Examen

mai 2016 - Durée 2h

Calculette et documents non autorisés

Exercice 1 :

1. Qu'est ce qu'une image en mode RGB 24 bits ?
2. Quelle est la taille mémoire d'une image en mode RGB 24 bits de dimensions 1000×1000 ?
3. Une image en mode RGB 24 bits de dimensions 1000×1000 est stockée au format JPEG. La taille du fichier JPEG est 200000 octets. A quel taux de compression correspond cette taille de fichier ?

Exercice 2 :
Pour une image en niveaux de gris :

1. quelle mesure statistique donne une information sur la luminosité globale de l'image ?
2. quelle mesure statistique donne une information sur le contraste global de l'image ?

Exercice 3 :
Expliquez ce qu'est une opération de ***seuillage***.

Exercice 4 :
En figure 4.1, on a une image de dimensions 3×2 avec les valeurs des différents pixels. On effectue un agrandissement pour obtenir une image de dimensions 9×6 (figure 4.2).

	x=0	x=1	x=2
y=0	0	200	160
y=1	80	120	240

figure 4.1

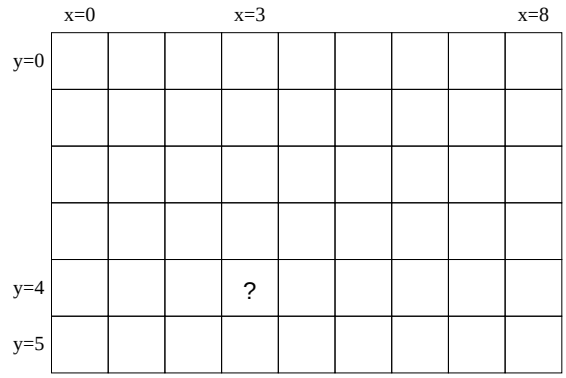


figure 4.2

En utilisant l'interpolation bilinéaire, quelle est la valeur du pixel d'indice $(x = 3, y = 4)$?
Expliquez comment vous calculez la valeur de ce pixel.

Exercice 5 :

Complétez le code C++ suivant

```
// image en niveaux de gris
// valeurs des pixels entre 0.0 et 1.0
ImageGrisF I(200,200);

// PARTIE A COMPLETER //

// afficher l'image à l'écran
I.afficher();
```

afin d'obtenir l'image ci-contre de dimensions 200×200 représentant un dégradé du noir (en haut à gauche de l'image) au blanc (en bas à droite de l'image).

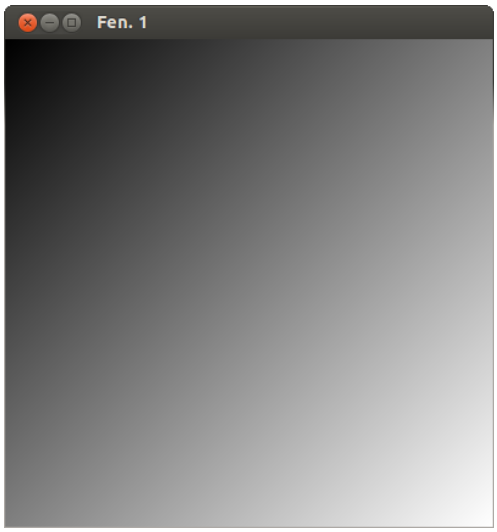


figure 5.1

Exercice 6 :

Déterminez la fonction affine permettant de corriger l'image de la figure 6.1 en l'image de la figure 6.2 (pour chaque image, l'histogramme correspondant est affiché). Expliquez comment vous déterminez cette fonction affine.

