Documentación técnica

Alumno: Sánchez Hernández Max Armando

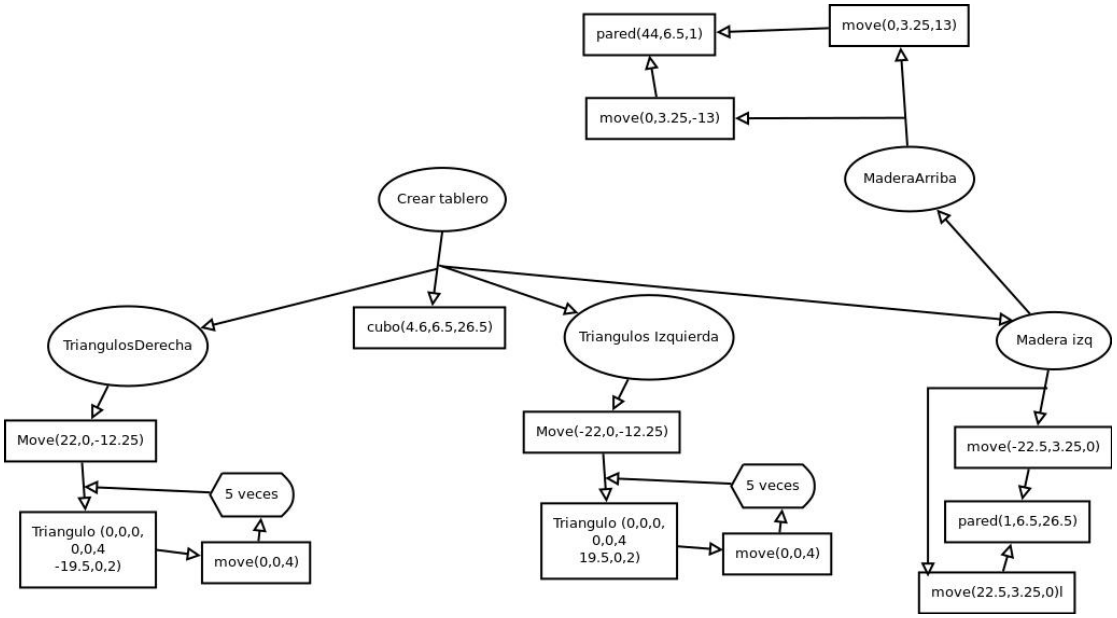
Proyecto Computación gráfica

Punto 1

Modelo Jerárquico de la caja de Backgammon, dicha caja tiene dos bisagras que unen las dos partes de la caja; en la parte interna de la caja está la zona del tablero de juego, dicho tablero consta de una textura de fondo que represente el paño verde o la madera, sobre dicha textura de fondo debe de haber planos con materiales que representen los triángulos alargados de dos colores intercalados (3 puntos: 1 punto por la caja que se puede abrir y cerrar, 1 punto por la textura de fondo, 1 punto por los planos triangulares)

Resolución

Esto se resolvio usando un arbol jerarquico de como se debian ir construyendo los elementos,



Codigo

```
void tableroCompleto(float anguloActual,GLuint bisagra,GLuint tapete,GLuint oro,G
    tablero(madera,tapete,triangulos,posiciones,24,textBlancas,textNegras);
    // aqui ponemos las bisagras
    glPushMatrix();
        glPushMatrix();
            glPushMatrix();
                glTranslatef(10.5,0.5,-13.01);
                glRotatef(-90,1,0,0);
                bisagra1(5,2,bisagra);
            glPopMatrix();
            glPushMatrix();
                glTranslatef(-10.5,0.5,-13.01);
                glRotatef(-90,1,0,0);
                bisagra1(5,2,bisagra);
            glPopMatrix();
        glPopMatrix();
        glTranslatef(0,1.5,-13.25);
        glRotatef(anguloActual,1,0,0);
        glTranslatef(0,-1.5,-13.25);
        tablero(madera,tapete,triangulos,posiciones,12,textBlancas,textNegras);
        glPushMatrix();
            glTranslatef(10.5,0.5,13.01);
            glPushMatrix();
                glRotatef(-90,1,0,0);
                bisagra1(5,2,bisagra);
            glPopMatrix();
            glTranslatef(0,1,0.25);
            glRotatef(90,0,0,1);
            cilindroBisagra(20,0.25,5,oro,bisagra);
        glPopMatrix();
        glPushMatrix();
            glTranslatef(-10.5,0.5,13.01);
            glPushMatrix();
                glRotatef(-90,1,0,0);
                bisagra1(5,2,bisagra);
            glPopMatrix();
            glTranslatef(0,1,0.25);
            glRotatef(90,0,0,1);
            cilindroBisagra(20,0.25,5,oro,bisagra);
        glPopMatrix();
    glPopMatrix();

}
```

