

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

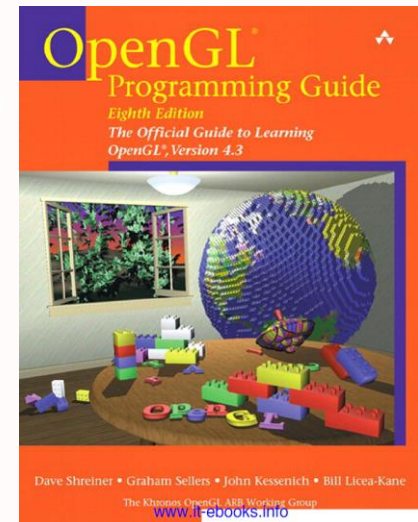
# References

- 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.

Computer Graphics  
Course, 3-6803430

LAB

T.Mariah Khayat



Kingdom of Saudi Arabia

المملكة العربية السعودية

Ministry of Education

Umm AlQura University

جامعة أم القرى

Adham University College

الكلية الجامعية بأضم

Computer Science Department

قسم الحاسب الآلي

# Lecture Four

## Drawing Triangles, Quads and Polygons using OpenGL

Computer Graphics  
Course, 3-6803430

LAB

T.Mariah Khayat

# content

1. Attributes in OpenGL
2. Triangles in OpenGL
3. Draw Triangles using OpenGL
4. Draw Triangle Fan using OpenGL
5. Quads in OpenGL
6. Draw a Square using OpenGL
7. Polygons in OpenGL
8. Draw an Ark Using OpenGL

# Attributes in OpenGL

- **Attributes:** are part of the OpenGL and determine the appearance of objects.
- **Color** deal with all objects (**points, lines, triangles, quads, polygons**).
- Each color component stored separately in the frame buffer.
- Usually 8 bits per component in buffer.
- Note: in **glColor3f** the color values range from 0.0 to 1.0 , while in **glColor3ub** the values range from 0 to 255.

# Attributes in OpenGL

CS

Department

- **Size** and **width** (points, lines).
- **Stipple** pattern (lines, polygons).
- Polygon mode:
  - Display as filled: (solid color) use the:
    - ❖ `glPolygonMode (GL_FRONT_AND_BACK, GL_FILL) ;`
  - Display edges use the functions:
    - ❖ `glPolygonMode (GL_FRONT_AND_BACK, GL_LINE) ;`

T.Mariah Khayat

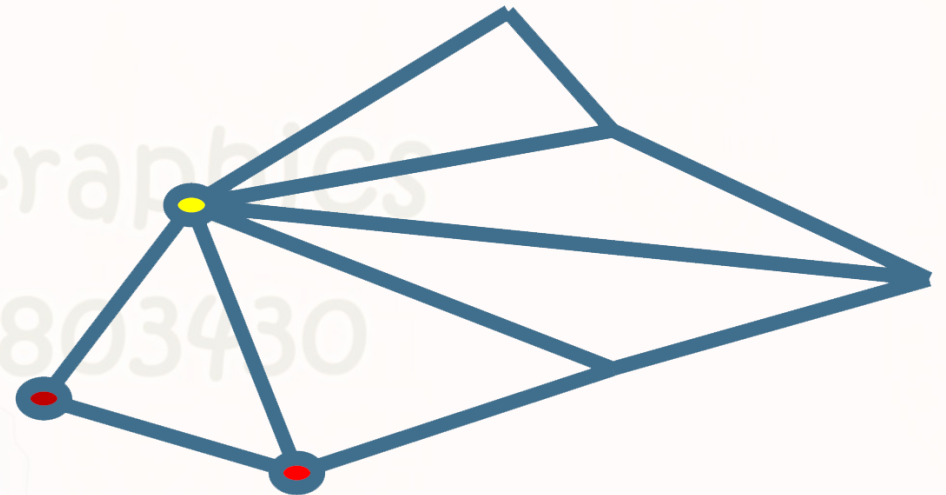
# Triangles in OpenGL

- **There are 3 functions** with Triangle: {S | \_STRIP | \_FAN}.
- **Triangle (`GL_TRIANGLES`):**
  - to draw normal Triangle. & Represent triangle as 3 vertices.
- **Triangle Strip : (`GL_TRIANGLE_STRIP`):**
  - Group of triangles sharing 2 vertices from previous triangle.
  - Use triangles to represent a solid object as a mesh.
  - Triangles frequently appear in strips:



# Triangles in OpenGL

- **Triangular Fans: (`GL_TRIANGLE_FAN`):**
  - Connected group sharing 1 common vertex, and 1 from previous triangle.



