

Новосибирский государственный технический университет

Кафедра вычислительной техники



**Пояснительная записка к  
Расчётно-графической работе.**

**Вариант 28**

**«Использование графической библиотеки  
“Graphics.h”»**

Группа: АМ-810

Студент: Корнилов Д. Е.

Новосибирск  
2008.

## 1. Постановка задачи

**Задание:** визуализировать движение вращающегося квадрата. Квадрат должен перемещаться по экрану, совершая вращение. Периодически направление вращения меняется на противоположное, скорость вращения тоже изменяется, постепенно линейная скорость движения по экрану уменьшается и он падает на «пол».

## 2. Основная идея

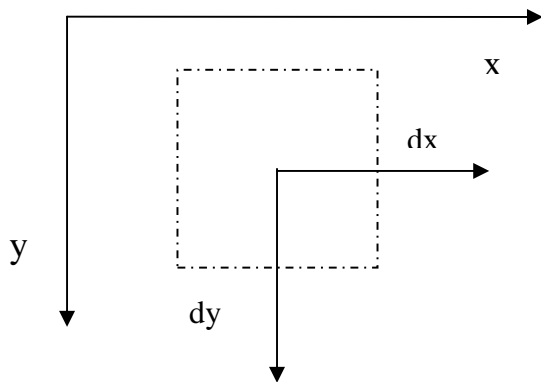
Для реализации движения квадрата необходимо решить следующие проблемы:

- Движение точки в системе координат относительно экрана.
- Вращение квадрата относительно перемещающейся точки и построение квадрата относительно перемещающейся точки.
- Необходимо предусмотреть все ситуации с изменением направления движения квадрата и отскока его от границ экрана.

Геометрическая модель:

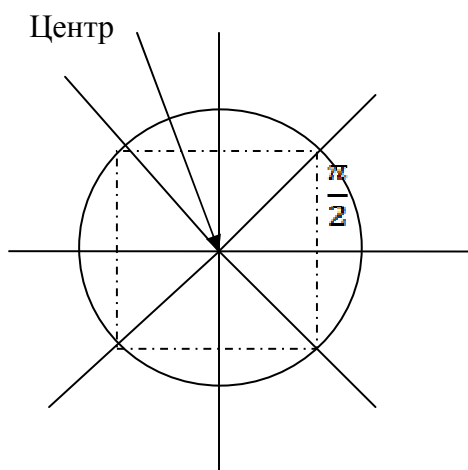
В процессе решения задачи задействовано 2 системы координат.

**Первая система координат:**



**Пояснение:** в этой системе координат происходит движение точки. (По определённым законам движения)

**Вторая система координат:**



Точка, вращающаяся вокруг центра движения

**Пояснение:** в системе координат , связанной с точкой движения, строятся 4 точки на окружности с поворотом на  $90^\circ$ .

### 3. Основные методы и решения

В соответствии с приведённой выше совокупностью систем координат реализуется выполнение поставленной задачи:

- 1) **Движение точки в системе координат относительно дисплея.** Первая система координат является базовой, в ней происходит определение принципов движения: вычисление координат текущего положения объекта, вычисление скорости, направления движения.
- 2) **Вращение квадрата относительно перемещающейся точки и построение квадрата относительно перемещающейся точки.** Во второй системе координат строится квадрат, относительно заданной начальными координатами точки на экране, задается половина диагонали квадрата, далее вычисляются четыре точки. Вращение квадрата осуществляется путём изменения угла поворота этих точек относительно центра. Полученные точки соединяются линиями и накладываются на первую систему координат.
- 3) **Контроль движения.** Слежение за процессом движения производится рядом условий, в которых задействованы координаты вершин квадрата. Условия необходимы для удержания фигуры в ограниченной на экране области. Также используется переменная, содержащая информацию о направлении движения, для реализации линейного ускорения и замедления.

