

respuestasexamenteoria210212grup...



PruebaAlien



Informática Gráfica



3º Grado en Ingeniería Informática

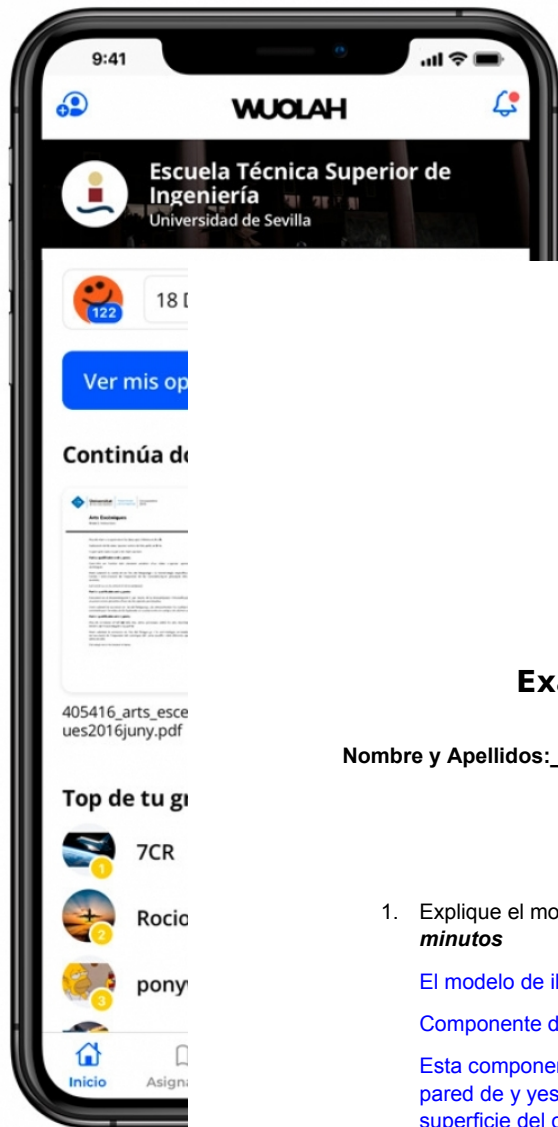


Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.





Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



Examen Informática Gráfica (12/02/21)

Nombre y Apellidos: _____ Grupo B

Teoría

(Puntuación expresada sobre 10. Mínimo 3,5 para que cuente)

1. Explique el modelo de iluminación que se usa en OpenGL. (2 pts) **Tiempo máximo: 15 minutos**

El modelo de iluminación de OpenGL tiene tres componentes:

Componente difusa:

Esta componente modela la reflexión de objetos que no son brillantes, como por ejemplo una pared de yeso o la superficie de un árbol. La reflexión depende de la orientación del objeto o superficie del objeto, por ejemplo un triángulo, y de la posición de la fuente de luz. La orientación del objeto se modela con el vector normal. La posición de la luz se modela con el vector entre la posición de la luz y un punto del objeto. La reflexión depende del ángulo que forman ambos vectores. Si ambos vectores están normalizados, el ángulo se calcula con el producto escalar. Además hay que tener en cuenta una constante de reflexividad difusa del objeto, y la componente difusa de la luz

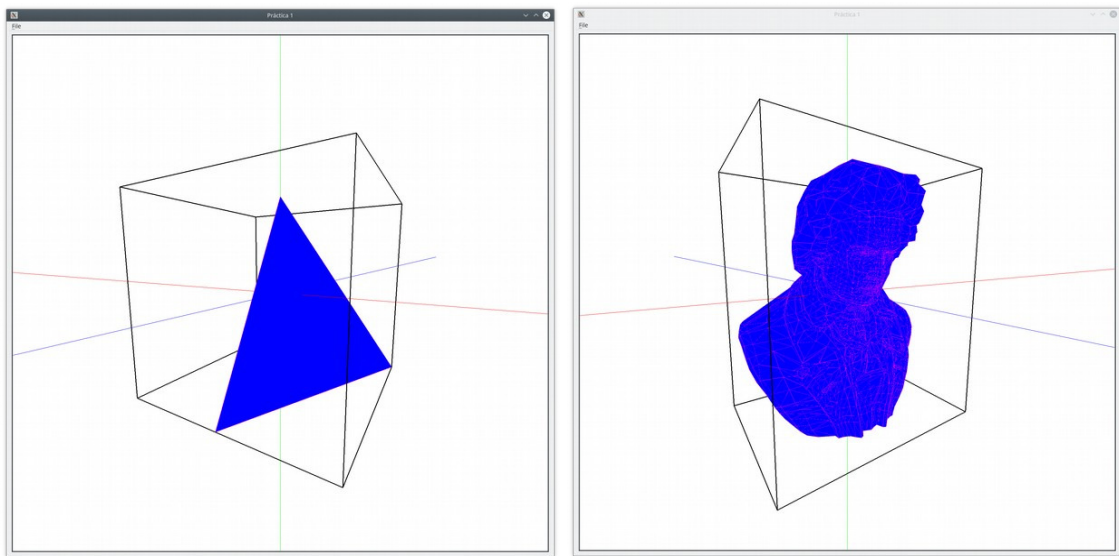
Componente especular:

Esta componente modela la reflexión de objetos que son brillantes como por ejemplo espejo. La reflexión depende de la orientación del objeto o superficie del objeto, por ejemplo un triángulo, y de la posición de la fuente de luz. También de la posición del observador. La orientación del objeto se modela con el vector normal. La posición de la luz se modela con el vector entre la posición de la luz y un punto del objeto. Dado el rayo de luz que incide con un ángulo Alfa con respecto a la normal, se produce un rayo reflejado con el mismo ángulo Alfa de salida. Además tenemos el vector que se forma entre la posición del observador y el punto de incidencia del rayo. La reflexión depende del ángulo que forman el vector del rayo reflejado y el vector del observador. Si ambos vectores están normalizados, el ángulo se calcula con el producto escalar. Además, para modelar la variación de la reflexión especular, se exponencia el producto escalar. Además hay que tener en cuenta una constante de reflexividad especular del objeto, y la componente especular de la luz

Componente ambiental

Esta componente modela la inter reflexión de la luz en los múltiples objetos de la escena. Esto implica que la esta luz reflejada viene de todas direcciones. Se crea para hacer que la parte de los objetos a los que no le llega la luz directamente no aparezca de color negro. La reflexión depende de una constante reflexividad ambiental del objeto, y la componente ambiental de la luz