- **For tri-strips and fans,**
  - A new triangle is defined by 1 new vertex added to the strip or fans.



# Draw Triangles using OpenGL

CS

Department

Computer Graphics  
Course, 3-6803430

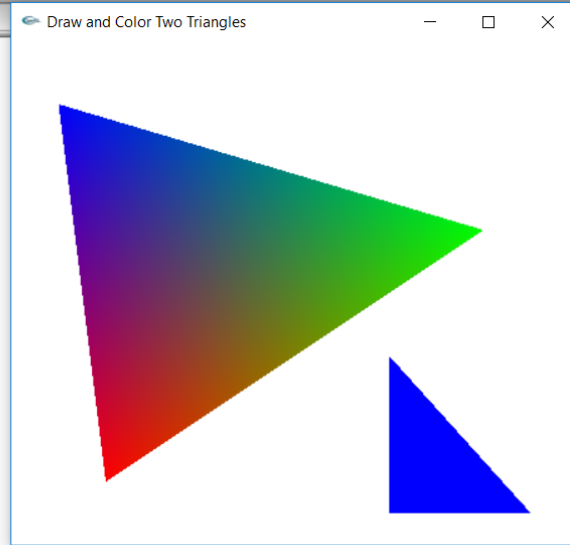
LAB

T.Mariah Khayat

```
#include<windows.h>
#include<GL/glut.h>
void Triangle() {
glClear(GL_COLOR_BUFFER_BIT);
glLineWidth(14);
glBegin(GL_TRIANGLES);
//First Triangle
glColor3f(1, 0, 0);
glVertex2f(100, 100);
glColor3f(0, 1, 0);
glVertex2f(500, 500);
glColor3f(0, 0, 1);
glVertex2f(50, 700);
//Second Triangle
glVertex2f(400, 50);
glVertex2f(550, 50);
glVertex2f(400, 300);
glEnd();
glFlush();
}
void Initial() {
glClearColor(1, 1, 1, 0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho2D(0, 600, 0, 800);
}
int main() {
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(450, 400);
glutInitWindowPosition(0, 0);
glutCreateWindow("Draw and Color Two Triangles");
Initial();
glutDisplayFunc(Triangle);
glutMainLoop();
return 0; }
```

# Draw Triangles using OpenGL

```
Blocks 17.12
Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
*Triangles.cpp
1 #include<windows.h>
2 #include<GL/glut.h>
3 void Triangle() {
4   glClear(GL_COLOR_BUFFER_BIT);
5   glLineWidth(14);
6   glBegin(GL_TRIANGLES);
7   //First Triangle
8   glColor3f(1, 0, 0);
9   glVertex2f(100, 100);
10  glColor3f(0, 1, 0);
11  glVertex2f(500, 500);
12  glColor3f(0, 0, 1);
13  glVertex2f(50, 700);
14  //Second Triangle
15  glVertex2f(400, 50);
16  glVertex2f(550, 50);
17  glVertex2f(400, 300);
18  glEnd();
19  glFlush();
20 }
21 void Initial() {
22   glClearColor(1, 1, 1, 0);
23   glMatrixMode(GL_PROJECTION);
24   glLoadIdentity();
25   gluOrtho2D(0, 600, 0, 800);
26 }
27 int main() {
28   glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
29   glutInitWindowSize(450, 400);
30   glutInitWindowPosition(0, 0);
31   glutCreateWindow("Draw and Color Two Triangles");
32   Initial();
33   glutDisplayFunc(Triangle);
34   glutMainLoop();
35   return 0; }
Logs & others
esters\Summer Semester 2019\Computer Graphics\CG Course LAB Le\C++ Windows (CR+LF) default Line 1, Col 20, Pos 19 Insert Modified Read/Write default
```



