Sistemas Gráficos – 66.71

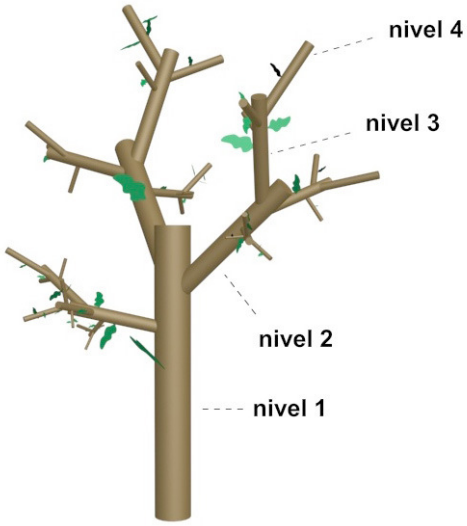
Trabajo Práctico Nº2
Curvas y Transformaciones – 1er. Cuat. 2009

Punto 1

Implementar una aplicación en OpenGL que permita diseñar y generar una serie de árboles 3D siguiendo un algoritmo fractal y 2 curvas dibujadas por el usuario.

Conceptos básicos aplicados

Se deberá crear un modelo o patrón básico (rama patrón) que se repetirá en los distintos niveles del árbol con diferente escala, rotación y posición. Este patrón consistirá de un cilindro (usar primitiva OpenGL para cilindros) y 3 hojas que se modelarán según una curva Bezier detallada más adelante.
Luego el árbol de conformará agregando 3 ramas secundarias que se insertarán en la rama principal en puntos específicos, con un ángulo y una escala proporcionalmente menor relativa a la rama padre, este proceso se repetirá recursivamente según la cantidad de niveles que tenga el árbol.



Arbol (4 niveles)

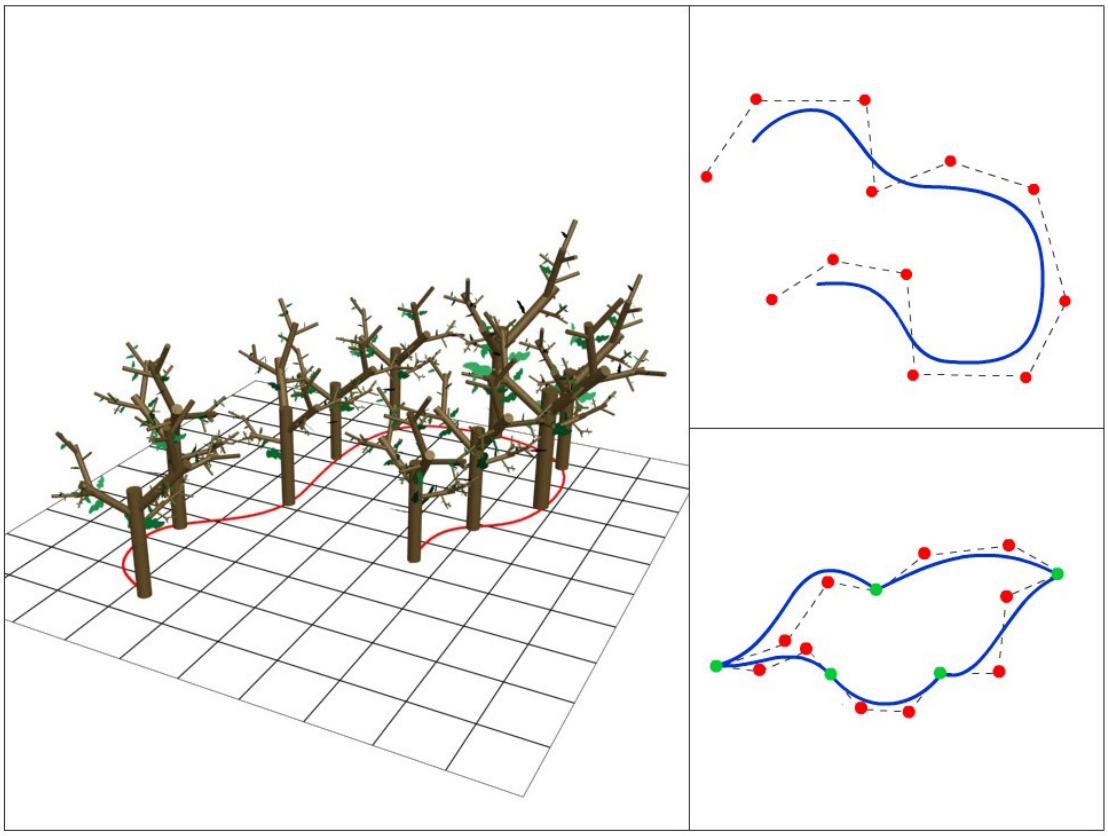
Rama patrón

Rama patrón
(vista superior)

Como se observa en la figura, los 3 cilindros con bandas rojas y blancas representan a las ramas secundarias cuya forma en realidad es la misma que la de la rama patrón (un cilindro con 3 hojas).
Se deberá respetar la forma general planteada en este ejemplo aunque los ángulos y escalas relativos de las ramas secundarias no necesitan ser exactamente idénticos. Es importante que en el resultado final las hojas y el tronco sean claramente visibles.

