Pràctica 0

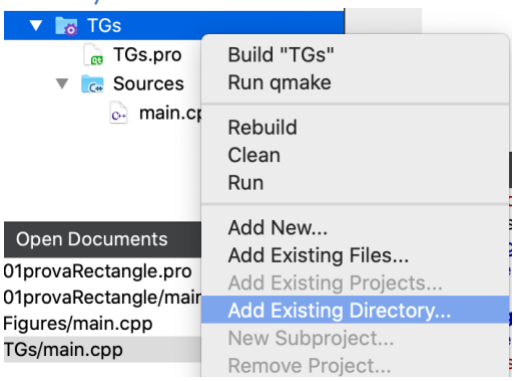
Fitxa 4: Repàs de Transformacions Geomètriques

GiVD - curs 2022-2023

En aquesta pràctica es repassen conceptes bàsics de geometria que s'utilitzaran al llarg del curs, mitjançant un projecte en C++ usant l'IDE QtCreator.

Abans de començar:

1. Crea un subprojecte de la practica0Pas2 en l'IDE QtCreator (veure Fitxa 3 per saber crear un subprojecte) anoment TGs
2. Afegeix la carpeta glm al projecte que trobaràs al campus virtual. Aquesta carpeta conté definicions de tipus i operacions que s'utilitzaran al llarg d'aquest tutorial. Per a això, segueix els següents passos:
   a. Descarrega el fitxer glm.tgz del campus virtual
   b. Descomprimeix el fitxer en una carpeta on s'ha creat el subprojecte
   c. Afegeix aquesta carpeta al projecte des de QtCreator, clic dret sobre el projecte - Add Existing Directory...



Al llarg d'aquest tutorial, s'aniran programant, en el projecte creat, els diferents conceptes segons es vagin introduint.

Material de suport:

- Transparències que resumeixen Transformacions Geomètriques:
- https://www.embibe.com/learn/mathematics/unit-coordinate-geometry-68
- https://www.geogebra.org/3d

1. Vectors i punts

Un punt és la representació més bàsica en geometria. Serveix per definir vèrtexs de triangles o posicions d'objectes.

1.1. Vectors i punts Geogebra

GeoGebra permet abordar la geometria des d'una forma dinàmica i interactiva. Ofereix representacions diverses dels objectes des de cadascuna de les seves possibles perspectives. Usarem aquesta eina per visualitzar els diferents conceptes de geometria. Realitzarem construccions de manera fàcil i ràpida, permetent-nos la transformació dinàmica dels objectes que creem.

Podem usar aquesta aplicació directament en línia:

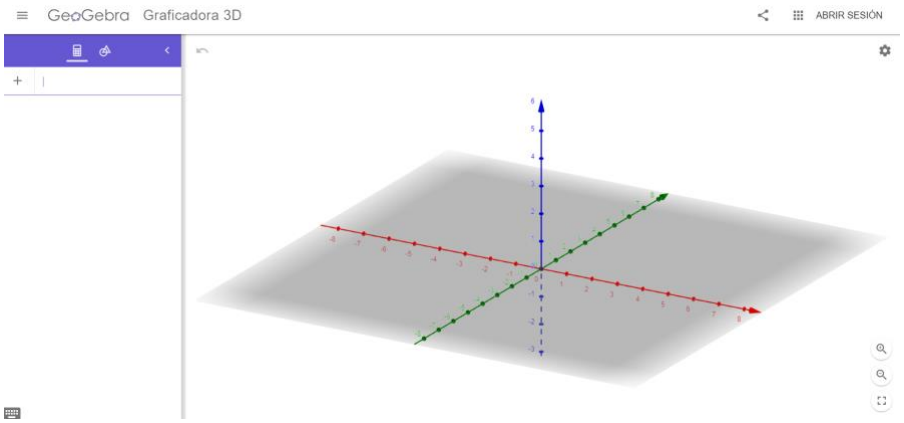
https://www.geogebra.org/3d

O bé, descarregar el programa al següent enllaç:

https://www.geogebra.org/download?lang=es-es

Exercici Geogebra vectors i punts:

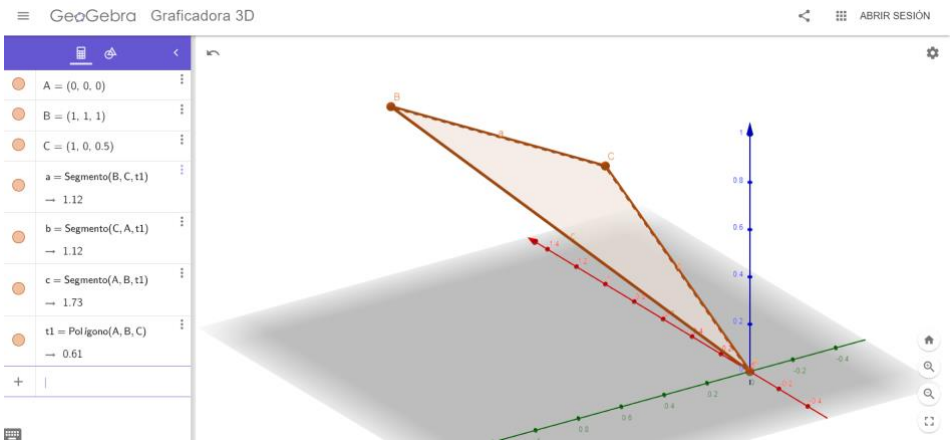
1. Obrir la pàgina de GeoGebra.



2. Afegirem un triangle de vèrtexs A=(0,0,0), B=(1,1,1) i C=(1,0,0.5).

   Per a agregar un punt, simplement hem d'afegir un ítem definint-ho de la següent manera: P=(x, y, z).

   Una vegada introduïts els punts del triangle, el crearem amb la comanda t1 = Polígon(A,B,C) (vigileu l'idioma amb el que executeu el GeoGebra per què poden canviar les comandes). En crear-ho, es generen els tres vectors que conformen el triangle.



3. Crearem un segon triangle amb els mateixos punts però en sentit invers, t2=Polígon(C,B,A).
4. A continuació, calcularem el vector normal del triangle t1. Un vector normal és un vector perpendicular a qualsevol objecte geomètric (línia, corba, superfície, etc). Per calcular-ho usarem la comanda n1=VectorNormal(t1).
5. Calcularem també el vector normal del triangle t2. Quina diferència hi ha amb el vector normal del triangle t1? Perquè?
6. Calcula el vector normal del triangle (B,C,A). Quines similituds i diferències hi ha amb els vectors n1 i n2?

1.2. Vectors i punts C++

Punts 3D

Representarem els punts amb 4 coordenades, les 3 primeres corresponen a les posicions en els eixos x, y, z mentre que la quarta component, w, s'utilitza per a l'homogeneïtzació.

Per declarar un punt en un programa, s'utilitzarà la classe proporcionada vec4. Per a això, has de descarregar del campus l'arxiu glm.zip, descomprimir-ho en el directori arrel del projecte i incloure la carpeta glm en el projecte. Després, en els arxius del programa que ho necessitis, s'ha d'incloure l'arxiu de capçalera:

```
#include "glm/glm.hpp"
```

I per declarar i inicialitzar un punt es pot fer de la següent manera:

```
glm::vec4 vertice(0,0,0,1);
```

Vectors

Els vectors serveixen per indicar direccions, per exemple la direcció de la llum, o la recta normal d'un objecte. Les accions més utilitzades en les properes activitats i pràctiques són: el mòdul d'un vector, la seva normalització, el producte escalar i el producte vectorial. La llibreria glm que t'has descarregat conté funcions per normalitzar (`glm::normalize`), realitzar el producte escalar de 2 vectors (`glm::dot`) i el producte vectorial (`glm::cross`). També pot interessar-te utilitzar la funció arc cosinus (`glm::acos`) inclosa en la mateixa llibreria.

Exercici amb vectors i punts:

1. Crea una nova classe `Figura` que tingui el següent mètode abstracte:

```
virtual glm::vec4 obteNormal() = 0;
```

2. Crea una nova classe `Triangle` amb els seus atributs corresponents que hereti de `Figura` i implementi els mètodes:

```
glm::vec4 obteNormal();
float obteAngle(int idxVexterx);
```

3. Crea un triangle en el main que tingui els vèrtexs (0,0,0) (1,1,1) (1,0,0.5). Calcula i mostra per consola el seu normal i els seus 3 angles.

