TESIS
II 2005
M367
v.2

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

DISEÑO E IMPLEMENTACIÓN DE EJERCICIOS PARA ENTRENAMIENTO QUIRÚRGICO VIRTUAL EN CIRUGÍA MÍNIMA INVASIVA(*) (APÉNDICES Y ANEXOS)

TRABAJO ESPECIAL DE GRADO

presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

REALIZADO POR Martínez, Marvi.
Motezuma, José.

PROFESOR GUIA Lic. Rodríguez, Omaira.

FECHA Octubre de 2005.

* Parcialmente financiado por el CDCH, UCV. Proyecto PTP 08.12.5294.2004

APÉNDICE A. CASOS DE USO

A.1 SEGUIR EL CAMINO

Actor: Usuario.

Propósito: Seguir con la punta de la pinza de Laparoscopia el camino que el sistema dibuja.

Precondición: El usuario se encuentra dentro del ejercicio Seguir el Camino, éste le muestra una ruta aleatoria con el primer tramo del camino coloreado de verde.

Curso normal de los eventos:

Acción del actor	Respuesta del sistema
1. El usuario mueve la pinza en la dirección del último tramo del camino coloreado de verde.	
2. El usuario toca con la punta de la pinza el inicio del último tramo del camino coloreado de verde y se mueve en dirección al final del mismo, sin dejar de tocar la línea.	
3. El usuario recorre con la pinza el tramo hasta su final.	4. El siguiente tramo del camino se colorea de verde, para indicarle al usuario que ha alcanzado otro tramo del camino de forma exitosa.

Cursos alternos:

3.a: El usuario mueve la pinza fuera del tramo.

4.a: El sistema registra un error cometido por el usuario.

A.2 PASAR A TRAVÉS DE LOS AROS

Actor: Usuario.

Propósito: El usuario debe utilizar la pinza de Laparoscopia para atravesar una sucesión de aros colocados en la escena sin tocar los bordes.

Precondición: El usuario se encuentra dentro del ejercicio Pasar a través de los Aros.

Curso normal de los eventos:

Acción del actor	Respuesta del sistema
1. El usuario mueve la pinza en la dirección a la sucesión de aros.	
2. El usuario pasa la pinza dentro de la sucesión de aros sin tocar los bordes de éstos.	
3. El usuario toca con la punta de pinza el objeto que se encuentra al final de la sucesión de aros.	4. El objeto colocado al final de la sucesión de aros cambia de color.

Cursos alternos:

2.a: El usuario toca algún(os) borde(s) de los aros.

3.a: Cada aro cuyo borde fue tocado se colorea de rojo, y el sistema registra un error por cada uno de ellos.

A.3 COLOCAR LAS FIGURAS

Actor: Usuario.

Propósito: Colocar una esfera en la sección que el sistema coloree aleatoriamente.

Precondición: El usuario se encuentra dentro del ejercicio Colocar las Figuras y el sistema ha coloreado una sección con un color aleatorio.

Curso normal de los eventos:

Acción del actor	Respuesta del sistema
1. El usuario mueve la pinza de Laparoscopia hacia una esfera del mismo color que la sección coloreada por el sistema.	
2. El usuario captura la esfera del mismo color que la sección coloreada por el sistema.	
3. El usuario se mueve en dirección a la sección coloreada sin soltar la esfera.	
4. El usuario coloca la esfera en la sección coloreada.	5. La sección es coloreada con otro color para indicarle al usuario que colocó la esfera exitosamente.
	6. La esfera colocada exitosamente desaparece de la escena y se decolora la sección.

Cursos alternos:

2.a. El usuario captura y coloca una esfera de un color distinto al de la sección indicada.

3.a. La esfera desaparece de la escena y se registra un error para el usuario.

3.b. El usuario suelta la esfera sin haber llegado a la sección coloreada.

4.b. La esfera desaparece de la escena y se registra un error para el usuario.

4.c. El usuario coloca la figura en una sección no coloreada.

5.c. La esfera desaparece de la escena y se registra un error para el usuario.

4.d. El usuario no coloca la figura en el tiempo estipulado.

5.d. El sistema registra un error y colorea aleatoriamente una nueva sección.

A.4 CORTAR LAS CUERDAS

Actor: Usuario.

Propósito: Cortar con la tijera de Laparoscopia una cuerda en la sección coloreada por el sistema.

Precondición: El usuario se encuentra dentro del ejercicio Cortar las Cuerdas y el sistema ha coloreado una sección de la cuerda.

Curso normal de los eventos:

Acción del actor	Respuesta del sistema
1. El usuario mueve la pinza de Laparoscopia hacia la sección coloreada, en la cuerda, por el sistema.	
2. El usuario corta con la tijera la sección coloreada en la cuerda.	3. La sección coloreada cambia de color indicándole al usuario que esta ha sido cortada exitosamente y luego la cuerda desaparece.

Cursos alternos:

2.a: El usuario corta con la tijera una sección no coloreada en la cuerda.

3.a: Se colorea la sección cortada indicándole al usuario que cortó una sección errónea y registra un error al usuario.

