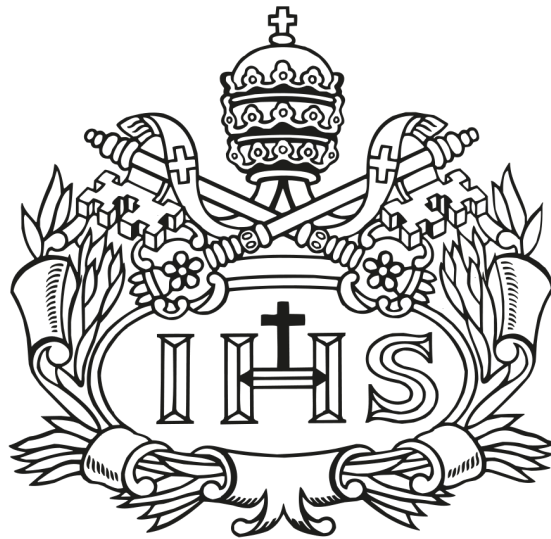


Documento de Diseño Para Proyecto Laboratorio de Marcha

Juan Pablo Peñaloza Botero y José Rafael Domínguez Nolasco



Pontificia Universidad
JAVERIANA
— Colombia —

Introducción a la Computación Gráfica

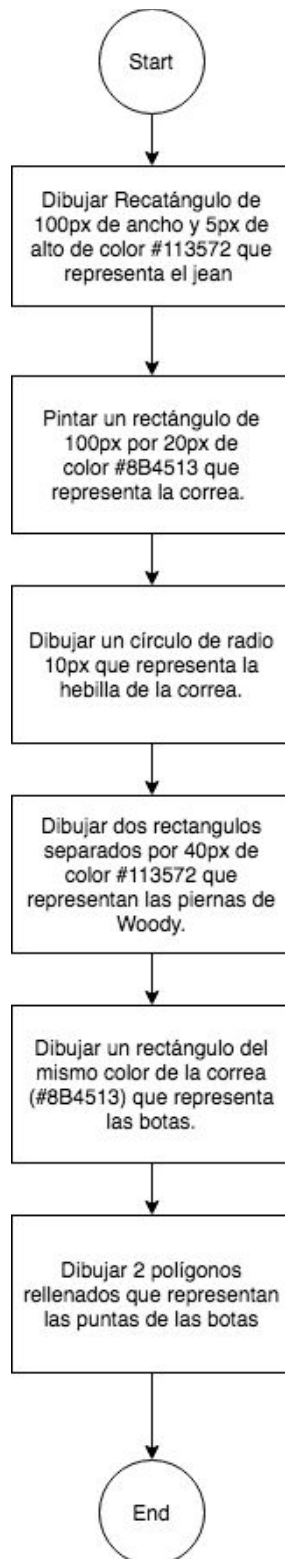
Análisis

Objetos Conocidos	<ul style="list-style-type: none">● La silueta del personaje a dibujar.● Algoritmos para dibujar:<ul style="list-style-type: none">○ Líneas○ Elipses○ Círculos● Dimensiones del avatar.
Objetos Desconocidos	<ul style="list-style-type: none">● Coordenadas exactas de los detalles del personaje.● Tamaño en píxeles de las extremidades.● Colores del personaje en formato openGl.
Condiciones	<ul style="list-style-type: none">● El personaje debe tener piernas y articulaciones definidas para que más adelante se pueda animar.● Debe caber en la ventana.● Utilizar los algoritmos vistos en clase.