2. Treballar amb Matrius i Transformacions Geomètriques

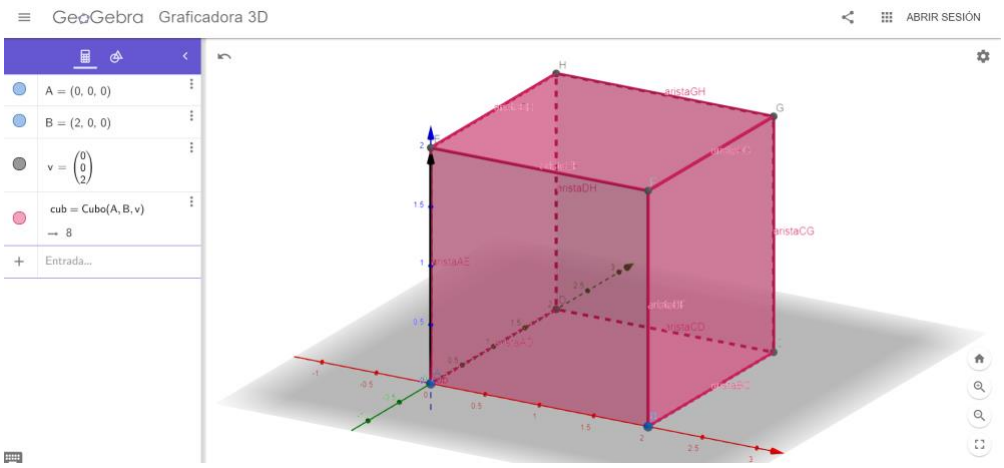
Les transformacions geomètriques són operacions o funcions que ens permeten crear un altre objecte homòleg a l'original. Per realitzar transformacions geomètriques utilitzarem matrius (veure transparències de suport al campus).

2.1. Vectors i punts GeoGebra

Exercici GeoGebra matrius i transformacions:

Crearem un cub de vèrtexs A=(0,0,0), B=(2,0,0), C=(2,2,0), D=(0,2,0), I=(0,0,2), F=(2,0,2), G=(2,2,2), H=(0,2,2). A GeoGebra, per crear un cub necessitem dos punts i un vector director.

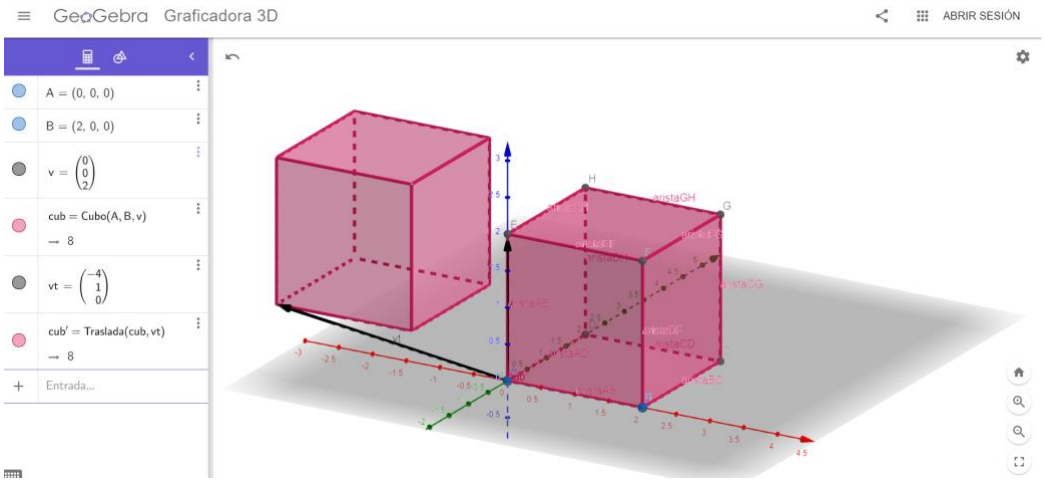
1. Crea els punts A i B.
2. Crea el vector v = $\overrightarrow{AE}$
3. Crea el cub, fent l'assignació cub=Cub(A,B,v).



