

BASES OPENGL

35

Syntaxe OpenGL

Types GL

GLtype

ex : GLint, GLfloat

Fonctions GL

glFonction [NB] [Type] ()

ex : glMatrixMode()

glColor3f(1.0, 1.0, 1.0)

36

3/27/20
S. Lanquetin – L3 Info, Synthèse d'images
37

Les Couleurs

Variable d'état	2 modes de couleurs	RVBA
<ul style="list-style-type: none"> modifiée avec glColor*() couleur courante 	<ul style="list-style-type: none"> RVBA : glColor*() Indexé : glIndex*() 	<ul style="list-style-type: none"> 3 valeurs pour la couleur 1 (optionnelle) pour l'opacité = 1 par défaut

```
glColor4f(0.0,0.0,0.0,1.0); //Noir
glColor3f(1.0,0.0,0.0); //Rouge
glColor3f(1.0,1.0,1.0); //Blanc
```

37

3/27/20
S. Lanquetin – L3 Info, Synthèse d'images
38

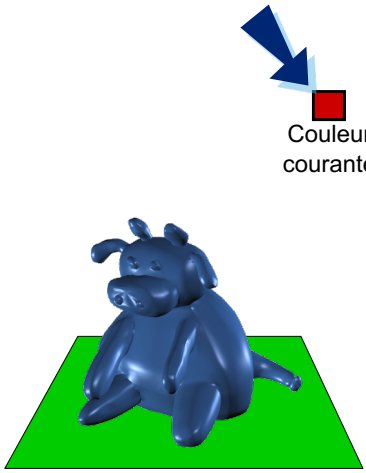
Exemple

```
// couleur courante fixée à bleu
glColor3f( 0.0f, 0.0f, 1.0f );

// vache
afficherVache();

// couleur courante fixée à vert
glColor3f( 0.0f, 1.0f, 0.0f );

// sol
afficherSol();
```



Couleur courante

38

3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

39

La couleur de fond

Détermination de la couleur de fond

```
void glClearColor(GLfloat, GLfloat, GLfloat, GLfloat)
```

Ex : `glClearColor(0.0, 0.0, 0.0, 1.0) ; // couleur noire`

Affectation à un variable d'état

Utilisation de cette couleur

```
glClear(GL_COLOR_BUFFER_BIT)
```

39

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

40

OpenGL : Dessin

Un Point



Un sommet



Un sommet



Un Segment



40

OpenGL : Dessin

Délimiter un tracé

```
glBegin(mode);
glEnd();
```

Définir un sommet : syntaxe

```
glVertex[{234}]{sifd}(coords);
```

Nombre de paramètres

Types de paramètres

Valeurs par défaut

Valeur de z par défaut : 0

Valeur de w par défaut : 1

41

OpenGL : Dessin - Exemple

➡ Ex : définir un sommets

```
glVertex3f(1.1, 1.0, 3.5); //S(1.1, 1.0, 3.5)
```



Ce sommet ne sera pas affiché

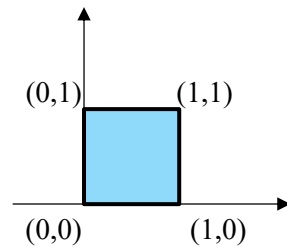
➡ Ex : Afficher un sommet

```
glBegin (GL_POINTS);
glVertex3f(1.1, 1.0, 3.5);
glEnd();
```

42

OpenGL : Dessin

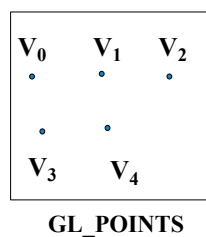
➔ Ex : Afficher un quadrilatère



```
glBegin (GL_POLYGON);  
glVertex2d(0,0)  
glVertex2d(1,0);  
glVertex2d(1,1);  
glVertex2d(0,1);  
glEnd();
```

