

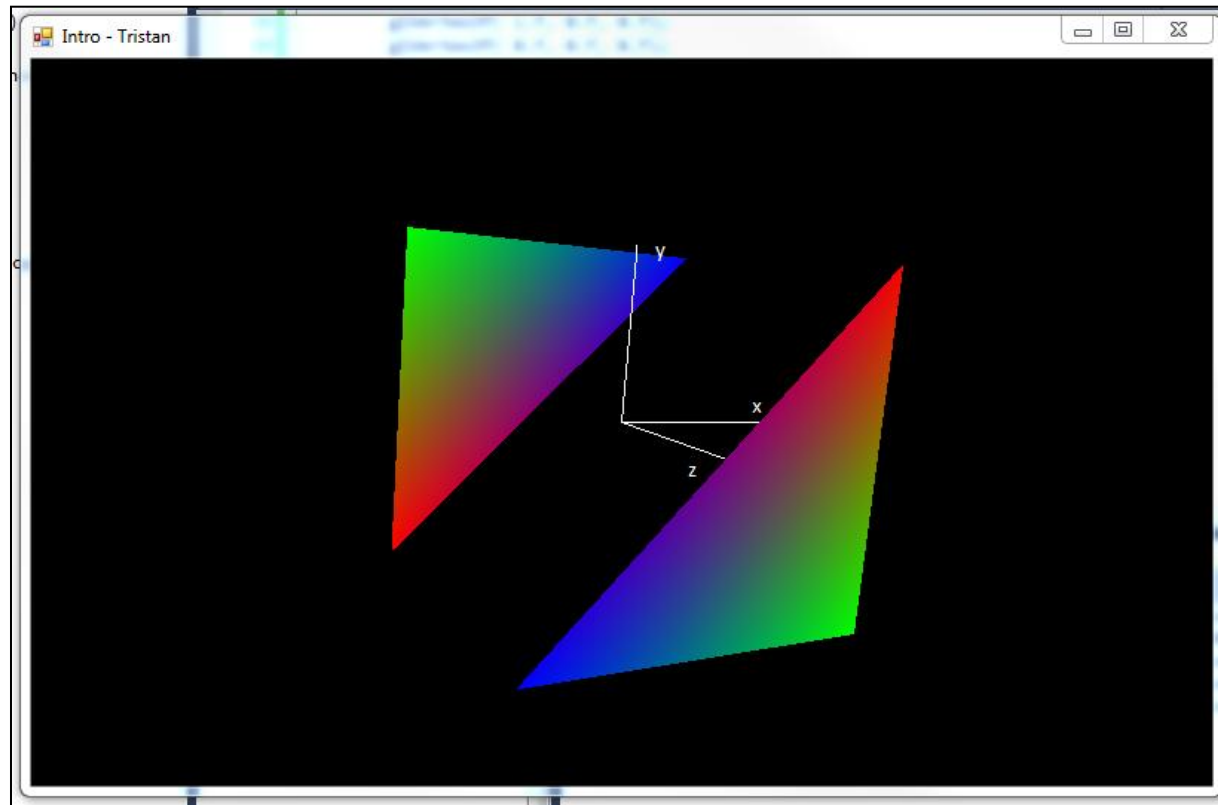
Chapitre 1

Présentation et introduction à l'infographie 3D

Deuxième exemple - 3D.

Passons au 3D – La perspective.