Kingdom of Saudi Arabia

المملكة العربية السعودية

Ministry of Education

# Draw Triangle Fan using OpenGL

```
#include<windows.h>
#include<GL/glut.h>
void Triangle_Fan() {
glClear(GL_COLOR_BUFFER_BIT);
glLineWidth(16);
glBegin(GL_TRIANGLE_FAN);
//Triangle Fan
glColor3f(1, 0, 0);
glVertex2f(300, 400);

glColor3f(1, 0.5, 0.5);
glVertex2f(550, 400);
glVertex2f(500, 600);

glColor3f(0.5, 0.3, 0.5);
glVertex2f(400, 700);

glColor3f(1, 0.5, 0.5);
glVertex2f(300, 700);

glColor3f(0.5, 0.3, 0.5);
glVertex2f(200, 700);

glColor3f(1, 0.5, 0.5);
glVertex2f(100, 600);

glColor3f(0.5, 0.3, 0.5);
glVertex2f(50, 400);

glColor3f(1, 0.5, 0.5);
glVertex2f(100, 200);

glEnd();
glFlush();
}
void Initial() {
glClearColor(1.0, 0.9, 0.9, 0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho2D(0, 600, 0, 800);
}
int main() {
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(400, 400);
glutInitWindowPosition(0, 0);
glutCreateWindow("Draw Triangle Fan");
Initial();
glutDisplayFunc(Triangle_Fan);
glutMainLoop();
return 0; }
```

CS

Department

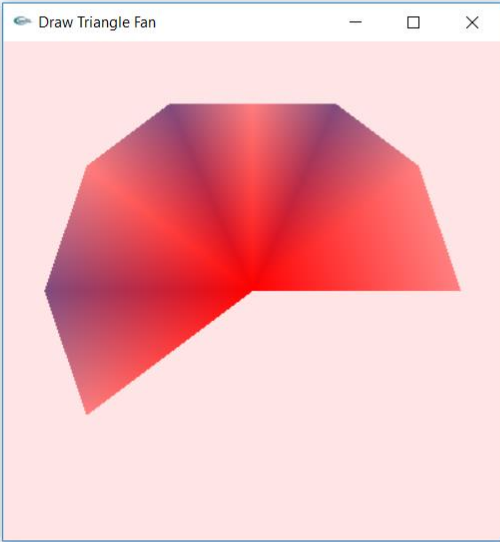
Computer Graphics  
Course, 3-6803430

LAB

T.Mariah Khayat

# Draw Triangle Fan using OpenGL

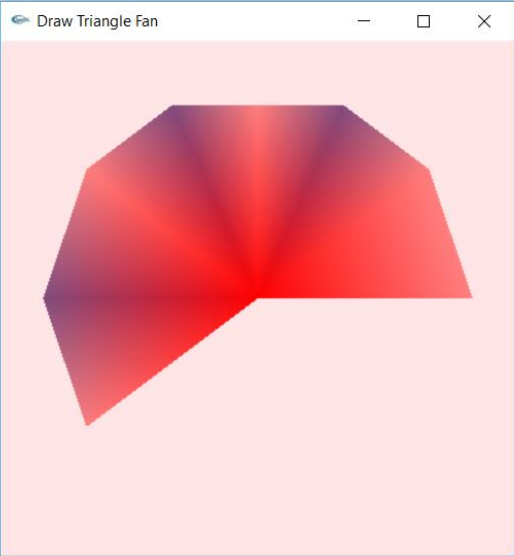
```
ide::Blocks 17.12
Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
Fan_Triangle.cpp
1 #include<windows.h>
2 #include<GL/glut.h>
3 void Triangle_Fan() {
4   glClear(GL_COLOR_BUFFER_BIT);
5   glLineWidth(16);
6   glBegin(GL_TRIANGLE_FAN);
7   //Triangle Fan
8   glColor3f(1, 0, 0);
9   glVertex2f(300, 400);
10
11  glColor3f(1, 0.5, 0.5);
12  glVertex2f(550, 400);
13  glVertex2f(500, 600);
14
15  glColor3f(0.5, 0.3, 0.5);
16  glVertex2f(400, 700);
17
18  glColor3f(1, 0.5, 0.5);
19  glVertex2f(300, 700);
20
21  glColor3f(0.5, 0.3, 0.5);
22  glVertex2f(200, 700);
23
24  glColor3f(1, 0.5, 0.5);
25  glVertex2f(100, 600);
26
27  glColor3f(0.5, 0.3, 0.5);
28  glVertex2f(50, 400);
29
30  glColor3f(1, 0.5, 0.5);
31  glVertex2f(100, 200);
32
33  glEnd();
34  glFlush();
35 }
36 void Initial() {
```



```
Logs & others
C/C++ Windows (CR+LF) default Line 3, Col 21, Pos 62 Insert Read/Write default
```