APÉNDICE B. DIAGRAMAS DE ACTIVIDADES

B.1 SEGUIR EL CAMINO

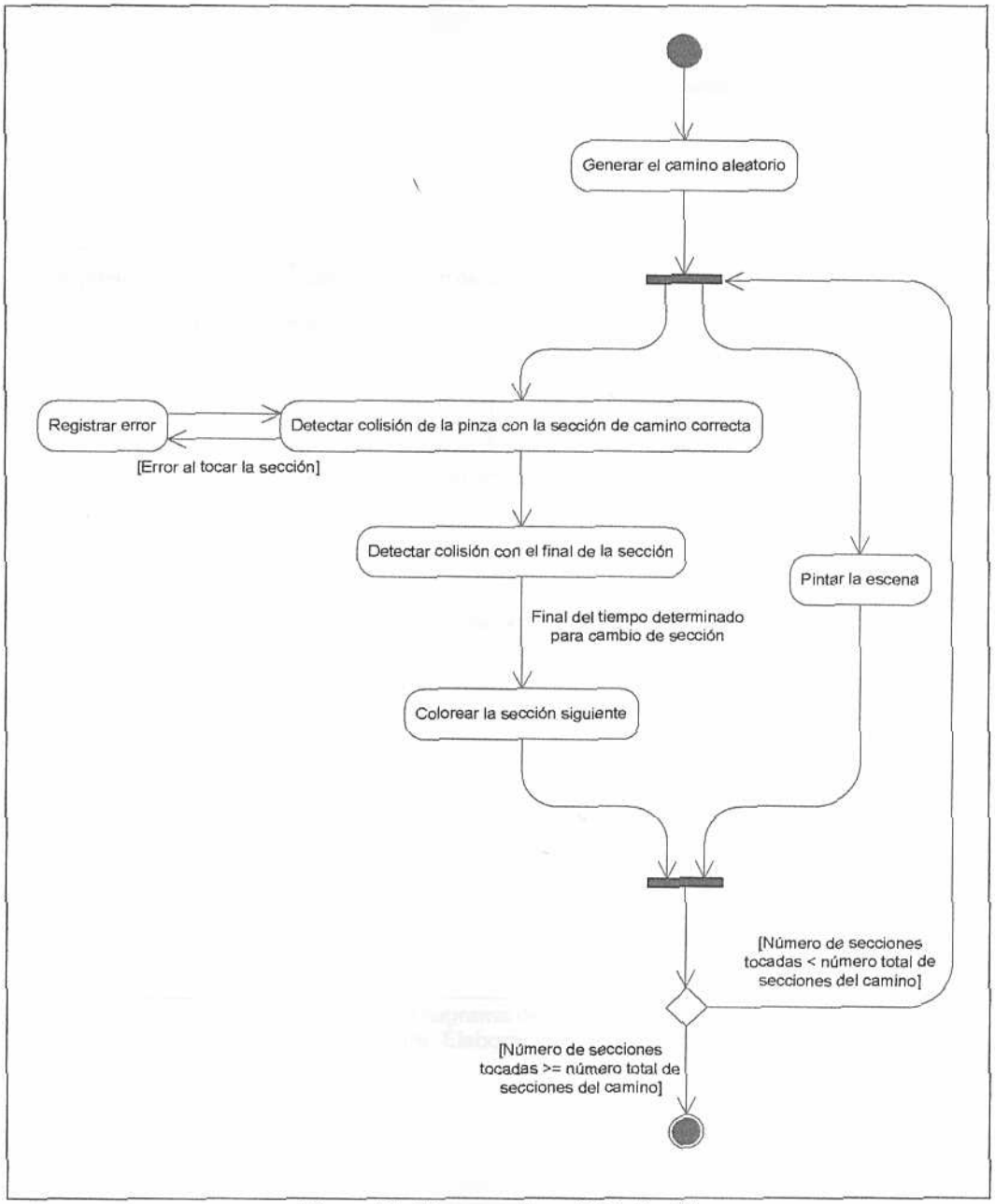

Figura 21. Diagrama de actividades #2
Fuente: Elaboración Propia

B.2 PASAR A TRAVÉS DE LOS AROS

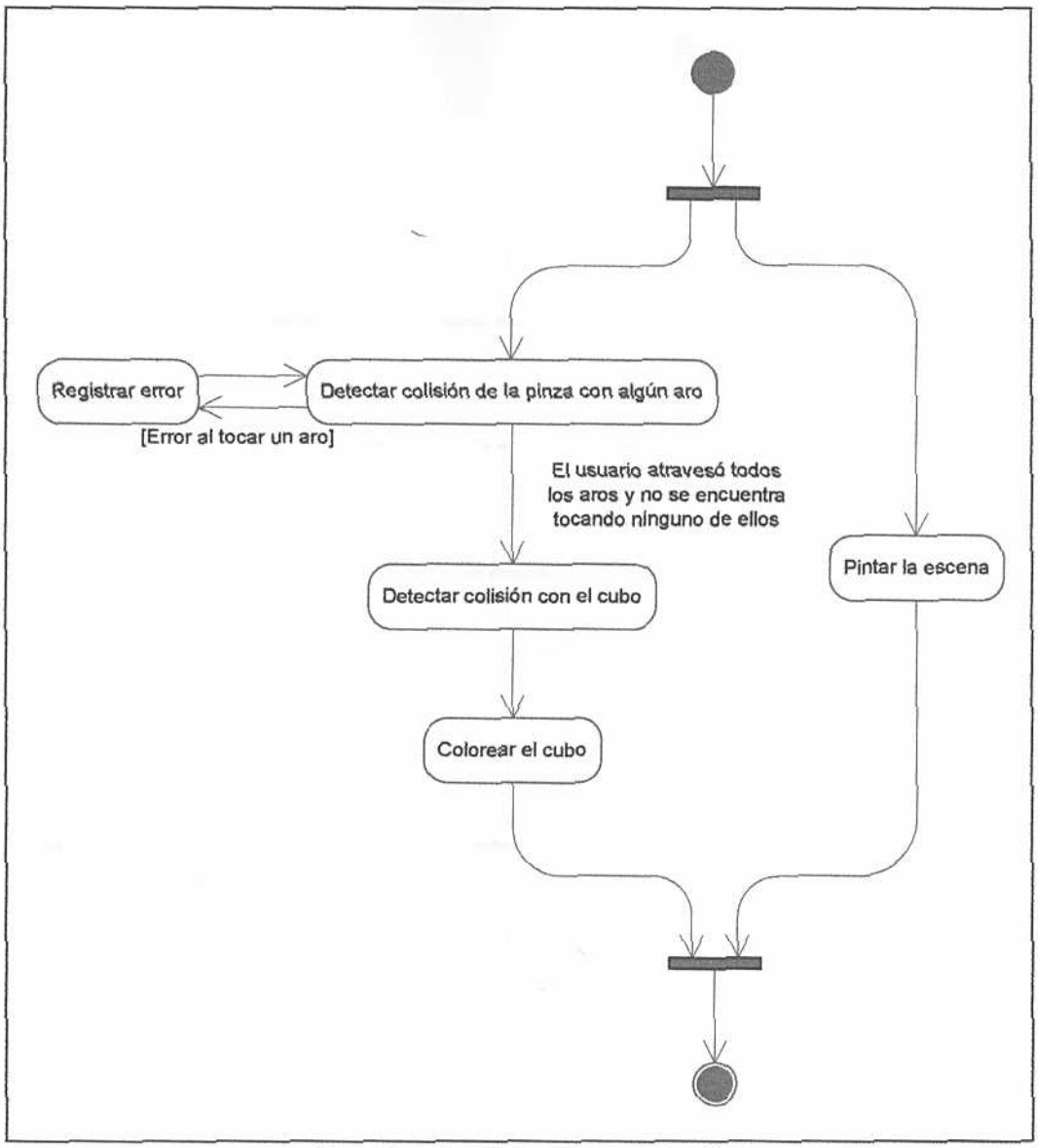

Figura 22. Diagrama de actividades #3
Fuente: Elaboración Propia

B.3 COLOCAR LAS FIGURAS

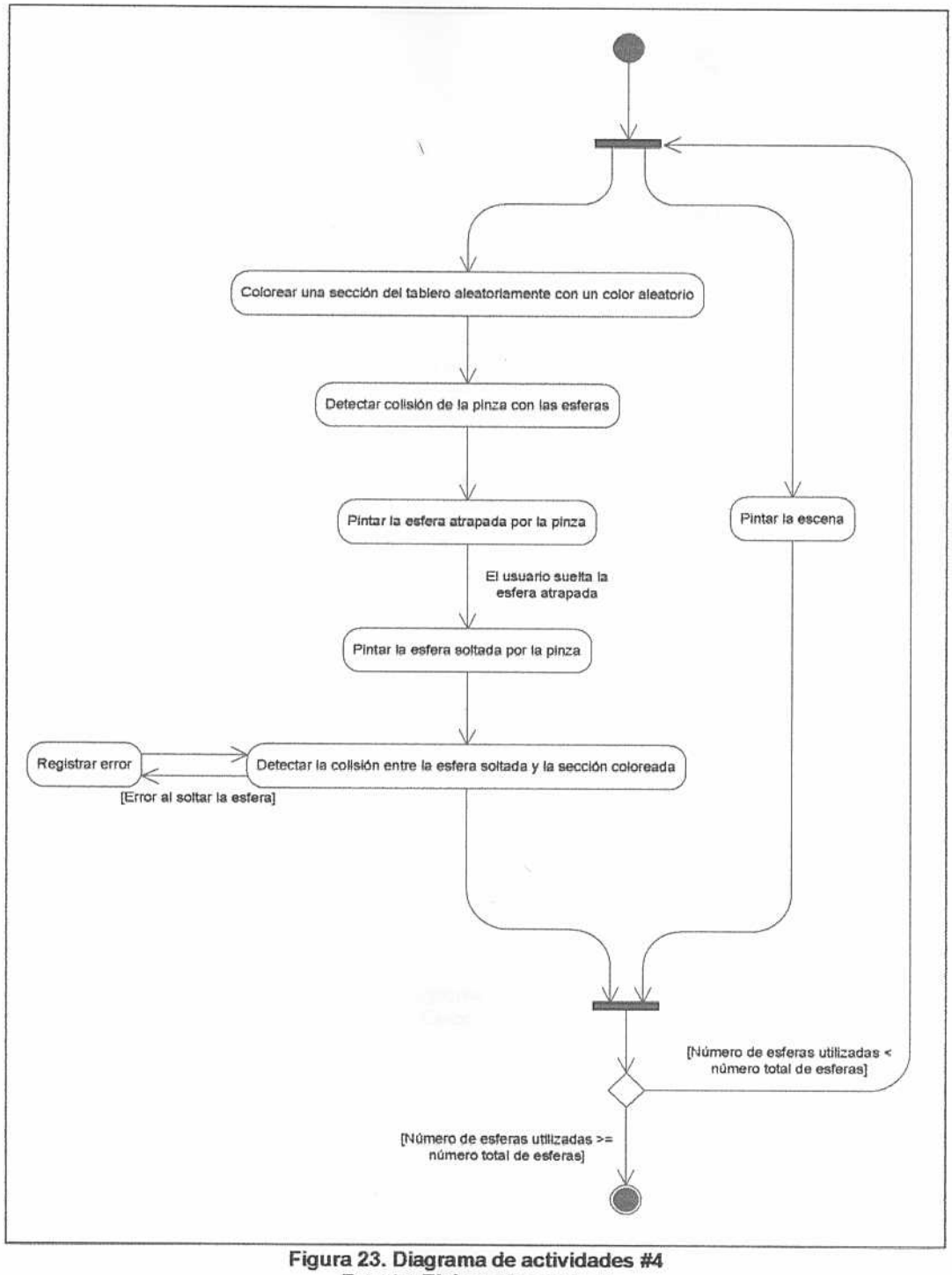

Figura 23. Diagrama de actividades #4
Fuente: Elaboración Propia

B.4 CORTAR LAS CUERDAS

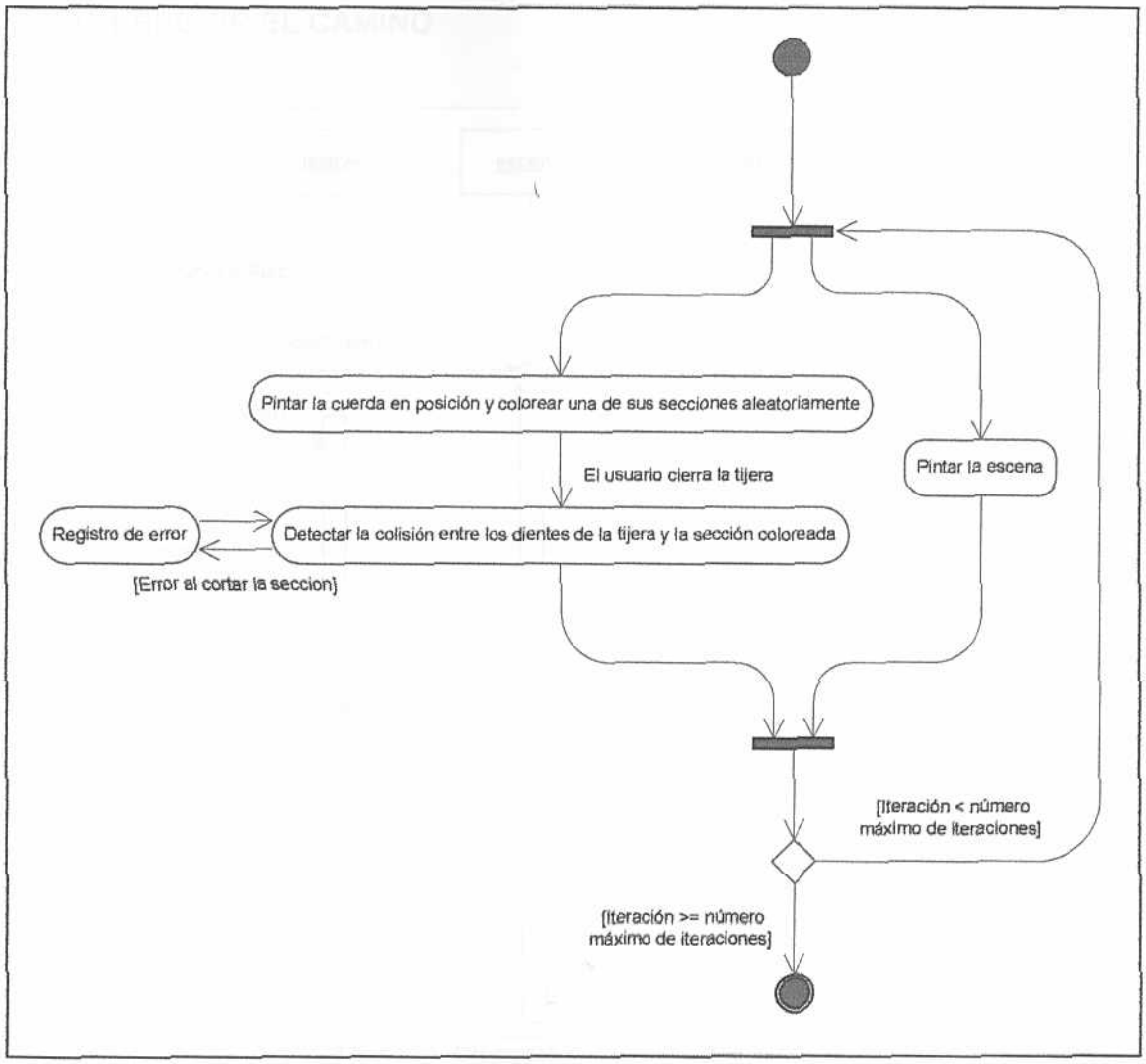

Figura 24. Diagrama de actividades #5
Fuente: Elaboración Propia

APÉNDICE C. DIAGRAMAS DE SECUENCIA

C.1 SEGUIR EL CAMINO

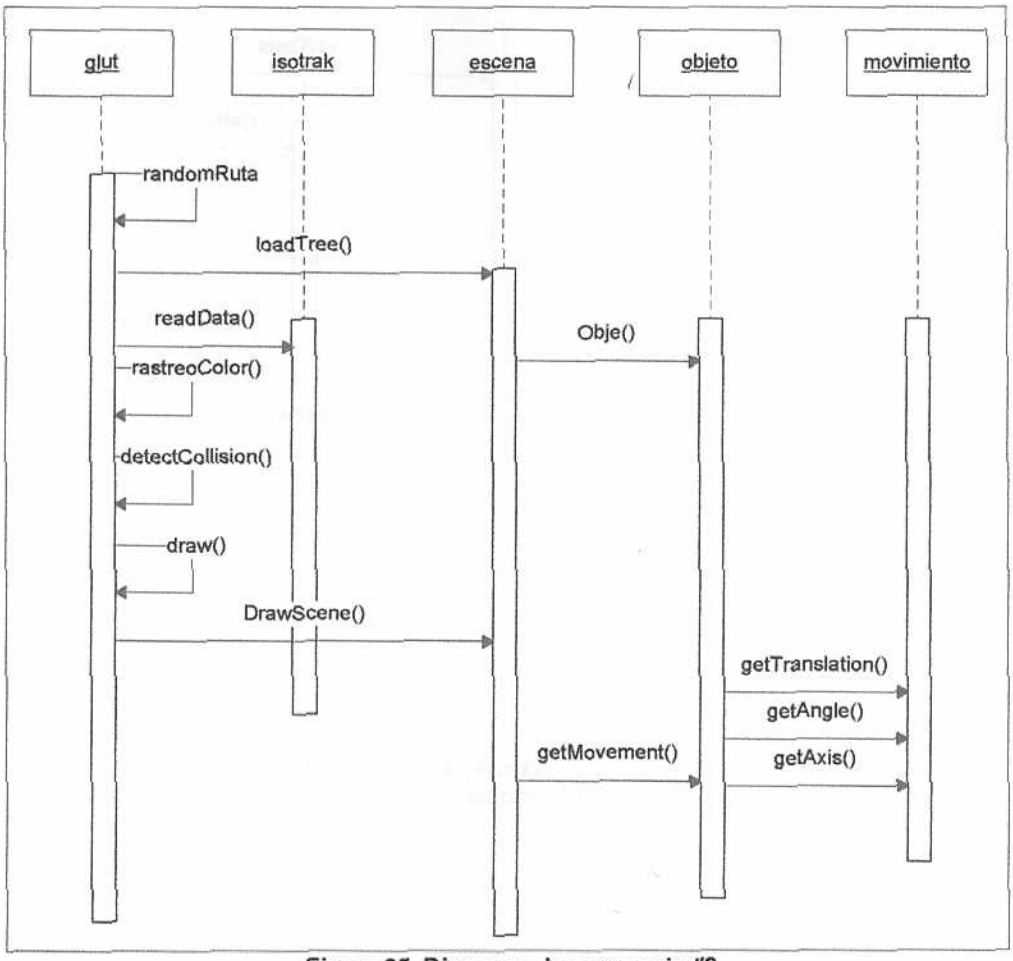

Figura 25. Diagrama de secuencia #2
Fuente: Elaboración Propia

C.2 PASAR A TRAVÉS DE LOS AROS

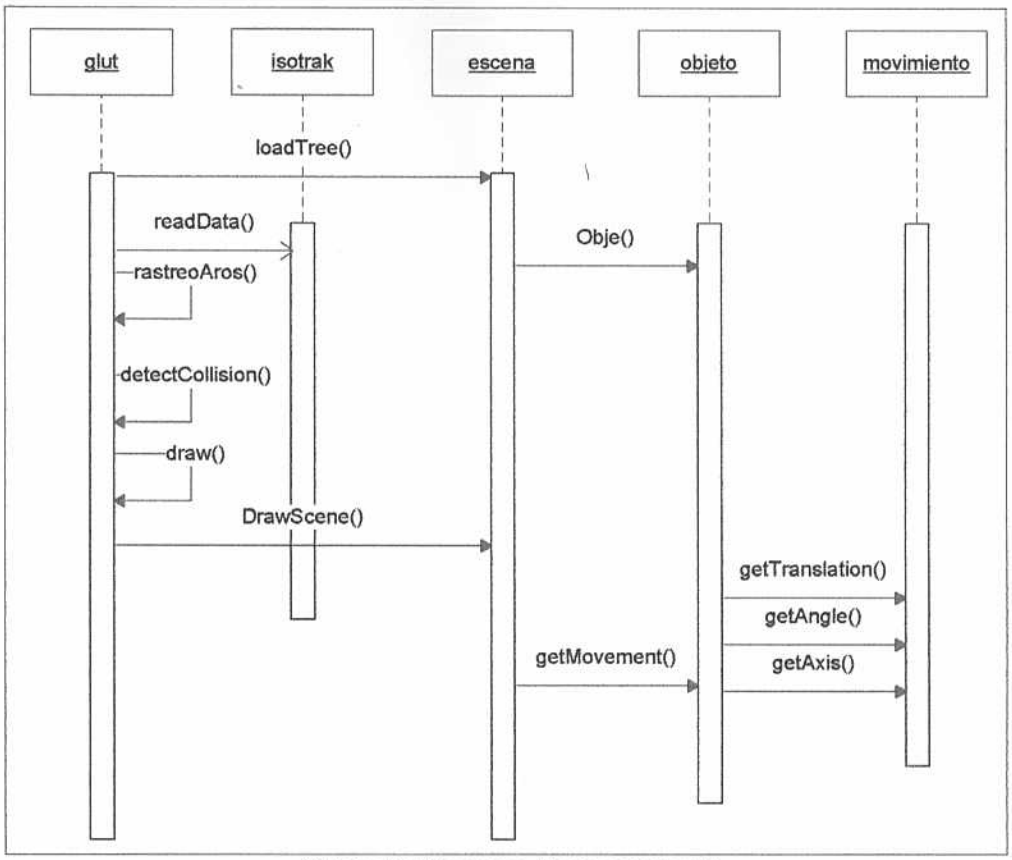

Figura 26. Diagrama de secuencia #3
Fuente: Elaboración Propia

C.3 COLOCAR LAS FIGURAS

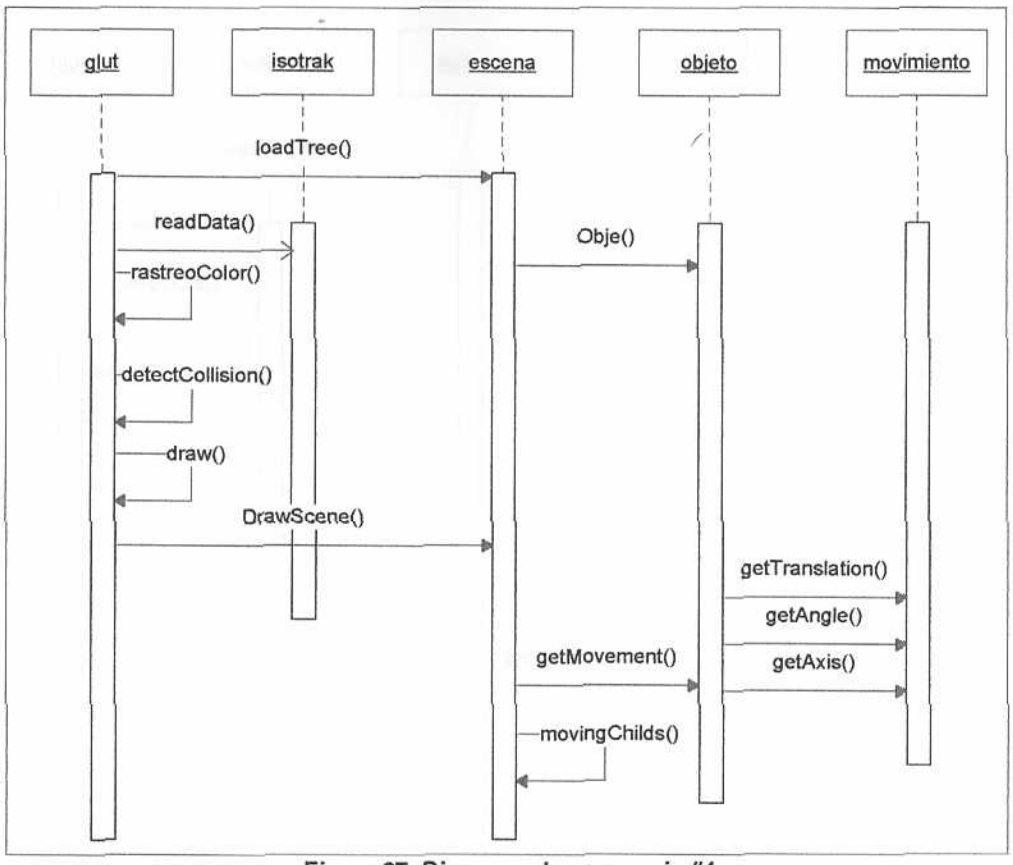

Figura 27. Diagrama de secuencia #4
Fuente: Elaboración Propia

C.4 CORTAR LAS CUERDAS

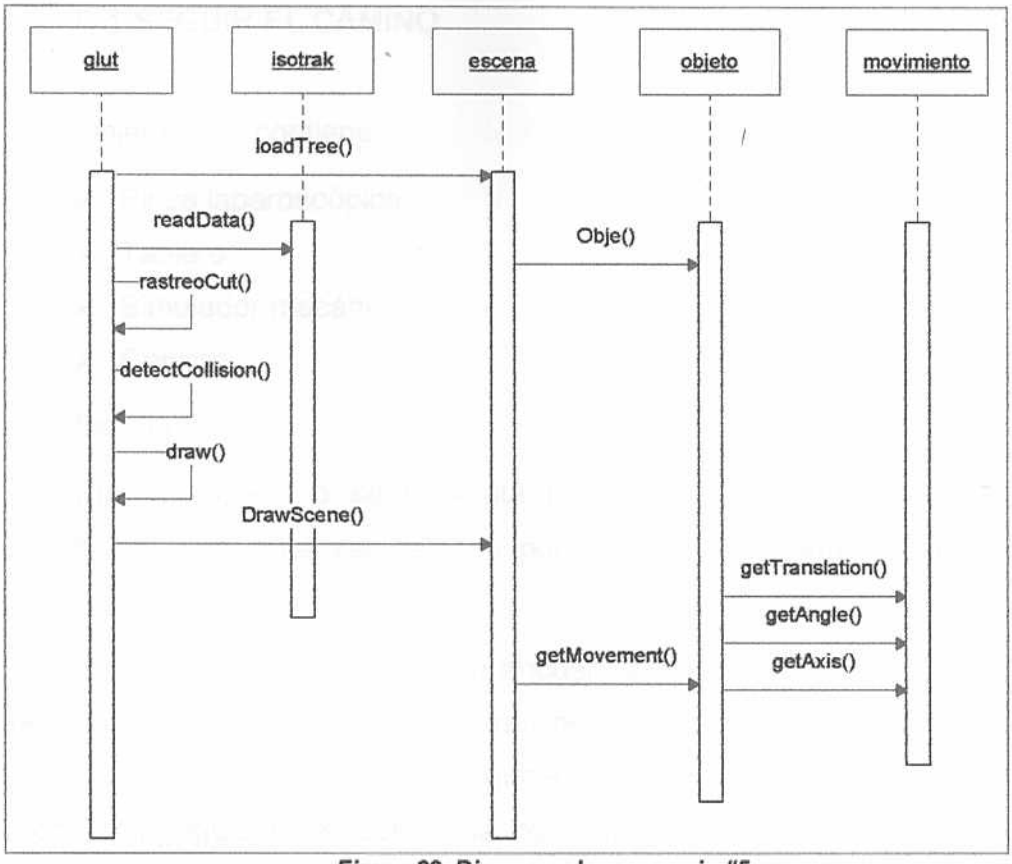

Figura 28. Diagrama de secuencia #5
Fuente: Elaboración Propia

APÉNDICE D. DISEÑO DE LOS EJERCICIOS

D.1 SEGUIR EL CAMINO

Objetos que contiene:

- Pinza laparoscópica.
- Tablero.
- Simulador mecánico.
- Camino.

Descripción:

En este ejercicio se presenta un tablero con un camino aleatorio pintado, el cual debe ser seguido por el usuario, tocándolo con la pinza laparoscópica.

En un principio el camino se encuentra coloreado de rojo y la primera sección a ser tocada por el usuario, de verde. En el momento en que el sistema detecta que el usuario generó una colisión con el final de una sección, la siguiente es coloreada de verde. El ejercicio finaliza cuando el usuario ha colisionado con todas las secciones del camino satisfactoriamente.

El sistema registrará un error por cada vez que los dientes de la pinza pierdan contacto con la línea que le corresponde.

D.2 PASAR A TRAVÉS DE LOS AROS

Objetos que contiene:

- Pinza laparoscópica.
- Simulador mecánico.

- Aros.
- Cubo.

Descripción:

En este ejercicio se presentará un tablero con 5 aros colocados de forma tal que el usuario pueda pasar la pinza entre ellos y un cubo al final de los aros.

Cuando el sistema detecta una colisión entre la pinza laparoscópica y un aro, registra un error y colorea el aro colisionado para que el usuario tenga conocimiento de su error. El ejercicio finaliza cuando el usuario logra tocar el cubo mientras se encuentra atravesando todos los aros de la escena.

D.3 COLOCAR LAS FIGURAS

Objetos que contiene:

- Pinza laparoscópica.
- Tablero.
- Simulador mecánico.
- Esfera.

Descripción:

En este ejercicio se presentará un tablero donde se pintará aleatoriamente, una sección del mismo, y 20 esferas, de 4 colores distintos, colocadas alrededor del tablero.