2. Dado un objeto 3D que se define con definido por sus vértices, `vector<_vertex3f> Vertices`, y triángulos, `vector<_vertex3ui> Triangles`, implemente mediante pseudocódigo o C++ el programa que calcularía la caja frontera (la caja frontera es el menor paralelepípedo que incluye a la figura; basta con salvar el vértice con los coordenadas menores y el vértice con las mayores coordenadas) **(2 pts) Tiempo máximo: 15 minutos**



```
Pos_min=_vertex3f(1e8,1e8,1e8);
Pos_max=_vertex3f(-1e8,-1e8,-1e8);

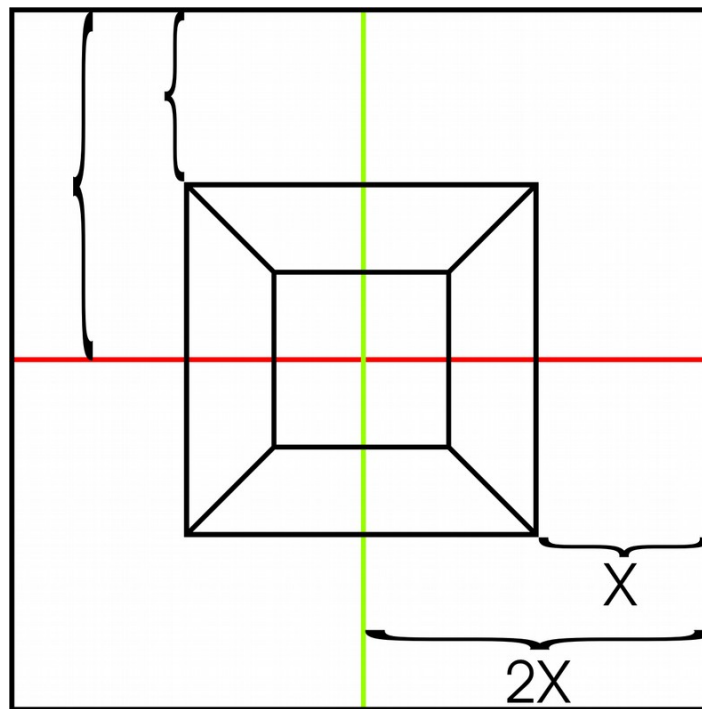
for (unsigned int i=0;i<Vertices.size();i++){
    if (Vertices[i].x<Pos_min.x) Pos_min.x=Vertices[i].x;
    if (Vertices[i].y<Pos_min.y) Pos_min.y=Vertices[i].y;
    if (Vertices[i].z<Pos_min.z) Pos_min.z=Vertices[i].z;

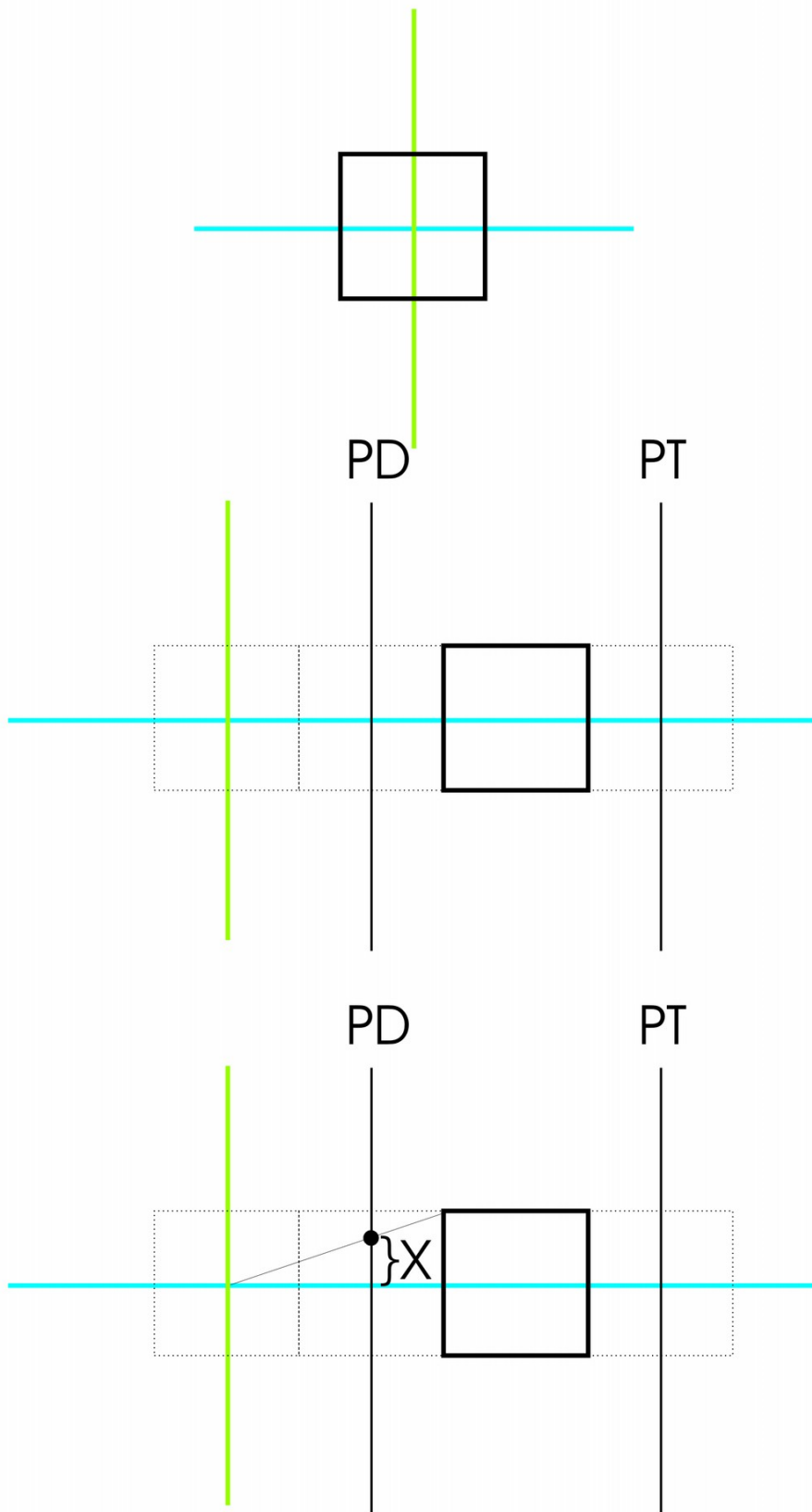
    if (Vertices[i].x>Pos_max.x) Pos_max.x=Vertices[i].x;
    if (Vertices[i].y>Pos_max.y) Pos_max.y=Vertices[i].y;
    if (Vertices[i].z>Pos_max.z) Pos_max.z=Vertices[i].z;
}
```



**KEEP
CALM
AND
ESTUDIA
UN POQUITO**

3. Tenemos un cubo unitario centrado en el origen. La transformación de vista consiste en una traslación, $\text{glTranslate}(0,0,-2)$. Dado que el plano delantero está a una distancia 1, el plano trasero a una distancia 3, calcular los valores de la ventana ($x_{\min}, y_{\min}, x_{\max}, y_{\max}$), de tal manera que la proyección ocupe la mitad de la ventana (ver figura). **(2 ptos) Tiempo máximo: 15 minutos**





$$\frac{X}{1} = \frac{0.5}{1.5} \quad X = 0.3333$$



Descarga la APP de Wuolah.