43

OpenGL : Points



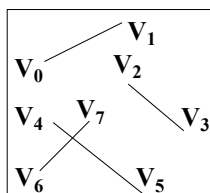
44

3/26/20

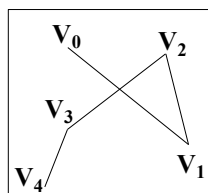
S. Lanquetin – L3 Info, Synthèse d'images

45

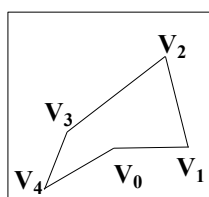
OpenGL : Lignes



GL_LINES



GL_LINE_STRIP



GL_LINE_LOOP

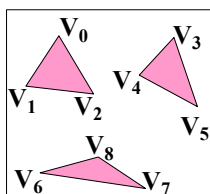
45

3/26/20

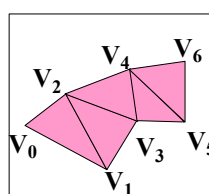
S. Lanquetin – L3 Info, Synthèse d'images

46

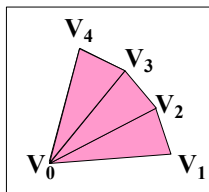
OpenGL : Triangles



GL_TRIANGLES



GL_TRIANGLE_STRIP



GL_TRIANGLE_FAN

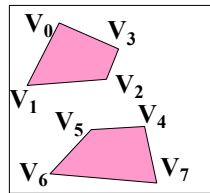
46

3/26/20

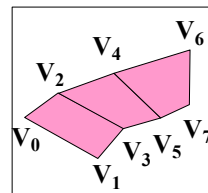
S. Lanquetin – L3 Info, Synthèse d'images

47

OpenGL : Quadrilatères



GL_QUADS



GL_QUAD_STRIP

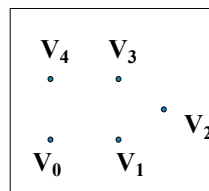
47

3/26/20

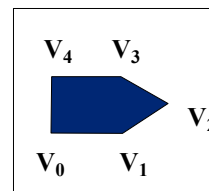
S. Lanquetin – L3 Info, Synthèse d'images

48

OpenGL : Polygone



GL_POINTS



GL_POLYGON

48

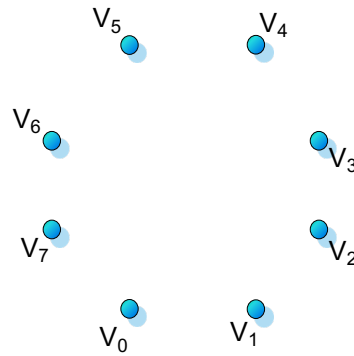
3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

49

OpenGL : glBegin() & glEnd()

```
glBegin (GL_POINTS);
glVertex2d(V0);
glVertex2d(V1);
glVertex2d(V2);
glVertex2d(V3);
glVertex2d(V4);
glVertex2d(V5);
glVertex2d(V6);
glVertex2d(V7);
glEnd();
```



49

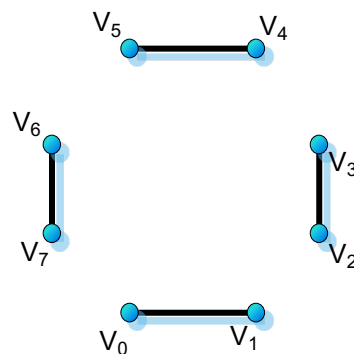
3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

50

OpenGL : glBegin() & glEnd()

```
glBegin (GL_LINES);
glVertex2d(V0);
glVertex2d(V1);
glVertex2d(V2);
glVertex2d(V3);
glVertex2d(V4);
glVertex2d(V5);
glVertex2d(V6);
glVertex2d(V7);
glEnd();
```



50

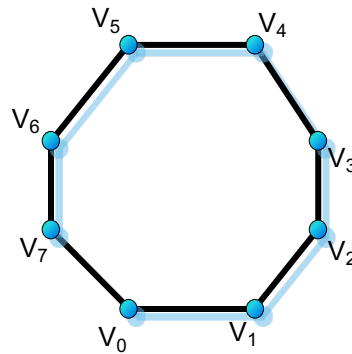
3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

