

Université L'arbi en mhidi
Faculté des sciences exactes et des sciences de la nature et de la vie
Département d'informatique
Master01 : Vision artificielle
Synthèse d'images

Solution TP 06

Le code qui consiste à appliquer les différents effets lumineux sur le cube est le suivant :

```
#include <SDL/SDL.h>
#include <GL/gl.h>
#include <GL/glu.h>
#include <cstdlib>
GLfloat L0dif[]={ 0.3,0.3,0.8};
GLfloat L1pos[]={ 2.0,2.0,2.0};
GLfloat L0pos[]={ 0.0,2.0,-1.0};
GLfloat L1dif[]={ 0.5,0.5,0.5};
GLfloat Mspec[]={0.5,0.5,0.5};
GLfloat Mshiny=100;
void Dessiner();

int main(int argc, char *argv[])
{
    SDL_Event event;
    SDL_Init(SDL_INIT_VIDEO);
    atexit(SDL_Quit);
    SDL_WM_SetCaption("SDL GL Application", NULL);
    SDL_SetVideoMode(640, 480, 32, SDL_OPENGL);
    glMatrixMode( GL_PROJECTION );
    glLoadIdentity();
    gluPerspective(70,(double)640/480,1,1000);
    glEnable(GL_DEPTH_TEST);
    /* Paramétrage des lumières */
    glShadeModel(GL_SMOOTH);
    glLightModeli(GL_LIGHT_MODEL_LOCAL_VIEWER,GL_TRUE);
    /*This call places the viewpoint at (0, 0, 0) in eye coordinates. To switch back to an infinite viewpoint, pass in GL_FALSE as the argument.*/

    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_LIGHT1);
    glLightfv(GL_LIGHT0,GL_DIFFUSE,L0dif);
    glLightfv(GL_LIGHT0,GL_SPECULAR,L0dif);
    glLightfv(GL_LIGHT1,GL_DIFFUSE,L1dif);
    glLightfv(GL_LIGHT1,GL_SPECULAR,L1dif);
    /* Paramétrage du matériau */
    glMaterialfv(GL_FRONT_AND_BACK,GL_SPECULAR,Mspec);
    glMaterialf(GL_FRONT_AND_BACK,GL_SHININESS,Mshiny);
    /*Dessiner()*/
    for (;;)
    {
        SDL_WaitEvent(&event);
        switch(event.type)
        {
            case SDL_QUIT:
                exit(0);
                break;
        }
        }Dessiner(); }
    return 0;}
```

```

void Dessiner()
{
    glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
    glMatrixMode( GL_MODELVIEW );
    glLoadIdentity( );
    gluLookAt(3,4,2,0,0,0,0,0,1);
    glLightfv(GL_LIGHT0,GL_POSITION,L0pos);
    glLightfv(GL_LIGHT1,GL_POSITION,L1pos);
    glBegin(GL_QUADS);
    glColor3ub(255,0,0); //face rouge
    glVertex3d(1,1,1);
    glVertex3d(1,1,-1);
    glVertex3d(-1,1,-1);
    glVertex3d(-1,1,1);
    glColor3ub(0,255,0); //face verte
    glVertex3d(1,-1,1);
    glVertex3d(1,-1,-1);
    glVertex3d(1,1,-1);
    glVertex3d(1,1,1);
    glColor3ub(0,0,255); //face bleue
    glVertex3d(-1,-1,1);
    glVertex3d(-1,-1,-1);
    glVertex3d(1,-1,-1);
    glVertex3d(1,-1,1);
    glColor3ub(255,255,0); //face jaune
    glVertex3d(-1,1,1);
    glVertex3d(-1,1,-1);
    glVertex3d(-1,-1,-1);
    glVertex3d(-1,-1,1);
    glColor3ub(0,255,255); //face cyan
    glVertex3d(1,1,-1);
    glVertex3d(1,-1,-1);
    glVertex3d(-1,-1,-1);
    glVertex3d(-1,1,-1);
    glColor3ub(255,0,255); //face magenta
    glVertex3d(1,-1,1);
    glVertex3d(1,1,1);
    glVertex3d(-1,1,1);
    glVertex3d(-1,-1,1);
    glEnd();glFlush();
    SDL_GL_SwapBuffers();
}

```