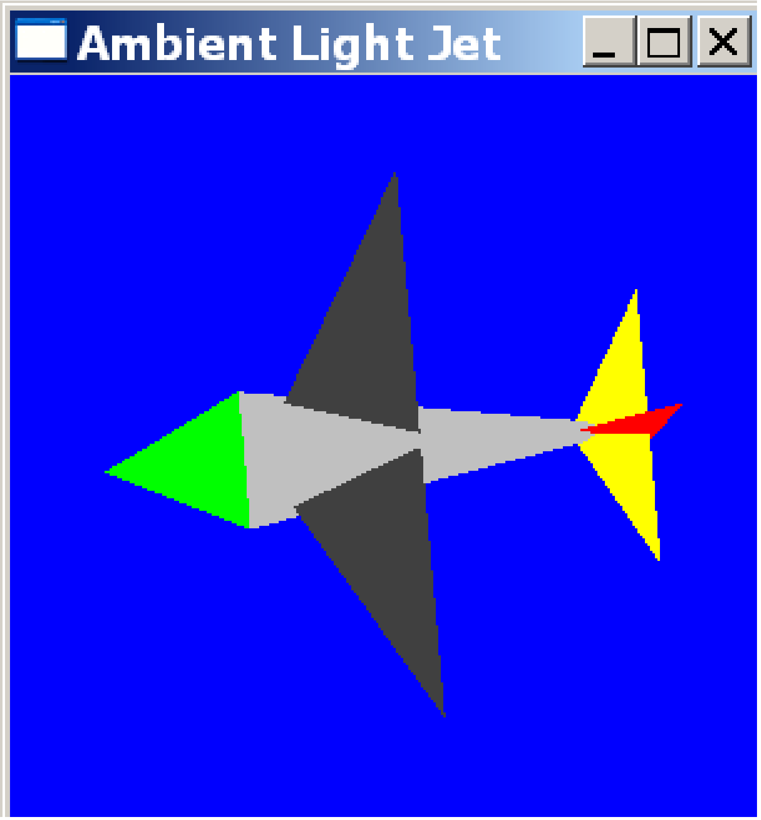
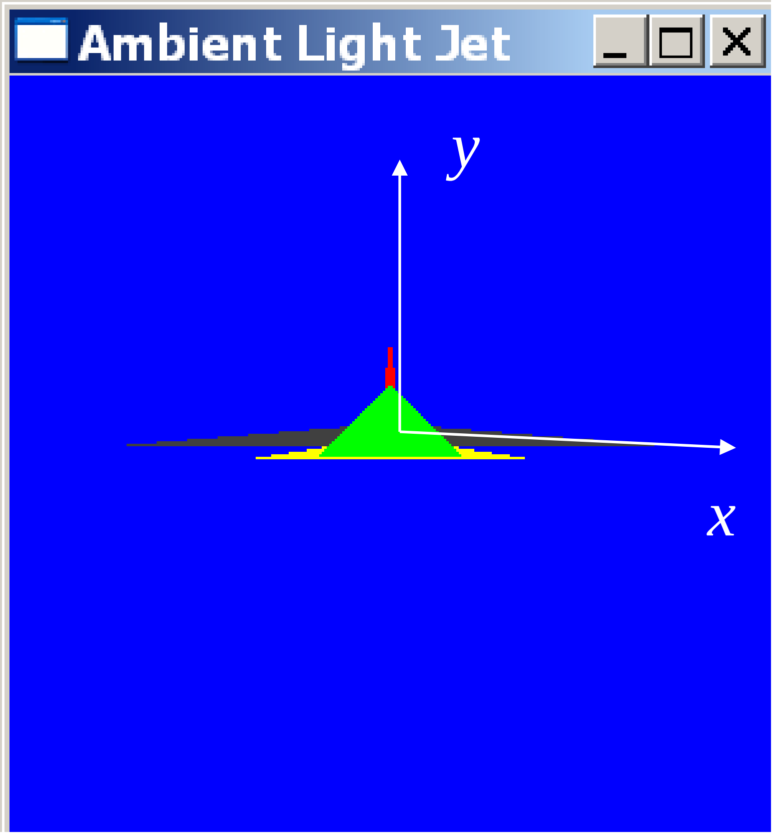
10-1 簡易型飛機 (ambient.c)



```
// Ambient.c
// OpenGL SuperBible, Chapter 6
// Beginning of OpenGL lighting sample
// Demonstrates Ambient Lighting
// Program by Richard S. Wright Jr.

#include <windows.h>
#include <gl/gl.h>
#include <gl/glu.h>
#include <gl/glut.h>
#include <math.h>

// Define a constant for the value of PI
#define GL_PI 3.1415f

// Rotation amounts
static GLfloat xRot = 0.0f;
static GLfloat yRot = 0.0f;
```

