Kingdom of Saudi Arabia
المملكة العربية السعودية
Ministry of Education
وزارة التعليم
Umm AlQura University
جامعة أم القرى
Adham University College
الكلية الجامعية بأضم
Computer Science Department
قسم الحاسب الآلي

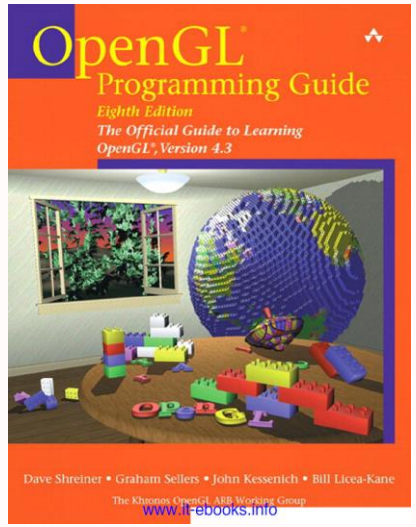
CS Department

Computer Graphics Course, 3-6803430

T.Mariah Khayat

References

- *Lab Lectures, Computer Graphics, Taif University, Faculty Of Computers And Information Technology, TA. Maha Thafar &TA. Haifa Alshehri, TA.Sohair Soliman & L.Shakila Bano.*
- *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 4.3, 8th edition, Dave Shreiner, Graham Sellers, John Kessenich, Bill Licea-Kane & The Khronos OpenGL ARB Working Group, Addison-Wesley.*



Lecture Eight

The Other Transformations in OpenGL

content

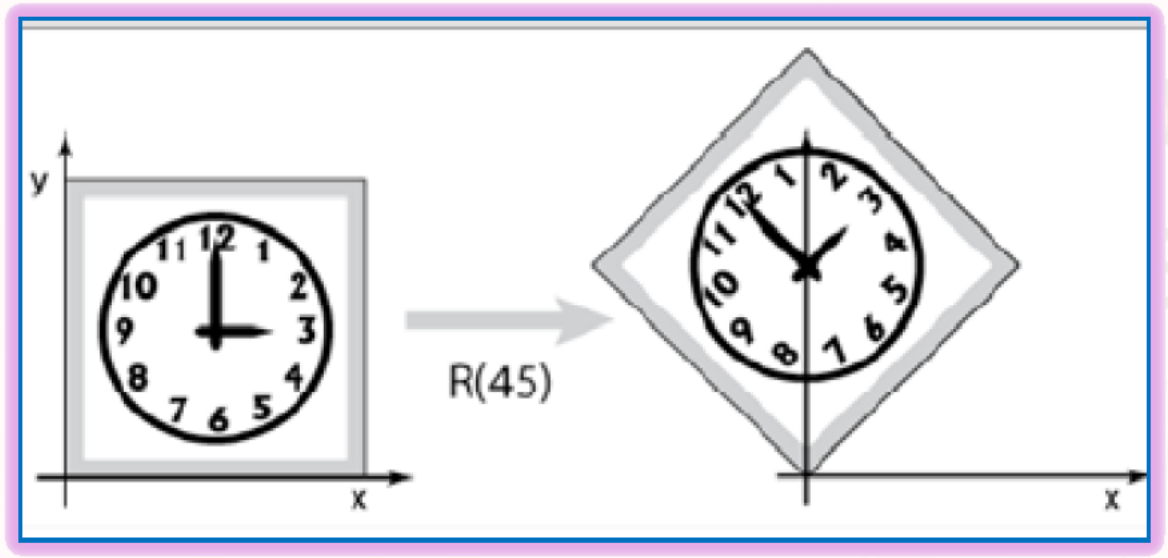
1. Rotate Objects using Modeling Transformation
2. Scale Objects using Modeling Transformation
3. Rotate 3D Teapot Program
4. Order of Transformation
5. Viewport Transformation
6. Put Multiple Images in One Window Using View Port Functions Program

Rotate Objects using Modeling Transformation

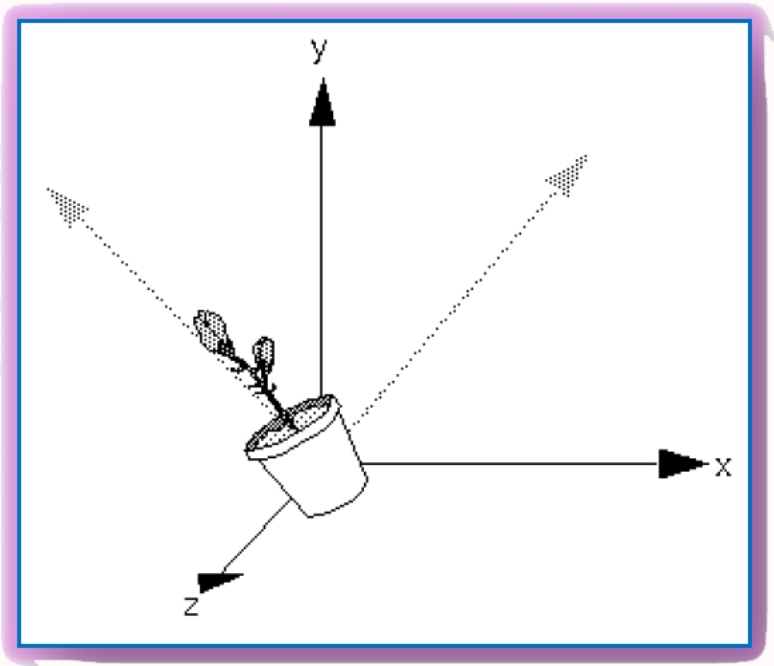
- In order to **rotate** an object
- You have to rotate the coordinate system (e.g. rotation of 40° around the Z axis)
- You draw the object in the rotated coordinate system
- The Rotation function: **void glRotatef(theta, x,y,z)**
- **Example: glRotatef(45 ,0 ,1 ,0);** //Rotate 45 deg around the y axis
- **Note:** The argument of x , y and z can take only the values (**0 , 1 or -1)** means rotate around the given axis.

