

```
1: #include <iostream>
2: #include <cmath>
3: #include <SFML/Graphics.hpp>
4: #include <SFML/Window.hpp>
5:
6: #include "original.hpp"
7:
8:
9: void Fractal::draw(sf::RenderTarget& target, sf::RenderStates states) const{
10:     target.draw(square, states);
11:     fractal_rec(square, depth, target);
12: }
13:
14: Fractal::Fractal(int N, int size_frac){
15:     side = size_frac;
16:     depth = N;
17:
18:     square.setPointCount(4);
19:     square.setPoint(0, sf::Vector2f(side/2, 0)); //top
20:     square.setPoint(1, sf::Vector2f(0, side/2)); //left
21:     square.setPoint(2, sf::Vector2f(side/2, side)); //bottom
22:     square.setPoint(3, sf::Vector2f(side, side/2)); //right
23:
24:     square.setFillColor(sf::Color::Red);
25:     square.setOutlineColor(sf::Color::White);
26:     square.setOutlineThickness(5);
27: }
28: void Fractal::fractal_rec(sf::ConvexShape fractal_shape, int recursion, sf::
RenderTarget &target) const{
29:     sf::Vector2f left,top,right,bottom;
30:     sf::Vector2f mid_lefttop, mid_topright, mid_rightbottom, mid_bottoml
eft;
31:
32:     sf::ConvexShape temp1_sqr, temp2_sqr, temp3_sqr, temp4_sqr;
33:
34:     if(recursion == 0){
35:         return;
36:     }
37:     else{
38:         top = fractal_shape.getPoint(0);
39:         left = fractal_shape.getPoint(1);
40:         bottom = fractal_shape.getPoint(2);
41:         right = fractal_shape.getPoint(3);
42:
43:         mid_lefttop.x = (left.x + top.x)/2;
44:         mid_lefttop.y = (left.y + top.y)/2;
45:
46:         mid_topright.x = (top.x + right.x)/2;
47:         mid_topright.y = (top.y + right.y)/2;
48:
49:         mid_rightbottom.x = (right.x + bottom.x)/2;
50:         mid_rightbottom.y = (right.y + bottom.y)/2;
51:
52:         mid_bottomleft.x = (left.x + bottom.x)/2;
53:         mid_bottomleft.y = (left.y + bottom.y)/2;
54:
55:         temp1_sqr = filledFractal(mid_lefttop, mid_topright, mid_rig
htbottom, mid_bottomleft, target);
56:
57:         fractal_rec(temp1_sqr, recursion - 1, target);
58:     }
```

```
59:
60: }
61: sf::ConvexShape Fractal::filledFractal(sf::Vector2f point1,sf::Vector2f poin
t2, sf::Vector2f point3, sf::Vector2f point4,sf::RenderTarget &target) const{
62:     sf::ConvexShape mini_square;
63:
64:     mini_square.setPointCount(4);
65:     mini_square.setPoint(0, point1);
66:     mini_square.setPoint(1, point2);
67:     mini_square.setPoint(2, point3);
68:     mini_square.setPoint(3, point4);
69:
70:     mini_square.setFillColor(sf::Color::Black);
71:     mini_square.setOutlineColor(sf::Color::White);
72:     mini_square.setOutlineThickness(5);
73:
74:     target.draw(mini_square);
75:
76:     return mini_square;
77:
78: }
79:
80: Fractal::~Fractal(){
81:
82: }
```