Instituto Tecnológico y de Estudios Superiores de Monterrey Campus Estado de México
Escuela de Ingeniería y Ciencias
Departamento de Computación

Proyecto 03. Curvas. Gráficas computacionales.

Apegándome al Código de Ética de los Estudiantes del Tecnológico de Monterrey, me comprometo a que mi actuación en este examen esté regida por la honestidad académica. Acepto: _____

Profesor: Oriam Renan De Gyves López	Clave y grupo: TC3022. ____
Alumno: _____	Matrícula: _____

Lee cuidadosamente todas las instrucciones antes de comenzar a resolver cada uno de los problemas.

Fecha de entrega: 10 de marzo, antes de las 23:59 horas.

NOTA MUY IMPORTANTE: Este proyecto debe ser elaborado de manera individual. Puedes trabajar de manera colaborativa para resolver dudas conceptuales, pero el código debe ser realizado únicamente por ti. Recuerda citar correctamente todas las fuentes de inspiración que ocupaste para la realización del mismo. Cualquier indicio de copia, trampa o fraude será penalizado con 1 de calificación y será reportado al comité de integridad académica. Se penalizará tanto al copiado como al copiador.

Documentación:
Manual de referencia de C++: http://www.cplusplus.com/reference/
Documentación de OpenGL: http://docs.gl/

Deberán leer:
`glGenVertexArrays`: http://docs.gl/gl4/glGenVertexArrays
`glDeleteVertexArrays`: http://docs.gl/gl4/glDeleteVertexArrays
`glBindVertexArray`: http://docs.gl/gl4/glBindVertexArray
`glGenBuffers`: http://docs.gl/gl4/glGenBuffers
`glDeleteBuffers`: http://docs.gl/gl4/glDeleteBuffers
`glBindBuffer`: http://docs.gl/gl4/glBindBuffer
`glBufferData`: http://docs.gl/gl4/glBufferData
`glEnableVertexAttribArray`: http://docs.gl/gl4/glEnableVertexAttribArray
`glVertexAttribPointer`: http://docs.gl/gl4/glVertexAttribPointer
Concoide de Nicomedes: http://mathworld.wolfram.com/ConchoidofNicomedes.html
Concoide de Nicomedes: http://xahlee.info/SpecialPlaneCurves_dir/ConchoidOfNicomedes_dir/conchoidOfNicomedes.html

Evaluación
El proyecto será evaluado utilizando los siguientes criterios:

100	El proyecto cumple con todos los requerimientos.
1	El programa fuente contiene errores sintácticos.
1 - 99	El programa produce errores al momento de correrlo.
Falta a la integridad académica	La solución es un plagio.

Nota: *Todas las escenas deberán estar cargadas en el* **`scene_manager`**.

I. (50 puntos) Dibuja una configuración del Concoide de Nicomedes. Puedes conocer más acerca del Concoide en la sección "***Deberán leer***" de este documento. Codifica tu resultado en una escena llamada **`scene_conchoid`**.

II. (50 puntos) Utilizando curvas de Chaikin, dibuja a tu personaje de "caricatura" favorito. Agrega todo el detalle necesario (no sólo el contorno) para que sea reconocible fácilmente. Dibuja el polígono original utilizando una línea discontinua, y la curva final utilizando una línea continua. Permite prender o apagar el dibujado de estas líneas utilizando input de teclado. Utiliza el número de refinamientos que consideres necesarios para tu dibujo, sin embargo, deberá ser mínimo 1 refinamiento. Codifica tu resultado en una escena llamada **`scene_chaikin`**.

Algoritmo de Chaikin para curvas

General

En 1974, George Chaikin dio una clase en la Universidad de Utah en la que presentó una técnica para generar curvas a partir de un número limitado de puntos. Este algoritmo es interesante ya que fue uno de los primeros algoritmos de refinamiento ("corner cutting"), especificado para generar una curva a partir de un conjunto de puntos de control. El algoritmo fue olvidado durante muchos años por la comunidad de gráficas computacionales debido a la gran actividad de estudio sobre curvas B-spline. En los últimos años, los investigadores se han alejado de los modelos analíticos y el paradigma de Chaikin es ahora utilizado ampliamente para generar distintos tipos de curvas y superficies.