51

glBegin() & glEnd()

```
glBegin (GL_LINE_LOOP);
glVertex2d(V0);
glVertex2d(V1);
glVertex2d(V2);
glVertex2d(V3);
glVertex2d(V4);
glVertex2d(V5);
glVertex2d(V6);
glVertex2d(V7);
glEnd();
```



51

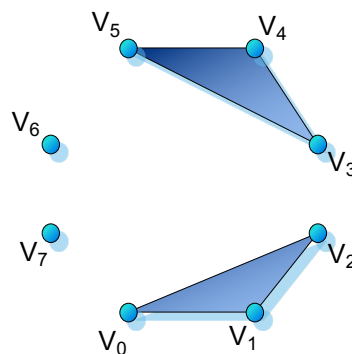
3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

52

glBegin() & glEnd()

```
glBegin (GL_TRIANGLES);
glVertex2d(V0);
glVertex2d(V1);
glVertex2d(V2);
glVertex2d(V3);
glVertex2d(V4);
glVertex2d(V5);
glVertex2d(V6);
glVertex2d(V7);
glEnd();
```



52

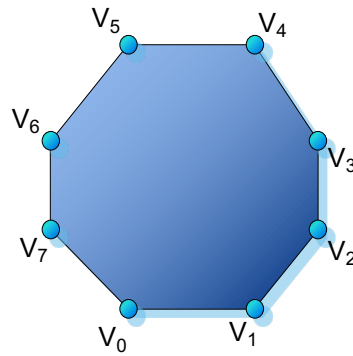
3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

53

OpenGL : glBegin() & glEnd()

```
glBegin (GL_POLYGON);
glVertex2d(V0);
glVertex2d(V1);
glVertex2d(V2);
glVertex2d(V3);
glVertex2d(V4);
glVertex2d(V5);
glVertex2d(V6);
glVertex2d(V7);
glEnd();
```



53

3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

54

OpenGL : Que dessine cet exemple ?



Exemple

```
#define PI 3.141592
#define n 20
...
glBegin(GL_LINE_LOOP);
for(int i=0;i<n;i++)
{
    angle = 2.0*PI*i/n;
    glVertex2f(cos(angle),sin(angle));
}
glEnd();
...
```

54

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

55

OpenGL : Polygones

→ Un polygone peut être dessiné en mode

- ☐ Fil de fer **GL_LINE**
- ☐ Rempli avec la couleur courante **GL_FILL**

→ Un polygone possède deux faces

- ☐ Avant **GL_FRONT**
- ☐ Arrière **GL_BACK**

GL_FRONT_AND_BACK

55

55

3/26/20

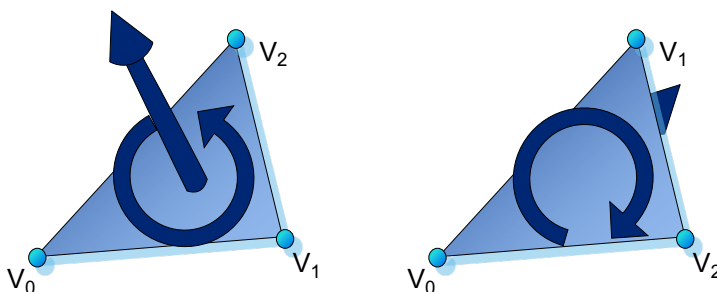
S. Lanquetin – L3 Info, Synthèse d'images

56

Faces avant / arrière ?