Interface de usuario

Se deberá implementar una aplicación con las siguientes vistas:



- Editor de recorrido (ventana superior derecha): Deberá permitir editar el grafo de control de una curva B-Spline cúbica utilizando el mouse. Funcionalidades mínimas: agregar punto de control, borrar curva. Deberá representarse la curva resultante, los puntos de control y el grafo de manera clara como se ve en la figura. La curva generada definirá la ubicación de los árboles sobre el plano.

- Editor de hoja (ventana inferior derecha): Deberá permitir editar una curva de Bezier con N puntos de control que deberá ser cerrada. Dado que cada tramo de Bezier requiere 4 puntos de control, se aceptará que el cierre se realice uniendo el último punto de la curva con el comienzo mediante una línea recta. El cierre podría ejecutarse con un comando de teclado ya que la cantidad de puntos de control total no está predefinida.
  La curva resultante se utilizará para generar una hoja usando la primitiva de OpenGL “GL_TRIANGLE_FAN”.
  Funcionalidades mínimas: agregar punto de control, borrar curva.
  Notar que el primer punto de cada tramo tiene color verde.

- Escena 3D (ventana izquierda): en base a las curvas de los paneles del lado derecho se deberán construir: la rama patrón (tronco mas hojas), modelo del árbol, las múltiples instancias del árbol siguiendo el recorrido.

Además deberá ser visible la grilla y la curva de recorrido.
La escala y orientación de cada instancia de árbol sobre el recorrido deberá variar en forma aleatoria. La cantidad de árboles sobre la curva será un parámetro (ver más adelante).
La generación de los árboles se ejecutará con un comando de teclado (tecla "g"). Al arrastrar el mouse sobre esta ventana el punto de vista deberá orbitar alrededor del centro de la grilla (variación en 2 ejes según movimiento X e Y del mouse). Es optativa la implementación de la función de zoom.

Parámetros globales de la aplicación

Podrán ser constantes globales en el código fuente o levantadas desde un archivo de texto. Es importante que el día de la entrega estos parámetros puedan modificarse para hacer diferentes pruebas.

- Número de niveles del árbol: controla el nivel de recursividad del algoritmo que genera los árboles.

- Subdivisiones por tramo de curva BSpline: define el paso de discretización de cada tramo BSpline. Un tramo corresponde al generado por 4 ptos. de control. Si el parámetro es 10, quiere decir que se evaluará la curva en u=0, 0.1, 0.2 etc.

- Subdivisiones por tramo de curva Bezier: idem anterior pero para Bezier.

- Cantidad de árboles por tramo BSpline: define cuantos árboles hay para el rango U=0 .. 1 del tramo BSpline del recorrido. Si por ejemplo el valor es 4, el cada tramo deberá ubicarse un árbol en U=0, 0.25, 0.50 y 0.75.

Implementación

Deberán implementarse los algoritmos de generación de las curvas BSpline y Bezier requeridas en la aplicación.
Junto con este enunciado se adjunta el framework necesario que incluye la el dibujado grilla y la configuración del modo de iluminación en OpenGL.
Para al algoritmo recursivo de construcción del árbol deberán utilizarse transformaciones de OpenGL junto con los mecanismos glPopMatrix y glPushMatrix. Para modelar cada rama (rama patrón mas hojas) no hace falta crear más que una sola función que llamada en diferentes contextos (diferentes estados de la matriz de transformación) generará las ramas de los diferentes niveles.

Punto 2

Extender el algoritmo de Liang-Barsky para un viewport definido por un polígono en lugar de un rectángulo. Explicar que cambios son necesarios y detallar el algoritmo modificado.

Fecha de entrega

22 de Mayo de 2009

Informe

Explicar la arquitectura de la aplicación, detallando la estructura de clases y decisiones de diseño, que se hayan tomado. No se requiere incluir el código fuente. Se deberán Incluir en el pie de cada página los siguientes datos:

- Trabajo práctico: Nro. 2
- Cuatrimestre: 1er cuatrimestre. 2009
- Integrantes del grupo: padrón, nombre y apellido
- Fecha de entrega
- Nro. de página