```
    1: //
    2: //  sierpinski.cpp
    3: //  ps1
    4: //
    5: //  Created by Jingxian Shi on 1/30/18.
    6: //  Copyright Â© 2018 Jingxian Shi. All rights reserved.
    7: //
    8:
    9: #include <iostream>
   10: #include <cmath>
   11: #include "sierpinski.hpp"
   12:
   13: Sierpinski::Sierpinski(int depth, double side)
   14: {
   15:     recursion_depth = depth;
   16:     double height = (double)((sqrt(3.)/2.0)*(double)side);
   17:     _top = sf::Vector2f(side/2, 0);
   18:     _left = sf::Vector2f(0, height);
   19:     _right = sf::Vector2f(side, height);
   20:
   21:     sf::Vector2f top_left = sf::Vector2f((_top.x+_left.x)/2, (_left.y+_top.y)/2);
   22:     sf::Vector2f top_right = sf::Vector2f((_right.x+_top.x)/2, (_right.y+_top.y)/
2);
   23:     sf::Vector2f bottom_mid = sf::Vector2f((_right.x+_left.x)/2, (_right.y+_left.
y)/2);
   24:
   25:     setPointCount(3);
   26:     setPoint(0, top_left);
   27:     setPoint(1, top_right);
   28:     setPoint(2, bottom_mid);
   29:     setFillColor(sf::Color::Black);
   30: }
   31:
   32: Sierpinski::Sierpinski(int depth, sf::Vector2f top, sf::Vector2f left, sf::Vector
2f right)
   33: {
   34:     recursion_depth = depth;
   35:     _top = top;
   36:     _left = left;
   37:     _right = right;
   38:
   39:     sf::Vector2f top_left = sf::Vector2f((_top.x+_left.x)/2, (_left.y+_top.y)/2);
   40:     sf::Vector2f top_right = sf::Vector2f((_right.x+_top.x)/2, (_right.y+_top.y)/
2);
   41:     sf::Vector2f bottom_mid = sf::Vector2f((_right.x+_left.x)/2, (_right.y+_left.
y)/2);
   42:
   43:     setPointCount(3);
   44:     setPoint(0, top_left);
   45:     setPoint(1, top_right);
   46:     setPoint(2, bottom_mid);
   47:     setFillColor(sf::Color::Black);
   48: }
   49:
   50: void Sierpinski::draw(sf::RenderTarget& target, sf::RenderStates states) const
   51: {
   52:     target.draw((sf::ConvexShape)(*this), states);
   53:     if(recursion_depth <= 0)
   54:     {
   55:         return;
   56:     }
   57:     else
   58:     {
   59:         Sierpinski top(recursion_depth-1,
   60:                        _top, //top
```

```
   61:                         sf::Vector2f((_top.x+_left.x)/2.0, (_left.y+_top.y)/2.0),
//bottom left
   62:                         sf::Vector2f((_right.x+_top.x)/2.0, (_right.y+_top.y)/2.0)
); //bottom right
   63:         top.draw(target, states);
   64:
   65:         Sierpinski left(recursion_depth-1,
   66:                         sf::Vector2f((_top.x+_left.x)/2.0, (_top.y+_left.y)/2.0),
   67:                         _left,
   68:                         sf::Vector2f((_left.x+_right.x)/2.0, (_left.y+_right.y)/2
.0));
   69:         left.draw(target, states);
   70:
   71:         Sierpinski right(recursion_depth-1,
   72:                          sf::Vector2f((_right.x+_top.x)/2, (_right.y+_top.y)/2),
   73:                          sf::Vector2f((_right.x+_left.x)/2, (_right.y+_left.y)/2)
,
   74:                          _right);
   75:         right.draw(target, states);
   76:     }
   77: }
   78:
   79:
```

```
    1: //
    2: //  sierpinski.hpp
    3: //  ps1
    4: //
    5: //  Created by Jingxian Shi on 1/30/18.
    6: //  Copyright Â© 2018 Jingxian Shi. All rights reserved.
    7: //
    8:
    9: #ifndef sierpinski_hpp
   10: #define sierpinski_hpp
   11:
   12: #include <stdio.h>
   13: #include <SFML/Graphics.hpp>
   14: #include <SFML/Graphics/ConvexShape.hpp>
   15: #include <iostream>
   16:
   17: using namespace std;
   18:
   19: class Sierpinski : public sf::ConvexShape
   20: {
   21: public:
   22:     Sierpinski(int depth, double side);
   23:     Sierpinski(int depth, sf::Vector2f top, sf::Vector2f left, sf::Vector2f right
);
   24:     virtual void draw(sf::RenderTarget& target, sf::RenderStates states) const;
   25: private:
   26:     int recursion_depth;
   27:     sf::Vector2f _top, _left, _right;
   28: };
   29:
   30: #endif /* sierpinski_hpp */
   31:
   32:
```