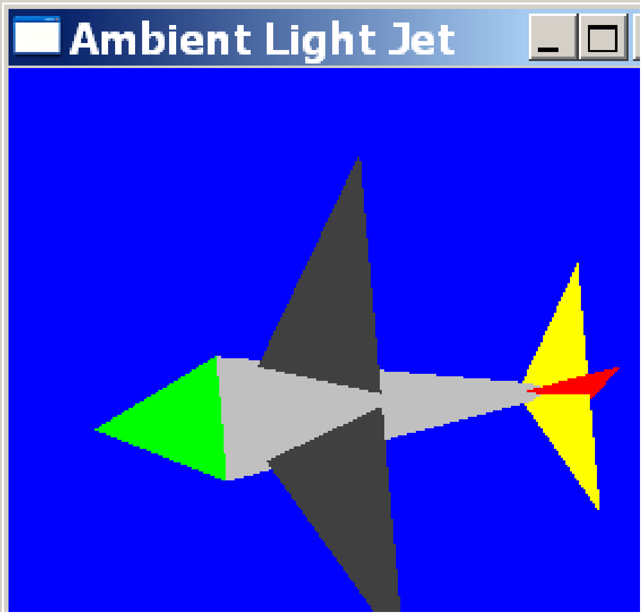
Ambient.c 2/13

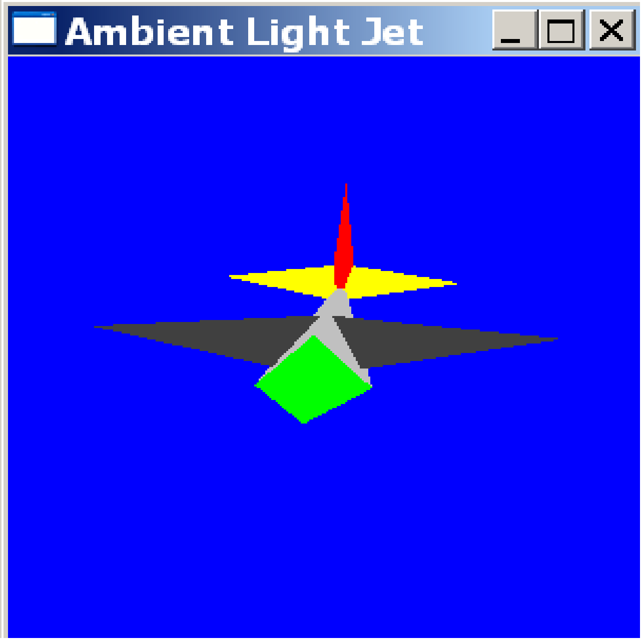
```
// Called to draw scene
void RenderScene(void)
 {
   // Clear the window with current clearing color
  glClear(GL_COLOR_BUFFER_BIT |GL_DEPTH_BUFFER_BIT);

  // Save the matrix state
  glPushMatrix();
  glRotatef(xRot, 1.0f, 0.0f, 0.0f);
  glRotatef(yRot, 0.0f, 1.0f, 0.0f);
```



