

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

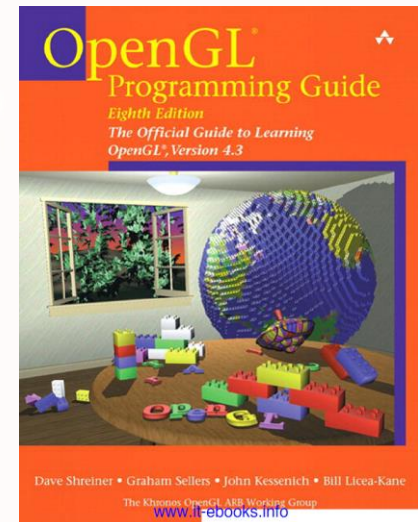
References

- 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 Three

Drawing Lines using OpenGL

Computer Graphics
Course, 3-6803430



T.Mariah Khayat

content

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

1. Line

2. Draw a Line using OpenGL

3. Stipple Line

4. Draw a stipple line using OpenGL

5. Draw a strip line using OpenGL

T.Mariah Khayat

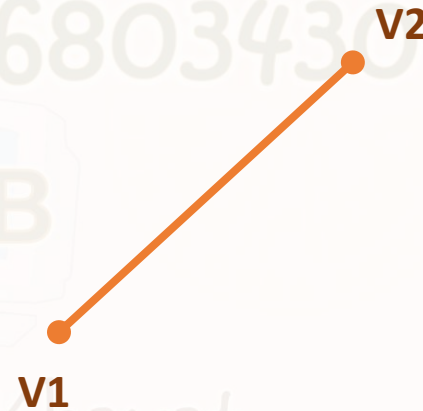
CS
Department

Computer Graphics

Course: 3-6803430

Lines

- **A Line:** is a straight line in space.
- It is defined by its two end-points.
- Each end-point is called a **vertex**.
- Each vertex has its own 3-D/2-D position defined by its 3-D/2-D coordinates.
- We draw lines by sampling at intervals of one pixel and drawing the closest pixels.



Drawing a Line Using OpenGL

```
#include<windows.h>
#include<GL/glut.h>
void DrawLine(){
    glClear(GL_COLOR_BUFFER_BIT);
    glLineWidth(8);
    glBegin(GL_LINES);
    glVertex2d(400, 700);
    glVertex2d(100, 50);
    glEnd();
    glFlush();
}
void Initial(){
    glClearColor(0.0, 1.0, 1.0, 0.0);
    glColor3f(0,1,0);
    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 Line");
    Initial();
    glutDisplayFunc(DrawLine);
    glutMainLoop();
    return 0;
}
```



CS
Department

Computer Graphics
Course, 3-6803430



T.Mariah Khayat

Drawing a Line Using OpenGL

The screenshot displays the Code::Blocks IDE interface. The main editor window shows the source code for 'Draw a Line.cpp'. The code includes headers for windows.h and GL/glut.h, and defines a DrawLine() function that clears the buffer, sets line width, and draws a line from (400, 700) to (100, 50). The Initial() function sets the background color to cyan and the projection mode. The main() function initializes GLUT and enters the main loop.

```
1 #include<windows.h>
2 #include<GL/glut.h>
3 void DrawLine(){
4   glClear(GL_COLOR_BUFFER_BIT);
5   glLineWidth(8);
6   glBegin(GL_LINES);
7   glVertex2d(400, 700);
8   glVertex2d(100, 50);
9   glEnd();
10  glFlush();
11 }
12 void Initial(){
13   glClearColor(0.0, 1.0, 1.0, 0.0);
14   glColor3f(0,1,0);
15   glMatrixMode(GL_PROJECTION);
16   glLoadIdentity();
17   gluOrtho2D(0, 600, 0, 800);
18 }
19 int main(){
20   glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
21   glutInitWindowSize(400, 400);
22   glutInitWindowPosition(0,0);
23   glutCreateWindow("Draw a Line");
24   Initial();
25   glutDisplayFunc(DrawLine);
26   glutMainLoop();
27   return 0;
28 }
```

The preview window titled 'Draw a Line' shows a cyan background with a thick red diagonal line extending from the top-right towards the bottom-left.

The bottom status bar shows the current file is 'C/C++', the window title is 'Windows (CR+LF)', and the cursor is at 'Line 6, Col 16, Pos 125'.

Stipple Line

- To determine the type /stipple of line use 3 functions:

1. `glEnable(GL_LINE_STIPPLE);`
2. `glLineStipple(GLint factor, GLushort pattern);`
3. `glDisable(GL_LINE_STIPPLE);`

○ **First:** `glEnable(GL_LINE_STIPPLE)` Function:

- To make active line stipple.

○ **Second:** `glLineStipple` Function:

- **Example:** `glLineStipple(2, 0x3f07);`

➤ Parameter 1: Factor:

*Specifies a multiplier for each bit in the line stipple pattern.

In above example factor =2 means: each bit in the pattern is used 2 times before the next bit in the pattern is used.

*Factor is clamped to the range [1, 256] and defaults to 1.