Punto 2

Se debe de contar con 8 fichas de juego (4 fichas de cada jugador y se visualizan con materiales diferentes para los colores, dichas fichas deben de estar acomodadas en las 4 esquinas del tablero (2 puntos: 1 punto por las fichas y 1 punto por el acomodo de las fichas)

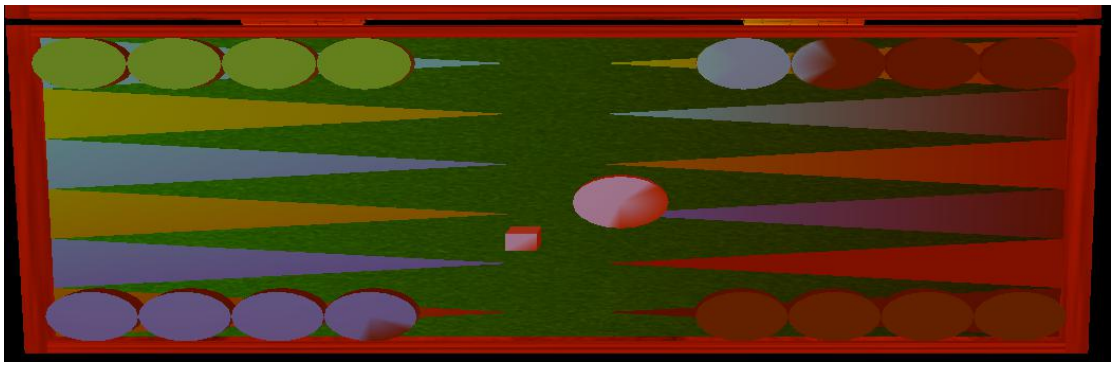
Resolucion: Se creo un arreglo que pudiera indicar la posición de las fichas en el tablero, las puntas estaban enumeradas lógicamente,de manera que con recorrer el arreglo se indicará la posición de la ficha

Codigo

```
248     // aqui se incluyen las fichas de tablero
249     glPushMatrix();
250         glTranslatef(-22,1.27,-12);
251         glTranslatef(2,0,2);
252         for (int i=limite-12,signo=1;i<limite;i++){
253             if (i==6 or i==18){
254                 glPopMatrix();
255                 glPushMatrix();
256                 glTranslatef(22,1.27,-12);
257                 glTranslatef(-2,0,2);
258                 signo=-1;
259             }
260             glPushMatrix();
261             for (int j=0;j<5;j++){
262                 if (posiciones[i][j]==1)
263                     ficha(matFiha1Am);
264                 else if (posiciones[i][j]==2)
265                     ficha(matFiha2Am);
266                 glTranslatef(4*signo,0,0);
267             }
268             glPopMatrix();
269             glTranslatef(0,0,4);
270         }
271     glPopMatrix();
```

```
125  int posiciones[24][5]={
126
127  //tablero 2
128  {0,0,0,0,0}
129  ,{0,0,0,0,0}
130  ,{0,0,0,0,0}
131  ,{0,0,0,0,0}
132  ,{0,0,0,0,0}
133  ,{0,0,0,0,0}
134  //lado 2 tablero 2
135  ,{0,0,0,0,0}
136  ,{0,0,0,0,0}
137  ,{0,0,0,0,0}
138  ,{0,0,0,0,0}
139  ,{0,0,0,0,0}
140  ,{0,0,0,0,0}
141  ,{1,1,1,1,0}
142  ,{0,0,0,0,0}
143  ,{0,0,0,0,0}
144  ,{0,0,0,0,0}
145  ,{0,0,0,0,0}
146  ,{2,2,2,2,0}
147  //el otro lado tablero 1
148  ,{2,2,2,2,0}
149  ,{0,0,0,0,0}
150  ,{0,0,0,0,0}
151  ,{0,0,0,0,0}
152  ,{0,0,0,0,0}
153  ,{1,1,1,1,0}};
```

