



**UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
INGENIERÍA DE SOFTWARE**

NRC: 14774

Materia: Computación Gráfica

Tema: Aplicaciones de Computación Gráfica

Grupo 2

Albán Richard

Escobar Aymé

Guacán Alexander

Macas Karol

Ing. César Javier Villacís Silva

1. Descripción del problema

Escriba un programa para dibujar un panel de abejas, como se muestra en la Figura 1. Para dibujar esta figura se debe leer el valor del lado de un hexágono como, por ejemplo: lado igual a 2. Se recomienda encerrar la figura dentro de un rectángulo y realizar los cálculos de los segmentos que encierran a la figura para poder graficarla.

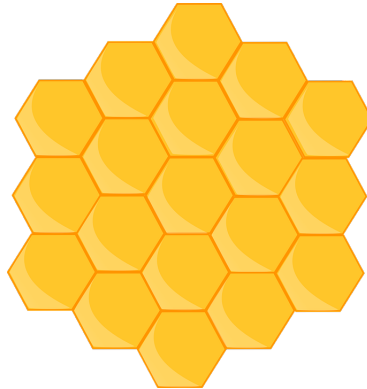


Figura 1: Panel de abejas (kclipartz, 2023).

2. Geometría de la figura

2.1. Hexágono

Para graficar el panel de abejas completo, primero debemos poder graficar un hexágono. Para ello debemos saber que un hexágono se construye usando triángulos equiláteros como puedes observar en la Figura 2, por lo que como dato principal sabemos que $\theta = 60^\circ$, además se ha encerrado al hexágono en un rectángulo, el cual representa el marco donde dibujaremos la figura, donde el centro de la misma, es el centro de este marco de dibujo.

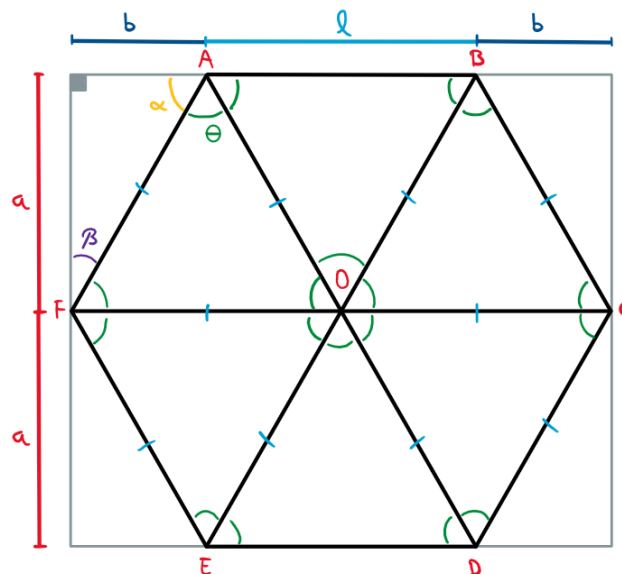


Figura 2: Vertices y lados del hexágono.

Calculamos los ángulos del triángulo rectángulo exterior al hexágono:

$$2\theta + \alpha = 180$$

$$\alpha = 180 - 2(30)$$

$$\alpha = 60$$

$$\alpha + \beta = 90$$

$$\beta = 90 - 60$$

$$\beta = 60$$

Utilizamos identidades trigonométricas en el triángulo externo (Figura 3) para obtener el valor de la apotema del hexágono (a) y de la distancia de la esquina superior izquierda del marco de dibujo a el vertice A .