Rotate Objects using Modeling Transformation

2D



3D

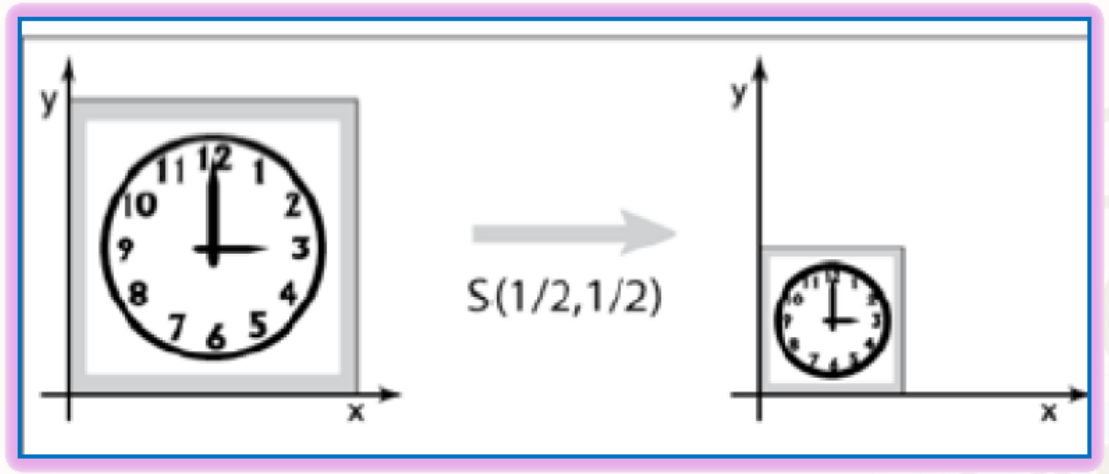


Scale Objects using Modeling Transformation

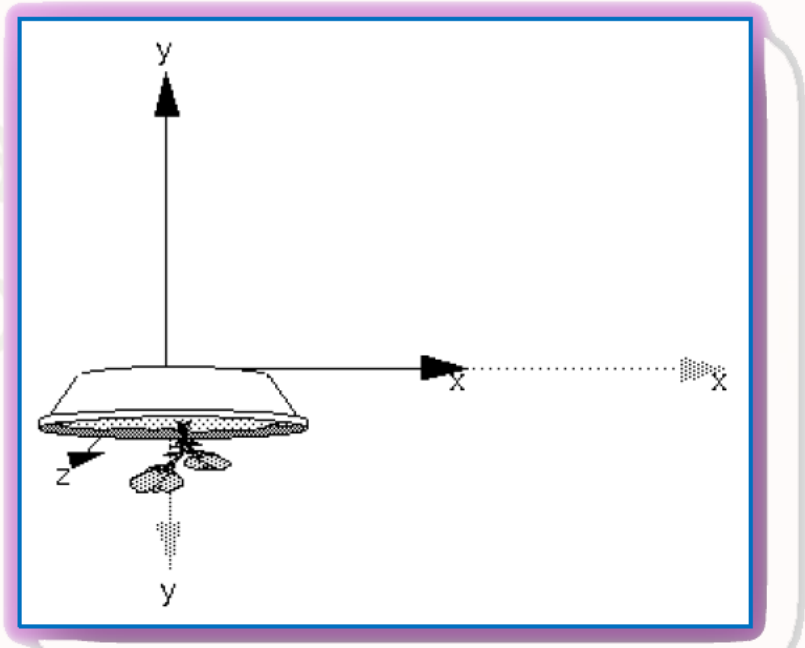
- In order to scale an object
- You first scale the coordinate system (e.g. uniform scaling of 1/2)
- You draw the object in the scaled coordinate system
- the Scale function: **void glScalef(sx ,sy , sz).**
- **Example: glScalef(2, 1 ,1); //scale the x-axis by 2, don't scale the y and z axes**

