

UNIVERSITAT DE LES ILLES BALEARS

ENGINYERIA INFORMÀTICA 2020-2021

INFORMÀTICA GRÀFICA

Etapas

Alumnes:

Joan Sart Vizcaíno

Carles Serrano Rueda

Lluc Valdés Carrasco

Data:

16 de Juny 2021

Índice

	Pàgina
1. Etapa 1	2
2. Etapa 2	4
2.1. Doble Buffer	4
2.2. Manteniment d'aspecte/ratio	5

1. Etapa 1

Aquesta primera etapa consisteix en podem compilar un projecte emprant *OpenGL*. El nostre entorn de treball ha estat **Visual Studio ++** en **Windows**.

El primer que tenim és una carpeta que és diu *BasicGLUT32-master* amb els següents arxius:

.vs	20/3/2021 15:47	Carpeta de fitxers	
Debug	20/3/2021 15:47	Carpeta de fitxers	
GL	20/3/2021 15:47	Carpeta de fitxers	
BasicGLUT32.vcxproj	20/3/2021 15:47	VC++ Project	8 kB
BasicGLUT32.vcxproj.user	20/3/2021 15:47	Per-User Project Opti...	1 kB
glut32.dll	20/3/2021 15:47	Extensió de l'aplicació	232 kB
glut32.lib	20/3/2021 15:47	Object File Library	29 kB
main.cpp	20/3/2021 15:47	C++ Source	3 kB
README.md	20/3/2021 15:47	Document MD	1 kB

Figura 1: Carpeta

Aquesta carpeta ens possibilita poder emprar Open GL.

Per poder-ho emprar al teu projecte, hauràs d'anar dins la solució del projecte a l'arxiu i anar a propietats del document:

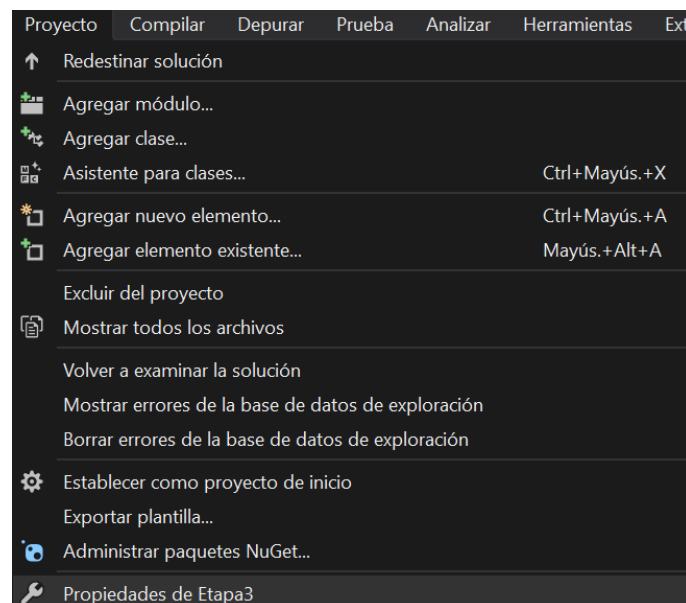


Figura 2: Propietats

Després hem de ficar el path on indica la següent imatge:

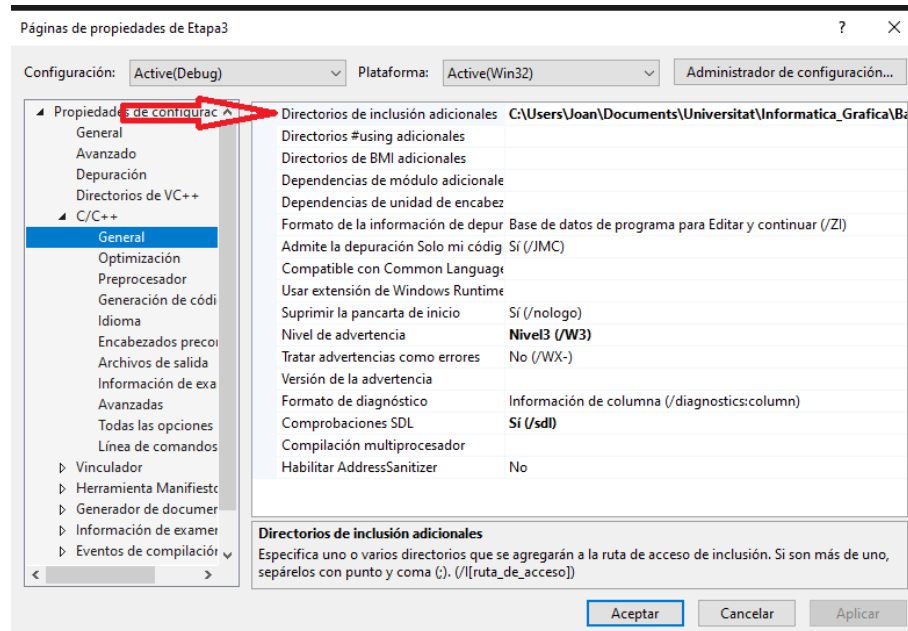


Figura 3: Primera passa

La següent passa no te perquè fer falta però la recomanem perquè d'aquesta manera sempre funciona. És posar el mateix directori que hem posat abans al següent lloc: Després hem de ficar el path on indica la següent imatge:

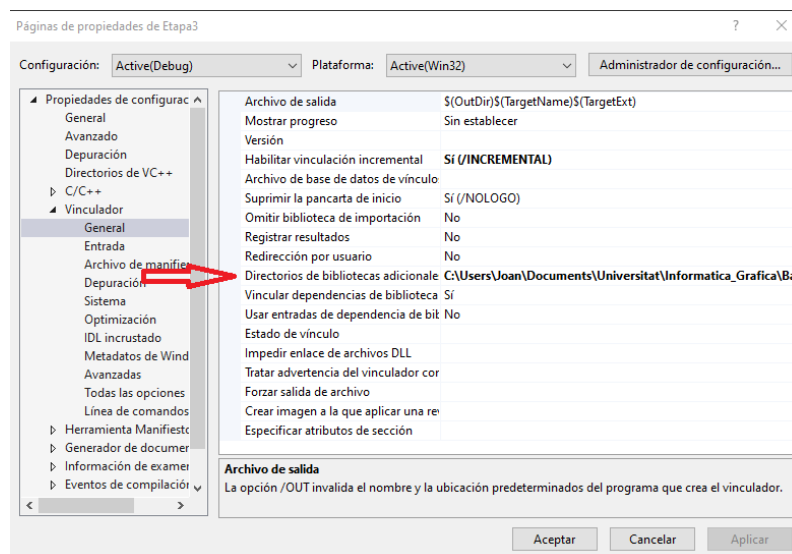


Figura 4: Segona passa

Per continuar hem d'escriure (`glut32.lib`) dins les dependències addicionals:

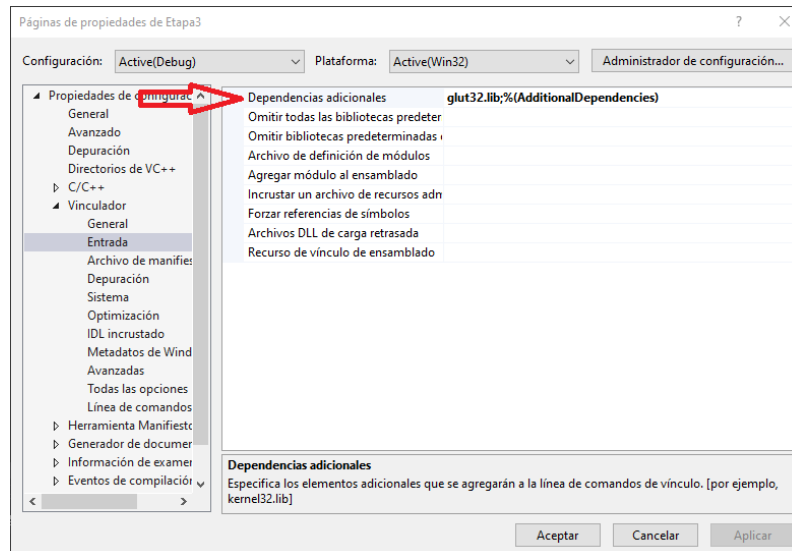


Figura 5: Tercera passa

Per finalitzar haurem de copiar l'arxiu `glut32.dll` a la carpeta del projecte on tenim el nostre **Main.cpp**

2. Etapa 2

Quan ja hem pogut executar un programa emprant **OpenGL** pasam a mantenir l'escalat i emprar el doble buffer.

2.1. Doble Buffer

Per aplicar el doble buffer el que hem fet just al final de la funció de `Display` (la funció de la pantalla):

```
1 | glutSwapBuffers();
```

I posam la següent línia al `main`:

```
1 | glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE);
```

2.2. Manteniment d'aspecte/ratio

Per aquesta part hem creat una funció *reshape()* que el que fa és modificar la zona on és pinta la pantalla. La primer versió ha estat:

```
1 void reshape(int width, int height) {
2     if (width > height) {
3         glViewport(width / 2 - height / 2, 0,
4             height, height);
5         W_WIDTH = height;
6         W_HEIGHT = height;
7         origenX = width / 2 - height / 2;
8         origenY = 0;
9     }else {
10        glViewport(0, height / 2 - width / 2,
11            width, width);
12        W_WIDTH = width;
13        W_HEIGHT = width;
14        origenX = 0;
15        origenY = height / 2 - width / 2;
16    }
17 }
```

En versions posteriors hem evolucionat el *reshape()* per que funcionas millor amb una escena més completa que amb únicament una figura amb dues dimensions:

```
1 void reshape(int w, int h) {
2     // Prevent a divide by zero, when window is too short
3     // (you cant make a window of zero width).
4     if (h == 0) {
5         h = 1;
6     }
7     float ratio = w * 1.0 / h;
8
9     // Use the Projection Matrix
10    glMatrixMode(GL_PROJECTION);
11
12    // Reset Matrix
13    glLoadIdentity();
14
15    // Set the viewport to be the entire window
16    glViewport(0, 0, w, h);
17 }
```

```
18      // Set the correct perspective.  
19      gluPerspective(45.0f, ratio, 0.1f, 100.0f);  
20  
21      // Get Back to the Modelview  
22      glMatrixMode(GL_MODELVIEW);  
23  
24  }
```

Després de tenir fet el reshape hem fet un pèndul doble. La idea bàsicament ha estat agafar una un rectangle que rotàs sobre un extrem i després després feim rotar un altre rectangle sobre l'altre extrem inferior del segon rectangle. Això és pot veure al document [Etapa2.cpp](#).