# Draw Triangle Fan using OpenGL

```
ode::Blocks 17.12
Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
Fan_Triangle.cpp
17
18 glColor3f(1, 0.5, 0.5);
19 glVertex2f(300, 700);
20
21 glColor3f(0.5, 0.3, 0.5);
22 glVertex2f(200, 700);
23
24 glColor3f(1, 0.5, 0.5);
25 glVertex2f(100, 600);
26
27 glColor3f(0.5, 0.3, 0.5);
28 glVertex2f(50, 400);
29
30 glColor3f(1, 0.5, 0.5);
31 glVertex2f(100, 200);
32
33 glEnd();
34 glFlush();
35 }
36 void Initial() {
37     glClearColor(1.0, 0.9, 0.9, 0);
38     glMatrixMode(GL_PROJECTION);
39     glLoadIdentity();
40     gluOrtho2D(0, 600, 0, 800);
41 }
42 int main() {
43     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
44     glutInitWindowSize(400, 400);
45     glutInitWindowPosition(0, 0);
46     glutCreateWindow("Draw Triangle Fan");
47     Initial();
48     glutDisplayFunc(Triangle_Fan);
49     glutMainLoop();
50     return 0; }
51
```



# Quads using OpenGL

- **There are 2 functions** with Quad: {**S** | **\_STRIP**}.
- **Quad :(GL\_QUADS):**
  - To draw normal Quad & Represent as 4 vertices.
  - You can draw more than one quad by determine 4 vertex of each Quad.



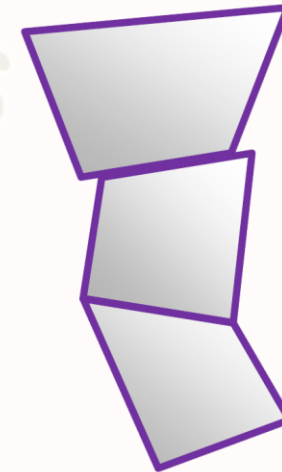
Quad list

# Quads using OpenGL

- **Quad Strip : (GL\_QUAD\_STRIP):**

- Group of Quads sharing 2 vertices from previous Quad.

Quad strip



2 vertices per quad

# Draw a Square using OpenGL

```
#include<windows.h>
#include<GL/glut.h>
void Quad() {
glClear(GL_COLOR_BUFFER_BIT);
glLineWidth(20);
glBegin(GL_QUADS);
glColor3f(0.8, 0.6, 0.6);
glVertex2f(200, 200);
glVertex2f(500, 200);
glColor3f(1, 0, 0);
glVertex2f(500, 500);
glVertex2f(200, 500);
glEnd();
glFlush();
}
void Initial() {
glClearColor(1, 1, 1, 0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho2D(0, 600, 0, 800);
}
int main() {
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(400, 400);
glutInitWindowPosition(0, 0);
glutCreateWindow("Draw a Quad");
Initial();
glutDisplayFunc(Quad);
glutMainLoop();
return 0; }
```