Scale Objects using Modeling Transformation

2D



3D



Rotate 3D Teapot Program

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

void Draw() {
glClear(GL_COLOR_BUFFER_BIT);
gluLookAt(0, 3, 5, 0, 1, 0, 0, 1, 2);
glColor3f(1.0,0.5,0.5); //Pink Color
glutSolidTeapot(4); //Draw the Teapot
glRotatef(60, 1, 1, 0); //Rotate the Teapot
glColor3f(1, 0.2, 0.2); //Color after the rotation
glutSolidTeapot(4); //Redraw the Teapot
glFlush();
}
void init() {
    glClearColor(1, 0.9, 0.8, 0.0);
    glShadeModel(GL_FLAT); //Use one Color.
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glFrustum(-2, 2, -3, 3, 1, 20);
    glMatrixMode(GL_MODELVIEW);
}
int main() {
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(500, 500);
glutInitWindowPosition(0,0);
glutCreateWindow("Rotating a Teapot");
init();
glutDisplayFunc(Draw);
glutMainLoop();
return 0;
}
```

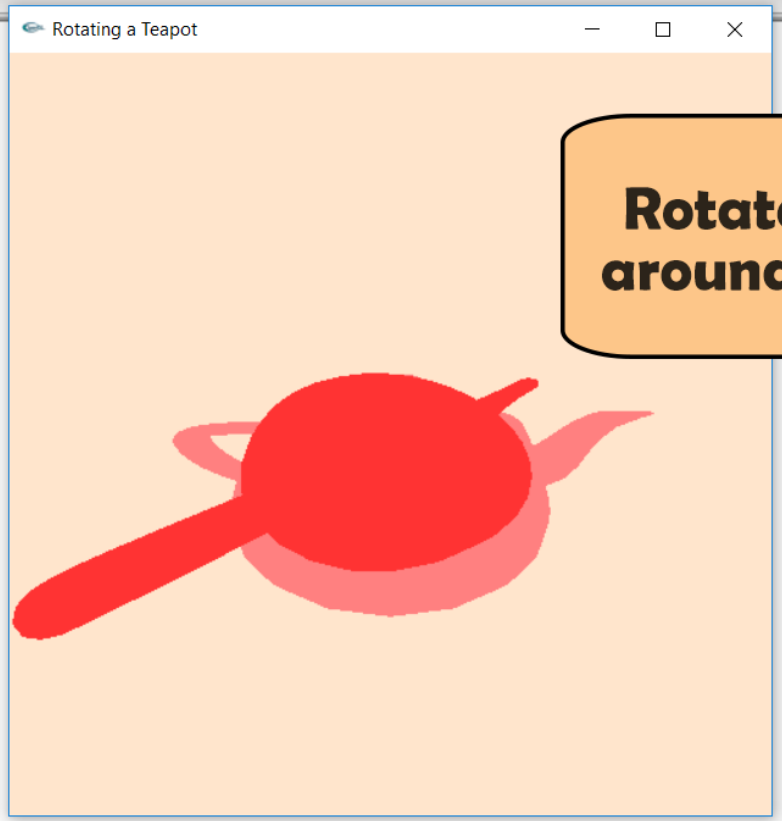
Rotate 3D Teapot Program

aPot] - Code::Blocks 17.12

Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

*Rotate3DTeapot.cpp

```
1  #include<windows.h>
2  #include<GL/glut.h>
3
4  void Draw() {
5  glClear(GL_COLOR_BUFFER_BIT);
6  gluLookAt(0, 3, 5, 0, 1, 0, 0, 1, 2);
7  glColor3f(1.0,0.5,0.5); //Pink Color
8  glutSolidTeapot(4); //Draw the Teapot
9  glRotatef(60, 1, 1, 0); //Rotate the Teapot
10 glColor3f(1, 0.2, 0.2); //Color after the rotation
11 glutSolidTeapot(4); //Redraw the Teapot
12 glFlush();
13 }
14 void init() {
15     glClearColor(1, 0.9, 0.8, 0.0);
16     glShadeModel(GL_FLAT); //Use one Color.
17     glMatrixMode(GL_PROJECTION);
18     glLoadIdentity();
19     glFrustum(-2, 2, -3, 3, 1, 20);
20     glMatrixMode(GL_MODELVIEW);
21 }
22 int main() {
23 glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
24 glutInitWindowSize(500, 500);
25 glutInitWindowPosition(0,0);
26 glutCreateWindow("Rotating a Teapot");
27 init();
28 glutDisplayFunc(Draw);
29 glutMainLoop();
30 return 0;
31 }
```



Rotate 60 degree around x and y axis

Rotate 3D Teapot Program

ot] - Code::Blocks 17.12

Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

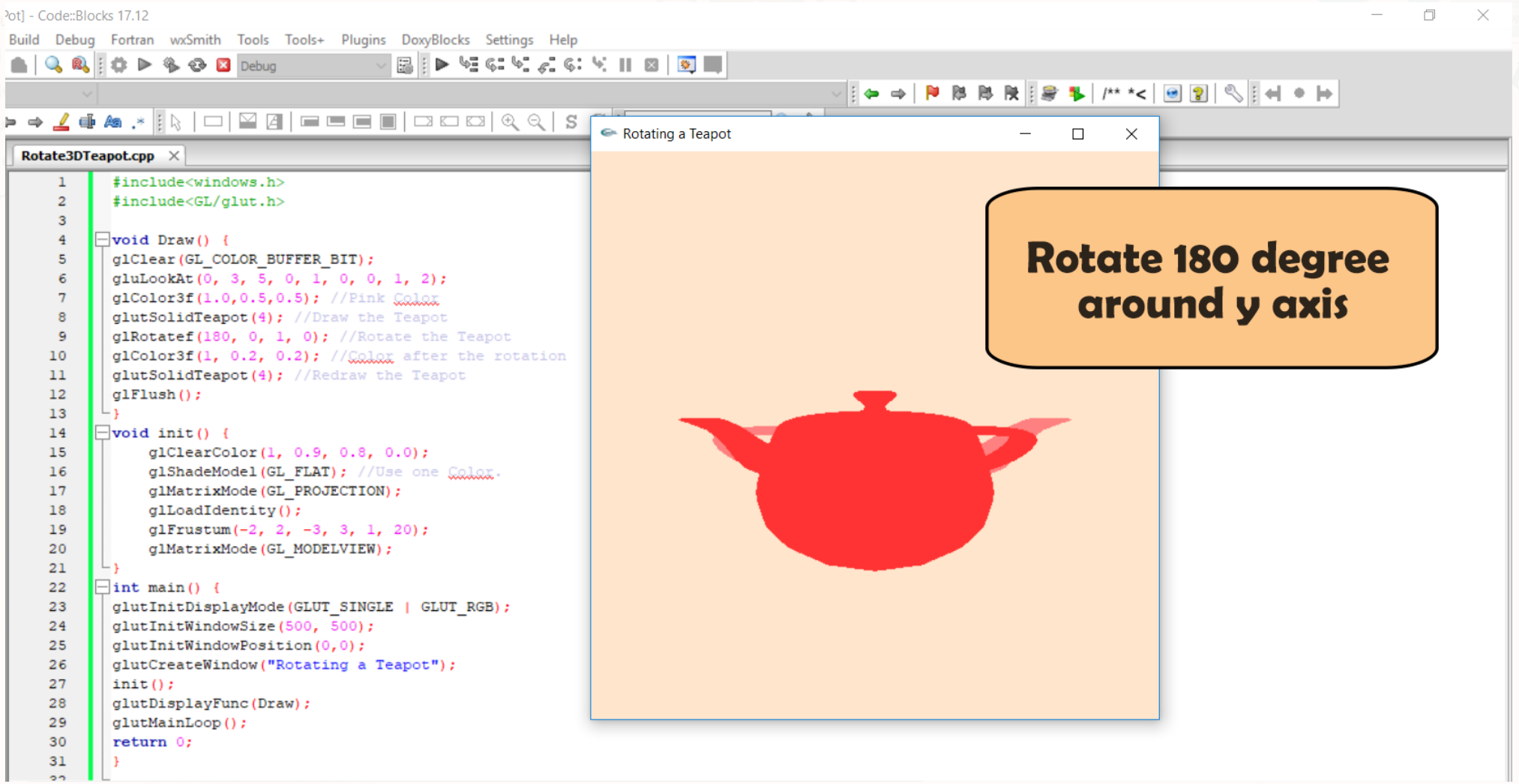
Debug

Rotate3DTeapot.cpp