Ordre trigo ⇔ Face avant

Ordre "Clockwise" ⇔ Face arrière



56

3/30/20

S. Lanquetin – L3 Info, Synthèse d'images

57

OpenGL : Polygones

glPolygonMode(face, mode);

- ➔ Contrôle le mode de dessin des **faces avant et arrière** d'un polygone
- ➔ face : `GL_FRONT`, `GL_BACK`, `GL_FRONT_AND_BACK`
- ➔ mode : `GL_POINT`, `GL_LINE`, `GL_FILL`

glCullFace (GL_BACK);

- ➔ Désigner des faces non visibles (l'arrière)

glDisable (GL_CULL_FACE);

- ➔ Les tracer

glEnable (GL_CULL_FACE);

- ➔ Ne pas les tracer

57

3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

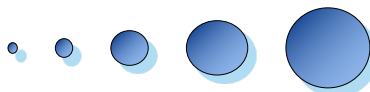
58

OpenGL : Taille

➔ Contrôler la taille d'un point

- ☐ taille par défaut 1.0
- ☐ représentation par un carré

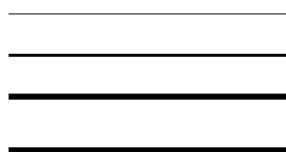
void glPointSize(GLfloat Size);



➔ Contrôler la largeur d'une ligne

- ☐ largeur par défaut 1.0

void glLineWidth (GLfloat Size);



➔ Lisser (modèle de Gouraud)

glShadeModel(GL_FLAT) ; surface avec facettes

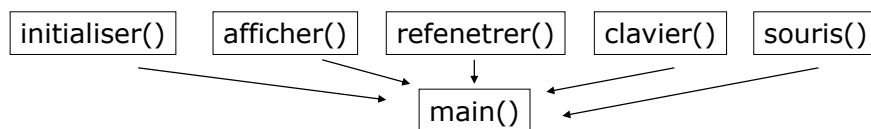
glShadeModel(GL_SMOOTH); surface lisse (dégradé de couleur)

58

OpenGL : Programme type

Un programme est généralement constitué de 5 parties :

1. Une fonction d'initialisation (void initialiser()).
2. Une fonction d'affichage (void afficher()).
3. Une fonction de fenêtrage (void refenêtrer()).
4. Une fonction de gestion de clavier (void clavier()).
5. Une fonction de gestion de la souris (void souris()).
6. La partie principale (main).



59

Initialiser()

```
void Initialiser()
{
    glClearColor(0,0,0,0);
    glColor3f(1.0,2.0,3.0);
}
```

60

60

3/27/20

S. Lanquetin – L3 Info, Synthèse d'images

61

afficher()

```
void afficher()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_POLYGON)
        glVertex3f(10.0,12.0,15.0) ;
    ....
    glEnd();
    glutSwapBuffer(); //ou glFlush();
}
```

61

61

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

62

refenetrer()

```
void refenetrer(...)
{
}
```

62

62

3/27/20

S. Lanquetin – L3 Info, Synthèse d' images

63

GLUT : Actualisation Affichage

```

void clavier(...)
{
    switch(touche){
        case 27 :
            exit(0) ;
        case 'a' : case 'A' :
            ....
            glutPostRedisplay();
            break;
    }
}

```

63

3/26/20

S. Lanquetin – L3 Info, Synthèse d' images

64

souris()

```

void souris(int btn, int state, int x, int y)
{
    switch(bouton){
        case GLUT_LEFT_BUTTON :
            if (etat==GLUT_DOWN)
                ....
            break ;
        case GLUT_MIDDLE_BUTTON :
            if (etat==GLUT_DOWN)
                ....
            break ;
        case GLUT_RIGHT_BUTTON :
            if (etat==GLUT_DOWN)
                ....
            break ;
    }
}

```

64

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

65

main(int argc, char** argv)

```
void main(int argc, char** argv)
{
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_RGB | GLUT_DEPTH | GLUT_DOUBLE);
    glutInitWindowSize(500,500) ; //taille initiale de la fenetre
    glutInitWindowPosition(200,200); //position de la fenetre
    glutCreateWindow(argv[0]) ; //création de la fenetre
    Init() ;
    glutDisplayFunc(afficher);
    glutDisplayFunc(refenetrer);
    glutDisplayFunc(clavier);
    glutDisplayFunc(souris);
    glutMainLoop(); //lancement de la boucle principale
    return (0);
}
```