Las secciones se pintarán, durante un tiempo determinado, de los mismos colores que las esferas; este color será determinado aleatoriamente. Esto se hace así con la intención que el usuario desconozca la próxima sección a colorearse y qué color de esfera debe colocar sobre ésta. Al

momento que se pinte una sección, el usuario debe tomar con la pinza laparoscópica una esfera del mismo color pintado en la sección, si el sistema detecta una colisión entre los dientes de la pinza y la esfera, ésta será transformada de manera tal que se vea tomada por la pinza. Luego el usuario debe colocar la esfera sobre la sección, generando una colisión entre la esfera y la sección coloreada.

Los errores son registrados por el sistema en los siguientes casos:

- El usuario coloca una esfera de un color incorrecto.
- El usuario coloca una esfera colisionando con una sección incorrecta.
- El usuario al soltar la esfera no logra que ésta colisione con la sección.
- El usuario no coloca la esfera y se termina el tiempo especificado.
- El usuario coloca la esfera dentro de la sección y no sobre ella.

En el momento en que el usuario suelta una esfera, ésta es pintada de color verde si el usuario la colocó correctamente, o de rojo, en caso contrario. Luego de un corto tiempo la esfera no es cargada nunca más en la escena

El ejercicio finaliza cuando el usuario ha utilizado todas las esferas de la escena.

D.4 CORTAR LAS CUERDAS

Objetos que contiene:

- Tijera laparoscópica.

- Cuerda.
- Simulador mecánico.

Descripción:

En este ejercicio se presentará una cuerda, con una sección de la misma coloreada, esta cuerda cambiará cada cierto tiempo definido. El usuario debe cortar con la tijera la sección coloreada, este corte será satisfactorio si el sistema detecta una colisión entre los dos dientes de la tijera y la sección coloreada solamente.

En caso que el usuario corte la cuerda en la sección correcta, el sistema coloreará la sección de verde durante un corto tiempo, luego presentará una nueva cuerda a ser cortada; si por el contrario es cortada una sección incorrecta o no es cortada la correcta en un tiempo determinado, el sistema colorea dicha sección y registrará un error.

La posición, inclinación y sección coloreada de la cuerda son obtenidas de manera aleatoria. El ejercicio finaliza cuando el sistema ha mostrado una cantidad determinada de cuerdas.

APÉNDICE E. IMPLEMENTACIÓN DE LOS EJERCICIOS

E.1 SEGUIR EL CAMINO

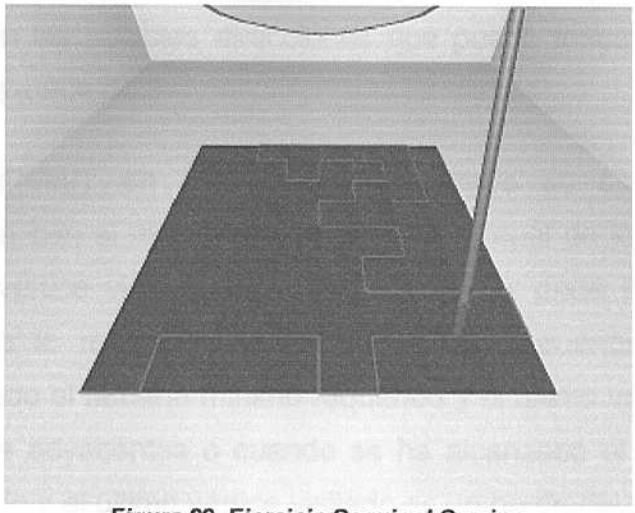

Figura 29. Ejercicio Seguir el Camino
Fuente: Elaboración Propia

En este ejercicio el usuario interactúa con el camino dibujado sobre el tablero que se observa en la Figura 29.