Translació

Una translació consisteix en un desplaçament de l'objecte, sense canvis d'orientació, definit per un vector. Per definir una translació necessitarem l'objecte a moure i un vector per a indicar l'adreça del moviment.

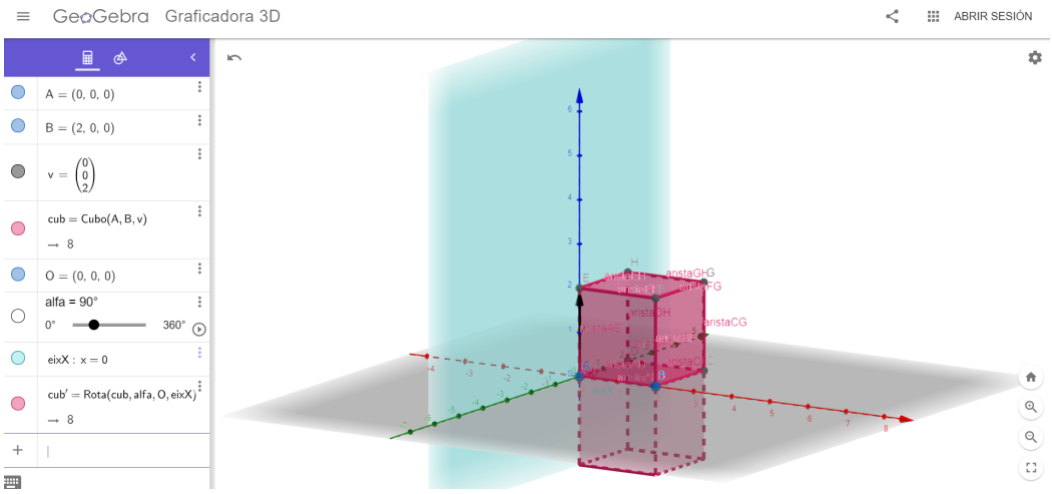
4. Definim el vector vt=(-4,1-0)
5. Traslladem l'objecte amb la funció Traslada(<Objecte>, <Vector>). Quins punts s'han traslladat?



Rotació

La rotació permet girar cada punt d'un objecte un mateix angle α al voltant del centre O sobre un eix de rotació.

6. Definim el centre de la rotació O=(0,0,0), l'angle alfa = 90º i l'eix eixX: x=0.
7. Rotem l'objecte amb la funció Rota(<Objecte>, <Angle>, <Punt sobre eix>, <Eix Director>)



8. Traslladem l'objecte aquesta vegada amb el centre O=(3,0,3), alfa 180º i eix i=0.
9. Obtenim el mateix resultat si roten i traslladem que si traslladem i després rotem?

2.2. Vectors i punts C++

Per a declarar una matriu, ho podeu fer de la següent manera:

```
glm::mat4 transforma;
```

Exercicis amb matrius i transformacions

1. Afegeix a la classe **`Figura`** els mètodes abstractes:

```
virtual void trasllada(glm::vec3 desplazament) = 0;
virtual void escala(glm::vec3 factor_escalat) = 0;
virtual void rota(float radians, glm::vec3 punt_rotacio,
                  glm::vec3 eix_rotacio) = 0;
```

2. Implementa'ls per a la classe **Triangle**. I realitza els següents càlculs per al triangle que ja tens definit:

2.1. Trasllada el triangle 5 unitats en l'eix x, 9 unitats en l'eix i, i 2 unitats en l'eix z.

2.2. Escala el triangle en 2 unitats en l'eix de les x, 0.5 unitats en l'eix i, i 5 unitats en l'eix z.

2.3. Rota el triangle PI / 2 radiants en l'eix de les x al voltant del seu centre.

4. Crea una nova classe **Rectangle** que hereti de **Figura** i implementa tots els mètodes anteriors. Crea un rectangle i comprova els diferents mètodes implementats.