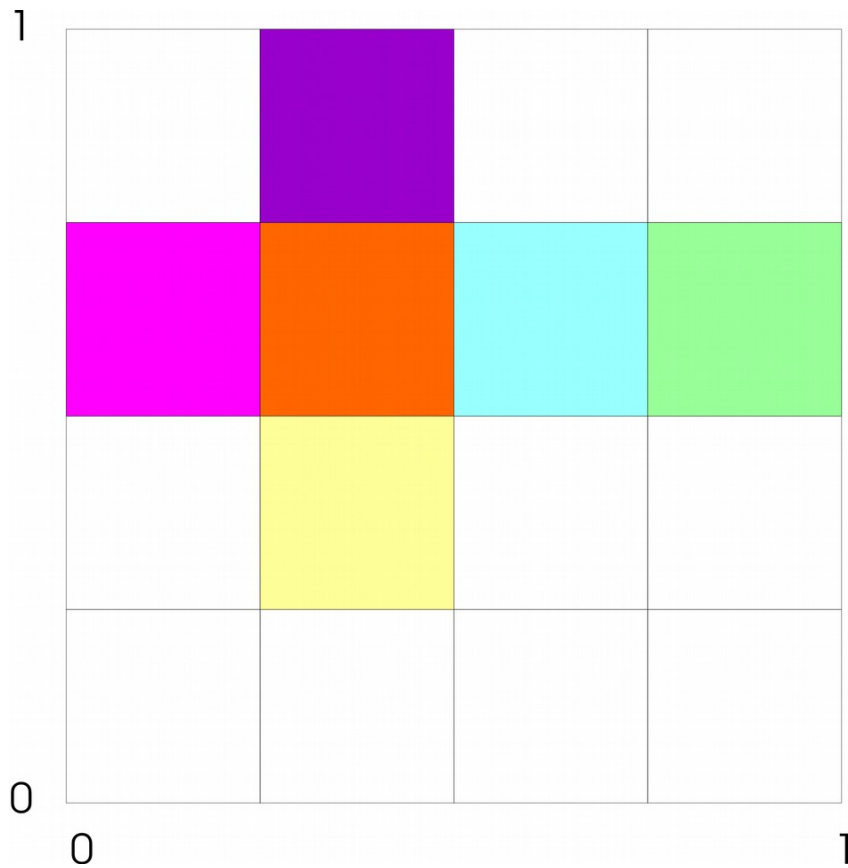
Ya disponible para el móvil y la tablet.



`glFrustum(-0.6666,0.6666,-0.6666,0.6666,1,3)`

4. Tenemos un cubo definido por sus 8 vértices y 12 triángulos y queremos aplicarle la textura de esta manera (ver imagen). Indicar si se puede hacer o no. En caso de que no se pueda, exponer cómo se resolvería. Indicar los valores de las coordenadas de textura para cada vértice.

(3 pts) Tiempo máximo: 20 minutos



Solución

Como se puede ver en la imagen, el problema es que un mismo vértice, y por tanto sus coordenadas de textura, aparecen repetidos. Por ejemplo, el V0 se repite 3 veces. Eso implica que la posición del vértice es la misma pero para cada cara tiene unas coordenadas de textura.

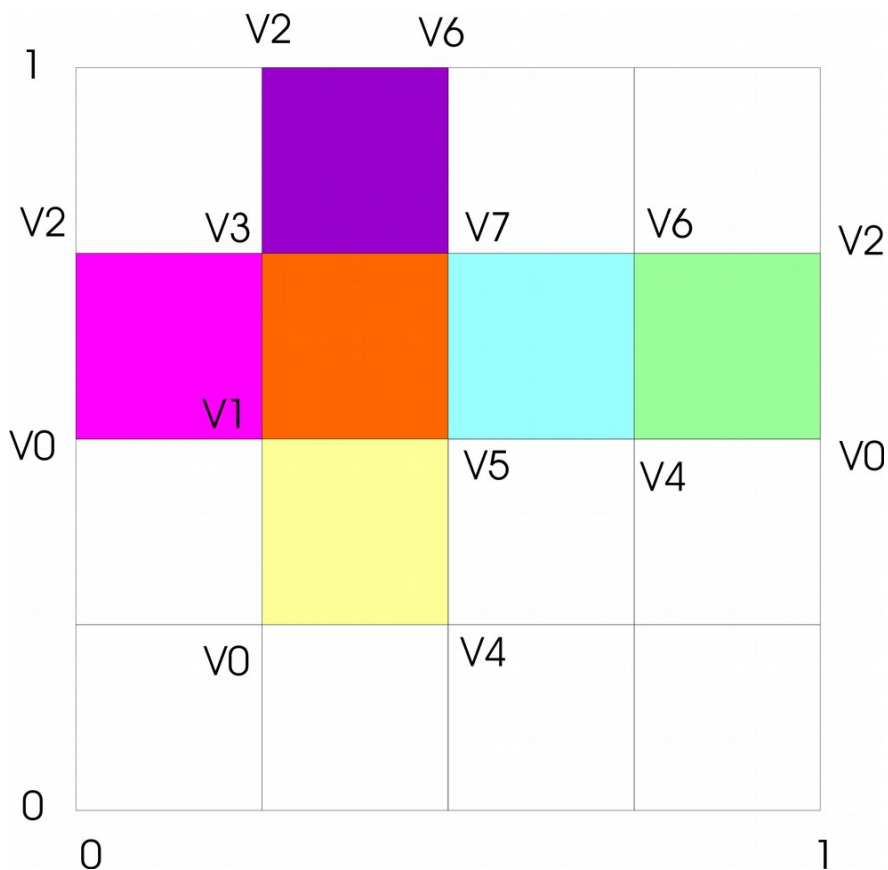
Por tanto la solución pasa por repetir los vértices y tener una descripción para los triángulos que usa esos vértices nuevos. Para cada uno de estos nuevos vértices debe haber un elemento que guarde las coordenadas de textura correspondientes.

Por ejemplo, tendríamos

$V0a \rightarrow (0,0.5)$

$V0b \rightarrow (0.25,0.25)$

$V0c \rightarrow (1,0.5)$



5. Explique los diferentes métodos que se pueden usar para realizar la selección o pick. (2 pts)
Tiempo máximo: 20 minutos

Hay tres formas de realizar la selección:

a) Identificación por color

Este método consiste en lo siguiente. A cada objeto que se quiere identificar se le asigna un identificador, un número natural. Este número es convertido a un color. Se activa la eliminación de partes ocultas. Cuando se dibuja el objeto se usa el color que tiene asociado. Al mover el cursor y pulsar para realizar la selección se guardan las coordenadas x e y del pixel seleccionado. Se lee el pixel del framebuffer en la posición x e y. Se convierte el color al identificador.

