

**Department of Computer Science &Engineering(CSE)**

**LAB - 6**

Name : Shah Ibne Fahad

Student ID : C193048

Semester : 7th

Section : 7BM

Email : c193048@ugrad.iiuc.ac.bd

Contact : 01860793742

Course Code : CSE-4742

Course Title : Computer Graphics Lab

Name of the course Teacher :

**Mahadi Hassan**

Associate Professor

Dept of Computer Science and Engineering,IIUC

1. Flood Fill algorithm & 2.Boundary Fill

#include <graphics.h>

#include <stdio.h>

using namespace std;

void flood(int x, int y, int new\_col, int old\_col)

{

if (getpixel(x, y) == old\_col)

{

putpixel(x, y, new\_col);

flood(x + 1, y, new\_col, old\_col);

flood(x - 1, y, new\_col, old\_col);

flood(x, y + 1, new\_col, old\_col);

flood(x, y - 1, new\_col, old\_col);

}

}

void flood\_algo()

{

int gd, gm = DETECT;

initgraph(&gd, &gm, "");

int top, left, bottom, right;

top = left = 50;

bottom = right = 100;

rectangle(left, top, right, bottom);

int x = 51;

int y = 51;

int newcolor = 14;

int oldcolor = 0;

flood(x, y, newcolor, oldcolor);

getch();

}

void boundaryFill4(int x, int y, int fill\_color,int boundary\_color)

{

if(getpixel(x, y) != boundary\_color &&

getpixel(x, y) != fill\_color)

{

putpixel(x, y, fill\_color);

boundaryFill4(x + 1, y, fill\_color, boundary\_color);

boundaryFill4(x, y + 1, fill\_color, boundary\_color);

boundaryFill4(x - 1, y, fill\_color, boundary\_color);

boundaryFill4(x, y - 1, fill\_color, boundary\_color);

}

}

void boundary\_algo()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

int x = 250, y = 200, radius = 50;

circle(x, y, radius);

boundaryFill4(x, y, 6, 15);

delay(10000);

getch();

closegraph();

}

int main()

{

//flood\_algo();

//boundary\_algo();

return 0;

}

1. Bitmap font

#include <graphics.h>

int bitmap\_B[12][12] = {

{0, 1, 1, 1, 1, 1, 0, 0,0, 0,0,0},

{0, 1, 0, 0, 0, 1, 0, 0,0, 0,0,0},

{0, 1, 0, 0, 0, 1, 0, 0,0, 0,0,0},

{0, 1, 0, 0, 0, 1, 0, 0,0, 0,0,0},

{0, 1, 0, 0, 0, 1, 0, 0,0, 0,0,0},

{0, 1, 0, 0, 0, 1, 0, 0,0, 0,0,0},

{0, 1, 1, 1, 1, 1, 0, 0,0, 0,0,0},

{0, 1, 0, 0, 0, 1, 0, 0,0, 0,0,0},

{0, 1, 0, 0, 0, 1, 0, 0,0, 0,0,0},

{0, 1, 0, 0, 0, 1, 0, 0,0, 0,0,0},

{0, 1, 0, 0, 0, 1, 0, 0,0, 0,0,0},

{0, 1, 1, 1, 1, 1, 0, 0,0, 0,0,0},

};

void draw\_char\_B( int x, int y, int color) {

for (int i = 0; i < 12; i++) {

for (int j = 0; j < 12; j++) {

if (bitmap\_B[i][j] == 1) {

putpixel(x + j, y + i, color);

}

}

}

}

int main() {

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

draw\_char\_B( 200, 200, WHITE);

getch();

closegraph();

return 0;

}

1. Outline font

#include <graphics.h>

// Define a bitmap font for the letter 'A'

int bitmap\_A[17][12] =

{

{1, 1,1, 1, 1, 1, 1, 1,1,1,1, 0},

{1, 0,0, 0, 0, 0, 0, 0,0,0,0, 1},

{1, 0,0, 0, 0, 0, 0, 0,0,0,0, 1},

{1, 0,0, 0, 0, 0, 0, 0,0,0,0, 1},

{1, 0,0, 0, 0, 0, 0, 0,0,0,0, 1},

{1, 0,0, 0, 0, 0, 0, 0,0,0,0, 1},

{1, 0, 1, 1, 1, 1, 1, 1,1,1,1, 0},

{1, 0, 1, 0, 0, 0, 0, 0,0,0,0, 0},

{1, 0,1, 0, 0, 0, 0, 0,0,0,0, 0},

{1, 0,1, 0, 0, 0, 0, 0,0,0,0, 0},

{1, 0,1, 0, 0, 0, 0, 0,0,0,0, 0},

{1, 0,1, 0, 0, 0, 0, 0,0,0,0, 0},

{1, 0,1, 0, 0, 0, 0, 0,0,0,0, 0},

{1, 0,1, 0, 0, 0, 0, 0,0,0,0, 0},

{1, 0,1, 0, 0, 0, 0, 0,0,0,0, 0},

{1, 0,1, 0, 0, 0, 0, 0,0,0,0, 0},

{1, 1,1, 0, 0, 0, 0, 0,0,0,0, 0}

};

void draw\_char\_A( int x, int y, int color)

{

for (int i = 0; i < 17; i++)

{

for (int j = 0; j < 12; j++)

{

if (bitmap\_A[i][j] == 1)

{

putpixel(x + j, y + i, color);

}

}

}

}

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

// Draw the letter 'A' at (100, 100) in red

draw\_char\_A( 100, 100, WHITE);

getch();

closegraph();

return 0;

}