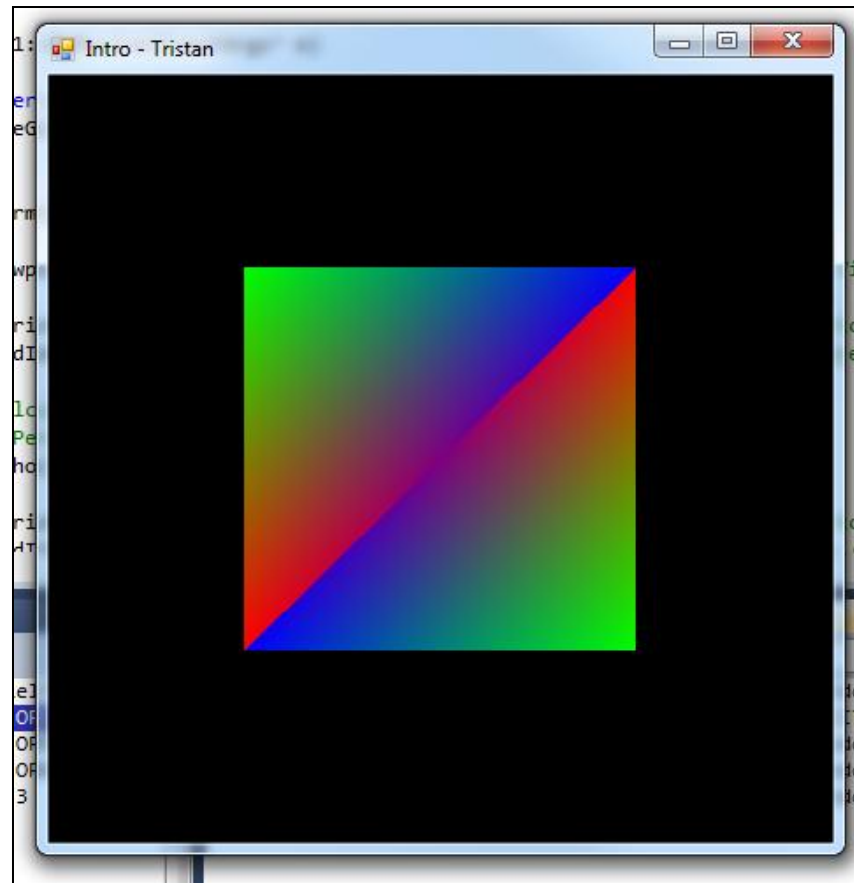
Nous allons créer la scène suivante qui consiste en deux triangles et un repère géométrique pour l'origine de notre système cartésien.



Étape 1 – Les deux triangles.

À partir de ce que nous avons de l'exemple précédant, remplaçons les lignes par deux triangles. Donc dans « Render() »

```
76
77 System::Void Form1::Render(System::Void)
78 {
79     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // Vide le tampon de couleur et de profondeur
80     glLoadIdentity(); // Réinitialise la matrice modèle/vue
81
82     glBegin(GL_TRIANGLES); // Début - en mode triangle
83
84     glColor3f(1.f,0.f,0.f); // Rouge
85     glVertex3f( 1.f, 1.f, 1.f); //
86     glColor3f(0.f,1.f,0.f); // Vert
87     glVertex3f( 1.f,-1.f, 1.f); //
88     glColor3f(0.f,0.f,1.f); // Bleu
89     glVertex3f( -1.f,-1.f, 1.f); //
90
91     glColor3f(0.f,0.f,1.f); // Bleu
92     glVertex3f( 1.f, 1.f, -1.f); //
93     glColor3f(1.f,0.f,0.f); // Rouge
94     glVertex3f( -1.f,-1.f, -1.f); //
95     glColor3f(0.f,1.f,0.f); // Green
96     glVertex3f( -1.f,1.f, -1.f); //
97
98     glEnd(); // Carré complet
99
00     SwapBuffers(m_hDC); // on affiche le tout à l'écran
01 }
```



Les deux triangles ont la même taille. Pourquoi ?

Notre projection est toujours orthographique. Passons à une perspective.

Dans « ResizeGLScene »

```
44  
45 GLvoid Form1::ReSizeGLScene(GLsizei width, GLsizei height)  
46 {  
47     glViewport(0,0,width,height);           // Re-initialise le Viewport  
48  
49     glMatrixMode(GL_PROJECTION);           // On active la matrice de projection  
50     glLoadIdentity();                     // La matrice de projection est remise à zero  
51  
52     // Calcul du rapport d'aspect  
53     gluPerspective(45.f,(GLfloat)width/(GLfloat)height,0.1f,100.f);  
54  
55     glMatrixMode(GL_MODELVIEW);           // On active la matrice modèle / vue  
56     glLoadIdentity();                     // La matrice modèle / vue est remise à zero  
57  
58     Render();  
59 }  
60
```

`gluPerspective(45.f,(GLfloat)width/(GLfloat)height,0.1f,100.f);`

- **fovy = 45.f** Ouverture de la camera (Field of view) ;
- **aspect = w/h** Rapport entre la largeur et la hauteur (Aspect ratio) ;
- **zNear = 0.1f** Plan rapproché (near plane) ;
- **zFar = 100.f** Plan éloigné (Far plane).