```
1  #include<windows.h>
2  #include<GL/glut.h>
3
4  void Draw() {
5  glClear(GL_COLOR_BUFFER_BIT);
6  gluLookAt(0, 3, 5, 0, 1, 0, 0, 1, 2);
7  glColor3f(1.0,0.5,0.5); //Pink Color
8  glutSolidTeapot(4); //Draw the Teapot
9  //glRotatef(60, 1, 1, 0); //Rotate the Teapot
10 //glColor3f(1, 0.2, 0.2); //Color after the rotation
11 //glutSolidTeapot(4); //Redraw the Teapot
12 glFlush();
13 }
14 void init() {
15     glClearColor(1, 0.9, 0.8, 0.0);
16     glShadeModel(GL_FLAT); //Use one Color.
17     glMatrixMode(GL_PROJECTION);
18     glLoadIdentity();
19     glFrustum(-2, 2, -3, 3, 1, 20);
20     glMatrixMode(GL_MODELVIEW);
21 }
22 int main() {
23 glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
24 glutInitWindowSize(500, 500);
25 glutInitWindowPosition(0,0);
26 glutCreateWindow("Rotating a Teapot");
27 init();
28 glutDisplayFunc(Draw);
29 glutMainLoop();
30 return 0;
31 }
```

One Teapot without Rotate

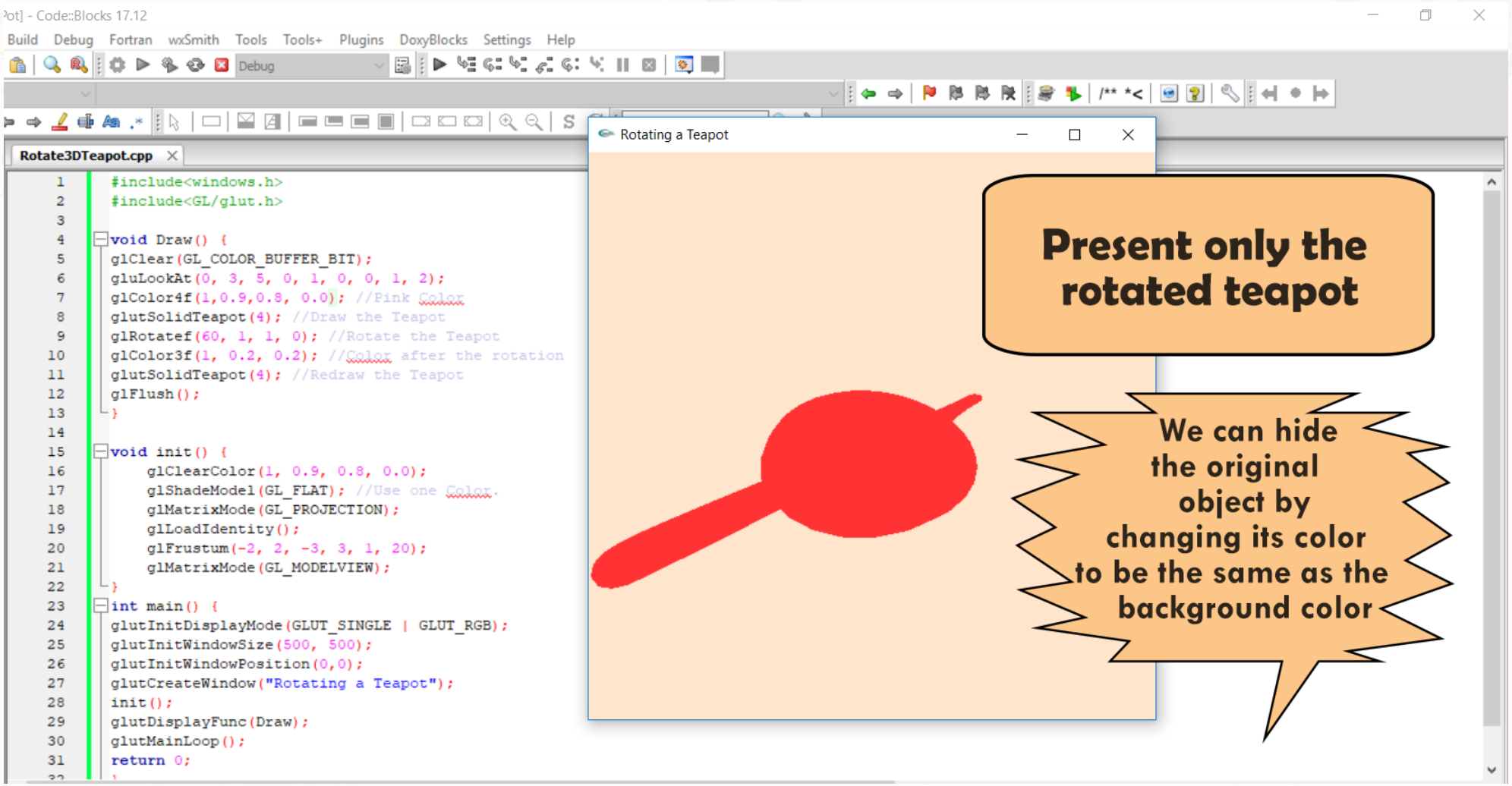
Rotate 3D Teapot Program




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

void Draw() {
glClear(GL_COLOR_BUFFER_BIT);
gluLookAt(0, 3, 5, 0, 1, 0, 0, 1, 2);
glColor3f(1.0,0.5,0.5); //Pink Color
glutSolidTeapot(4); //Draw the Teapot
glRotatef(180, 0, 1, 0); //Rotate the Teapot
glColor3f(1, 0.2, 0.2); //Color after the rotation
glutSolidTeapot(4); //Redraw the Teapot
glFlush();
}
void init() {
    glClearColor(1, 0.9, 0.8, 0.0);
    glShadeModel(GL_FLAT); //Use one Color.
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glFrustum(-2, 2, -3, 3, 1, 20);
    glMatrixMode(GL_MODELVIEW);
}
int main() {
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(500, 500);
glutInitWindowPosition(0,0);
glutCreateWindow("Rotating a Teapot");
init();
glutDisplayFunc(Draw);
glutMainLoop();
return 0;
}
```