Captura de pantalla



Punto 3

Debe de haber un dado de 6 caras texturizado el cual se animará con animación continua de tipo glutIdleFunc (no es keyframes) (2 puntos: 1 punto por el dado texturizado correctamente, 1 punto por la animación contínua)

Resolución: el dado se crea con GL_POLYGON, y se texturiza con dice.tga también se le aplica propiedades de un material para que interactue con la luz, se incluyen rotaciones en el eje x y en el eje z y se genera un numero aleatorio entre 1 y 4 que se multiplica por 90 para asegurar que siempre quede una cara mirando hacia arriba, así entonces se usa una funcion de interpolación que calcula cual debe ser el ángulo siguiente para que en 10 pasos se llegue al ángulo final.

Código

```
void cubo(GLuint textura,float x, float z){
    float a=1; //arista
    //rotacion aleatoria en x y en z
    GLfloat matCuboAm[]={0.5,0.5,0.5,1};
    GLfloat matShin[]={50.0};
    // glMaterialfv(GL_FRONT_AND_BACK,GL_AMBIENT,matCuboAm);
    glMaterialfv(GL_FRONT_AND_BACK,GL_DIFFUSE,matCuboAm);
    glMaterialfv(GL_FRONT_AND_BACK,GL_SPECULAR,matCuboAm);
    glMaterialfv(GL_FRONT,GL_SHININESS,matShin);

    glPushMatrix();
        glRotatef(x,1,0,0);
        glRotatef(z,0,0,1);
        glBindTexture(GL_TEXTURE_2D, textura);
        glBegin(GL_POLYGON);
                //cara trasera con 2
                glNormal3f(0,0,-1);
                // producto(resta(-a,-a,-a,-a,a,-a),resta(-a,a,-a,a,a,-a));
                glTexCoord2f(0.33,0.5); glVertex3f(-a,-a,-a);
                glTexCoord2f(0.66,0.5); glVertex3f(-a,a,-a);
                glTexCoord2f(0.66,1); glVertex3f(a,a,-a);
                glTexCoord2f(0.33,1); glVertex3f(a,-a,-a);
        glEnd();
        glBegin(GL_POLYGON);
                //cara frontral con 5
                glNormal3f(0,0,1);
                // producto(resta(-a,-a,a,-a,a,a),resta(-a,a,a,a,a,a));
                glTexCoord2f(0.33,0); glVertex3f(-a,-a,a);
                glTexCoord2f(0.33,0.5); glVertex3f(-a,a,a);
                glTexCoord2f(0.66,0.5); glVertex3f(a,a,a);
                glTexCoord2f(0.66,0); glVertex3f(a,-a,a);
            glEnd();
            glBegin(GL_POLYGON);
                //cada inferior con 4
                glNormal3f(0,-1,0);
                // producto(resta(-a,-a,-a,a,-a,-a),resta(a,-a,-a,a,-a,a));
                glTexCoord2f(0,0); glVertex3f(-a,-a,-a);
                glTexCoord2f(0,0.5); glVertex3f(a,-a,-a);
                glTexCoord2f(0.33,0.5); glVertex3f(a,-a,a);
                glTexCoord2f(0.33,0); glVertex3f(-a,-a,a);
            glEnd();
            glBegin(GL_POLYGON);
                //cara superior con 3
                glNormal3f(0,1,0);
                // producto(resta(-a,a,-a,a,a,-a),resta(a,a,-a,a,a,a));
                glTexCoord2f(0.66,0.5); glVertex3f(-a,a,-a);
```

```
    if (tiro){
        if (rAx!=rotacionAleatoriax ||rAz!=rotacionAleatoriaz  ){
            rAx+=interpolacionCubo(rAxAux,rotacionAleatoriax,10);
            if (rAx>360)
                rAx-=360;
            rAz+=interpolacionCubo(rAzAux,rotacionAleatoriaz,10);
            if (rAz>360)
                rAz-=360;
        }
        if (rAx2!=rotacionAleatoriax2 || rAz2!=rotacionAleatoriaz2  ){
            rAx2+=interpolacionCubo(rAxAux2,rotacionAleatoriax2,10);
            if (rAx2>360)
                rAx2-=360;
            rAz2+=interpolacionCubo(rAzAux2,rotacionAleatoriaz2,10);
            if (rAz2>360)
                rAz2-=360;
        }
        if ((rAx==rotacionAleatoriax  &&rAx2==rotacionAleatoriax2 )||(rAz==rotacionA
            tiro=false;
            rAxAux=rAx;
            rAzAux=rAz;
            rAxAux2=rAx2;
            rAzAux2=rAz2;
        }
    }
```