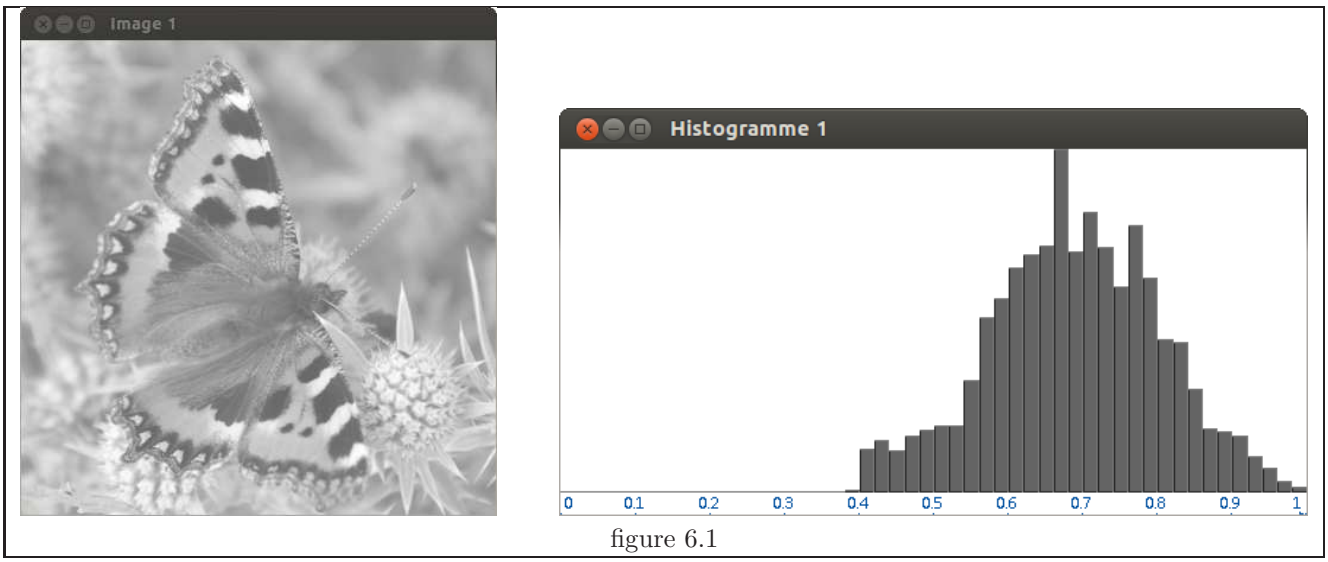


figure 6.1

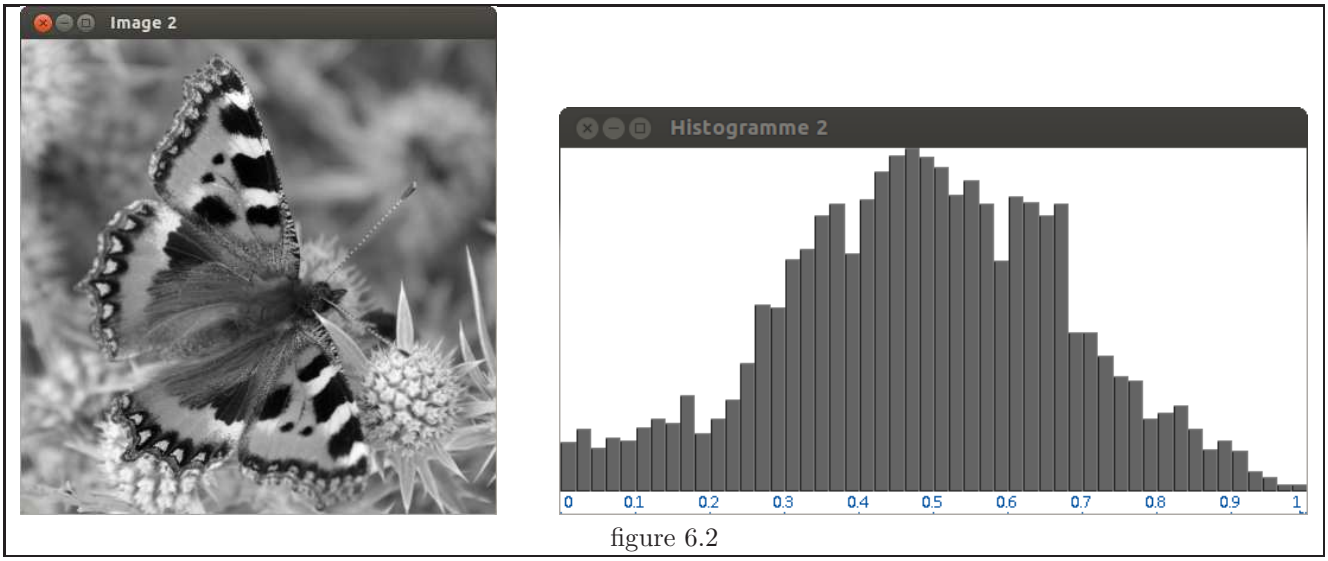
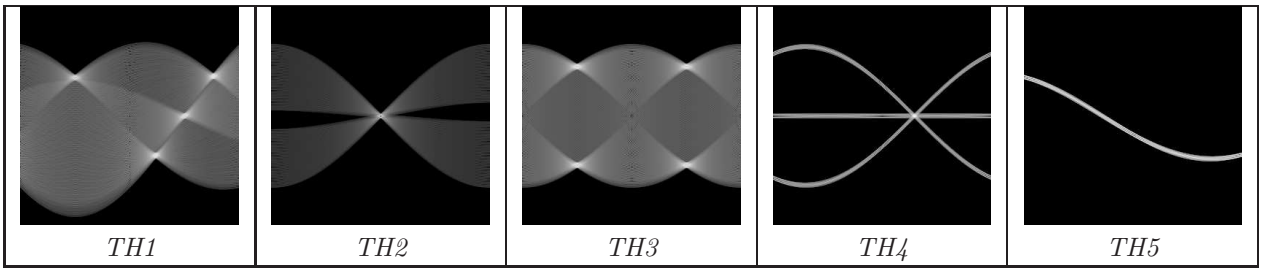
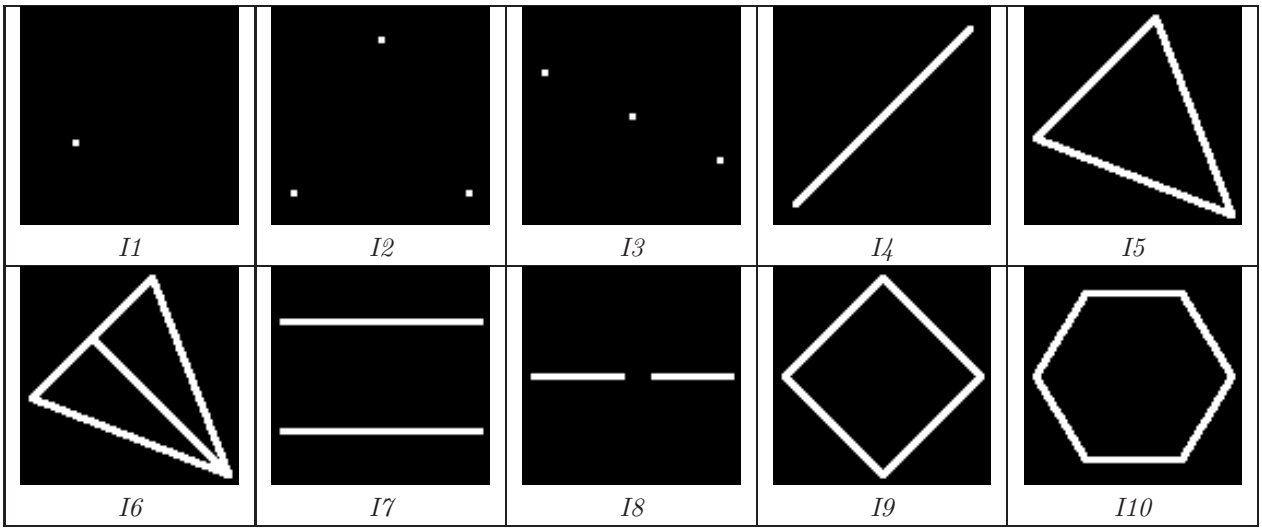


figure 6.2

Exercice 7 :
Pour chacune des 5 images *TH1* à *TH5* représentant une transformée de Hough, associez parmi les 10 images *I1* à *I10* l'image correspondante.



TH1 | *TH2* | *TH3* | *TH4* | *TH5*



I1 | *I2* | *I3* | *I4* | *I5*

I6 | *I7* | *I8* | *I9* | *I10*

Exercice 8 :
Soit le filtre linéaire dont le noyau K correspond aux instructions C++ suivantes :

```
// noyau du filtre
float K[3][3] = {
 {1.0/9.0, 1.0/9.0, 1.0/9.0},
 {1.0/9.0, 1.0/9.0, 1.0/9.0},
 {1.0/9.0, 1.0/9.0, 1.0/9.0}
};
```

Quel est le nom de ce filtre, et quel est son effet lors qu'on l'applique à une image ?