Rotate 3D Teapot Program




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

void Draw() {
glClear(GL_COLOR_BUFFER_BIT);
gluLookAt(0, 3, 5, 0, 1, 0, 0, 1, 2);
glColor4f(1,0.9,0.8, 0.0); //Pink Color
glutSolidTeapot(4); //Draw the Teapot
glRotatef(60, 1, 1, 0); //Rotate the Teapot
glColor3f(1, 0.2, 0.2); //Color after the rotation
glutSolidTeapot(4); //Redraw the Teapot
glFlush();
}

void init() {
    glClearColor(1, 0.9, 0.8, 0.0);
    glShadeModel(GL_FLAT); //Use one Color.
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glFrustum(-2, 2, -3, 3, 1, 20);
    glMatrixMode(GL_MODELVIEW);
}
int main() {
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(500, 500);
glutInitWindowPosition(0,0);
glutCreateWindow("Rotating a Teapot");
init();
glutDisplayFunc(Draw);
glutMainLoop();
return 0;
```

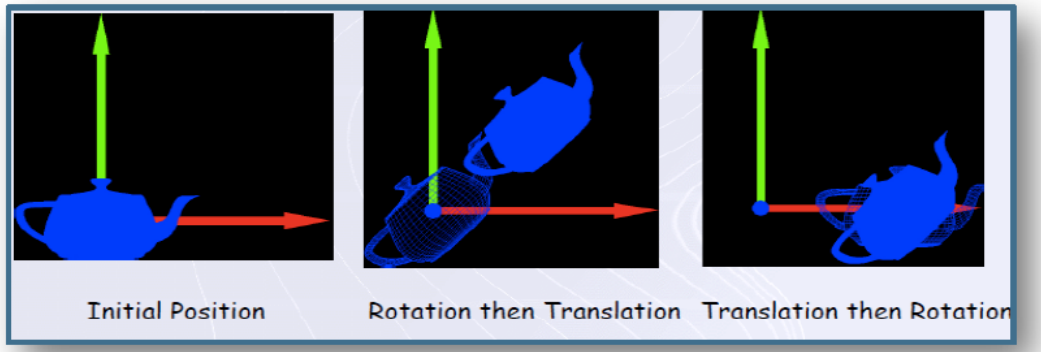
Order of Transformation

- The commands of **modelling transformation** are issued in reverse order of operation.
- Consider the following code fragment:

```
glScalef(2.0f,1.0f,1.0f);              //This is performed last
glTranslatef(5.0f,0.0f,0.0);           //This is performed second
glRotatef(45.0f,0.0f,1.0f,0.0f);       //This is performed first
glutWireCube(1.0);
```

- The first operation performed on the cube is the Rotation, followed by the translation followed by the scaling.

  - See the different when we change the order of transformation:



Viewport Transformation

- View port transformation transforms **x** and **y** from normalized device coordinates to window coordinates.

- Note that **z** value are not transformed in this stage , because **z** axis is orthogonal to the window (2D) and **Z** values have no effect in the mapping.

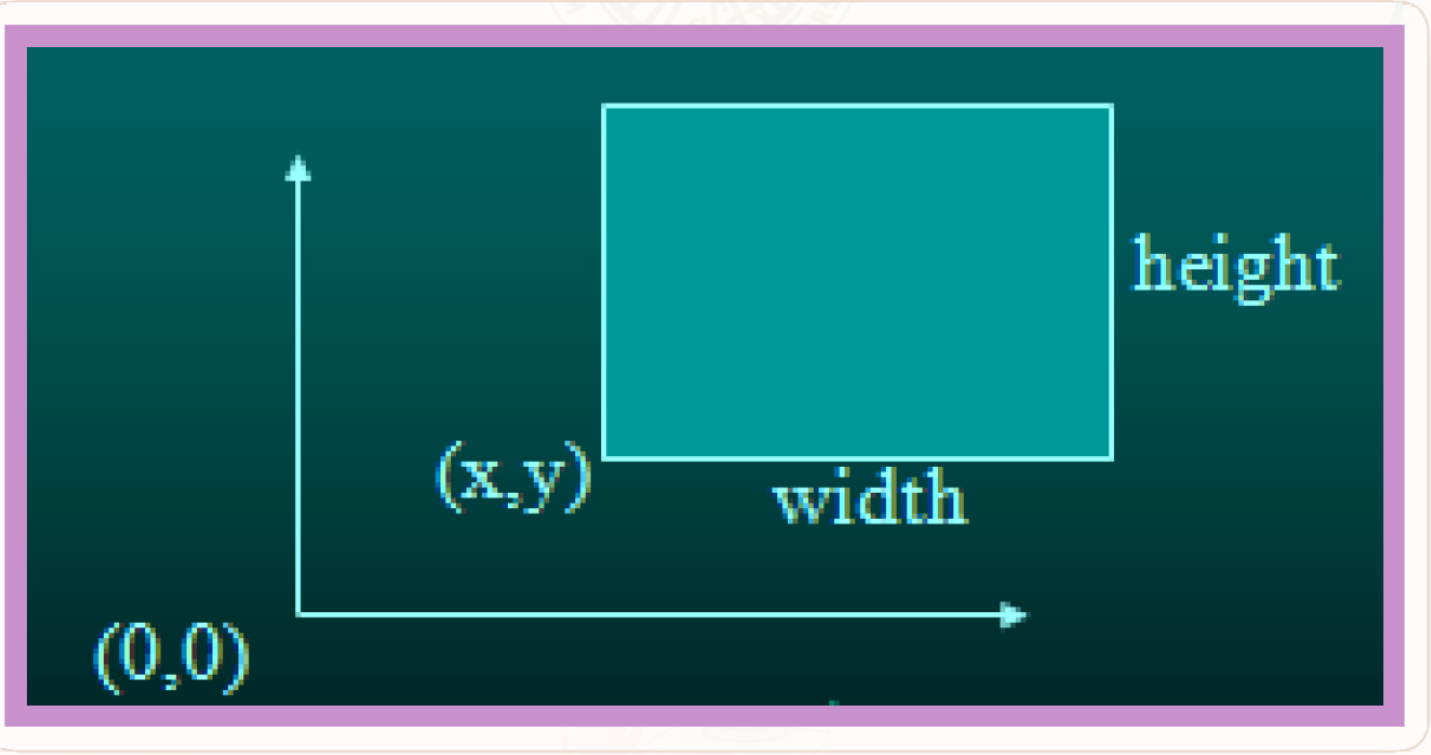
Viewport Transformation

- The function:

void glViewport(GLint x, GLint y, GLsizei width,GLsizei height)

- **x, y:** are normally set to **(0, 0).**
- **width, height:** are normally set to window width and height.
- This means your 2D image size matches your window size. **(view port usually same as window size)**
- You can make your image smaller or bigger than the window by adjusting those values (**E.g**. put multiple images in one window)
- We can use viewport to divide the main window as the games that support tow players.

Viewport Transformation



Put Multiple Images in One Window Using View Port Functions Program

```
#include<windows.h>
#include<GL/glut.h>
#include<math.h>
void Display(void) {
     glColor3f(1, 0.6, 0); //رأس الطائر
     glBegin(GL_POLYGON);
     {    float angle, x2 = 75, y2 = 35, radius = 5;
          for (angle = 0; angle<= 360; angle++)
          glVertex2f(x2+sin(angle)*radius, y2+cos(angle)*radius); }
     glEnd();
     glColor3f(1,1,1); //العيون، الجزء الأبيض
     glBegin(GL_POLYGON);
     {  float angle, x2 = 77, y2 = 37, radius = 2;
        for (angle = 0; angle<= 360; angle++)
        glVertex2f(x2+sin(angle)*radius, y2+cos(angle)*radius); }
     glEnd();
     glColor3f(0,0,0); //العيون، الجزء الأسود
          glBegin(GL_POLYGON);
     {  float angle, x2 = 77, y2 = 37, radius = 1;
        for (angle = 0; angle<= 360; angle++)
        glVertex2f(x2+sin(angle)*radius, y2+cos(angle)*radius); }
     glEnd();
     glColor3f(1, 0.6, 0); //جسم الطائر
     glBegin(GL_POLYGON);
     glVertex2f(73,30);
     glVertex2f(77,30);
     glVertex2f(80,25);
     glVertex2f(75,20);
     glVertex2f(60,20);
     glEnd();
     glColor3f(1, 0.6, 0); //الذيل
     glBegin(GL_TRIANGLES);
     glVertex2f(55,25);
     glVertex2f(50,23);
     glVertex2f(60,20);
     glEnd();
     glColor3f(0.5, 0.2, 0.0);  //المنقار
     glLineWidth(2);
     glBegin(GL_LINES);
     glVertex2f(80, 33);
     glVertex2f(84, 31);
     glEnd();
     glColor3f(0.5, 0.2, 0.0);
     glBegin(GL_TRIANGLES);
     glVertex2f(80,33);
     glVertex2f(80,35);
     glVertex2f(85,35);
     glEnd();
     glLineWidth(3);  //أرجل الطائر
     glBegin(GL_LINES);
     glColor3f(0.5, 0.2, 0.0);
     glVertex2i(75,20);
     glVertex2i(75,13);
     glVertex2i(75,15);
     glVertex2i(77,15);
     glVertex2i(75,15);
     glVertex2i(73,15);
     glVertex2i(70,20);
     glVertex2i(70,13);
     glVertex2i(70,15);
     glVertex2i(72,15);
     glVertex2i(70,15);
     glVertex2i(68,15);
    glEnd();}
```