En el caso de representar el color con el modelo RGB y 24 bits, se tienen la posibilidad de identificar $2^{24}-1$ objetos diferentes. El color blanco implica que no se ha seleccionado nada

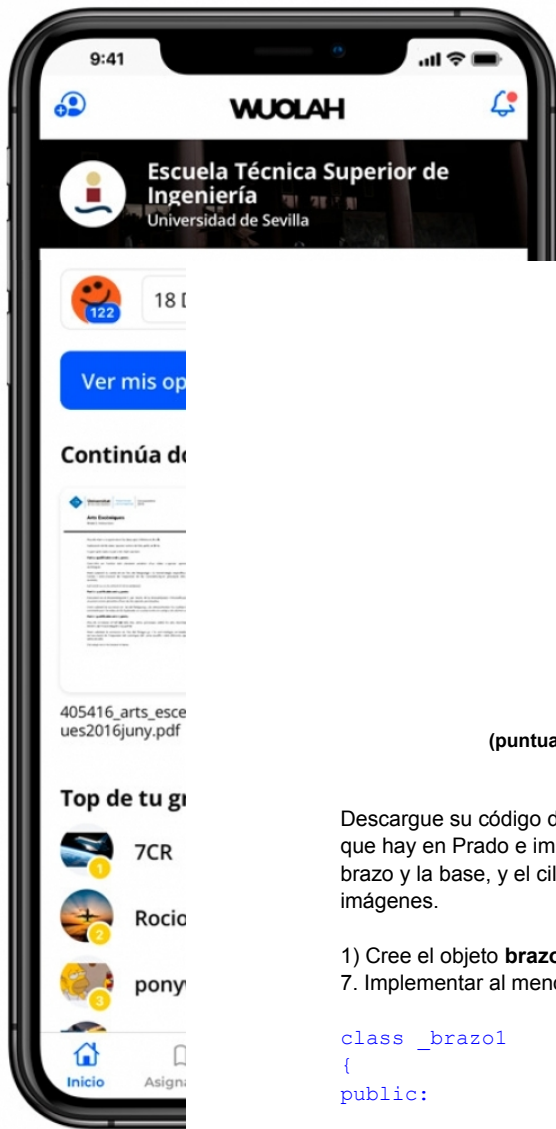
Para pasar del identificador al color se usan mascarar de bits para obtener cada parte. Para pasar de color a ID se hacen los pasos inversos

b) Lanzando un rayo

La idea básica es que cuando pulso el botón del ratón voy a seleccionar el objeto más cercano que está en la posición del cursor. Para ello, obtenemos la posición x e y del cursor en coordenadas de dispositivo y las convertimos a coordenadas de vista. Hacemos pasar una línea recta (o rayo) por el centro de proyección y la nueva posición. Esto es, obtenemos la ecuación de una recta. Calculamos la intersección con los objetos. Si hay intersección se añade a la lista, guardando el identificador del objeto y la profundidad. Por último ordenamos en profundidad y devolvemos el identificador de la intersección más cercana.

c) Por ventana

Este método consiste en aprovechar la etapa de discretización del cauce visual. Esto es, cuando pasamos de fórmulas (líneas, triángulos, etc) a píxeles. La idea consiste en que cuando se marca la posición con el ratón, se obtiene una posición x e y. Alrededor de dicha posición se crea una pequeña ventana (normalmente unos pocos píxeles). Una vez identificados los píxeles que conforman la ventana, lo único que hay que hacer es dibujar cada objeto, al cual se le asigna un identificador. Si al convertir el objeto en píxeles, coincide con alguno o varios de la ventana, entonces hay selección. Se guarda el identificador del objeto y la profundidad. Finalmente podemos hacer una ordenación por profundidad y quedarnos con el identificador del más cercano.



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



Prácticas

(puntuación expresada sobre 10. Mínimo 3,5 para que cuente en la nota final)

Descargue su código de la **práctica 2** o en caso de que no la tenga hecha, el **esqueleto de la práctica 1** que hay en Prado e implemente el modelo de un brazo articulado. Para ello se usará el cubo para el brazo y la base, y el cilindro para el soporte de giro (o el cubo si no tiene cilindro). Fijarse en las imágenes.

- 1) Cree el objeto **brazo1** usando el cubo con las siguientes dimensiones (1,10,1) . Se activa con la tecla 7. Implementar al menos el modo de visualización chess. (10% de la nota total)

```
class _brazo1
{
public:

    void draw_chess();
protected:
    _cube Cube;
};

void _brazo1::draw_chess()
{
    glPushMatrix();
    glTranslatef(5,0,0);
    glScalef(10,1,1);
    Cube.draw_chess();
    glPopMatrix();
}
```

- 2) Cree el objeto **brazo2** usando el cubo las siguientes dimensiones (1.5,15,1.5). Añadir el objeto brazo1 en el extremo. Se activa con la tecla 8. Implementar al menos el modo de visualización chess. (10% de la nota total)

```
class _brazo2
{
public:

    void draw_chess();

    void increase_displacement(){Position=Position+1;}
    void decrease_displacement(){Position=Position-1;}

protected:
    _brazo1 Brazo1;
    _cube Cube;
```

```

    float Position=0;
};

void _brazo2::draw_chess()
{
    glPushMatrix();
    glTranslatef(7.5,0,0);
    glScalef(15,1.5,1.5);
    Cube.draw_chess();
    glPopMatrix();

    glPushMatrix();
    glTranslatef(Position,0,0);
    glTranslatef(15,0,0);
    Brazo1.draw_chess();
    glPopMatrix();
}

```

3) Permita que el **brazo1** incluido en el objeto **brazo2** se pueda mover hacia la derecha y la izquierda las teclas Q y W. (25% de la nota total)

```

case Qt::Key_Q:Brazo2.increase_displacement();break;
case Qt::Key_W:Brazo2.decrease_displacement();break;

```

4) Cree el objeto **pivote** con un cilindro con las siguientes dimensiones (1.5,1.5,1.5), orientado y posicionado de la forma correcta. Añada el objeto **brazo2** de tal manera que quede en el borde del cilindro.(En caso de usar un cubo, se debe colocar a la misma distancia que si se hubiera usado el cilindro) (30% de la nota total)

```

class _pivote
{
public:

    _pivote();

    void draw_chess();

    void increase_displacement(){Brazo2.increase_displacement();}
    void decrease_displacement(){Brazo2.decrease_displacement();}

    void increase_angle(){Angle=Angle+1;}
    void decrease_angle(){Angle=Angle-1;}

protected:
    _brazo2 Brazo2;
    _cylinder Cylinder;

    float Angle=0;