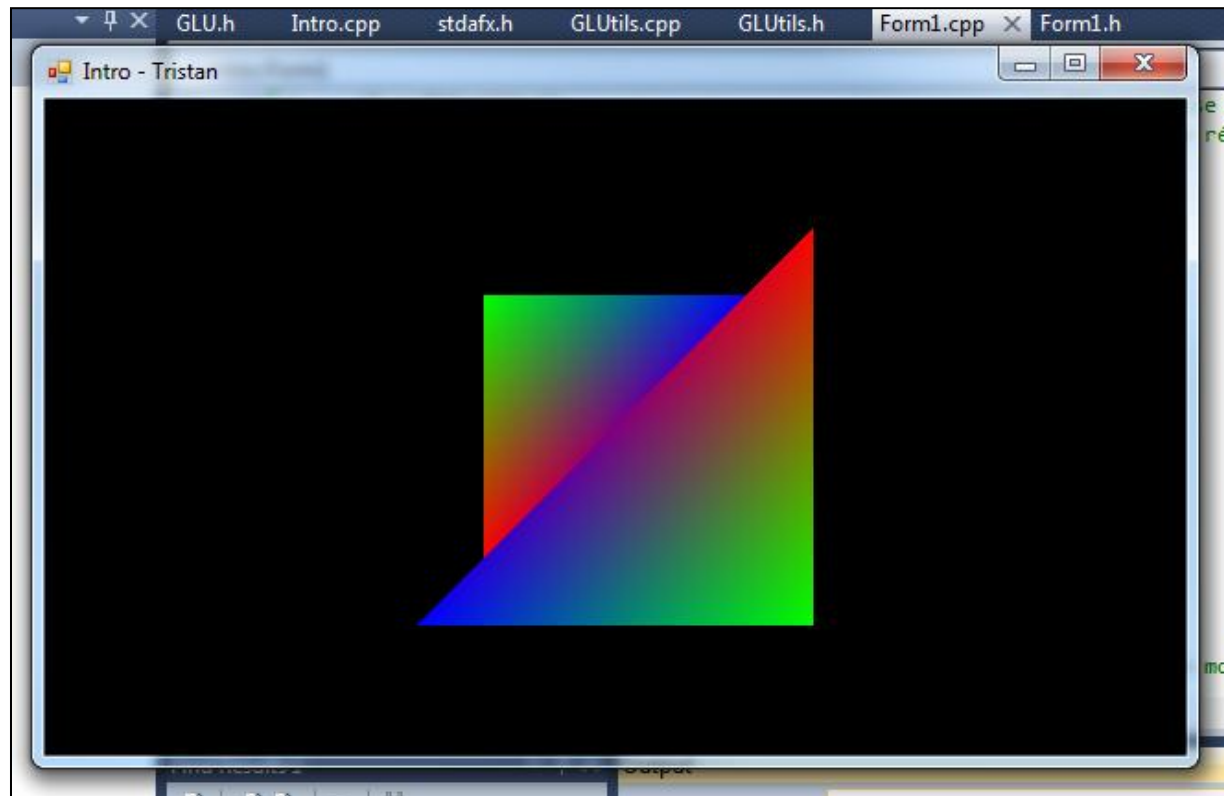
On ne voit qu'un seul triangle. Pourquoi ?

Notre caméra se trouve en $0.f, 0.f, 0.f$ et elle regarde vers les z négatifs, on ne voit que le deuxième triangle (composante $z = -1.f$).

Une autre modification est nécessaire. Reculer la scène.