El sistema genera un camino completamente aleatorio donde no se cruzan nunca las líneas entre sí. Para ello se tiene una matriz de 9x9 que indica cuáles vértices de los cubos ya han sido visitados y se encuentran dentro del camino, para así poder determinar, de acuerdo a la ubicación de los cubos, qué dirección debe tomar el camino.

Se tienen varios procedimientos para la generación del camino aleatorio:

- isBorder(): determina si un vértice es borde del tablero, lo cual disminuye la cantidad de posibles direcciones que puede tomar el camino, por ejemplo, si es un vértice que pertenece a uno de los cubos de la primera fila, no existe posibilidad de que el camino siga

hacia abajo, o sube, o sigue a la derecha o a la izquierda, dependiendo de cuál es el origen del mismo.

- getAdyacentes(): determina cuáles son los vértices adyacentes al último vértice que ha sido visitado por el camino, de esta forma, se obtienen las posibles direcciones que puede tomar el camino, que nunca excede de 3.

- randomRuta(): en este método recursivo se escoge de forma aleatoria con el uso de la función rand(), cuál de los adyacentes al último vértice visitado es el próximo a ser parte del camino y va llenando la matriz. El método se detiene cuando el camino ha alcanzado el tamaño mínimo requerido y el último vértice visitado ya no tiene adyacentes o cuando se ha alcanzado el tamaño mínimo requerido y el último vértice visitado es un borde del tablero.

Todo el proceso para la generación del camino se lleva a cabo al momento del inicio del ejercicio, antes de la visualización de la escena, así cuando se llame a drawScene(), el camino esté cargado en memoria y pueda ser visualizado con el resto de los objetos que conforman la escena.

Entonces el sistema se encarga de detectar colisión entre los dientes de la pinza y las líneas que conforman el camino. En un principio el inicio del camino (primera línea a tocar), se encuentra con material verde, y una vez que el usuario entre en contacto con ésta, no debe alejarse del camino, ya que en todo momento se está invocando al método detectCollision() y si éste, no detecta la colisión con la línea que corresponde, se incrementa la variable de error. Para pasar correctamente de una línea a otra, detectCollision() debe determinar que existe colisión simultánea de ambas con la pinza, es decir, el final de una y el inicio de la otra.

E.2 PASAR A TRAVÉS DE LOS AROS

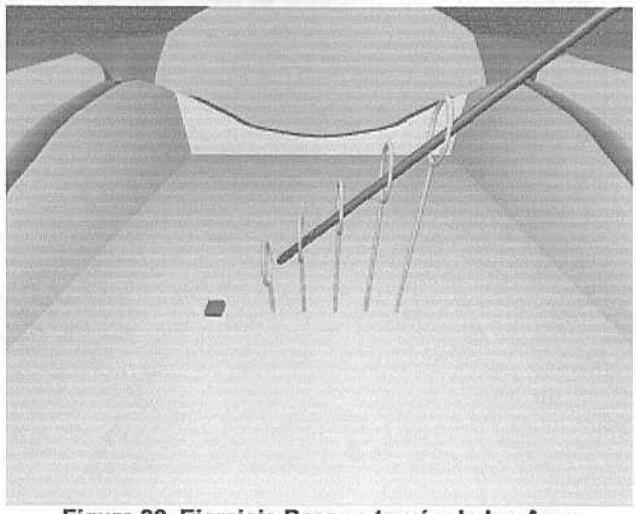

Figura 30. Ejercicio Pasar a través de los Aros
Fuente: Elaboración Propia

En este ejercicio el usuario interactúa con los aros y el cubo (Figura 30), de forma tal, que pueda alcanzar este último con la punta de la pinza, estando dentro de los aros y sin tener contacto con los bordes de éstos.

El sistema se encarga de detectar colisión entre los aros y la pinza, y si es así, el material del aro o los aros que son colisionados, cambia a color rojo, utilizando el método setMaterial(), durante todo el tiempo que el usuario mantenga la colisión.

Para determinar que la pinza se encuentra dentro de las circunferencias de los aros, se cargaron en la escena cinco objetos transparentes con forma de círculo, cuyo diámetro es un poco menor a la de los aros. En todo momento se llama al método detectCollision() y si éste determina que los dientes de la pinza tocan el cubo, se invoca de nuevo al mismo método, pero para determinar si la pinza esta colisionando simultáneamente con los 5 objetos transparentes. De otro modo el ejercicio no finaliza y cada vez que un aro sea coloreado de rojo la variable de error se ve incrementada.

E.3 COLOCAR LAS FIGURAS

En la Figura 31, se observa la escena que el usuario puede visualizar al realizar este ejercicio.

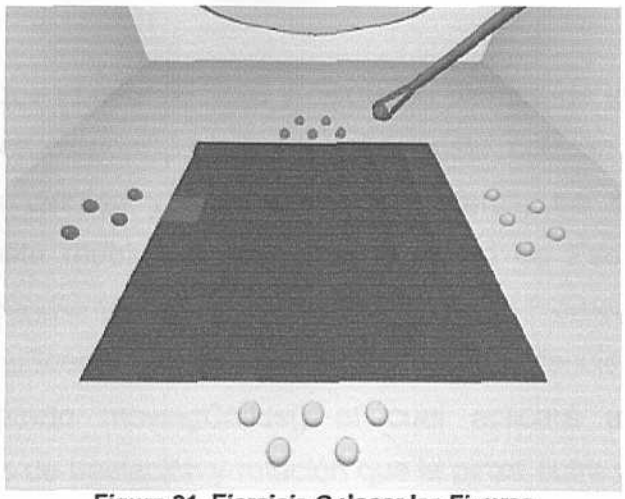

Figura 31. Ejercicio Colocar las Figuras
Fuente: Elaboración Propia

En este ejercicio el usuario interactúa con las esferas a través la pinza. El sistema aquí debe detectar diversas colisiones y modificar el Grafo de Escena.

El sistema carga en la escena esferas con colores especificos, y se encarga de colorear en una secuencia y en un color aleatorio (los mismos con los que fueron cargadas las esferas) los cubos del tablero, esta operación la realiza con la ayuda de la función rand().

```
painted= ((float)rand()/RAND_MAX)*63;

  a=((float)rand()/RAND_MAX);
  if ((a<0.25)&&(contOrange<5))
  {
        escena.painted_color=10;
  }
  if ((a>=0.25)&&(a<0.5)&&(contYellow<5))
  {
        escena. painted_color =11;
  }
  if ((a>=0.5)&&(a<0.75)&&(contGreen<5))
```

```
{
        escena. painted_color =9;
}
if ((a>=0.75)&&(contBlue<5))
{
        escena. painted_color =12;
}
```

Cada vez que el usuario abre y cierra los dientes de la pinza, se llama al método detectCollision(), y en caso de existir colisión entre las tapas internas de los dientes y alguna de las esferas, se llama al método reorderTree(), éste reubica la esfera en el Grafo de Escena colocándola como nodo descendiente de la pinza. A partir de este momento cada vez que se llame a drawScene(), éste verificará la descendencia de la pinza, e invocará el método movingChilds() el cual aplicará a la esfera las transformaciones de traslación y rotación que la pinza sufra por motivos de la interacción del usuario.

```
If (root[0]->descendant != NULL)
{
                movingChilds();
}
```

En el momento que el sistema detecta que ha sido soltada una esfera, invoca a detectCollision(), el cual verifica que la esfera fue bien ubicada sobre un cubo y si éste último se encuentra coloreado con el mismo tono de color que luce la esfera. El material de la esfera es cambiado a verde por un corto instante de tiempo, para indicarle al usuario que colocó la esfera de forma exitosa. Luego de esto, reorderTree() elimina la esfera del árbol, logrando así que ésta no sea visualizada nunca más en la escena.

La variable de error es incrementada cada vez que el usuario incurra en alguno de los casos de error que fueron señalados en la etapa de diseño de este ejercicio.

E.4 CORTAR LAS CUERDAS

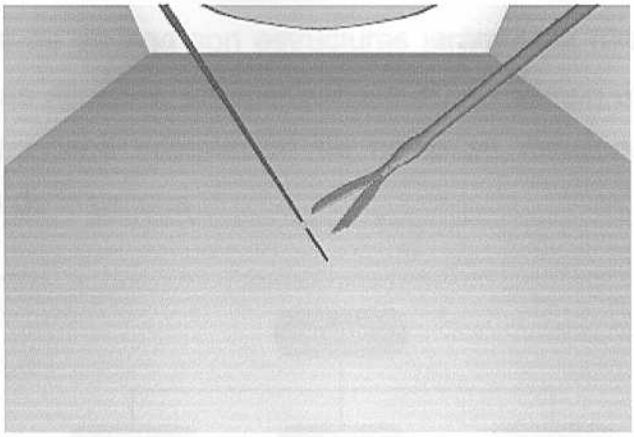

Figura 32. Ejercicio Cortar las Cuerdas
Fuente: Elaboración Propia

En este ejercicio el usuario interactúa con una cuerda a la vez, la cual está conformada por un conjunto de objetos del tipo línea.

Existen 36 posibles cuerdas a cargarse, ya que dependen de la ubicación y la inclinación que lleven. Para ello se escoge una de estas cuerdas por medio de un número aleatorio determinado por la función rand(). Igualmente se usa esta función para determinar que línea de la cuerda es la que va a colorearse, utilizando setMaterial(), para indicarle al usuario cuál sector de la cuerda debe cortar.

El método detectCollision() determina si cada vez que el usuario abre y cierra los dientes, éstos se encuentran simultáneamente colisionando con alguna línea de la cuerda y si ésta es la que se encuentra coloreada en la escena. En el caso que se detecte colisión con la línea coloreada y con otra adyacente a ésta, sea arriba o abajo, se considera un error para el usuario.

La variable de error se incrementa cada vez que el usuario cierra los dientes y no lo hace exclusivamente sobre la línea coloreada y en el caso de que se cargue una línea y la anterior no haya sido cortada en el sector que fue indicado.

APÉNDICE F. GRAFOS DE ESCENA

Los grafos de escena son estructuras jerárquicas (árboles) utilizadas para gestionar las escenas, en aplicaciones 3D en tiempo real. La jerarquía de la escena contiene información de cómo se relacionan entre sí los distintos objetos de la escena.

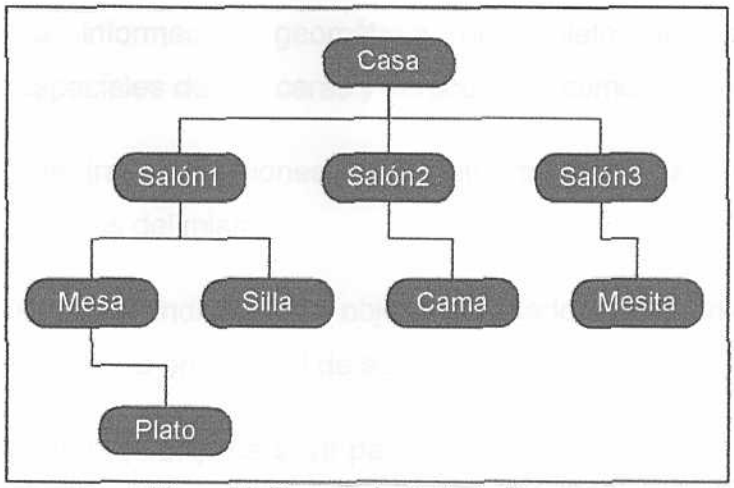

Figura 33. Ejemplo de Grafo de Escena
Fuente: Elaboración Propia

El árbol que representa una escena está conformado por una serie de nodos que contiene la información de cada uno de los objetos presentes en la escena, y se recorre en profundidad desde el nodo raíz hasta llegar a los nodos hojas, de arriba abajo y de izquierda a derecha, para realizar las operaciones necesarias sobre estos.

F.1 ESTRUCTURA DEL GRAFO DE ESCENA

La estructura del grafo de escena utilizado para la realización de este sistema es la siguiente:

- Nodo raíz: Es la raíz del árbol que contiene la jerarquía de la escena, bajo él se encuentran todos los nodos pertenecientes al grafo.
- Nodos: Almacenan la información referida al objeto poligonal que se plasmará en pantalla, éstos contienen:
  - Las propiedades del objeto: nombre y material.
  - La información geométrica del objeto: las coordenadas espaciales de sus caras y vértices, así como el color del mismo.
  - Las transformaciones del objeto: translaciones, rotaciones y escalas del mismo.
  - Los descendientes del objeto: apuntadores a los descendientes del objeto en el árbol de escena.