Punto 4

El tablero debe de poder abrir y cerrarse con animación de Keyframes, al cerrar el tablero, las fichas se quedarán dentro del tablero (2 puntos: 1 punto por la animación por keyframes, 1 punto por que las fichas se queden dentro de la caja posicionadas al abrir y cerrar)

Resolución:Se uso una estructura para almacenar los estados del giro con

respecto a al bisagra, se uso también una función de interpolación para alcanzar dichos estados.

```
float interpolacion(float actual,float destino){
    float aumento=0.5;
    return ((actual-destino)/abs(actual-destino))*aumento;
}

float interpolacionCubo(float origen,float destino,float tiempo){
    return (destino-origen)/tiempo;
}

void animacion()
{
    if (play){
        if (abrir){
            if (anguloActual<keyFrame[0].angulo)
                indiceK--;
                if (indiceK==0){
                    abrir=false;
                    play=false;
                }
            else
                anguloActual-=interpolacion(keyFrame[indiceK].angulo,keyFrame[indiceK-1]
        }
        else if (cerrar){
            if (anguloActual>=keyFrame[1].angulo)
                indiceK++;
                if (indiceK==totalK-1){
                    play=false;
                    cerrar=false;
                }
            else
                anguloActual-=interpolacion(keyFrame[indiceK].angulo,keyFrame[indiceK+1]
        }
    }
```

Punto 5

Debe de haber tres luces, una luz que ilumine todo el tablero, una luz posicionada en una esquina de la caja que ilumine sólo a las fichas que están en dicha esquina y una luz que se pueda desplazar por todo el tablero, el color de las 3 luces debe de ser diferente y ninguna puede ser totalmente blanca (3 puntos: 1 punto por cada luz y la interacción de dicha luz con los objetos con materiales)

Resolución: se crean tres luces una direccional(roja) y dos posicionales, se asignan variables para cambiar la posición de una y la otra se pone sobre una esquina

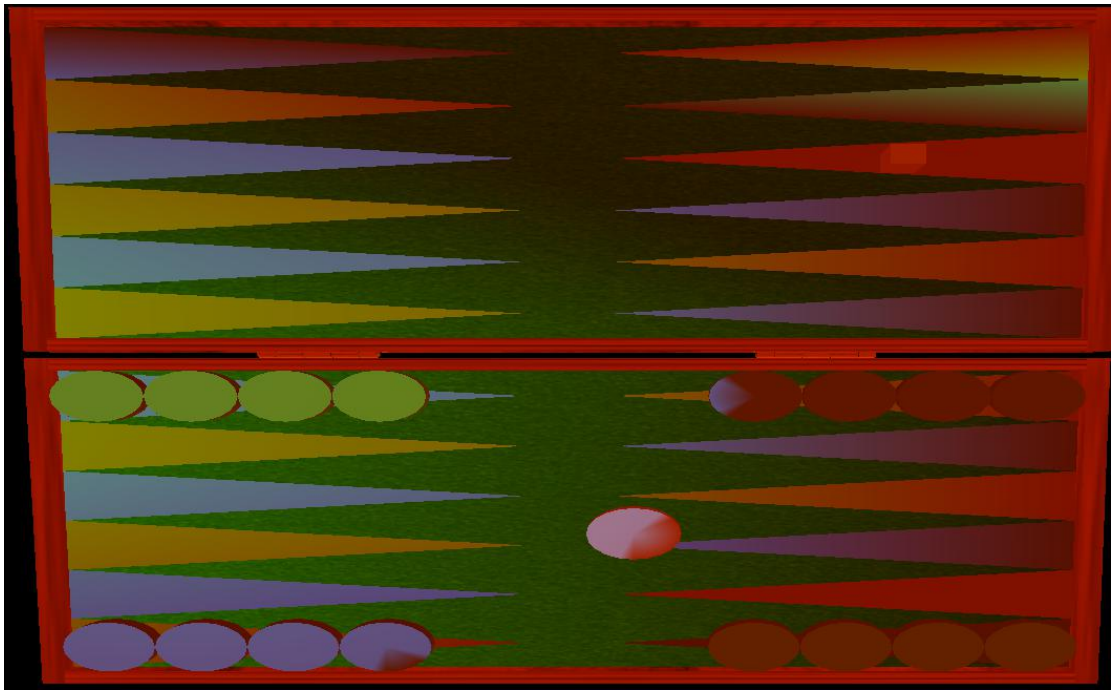
Codigo

```
        glRotatef(g_lookupdown,1.0f,0,0);
        light_position2[0] = ax;
        light_position2[1] = ay;
        light_position2[2] = az;
        glLightfv(GL_LIGHT0, GL_POSITION, light_position);
        glLightfv(GL_LIGHT0,GL_AMBIENT,ambLiDir);
        glLightfv(GL_LIGHT0,GL_DIFFUSE,difLiDir);
        glLightfv(GL_LIGHT0,GL_SPECULAR,espLiDir);

        glLightfv(GL_LIGHT1, GL_AMBIENT, light1_ambient);
        glLightfv(GL_LIGHT1, GL_DIFFUSE, light1_diffuse);
        glLightfv(GL_LIGHT1, GL_SPECULAR, light1_specular);
        glLightfv(GL_LIGHT1, GL_POSITION, light1_position);
        glLightf(GL_LIGHT1, GL_SPOT_CUTOFF, 22.0);
        glLightfv(GL_LIGHT1, GL_SPOT_DIRECTION, spot_direction);

        glLightfv(GL_LIGHT2, GL_AMBIENT, light_ambient2);
        glLightfv(GL_LIGHT2, GL_DIFFUSE, light_diffuse2);
        glLightfv(GL_LIGHT2, GL_SPECULAR, light_specular2);
        glLightfv(GL_LIGHT2, GL_POSITION, light_position2);
        glLightf(GL_LIGHT2, GL_SPOT_CUTOFF, 10.0);
        glLightfv(GL_LIGHT2, GL_SPOT_DIRECTION, spot_direction);
```