CS  
Department

Computer Graphics  
Course, 3-6803430

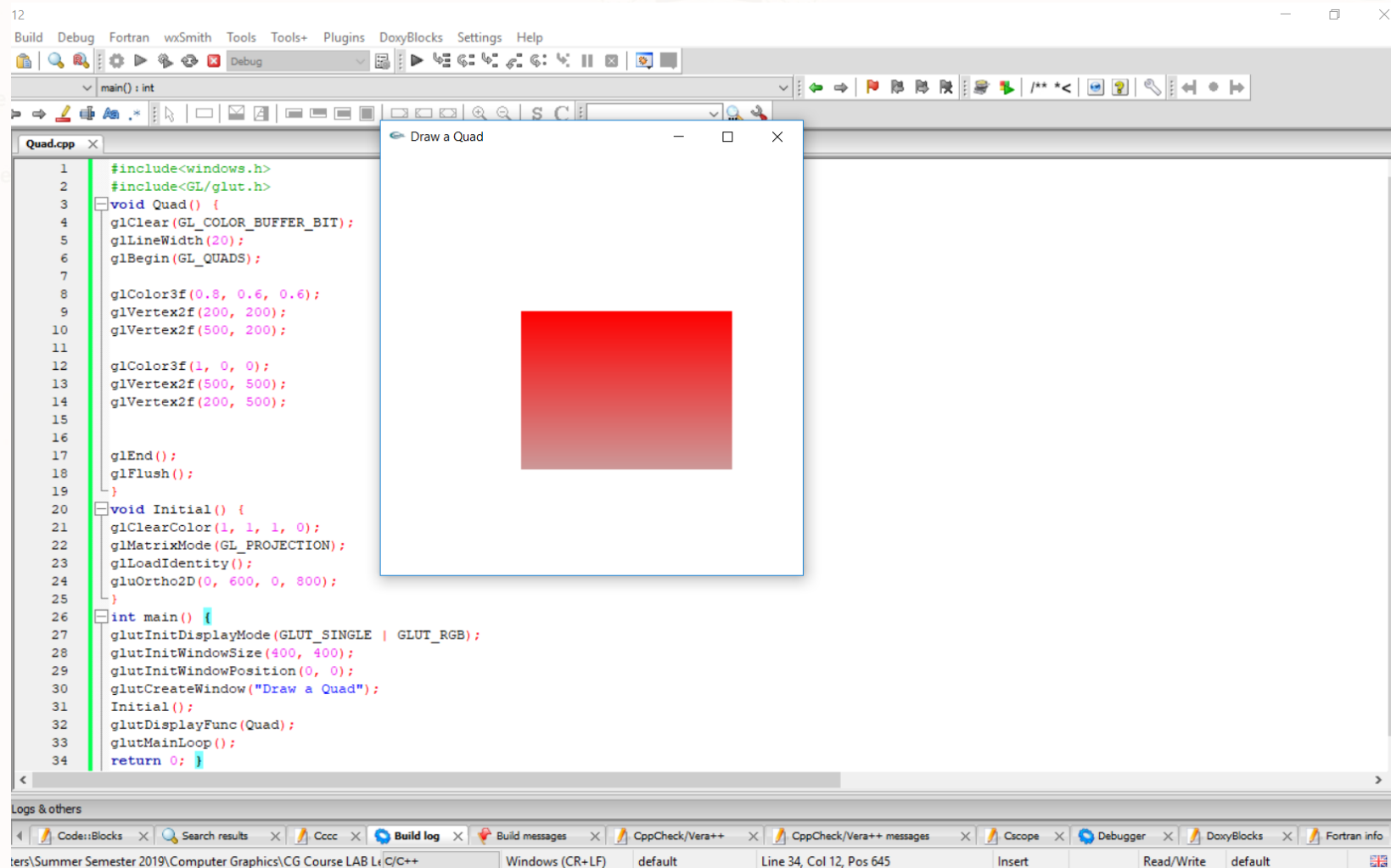
LAB

T.Mariah Khayat



# Draw a Square using OpenGL

```
12
Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main(): int
Quad.cpp
1 #include<windows.h>
2 #include<GL/glut.h>
3 void Quad() {
4   glClear(GL_COLOR_BUFFER_BIT);
5   glLineWidth(20);
6   glBegin(GL_QUADS);
7
8   glColor3f(0.8, 0.6, 0.6);
9   glVertex2f(200, 200);
10  glVertex2f(500, 200);
11
12  glColor3f(1, 0, 0);
13  glVertex2f(500, 500);
14  glVertex2f(200, 500);
15
16
17  glEnd();
18  glFlush();
19 }
20 void Initial() {
21  glClearColor(1, 1, 1, 0);
22  glMatrixMode(GL_PROJECTION);
23  glLoadIdentity();
24  gluOrtho2D(0, 600, 0, 800);
25 }
26 int main() {
27  glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
28  glutInitWindowSize(400, 400);
29  glutInitWindowPosition(0, 0);
30  glutCreateWindow("Draw a Quad");
31  Initial();
32  glutDisplayFunc(Quad);
33  glutMainLoop();
34  return 0; }
```