Ajoutons une ligne à la méthode « Render() »

```
7 System::Void Form1::Render(System::Void)
8 {
9     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // Vide le tampon de couleur et de profondeur
10    glLoadIdentity(); // Réinitialise la matrice modèle/vue
11    glTranslatef(0.f,0.f,-5.f); // Déplace le référentiel de 5.0 dans l'écran
```

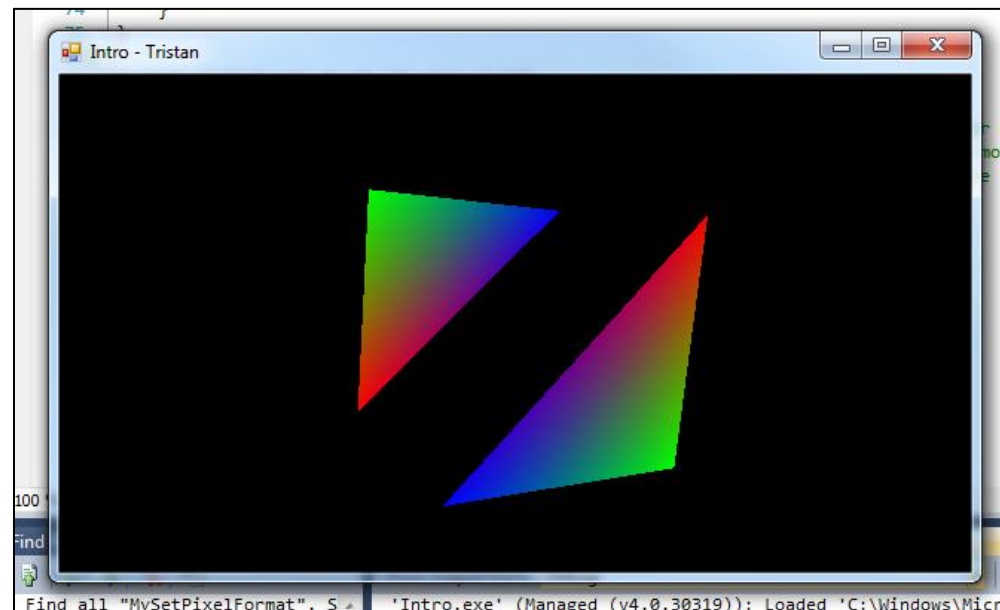


Nous voulons maintenant légèrement tourner notre scène, toujours dans « Render() », ajoutons :

```
7 System::Void Form1::Render(System::Void)
8 {
9     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // Vide le tampon de couleur et de profondeur
10    glLoadIdentity(); // Réinitialise la matrice modèle/vue
11    glTranslatef(0.f,0.f,-5.f); // Déplace le référentiel de 5.0 dans l'écran
12    glRotatef(30.f, 0.f,1.f,0.f);
13    glRotatef(10.f, 1.f,0.f,0.f);
```

glRotatef (GLfloat angle, GLfloat x, GLfloat y, GLfloat z);

- Premier paramètre : l'angle en degrés ;
- Les trois suivants : l'axe de rotation ;



Finalement, ajoutons le référentiel à l'origine. Dans « Render() »

```
77 System::Void Form1::Render(System::Void)
78 {
79     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // Vide le tampon de couleur et de profondeur
80     glLoadIdentity(); // Réinitialise la matrice modèle/vue
81     glTranslatef(0.f,0.f,-5.f); // Déplace le référentiel de 5.0 dans l'écran
82     glRotatef(30.f, 0.f,1.f,0.f);
83     glRotatef(10.f, 1.f,0.f,0.f);
84
85     glBegin(GL_LINES);
86         glColor3f(1.f,1.f,1.f);
87         glVertex3f( 0.f, 0.f, 0.f);
88         glVertex3f( 1.f, 0.f, 0.f);
89         glVertex3f( 0.f, 0.f, 0.f);
90         glVertex3f( 0.f, 1.f, 0.f);
91         glVertex3f( 0.f, 0.f, 0.f);
92         glVertex3f( 0.f, 0.f, 1.f);
93     glEnd();
94
95     glBegin(GL_TRIANGLES); // Début - en mode triangle
96
97     glColor3f(1.f,0.f,0.f); // Rouge
```

Notre scène est complète. Il ne manque que le lettre x, y, z.

Nous y reviendrons dans quelques semaines.