```
    1: #include <SFML/Graphics.hpp>
    2: #include <SFML/Window.hpp>
    3: #include <cmath>
    4: #include <math.h>
    5: #include <iostream>
    6: #include "func.hpp"
    7:
    8: using namespace sf ;
    9: using namespace std ;
   10:
   11: int Sierpinski( double side, double depth, double p2x, double p2y, sf::Rende
rWindow& a ) {
   12:   //reminder side = side/3
   13:   if(depth == 0) return 0 ;
   14:
   15:   double p0y, p0x, p1y, p1x, p3y, p3x;
   16:
   17:   double s = .866 ;
   18:   double c = .5 ;
   19:
   20:   //p0's values, left circ
   21:   p0y = p2y - side*2*s ;
   22:   p0x = p2x - side*2*c ;
   23:
   24:   //p1's values, right circ pt
   25:   p1y = p2y  ;
   26:   p1x = p2x + side*2 ;
   27:
   28:   //buttom circle
   29:   p3y = p2y + side*2*s ;  //side*2 - side*(1/3)  ;
   30:   p3x = p2x - side*2*c ;
   31:
   32:   CircleShape circle ;
   33:   circle.setRadius(side) ;
   34:   circle.setOrigin(side, side) ;
   35:   circle.setPosition( (float)p2x, (float)p2y) ;
   36:   circle.setFillColor(Color::Black);
   37:   a.draw(circle);
   38:
   39:   CircleShape cb ;
   40:   cb.setRadius(side) ;
   41:   cb.setOrigin(side,side) ;
   42:   cb.setPosition((float)p0x,(float)p0y) ;
   43:   cb.setFillColor(Color::Green);
   44:   cb.setOutlineThickness(1);
   45:   cb.setOutlineColor(sf::Color(250, 255, 255));
   46:   a.draw(cb);
   47:
   48:   CircleShape cc ;
   49:   cc.setRadius(side) ;
   50:   cc.setOrigin(side,side) ;
   51:   cc.setPosition(p1x, p1y) ;
   52:   cc.setFillColor(Color::Blue);
   53:   cc.setOutlineThickness(1);
   54:   cc.setOutlineColor(sf::Color(250, 255, 255));
   55:   a.draw(cc);
   56:
   57:   CircleShape cd ;
   58:   cd.setRadius(side) ;
   59:   cd.setOrigin(side,side) ;
   60:   cd.setPosition(p3x, p3y) ;
```

```
   61:   cd.setFillColor(Color::Red);
   62:   cd.setOutlineThickness(1);
   63:   cd.setOutlineColor(sf::Color(250, 255, 255));
   64:   a.draw(cd);
   65:   //p2's val, bottum circ pt
   66:
   67:   /*    if( depth == 1 ) {
   68:       cout << "trval test " << endl ;
   69:       cout << p0x << " "<< p0y << endl ;
   70:       cout << p1x << " "<< p1y << endl ;
   71:       cout << p2x << " "<< p2y << endl ;
   72:       cout << side  << endl ;
   73:     }
   74:   */
   75:
   76:   //Draw picture of middle tri

   77:   ConvexShape T ;
   78:   T.setPointCount(3) ;
   79:   T.setPoint(0, Vector2f(p2x + side,p2y));
   80:   T.setPoint(1, Vector2f(p2x - side*c,p2y - side*s));
   81:   T.setPoint(2, Vector2f(p2x - side*c,p2y + side*s));
   82:   T.setPosition(0,0);
   83:   T.setFillColor(Color::Cyan);
   84:   a.draw(T) ;
   85:
   86:   Sierpinski( side/3 , depth - 1, p0x, p0y, a ) ;
   87:   Sierpinski( side/3 , depth - 1, p1x, p1y, a ) ;
   88:   Sierpinski( side/3 , depth - 1, p3x, p3y, a ) ;
   89:
   90:   return 0 ;
   91:
   92: }
```