Esta estructura jerárquica sirve para manipular los objetos de la escena que dependen de otros objetos, de esta manera, si un objeto que posee descendientes es transformado en la escena, los objetos descendientes de él sufren las mismas transformaciones.

APÉNDICE G. ESTRUCTURAS DE ARCHIVOS UTILIZADOS

G.1 ARCHIVO "OBJECTS.TXT"

La estructura de este archivo es creada por personal del LCG:

object [nombre del objeto] El nombre del objeto debe coincidir con un objeto existente.
geometry "[ruta]" Debe contener la ruta donde se encuentra el archivo ".off" del modelo del objeto.
aabbtree "[ruta]" Debe contener la ruta donde se encuentra el archivo ".aabb" del modelo del objeto.

Estos 3 datos van tantas veces como objetos se utilicen en el sistema.

Ejemplo:

```
object cubo
geometry "..\\..\\data\\cubo.off"
aabbtree "..\\..\\data\\cubo.aabb"

object aros
geometry "..\\..\\data\\aros.off"
aabbtree "..\\..\\data\\aros.aabb"
```

G.2 ARCHIVOS CON EXTENSIÓN "OFF" [38]

OFF
[Número de Vértices] [Número de Caras] [Número de Bordes]
[x1] [y1] [z1] (vértices)

.
.
[v1] [v2] [v3] .....[vn] (caras)

.
.

Contiene tantas líneas de vértices (x,y,z) como lo indique el Número de vértices y tantas líneas de caras cómo lo indique el Número de Caras.

El Número de Vértices empieza desde 0 y el Número de Bordes es siempre cero.

Ejemplo:

```
OFF
8 6 0
-0.500000 -0.500000 0.500000
0.500000 -0.500000 0.500000
-0.500000 0.500000 0.500000
0.500000 0.500000 0.500000
-0.500000 0.500000 -0.500000
0.500000 0.500000 -0.500000
-0.500000 -0.500000 -0.500000
0.500000 -0.500000 -0.500000
4 0 1 3 2
4 2 3 5 4
4 4 5 7 6
4 6 7 1 0
4 1 7 5 3
4 6 0 2 4
```

G.3 ARCHIVOS CON EXTENSIÓN "SCL"

Estos archivos de elaboración propia poseen la siguiente estructura:

[Número de la escena] Del 1 al 5.

[Parámetros de entrada de la función LookAt()] Son 9 números flotantes separados por un espacio entre sí.

[Cantidad de objetos de la escena] Es un número entero que puede ir desde 1 a 200. En el caso de este sistema va de 5 a 200 ya que los primeros 5 son objetos obligatorios a cargar (simulador, barra, 2 dientes internos y 2 externos del instrumento laparoscópico.

[Nombre de objeto] Debe coincidir con alguno de los nombres de objetos existentes dentro del archivo objects.txt.

[Número de traslaciones iniciales del objeto] Es un número entero mayor que 0.

[Parámetros para traslación] Son 3 números flotantes separados por espacio. Deben existir tantas traslaciones como el número que se indica y deben ir separadas por un fin de línea.

[Número de rotaciones iniciales del objeto] Es un número entero mayor que 0.

[Parámetros para rotación] Compuesto por 4 números flotantes separados por espacio. Deben existir tantas rotaciones como el número que se indica y deben ir separadas por un fin de línea.

[Número de escalas iniciales del objeto] Es un número entero mayor que 0.

[Parámetros para escala] Son 3 números flotantes separados por espacio. Deben existir tantas escalas como el número que se indica y deben ir separadas por un fin de línea.

Los datos indicados desde Nombre de objeto en adelante deben ir tantas veces como lo indique la cantidad de objetos.

Ejemplo:

```
1
-.78 1.63 1. -0.95 1.4 0. 0. 1. 0.
2
Objeto1
1
0.95 -4.0 -0.95
1
-90 1. 0. 0.
1
1 1 1
Objeto2
1
0. -25.5 0.
2
-90 1. 0. 0.
180 0. 1. 0.
1
1 1 1
```

APENDICE H. CASOS DE PRUEBA

H.1 SEGUIR EL CAMINO

Prueba 2.1 Curso normal	
Descripción	Prueba que el usuario puede seguir el camino que el sistema dibuja.
Entrada	El usuario toca con la punta de la pinza la sección del camino que corresponde y lo recorre sin dejar de tocarlo.
Salida esperada	La sección del camino cambia de color indicando que la colisión fue exitosa.
Prueba 2.2 Curso alternativo: Salir del camino	
Descripción	Prueba que cuando el usuario se sale del camino se registra un error y ocurre una retroalimentación visual para indicar el error.
Entrada	El usuario deja de tocar el camino.
Salida esperada	La sección vuelve a su color original y se registra un error.

Tabla 9. Caso de prueba #2: Seguir el Camino
Fuente: Elaboración Propia

H.2 PASAR A TRAVÉS DE LOS AROS

Prueba 3.1 Curso normal	
Descripción	Prueba que el usuario puede pasar a través de los aros sin tocarlos y tocar el cubo que se encuentra al final de la sucesión de aros.
Entrada	El usuario toca el cubo al final de la sucesión de aros, al mismo tiempo que atraviesa esta sucesión.
Salida esperada	El cubo cambia de color indicándole al usuario que la colisión fue exitosa.
Prueba 3.2 Curso alternativo: Tocar el borde de un aro	
Descripción	Prueba que cuando el usuario toca el borde de un aro se registra un error y ocurre una retroalimentación visual para indicar el error.
Entrada	El usuario toca el borde de un aro.
Salida esperada	El aro se colorea de rojo y se registra un error.

Tabla 10. Caso de prueba #3: Pasar a través de los Aros
Fuente: Elaboración Propia

H.3 COLOCAR LAS FIGURAS

Prueba 4.1 Curso normal	
Descripción	Prueba que el usuario puede tomar una figura y colocarla sobre la sección indicada con el mismo color que la figura.
Entrada	El usuario coloca una figura, previamente atrapada con la pinza, sobre la sección indicada con el mismo color que la figura.
Salida esperada	La figura cambia de color indicándole al usuario que la colisión fue exitosa.
Prueba 4.2 Curso alternativo: Figura equivocada	
Descripción	Prueba que cuando el usuario coloca una figura equivocada se registra un error y ocurre una retroalimentación visual para indicar el error.
Entrada	Coloca una figura del color equivocado sobre la sección indicada.
Salida esperada	La figura se colorea, luego desaparece y se registra un error.
Prueba 4.3 Curso alternativo: Soltar figura	
Descripción	Prueba que cuando el usuario suelta una figura sin colocarla en la sección indicada se registra un error y ocurre una retroalimentación visual para indicar el error.
Entrada	El usuario suelta la figura sin tocar la sección indicada.
Salida esperada	La figura se colorea, luego desaparece y se registra un error.
Prueba 4.4 Curso alternativo: Sección equivocada	
Descripción	Prueba que cuando el usuario coloca una figura en una sección no indicada se registra un error y ocurre una retroalimentación visual para indicar el error.
Entrada	El usuario coloca la figura en una sección que no es la indicada.
Salida esperada	La figura se colorea, luego desaparece y se registra un error.
Prueba 4.3 Curso alternativo: No Coloca	
Descripción	Prueba que cuando el usuario no coloca la figura correctamente en el tiempo previsto se registra un error y ocurre una retroalimentación visual para indicar el error.
Entrada	El usuario no coloca la figura en la sección indicada y finaliza el tiempo previsto.
Salida esperada	Se colorea una nueva sección aleatoriamente y se registra un error.

Tabla 11. Caso de prueba #4: Colocar las Figuras
Fuente: Elaboración Propia

H.4 CORTAR LAS CUERDAS

Prueba 4.1 Curso normal	
Descripción	Prueba que el usuario puede cortar una cuerda en la sección indicada.
Entrada	El usuario cierra la tijera en torno a la sección indicada.
Salida esperada	La sección cambia color indicándole al usuario que la colisión fue exitosa.
Prueba 4.2 Curso alternativo: Sección equivocada	
Descripción	Prueba que cuando el usuario corta una sección equivocada se registra un error y ocurre una retroalimentación visual para indicar el error.
Entrada	El usuario cierra la tijera en torno a una sección no indicada.
Salida esperada	La sección cambia color y se registra un error.
Prueba 4.3 Curso alternativo: No corta	
Descripción	Prueba que cuando el usuario no corta la sección indicada en el tiempo previsto se registra un error y ocurre una retroalimentación visual para indicar el error.
Entrada	El usuario cierra la tijera un torno a la sección indicada y se acaba el tiempo previsto para esto.
Salida esperada	Se pinta una nueva cuerda y se registra un error.

Tabla 12. Caso de prueba #5: Cortar Cuerdas
Fuente: Elaboración Propia

APÉNDICE I. ENCUESTAS APLICADAS DURANTE LAS PRUEBAS

I.1 MEDICIONES DE PRUEBAS FUNCIONALES

I.1.1 Segunda medición a estudiantes

Nombre del Practicante	Ejercicio	Fin	Errores
Olga Gómez	Ej. #1 - Tocar los Colores	Si	2
	Ej. #2 - Seguir el Camino	Si	47
	Ej. #3 - Pasar a través de los Aros	Si	440
	Ej. #4 - Colocar las Figuras	Si	31
	Ej. #5 - Cortar las Cuerdas	Si	17
Yohana Pérez	Ej. #1 - Tocar los Colores	Si	5
	Ej. #2 - Seguir el Camino	Si	109
	Ej. #3 - Pasar a través de los Aros	Si	260
	Ej. #4 - Colocar las Figuras	Si	38
	Ej. #5 - Cortar las Cuerdas	Si	15
Mildred Martínez	Ej. #1 - Tocar los Colores	Si	11
	Ej. #2 - Seguir el Camino	Si	33
	Ej. #3 - Pasar a través de los Aros	Si	120
	Ej. #4 - Colocar las Figuras	Si	35
	Ej. #5 - Cortar las Cuerdas	Si	14
Daniel Pereira	Ej. #1 - Tocar los Colores	Si	2
	Ej. #2 - Seguir el Camino	Si	88
	Ej. #3 - Pasar a través de los Aros	Si	90
	Ej. #4 - Colocar las Figuras	Si	39
	Ej. #5 - Cortar las Cuerdas	Si	17
Juan Carlos de Abreu	Ej. #1 - Tocar los Colores	Si	7
	Ej. #2 - Seguir el Camino	Si	48
	Ej. #3 - Pasar a través de los Aros	Si	230
	Ej. #4 - Colocar las Figuras	Si	31
	Ej. #5 - Cortar las Cuerdas	Si	26

Tabla 13. Segunda medición del sistema con estudiantes
Fuente: Elaboración Propia

I.2 ENCUENTAS

I.2.1 Encuestas de Estudiantes

I.2.1.1 Encuesta # 1

Datos Personales del Estudiante:

Nombre y Apellido: Olga Gómez

Universidad: UCV

Año: 5to

¿Con cuánta frecuencia utiliza el computador?

Diariamente X Una vez a la semana___ Una vez al mes___ Nunca___

Ejercicio #1 - Tocar los Colores:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Cuando aparecían cuadritos cerca y luego aparecía uno más lejos

¿Qué le agregaría al ejercicio?

Sonido

Otras observaciones:

Ejercicio #2 – Seguir el Camino:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Cuando pasaba de una línea a la otra

¿Qué le agregaría al ejercicio?

Que avisara cuando se produce un error

Otras observaciones:

Es difícil saber cuándo se está tocando la línea

Ejercicio #3 - Pasar a través de los Aros:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Todo

¿Qué le agregaría al ejercicio?

Un color diferente cuando se esté dentro de los aros

Otras observaciones:

Ejercicio #4 - Colocar las Figuras:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Colocar las pelotas a la altura ideal para que contara como buena

¿Qué le agregaría al ejercicio?

Otras observaciones:

Ejercicio #5 - Cortar las Cuerdas:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Ubicar dónde estaba la tijera con respecto a la cuerda

¿Qué le agregaría al ejercicio?

Otras observaciones:

Sistema Virtual en General:

¿Qué le pareció?

Ficticio			Real
1	2	3	4

El tiempo de respuesta del sistema lo considera:

Lento			Rápido
1	2	3	4

Otras observaciones:

_____

_____

I.2.1.2 Encuesta # 2

Datos Personales del Estudiante:

Nombre y Apellido: Yohana Pérez

Universidad: UCV

Año: 5to

¿Con cuánta frecuencia utiliza el computador?

Diariamente X Una vez a la semana___ Una vez al mes___ Nunca___

Ejercicio #1 - Tocar los Colores:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Coordinar el movimiento de la pinza y el espacio visual y coordinarlo
Tiempo

¿Qué le agregaría al ejercicio?

Sonido

_____

Otras observaciones:

Ejercicio #2 – Seguir el Camino:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Seguir el camino con la pinza sin alejarla del tablero

¿Qué le agregaría al ejercicio?