Stipple Line

o Second: `glLineStipple` Function:

➤ Parameter 2: Pattern:

*Specifies in hexadecimal transform to binary number (16-bit integer) whose bit pattern determines which fragments of a line will be drawn

*0 is OFF , 1 is ON

*Bit zero is used first; the default pattern is all 1s.

*pattern = `0x3f07` -----> `0011111100000111`

PATTERN	FACTOR	
<code>0x00FF</code>	1	_____
<code>0x00FF</code>	2	_____
<code>0x0C0F</code>	1	___ _ _ _
<code>0x0C0F</code>	3	_____
<code>0xAAAA</code>	1	-----
<code>0xAAAA</code>	2	___ _ _ _
<code>0xAAAA</code>	3	___ _ _ _
<code>0xAAAA</code>	4	___ _ _ _

Kingdom of Saudi Arabia

المملكة العربية السعودية

Ministry of Education

Stipple Line

Umm AlQura University

جامعة أم القرى

Adham University College

CS

Department

- **Third** : `glDisable(GL_LINE_STIPPLE)` Function:
 - To stop active (close) line stipple.

Computer Graphics
Course, 3-6803430

LAB

T.Mariah Khayat

Drawing Stipple Line Using OpenGL

```
#include<windows.h>
#include<GL/glut.h>
void DrawLine(){
    glClear(GL_COLOR_BUFFER_BIT);
    glEnable(GL_LINE_STIPPLE); //Determining the Line Type.
    glLineWidth(10);
    glLineStipple(2, 0XF2F0); //Factor and Pattern.
    glBegin(GL_LINES);
    glVertex2d(50, 200);
    glVertex2d(450, 200);
    glEnd();
    glDisable(GL_LINE_STIPPLE); //Disabling the Line Type.
    glFlush();
}
void Initial(){
    glClearColor(0.8, 1.0, 1.0, 0.0);
    glColor3f(1,0,0);
    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 Stipple Line");
    Initial();
    glutDisplayFunc(DrawLine);
    glutMainLoop();
    return 0;
}
```

CS
Department

Computer Graphics
Course, 3-6803430

LAB

T.Mariah Khayat

Drawing Stipple Line Using OpenGL



The screenshot displays the Code::Blocks IDE with a C++ project named "Draw a Stipple Line". The code in the editor is as follows:

```
1 #include<windows.h>
2 #include<GL/glut.h>
3 void DrawLine(){
4     glClear(GL_COLOR_BUFFER_BIT);
5     glEnable(GL_LINE_STIPPLE); //Determining the Line Type.
6     glLineWidth(10);
7     glLineStipple(2, 0XF2F0); //Factor and Pattern.
8     glBegin(GL_LINES);
9     glVertex2d(50, 200);
10    glVertex2d(450, 200);
11    glEnd();
12    glDisable(GL_LINE_STIPPLE); //Disabling the Line Type.
13    glFlush();
14 }
15 void Initial(){
16     glClearColor(0.8, 1.0, 1.0, 0.0);
17     glColor3f(1,0,0);
18     glMatrixMode(GL_PROJECTION);
19     glLoadIdentity();
20     gluOrtho2D(0, 600, 0, 800);
21 }
22 int main(){
23     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
24     glutInitWindowSize(400, 400);
25     glutInitWindowPosition(0,0);
26     glutCreateWindow("Draw a Stipple Line");
27     Initial();
28     glutDisplayFunc(DrawLine);
29     glutMainLoop();
30     return 0;
31 }
```

The IDE also shows a "Management" pane on the left with "Draw a Stipple Line" selected. At the bottom, a "Logs & others" pane shows the file path: G:\08.04.2019\Documents\UQU\Semesters\Summer Semester 2019\Computer Graphics\CG Course LAB L\C/C++\Lectures\CGLABLectures\CGLABLectureThree\Draw a Line\Draw a Stipple Line\.

On the right side of the IDE, a window titled "Draw a Stipple Line" is open, displaying a light blue background with a horizontal red stippled line. The line consists of a series of small red squares, demonstrating the effect of the `glLineStipple` function.

Drawing Strip Line Using OpenGL

```
#include<windows.h>
#include<GL/glut.h>
void DrawLine(){
    glClear(GL_COLOR_BUFFER_BIT);
    glLineWidth(4);
    glBegin(GL_LINE_STRIP); //Determining the Line Type.
    //Line One
    glColor3f(1,0,0); //Color of Line One
    glVertex2d(0,200);
    glVertex2d(100,600);
    //Line Two
    glColor3f(0,0,1); //Color of Line Two
    glVertex2d(200,200);
    glVertex2d(300,600);
    //Line Three
    glColor3f(0,1,0); //Color of Line Three
    glVertex2d(400,200);
    glVertex2d(500,600);
    //Line Four
    glColor3f(1, 0, 0); //Color of Line Four
    glVertex2d(600, 200);
    glVertex2d(700, 600);
    glEnd();
    glFlush();
}
void Initial(){
    glClearColor(1.0, 0.9, 0.9, 0.0);
    glLoadIdentity();
    gluOrtho2D(0, 600, 0, 800);
}
int main(){
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(0,0);
    glutCreateWindow("Draw a Strip Line");
    Initial();
    glutDisplayFunc(DrawLine);
    glutMainLoop();
    return 0;
}
```