Exercice 9 :
Les instructions suivantes correspondent à une partie d'un programme C++ utilisant la librairie OpenGL :

```
1  ////////////////////////////////////////////////////////////////////////
2  // Rappel des ordres graphiques :
3  //  glVertex2d(x,y) équivalent à glVertex3d(x,y,0.0) (coordonnée z nulle)
4  //  glPushMatrix();          // début d'une transformation géométrique
5  //  glPopMatrix();           // fin d'une transformation géométrique
6  //  glTranslated(Vx,Vy,Vz); // translation de vecteur (Vx,Vy,Vz)
7  //  glScaled(hx,hy,hz);      // homothétie de rapport hx,hy,hz (respectivement en x,y,z)
8  //  glRotated(a,0,0,1);      // rotation d'angle a (en degrés) dans le plan Oxy
9                               //   (autour du vecteur (0,0,1) )
10
11 // tracé d'un cercle de centre (0,0) et rayon 1 - cf figure 10-1
12 void cercle()
13 {
14     int N=100;
15     glBegin(GL_POLYGON);
16         for (int i=0; i<N; i++)
17         {
18             GLdouble a = 2.0*M_PI*(GLdouble)i/(GLdouble)N;
19             glVertex2d( cos(a), sin(a));
20         }
21     glEnd();
22 }
23
24 // tracé d'un carré de sommets (1,1),(-1,1),(-1,-1),(1,-1)
25 void carre()
26 {
27     // PARTIE A COMPLETER - cf figure 10-2
28 }
29
30 // tracé d'un demi-cercle de centre (0,0) et rayon 1
31 // au-dessus de l'axe horizontal
32 void demi_cercle()
33 {
34     // PARTIE A COMPLETER - cf figure 10-3
35 }
36
37 void dessin()
38 {
39     glMatrixMode(GL_MODELVIEW);
40     glLoadIdentity();
41
42     glClearColor(1.0,1.0,1.0,0.0); // fond blanc
43     glClear(GL_COLOR_BUFFER_BIT);
44
45     // dessin de la figure dans le plan z=0
46     glColor3d(0.0, 0.0, 0.0);
47
48     // PARTIE A COMPLETER - cf figures 10-4 et 10-5
49
50     glFlush();
51 }
52
53 ////////////////////////////////////////////////////////////////////////
54 // prise en compte d'un redimensionnement de la fenetre
55 void redimensionnement(int w, int h)
56 {
57     ...
58     glOrtho(xmin,xmax,ymin,ymax,zmin,zmax);
59     ...
60 }
61
62 ...
```

1. Complétez les fonctions `carre` (ligne 27) et `demi_cercle` (ligne 34) afin de représenter dans le plan $z = 0$ les objets correspondant aux figures 9-2 et 9-3.

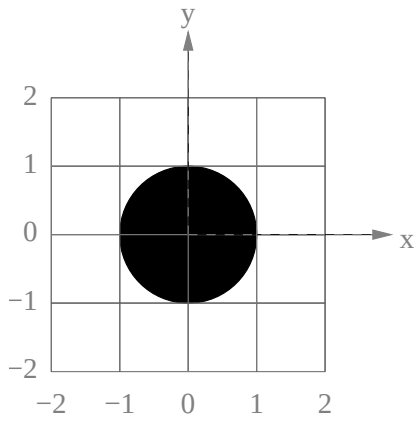


figure 9-1

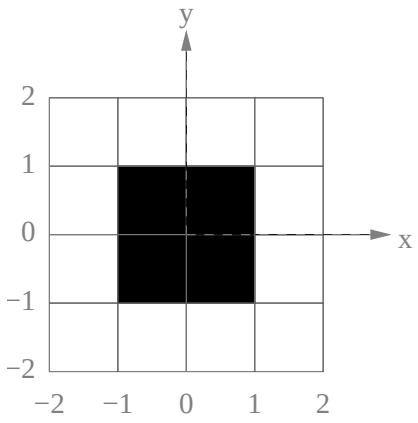


figure 9-2

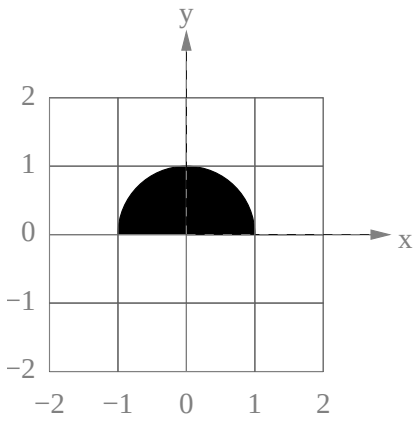


figure 9-3

2. Complétez la fonction `dessin` (ligne 48) afin de dessiner la scène représentée en figure 9-4 ci-dessous (la figure 9-5 vous permet d'avoir les coordonnées de la scène dans le plan Oxy)