### 4. Принцип работы

Прежде всего нужно определить переменные и их значение в процессе работы программы:

```
double x=30, y=15      // координаты центра относительно которого вращается
                        // квадрат
dx=2, dy=0             // приращения по соответствующим координатам
ay=0.1                 // приращение угла поворота
double FI=45,          // начальный угол поворота
int r=50               // половина диагонали квадрата
double X[5], Y[5]      // целочисленные массивы хранения координат квадрата
X[0], Y[0]             // максимальные координаты (размеры дисплея)
double OM, a           // угловая скорость и коэффициент изменения направления и
                        // скорости вращения квадрата
int xold, yold         // последние целочисленные координаты
```

Программа состоит из главного цикла в void main() , функции show() и hide():

- *show ()* – функция рисования квадрата по вычисленным точкам;
- *hide ()* – функция стирания квадрата на старых координатах.

### **Главный цикл**

В теле главного цикла выполняется: вычисление новых значений переменных, вычисление новых координат квадрата, вызов функций *show()* и *hide()* с задержкой между ними, проверка условия движения, определение направления движения.

```
void main()
{
double x=30,y=15,dx=2,dy=0,ay=0.1;      // определение переменных
double FI=25,OM=0,a=0;
double PI=3.1415926;
int xold=x,yold=y, r=50;

int gdriver = DETECT, gmode;
initgraph(&gdriver, &gmode, "e:\\bc31\\bgi");
setfillstyle(1,BLUE);

X[0]=getmaxx();                          // вычисление максимальных координат
Y[0]=getmaxy();
bar(1,1,X[0]-1,Y[0]-1);                  // задание ограниченной области экрана

while (!kbhit()){                         // пока нет нажатия клавиши цикл выполняется
dy+=ay;                                  // приращение скорости движения квадрата
x+=dx;                                    // приращение координат по x и y
y+=dy;
OM+=a;                                    // приращение угловой скорости
FI+=OM;                                   // приращение угла поворота
a=-0.00005*FI;                            // изменение коэффициента
if (dx<0 && x<=(r+2)) dx=-dx;             // условия на изменение направления
if (dx>0 && x>X[0]-(r+2)) dx=-dx;          // движения после столкновения с
if (dy<0 && y<=(r+2)) dy=-dy;             // границей экрана
if (dy>0 && y>Y[0]-(r+2)) dy=-dy*0.98;    // отскок с затуханием
if ((int)x!=xold || (int)y!=yold){
xold=x;
yold=y;
X[1]=xold+r*cos(FI);                      // вычисление координат вершин квадрата
Y[1]=yold+r*sin(FI);
X[2]=xold+r*cos(FI+PI/2);
Y[2]=yold+r*sin(FI+PI/2);
X[3]=xold+r*cos(FI+PI);
Y[3]=yold+r*sin(FI+PI);
X[4]=xold+r*cos(FI+3*PI/2);
Y[4]=yold+r*sin(FI+3*PI/2);
show();                                   // нарисовать квадрат
delay(10);
hide();                                   // стереть квадрат на старых координатах
```

```

if(X[1]>=X[0]){
if(OM<0) OM=OM*0.8;      // условие, что если угловая скорость будет
else OM=-OM;             // положительна ее умножить на коэффициент
}                         // как только она станет отрицательной
                          //поменять знак
}

```

### **Функция определения координат и прорисовки куба ( show() )**

В качестве параметров функция получает координаты вершин из глобально определенного массива и рисует по ним квадрат.

```

void show{                // рисует четыре линии по четырем точкам
setcolor (WHITE);        // цвет квадрата
line(X[1],Y[1],X[2],Y[2]);
line(X[2],Y[2],X[3],Y[3]);
line(X[3],Y[3],X[4],Y[4]);
line(X[4],Y[4],X[1],Y[1]);
}

```

### **Функция clean().**

Данная функция меняет параметр цвета рисования на фоновый и перерисовывает квадрат. Таким образом происходит стирание квадрата.

```

void hide(){              //перерисовывает квадрат другим цветом
setcolor (BLUE);
line(X[1],Y[1],X[2],Y[2]);
line(X[2],Y[2],X[3],Y[3]);
line(X[3],Y[3],X[4],Y[4]);
line(X[1],Y[1],X[4],Y[4]);
}

```