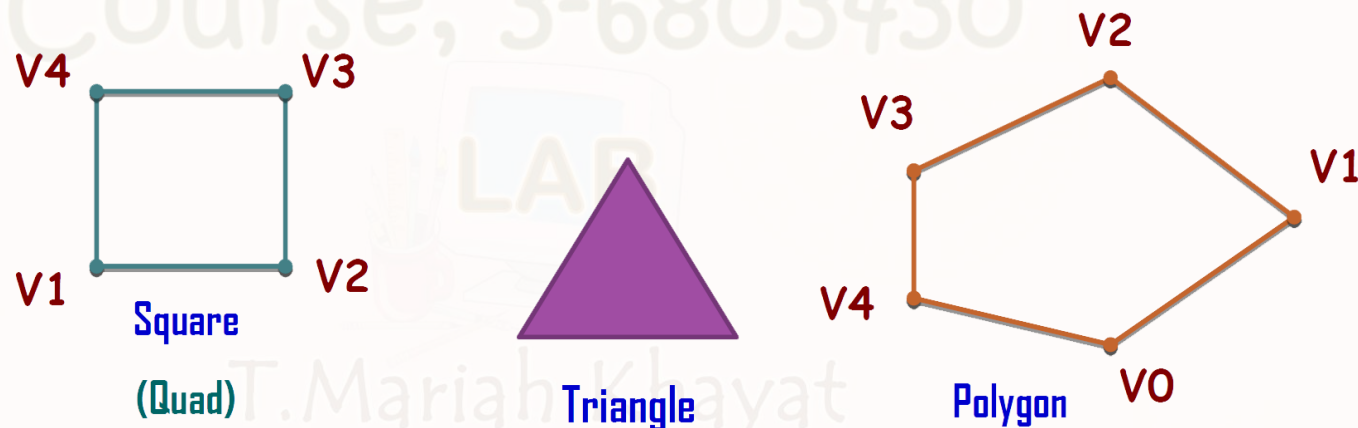


Logs &amp; others

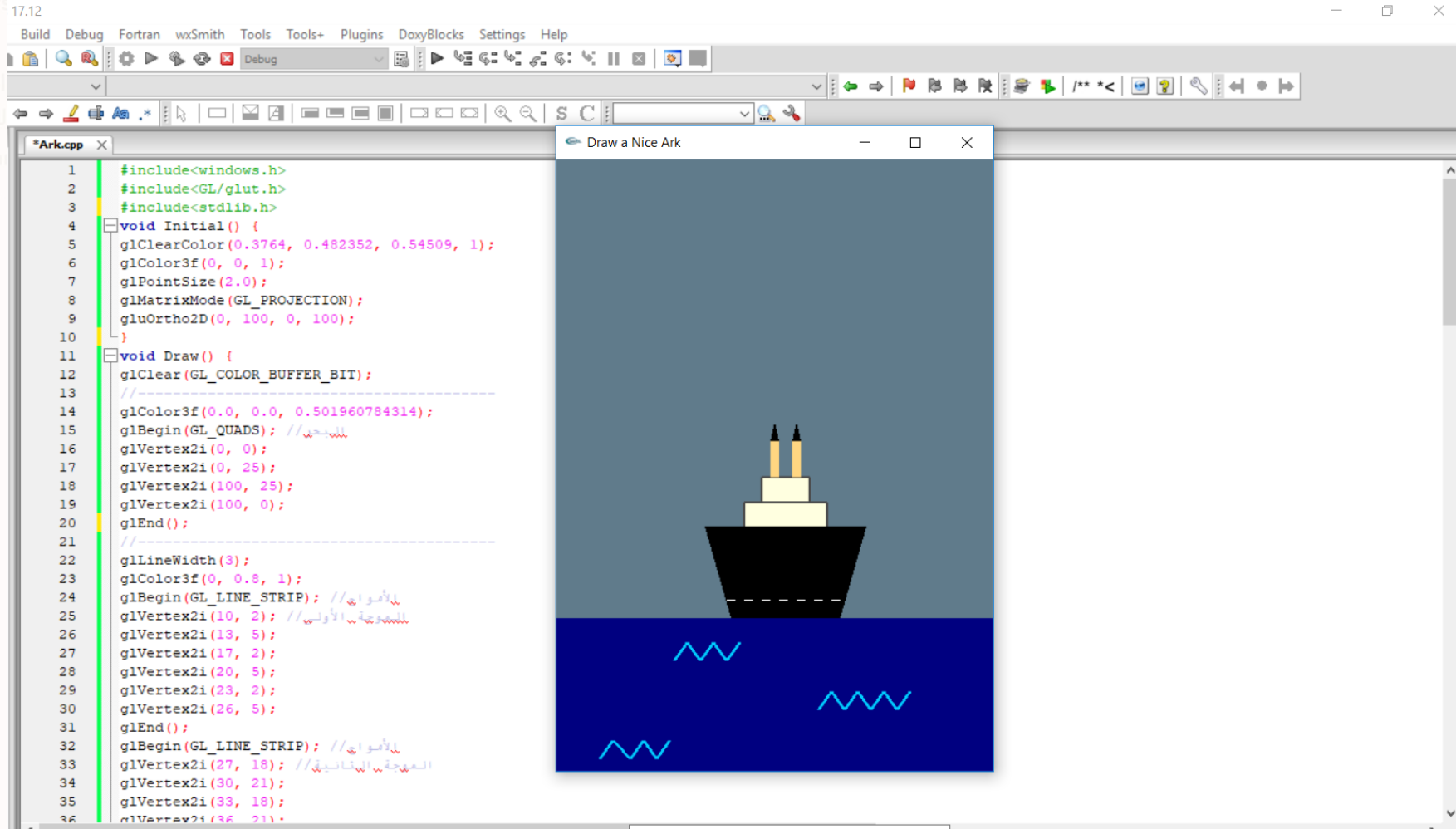
Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger DoxyBlocks Fortran info  
ters\Summer Semester 2019\Computer Graphics\CG Course LAB Le\C\C++ Windows (CR+LF) default Line 34, Col 12, Pos 645 Insert Read/Write default

