

Plan

4 Synthèse d'image

- Présentation
- **OpenGL**
- De la scène 3D à l'image 2D
- Rendu

OpenGL

OpenGL

Librairie graphique développée par SGI (Silicon Graphics) - 1992

OpenGL

Librairie graphique développée par SGI (Silicon Graphics) - 1992

Version actuelle 4.5

site Web : www.opengl.org

OpenGL

Librairie graphique développée par SGI (Silicon Graphics) - 1992

Version actuelle 4.5

site Web : www.opengl.org

Librairie de bas niveau dédiée uniquement au graphique 3D

OpenGL

Librairie graphique développée par SGI (Silicon Graphics) - 1992

Version actuelle 4.5

site Web : www.opengl.org

Librairie de bas niveau dédiée uniquement au graphique 3D

- Tracé de pixels, segments. Remplissage de triangles.
(possibilité de quadrangles et polygones convexes)

OpenGL

Librairie graphique développée par SGI (Silicon Graphics) - 1992

Version actuelle 4.5

site Web : www.opengl.org

Librairie de bas niveau dédiée uniquement au graphique 3D

- Tracé de pixels, segments. Remplissage de triangles.
(possibilité de quadrangles et polygones convexes)
- Gestion des opérations géométriques 3D et de projection.
Définition de caméra.

OpenGL

Librairie graphique développée par SGI (Silicon Graphics) - 1992

Version actuelle 4.5

site Web : www.opengl.org

Librairie de bas niveau dédiée uniquement au graphique 3D

- Tracé de pixels, segments. Remplissage de triangles.
(possibilité de quadrangles et polygones convexes)
- Gestion des opérations géométriques 3D et de projection.
Définition de caméra.
- Gestion des couleurs, transparence, lumière, réflexions, textures

GLU (OpenGL Utility Library)

GLU (OpenGL Utility Library)

Librairie englobant OpenGL ajoutant des fonctionnalités

GLU (OpenGL Utility Library)

Librairie englobant OpenGL ajoutant des fonctionnalités

- Gestion des matrices de projection et visualisation

GLU (OpenGL Utility Library)

Librairie englobant OpenGL ajoutant des fonctionnalités

- Gestion des matrices de projection et visualisation
- Redimensionnement d'images

GLU (OpenGL Utility Library)

Librairie englobant OpenGL ajoutant des fonctionnalités

- Gestion des matrices de projection et visualisation
- Redimensionnement d'images
- Gestion d'objets quadriques (sphère, cylindre, disque)

GLU (OpenGL Utility Library)

Librairie englobant OpenGL ajoutant des fonctionnalités

- Gestion des matrices de projection et visualisation
- Redimensionnement d'images
- Gestion d'objets quadriques (sphère, cylindre, disque)
- Gestion des courbes et des surfaces de Bézier et NURBS

GLU (OpenGL Utility Library)

Librairie englobant OpenGL ajoutant des fonctionnalités

- Gestion des matrices de projection et visualisation
- Redimensionnement d'images
- Gestion d'objets quadriques (sphère, cylindre, disque)
- Gestion des courbes et des surfaces de Bézier et NURBS
- ...

GLUT (OpenGL Utility Toolkit)

GLUT (OpenGL Utility Toolkit)

Librairie englobant GLU offrant un jeu de procédures pour

GLUT (OpenGL Utility Toolkit)

Librairie englobant GLU offrant un jeu de procédures pour

- la gestion des fenêtres OpenGL,

GLUT (OpenGL Utility Toolkit)

Librairie englobant GLU offrant un jeu de procédures pour

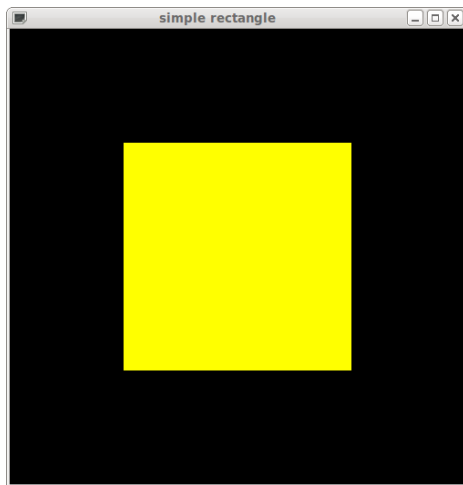
- la gestion des fenêtres OpenGL,
- les interactions avec le système d'exploitation (gestion clavier, souris, etc.)

GLUT (OpenGL Utility Toolkit)

Librairie englobant GLU offrant un jeu de procédures pour

- la gestion des fenêtres OpenGL,
- les interactions avec le système d'exploitation (gestion clavier, souris, etc.)
- indépendamment de celui-ci et du gestionnaire de fenêtres.