El paradigma de refinamiento (corner cutting)

Los investigadores desde Bezier habían estado trabajando en curvas generadas por medio de puntos de control, pero habían enfocado su análisis en la representación analítica, basada en los polinomios de Bernstein. Chaikin tenía una idea diferente y decidió utilizar modelos geométricos para trabajar con los puntos de control directamente. Su método de generación de curvas está basado en refinamiento, en donde el algoritmo genera nuevos puntos de control al cortar las esquinas del polígono original. La figura 1 ilustra esta idea, en donde los puntos de control originales han sido refinados al cortar las esquinas de la primera secuencia.

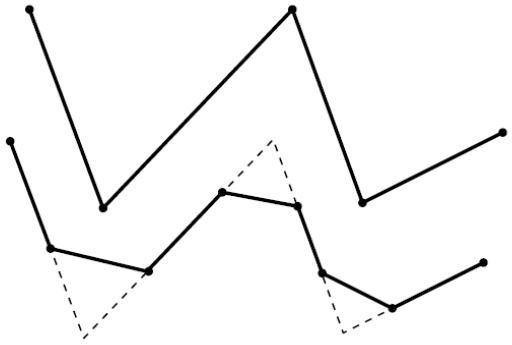
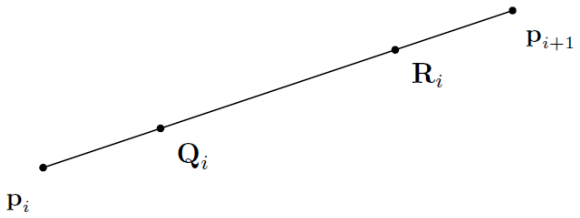


Figura 1. Claramente, se podría tomar el segundo polígono y cortar sus esquinas, produciendo una tercera secuencia. En el límite, tendríamos una curva. Esta es la idea de Chaikin.

El método de Chaikin

Chaikin utilizó razones fijas al cortar esquinas, de tal manera que todas las líneas fueran cortadas de la misma manera. Al representar esto matemáticamente, el método de Chaikin se expresa de la siguiente manera: dado un polígono de control $\{P_0, P_1, \dots, P_n\}$, lo refinamos al generar una nueva secuencia de puntos de control $\{Q_0, R_0, Q_1, R_1 \dots, Q_{n-1}, R_{n-1}\}$, en donde cada par de puntos Q_iR_i son calculados a razón de $\frac{1}{4}$ y $\frac{3}{4}$ entre los extremos del segmento de línea $\overline{P_iP_{i+1}}$.

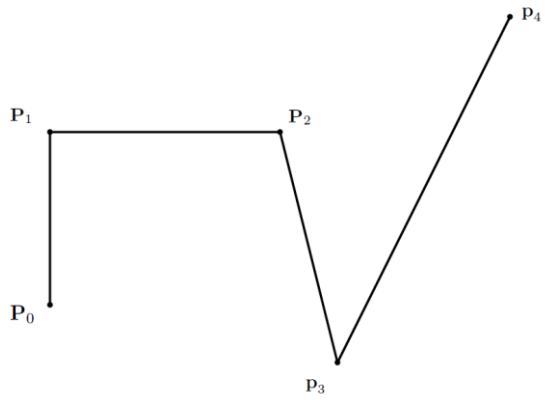


En donde:

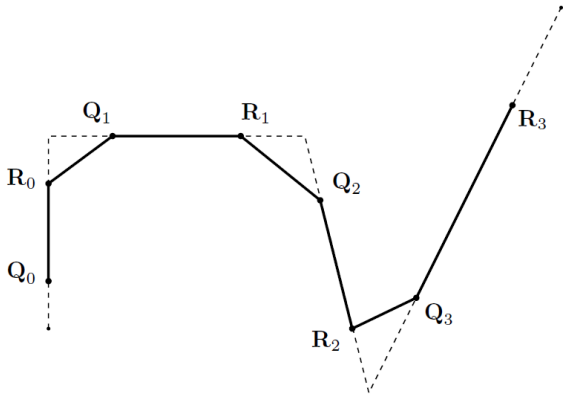
$$Q_i = \frac{3}{4}P_i + \frac{1}{4}P_{i+1}$$
$$R_i = \frac{1}{4}P_i + \frac{3}{4}P_{i+1}$$

Estos $2n$ nuevos puntos son considerados el nuevo polígono de control – una refinación del polígono de control original.