```

```
};
```

```
_pivote::_pivote()  
{  
    Cylinder.create(40);  
}
```

```
void _pivote::draw_chess()  
{  
    glPushMatrix();  
    glRotatef(90,1,0,0);  
    glScalef(1.5,1.5,1.5);  
    Cylinder.draw_chess();  
    glPopMatrix();  
  
    glPushMatrix();  
    glRotatef(Angle,0,0,1);  
    glTranslatef(0.75,0,0);  
    Brazo2.draw_chess();  
    glPopMatrix();  
}
```

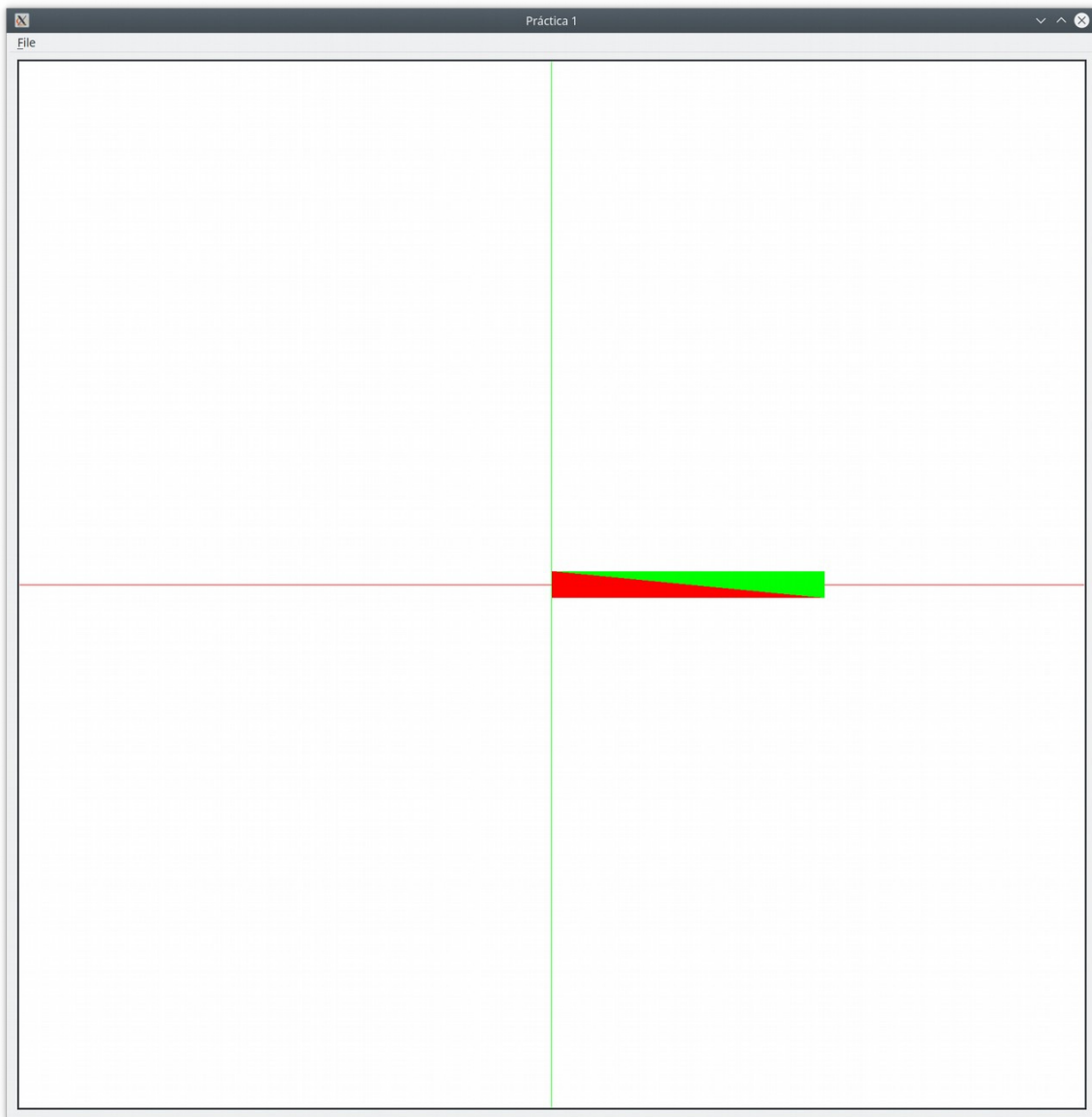
5) Permita que el objeto pivote pueda mover el **brazo1** hacia la izquierda y la derecha con las teclas A y S, y que el **brazo2** incluido se pueda mover hacia arriba y hacia abajo con las teclas Z y X. (25% de la nota total)

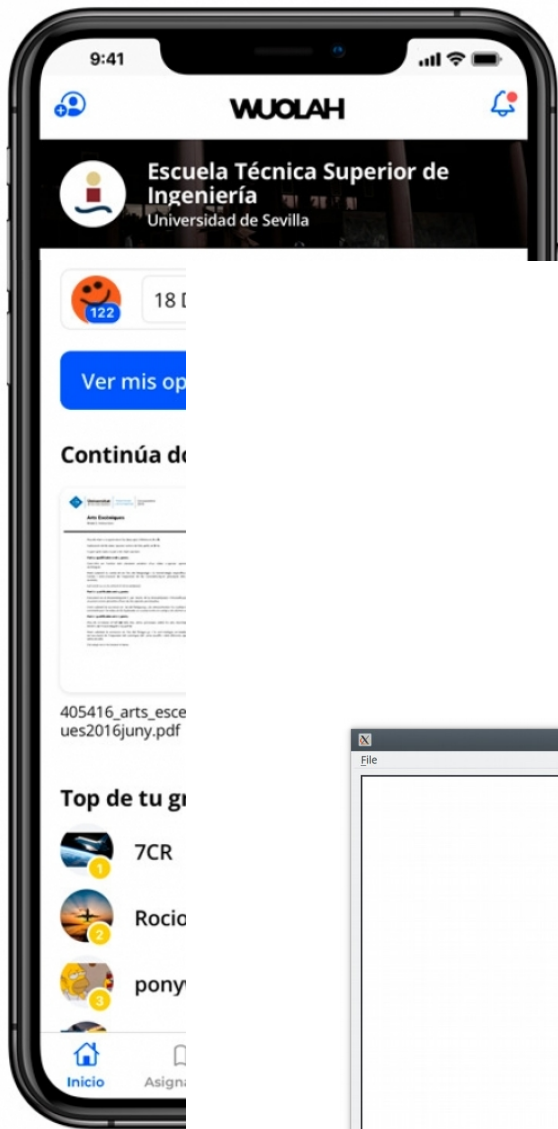
```
case Qt::Key_A: Soporte.increase_displacement(); break;  
case Qt::Key_S: Soporte.decrease_displacement(); break;  
  
case Qt::Key_Z: Soporte.increase_angle(); break;  
case Qt::Key_X: Soporte.decrease_angle(); break;
```

Para cada apartado, se valorará con la nota establecida el que se haya programado y se visualice correctamente y con 0 en otro caso. Cada apartado se valorará si el anterior está correcto.