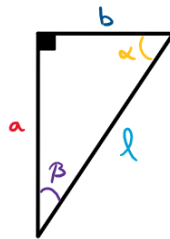


Figura 3: Triángulo externo al hexágono.

$$\sin(\alpha) = \frac{a}{l}$$

$$a = l * \sin 60$$

$$a = \frac{\sqrt{3}}{2}l \quad (1)$$

$$\sin(\beta) = \frac{b}{l}$$

$$b = l * \sin 30$$

$$b = \frac{l}{2} \quad (2)$$

Por lo tanto, si ubicamos cada uno de los vertices con respecto al vertice central O del hexágono, obtenemos:

$$O(x, y) = O(x_O, y_O)$$

$$A(x, y) = A(x_O - b, y_O - a)$$

$$B(x, y) = B(x_O + b, y_O - a)$$

$$C(x, y) = C(x_O + l, y_O) \quad (3)$$

$$D(x, y) = D(x_O + b, y_O + a)$$

$$E(x, y) = E(x_O - b, y_O + a)$$

$$F(x, y) = F(x_O - l, y_O)$$

Hay que aclarar que el punto $O(x_O, y_O)$ es un punto dinámico, esto ya que cada hexágono del panal de abejas tendrán el mismo tamaño, pero lo que cambia es el centro desde el cual se comienza a dibujar.

2.2. Panal de abejas

Una vez que aprendimos a graficar los hexágonos que conforman el panal de abejas, procederemos a determinar los centros de cada uno de ellos. Lo primero que haremos es dividir al panal por columnas, en cada una de ellas necesitaremos una cantidad diferente de hexagonos como se puede observar en la Figura 4.

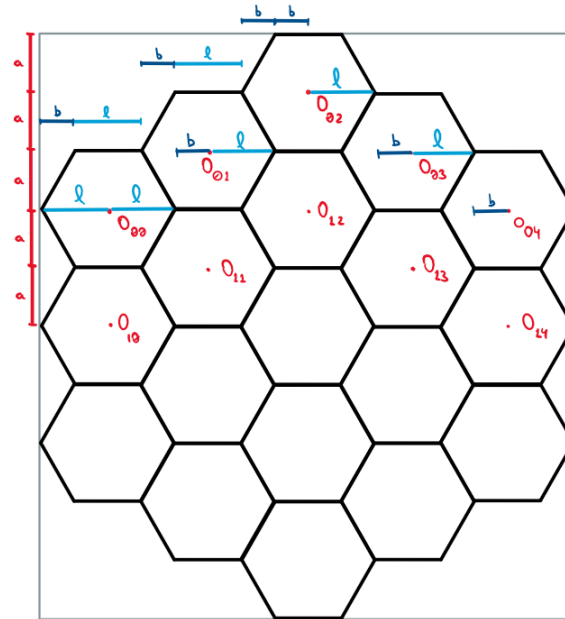


Figura 4: Panal de abejas con medidas.

Si tabulamos los hexágonos a dibujar por columna obtenemos que:

Columna (c)	Hexágonos (h)
0	3
1	4
2	5
3	4
4	3

Por lo tanto, la formula que nos permite determinar la cantidad de hexágonos a graficar por columna es:

$$h = -|c - 2| + 5 \quad (4)$$

Al tabular cada uno de los centros de los hexágonos del panal de abejas, podemos encontrar un patrón:

$O_{ij}(x, y)$	0	1	2	3
0	$(l, 3a)$	$(2l + b, 2a)$	$(3l + 2b, a)$	$(4l + 3b, 2a)$
1	$(l, 5a)$	$(2l + b, 4a)$	$(3l + 2b, 3a)$	$(4l + 3b, 4a)$
2	$(l, 7a)$	$(2l + b, 6a)$	$(3l + 2b, 5a)$	$(4l + 3b, 6a)$
n	$(l, (2i + 3)a)$	$(2l + b, (2i + 2)a)$	$(3l + 2b, (2i + 1)a)$	$(4l + 3b, (2i + 2)a)$

La formula para calcular el centro de cada uno de los hexágonos del panal de abejas si nos movemos por filas (i) y columnas (j) es:

$$O_{ij}(x, y) = O_{ij}((j + 1)l + jb, (2i + |j - 2| + 1)a) \quad (5)$$

3. Algoritmos

3.1. Algoritmo del método *ComputeApothem()*

1. Calcular valor del apotema utilizando la Ecuación (1).

3.2. Algoritmo del constructor *Hexagon()*

1. Asignar a la variable a el valor de la llamada a la función *ComputeApothem()*.
2. Calcular el valor de b del triángulo externo al hexágono, Ecuación (2).