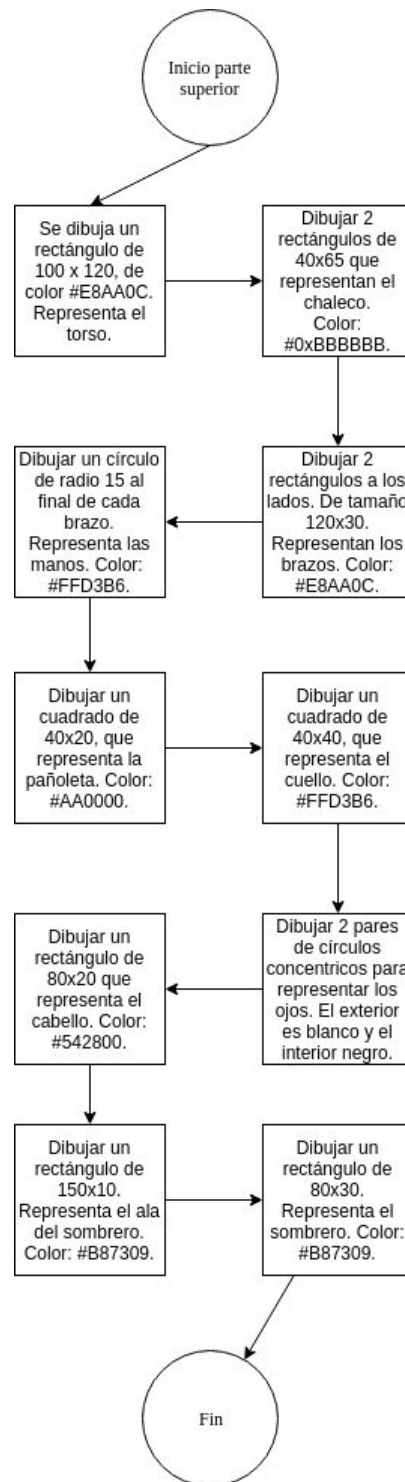
Diseño

Entradas	<ul style="list-style-type: none">● Imagen del personaje a dibujar.● Dimensiones del personaje.● Dimensiones de la pantalla.
Salidas	<ul style="list-style-type: none">● Coordenadas exactas de los detalles del personaje.● Dibujo del personaje en OpenGL.
Condiciones	<ul style="list-style-type: none">● Se debe utilizar los métodos vistos en clase de openGl● El personaje debe tener piernas para posteriormente animar.● Utilizar los algoritmos de elipse, círculo y línea vistos en clase.

Dibujar Cadera, Piernas y Pies



Dibujar Cabeza, Brazos y Torso



Problemas Encontrados

Problema: Cuando se fue a compilar la estructura básica del programa en el sistema operativo MacOS, cambiaban un poco el nombre de las librerías para Linux y para Mac.

Solución: Se agregó un `#if define` para que importe las librerías para Mac en caso de que se compile en Mac o de Linux en caso de que se compile en Linux.

Problema: Los colores en la mayoría de escenarios están descritos en RGB donde cada canal es un número de 0 a 255, en OpenGL también son los mismos tres canales pero de 0.0 a 1.0, por lo tanto al momento de buscar un color no se encontraba en términos de 0 a 1.0.

Solución: Se divide cada canal por 255 para sacar el porcentaje, pero para hacer las cosas más automáticas encontramos una página que convierte cualquier formato de color a formato de OpenGL.

Problema: Es común encontrar colores en notación hexadecimal, sin embargo, OpenGL no cuenta con funciones dedicadas para utilizar esta notación.

Solución: Se creó una función que recibe los valores hexadecimales de los colores como un número entero, y procede a hacer la conversión para poder ser utilizado con OpenGL.

Segunda Entrega

Análisis

Objetos Conocidos	<ul style="list-style-type: none">● La figura que se quiere dibujar en 3D que es Woody de Toy Story● Figuras geométricas en 3D para realizar el dibujo.● Dimensiones del avatar.
Objetos Desconocidos	<ul style="list-style-type: none">● Tamaño del escenario.● Posición exacta de cámaras
Condiciones	<ul style="list-style-type: none">● Marcar bien las articulaciones para más adelante relacionarlas con los marcadores.

Diseño

Entradas	<ul style="list-style-type: none">● Boceto de figura en 3D.● Posición de los marcadores.
Salidas	<ul style="list-style-type: none">● Escenario con paredes y pasarela.● Dibujo del personaje en OpenGL.● Cámaras.
Condiciones	<ul style="list-style-type: none">● Debe estar en 3D● Deben haber cámaras.● Definir las articulaciones

Acta de Evaluación

Observación: El documento de diseño es correcto a nivel del programa, pero no incluye información sobre el diseño del dibujo, sobre los elementos usados (líneas, polígonos, círculos), sus posiciones y colores.

Solución: Se creó una sección en el documento de diseño para explicar la forma en que decidimos diseñar e implementar el escenario y el avatar, ahí se mencionan los colores y las figuras utilizadas.

Observación: Esto involucra también un proceso creativo que hay que reflejar de alguna manera

Solución: Se creó una sección para explicar cómo se nos ocurrió el personaje y en las secciones siguientes esta explicada la creatividad utilizada para implementar el personaje.