# Polygons in OpenGL

- **A Polygon:** is a 2D shape that is made up of a number of vertices.
- An ordered set of vertices defines a polygon.
- You can also set other properties of the polygon, such as its color.
- **Quadrilateral & Triangle** is a special case of polygon.
- Use **GL\_POLYGON** argument to draw polygon by determine its vertexes.



# Draw Ark using OpenGL



# Draw Ark using OpenGL

```
17.12
Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
*Ark.cpp x Draw a Nice Ark
31 glEnd();
32 glBegin(GL_LINE_STRIP); // للأموال
33 glVertex2i(27, 18); // الموجة الثانية
34 glVertex2i(30, 21);
35 glVertex2i(33, 18);
36 glVertex2i(36, 21);
37 glVertex2i(39, 18);
38 glVertex2i(42, 21);
39 glEnd();
40 glBegin(GL_LINE_STRIP); // للأموال
41 glVertex2i(60, 10); // الموجة الثالثة
42 glVertex2i(63, 13);
43 glVertex2i(66, 10);
44 glVertex2i(69, 13);
45 glVertex2i(72, 10);
46 glVertex2i(75, 13);
47 glVertex2i(78, 10);
48 glVertex2i(81, 13);
49 glEnd();
50 //-----
51 glColor3f(0,0,0);
52 glBegin(GL_POLYGON); // الجزء السفلي للسفينة
53 glVertex2i(40, 25);
54 glVertex2i(34, 40);
55 glVertex2i(71, 40);
56 glVertex2i(65, 25);
57 glEnd();
58 //-----
59 glColor3f(1,1,0.878431372549);
60 glBegin(GL_QUADS); // منتصف السفينة
61 glVertex2i(43, 40);
62 glVertex2i(43, 44);
63 glVertex2i(62, 44);
64 glVertex2i(62, 40);
65 glEnd();
66
```

# Draw Ark using OpenGL

The image shows a code editor window titled "Ark.cpp" with the following code:

```
66 //-----
67 glLineWidth(2);
68 glColor3f(0.3, 0.3, 0.3); //المستطيق من جسد الهيكل
69 glBegin(GL_LINES);
70 glVertex2i(43, 40);
71 glVertex2i(43, 44);
72
73 glVertex2i(43, 44);
74 glVertex2i(62, 44);
75
76 glVertex2i(62, 40);
77 glVertex2i(62, 44);
78 glEnd();
79 //-----
80 glColor3f(1, 1, 0.878431372549);
81 glBegin(GL_QUADS); //الجزء العلوي
82 glVertex2i(47, 44);
83 glVertex2i(47, 48);
84 glVertex2i(58, 48);
85 glVertex2i(58, 44);
86 glEnd();
87 //-----
88 glColor3f(0.3, 0.3, 0.3);
89 glBegin(GL_LINES); //المستطيق من جسد الجزء العلوي
90 glVertex2i(47, 44);
91 glVertex2i(47, 48);
92
93 glVertex2i(47, 48);
94 glVertex2i(58, 48);
95
96 glVertex2i(58, 48);
97 glVertex2i(58, 44);
98
99 glVertex2i(47, 44);
100 glVertex2i(58, 44);
101 glEnd();
```

The rendered scene in the "Draw a Nice Ark" window shows a black ark with a yellow cabin and two black smokestacks. The ark is on a dark blue sea with light blue wavy lines representing water. The background is a light blue sky.