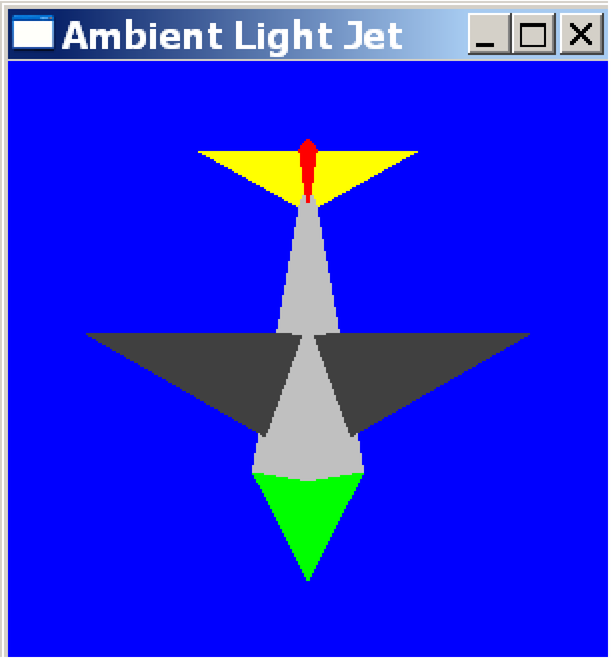
Ambient.c 3/13

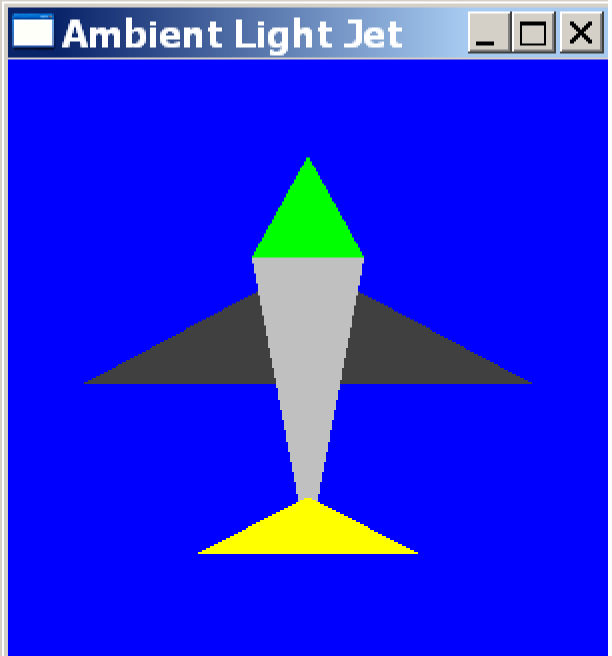
```
 // Nose Cone //////////////////////////////
// Bright Green
 glColor3ub(0, 255, 0);
 glBegin(GL_TRIANGLES);
   glVertex3f(0.0f, 0.0f, 60.0f);
   glVertex3f(-15.0f, 0.0f, 30.0f);
   glVertex3f(15.0f,0.0f,30.0f);

   glVertex3f(15.0f,0.0f,30.0f);
   glVertex3f(0.0f, 15.0f, 30.0f);
   glVertex3f(0.0f, 0.0f, 60.0f);

   glVertex3f(0.0f, 0.0f, 60.0f);
   glVertex3f(0.0f, 15.0f, 30.0f);
   glVertex3f(-15.0f,0.0f,30.0f);
```