Un exemple simple



Tracé d'un carré jaune dans une fenêtre

Un exemple simple

Code source (1/3)

```
// utilisation de la librairie GLUT  
#include <GL/glut.h>  
  
// les ordres graphiques  
// —> dessin de la scène 3D  
void dessin()  
{  
    glMatrixMode(GL_MODELVIEW);  
    glLoadIdentity();  
    glClear(GL_COLOR_BUFFER_BIT);  
    glColor3f(1.0, 1.0, 0.0);  
    glRectf(-25.0, -25.0, 25.0, 25.0);  
    glFlush();  
}
```

Un exemple simple

Code source (2/3)

```
// prise en compte d'un redimensionnement de la fenetre  
// —> définition de la caméra  
void redimensionnement(int w, int h)  
{  
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);  
    glMatrixMode(GL_PROJECTION); // repère de la caméra  
    glLoadIdentity();  
    glOrtho(-50.0, 50.0, -50.0, 50.0, -1.0, 1.0);  
}
```

Un exemple simple

Code source (3/3)

```
// la fonction principale
int main(int argc,char **argv)
{
    glutInit(&argc,argv);
    glutInitWindowSize(500,500);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutCreateWindow(" simple rectangle");
    glutDisplayFunc(dessin);
    glutReshapeFunc(redimensionnement);
    glutMainLoop();
    return (0);
}
```


Programme avec GLUT

Détail du programme-exemple

Programme avec GLUT

Détail du programme-exemple

Inclure les interfaces de la librairie GLUT (ainsi que GLU et Open GL)

```
#include <GL/glut.h>
```

Programme avec GLUT

Détail du programme-exemple - la fonction `dessin`

Programme avec GLUT

La fonction `dessin`

Appliquer les transformations géométriques au repère de la scène

```
glMatrixMode(GL_MODELVIEW);
```

Programme avec GLUT

Détail du programme-exemple - la fonction `dessin`

Initialiser le système de transformations géométriques de `GL_MODELVIEW`

```
glLoadIdentity();
```

Programme avec GLUT

Détail du programme-exemple - la fonction `dessin`

Effacer le buffer graphique de l'image

```
glClear(GL_COLOR_BUFFER_BIT);
```

Programme avec GLUT

Détail du programme-exemple - la fonction `dessin`

Définir la couleur jaune

et créer le carré de diagonale $[(-25, -25) , (25, 25)]$

```
glColor3f(1.0, 1.0, 0.0);  
glRectf(-25.0, -25.0, 25.0, 25.0);
```

Programme avec GLUT

Détail du programme-exemple - la fonction `dessin`

Afficher l'image à l'écran

```
glFlush();
```


Programme avec GLUT

Détail du programme-exemple - la fonction `redimensionnement`

Programme avec GLUT

Détail du programme-exemple - la fonction `redimensionnement`

Mettre en correspondant le buffer image avec la taille de la fenetre

```
glViewport (0, 0, (GLsizei) w, (GLsizei) h);
```

Programme avec GLUT

Détail du programme-exemple - la fonction `redimensionnement`

Appliquer les transformations géométriques au système de projection de la caméra

```
glMatrixMode(GL_PROJECTION);
```

Programme avec GLUT

Détail du programme-exemple - la fonction `redimensionnement`

Initialiser le système de transformations géométriques de `GL_PROJECTION`

```
glLoadIdentity();
```

Programme avec GLUT

Détail du programme-exemple - la fonction `redimensionnement`

Définir la caméra avec projection orthographique et volume englobant correspondant

```
glOrtho(-50.0, 50.0, -50.0, 50.0, -1.0, 1.0);
```

Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Initialiser la librairie GLUT avec les arguments du programme

```
glutInit(&argc , argv );
```

Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Définir une fenêtre graphique de dimension 500×500
avec un *buffer graphique simple* et en mode couleur RGB
et la créer avec le titre `simple rectangle`

```
glutInitWindowSize(500,500);  
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);  
glutCreateWindow(" simple rectangle");
```


Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Associer à la fenêtre précédemment créée, différents *callbacks* (*fonctions de rappels*)

Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Callback associé au dessin dans la fenêtre graphique

```
glutDisplayFunc(dessin);
```

Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Callback associé au (re)dimensionnement de la fenêtre graphique

```
glutReshapeFunc( redimensionnement );
```

Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Il existe d'autres procédures de GLUT permettant de prendre en compte d'événements. Les principales sont :

Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Il existe d'autres procédures de GLUT permettant de prendre en compte d'événements. Les principales sont :

- `glutKeyboardFunc` : pour gérer les touches du clavier correspondant à des caractères imprimables (lettres, chiffres, symboles, ...)

Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Il existe d'autres procédures de GLUT permettant de prendre en compte d'événements. Les principales sont :

- `glutKeyboardFunc` : pour gérer les touches du clavier correspondant à des caractères imprimables (lettres, chiffres, symboles, ...)
- `glutSpecialFunc` : pour gérer les touches du clavier correspondant à des caractères non imprimables (touches de fonction, flèches, ...)

Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Il existe d'autres procédures de GLUT permettant de prendre en compte d'événements. Les principales sont :

- `glutKeyboardFunc` : pour gérer les touches du clavier correspondant à des caractères imprimables (lettres, chiffres, symboles, ...)
- `glutSpecialFunc` : pour gérer les touches du clavier correspondant à des caractères non imprimables (touches de fonction, flèches, ...)
- `glutMouseFunc` : pour gérer les clics souris

Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Il existe d'autres procédures de GLUT permettant de prendre en compte d'événements. Les principales sont :

- `glutKeyboardFunc` : pour gérer les touches du clavier correspondant à des caractères imprimables (lettres, chiffres, symboles, ...)
- `glutSpecialFunc` : pour gérer les touches du clavier correspondant à des caractères non imprimables (touches de fonction, flèches, ...)
- `glutMouseFunc` : pour gérer les clics souris
- `glutMouseMotionFunc` : pour gérer les déplacements de la souris

Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Il existe d'autres procédures de GLUT permettant de prendre en compte d'événements. Les principales sont :

- `glutKeyboardFunc` : pour gérer les touches du clavier correspondant à des caractères imprimables (lettres, chiffres, symboles, ...)
- `glutSpecialFunc` : pour gérer les touches du clavier correspondant à des caractères non imprimables (touches de fonction, flèches, ...)
- `glutMouseFunc` : pour gérer les clics souris
- `glutMouseMotionFunc` : pour gérer les déplacements de la souris
- `glutIdle` : pour gérer l'absence d'évènement

Programme avec GLUT

Détail du programme-exemple - la fonction `main`

Appel de la fonction de boucle de gestion des évènements

```
glutMainLoop();
```

Programme avec GLUT

Principe de la boucle de gestion des évènements

Programme avec GLUT

Principe de la boucle de gestion des évènements

boucle

si la pile d'évènements est non vide **alors**

 dépiler l'évènement e

 :

 :

sinon_si un callback de type Idle est défini **alors**

 exécuter le callback d'inaction

fin_si

fin_boucle

Programme avec GLUT

Principe de la boucle de gestion des évènements

boucle

si la pile d'évènements est non vide **alors**

 dépiler l'évènement e

 :

si e de type MISE_A_JOUR_FENETRE **et**
 un callback de type Display est défini **alors**

 exécuter le callback

fin_si

 :

sinon_si un callback de type Idle est défini **alors**

 exécuter le callback d'inaction

fin_si

fin_boucle

Programme avec GLUT

Principe de la boucle de gestion des évènements

boucle

si la pile d'évènements est non vide **alors**

 dépiler l'évènement e

 :

si e de type REDIMENSIONNEMENT_FENETRE **et**
 un callback de type Reshape est défini **alors**

 exécuter le callback

fin_si

 :

sinon_si un callback de type Idle est défini **alors**

 exécuter le callback d'inaction

fin_si

fin_boucle

Programme avec GLUT

Principe de la boucle de gestion des évènements

boucle

si la pile d'évènements est non vide **alors**

 dépiler l'évènement e

 :

si e de type CLAVIER_TOUCHE_STANDARD **et**
 un callback de type Keyboard est défini **alors**

 exécuter le callback

fin_si

 :

sinon_si un callback de type Idle est défini **alors**

 exécuter le callback d'inaction

fin_si

fin_boucle

Programme avec GLUT

Les callbacks

Procédure avec certains arguments suivant le type d'évènement associé.

Programme avec GLUT

Les callbacks

Procédure avec certains arguments suivant le type d'évènement associé.

Callback associé à la mise à jour de la fenêtre

```
void dessin()
```

Programme avec GLUT

Les callbacks

Procédure avec certains arguments suivant le type d'évènement associé.

Callback associé au (re)dimensionnement de la fenêtre

```
void redimensionnement(int largeur, int hauteur)
```

Programme avec GLUT

Les callbacks

Procédure avec certains arguments suivant le type d'évènement associé.

Callback associé à la gestion des touches claviers

```
void touche_standard  
(int code_ASCII, int x_souris, int y_souris)
```

Programme avec GLUT

Les callbacks

Procédure avec certains arguments suivant le type d'évènement associé.

Callback associé à la gestion des touches claviers

```
void touche_speciale  
(int code_touche, int x_souris, int y_souris)
```

Programme avec GLUT

Les callbacks

Procédure avec certains arguments suivant le type d'évènement associé.

Callback associé à la gestion du click souris

```
void click_souris  
(int num_bouton, int etat, int x_souris, int y_souris)
```

Programme avec GLUT

Les callbacks

Procédure avec certains arguments suivant le type d'évènement associé.

Callback associé à la gestion du déplacement souris

```
void deplacement_souris(int x_souris, int y_souris)
```

Programme avec GLUT

Les callbacks

Procédure avec certains arguments suivant le type d'évènement associé.

Callback d'inaction

```
void inaction()
```