* [#include](https://vk.com/im?sel=206764991&st=%23include) <time.h>  
    
  [#define](https://vk.com/im?sel=206764991&st=%23define) size 15  
    
  using namespace sf;  
  using namespace std;  
    
  CircleShape triangle[size];  
    
  int main()  
    
  {  
    
  int lenght;  
  int width;  
  int freq;  
    
    
  int vx[size], vy[size], r[size], x[size], y[size], i, k;  
    
  int R[size], G[size], B[size];  
    
  setlocale(LC\_ALL, "Russian");  
    
  srand(time(NULL));  
    
  printf("Пожалуйста, введите длину и высоту окна: \n");  
    
  scanf("%d%d", &lenght, &width);  
  if (lenght<400){lenght=400;}  
  if (width<400){width=400;}  
    
  printf("Пожалуйста, введите частоту смены окна: \n");  
    
  scanf("%d", &freq);  
    
  printf("Введите необходимое количество треугольников на экране (от 5 до 15)\n");  
    
  scanf("%i", &k);  
  if (k<5){k=5;}  
  if (k>15){k=15;}  
    
  for (i = 0; i < k; i++)  
    
  {  
    
  r[i] = 10+ rand() % 80;  
    
  vx[i] = rand() % 21 + (-15);  
    
  vy[i] = rand() % 21 + (-15);  
    
  x[i] = rand() % (lenght - 2\*r[i]);  
    
  y[i] = rand() % (width - 2\*r[i]);  
  /\*for (int j=0;j<k;j++)  
  {  
  if (i==j) break;  
  while ( (x[i]<=(x[j]+2\*r[j]) ) || (y[i]<=(y[j]+2\*r[j]) ) )  
  {  
  r[i]=10+rand()%30;  
  x[i] = rand() % (lenght - 2\*r[i]);  
  y[i] = rand() % (width - 2\*r[i]);  
  }  
  while ( (vx[i]==vx[j]) && (vy[i]==vy[j]) )  
  {  
  r[i]=10+rand()%30;  
  vx[i] = rand() % 21 + (-10);  
  vy[i] = rand() % 21 + (-10);  
  }  
    
  }  
  \*/  
    
  triangle[i].setPosition(x[i], y[i]);  
    
  triangle[i].setPointCount(3);  
    
  triangle[i].setRadius(r[i]);  
    
  R[i] = rand() % 255;  
  G[i] = rand() % 255;  
  B[i] = rand() % 255;  
    
  triangle[i].setFillColor(Color(R[i], G[i], B[i], 255));  
  triangle[i].setOrigin(r[i], r[i]);  
    
  }  
    
    
  RenderWindow window(VideoMode( lenght, width), "Triangles");  
    
  window.setFramerateLimit(freq);  
    
  while (window.isOpen())  
    
  {  
    
  Event event;  
    
  while (window.pollEvent(event))  
    
  {  
    
  if (event.type == Event::Closed)  
    
  window.close();  
    
  }  
    
  window.clear();  
    
  for (i = 0; i < k; i++)  
    
  {  
    
  triangle[i].rotate(5);  
    
  window.draw(triangle[i]);  
    
    
  if ((triangle[i].getPosition().x - r[i] > lenght )  
    
  || (triangle[i].getPosition().x + r[i] <0 )  
  || (triangle[i].getPosition().y - r[i] > width )  
    
  || (triangle[i].getPosition().y + r[i] < 0 ))  
    
  { int p\_X;  
    
  p\_X=abs(triangle[i].getPosition().x-lenght);  
    
  int p\_Y;  
    
  p\_Y=abs(triangle[i].getPosition().y-width);  
    
  triangle[i].setPosition(p\_X,p\_Y);  
    
  }  
    
  triangle[i].move(vx[i], vy[i]);  
    
  }  
    
  for (i = 0; i < k; i++)  
    
  {  
    
  for (int j = i+1; j < k; j++)  
    
  {  
  int e1 = pow((triangle[i].getPosition().x+vx[i]-triangle[j].getPosition().x-vx[j]),2);  
  int e2 = pow((triangle[i].getPosition().y+vy[i]-triangle[j].getPosition().y-vy[j]),2);  
    
  if (sqrt(e1+e2) < (r[i]+r[j]))  
  //(triangle[i].getGlobalBounds().intersects(triangle[j].getGlobalBounds()))  
    
    
  {  
  triangle[i].setFillColor(Color::Red);  
  triangle[j].setFillColor(Color::Red);  
  printf("%d \t %d \n",i,j);  
    
  }  
    
  else  
  {  
  triangle[i].setFillColor(Color(R[i], G[i], B[i], 255));  
  triangle[j].setFillColor(Color(R[j], G[j], B[j], 255));  
    
  }  
  }  
  }  
    
  window.display();  
    
  }  
  return 0;  
  }