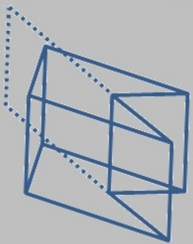
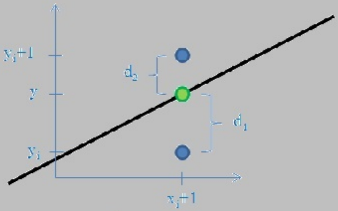
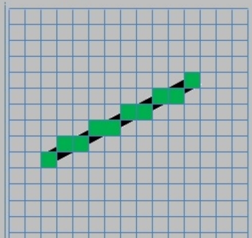






Universidad Autónoma del Estado de México
Centro Unviersitario UAEM Zumpango

Graficación
Unidad de Competencia III
Generación de objetos poligonales y poliédricos. Tema 1 y 2.

M. en C. Rafael Rojas Hernández

Septiembre 2017

Índice de la presentación

Información general

Unidad de Aprendizaje

Graficación

Propósito de la Unidad de Aprendizaje

El alumno establecerá los fundamentos y algoritmos básicos dentro del campo de la infografía y graficación, y elaborará herramientas o aplicaciones específicas como resolución a problemas del área de trabajo.

Estructura de la Unidad de Aprendizaje

1. Introducción a la Graficación e Infografía.
2. **Generación de objetos poligonales y poliédricos.**
3. Transformaciones elementales en 2D y 3D.
4. Modelado de objetos en 2D y 3D.
5. Visualización de objetos controlando las condiciones lumínicas.
6. Generación de imágenes escenarios virtuales y Animación.

Temario

Modelado poligonal y poliédrico

Dibujo de líneas

- Método básico
- Método incremental
- Método DDA
- Método de Bresenham o punto medio

Objetivo de la Unidad de Competencia

Describir y manipular los métodos básicos para construir objetos poligonales y poliédricos en la práctica.

Conocimientos

- Ilustra el modelado poligonal y poliédrico.
- Dibuja líneas a través de diversos métodos.
- Dibuja circunferencias a través de diversos métodos.
- Dibuja polígonos y poliedros.
- Emplea métodos de rellenado de áreas.
- Utiliza aproximaciones poligonales para el dibujo de curvas.
- Aplica operadores geométricos, topológicos y boléanos.

Habilidades

Diseña objetos elementales en perspectiva en la pantalla, soluciona problemas prácticos de dibujo de formas, y bosqueja objetos elementales.

Actitudes y valores

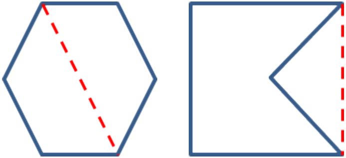
Autonomía, responsabilidad, respeto, tolerancia, puntualidad y trabajo. Selecciona métodos apropiados, explica conceptos, propone soluciones, resuelve problemas, pone en práctica los conocimientos adquiridos, actúa conforme a un plan.

Polígono

Un **polígono** en una figura formada por la reunión de tres o más segmentos de manera que no se crucen y solamente se toquen en los extremos y en donde ningún par de segmentos con un extremo común sean colineales.

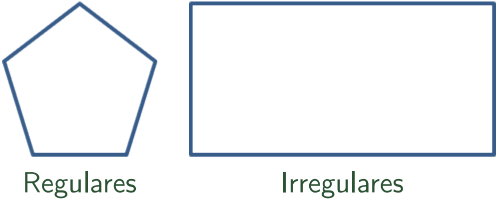
- Número de lados

- Número de ángulos



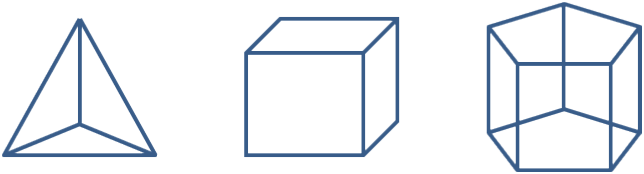
Convexos Cóncavos

- Relación entre lados y ángulos

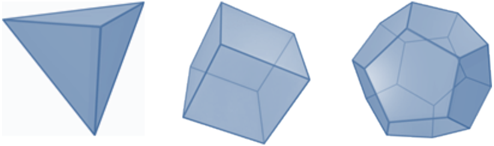


Poliedro

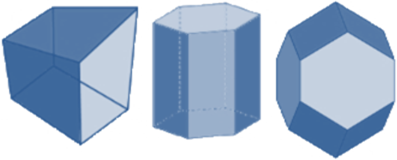
Un **poliédro** es un cuerpo geométrico limitado por cuatro o más regiones poligonales tales que cada uno de sus lados pertenece precisamente a dos regiones adyancentes no coplanares.



