**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

Факультет физико-математических и естественных наук

ОТЧЕТ

по лабораторной работе № 2

дисциплина: Компьютерная графика

Студент: Дорук Мерич Каймакджыоглу

Группа: НКНбд-01-20

**МОСКВА**

2022 г.

**Цел работу:** написать компьютерную программу для построения двумерных примитивов «Линия» и «Окружность». Программа должна соответствовать следующим требованиям:

**Ход работы:**

- Для рисования примитивов «Линия» и «Окружность» должны использоваться целочисленные алгоритмы Бразенхайма.

- За построение примитива «Линия» должна отвечать функция

void myline(int x1,int y1, int x2,int y2, int c), в которой

x1, y1 – растровые координаты начала линии,

x2, y2 – растровые координаты конца линии,

c – цвет линии;

*void* myline(*int* *x1*,*int* *y1*, *int* *x2*,*int* *y2*, *int* *c*){

*int* x = x1;

*int* y = y1;

*int* dx = abs(x2 - x1);

*int* dy = abs(y2 - y1);

*int* sx = x1<x2 ? 1 : -1;

*int* sy = y1<y2 ? 1 : -1;

*bool* flag = false;

   if (dy > dx){

*int* tmp = dx;

       dx = dy;

       dy = tmp;

       flag = true;

   }

*int* e = 2 \* dy - dx;

   for (*int* i = 1; i<=dx;i++){

       putpixel(x,y,c);

       while (e >= 0){

           if (flag)

               x = x + sx;

           else

               y = y + sy;

           e -= 2 \* dx;

       }

       if (flag)

           y = y + sy;

       else

           x = x + sx;

       e += 2 \* dy;

   }

}

- За построение примитива «Окружность» должна отвечать функция

void mycirc (int x0,int y0, int r, int c), в которой

x0, y0 – растровые координаты центра окружности,

r – радиус окружности,

c – цвет окружности;

*void* mycirc (*int* *x0*,*int* *y0*, *int* *r*, *int* *c*){

*int* x = 0;

*int* y = r;

*int* gap = 0;

*int* delta = 2 - 2 \* r;

   while (y >= 0){

       putpixel(x0 - x, y0 + y, c);

       putpixel(x0 + x, y0 + y, c);

       putpixel(x0 - x, y0 - y, c);

       putpixel(x0 + x, y0 - y, c);

       gap = 2 \* (delta + y) - 1;

       if (delta < 0 && gap <= 0){

           x++;

           delta += 2 \* x + 1;

       }

       else if (delta > 0 && gap > 0){

           y--;

           delta -= 2 \* y + 1;

       }

       else{

           x++;

           delta += 2 \* (x - y);

           y--;

       }

   }

}

- Функцию myline надо интегрировать в программу рисования дерева Пифагора.

// lines:

        myline(x0,y0,x1,y1, 2);

        myline(x0,y0,x3,y3, 2);

        myline(x1,y1,x2,y2, 2);

        myline(x2,y2,x3,y3, 2);

        myline(x3,y3,x4,y4, 2);

        myline(x2,y2,x4,y4, 2);

**-** инициализация графической библиотеки в main():

*int* main()

{

*int* gdriver = DETECT, gmode;

    initgraph(&gdriver,&gmode,(*char*\*)"");

    mycirc(200,200,25,2);

    mycirc(300,200,50,6);

    mycirc(300,200,20,8);

    pifagor(12,220,450,100,-pi/45.0,pi/3.5);