```
void init() {
glClearColor(0.6,0.9,0.5,0);
glClear(GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(40, 100, -10, 60, -20,
20);
glMatrixMode(GL_MODELVIEW);
glViewport(0,0,100,200);
//النافذة الأولى
glLoadIdentity();
glTranslatef(-30, -10, 0);
glScalef(1.5, 1.5, 1.5);
Display();
glViewport(100,0,100,100);
//النافذة الثانية
glLoadIdentity();
glTranslatef(-27, 0, 0);
glScalef(1.5, 1.5, 1.5);
Display();
glViewport(100,100,100,100);
//النافذة الثالثة
glLoadIdentity();
Display();
glTranslatef(10, -10, 0);
Display();
glutSwapBuffers(); }
int main() {
glutInitDisplayMode(GLUT_DEPTH |
GLUT_DOUBLE | GLUT_RGB);
glutInitWindowSize(300, 300);
glutInitWindowPosition(100, 100);
glutCreateWindow("Some Birds");
glutDisplayFunc(init);
glutMainLoop();}
```

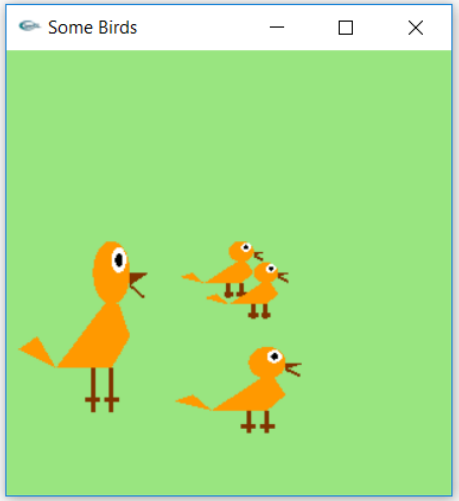
Put Multiple Images in One Window Using View Port Functions Program

iewPort.cpp [MultipleImagesInOneWindowUsingViewPort] - Code::Blocks 17.12

Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

MultipleImagesInOneWindowUsingViewPort.cpp

```
#include<windows.h>
#include<GL/glut.h>
#include<math.h>
void Display(void) {
    //رأس الطائر
    glColor3f(1, 0.6, 0);
    glBegin(GL_POLYGON);
    {
        float angle, x2 = 75, y2 = 35, radius = 5;
        for (angle = 0; angle<= 360; angle++)
           glVertex2f(x2+sin(angle)*radius, y2+cos(angle)*radius);
    }
    glEnd();
    //العين الجزء الأبيض
    glColor3f(1,1,1);
    glBegin(GL_POLYGON);
    {
       float angle, x2 = 77, y2 = 37, radius = 2;
       for (angle = 0; angle<= 360; angle++)
       glVertex2f(x2+sin(angle)*radius, y2+cos(angle)*radius);
    }
    glEnd();
    //العين الجزء الأسود
    glColor3f(0,0,0);
        glBegin(GL_POLYGON);
    {
       float angle, x2 = 77, y2 = 37, radius = 1;
       for (angle = 0; angle<= 360; angle++)
       glVertex2f(x2+sin(angle)*radius, y2+cos(angle)*radius);
    }
    glEnd();
    //جسم الطائر
```



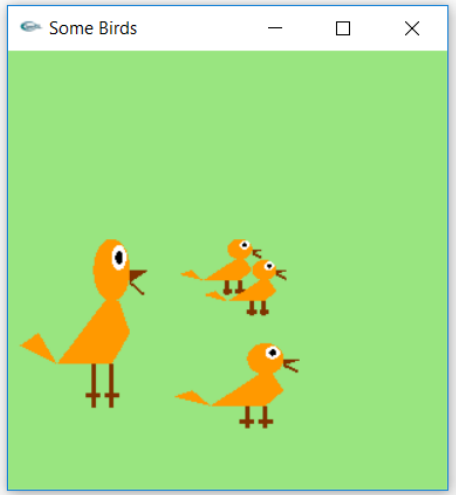
Put Multiple Images in One Window Using View Port Functions Program

ViewPort.cpp [MultipleImagesInOneWindowUsingViewPort] - Code::Blocks 17.12

Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

MultipleImagesInOneWindowUsingViewPort.cpp

