Apellido y nombre: [redacted]

4 HOJAS + ENUNCIADO (2)

COMPUTACIÓN GRÁFICA
Primer Parcial- 28 de abril de 2010

Resuelva cada problema en una hoja distinta. Escriba en cada hoja a entregar su nombre y apellido.

1. Implemente en C# , creando solo dos instancias de arco (círculo y semicírculo) y usando transformaciones, el método ***dibujar()*** de la clase **Figura**, para que se vea en la pantalla la Figura1. Dispone para ello de la clase **Arco**, con su ***constructor*** y su método ***dibujar()***. Los parámetros del constructor de la clase Arco son los mostrados en la Figura2.

Figura 1

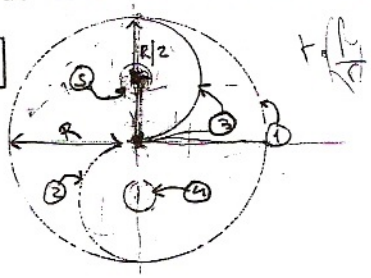
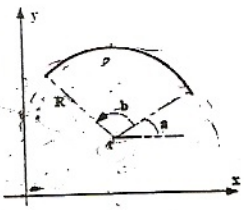
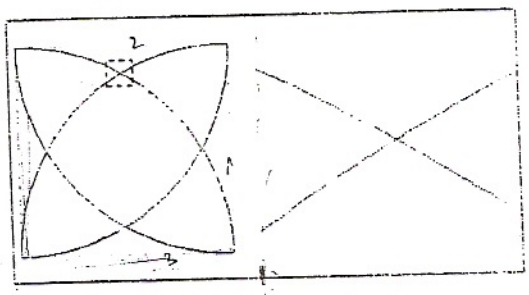


Figura 2



2. Implemente en C# el procedimiento ***setupViewport(...)*** , de modo que la Figura3 se vea completa y sin deformarse, aún cambiando el tamaño de la ventana de pantalla. Las llamadas a este procedimiento pueden verse en el código adjunto.

Figura 3



3. Indique cuáles son los tres parámetros de gluUnProject marcados en el código adjunto. Muestre sobre el mismo código dónde y cómo los obtiene.

4. La siguiente tabla representa la estructura de caras del modelo de la Figura 4. Indique cómo modificaría las normales, donde sea necesario para obtener, una vez sombreado e iluminado, el modelo de la figura 5. Escriba la/s tabla/s resultante/s.

	CARAS											
	Ind_vértices						Ind_normales					
C0	0	1	7	6			0	0	0	0		
C1	1	2	8	7			1	1	1	1		
C2	2	3	9	8			2	2	2	2		
C3	3	4	10	9			3	3	3	3		
C4	4	5	11	10			4	4	4	4		
C5	5	0	6	11			5	5	5	5		
C6	0	5	4	3	2	1	6	6	6	6	6	6
C7	6	7	8	9	10	11	7	7	7	7	7	7

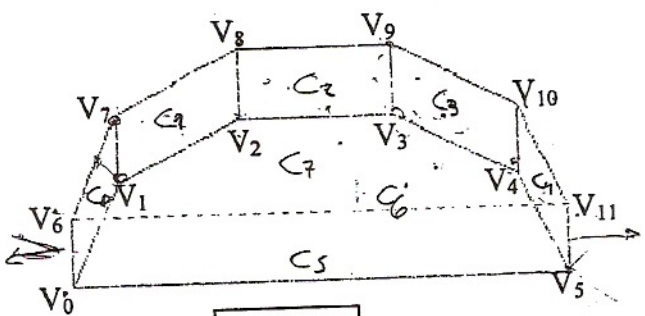


Figura 4

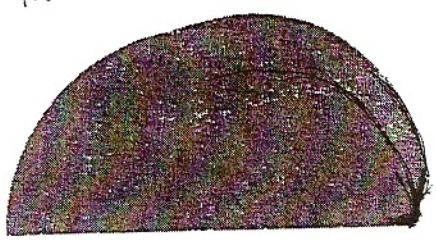
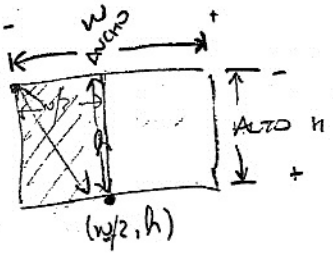


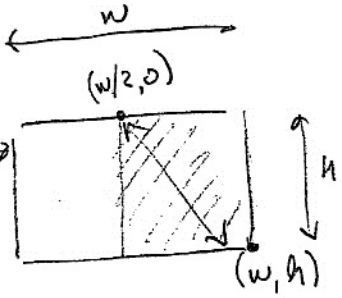
Figura 5

Apellido y nombre: [redacted]

COMPUTACIÓN GRÁFICA
Primer Parcial- 28 de abril de 2010

```
private void glControl1_Paint(object sender, PaintEventArgs e)
        {
            GL.Clear(ClearBufferMask.ColorBufferBit |
            ClearBufferMask.DepthBufferBit);

            miFigura = new Figura(origenX, origenY, ang1, ang2, radio);

            setupVentana( 0, 0 , ancho/2(todo!

            int w = glControl1.Width;
            int h = glControl1.Height;
            setupViewport(0, w/2, 0, h);
                          Xo, ANCHO, Yo, ALTO
            GL.MatrixMode(MatrixMode.Modelview);
            GL.LoadIdentity();
            miFigura.dibujar();

            if (zoom)
            {
             areaZoom = new Cuadrado(  ,   ,   ,  );
            GL.Color3(0, 0, 0.7);
            GL.Enable(EnableCap.LineStipple);
            GL.LineStipple(5, 0x5555);

            areaZoom.dibujar();
            GL.Disable(EnableCap.LineStipple);

            setupVentana(  ,   ,   ,   );

            setupViewport(w / 2, w, 0, h);
            GL.MatrixMode(MatrixMode.Modelview);
            GL.LoadIdentity();
            miFigura.dibujar();

            }

            glControl1.SwapBuffers();
        }

        private void glControl1_MouseDown(object sender, MouseEventArgs e)
        {
            double zoomX;
            double zoomY;
            double zoomZ;

        Glu.gluUnProject(x, height - y, ___, ___, ___, out zoomX, out zoomY, out zoomZ);
```





PROJECTION MATRIX

VIEWPORT MATRIX

MODELVIEW MATRIX