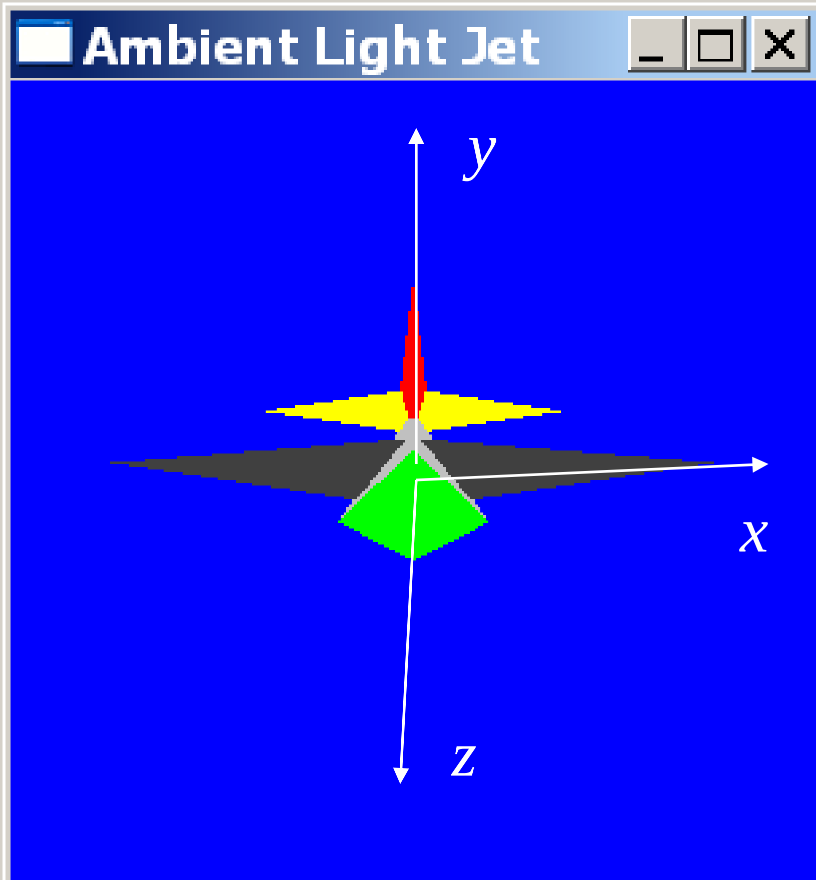
```
// Body of the Plane ////////////////////////
// light gray
  glColor3ub(192,192,192);
  glVertex3f(-15.0f,0.0f,30.0f);
  glVertex3f(0.0f, 15.0f, 30.0f);
  glVertex3f(0.0f, 0.0f, -56.0f);

  glVertex3f(0.0f, 0.0f, -56.0f);
  glVertex3f(0.0f, 15.0f, 30.0f);
  glVertex3f(15.0f,0.0f,30.0f);

  glVertex3f(15.0f,0.0f,30.0f);
  glVertex3f(-15.0f, 0.0f, 30.0f);
  glVertex3f(0.0f, 0.0f, -56.0f);
```





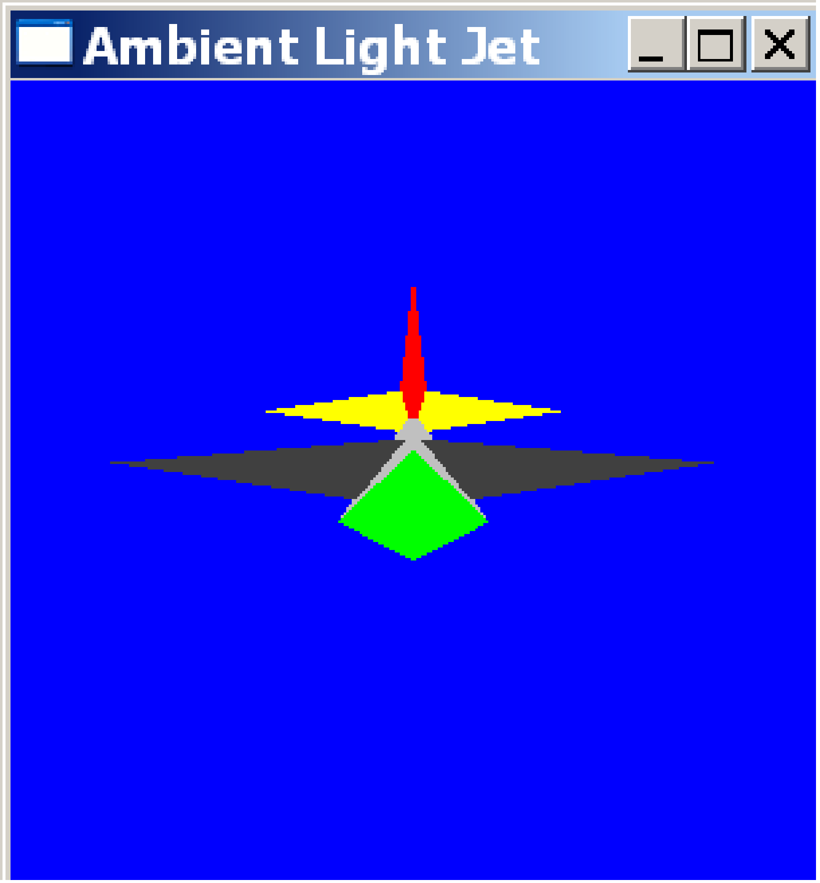
Ambient.c 5/13

```
///////////////////////////////////////////////
// Left wing
// Dark gray
  glColor3ub(64,64,64);
  glVertex3f(0.0f,2.0f,27.0f);
  glVertex3f(-60.0f, 2.0f, -8.0f);
  glVertex3f(60.0f, 2.0f, -8.0f);

  glVertex3f(60.0f, 2.0f, -8.0f);
  glVertex3f(0.0f, 7.0f, -8.0f);
  glVertex3f(0.0f,2.0f,27.0f);

  glVertex3f(60.0f, 2.0f, -8.0f);
  glVertex3f(-60.0f, 2.0f, -8.0f);
  glVertex3f(0.0f,7.0f,-8.0f);
```



