Министерство образования и науки Российской Федерации Федеральное агентство по образованию

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №1 дисциплины

«Компьютерная графика»

Выполнил студент группы ИВТ-22 /Крючков И. С/ Проверил /Коржавина А. С./

Киров 2021

**Цель работы:**

# закрепить лекционный материал по изучению базовых алгоритмов компьютерной графики – разложению отрезков и окружностей в растр.

# Задание:

1. Написать на языке Pascal программу, реализующую алгоритмы построения прямой: простой пошаговый алгоритм и алгоритмы Брезенхема для четырех- и восьмисвязной развертки;
2. Проверить правильность работы программы, нарисовав, например, каждым алгоритмом семейство радиальных прямых, выходящих из одной точки с шагом 15 градусов;
3. Написать и отладить программу, реализующую два алгоритма построения окружности: по формуле Y=+-SQRT(r\*r-x\*x) и Брезенхема. В обоих случаях использовать свойство симметрии окружности (в первом - найдя точки четверти окружности, остальные - отразив симметрично; во втором - свойство симметрии использовать полностью);
4. Реализовать демонстрационный ролик (анимацию) движения объектов прямолинейно и по окружности.

**Теория:**

Разложение отрезка в растр - процесс последовательной инициализации множества пикселов экрана, составляющих отрезок, называется его растровой разверткой, а само это множество – растровым представлением отрезка.

Простой пошаговый алгоритм. Если увеличивать с определенным шагом координату Х, а затем находить координату *Y* используя уравнение прямой *Y=m\*X+b* и подкрашивать пиксел с координатами *(Х,ROUND(Y))*, то потребуется много времени (на нецелочисленные операции). Если шаг по Х принять равным единице, то *m=dY/dX* сводится к *m=dY*, т.е.изменение Х на единицу приведет к изменению углового коэффициента на *m*. Т.о., если *Х(I+1)=X(I)+1*, то Y(I+1)=Y(I)+m. Алгоритм корректно работает только для отрезков в первом и восьмом ок­тандах.

Целочисленные алгоритмы Брезенхема

Суть алгоритма в следующем: в процессе работы одна из координат либо х, либо y (в зависимости от углового коэффициента) изменяется на единицу. Изменение другой координаты (на 0 или 1) зависит от расстояния - (е) между действительным положением отрезка и ближайшими координатами растра (е назовем управляющей переменной). Алгоритм построен так, что на каждом шаге проверяется лишь знак е и корректируется ее значение после каждого изменения очередной координаты. Значение исходной управляющей переменной:

*е=2\*(y2-y1)-(x2-x1),*

где *x1, y1, x2, y2* – координаты начальной и конечной точек отрезка. В каждом шаге при *e>=0* значение y от предыдущего увеличивается на единицу, а е уменьшается на *2\*(x2-x1)*, в противном случае – y не меняется, а значение e увеличивается на *2\*(y2-y1)*. В обоих случаях координата х следующего пиксела увеличивается на единицу от предыдущего значения.

Преобразование окружности в растровую форму.

Существуют несколько простых способов преобразования окружности в растровую форму. Например, по формуле X\*X+Y\*Y=R\*R для окружности с центром в начале координат. Чтобы изобразить четверть такой ок ­ружности, на каждом шаге следует поменять Х от 0 до R на единицу и вычислить Y как SQRT(R\*R-X\*X). Остальные четверти изображают симметрично. Этот метод содержит операции умножения и извлечения корня, потому не эффективен.

Процесс можно улучшить, если вычислять одну восьмую часть окружности, а остальные семь частей отображать симметрично (в предыдущем случае Х менять от 0 до R/SQRT(2)). Но необходимый эффект можно получить только при работе с целыми числами (алгоритмами, подобными рассмотренным выше).

**Схемы алгоритма:**

****

Рисунок 1 – Схема простого пошагово алгоритма построения прямой.