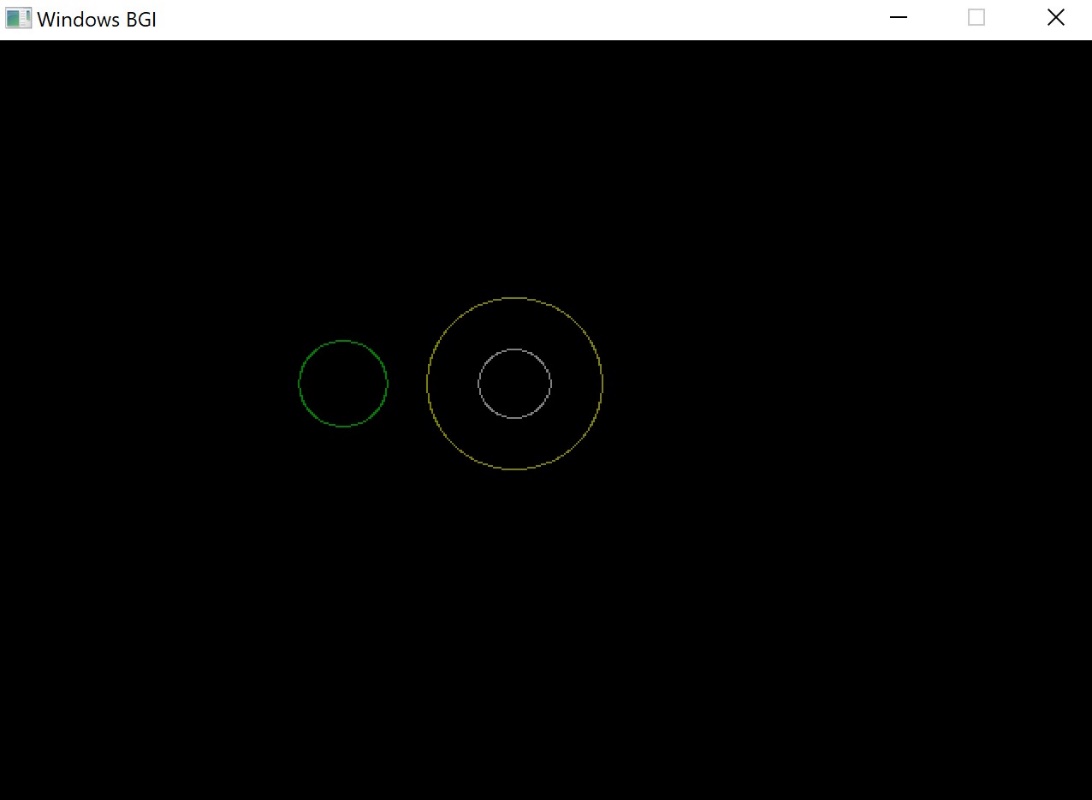
    getch();

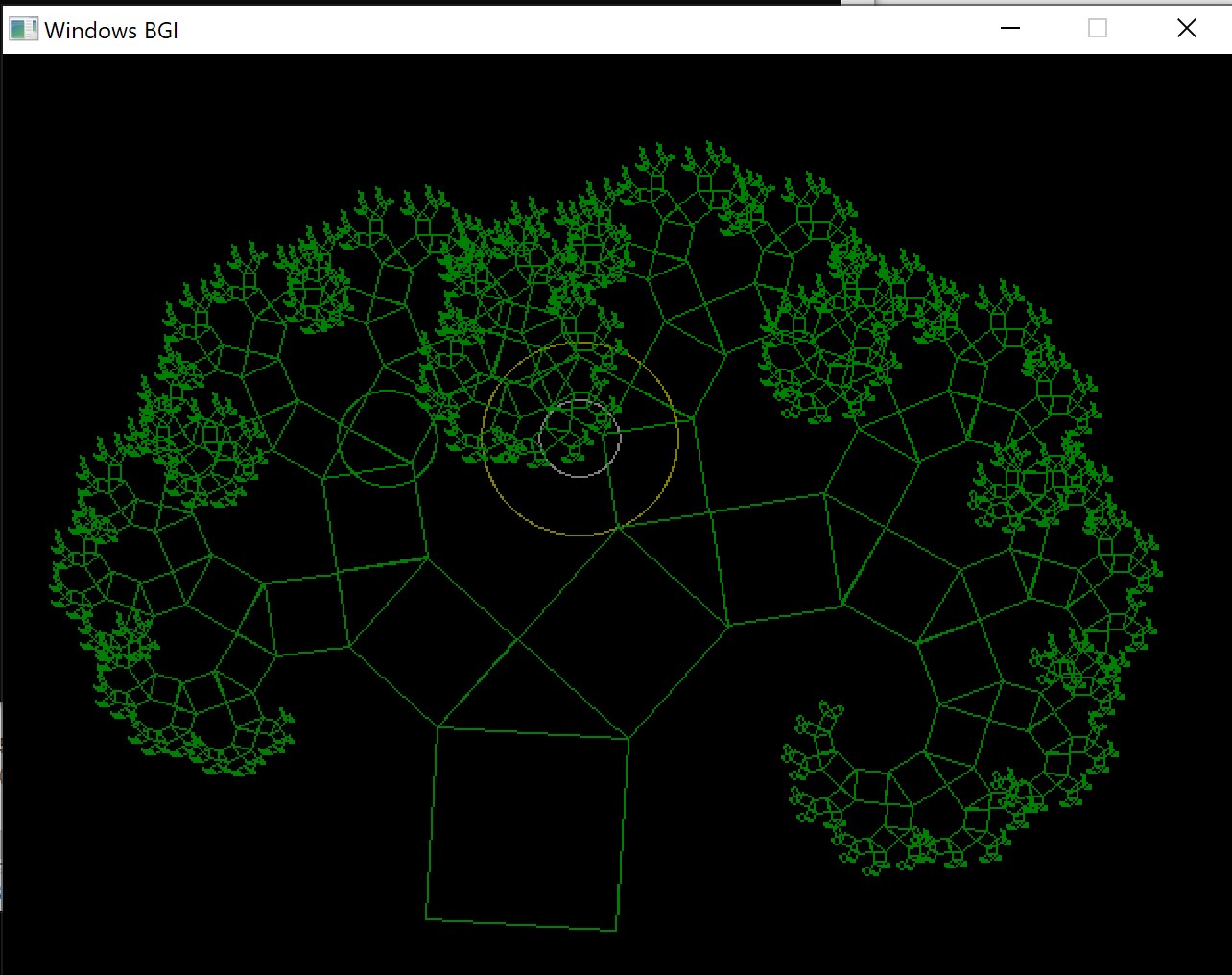
    closegraph();