```
32      //جسم الطائر
33      glColor3f(1, 0.6, 0);
34      glBegin(GL_POLYGON);
35      glVertex2f(73,30);
36      glVertex2f(77,30);
37      glVertex2f(80,25);
38      glVertex2f(75,20);
39      glVertex2f(60,20);
40      glEnd();
41      //الذيل
42      glColor3f(1, 0.6, 0);
43      glBegin(GL_TRIANGLES);
44      glVertex2f(55,25);
45      glVertex2f(50,23);
46      glVertex2f(60,20);
47      glEnd();
48      //المنقار
49      glColor3f(0.5, 0.2, 0.0);
50      glLineWidth(2);
51      glBegin(GL_LINES);
52      glVertex2f(80, 33);
53      glVertex2f(84, 31);
54      glEnd();
55      glColor3f(0.5, 0.2, 0.0);
56      glBegin(GL_TRIANGLES);
57      glVertex2f(80,33);
58      glVertex2f(80,35);
59      glVertex2f(85,35);
60      glEnd();
61      //أرجل الطائر
62      glLineWidth(3);
63      glBegin(GL_LINES);
```



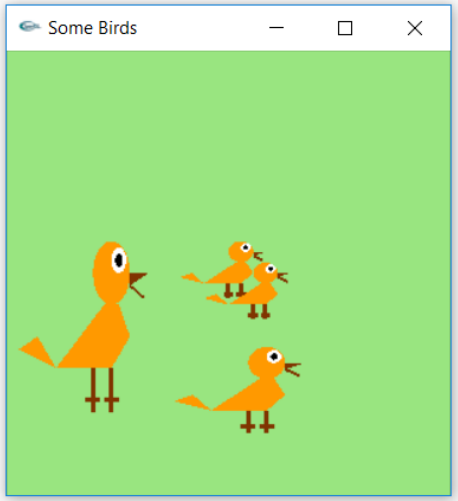
Put Multiple Images in One Window Using View Port Functions Program

ViewPort.cpp [MultipleImagesInOneWindowUsingViewPort] - Code::Blocks 17.12

Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

MultipleImagesInOneWindowUsingViewPort.cpp

```
61          //أسمك بالطلب
62          glLineWidth(3);
63          glBegin(GL_LINES);
64          glColor3f(0.5, 0.2, 0.0);
65          glVertex2i(75,20);
66          glVertex2i(75,13);
67          glVertex2i(75,15);
68          glVertex2i(77,15);
69          glVertex2i(75,15);
70          glVertex2i(73,15);
71          glVertex2i(70,20);
72          glVertex2i(70,13);
73          glVertex2i(70,15);
74          glVertex2i(72,15);
75          glVertex2i(70,15);
76          glVertex2i(68,15);
77        glEnd();
78
79          }
80  void init() {
81  glClearColor(0.6,0.9,0.5,0);
82  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
83  glMatrixMode(GL_PROJECTION);
84  glLoadIdentity();
85  glOrtho(40, 100, -10, 60, -20, 20);
86  glMatrixMode(GL_MODELVIEW);
87  //للنافذة الأولى
88  glViewport(0,0,100,200);
89  glLoadIdentity();
90  glTranslatef(-30, -10, 0);
91  glScalef(1.5, 1.5, 1.5);
92  Display();
```



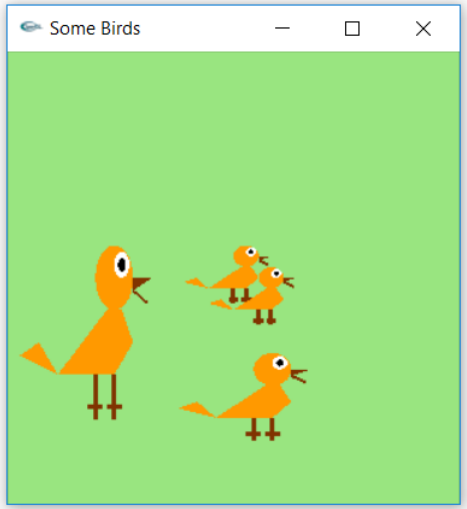
Put Multiple Images in One Window Using View Port Functions Program

ViewPort.cpp [MultipleImagesInOneWindowUsingViewPort] - Code::Blocks 17.12

Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

MultipleImagesInOneWindowUsingViewPort.cpp

```
84  glLoadIdentity();
85  glOrtho(40, 100, -10, 60, -20, 20);
86  glMatrixMode(GL_MODELVIEW);
87  //النافذة الأولى
88  glViewport(0,0,100,200);
89  glLoadIdentity();
90  glTranslatef(-30, -10, 0);
91  glScalef(1.5, 1.5, 1.5);
92  Display();
93  //النافذة الثانية
94  glViewport(100,0,100,100);
95  glLoadIdentity();
96  glTranslatef(-27, 0, 0);
97  glScalef(1.5, 1.5, 1.5);
98  Display();
99  //النافذة الثالثة
100 glViewport(100,100,100,100);
101 glLoadIdentity();
102 Display();
103 glTranslatef(10, -10, 0);
104 Display();
105 glutSwapBuffers();
106 }
107
108 int main() {
109 glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGB);
110 glutInitWindowSize(300, 300);
111 glutInitWindowPosition(100, 100);
112 glutCreateWindow("Some Birds");
113 glutDisplayFunc(init);
114 glutMainLoop();
115 }
```



Exercise

- **Using OpenGL, draw a brown Teapot with size = 5, then perform the following:**
  - **Rotate the Teapot 150 degree aroud y axis**
  - **Change the color of the Rotated Teapot to Orange.**
  - **Display only the Rotated Teapot in Window.**

وصلى الله وبارك على نبينا محمد

The End Summary of Lecture Eight

T.Mariah Khayat
الأستاذة/ مارية خياط
Adham University College
الكلية الجامعية بأضم

mskhayat@uqu.edu.sa