Otras observaciones:

Ejercicio #3 - Pasar a través de los Aros:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Todo

¿Qué le agregaría al ejercicio?

Algún indicador para saber cuando se esté dentro de los aros

Otras observaciones:

Ejercicio #4 - Colocar las Figuras:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Colocar las figuras a tiempo

¿Qué le agregaría al ejercicio?

Otras observaciones:

Ejercicio #5 - Cortar las Cuerdas:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Ubicar la tijera con respecto a la cuerda

¿Qué le agregaría al ejercicio?

Otras observaciones:

Sistema Virtual en General:

¿Qué le pareció?

Ficticio			Real
1	2	3	4

El tiempo de respuesta del sistema lo considera:

Lento			Rápido
1	2	3	4

Otras observaciones:

I.2.1.3 Encuesta # 3

Datos Personales del Estudiante:

Nombre y Apellido: Mildred Martínez

Universidad: UCV

Año: 5to

¿Con cuánta frecuencia utiliza el computador?

Diariamente X Una vez a la semana___ Una vez al mes___ Nunca___

Ejercicio #1 - Tocar los Colores:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Coordinar el movimiento de la pinza y pensar en el tiempo que corre

¿Qué le agregaría al ejercicio?

Sonido, y que se mostrar la distancia que se tiene entre la pinza y el a tocar

Otras observaciones:

Ejercicio #2 – Seguir el Camino:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Mantener el pulso

¿Qué le agregaría al ejercicio?

Sonido que diga cuando lo haces bien o mal

Otras observaciones:

Ejercicio #3 - Pasar a través de los Aros:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Mantener la pinza dentro de los aros

¿Qué le agregaría al ejercicio?

Otras observaciones:

Ejercicio #4 - Colocar las Figuras:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Todo

¿Qué le agregaría al ejercicio?

Otras observaciones:

Ejercicio #5 - Cortar las Cuerdas:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Mantener la posición, rotar la tijera

¿Qué le agregaría al ejercicio?

Otras observaciones:

Sistema Virtual en General:

¿Qué le pareció?

Ficticio			Real
1	2	3	4

El tiempo de respuesta del sistema lo considera:

Lento			Rápido
1	2	3	4

Otras observaciones:

I.2.1.4 Encuesta # 4

Datos Personales del Estudiante:

Nombre y Apellido: Daniel Pereira

Universidad: UCV

Año: 5to

¿Con cuánta frecuencia utiliza el computador?

Diariamente X Una vez a la semana___ Una vez al mes___ Nunca___

Ejercicio #1 - Tocar los Colores:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Coordinar la pinza para alcanzar los cuadros sitiados más profundo

¿Qué le agregaría al ejercicio?

Le agregaría sonido

Otras observaciones:

Ejercicio #2 – Seguir el Camino:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Poder relajar el brazo, lo sentía muy tenso

¿Qué le agregaría al ejercicio?

Le agregaría sonido o algún tipo de señal que me indicara cuando cometiendo errores

Otras observaciones:

Ejercicio #3 - Pasar a través de los Aros:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Mantener el pulso

¿Qué le agregaría al ejercicio?

Diferenciar mediante un cambio de colores cuando estoy por fuera o dentro de los aros independientemente si cometo error o no

Otras observaciones:

_____

_____

Ejercicio #4 - Colocar las Figuras:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

_____

_____

¿Qué le agregaría al ejercicio?

_____

_____

Otras observaciones:

_____

_____

Ejercicio #5 - Cortar las Cuerdas:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Ubicarme en el espacio

_____

¿Qué le agregaría al ejercicio?

Otras observaciones:

Sistema Virtual en General:

¿Qué le pareció?

Ficticio			Real
1	2	3	4

El tiempo de respuesta del sistema lo considera:

Lento			Rápido
1	2	3	4

Otras observaciones:

I.2.2 Encuestas de Expertos

I.2.2.1 Encuesta # 1

Datos Personales del Experto:

Nombre y Apellido: Eduardo Troconis

Especialización: Cirujano Pediatra

Lugar donde trabaja actualmente: Instituto de Cirugía Experimental

Número de cirugías abiertas realizadas: más de 1500

Número de cirugías por laparoscopia que ha realizado: más de 50

¿Con cuánta frecuencia utiliza el computador?

Diariamente X Una vez a la semana___ Una vez al mes___ Nunca___

Ejercicio #1 – Tocar los Colores:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	**3**	4
Difícil			Fácil
1	2	3	**4**

¿Qué fue lo más difícil del ejercicio?

El frenado de la goma al contacto con la pinza

¿Qué le agregaría al ejercicio?

El frenado de la goma al contacto con la pinza

Otras observaciones:

Usar un trocar para simular un efecto más real

Ejercicio #2 – Seguir el Camino:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	**3**	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Saber si se está tocando el camino

¿Qué le agregaría al ejercicio?

Otras observaciones:

Ejercicio #3 - Pasar a través de los aros:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

¿Qué le agregaría al ejercicio?

Más precisión a la trayectoria de la pinza

Otras observaciones:

Ejercicio #4 - Colocar las Figuras:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Colocar las figuras en la ubicación correcta

¿Qué le agregaría al ejercicio?

Otras observaciones:

Ejercicio #5 - Cortando las líneas:

¿Qué le pareció el ejercicio?

Frustrante			Satisfactorio
1	2	3	4
Difícil			Fácil
1	2	3	4

¿Qué fue lo más difícil del ejercicio?

Colocar las figuras en la ubicación correcta

¿Qué le agregaría al ejercicio?

Otras observaciones:

Agregaría un nivel de dificultad donde se tuviese que cortar la cuerda punta de la pinza

Sistema Virtual en General:

¿Qué le pareció?

Ficticio			Real
1	2	3	4

El tiempo de respuesta del sistema lo considera:

Lento			Rápido
1	2	3	4

Otras observaciones:

APÉNDICE J. PAPER ENVIADO A CONGRESO CIMENICS 2006

MEASUREMENT OF THE IMPROVEMENT IN ABILITY LAPAROSCOPY USING A VIRTUAL SIMULATOR

Bricelis Urbina
bricelis@opalo.ciens.ucv.ve
José Manuel Motezuma
jmmotezuma@gmail.com
Marvi Martinez
marvi.martinez@gmail.com
Omaira Rodríguez
omaira@opalo.ciens.ucv.ve
Computer Graphics Lab, Faculty of Sciences. Central University of Venezuela.

Abstract. *Laparoscopic surgery is a surgical technique which demands surgeon visuospatial and motor skills that seldom are exercised. The surgical instruments handling and laparoscopic camera navigation are complicated tasks in small spaces such as the abdominal cavity, where the visualized image is magnified and the size of the instruments reduces the skill, eliminates the tactile sensation and reduces kinesthetic force feedback to the surgeon. In addition, the surgeon must interpret a 3D space from images transmitted by video to a 2D monitor. LAPAROS is a developed tool thanks to the multidisciplinary work of the Computer Graphics Laboratory, the Bioengineering Center and the Experimental Surgery Institute at the Central University of Venezuela. This system for laparoscopic surgery training presents a dual platform: mechanical and virtual simulator. The mechanical simulator SIMULAP v.1 has been developed bearing in mind design aspects resembling to the human torso. The virtual simulator provides three training levels: basic, average and advanced, each one focused to develop specific abilities and skills in the surgeons. In this work we present the measurement of the effectiveness of the basic level of our system in the laparoscopic training process. The tests were applied to a group of seven expert surgeons in laparoscopy and to eleven medicine students without any knowledge in the area. These tests consisted in the accomplishment of seven training sessions in the virtual simulator and one in the mechanical simulator. The results were obtained through careful measurement of the movement precision, ability in handling of instruments and speed of execution of the exercises proposed in the mechanical simulator.*

Keywords: *Laparoscopy, Simulation, Virtual Reality, Surgical Training, Tests.*

ANEXO A. UNIFIED MODELING LANGUAJE – UML (LENGUAJE UNIFICADO DE MODELADO)

Es un lenguaje que permite modelar, construir y documentar los elementos que comprende el desarrollo de software. Se ha convertido en el estándar de facto de la industria y su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyecto [43,44].

A.1 DIAGRAMAS UML

A continuación se detallan los diagramas que se utilizan para el desarrollo del sistema planteado:

A.1.1Diagramas de Casos de Uso [21,44,45]

El modelado de Casos de Uso es la técnica más efectiva y simple para modelar los requerimientos del sistema desde la perspectiva del usuario. Se utilizan para definir cómo debería ser el comportamiento del sistema, ya que sólo especifica cómo debe comportarse y no cómo están implementadas las partes que define. El modelo está compuesto por actores y casos de uso

Un actor es un ente con comportamiento, como una persona (identificada por un rol), un sistema u organización y que realiza algún tipo de interacción con el sistema. Es importante destacar el uso de la palabra rol, pues un actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema. Se representa mediante una figura que asemeja la silueta humana.

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema

para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad y se representa mediante una elipse con el nombre del caso de uso en su interior.

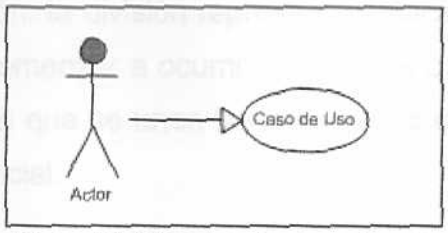

Figura 34. Diagrama de casos de uso
Fuente: Elaboración Propia

A.1.2 Diagrama de Actividades

El diagrama de actividades sirve para modelar el flujo de control entre actividades. En él se puede ver el flujo de actividades a lo largo del tiempo y las tareas que pueden llegar a realizarse a la vez [37].

El diagrama de actividades, gráficamente es un conjunto de arcos y nodos. Desde el punto de vista conceptual, muestra la manera en que ocurren las respuestas a las acciones generadas internamente. Puede estar relacionado con una clase o caso de uso específico y muestra los distintos pasos que se deben realizar para una operación determinada. [37,21]

Este diagrama contiene: [37] (Ver figura 35)

- Actividades: se representan gráficamente con un rectángulo con las puntas curvas, con el nombre de la actividad escrita en su interior. Una actividad, se puede descomponer en sub-actividades representadas con otros diagramas de actividades.

- Transiciones: reflejan el paso de un estado a otro. Se representa gráficamente con un arco dirigido.

- Bifurcaciones: representan una condición en la cual se evalúa una expresión booleana y dependiendo del resultado se puede tomar distintos caminos.

- División y unión: la división representa gráficamente el momento en que pueden comenzar a ocurrir actividades concurrentes, y la unión el momento en que se unen las actividades concurrentes al flujo de control secuencial.

- Carriles: separan las actividades en grupos, cada uno de éstos posee un nombre. Estos grupos representan los responsables en realizar las actividades que se encuentran dentro de su carril.

- Estado inicial: indica el inicio del diagrama de actividades, se representa gráficamente con un círculo relleno.

- Estado final: indica el final del diagrama de actividades, dependiendo de los distintos flujos que se puedan tomar en el diagrama, se puede llegar a estados finales diferentes. Se representa gráficamente con un círculo relleno dentro de otro sin relleno.

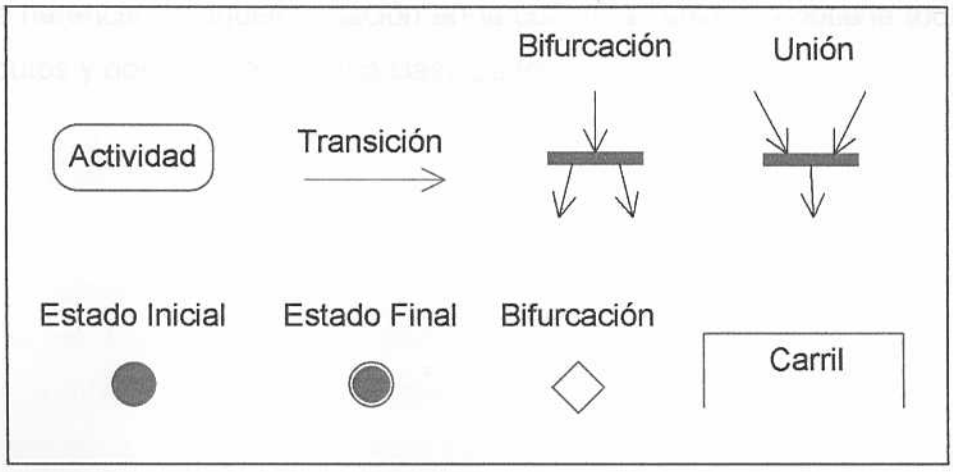

Figura 35. Elementos de un diagrama de actividades
Fuente: Elaboración Propia

A.1.3 Diagrama de clases [30]

El Diagrama de Clases por lo general es el utilizado como base para el análisis y diseño. En él es posible presentar tanto las clases del sistema como las distintas relaciones entre ellas.