65

65

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

66

LES MATRICES

openGL

66

3/30/20

S. Lanquetin – L3 Info, Synthèse d'images

67

OpenGL : Matrices

Sélection de la matrice active :

`glMatrixMode(...)`

`GL_MODELVIEW` `GL_PROJECTION` `GL_TEXTURE`

Initialisation de la matrice active :

`glLoadIdentity()`

Modification de la matrice active :

`glLoadMatrix(m), glMultMatrix(p)`

67

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

68

`glLoadMatrix(m)`

- Tableau `m[0],...m[15]`

$$MatC = \begin{bmatrix} m[0] & m[4] & m[8] & m[12] \\ m[1] & m[5] & m[9] & m[13] \\ m[2] & m[6] & m[10] & m[14] \\ m[3] & m[7] & m[11] & m[15] \end{bmatrix}$$

68

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

69

glMultMatrix(m)

- Tableau m[0],...m[15]

$$MatC = MatC \times \begin{bmatrix} m[0] & m[4] & m[8] & m[12] \\ m[1] & m[5] & m[9] & m[13] \\ m[2] & m[6] & m[10] & m[14] \\ m[3] & m[7] & m[11] & m[15] \end{bmatrix}$$

69

3/30/20

S. Lanquetin – L3 Info, Synthèse d'images

70

OpenGL : Matrices

Règle :

Définir les transformations dans l'ordre inverse

Exemple :

```
glMatrixMode( GL_MODELVIEW);
glLoadIdentity();
glMultMatrixf(N);
glMultMatrixf(M);
glMultMatrixf(L);
glBegin(GL_POINTS);
    glVertex3f(v);
glEnd();
```

L est appliqué en premier : N(M(Lv))

70

3/30/20

S. Lanquetin – L3 Info, Synthèse d'images

71

OpenGL : Matrices

Gestion de la pile des matrices :

`glPushMatrix()` empiler
`glPopMatrix()` dépiler

71

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

72

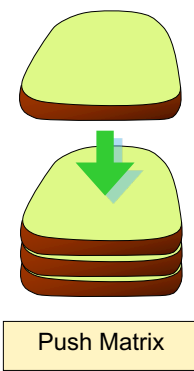
Matrice Push / Pop

`void glPushMatrix(void);`

“rappelle-toi où tu es”

Matrice du haut copiée :

*Contenu 1ère matrice (matrice courante)
 =contenu 2ème matrice*



72

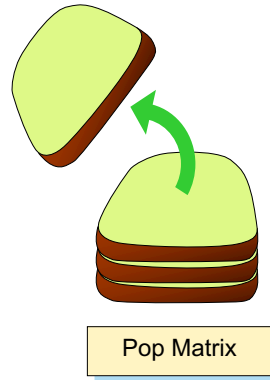
3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

73

Matrice Push / Pop

*void **glPopMatrix**(void);*
“retourne où tu étais”



73

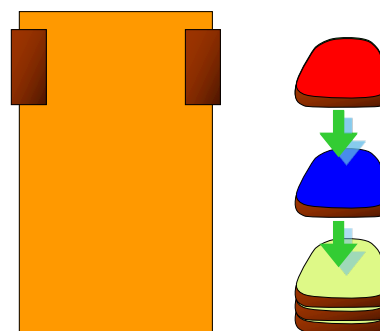
3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

74

Matrice Push / Pop

```
affichageVoiture()
{
    glTranslatef( c_x, c_y, c_z);
    affichageCarrosserie();
    glPushMatrix();
    glTranslate( f_x, f_y, f_z);
    glPushMatrix();
    glTranslate( l_x, l_y, l_z);
    affichagePneu();
    glPopMatrix();
    glPushMatrix();
    glTranslate( r_x, r_y, r_z);
    affichagePneu();
    glPopMatrix();
    ...
}
```