    return 0;

}

**Вывод:**

****

****

**КОД:**

#include <iostream>

#include <conio.h>

#include "graphics.h"

#include <math.h>

using *namespace* std;

#define pi 3.141592

*void* mycirc (*int* *x0*,*int* *y0*, *int* *r*, *int* *c*){

*int* x = 0;

*int* y = r;

*int* gap = 0;

*int* delta = 2 - 2 \* r;

   while (y >= 0){

       putpixel(x0 - x, y0 + y, c);

       putpixel(x0 + x, y0 + y, c);

       putpixel(x0 - x, y0 - y, c);

       putpixel(x0 + x, y0 - y, c);

       gap = 2 \* (delta + y) - 1;

       if (delta < 0 && gap <= 0){

           x++;

           delta += 2 \* x + 1;

       }

       else if (delta > 0 && gap > 0){

           y--;

           delta -= 2 \* y + 1;

       }

       else{

           x++;

           delta += 2 \* (x - y);

           y--;

       }

   }

}

*void* myline(*int* *x1*,*int* *y1*, *int* *x2*,*int* *y2*, *int* *c*){

*int* x = x1;

*int* y = y1;

*int* dx = abs(x2 - x1);

*int* dy = abs(y2 - y1);

*int* sx = x1<x2 ? 1 : -1;

*int* sy = y1<y2 ? 1 : -1;

*bool* flag = false;

   if (dy > dx){

*int* tmp = dx;

       dx = dy;

       dy = tmp;

       flag = true;

   }

*int* e = 2 \* dy - dx;

   for (*int* i = 1; i<=dx;i++){

       putpixel(x,y,c);

       while (e >= 0){

           if (flag)

               x = x + sx;

           else

               y = y + sy;

           e -= 2 \* dx;

       }

       if (flag)

           y = y + sy;

       else

           x = x + sx;

       e += 2 \* dy;

   }

}

*void* pifagor(*int* *n*, *int* *x0*, *int* *y0*, *int* *a*, *double* *fi*, *double* *alpha*){

*int* x1,x2,x3,x4,y1,y2,y3,y4;

*double* b = a\*cos(alpha);

*double* c = a\*sin(alpha);

    if(n != 0){

        n--;

*int* color = 7;

        setcolor(color);

        // assigning x:

        x1 = (*int*) (x0 + a\*cos(fi));

        x2 = (*int*) (x1 - a\*sin(fi));

        x3 = (*int*) (x0 - a\*sin(fi));

        x4 = (*int*) (x3 + b\*cos(fi + alpha));

        // assigning y:

        y1 = (*int*) (y0 - a\*sin(fi));

        y2 = (*int*) (y1 - a\*cos(fi));

        y3 = (*int*) (y0 - a\*cos(fi));

        y4 = (*int*) (y3 - b\*sin(fi + alpha));

        // lines:

        myline(x0,y0,x1,y1, 2);

        myline(x0,y0,x3,y3, 2);

        myline(x1,y1,x2,y2, 2);

        myline(x2,y2,x3,y3, 2);

        myline(x3,y3,x4,y4, 2);

        myline(x2,y2,x4,y4, 2);

        // callback func of pifagor 2 times as asked:

        pifagor(n, x3, y3, (*int*)b, fi + alpha, alpha);

        pifagor(n, x4, y4, (*int*)c, (-pi/2+fi+alpha), alpha);

    }

    return;

}

*int* main()

{

*int* gdriver = DETECT, gmode;

    initgraph(&gdriver,&gmode,(*char*\*)"");

    mycirc(200,200,25,2);

    mycirc(300,200,50,6);

    mycirc(300,200,20,8);

    pifagor(12,220,450,100,-pi/45.0,pi/3.5);

    getch();

    closegraph();

    return 0;

}