(Nota total: 10 puntos)

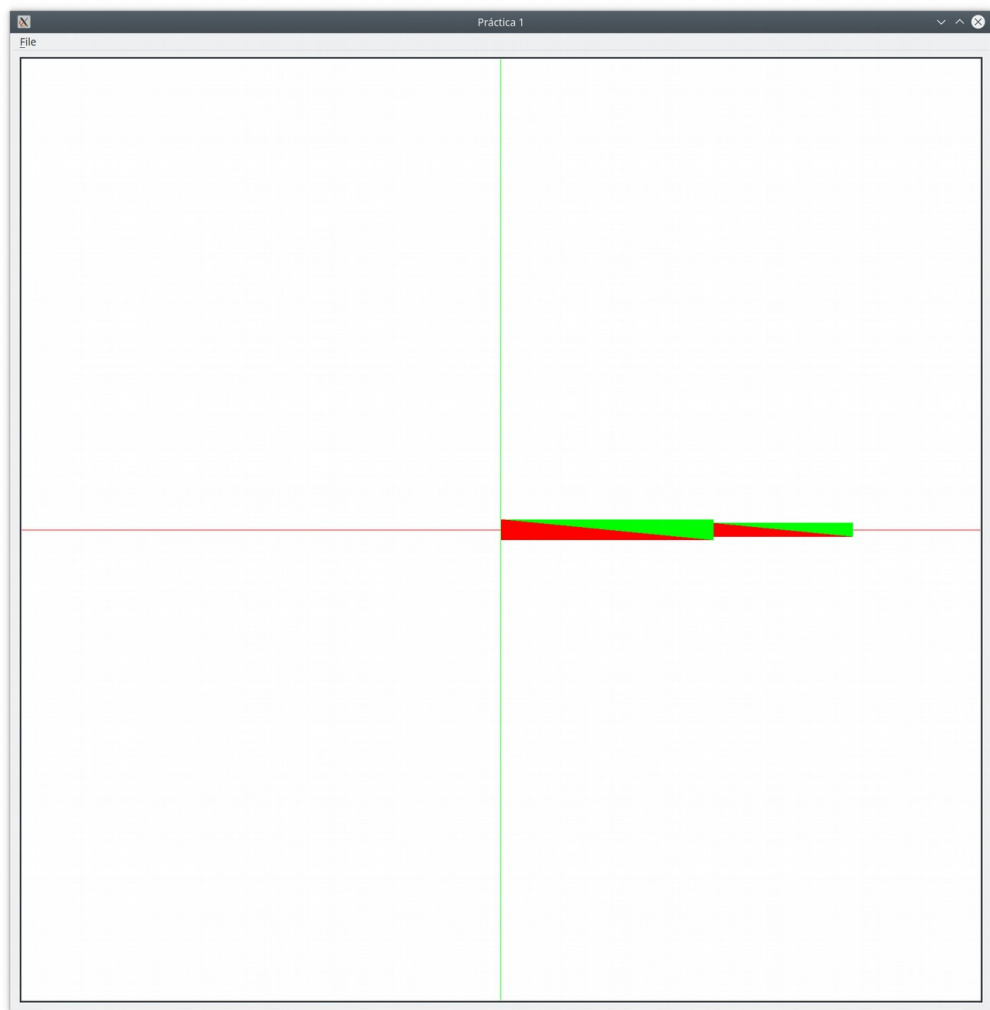
(Tiempo máximo: 75 minutos)