Este método de representación de clases incluye definiciones para atributos y operaciones, que pueden ser representados con distintos niveles de visibilidad: (-) privado, (#) protegido y (+) público.

Existen dos tipos de relación a representarse con estos diagramas: la Asociación y la Herencia.

Las asociaciones son las conexiones o enlaces existentes entre las clases, se les pueden especificar las restricciones de multiplicidad:

- Uno y sólo uno.
- (0..1) Cero o uno.
- (M..N) Desde M hasta N.
- (*) Cero o muchos.
- (1..*) Uno o muchos (al menos uno).

La herencia es aquella relación en la cual una clase hijo obtiene todos los atributos y operaciones de una clase padre.

ANEXO B. SIMULAP *V1.0*

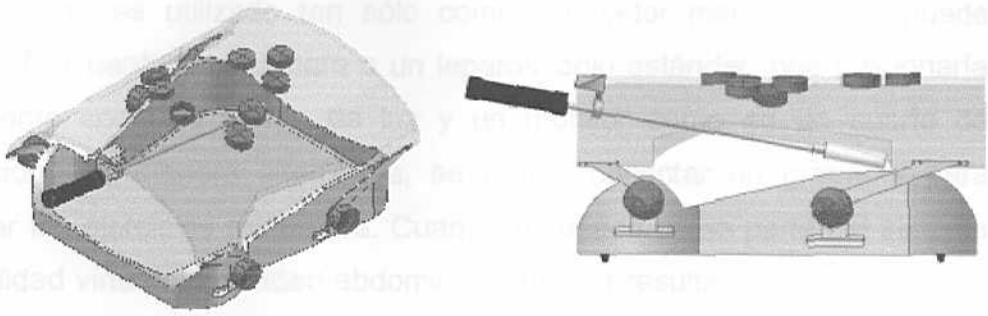

Figura 36. Simulador SIMULAP v1.0
Fuente: [4]

Como parte vital y fundamental del proyecto LAPAROS se propuso utilizar SIMULAP v1.0, un simulador mecánico que se ajusta al tamaño y forma del abdomen del paciente y adicionalmente resulta fácil y cómodo ejercitarse con él.

Este simulador surge como resultado de un Trabajo Especial de Grado desarrollado por 2 estudiantes en el Centro de Bioingeniería de la Universidad Central de Venezuela. Uno de sus objetivos primordiales es proveer un equipo de simulación de uso permanente, amigable y de bajo costo, que permita simular los escenarios de las operaciones laparoscópicas más realizadas en Venezuela.

SIMULAP v1.0 tiene la forma del abdomen masculino. La tapa superior representa la pared abdominal anterior y posee diversas aberturas que simulan las incisiones que un cirujano realiza durante la cirugía. Estos agujeros poseen gomas especiales que simulan resistencia del tejido a la inserción de los instrumentos, donde pueden ser insertados 2 o más trócares, y a través de éstos los instrumentos utilizados en Laparoscopia. La inclinación de la tapa superior puede ser fácilmente modificada, a través de

los soportes de la misma, con la finalidad de simular la inclinación del paciente sobre la mesa de cirugía.

Cuando es utilizado tan sólo como simulador mecánico, se puede acoplar fácilmente una cámara o un laparoscopio estándar, que funcionaría igualmente con una fuente de luz y un monitor como en un cuarto de operación. De manera alternativa, se puede conectar un grabador para analizar los ejercicios realizados. Cuando es usado como parte del sistema de realidad virtual, la cavidad abdominal simulada resulta lo bastante amplia como para emplazar el equipo necesario.

Todos los materiales de los que está hecho el simulador son de bajo costo, fáciles de limpiar y mantener así como también resultan bastante livianos. La tapa superior es hecha en acrílico, la base es de fibra de vidrio y los soportes de aluminio.

Todas estas características hacen de SIMULAP v1.0 un simulador mecánico excelente, que puede ser utilizado como parte de un sistema de realidad virtual como lo es LAPAROS.

ANEXO C. QUATERNIONES

La primera vez que se habló de quaternions se hizo en 1843 por William Rowan Hamilton. Un quaternion es una forma alternativa de representar rotaciones a través de cualquier eje, siendo una extensión de los números reales.

Mientras que los números complejos se forman por la adición de la unidad imaginara *i*, tal que $i^2=-1$, los quaternions se generan de manera similar añadiéndole las unidades imaginarias *i, j, y k* a los números reales tal que $i^2= j^2= k^2= ijk=-1$ [26]. Todo esto puede ser resumido en la siguiente tabla:

	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

Tabla 14. Quaternions
Fuente: [26]

Igualmente los quaternions pueden ser representados por matrices complejas de 2x2 y por matrices reales de 4x4.

El quaternion de la forma $a + bi + cj + dk$ puede ser fácilmente representado de las siguientes maneras [26]:

$$\begin{pmatrix} a-di & -b+ci \\ b+ci & a+di \end{pmatrix}$$

$$\begin{pmatrix} a & -b & d & -c \\ b & a & -c & -d \\ -d & c & a & -b \\ c & d & b & a \end{pmatrix}$$

Se puede pensar en un quaternion como en un vector 4 dimensional con la forma:

$$q = \{ w, x, y, z \} = w + xi + yj + zk$$

Para tener una idea más clara de la representación de una rotación a través de un quaternion se presenta la siguiente figura:

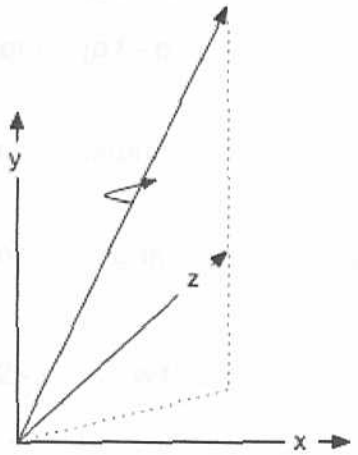

Figura 37. Ejemplo de Quaternion
Fuente: [28]

El vector que describe el quaternion mostrado en la Figura 37, podría denotarse aproximadamente por v = (1.5, 3, 0.5). Por ejemplo, si el ángulo dado es de 30 grados, la escena será rotada estos mismos grados alrededor del eje definido por el vector v [28]. En vez de rotar un objeto a través de una serie de rotaciones, los quaterniones permiten al programador rotarlo a través de un eje y un ángulo de rotación [36].

Los quaterniones poseen varias ventajas comparado con las rotaciones por matrices. Concatenar quaterniones exige menos operaciones y requieren menos espacio para almacenarlos que una matriz.

C.1 OPERACIONES MÁS USADAS [33]

C.1.1 Suma y resta de quaterniones

Teniendo 2 quaterniones

$$p = p_0 + p_1 i + p_2 j + p_3 k \text{ y } q = q_0 + q_1 i + q_2 j + q_3 k$$

$$p + q = (p_0 + q_0) + (p_1 + q_1)i + (p_2 + q_2)j + (p_3 + q_3)k$$

$$p - q = (p0 - q0) + (p1 - q1)i + (p2 - q2)j + (p3 - q3)k$$

C.1.2 Multiplicación de quaterniones

Dados los quaterniones p y q la idea es calcular la rotación combinada *p.q*

$$p.q = (w1.w2 - v1.v2,\ w1.v2 + w2.v1 + v1 \times v2)$$

Donde:

$v1 = (p_1 + p_2 + p_3)$ $\quad$ $w1 = (p_0)$

$v2 = (q_1 + q_2 + q_3)$ $\quad$ $w2 = (q_0)$

C.1.3 Conjugada de un quaternion

Para hacer la conjugada de un quaternion se cambia cada parte imaginaria por su negativo.

Entonces la conjugada de *q* representada por q^* sería:

$$q^* = q_0 - q_{1i} - q_{2j} - q_{3k}$$

C.1.4 Módulo de un quaternion

El módulo de q representado por $|q| = \sqrt{(qq)}$

ANEXO D. DETECCIÓN DE COLISIONES Y ÁRBOLES AABB [3]

La detección de colisiones consiste en determinar en que momento dos o más objetos están intersecándose en un ambiente virtual. Si los objetos son relativamente pequeños (10 o menos triángulos), es posible determinar directamente cuáles triángulos están colisionando. Sin embargo, en objetos un poco más complejos es necesario utilizar algún tipo de procedimiento para poder descartar aquellos triángulos que no están colisionando y probar únicamente aquellos que posiblemente estén colisionando.

El AABB es uno de los volúmenes contenedores más fáciles de construir, consiste simplemente en buscar el menor y mayor valor de las coordenadas de los vértices en cada uno de los ejes (X, Y, Z). Esto puede hacerse con O(N) sin asumir ninguna topología en especial del objeto. Se utiliza la mediana de los puntos ordenados a lo largo del eje de partición para subdividir el AABB durante la construcción de la jerarquía, aunque pueden utilizarse otros estimadores como el promedio.

D.1 CÁLCULO DE INTERSECCION ENTRE AABBS.

Cuando los AABBs no se han rotado, la intersección entre dos AABBs es trivial ya que consiste en proyectar los extremos de los AABBs sobre cada eje y verificar que estos intervalos se solapen, como puede observarse en la Figura 38. Lamentablemente, en la mayoría de las aplicaciones, los AABBs suelen estar rotados, por lo que esta técnica no es aplicable. Cuando los AABBs están rotados debe utilizarse una técnica conocida como prueba de los ejes separadores (o SAT por Separating Axis Test).

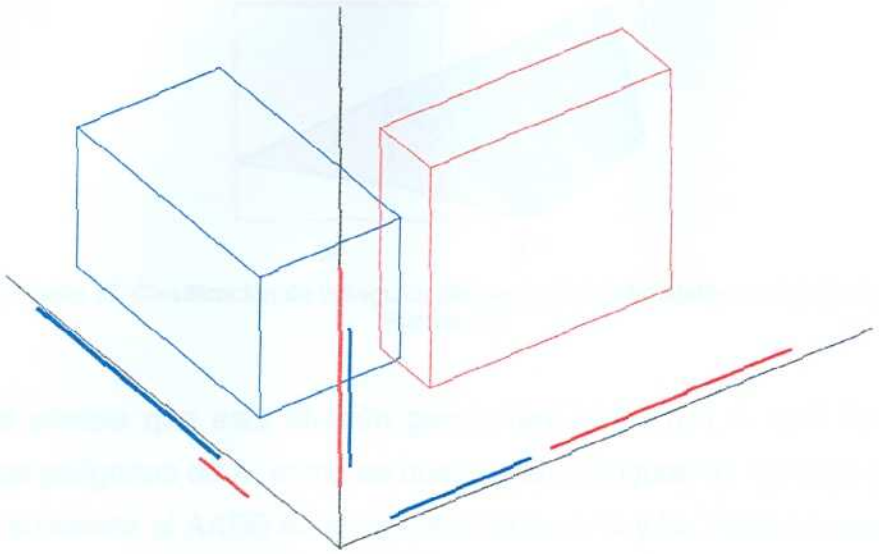

Figura 38. Verificación de intersección entre AABBs
Fuente: [3]

Cada AABB almacenado en memoria contiene el centro, y los medio largos de cada lado. Se almacena el centro en lugar de alguna de las esquinas debido a que el método de detección de intersecciones utilizado (Ejes Separadores) requiere conocer su centro.

Para la construcción de los árboles cada AABB se divide por la mitad en su lado mayor. Por ejemplo, si un AABB A tiene como límites inferior y superior (0, 0, 0) y (4, 2, 6) respectivamente, la primera división a intentar es a lo largo del eje Z, en el plano Z = 3. Para saber en cual AABB hijo quedará cada triángulo, se cuenta el número de vértices del triángulo que queda en cada AABB hijo y se selecciona aquel en el que queden más vértices. Como todo triángulo tiene 3 vértices siempre se podrá decidir en que AABB hijo quedará cada triángulo. Por ejemplo, en la Figura 39 se muestran los AABBs hijos A' y A''. Los triángulos 1 y 2 quedarán en A', mientras que 3 y 4 quedan en A''. Todos los vértices del triángulo 1 están en A'. Del triángulo 2, un sólo vértice está en A'' mientras que dos vértices están en A', por lo que el triángulo se asigna a A'.