Computer Graphics
Course, 3-6803430

T.Mariah Khayat

CS
Department

Drawing Strip Line Using OpenGL

CS

The screenshot displays a code editor window titled "Draw Strip Line.cpp [Draw a Strip Line] - Code::Blocks 17.12". The code defines a function `DrawLine()` that uses OpenGL to draw a zigzag line with four segments of different colors: red, blue, green, and red. The `Initial()` function sets the background to light pink and the window size to 600x800 pixels. A separate window titled "Draw a Strip Line" shows the rendered output, which is a zigzag line with four segments: red, blue, green, and red, set against a light pink background.

```
1 #include<windows.h>
2 #include<GL/glut.h>
3 void DrawLine(){
4   glClear(GL_COLOR_BUFFER_BIT);
5   glLineWidth(4);
6   glBegin(GL_LINE_STRIP); //Determining the Line Type.
7   //Line One
8   glColor3f(1,0,0); //Color of Line One
9   glVertex2d(0,200);
10  glVertex2d(100,600);
11  //Line Two
12  glColor3f(0,0,1); //Color of Line Two
13  glVertex2d(200,200);
14  glVertex2d(300,600);
15  //Line Three
16  glColor3f(0,1,0); //Color of Line Three
17  glVertex2d(400,200);
18  glVertex2d(500,600);
19  //Line Four
20  glColor3f(1, 0, 0); //Color of Line Four
21  glVertex2d(600, 200);
22  glVertex2d(700, 600);
23  glEnd();
24  glFlush();
25 }
26 void Initial(){
27   glClearColor(1.0, 0.9, 0.9, 0.0);
28   glLoadIdentity();
29   gluOrtho2D(0, 600, 0, 800);
30 }
```

Drawing Strip Line Using OpenGL

```
12  glColor3f(0,0,1); //Color of Line Two
13  glVertex2d(200,200);
14  glVertex2d(300,600);
15  //Line Three
16  glColor3f(0,1,0); //Color of Line Three
17  glVertex2d(400,200);
18  glVertex2d(500,600);
19  //Line Four
20  glColor3f(1, 0, 0); //Color of Line Four
21  glVertex2d(600, 200);
22  glVertex2d(700, 600);
23  glEnd();
24  glFlush();
25  }
26  void Initial(){
27      glClearColor(1.0, 0.9, 0.9, 0.0);
28      glLoadIdentity();
29      gluOrtho2D(0, 600, 0, 800);
30  }
31  int main(){
32      glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
33      glutInitWindowSize(400, 400);
34      glutInitWindowPosition(0,0);
35      glutCreateWindow("Draw a Strip Line");
36      Initial();
37      glutDisplayFunc(DrawLine);
38      glutMainLoop();
39      return 0;
40  }
41  }
```

The screenshot shows a code editor window titled "Draw Strip Line.cpp [Draw a Strip Line] - Code::Blocks 17.12". The code defines four lines with different colors and vertices. A window titled "Draw a Strip Line" displays the resulting zigzag line. The code is as follows:

```
12  glColor3f(0,0,1); //Color of Line Two
13  glVertex2d(200,200);
14  glVertex2d(300,600);
15  //Line Three
16  glColor3f(0,1,0); //Color of Line Three
17  glVertex2d(400,200);
18  glVertex2d(500,600);
19  //Line Four
20  glColor3f(1, 0, 0); //Color of Line Four
21  glVertex2d(600, 200);
22  glVertex2d(700, 600);
23  glEnd();
24  glFlush();
25  }
26  void Initial(){
27      glClearColor(1.0, 0.9, 0.9, 0.0);
28      glLoadIdentity();
29      gluOrtho2D(0, 600, 0, 800);
30  }
31  int main(){
32      glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
33      glutInitWindowSize(400, 400);
34      glutInitWindowPosition(0,0);
35      glutCreateWindow("Draw a Strip Line");
36      Initial();
37      glutDisplayFunc(DrawLine);
38      glutMainLoop();
39      return 0;
40  }
41  }
```

The window titled "Draw a Strip Line" shows a pink background with a zigzag line. The line starts at (200, 200), goes to (300, 600), then to (400, 200), then to (500, 600), then to (600, 200), then to (700, 600), and finally to (800, 200). The lines are colored as follows: Line 2 (red), Line 3 (blue), Line 4 (green), and Line 5 (red).

Exercise

- **Draw a Stipple Line Based on the following specifications:**

- **Line Width = 5.**
- **Factor = 1, Pattern = 0XAAAA.**
- **Vertex 1 = (200, 400) and Vertex 2 = (500, 400).**
- **Line Color = White.**
- **Background Color = Red.**
- **Window Title Bar = “My White Stipple Line”.**

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 Three

T.Mariah Khayat

الأستاذة/ مارية خياط

Adham University College

الكلية الجامعية بأضم

mskhayat@uqu.edu.sa

T.Mariah Khayat