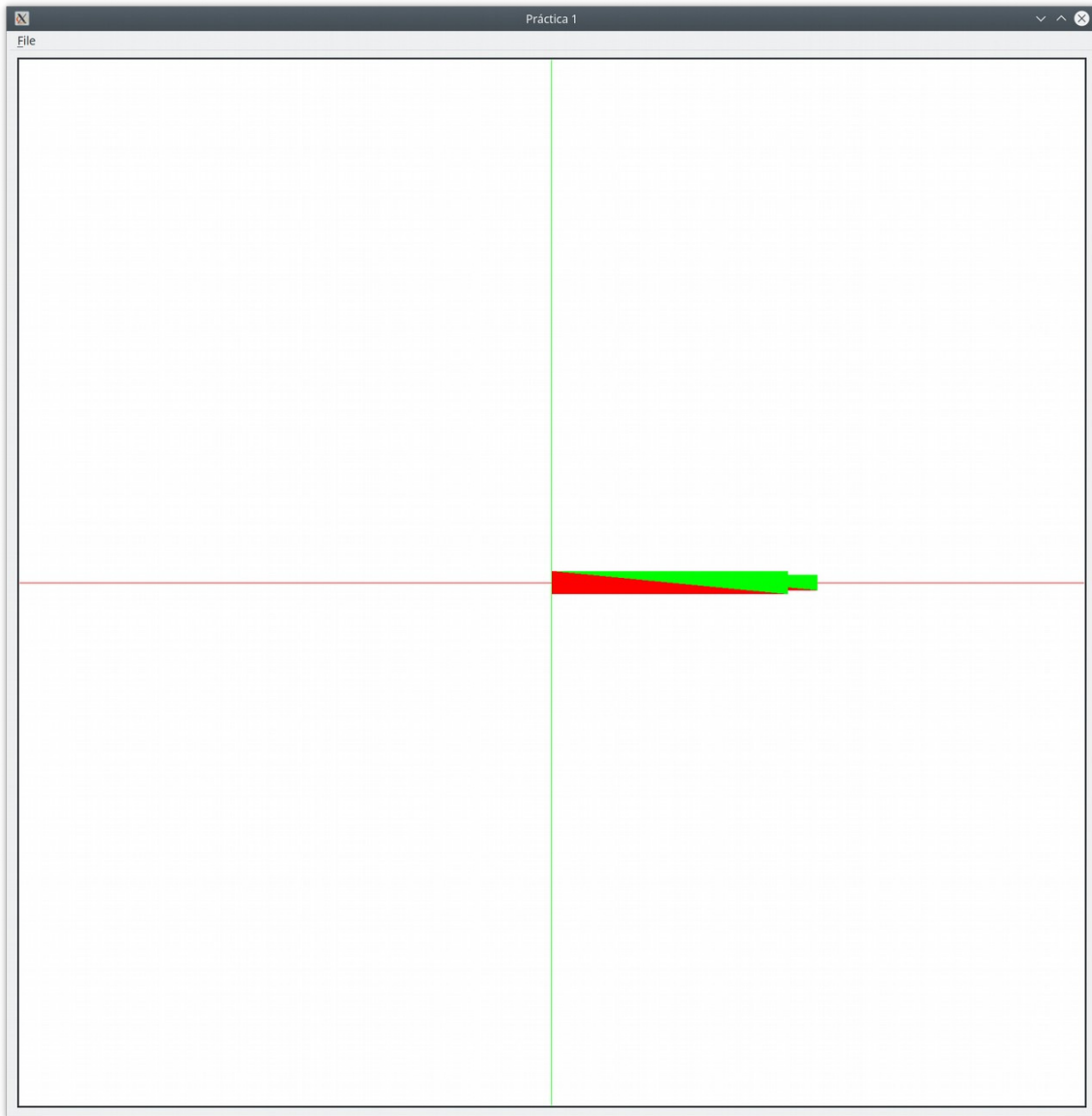


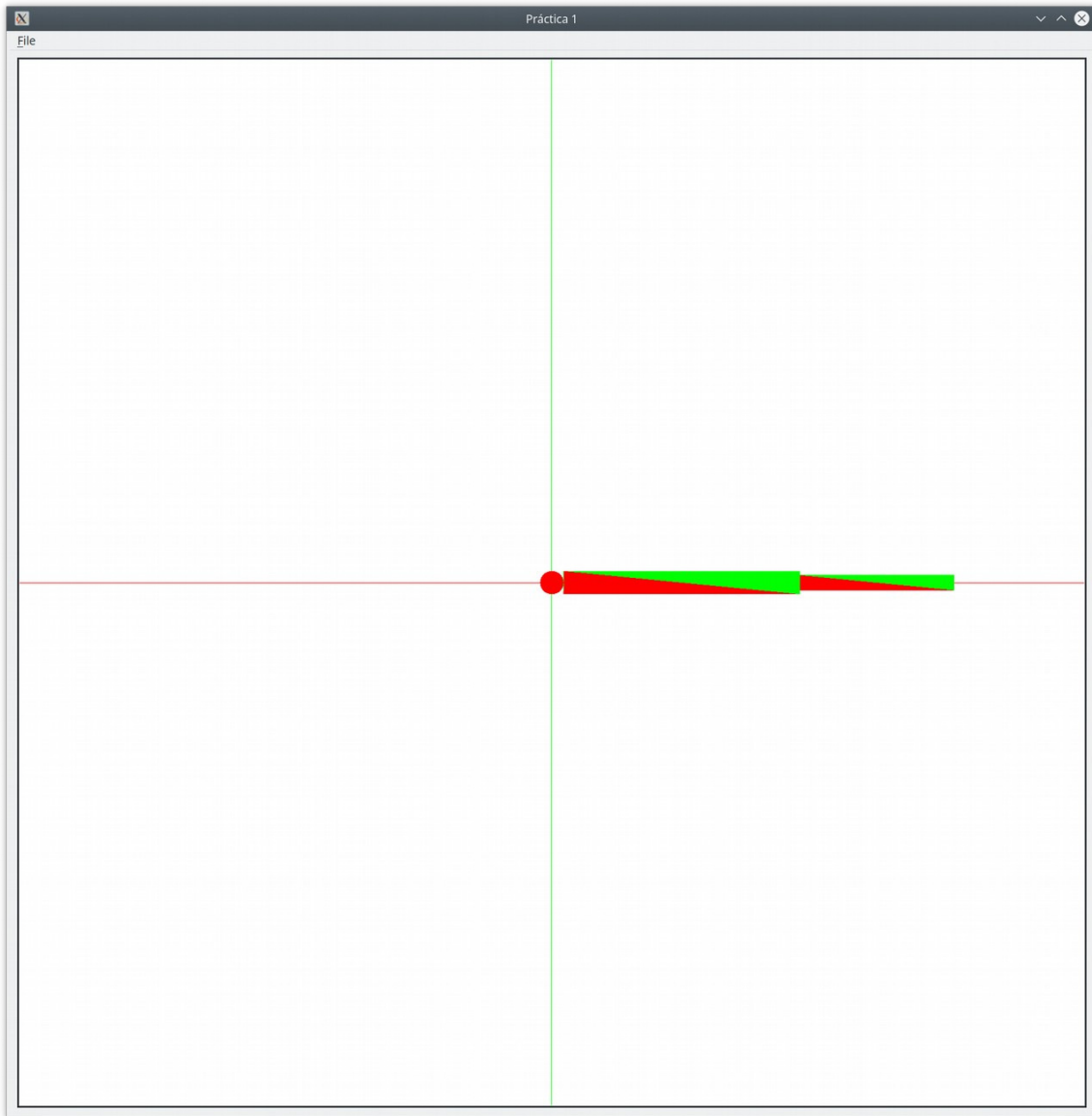
Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