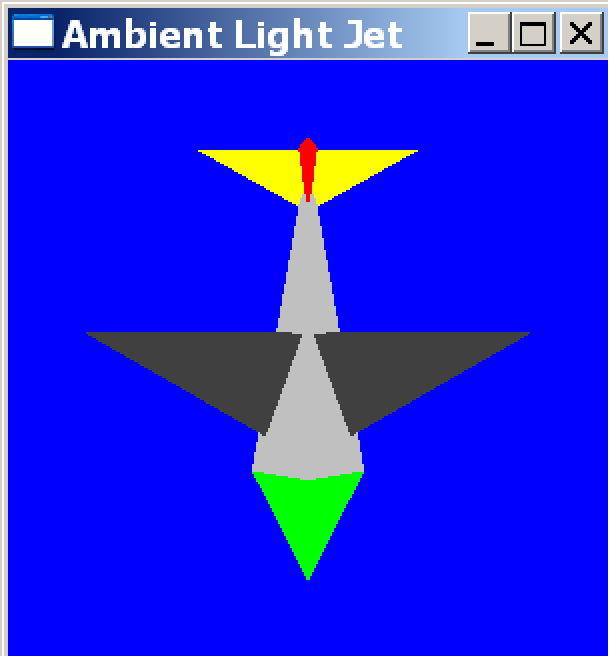
Ambient.c 6/13

```
// Other wing top section
  glVertex3f(0.0f,2.0f,27.0f);
  glVertex3f(0.0f, 7.0f, -8.0f);
  glVertex3f(-60.0f, 2.0f, -8.0f);

// Tail section///////////////////////////////
  // Bottom of back fin
    glColor3ub(255,255,0);
    glVertex3f(-30.0f, -0.50f, -57.0f);
    glVertex3f(30.0f, -0.50f, -57.0f);
    glVertex3f(0.0f,-0.50f,-40.0f);

  // top of left side
    glVertex3f(0.0f,-0.5f,-40.0f);
    glVertex3f(30.0f, -0.5f, -57.0f);
    glVertex3f(0.0f, 4.0f, -57.0f);
```



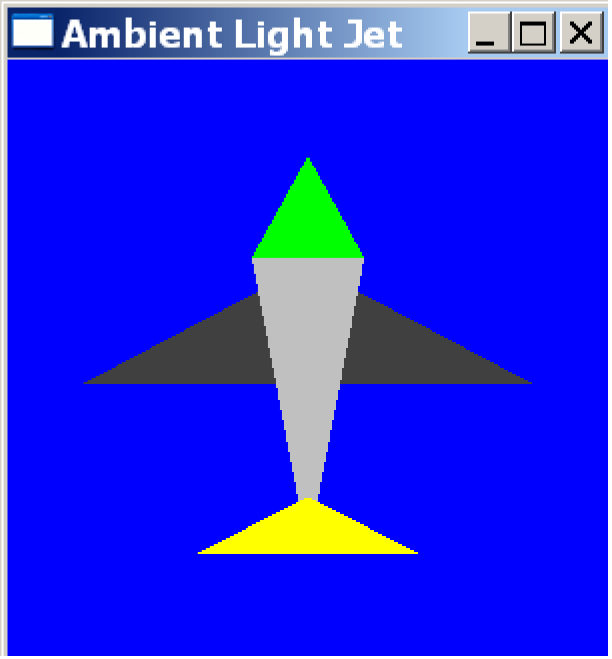
Ambient.c 7/13

```
// top of right side
   glVertex3f(0.0f, 4.0f, -57.0f);
   glVertex3f(-30.0f, -0.5f, -57.0f);
   glVertex3f(0.0f,-0.5f,-40.0f);

// back of bottom of tail
   glVertex3f(30.0f,-0.5f,-57.0f);
   glVertex3f(-30.0f, -0.5f, -57.0f);
   glVertex3f(0.0f, 4.0f, -57.0f);
// Top of Tail section left
   glColor3ub(255,0,0);
   glVertex3f(0.0f,0.5f,-40.0f);
   glVertex3f(3.0f, 0.5f, -57.0f);
   glVertex3f(0.0f, 25.0f, -65.0f);

   glVertex3f(0.0f, 25.0f, -65.0f);
   glVertex3f(-3.0f, 0.5f, -57.0f);
   glVertex3f(0.0f,0.5f,-40.0f);
```