74

3/30/20

S. Lanquetin – L3 Info, Synthèse d'images

75

OpenGL : Transformations

Rotation :

```
glRotatef(angle, axe_x, axe_y, axe_z);
```

Translation :

```
glTranslatef(t_x, t_y, t_z);
```

Changement d'échelle :

```
glScalef(s_x, s_y, s_z);
```

Repère de l'œil :

```
gluLookAt( ... );
```

75

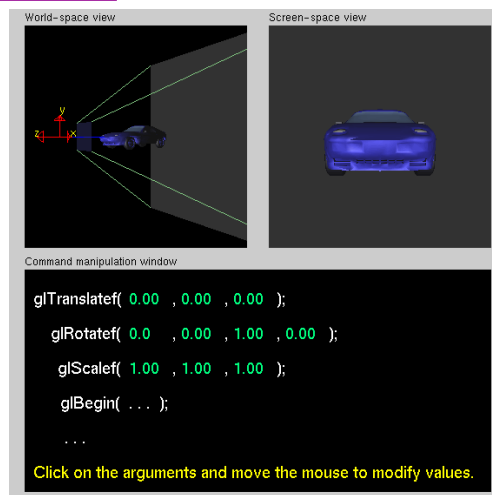
3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

76

Transformations

http://users.polytech.unice.fr/~buffa/cours/synthese_image/DOCS/www.xmission.com/Nate/tutors.html



76

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

77

INTERACTION

77

3/30/20

S. Lanquetin – L3 Info, Synthèse d'images

78

Clavier

- Modifier des variables dans **clavier()**
- Utiliser ces variables dans **affichage()**
- Mettre à jour avec **glutPostRedisplay()**

```
void clavier()
{
    switch(touche){
        case 27 :
            exit(0);
        case 'a' : case 'A' :
            ...
            glutPostRedisplay( );
            break;
    }
```

78

3/30/20

S. Lanquetin – L3 Info, Synthèse d'images

79

Souris

- Modifier des variables dans **souris()**
- Utiliser ces variables dans **affichage()**
- Mettre à jour avec **glutPostRedisplay()**

```
void souris(int btn, int state, int x, int y)
{
    switch(bouton){
        case GLUT_LEFT_BUTTON :
            if (etat==GLUT_DOWN)
                ....
                glutPostRedisplay( );
            break ;
    }
}
```

79

3/30/20

S. Lanquetin – L3 Info, Synthèse d'images

80

Animation

- Modifier des variables dans **anim1()**
- Utiliser ces variables dans **affichage()**
- Mettre à jour avec **glutPostRedisplay()**
- Appeler la fonction de rappel **glutIdleFunc (anim1)**

```
GLfloat angle=0;
void anim1( )
{
    angle+=2.0;
    if(angle>360)
        angle-=360;
    glutPostRedisplay( );
}
```

80

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

81

VISUALISATION

81

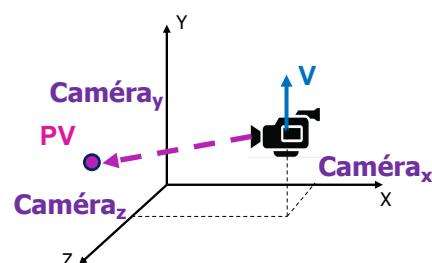
3/30/20

S. Lanquetin – L3 Info, Synthèse d'images

82

Position de la caméra

```
gluLookAt( Camérax, Caméray, Caméraz,
            PVx, PVy, PVz,
            Vx, Vy, Vz );
```



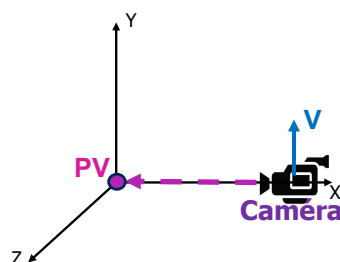
82

3/30/20

S. Lanquetin – L3 Info, Synthèse d'images

83

Position de la caméra



```
void affichage(){
    glMatrixMode( GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt( 0, 0, 5, 0, 0, 0, 0, 1, 0 );
    glScalef(1.0,2.0,1.0);
    ...//dessin de la scène
    glutSwapBuffers();
}
```