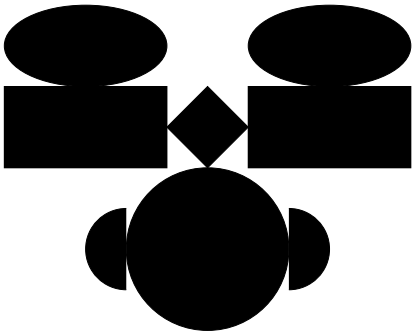


figure 9-4

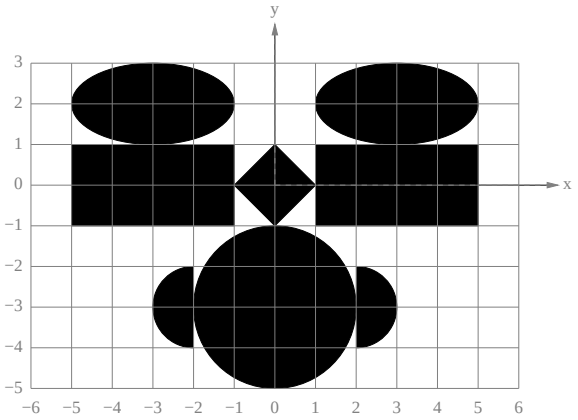


figure 9-5

3. La fenêtre graphique a pour dimension largeur= 500 et hauteur= 500.
Par quelles valeurs faut-il remplacer `xmin`, `xmax`, `ymin`, `ymax`, `zmax`, `zmax` dans l'appel de la fonction `glOrtho` (ligne 58) afin de voir l'ensemble de la scène, que celle-ci soit centrée dans la fenêtre, et en respectant le rapport d'échelle entre **x** et **y** (cf. figure 9.6 ci-dessous) ?

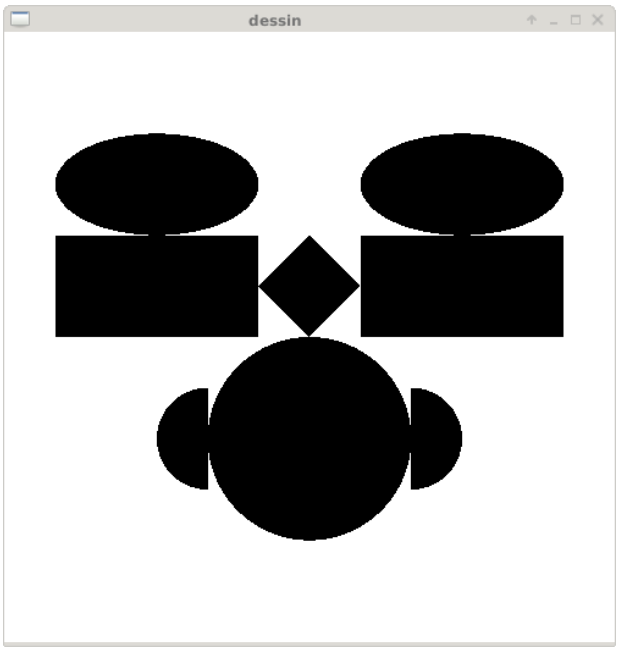


figure 9-6

Corrigé de l'examen

mai 2015 - Durée 2h

Calculette et documents non autorisés

Exercice 1 :

1. image couleur avec 3 composantes Rouge, Vert, Bleu, chaque composante est codée sur 8 bits (1 octet) soit 256 valeurs possibles par composante.
2. 1 pixel $\equiv$ 3 octets $\Rightarrow$ image $1000 \times 1000 \equiv 3000000$ octets $= 3$ Mo
3. taux de compression $= 3000000/200000 = 15$.

Exercice 2 :

1. il s'agit de la moyenne.
2. il s'agit de l'écart-type.