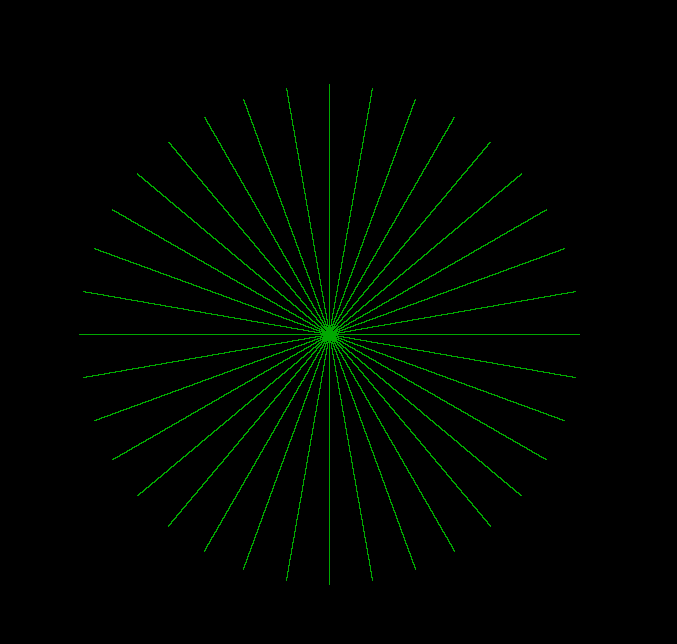
****

Рисунок 2 - Схема алгоритма Брезенхема для восьмисвязной развертки



Рисунок 3 - Схема алгоритма Брезенхема для четырехсвязной развертки.

**Экранная форма:**

****

**Листинг кода:**

procedure drawSimpleLine(x1, y1, x2, y2, col: integer);

var

x:integer;

m,y:real;

begin

if(x1 <> x2) then

begin

m:= (y2-y1)/(x2-x1);

y := y1;

for x := x1 to x2 do

begin

PutPixel(x, round(y), col);

y := y + m;

sleep(slp);

end;

end

else

begin

if(y1 = y2) then

begin

PutPixel(x1,y1,col);

sleep(slp);

end

else

writeln('Vertical');

end;

end;

procedure drawLineBrhm8(x1, y1, x2, y2, col: integer);

var

x,y,dx,dy,s1,s2,tmp, i, e:integer;

l:boolean;

begin

x:= x1;

y:= y1;

dx := abs(x2-x1);

dy := abs(y2-y1);

s1 := sign(x2-x1);

s2 := sign(y2-y1);

if(dy > dx) then

begin

tmp := dy;

dy := dx;

dx := tmp;

l := true;

end

else

l := false;

e := 2 \* dy - dx;

for i := 1 to dx do

begin

PutPixel(x,y,col);

sleep(slp);

while(e >= 0) do

begin

if(l = true) then

x := x + s1

else

y := y + s2;

e := e - 2 \* dx;

end;

if(l = true) then

y := y + s2

else

x := x + s1;

e := e + 2 \* dy;

end;

PutPixel(x, y, col);

sleep(slp)

end;

procedure drawLineBrhm4(x1, y1, x2, y2, col: integer);

var x, y, dx, dy, s1, s2, i, e: integer;

l: boolean;

begin

x := x1;

y := y1;

dx := abs(x2 - x1);

dy := abs(y2 - y1);

s1 := sign(x2 - x1);

s2 := sign(y2 - y1);

if (dy < dx) then

l := false

else

begin

dx := abs(y2 - y1);

dy := abs(x2 - x1);

l := true;

end;

e := 2 \* dy - dx;

for i := 1 to dx + dy do

begin

sleep(slp);

PutPixel(x, y, col);

if (e < 0) then

begin

if (l = true) then

y := y + s2

else

x := x + s1;

e := e + 2\*dy;

end

else

begin

if (l = true) then

x := x + s1

else

y := y + s2;

e := e - 2\*dx;

end;

end;

sleep(slp);

PutPixel(x, y, col);

end;

procedure drawCirle(x1, y1, rad, col: integer);

var x, y, e: integer;

begin

x := 0;

y := rad;

e := 3 - 2 \* rad;

while (x < y) do

begin

PutPixel(x1 + x, y1 + y, col);

PutPixel(x1 + y, y1 + x, col);

PutPixel(x1 + y, y1 - x, col);

PutPixel(x1 + x, y1 - y, col);

PutPixel(x1 - x, y1 - y, col);

PutPixel(x1 - y, y1 - x, col);

PutPixel(x1 - y, y1 + x, col);

PutPixel(x1 - x, y1 + y, col);

sleep(slp);

if (e < 0) then

e := e + 4 \* x + 6

else

begin

e := e + 4\*(x - y) + 10;

y := y - 1;

end;

x := x + 1;

end;

if (x = y) then

begin

PutPixel(x1 + x, y1 + y, col);

PutPixel(x1 + y, y1 + x, col);

PutPixel(x1 + y, y1 - x, col);

PutPixel(x1 + x, y1 - y, col);

PutPixel(x1 - x, y1 - y, col);

PutPixel(x1 - y, y1 - x, col);

PutPixel(x1 - y, y1 + x, col);

PutPixel(x1 - x, y1 + y, col);

sleep(slp);

end;

end;

**Вывод:** в ходе выполнения лабораторной работы был закреплен лекционный материал по изучению базовых алгоритмов компьютерной графики – разложению отрезков в растр. Реализованы алгоритмы построения прямой ­­– простой пошаговый, Брезенхема для четырехсвязной и восьмисвязной развертки