la rotación del pie alrededor de la articulación del tobillo. De los tres ángulos restantes, uno especificará la rotación de la base del cuello hacia adelante (atrás) respecto al torso/hombros, el segundo especificará la rotación de la base de la cabeza hacia adelante (atrás) respecto al cuello y el último especifica la rotación de la cabeza a la izquierda y derecha alrededor del cuello.

Todos los ángulos deben tener un valor mínimo y un máximo que será controlado por la interfaz. Por ejemplo, no debe ser posible rotar la rodilla o el codo en el sentido “equivocado” o rotar la mano o el pie más de 90 grados de su posición neutral.

Antes de hacer tu script o tu implementación te recomiendo hacer tu modelo, con todas las transformaciones de marcos en papel.

### 3 Interfaz

La interfaz de tu programa debe ofrecer un modo de modificar cualquier ángulo de las articulaciones de la marioneta así como poder deshacer o rehacer una rotación o translación a la marioneta. También deberás implementar una tecla o botón para que la marioneta regrese a su configuración inicial y un botón o tecla para salir del programa.

## Objetivos:

Fecha de entrega: lunes 12 de octubre, 23h59

Nombre: \_\_\_\_\_

- \_\_\_ 1: Las proporciones de la marioneta son razonables y las esferas están unidas de manera lógica. Las esferas están "atadas" a sus vecinas por las orillas, estas no rotan alrededor de su centro. (se reconoce la forma de la marioneta)
- \_\_\_ 2: La marioneta se puede rotar globalmente y trasladarse para efectos de vista
- \_\_\_ 3: La configuración de la marioneta puede reiniciarse a partir de un menú o teclado.
- \_\_\_ 4: Se utiliza una estructura jerárquica bien diseñada. Cada esfera se dibuja a partir del objeto de la clase esfera o alguna otra primitiva.
- \_\_\_ 5: Los movimientos de las articulaciones son correctos. La marioneta no debe deshacerse durante una secuencia o combinación de acciones.
- \_\_\_ 6: Los ángulos están restringidos de tal manera que no se permiten configuraciones no naturales.
- \_\_\_ 7: Están implementadas las funciones de profundidad (Z-buffer)
- \_\_\_ 8: Se mantiene correctamente una pila undo/redo.

Recuerda que si no pudiste hacer funcionar alguno de los objetivos anteriores puedes escribir en el archivo README por qué no funciona y tus ideas para solucionarlo. Si algún objetivo no está hecho y no hay explicación sobre este no será tomado en cuenta en las tareas.

Recuerda también que el código y la documentación que entregues es personal y deberás citar tus fuentes.