83

3/30/20

S. Lanquetin – L3 Info, Synthèse d'images

84

Transformations de projection

2 types de projection :

- perspective `glFrustum(...)`, `gluPerspective(...)`.
- parallèle `glOrtho(...)`

Objectif : définir un volume de vision

Passer en mode projection :

```
glMatrixMode(GL_PROJECTION);
```

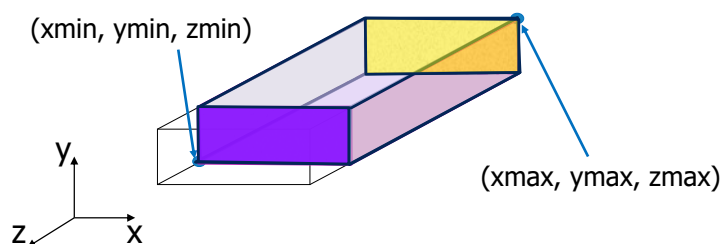
84

3/30/20

S. Lanquetin – L3 Info, Synthèse d'images

85

Transformations de projection : Parallèle



```
glOrtho(xmin, xmax, ymin, ymax, zmin, zmax)
```

```
glOrtho(gauche, droit, bas, haut, proche, eloigne);
```

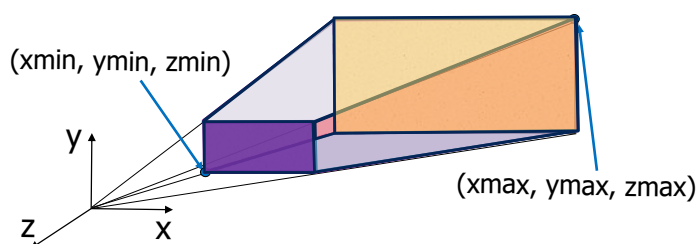
85

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

86

Transformations de projection : Perspective

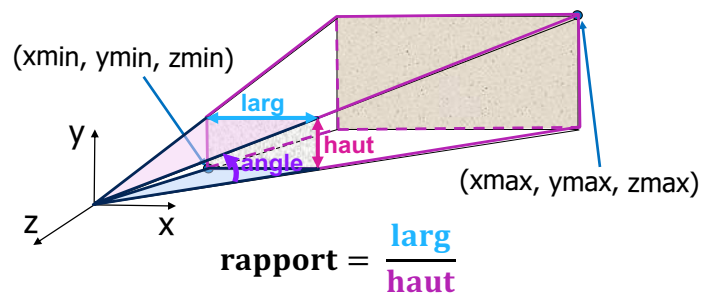


```
glFrustum(xmin, xmax, ymin, ymax, zmin, zmax)
```

```
glFrustum(gauche, droit, bas, haut, proche, eloigne)
```

86

Transformations de projection : Perspective



```
glFrustum( xmin, xmax, ymin, ymax, zmin, zmax)
```

```
glFrustum( gauche, droit, bas, haut, proche, eloigne)
```

```
gluPerspective(angle, rapport, zmin, zmax);
```

87

Transformations de projection : exemple

- Projection parallèle

```
glMatrixMode( GL_PROJECTION);
```

```
glLoadIdentity();
```

```
glOrtho(gauche, droit, bas, haut, proche, eloigne);
```

- Projection perspective

```
glMatrixMode( GL_PROJECTION);
```

```
glLoadIdentity();
```

```
gluPerspective(angle, rapport, zmin, zmax);
```