Observación: No vel en el documento tampoco relación de los elementos dibujados con los marcadores del proyecto.

Solución: En la sección que explica cómo se implentó el personaje están mencionadas las articulaciones y como vamos a resaltar las articulaciones relacionadas con los marcadores.

Información De Diseño Del Personaje

La elección del personaje fue muy rápida, queríamos hacer un avatar que sea familiar. Elegimos a Woody de la película Toy Story. Nos parece un personaje que se puede reconocer de inmediato y cuenta con la ventaja que ya está su figura en 3D. También discutimos sobre hacer personajes de Minecraft ya que su implementación sería simple de realizar pero no creíamos que contaba con el nivel de detalle que se requería para la clase.

Implementación del Escenario

Nuestro punto de partida fue implementar el escenario que comprendía dos paredes y una alfombra. Decidimos empezar por esto ya que nos queríamos familiarizar con la creación de elementos en OpenGL y que de alguna manera se volviera intuitivo para nosotros y pensamos que el escenario era más fácil de hacer que el avatar.

Primero empezamos en papel, dibujando un sistema de coordenadas y definiendo las dimensiones de nuestro espacio. Para empezar a tener una mejor perspectiva de la ventana, pusimos seis cubos en las diferentes esquinas de nuestro espacio y colocar la cámara de tal forma que los puntos se vean desde la misma perspectiva en la ventana como en el papel.

Para hacer las paredes lo que hicimos fue poner un GL_QUADS en el plano xy para la pared del fondo, otro en el plano yz para la pared izquierda y otro en el plano xz para representar el piso. La implementación de la alfombra fue más ingeniosa. Tomamos el piso y le aplicamos la matriz de escalamiento con un factor menor a 1 en el eje z al QUADS del piso y lo trasladamos de tal forma que quede en el centro. La pared izquierda la coloreamos de gris y la del fondo de blanco para obtener un mejor sentido de profundidad. El piso lo coloreamos de blanco y la alfombra de rojo.

Seudo Código:

1. Definir $maxX$, $maxY$, $maxZ$, $minX$, $minY$, $minZ$ como los valores máximo y mínimos de cada eje.
2. Hacer un Quads de $(minX, minY, maxZ)$, $(maxX, minY, maxZ)$, $(maxX, minY, minZ)$ y $(minX, minY, minZ)$ que representa el piso.
3. Hacer un Quads de $(minX, maxY, minZ)$, $(minX, maxY, maxZ)$, $(minX, minY, maxZ)$ y $(minX, minY, minZ)$ que representa la pared de la izquierda.
4. Hacer un Quads de $(minX, maxY, minZ)$, $(maxX, maxY, minZ)$, $(maxX, minY, minZ)$ y $(minX, minY, minZ)$ que representa la pared del fondo.
5. Escalar y trasladar el piso para que represente la alfombra.

Implementación de Woody

Esta sin duda fue la parte más desafiante del proyecto para nosotros. Para empezar a armar el avatar primero localizamos la cámara de tal forma que se vea de frente el avatar para tener una mejor perspectiva. Empezamos con una esfera para la cabeza y utilizamos la matriz de traslación para ponerla en la alfombra roja. Para el cuello del avatar utilizamos también una esfera que la escalamos de tal forma que quedara alargada y debajo de la cabeza.

El cuerpo del personaje fué lo que más nos dio trabajo y lo que más discutimos ya que no hay ninguna figura geométrica primitiva que sea parecida al cuerpo de nuestro personaje entonces recurrimos a utilizar un prisma como el tronco y otro prisma como la cadera.

La extremidades fueron también prismas a los cuales les aplicamos matrices de escalamiento y traslación y en el caso de los brazos la matriz de rotación. En cada articulación que leen los marcadores pusimos una esfera verde para identificarlos.

Seudo Código:

1. Dibujar, trasladar y escalar esfera para hacer cabeza.
2. Dibujar, trasladar y escalar esfera para hacer cuello.

3. Dibujar, trasladar y escalar un cubo para hacer el tronco.
4. Dibujar, trasladar y escalar un cubo para hacer la cadera.
5. Dibujar, trasladar, escalar y rotar cubo para hacer brazo izquierdo.
6. Dibujar, trasladar, escalar y rotar cubo para hacer brazo derecho.
7. Dibujar, trasladar, escalar y rotar pirámides para hacer las botas