Captura de pantalla



Punto 6

Debe de haber cambio de cámaras: Una cámara enseña todo el tablero desde la perspectiva del jugador y la otra cámara enseña desde la perspectiva de alguna ficha, al tener la cámara en la posición de la ficha la cámara se puede mover, al hacer cambio entre cámara de vista de todo el tablero y cámara ligad a la ficha debe de respetarse la posición en donde se dejó posicionada la ficha, no siempre regresar a un estado inicial (1 punto por la implementación de las dos cámaras y el correcto cambio de vista entre las dos, 1 punto adicional si la cámara de la ficha se mueve con mouse en lugar de con teclado)

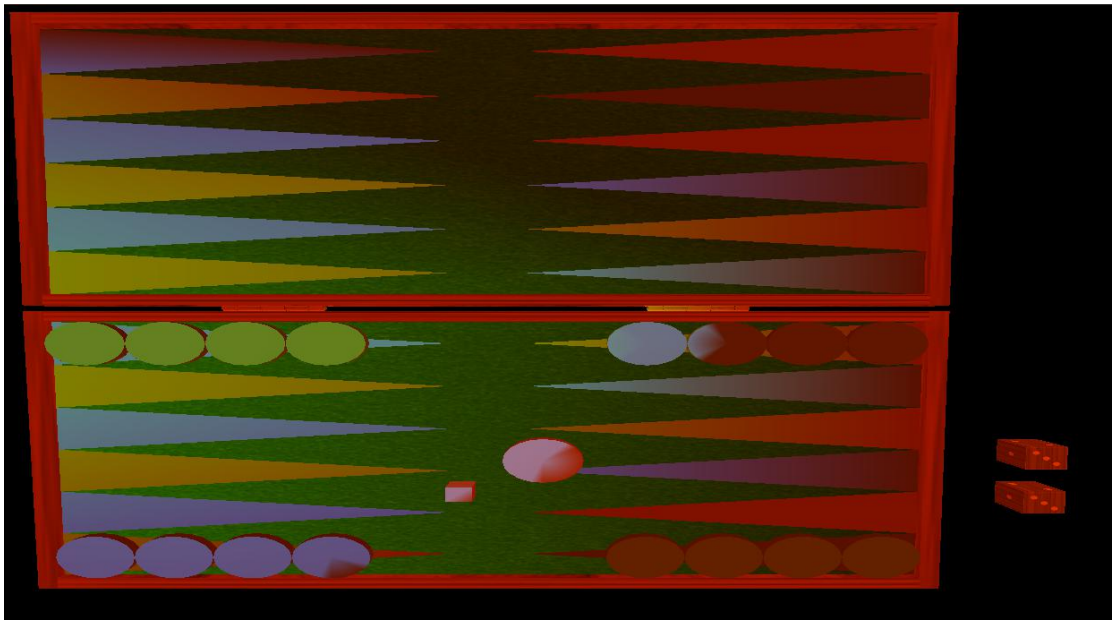
Resolución: se obtuvieron las coordenadas que la cámara deberá tener para poder ver todo el escenario, después con la tecla l(ele) se cambia entre la perspectiva de una ficha y la total.