- Según sus caras



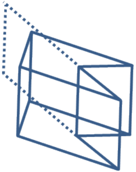
Regulares



Irregulares

- Según sus ángulos

Convexos



Cóncavos

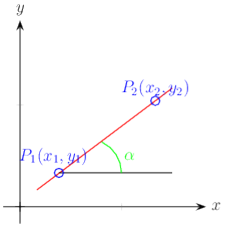
- Según sus caras

Línea recta

Analíticamente es una ecuación lineal o de primer grado en dos variables.
Representación gráfica del lugar geométrico cuya ecuación sea de primer grado con dos variables.

Una recta queda determinada si son conocidas al menos dos condiciones:

- Dos de sus puntos
- Un punto y su dirección



Ecuaciones de la línea recta

Punto-Pendiente

La ecuación de la recta que pasa por el punto $P(x_1, y_1)$ y cuya pendiente sea m es:

$$y - y_1 = m(x - x_1)$$

Pendiente-Ordenada al origen

La ecuación de la recta de pendiente m y que corta al eje y en el punto $P(0, b)$ es:

$$y = mx + b$$

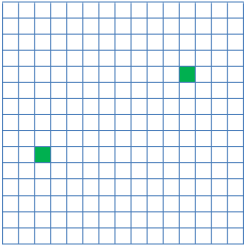
Ecuaciones de la línea recta

Cartesiana

La ecuación de la recta que pasa por los punto $P(x_1, y_1)$ y $Q(x_2, y_2)$ es:

$$\frac{y - y_1}{x - x_1} = \frac{y_1 - y_2}{x_1 - x_2}$$

Las ecuaciones anteriores funcionan cuando se tiene el caso de un plano continuo, sin embargo, para la gráficar una línea en el plano discreto surge el siguiente problema:



Para trazar una línea del punto $P_1(x_1, y_1)$ al punto $P_2(x_2, y_2)$, con $x_1 \leq x - 2$, la ecuacón de la recta estara dada por:

$$y_2 = m(x_2 - x_1) + y_1$$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Al obtener la pendiente, es posible trazar pequeñas línes incrementando a x_1 en 1 hasta llegar a x_2:

$$y = m(x - x_1) + y_1$$

Algoritmo básico

Algorithm 2.1: Básico

Data: Puntos $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$

Result: Línea

$m = \frac{y_2 - y_1}{x_2 - x_1}$;

for $x = x_1, \ldots, x_2$ **do**

  $y = m(x - x_1) + y_1$;

  putPixel(x, $\lfloor y + 0.5 \rfloor$);

  $x = x + 1$;

Si se considerá que una línea puede ser representa por un conjunto de líneas sucesivas, con la misma pendiente, es posible dividir el trazado en intervalos de x que se incrementan a partir de x_1 hasta x_2 para determinar a y, por lo que las ecuaciones las podemos reescribir como:

$$\begin{aligned} y_{i+1} &= m(x_{i+1} - x_i) + y_i \\ &= m((x_i + \Delta x) - x_i) + y_i \\ &= m \cdot \Delta x + y_i \end{aligned}$$

Si consideramos a $\Delta x = 1$ se tiene:

$$y_{i+1} = m + y_i$$

Dibujo de líneas
Método incremental

Algorithm 2.2: Incremental

Data: Puntos $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$
Result: Línea

$m = \frac{y_2 - y_1}{x_2 - x_1}$;
$y = y_1$;
for $x = x_1, \ldots, x_2$ **do**
    putPixel(x, $\lfloor y + 0.5 \rfloor$);
    $x = x + 1$;
    $y = y + m$;

El Analizador Diferencial Digital (*Digital Diferencial Analizer*) es un método que sirve para calcular las posiciones de los pixeles a lo largo de una línea recta considerando pasos unitarios de una coordenada y calculando el valor correspondiente de la otra; con la ecuación de la recta de la forma:

$$\Delta y = m \cdot \Delta x$$

Sin embargo debido a que se analiza los valores de la pendiente se tiene dos casos en partícular.