3. Calcular las coordenadas del vértice A con respecto al vertice central O .
4. Calcular las coordenadas del vértice B con respecto al vertice central O .
5. Calcular las coordenadas del vértice C con respecto al vertice central O .
6. Calcular las coordenadas del vértice D con respecto al vertice central O .
7. Calcular las coordenadas del vértice E con respecto al vertice central O .
8. Calcular las coordenadas del vértice F con respecto al vertice central O .

3.3. Algoritmo del método *Plot()* de la clase *Hexagon*

1. Asignar al objeto *illustrator* la funcionalidad de crear gráficos del *PictureBox* llamado *canvas*.
2. Crear un bolígrafo de color negro (Black).
3. Crear una brocha solida de color amarillo (Yellow).
4. Crear un arreglo de puntos con los vértices del hexágono: A, B, C, D, E, F, A.
5. Graficar un polígono relleno con la brocha y el arreglo de puntos del paso anterior.
6. Graficar un polígono con el bolígrafo y el arreglo de puntos del paso anterior.

3.4. Algoritmo del constructor *BeePanel()*

1. Calcular el centro de cada hexágono que conforma el panal de abejas, utilizando la Ecuación (5) y (4).
2. Crear un nuevo hexagono por cada centro calculado y asignarlos a un arreglo de hexágonos llamado *Panels*.

3.5. Algoritmo del método *Plot()* de la clase *BeePanel*

1. Llamado al método *Plot()* de cada hexágono almacenado en el arreglo *Panels*.

4. Código de la aplicación

A continuación se muestra los métodos de cada una de las clases principales necesarias para resolver el problema.

4.1. Clase *Hexagon*

```
1 public static float ComputeApothem(float side) {  
2     return (float)Math.Sqrt(3) * side / 2.0f;  
3 }
```

Figura 5: Calcular apotema del hexágono.

```
1 public Hexagon(PointF center, float side) {
2     Side = side;
3     Center = center;
4
5     a = ComputeApothem(Side);
6     b = Side / 2.0f;
7
8     Vertices = new Dictionary<char, PointF>
9     {
10         { 'A', new PointF((Center.X - b) * ScalarFactor, (Center.Y - a) * ScalarFactor) },
11         { 'B', new PointF((Center.X + b) * ScalarFactor, (Center.Y - a) * ScalarFactor) },
12         { 'C', new PointF((Center.X + Side) * ScalarFactor, Center.Y * ScalarFactor) },
13         { 'D', new PointF((Center.X + b) * ScalarFactor, (Center.Y + a) * ScalarFactor) },
14         { 'E', new PointF((Center.X - b) * ScalarFactor, (Center.Y + a) * ScalarFactor) },
15         { 'F', new PointF((Center.X - Side) * ScalarFactor, Center.Y * ScalarFactor) }
16     };
17 }
```

Figura 6: Constructor de la clase *Hexagon*.

```
1 public void Plot(PictureBox canvas, Color outline, Color fill) {
2     _illustrator = canvas.CreateGraphics();
3     _pen = new Pen(outline);
4     _brush = new SolidBrush(fill);
5
6     PointF[] vertices = Vertices.Values.ToArray();
7     _ = vertices.Append(vertices[0]);
8
9     _illustrator.FillPolygon(_brush, vertices);
10    _illustrator.DrawPolygon(_pen, vertices);
11 }
```

Figura 7: Graficar hexágono.

4.2. Clase *BeePanel*

```
1 private int ComputeRows(int currentColumn) {
2     return -Math.Abs(currentColumn - Columns / 2) + Columns;
3 }
```

Figura 8: Calcular hexágonos por columna.

```
1 public BeePanel(float hexagonSide) {
2     for (int j = 0; j < Columns; ++j) {
3         for (int i = 0; i < ComputeRows(j); ++i) {
4             PointF hexagonCenter = new() {
5                 X = ((j + 1) * hexagonSide) + (j * hexagonSide / 2),
6                 Y = (2 * i + Math.Abs(j - Columns / 2) + 1) * Hexagon.ComputeApothem(hexagonSide)
7             };
8
9             Panels.Add(new Hexagon(hexagonCenter, hexagonSide));
10        }
11    }
12 }
```

Figura 9: Constructor de la clase *BeePanel*.

```
1 public void Plot(PictureBox canvas, Color outline, Color fill) {
2     foreach (var panel in Panels) {
3         panel.Plot(canvas, outline, fill);
4     }
5 }
```

Figura 10: Graficar panel de abejas.