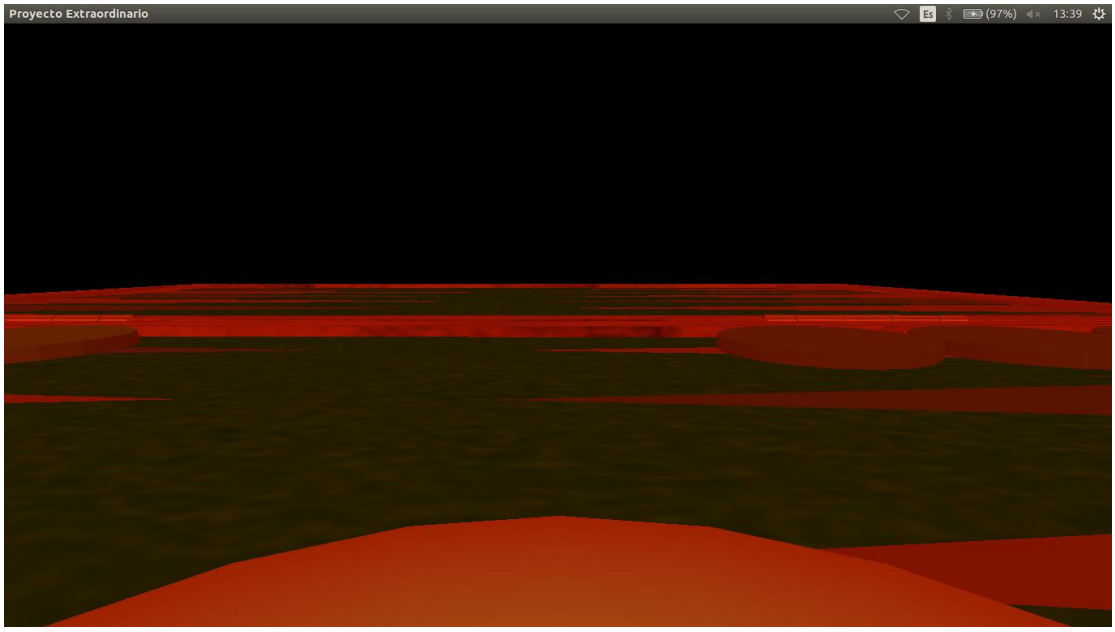
Código:

```
if (!changeCamera)
    objCamera.Position_Camera(4.229999, 31.900013, -11.579988,
      4.229999, 31.900013, -14.579986,
      0.000000, 1.000000, 0.000000);
else{

    if (fichaDryAnterior<fichaDry)
        objCamera.Position_Camera(fichaDummyx,fichaDummyy,fichaDummyz,
            fichaDviewx,fichaDummyy,fichaDviewz,0,1,0,-CAMERASPEED);
    else if (fichaDryAnterior>fichaDry)
        objCamera.Position_Camera(fichaDummyx,fichaDummyy,fichaDummyz,
            fichaDviewx,fichaDummyy,fichaDviewz,0,1,0,CAMERASPEED);
    else
        objCamera.Position_Camera(fichaDummyx,fichaDummyy,fichaDummyz,
            fichaDviewx,fichaDummyy,fichaDviewz,0,1,0,0);
    fichaDryAnterior=fichaDry;
}

gluLookAt(  objCamera.mPos.x,  objCamera.mPos.y,  objCamera.mPos.z,
        objCamera.mView.x, objCamera.mView.y, objCamera.mView.z,
        objCamera.mUp.x,   objCamera.mUp.y,   objCamera.mUp.z);
```




Proyecto Extraordinario
Es
(97%)
13:39