Para rectas con pendiente menor o igual a 1, se toman las variaciones en x como 1 y se calcula y:

$$y_{i+1} = y_i + m$$

Para rectas con pendiente positiva mayor a 1, se toman las variaciones en y como 1 y se calcula x:

$$x_{i+1} = y_i + \frac{1}{m}$$

Lo anterior se considera si $x_1 < x_2$, si se da el caso contrario se tiene que:

$$y_{i+1} = y_i - m$$

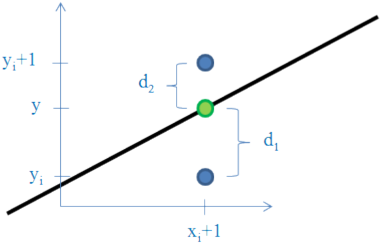
$$x_{i+1} = y_i - \frac{1}{m}$$

Algorithm 2.3: DDA

Data: Puntos $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$
Result: Línea

$dx = x_2 - x_1$;
$dy = y_2 - y_1$;
if $abs(dx) > abs(dy)$ **then**
 $paso = abs(dy)$;
else
 $paso = abs(dx)$;
$xInc = \frac{dx}{paso}$;
$yInc = \frac{dy}{paso}$;
$x = x_1$;
$y = y_1$;
putPixel($\lfloor x \lfloor$, $\lfloor y \rfloor$);
for $k = 1, \ldots, paso$ **do**
 $x = x + xInc$;
 $y = y + yInc$;
 putPixel($\lfloor x \lfloor$, $\lfloor y \rfloor$)

Se basa en hallar las coordenadas enteras más próximas a la trayectoría de una recta.



| La posición de y y las dos distancias d_1 y d_2 puede calcularse de la siguiente manera:

$$y = m(x_i + 1) + b$$

$$\begin{aligned} d_1 &= y - y_1 \\ &= m(x_i + 1) + b - y_i \end{aligned}$$

$$\begin{aligned} d_2 &= (y_i + 1) - y \\ &= y_i + 1 - m(x_i + 1) - b \end{aligned}$$

La diferencia entre las dos distancias será:

$$d_1 - d_2 = 2m(x_i + 1) - 2y_i + 2b - 1$$

Para trabajar la ecuación anterior con aritmética entera es necesario realizar la sustición de $m = \frac{\Delta y}{\Delta x}$, teniendo:

$$\begin{aligned} p_i &= \Delta x(d_1 - d_2) \\ &= 2\Delta y \cdot x_i - 2\Delta x \cdot y_i + c \end{aligned}$$

con $c = 2\Delta y + \Delta x(2b - 1)$
El parámetro p_i tiene un valor negativo si el píxel en la posición y_i es el más próximo. Para el caso de los intervalos sucesivos se tendrá:

$$p_{i+1} = 2\Delta y \cdot x_{i+1} - 2\Delta x \cdot y_{i+1} + c$$

Para obtener el paramétro sucesivo a partir del anterior se debe hacer:

$$p_{i+1} - p_i = 2\Delta y(x_{i+1} - x_i) - 2\Delta x(y_{i+1} - y_i)$$

Pero se tiene que $x_{i+1} = x_i + 1$, por lo que:

$$p_{i+1} = p_i + 2\Delta y - 2\Delta x(y_{i+1} - y_i)$$

Algorithm 2.4: Bresenham

Data: Puntos $P_1(x_1, y_1)$ y $P_2(x_2, y_2)$
Result: Línea

$dx = abs(x_1 - x_2)$;
$dx = abs(y_1 - y_2)$;
$p = 2dy - dx$;
$c1 = 2dy$;
$c2 = 2(dy - dx)$;
if $x_1 > x_2$ **then**
  $x = x_2$;
  $y = y_2$;
  $xFin = x_1$;
else
  $x = x_1$;
  $y = y_1$;
  $xFin = x_2$;
putPixel(x,y);

Algorithm 2.5: Bresenham

```
while x < xFin do
    x = x + 1;
    if p < 0 then
        p = p + c1;
    else
        y = y + 1;
        p = p + c2;
    putPixel(x,y);
m = (y2 - y1)/(x2 - x1);
y = y1;
for x = x1, ..., x2 do
    putpixel(x, ⌊y + 0.5⌋);
    x = x + 1;
    y = y + m;
```

Resumen

- Polígono
- Poliedro
- Línea Recta
- Método bÃ¡sico
- Método incremental
- Método DDA
- Método Bresenham

Bibliografía

- D. Marsh, Applied Geometry for Computer Graphics and CAD, Springer 2005.
- D. Salomon, Curves and Surfaces for Computer Graphics, Springer 2005.
- P. Comninos, Mathematical and Computer Programming Techniques for Computer Graphics, Springer, 2004.
- F. S. Hill, Computer Graphics, Prentice-Hall, 1990.
- E. Angel, Interactive Computer Graphics, Addison Wesley, 2000.
- D. Hearn & M. P. Baker, Gráficas por Computadora, prentice-Hall, 1988.