# Draw Ark using OpenGL

The image shows a code editor window titled "Ark.cpp" with the following code:

```
101 glEnd();
102 //-----
103 glColor3ub(255, 214, 129);
104 glBegin(GL_QUADS); //المستطيل المعنق للهدجق
105
106 glVertex2f(49.0, 48.1); //المستطيل الأول
107 glVertex2i(49, 54);
108 glVertex2i(51, 54);
109 glVertex2f(51.0, 48.1);
110
111 glVertex2f(54.0, 48.1); //المستطيل الثاني
112 glVertex2f(54, 54);
113 glVertex2f(56, 54);
114 glVertex2f(56.0, 48.1);
115 glEnd();
116 //-----
117 glColor3f(0,0,0); //أس الهديجق الهديجق
118 glBegin(GL_TRIANGLES);
119
120 glVertex2f(49, 54); //المثلث الأول
121 glVertex2f(50, 57);
122 glVertex2f(51, 54);
123
124 glVertex2f(54, 54); //المثلث الثاني
125 glVertex2f(55, 57);
126 glVertex2f(56, 54);
127 glEnd();
128 //-----
129 glEnable(GL_LINE_STIPPLE); //رسم الخط الهديجق
130 glLineWidth(1.0);
131 glColor3f(1.0, 1.0, 1.0);
132 glLineStipple(1, 0x00FF);
133 glBegin(GL_LINES);
134 glVertex2i(39, 28);
135 glVertex2i(66, 28);
136 glEnd();
```

The rendered scene in the "Draw a Nice Ark" window shows a black Ark with two yellow masts on a dark blue sea with white wavy lines representing water. The background is a light blue sky.

# Draw Ark using OpenGL

```
17:12
Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
*Ark.cpp x Draw a Nice Ark
115 glEnd();
116 //-----
117 glColor3f(0,0,0); // ملء الهيكلتة السوداء
118 glBegin(GL_TRIANGLES);
119
120 glVertex2f(49, 54); // الهيكلتة الأولى
121 glVertex2f(50, 57);
122 glVertex2f(51, 54);
123
124 glVertex2f(54, 54); // الهيكلتة الثانية
125 glVertex2f(55, 57);
126 glVertex2f(56, 54);
127 glEnd();
128 //-----
129 glEnable(GL_LINE_STIPPLE); // رسم الخطب النقطية
130 glLineWidth(1.0);
131 glColor3f(1.0, 1.0, 1.0);
132 glLineStipple(1, 0x00FF);
133 glBegin(GL_LINES);
134 glVertex2i(39, 28);
135 glVertex2i(66, 28);
136 glEnd();
137 glDisable(GL_LINE_STIPPLE);
138 //-----
139 glFlush();
140 }
141 int main() {
142     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
143     glutInitWindowSize(400, 560);
144     glutInitWindowPosition(100, 100);
145     glutCreateWindow("Draw a Nice Ark");
146     Initial();
147     glutDisplayFunc(Draw);
148     glutMainLoop();
149     return 0; }
```

# Excercise

- **Draw a Square that satisfies the following specifications:**
  - **Line Width = 10.**
  - **Vertex1 = (200, 200), Vertex2 = (200, 600), Vertex3 = (600, 600), Vertex4 = (200, 600).**
  - **Color of the Square = Orange.**
  - **Background Color = White.**
  - **Window Title Bar = “My Orange Square”.**



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

ومصلى الله وبارك على نبينا محمد

## The End Summary of Lecture Four

T.Mariah Khayat

الأستاذة/ مارية خياط

Adham University College

الكلية الجامعية بأضم

[mskhayat@uqu.edu.sa](mailto:mskhayat@uqu.edu.sa)

T.Mariah Khayat