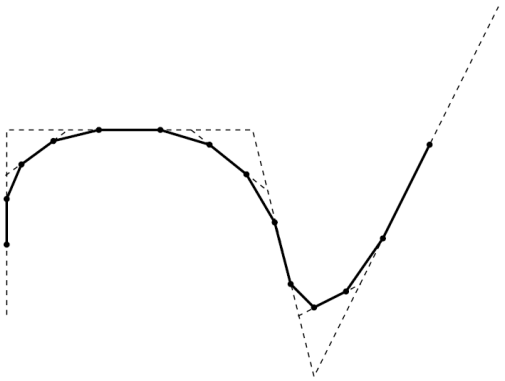
Ejemplo – Funcionamiento del algoritmo de Chaikin
Considere el siguiente polígono:



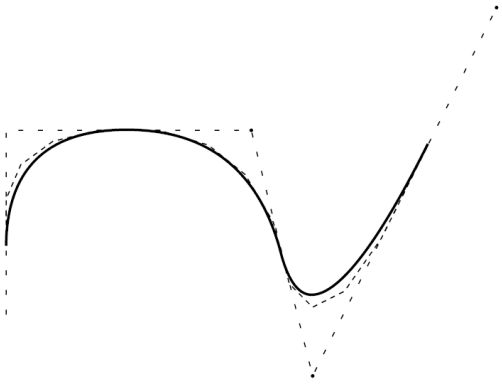
El algoritmo de Chaikin genera los puntos Q_i y R_i y usa estos puntos para refinar la curva y obtener el polígono de control mostrado en la siguiente imagen:



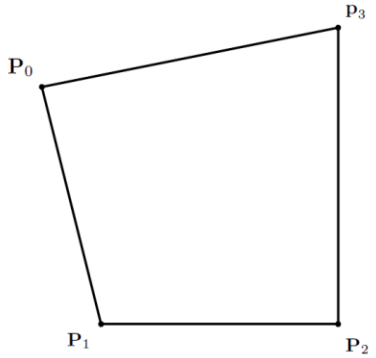
Estos puntos son ahora utilizados para generar una nueva refinación:



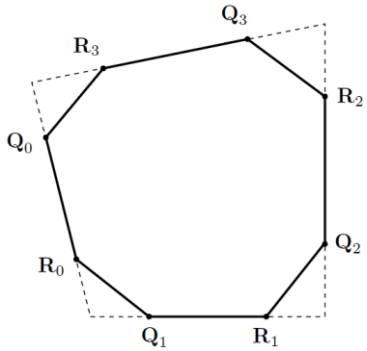
Y de nuevo, estos puntos son utilizados para generar una nueva refinación, etc. La siguiente ilustración muestra el polígono de control inicial, el tercer polígono de control en la secuencia, y la curva final:



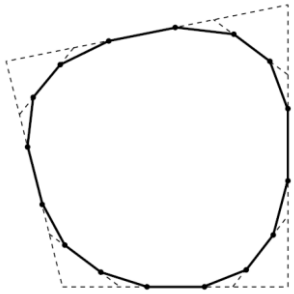
Ejemplo – Una curva cerrada
Para ilustrar la curva de Chaikin en un polígono de control cerrado, considera la siguiente figura:



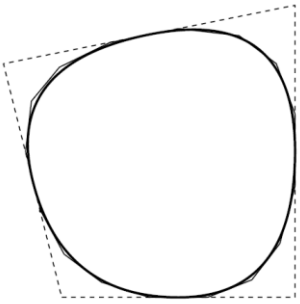
En este caso, los índices del punto de control son tomados con el módulo $n + 1$ (o 4, en el caso de esta figura). Podemos aplicar el método de esta figura, obteniendo:



Este nuevo polígono de control puede ser utilizado para obtener un segundo refinamiento como en la siguiente figura:



Es claro que podemos continuar este proceso indefinidamente. Para propósitos de gráficas computacionales, nos detenemos después de un número de refinamientos y aproximamos la curva conectando los puntos del polígono de control resultante con líneas rectas. El polígono de control inicial y la segunda refinación, con la curva resultante, son mostradas en la siguiente figura.



Resumen
Chaikin mostró un esquema simple en el cual se pueden especificar curvas a partir de un polígono de control. La idea es única dado que la descripción matemática es ignorada en favor de un algoritmo geométrico que simplemente selecciona nuevos puntos de control a lo largo de segmentos de línea del polígono de control original. Sin embargo, se ha demostrado que las curvas de Chaikin son equivalentes a una curva B-spline cuadrática (curva de Bezier).