Exercice 3 :
Opération sur une image $I = (I_k)$ en niveaux de gris qui consiste à calculer une image $S = (S_k)$ en noir & blanc telle que $S_k = 1$ si et ssi $I_k > s$ avec s seuil.

Exercice 4 :
Image I de dim. $3 \times 2 \rightarrow$ image $\bar{I}$ de dim. 9×6. La valeur de $\bar{I}_{3,4}$ est 118.

	x=0			x=3	x=4				x=8
y=0	0			150	200				160
y=4	64			**118**	136				
y=5	80			110	120				240

$\bar{I}_{4x,5y} = I_{x,y}$.
Puis on applique l'interpolation linéaire en x :

$$\begin{cases} \bar{I}_{3,0} = \frac{1}{4}\bar{I}_{0,0} + \frac{3}{4}\bar{I}_{4,0} = 150 \\ \bar{I}_{3,5} = \frac{1}{4}\bar{I}_{0,5} + \frac{3}{4}\bar{I}_{4,5} = 110 \end{cases}$$

et l'interpolation linéaire en y :

$$\bar{I}_{3,4} = \frac{1}{5}\bar{I}_{3,0} + \frac{4}{5}\bar{I}_{3,5} = 118$$

Exercice 5 :

```
for (int x = 0; x<200; x++)
for (int y = 0; y<200; y++)
    I(x,y) = (float)(x+y)/398.0;
```

Exercice 6 :
La fonction est de la forme $f(x) = a\,x + b$:

$$f(0,4) = 0 \text{ et } f(1) = 1 \Rightarrow f(x) = \frac{5}{3}x - \frac{2}{3}$$

Exercice 7 :
$TH1 \leftrightarrow I6$ $\quad TH2 \leftrightarrow I8$ $\quad TH3 \leftrightarrow I9$ $\quad TH4 \leftrightarrow I3$ $\quad TH5 \leftrightarrow I1$

Exercice 8 :
ce filtre est le *filtre moyenne* de taille 3, c'est un filtre *moyenneur* qui va flouter (lisser) l'image.

Exercice 9 :

1.
```
// tracé d'un carré de sommets (1,1),(-1,1),(-1,-1),(1,-1)
void carre()
{
    glColor3d(0.0, 0.0, 0.0);
    glBegin(GL_POLYGON);
        glVertex2d( 1.0, 1.0); glVertex2d(-1.0, 1.0);
        glVertex2d(-1.0,-1.0); glVertex2d( 1.0,-1.0);
    glEnd();
}

// tracé d'un demi-cercle de centre (0,0) et rayon 1
// au-dessus de l'axe horizontal
void demi_cercle()
{
    int N=100;
    glColor3d(0.0, 0.0, 0.0);
    glBegin(GL_POLYGON);
        for (int i=0; i<=N; i++)
        {
            GLdouble a = M_PI*(GLdouble)i/(GLdouble)N;
            glVertex2d( cos(a), sin(a));
        }
    glEnd();
}
```

```
    // dessin de la figure dans le plan z=0
    glPushMatrix();
        glTranslated(3,0,0); glScaled(2,1,1);
        carre();
    glPopMatrix();
    glPushMatrix();
        glTranslated(3,2,0); glScaled(2,1,1);
        cercle();
    glPopMatrix();
    glPushMatrix();
        glTranslated(-3,0,0); glScaled(2,1,1);
        carre();
    glPopMatrix();
    glPushMatrix();
        glTranslated(-3,2,0); glScaled(2,1,1);
        cercle();
    glPopMatrix();
    glPushMatrix();
        glRotated(45,0,0,1); glScaled(sqrt(0.5),sqrt(0.5),1);
        carre();
    glPopMatrix();
    glPushMatrix();
        glTranslated(0,-3,0); glScaled(2,2,1);
        cercle();
    glPopMatrix();
    glPushMatrix();
        glTranslated(-2,-3,0); glRotated(90,0,0,1);
        demi_cercle();
    glPopMatrix();
    glPushMatrix();
        glTranslated( 2,-3,0); glRotated(-90,0,0,1);
        demi_cercle();
    glPopMatrix();
```

3. Par exemple :

```
        glOrtho(-6.0      , 6.0,          // xmin , xmax
                -6.0-1.0  , 6.0-1.0,      // ymin , ymax
                -1.0      , 1.0);         // zmin , zmax
```