Ambient.c 8/13

```
// Back of horizontal section
      glVertex3f(3.0f,0.5f,-57.0f);
      glVertex3f(-3.0f, 0.5f, -57.0f);
      glVertex3f(0.0f, 25.0f, -65.0f);
  glEnd();

  glPopMatrix();

// Display the results
  glutSwapBuffers();
 }
```



```
// This function does any needed initialization on the rendering
// context.
void SetupRC()
 {
   // Light values
  // Bright white light
    GLfloat ambientLight[] = { 1.0f, 1.0f, 1.0f, 1.0f };

    glEnable(GL_DEPTH_TEST); // Hidden surface removal
    glEnable(GL_CULL_FACE);   // Do not calculate inside of jet
    glFrontFace(GL_CCW);
                              // Counter clock-wise polygons face out

   // Lighting stuff
    glEnable(GL_LIGHTING);      // Enable lighting
```

Ambient.c 10/13

```
// Set light model to use ambient light specified by ambientLight
glLightModelfv(GL_LIGHT_MODEL_AMBIENT,ambientLight);



glEnable(GL_COLOR_MATERIAL);
                                  // Enable Material color tracking

// Front material ambient and diffuse colors track glColor
glColorMaterial(GL_FRONT,GL_AMBIENT_AND_DIFFUSE);

// Nice light blue
  glClearColor(0.0f, 0.0f, 05.f,1.0f);
}
```

Ambient.c 11/13

```
void SpecialKeys(int key, int x, int y)
 {
    if (key == GLUT_KEY_UP)       xRot-= 5.0f;
    if (key == GLUT_KEY_DOWN) xRot += 5.0f;
    if (key == GLUT_KEY_LEFT)   yRot -= 5.0f;
    if (key == GLUT_KEY_RIGHT) yRot += 5.0f;
    if (key > 356.0f)  xRot = 0.0f;
    if (key < -1.0f)     xRot = 355.0f;
    if (key > 356.0f)  yRot = 0.0f;
    if (key < -1.0f)     yRot = 355.0f;

    // Refresh the Window
    glutPostRedisplay();
  }
```

Ambient.c 12/13

```
void ChangeSize(int w, int h)
{
  GLfloat nRange = 80.0f;
   // Prevent a divide by zero
      if(h == 0) h = 1;

   // Set Viewport to window dimensions
     glViewport(0, 0, w, h);
   // Reset coordinate system
     glMatrixMode(GL_PROJECTION);
     glLoadIdentity();
  // Establish clipping volume (left, right, bottom, top, near, far)
  if (w <= h)
    glOrtho (-nRange, nRange, -nRange*h/w, nRange*h/w, - nRange, nRange);
  else
    glOrtho (-nRange*w/h, nRange*w/h, -nRange, nRange, -nRange, nRange);
  glMatrixMode(GL_MODELVIEW);
  glLoadIdentity();
}
```

Ambient.c 13/13

```
int main(int argc, char* argv[])
{
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB |  GLUT_DEPTH);
   glutCreateWindow("Ambient Light Jet");
   glutReshapeFunc(ChangeSize);
   glutSpecialFunc(SpecialKeys);
   glutDisplayFunc(RenderScene);
   SetupRC();
   glutMainLoop();

   return 0;
}
```