4.3. Clase *Form1*

```
1 private void buttonPlotBeePanel_Click(object sender, EventArgs e) {
2     pictureBoxCanvas.Refresh();
3
4     try {
5         float hexagonSide = float.Parse(textBoxSideHexagon.Text);
6         BeePanel = new BeePanel(hexagonSide);
7
8         BeePanel.Plot(pictureBoxCanvas, Color.Black, Color.Yellow);
9     } catch (Exception) {
10        MessageBox.Show("El valor ingresado no es valido");
11    }
12
13 }
```

Figura 11: Función del botón “Graficar”.

5. Corrida del programa

En la Figura 12 se muestra un ejemplo de la ejecución del programa.

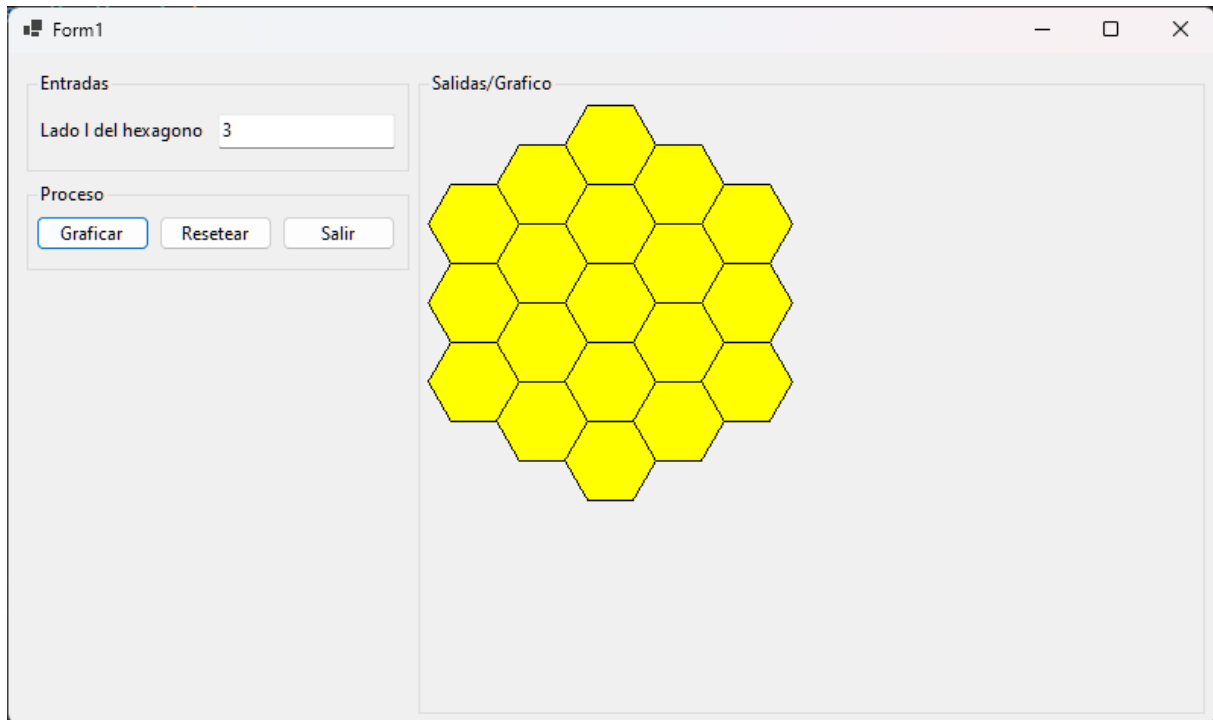


Figura 12: Dibujado del panal de abejas.

Referencias

klipartz. (2023). <https://www.klipartz.com/es/sticker-png-tigsh>