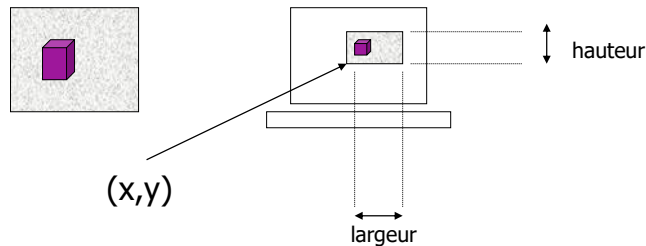
```
//glFrustum(gauche, droit, bas, haut, proche, eloigne);
```

88

Cadrage



Fenêtre d'affichage \neq cadre



glViewport(GLint x, GLint y, GLsizei largeur, GLsizei hauteur)



Conserver la ratio largeur / hauteur
pour image non déformée

89

Cadrage

- Adapter la projection de la scène
- Fonction de rappel **glutReshapeFunc(refenetrer);**

```
void refenetrer(int w, int h)
{ /* glViewport(0, 0, w, h) ; //taille du cadre d'affichage dans la fenêtre

  glMatrixMode(GL_PROJECTION) ;
  glLoadIdentity() ;
  gluPerspective(90., (float)w/(float)h, 5, 20) ;
  glMatrixMode(GL_MODELVIEW) ;
}
```

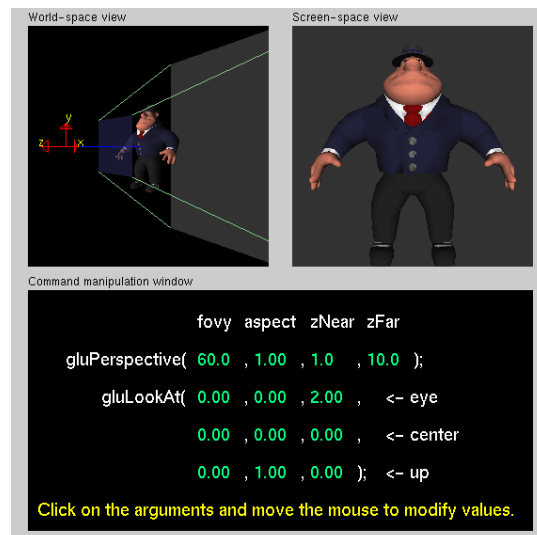
90

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

91

Projections



91

3/26/20

S. Lanquetin – L3 Info, Synthèse d'images

92

LISTES D'AFFICHAGE

OpenGL

92

Liste d'affichage

- Pré-compilation d'un groupe de commandes OpenGL
 - ➔ Objet dessiné plusieurs fois
 - ➔ Objet se déplace (animation)
- Commandes stockées et exécutées plus tard dans l'ordre de création
- Optimise les performances d'affichage
 - Affichage en une fois tout le contenu de la liste

93

Liste d'affichage : création

glNewList(numListe, mode) / glEndList()

- ➔ créer une liste d'affichage
- ➔ **numlist** entier de type **Guint**
- ➔ **mode** = **GL_COMPILE** ou **GL_COMPILE_AND_EXECUTE**

S :

```
Guint roue = 0 ; // index de liste manuel
glNewList (roue, GL_COMPILE) ;
// instructions pour dessiner une roue
glEndList ();
```

94

Liste d'affichage : appel

glCallList(numListe)

- ➡ Exécute une liste d'affichage

Ex :

```
Gluint velo, guidon, cadre, roue;
glNewList (velo, GL_COMPILE) ;
glCallList(guidon) ;
glCallList(cadre) ;
glTranslatef (1.0, 0.0, 0.0);
glCallList(roue) ;
glTranslatef (3.0, 0.0, 0.0) ;
glCallList(roue) ;
glEndList ();
```

95

Liste d'affichage : gestion

glGenLists (Gsizei range)

- ➡ générer une suite de nouveaux numéros de liste
- ➡ **roue = glGenLists (1)**

glIsList (Gluint list)

- ➡ retourne **GL_TRUE** si le numéro est déjà utilisé

glDeleteLists (Gluint list, Gsizei range)

- ➡ efface une suite de listes

96