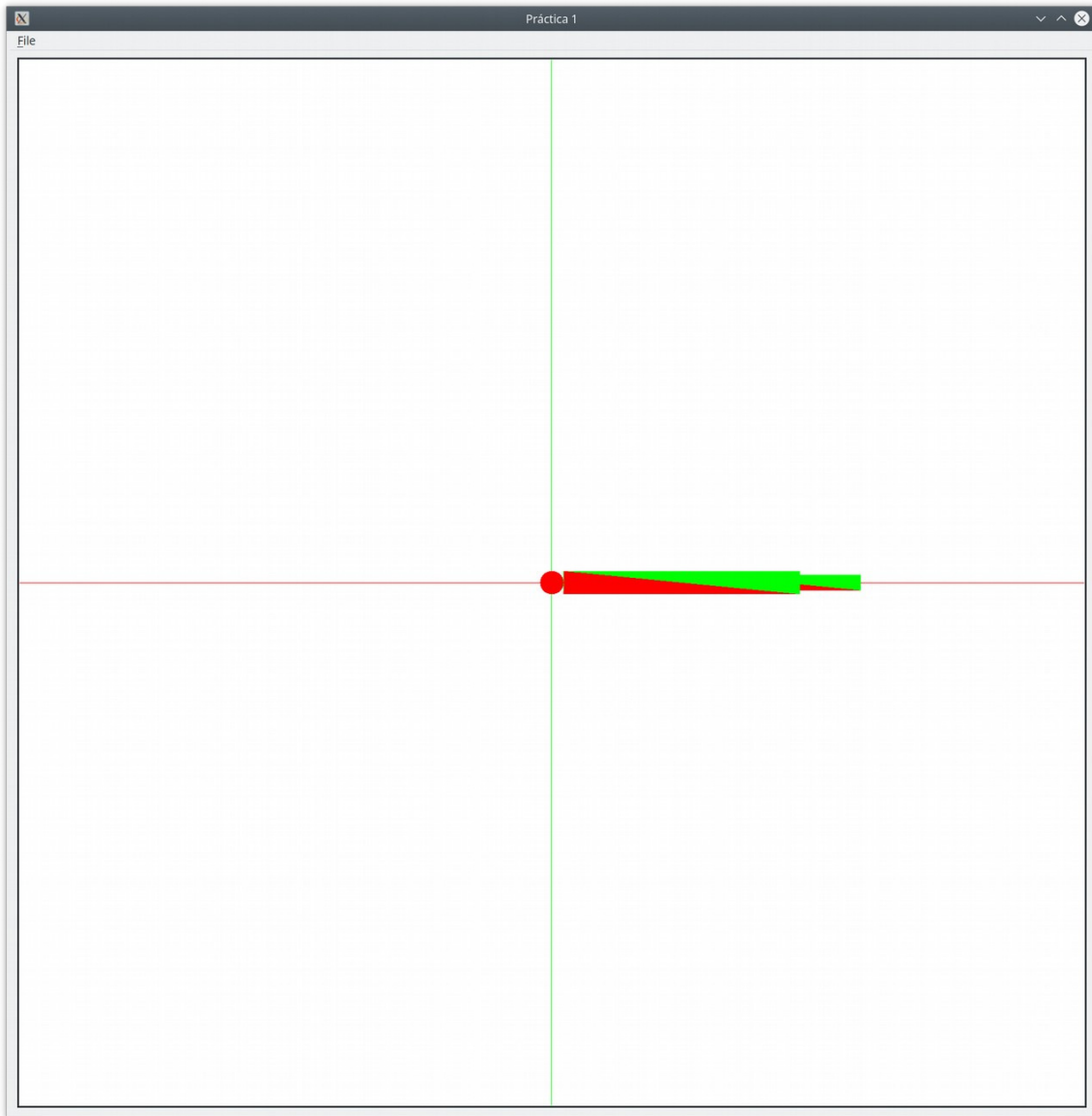


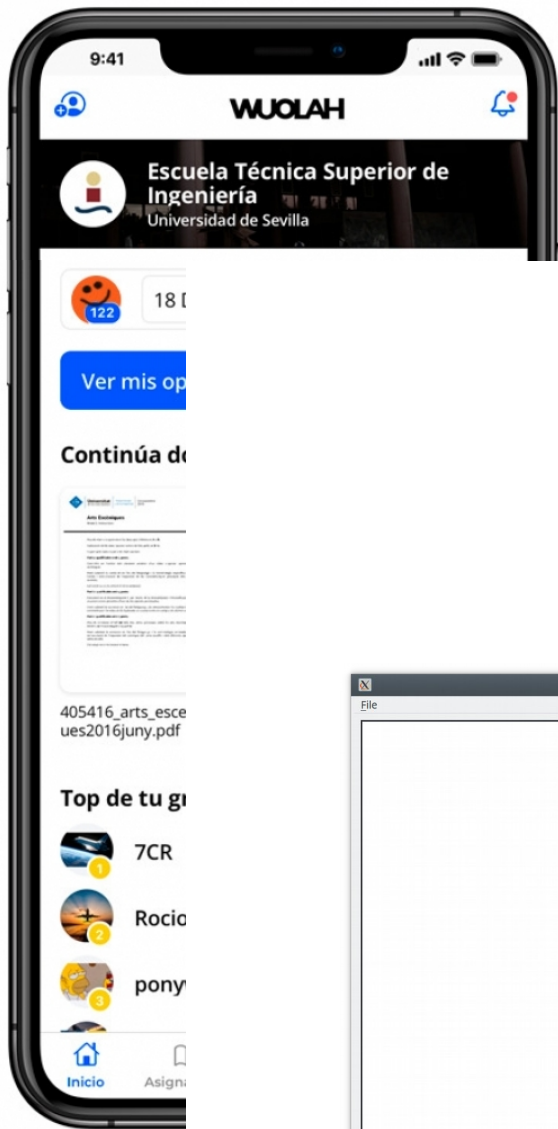
Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.





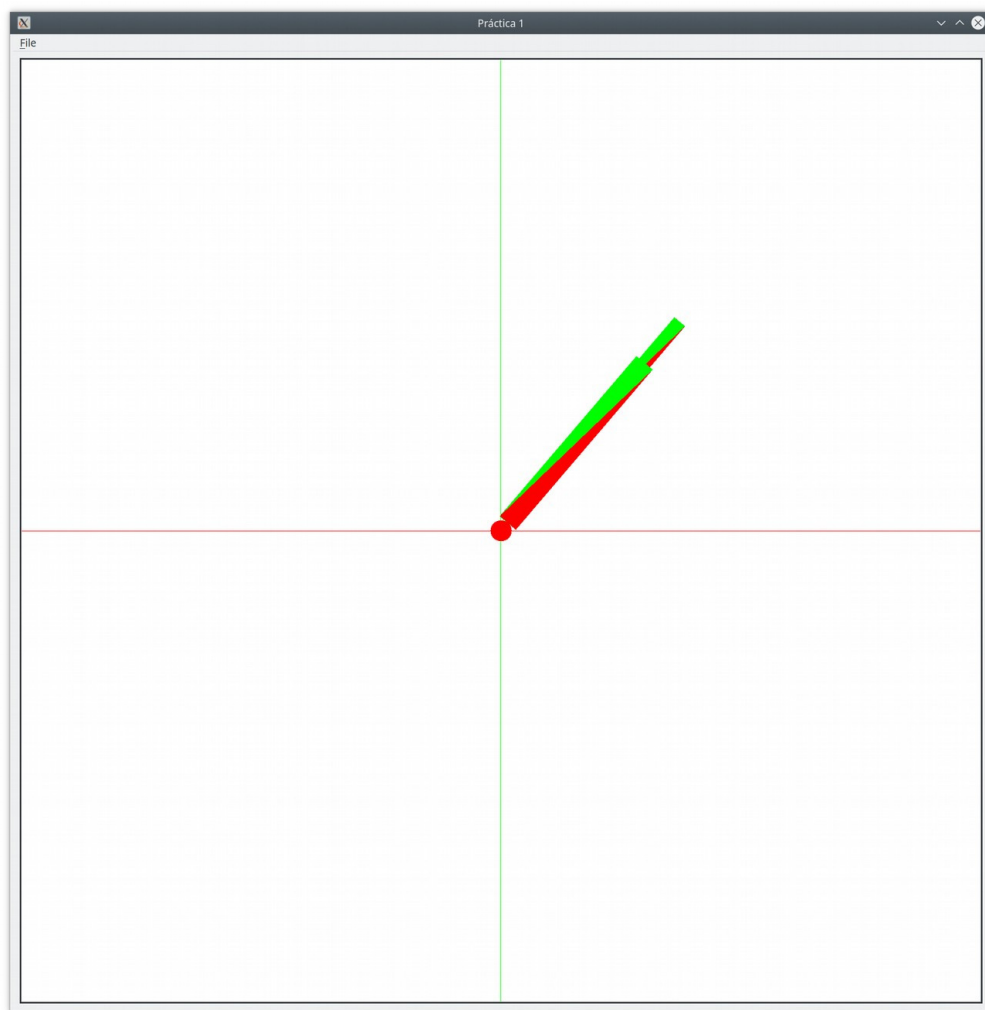
Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.





Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