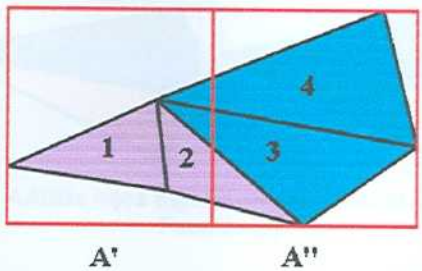

Figura 39. Clasificación de triángulos después de la subdivisión de un AABB
Fuente: [3]

Es posible que esta división genere un AABB hijo A' que encierra a todos los polígonos de A, como se observa en la Figura 40. En este caso, se intenta subdividir el AABB no en la mitad sino en ¼ y ¾. Puede suceder, que después de estas subdivisiones el problema no se resuelva como se muestra en la Figura 41. Cuando esto sucede, se intenta dividir el AABB a lo largo del próximo eje más largo. Siguiendo con el ejemplo propuesto, puede verse en la Figura 42 que en algunos casos esta división si es efectiva. En caso de que esta segunda división todavía no resuelva el problema, se intenta ahora con el último eje de coordenadas. Si este eje de coordenadas también genera hijos no vacíos, el AABB se considera indivisible.

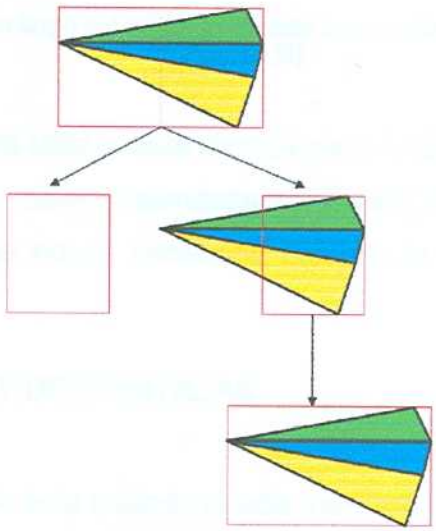

Figura 40. Uno de los hijos tiene a todos los triángulos, y el otro queda vacío
Fuente: [3]

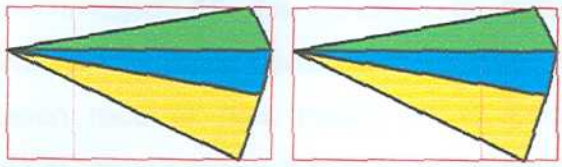

Figura 41. Los AABBs hijos siguen presentando el mismo problema
Fuente: [3]

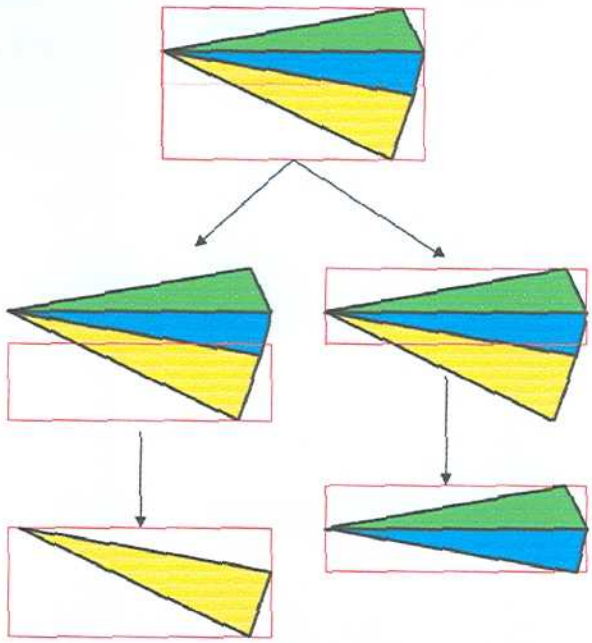

Figura 42. La división a lo largo del segundo eje más largo evita tener AABBs hijos vacíos
Fuente: [3]

La diferencia entre utilizar esta técnica para AABBs y para OBBs es que en AABBs todas las cajas delimitadoras tienen exactamente la misma orientación, por lo que no es necesario calcular la orientación para todas ellas.

D.2 VENTAJAS Y DESVENTAJAS

Como ventajas de éste método puede mencionarse que su construcción es sencilla, así como la detección de intersección entre ellos. Además puede utilizarse para modelos deformables con algunas restricciones.

Entre las desventajas de los AABBs, el tamaño del árbol es generalmente mayor a los árboles construidos usando OBBs, y por lo tanto los algoritmos suelen recorrer más nodos en el árbol. Aunque pueden utilizarse con modelos deformables, imponen restricciones que no siempre pueden cumplirse, como no permitir agregar o eliminar vértices al modelo.

ANEXO E. MATRICES Y TRANSFORMACIONES EN OPENGL® [19,39]

E.1 MATRICES

Es posible determinar un sistema de referencia con la ayuda de tres vectores, cada uno de éstos posee las coordenadas con que se determinan los ejes del sistema. Por ejemplo: el sistema de coordenadas canónico es el formado por los vectores: X= (1,0,0) , Y= (0,1,0) y Z= (0,0,1).

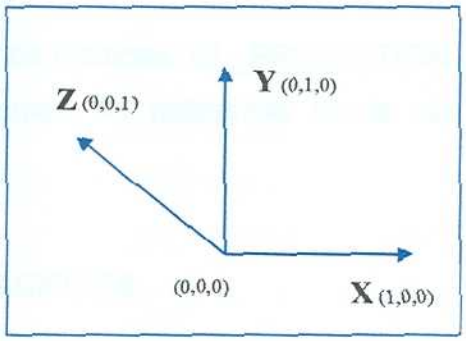

Figura 43. Sistema de coordenada canónico
Fuente: Elaboración Propia

El método utilizado por OpenGL® para la definición de sistemas de referencia es a través de matrices homogéneas 4X4. Estas matrices están formadas por 4 vectores:

$$\begin{pmatrix} a & a' & a'' & x \\ b & b' & b'' & y \\ c & c' & c'' & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Las 3 primeras columnas representan los vectores que forma la base. x= (a, b, c, 0), y=(a', b', c', 0) z=(a", b", c", 0). El cero (0) indica que es un vector.

La última columna indica el desplazamiento. Esto es un punto (x, y ,z, 1) y es el origen del sistema de coordenadas. El uno (1) indica que es un punto.

En el siguiente ejemplo puede observarse la matriz canónica, definida de la manera anteriormente explicada:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

OpenGL® maneja dos matrices: GL_PROJECTION y GL_MODELVIEW, éstas definen los sistemas de referencia de la cámara y los objetos respectivamente.

E.2 TRANSFORMACIONES

Las transformaciones son un aspecto muy importante de la programación gráfica. Existen tres tipos: las rotaciones, las translaciones y los escalados.

E.2.1 Traslaciones

Esta transformación es utilizada cuando deseamos mover una imagen de una posición a otra de la pantalla. La matriz de transformación que hay que aplicar para obtener un movimiento de Tx en el eje X, de Ty en el eje Y y de Tz en el eje Z es:

$$\begin{pmatrix} Bx \\ By \\ Bz \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & Tx \\ 0 & 1 & 0 & Ty \\ 0 & 0 & 1 & Tz \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} Ax \\ Ay \\ Az \\ 1 \end{pmatrix}$$

Esta transformación en muy simple y puede ser llevada a cabo sin necesidad de multiplicar las matrices con el punto y la transformación: basta simplemente con sumarle a cada componente de la matriz el desplazamiento que se desea trasladar.

$$\begin{pmatrix} Bx \\ By \\ Bz \\ 1 \end{pmatrix} = \begin{pmatrix} Tx + Ax \\ Ty + Ay \\ Tz + Az \\ 1 \end{pmatrix}$$

E.2.2 Escalado

El escalado aumenta o disminuye las dimensiones de una imagen. La escala aplicada no siempre produce una nueva imagen más grande o más pequeña proporcional a la original. Existen 3 factores (Ex, Ey, Ez) para especificar la cantidad en la que se ha de escalar el eje X, el Y y el Z. Únicamente en caso de que todos los factores coincidan la imagen resultante será proporcional a la original.

La matriz de transformación que hay que aplicar para obtener un escalado de Ex en el eje X, de Ey en el eje Y y de Ez en el eje Z es:

$$\begin{pmatrix} Bx \\ By \\ Bz \\ 1 \end{pmatrix} = \begin{pmatrix} Ex & 0 & 0 & 0 \\ 0 & Ey & 0 & 0 \\ 0 & 0 & Ez & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} Ax \\ Ay \\ Az \\ 1 \end{pmatrix}$$

El tamaño de cada eje es multiplicado por el factor que le corresponda. En el caso que un factor de escalado sea igual a 1, esa dimensión de la imagen permanecerá con las mismas dimensiones en la imagen original.

E.2.3 Rotaciones

Son giros que se le aplican al sistema de referencia en torno a un eje con un determinado ángulo de giro.

Al disponer de 3 ejes de coordenadas podemos realizar tres tipos de rotaciones: una rotación en el eje X, en el eje Y o en el eje Z. Por ejemplo, una rotación en el eje X quiere decir que se realiza la rotación moviendo el eje Y hacia el eje Z, de forma que el X permanece inmóvil.

Las matrices de transformación para las rotaciones dado un ángulo α son las siguientes:

Para rotaciones en el eje X:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{Cos}(\alpha) & -\text{Sen}(\alpha) & 0 \\ 0 & \text{Sen}(\alpha) & \text{Cos}(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Para rotaciones en el eje Y:

$$\begin{pmatrix} \text{Cos}(\alpha) & 0 & \text{Sen}(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ -\text{Sen}(\alpha) & 0 & \text{Cos}(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Para rotaciones en el eje Z:

$$\begin{pmatrix} \text{Cos}(\alpha) & -\text{Sen}(\alpha) & 0 & 0 \\ \text{Sen}(\alpha) & \text{Cos}(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Estas transformaciones son utilizadas para colocar los objetos en la posición, rotación y tamaño final deseado. Con la ayuda de éstas también es posible crear escenas animadas

Cuando se aplican varias transformaciones sobre un objeto, es muy importante el orden en que éstas se realicen. El resultado puede ser muy distinto si se cambia el orden en que se aplican las transformaciones. Por ejemplo: si se le aplican transformaciones de rotación y translación a un objeto el resultado sería distinto a aplicar primero la translación y luego la rotación.

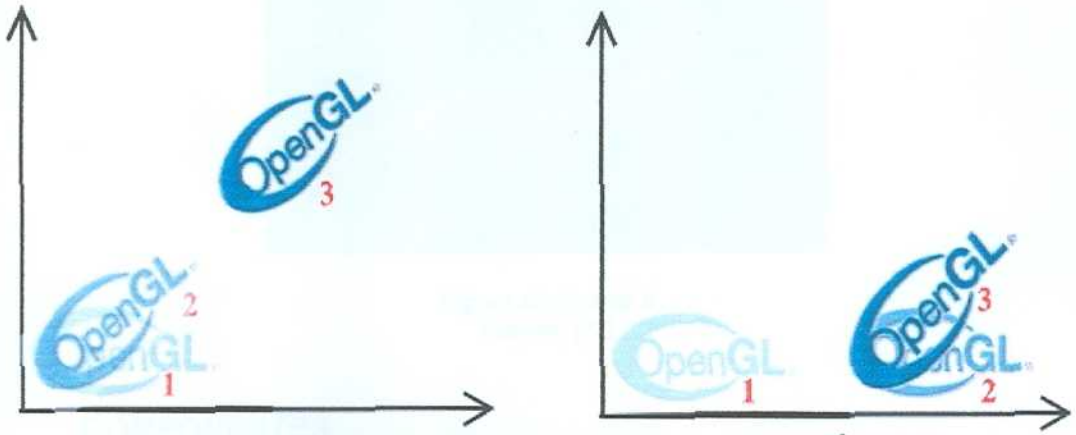

Figura 44. Secuencia de transformaciones en OpenGL®
Fuente: [18]

ANEXO F. POLHEMUS ISOTRAK II

El Polhemus Isotrak II es una solución rentable para rastreo y conversión digital 3D con seis grados de libertad. Es la respuesta perfecta para capturar posición y orientación, requisitos indispensables para las aplicaciones y ambientes 3D. Ideal para rastreo de la cabeza y el análisis biomecánico.

Figura 45. Isotrak II
Fuente: [22]

COMPONENTES

El sistema de ISOTRAK II incluye una Unidad Electrónica del Sistema (SEU por sus siglas en inglés), una fuente de poder, un receptor y un transmisor

- Unidad Electrónica del Sistema: Contiene el hardware y el software necesarios para generar y detectar los campos magnéticos, calcular la posición y orientación y la interfaz con el computador vía un puerto de alta velocidad Rs-232 (de hasta 115.2K baudios).

- Transmisor: El transmisor consta de un trío de bobinas electromagnéticas, dentro de una cáscara plástica, que emite los campos magnéticos. El transmisor es el sistema de referencia del sistema para las medidas del receptor.

- Receptor: El receptor consta de un trío de pequeñas bobinas electromagnéticas, dentro de una cáscara plástica, que detecta los campos magnéticos emitidos por el transmisor. El receptor es un cubo ligero al cual se le mide su posición y orientación exacta mientras se mueve. El receptor es totalmente pasivo, no teniendo ningún voltaje activo aplicado a él.