```
    1: #include "original.hpp"
    2: #include <SFML/Window.hpp>
    3: #include <cmath>
    4: #include <iostream>
    5:
    6: std::vector <sf::ConvexShape> square_vector;
    7: int count = 0;
    8:
    9: original::original(int size, int depth)
   10: {
   11:     float original_height = size/2;
   12:     //    original_depth = depth;
   13:     //   original_size = size;
   14:
   15:
   16:     sf::Vector2f p1, p2, p3, p4;
   17:     p1.x = size *.4;
   18:     p1.y = size *.35;
   19:     p2.x = size*.4;
   20:     p2.y = size*.55;
   21:     p3.x = size*.6;
   22:     p3.y = size*.55;
   23:     p4.x = size*.6;
   24:     p4.y = size*.35;
   25:
   26:     //Set Initial Triangle
   27:     sf::ConvexShape initial_square;
   28:     initial_square.setPointCount(4);
   29:     initial_square.setPoint(0, p1);
   30:     initial_square.setPoint(1, p2);
   31:     initial_square.setPoint(2, p3);
   32:     initial_square.setPoint(3, p4);
   33:     initial_square.setFillColor(sf::Color::Yellow);
   34:
   35:     square_vector.push_back(initial_square);
   36:
   37:     original(p1.x, p1.y, p2.x, p2.y, p3.x, p3.y, p4.x, p4.y, depth, size, or
iginal_height);
   38:
   39: }
   40:
   41:
   42: original::original(float x1, float y1, float x2, float y2, float x3, float y
3, float x4, float y4,
   43:                     int depth, float size, float height)
   44: {
   45:         /*if (depth not reached)
   46:          {
   47:          child (top, midleft (w/4) (h/2), midright (3w/4)(h/2))
   48:          child (midleft, left, middle (top w/2) (h))
   49:          child (midright, middle, right)
   50:          }
   51:          else
   52:          {
   53:          build triangle with current data
   54:          }
   55:          */
   56:
   57:
   58:
   59:         if (depth > 0)
```

```
   60:         {
   61:             depth--;
   62:
   63:             original(x1/4, y1/4, x2/4, y2/4, x3/4, y3/4, x4/4, y4/4, depth,
size, height);
   64:
   65:             original((x1+size*.5)/1.9, y1/4, (x2+size*.5)/1.9, y2/4, (x3+siz
e*.5)/2.1, y3/4, (x4+size*.5)/2.1, y4/4, depth, size, height);
   66:
   67:             original((x1 * 2.2), y1/4, (x2 * 2.2), y2/4, (x3 * 1.56), y3/4,
(x4 * 1.56), y4/4,
   68:                     depth, size, height);
   69:
   70:             original(x1/4, y1/4 + size*.63, x2/4, y2/4 + size*.63, x3/4, y3/
4 + size*.63, x4/4,
   71:                     y4/4 + size*.63, depth, size, height);
   72:
   73:             original((x1+size*.5)/1.9, y1/4 + size*.63, (x2+size*.5)/1.9, y2
/4 + size*.63, (x3+size*.5)/2.1, y3/4 + size*.63, (x4+size*.5)/2.1, y4/4 + size*.63
, depth, size, height);
   74:
   75:             original((x1 * 2.2), y1/4 + size*.63, (x2 * 2.2), y2/4 + size*.6
3, (x3 * 1.56), y3/4 + size*.63, (x4 * 1.56), y4/4 + size*.63, depth, size, height)
;
   76:
   77:             original(x1/4, y1/4 + size*.33, x2/4, y2/4 + size*.33, x3/4, y3/
4 + size*.33, x4/4,
   78:                     y4/4 + size*.33, depth, size, height);
   79:
   80:             original((x1 * 2.2), y1/4 + size*.33, (x2 * 2.2), y2/4 + size*.3
3, (x3 * 1.56), y3/4 + size*.33, (x4 * 1.56), y4/4 + size*.33, depth, size, height)
;
   81:
   82:
   83:         }
   84:         else
   85:         {
   86:             sf::Vector2f p1, p2, p3, p4;
   87:             p1.x = x1;
   88:             p1.y = y1;
   89:             p2.x = x2;
   90:             p2.y = y2;
   91:             p3.x = x3;
   92:             p3.y = y3;
   93:             p4.x = x4;
   94:             p4.y = y4;
   95:
   96:             //Set Initial Triangle
   97:             sf::ConvexShape initial_square;
   98:             initial_square.setPointCount(4);
   99:             initial_square.setPoint(0, p1);
  100:             initial_square.setPoint(1, p2);
  101:             initial_square.setPoint(2, p3);
  102:             initial_square.setPoint(3, p4);
  103:             initial_square.setFillColor(sf::Color::Yellow);
  104:
  105:             square_vector.push_back(initial_square);
  106:         }
  107: }
  108:
  109:
```

```
110: original::~original()
111: {
112:
113: }
114:
115:
116: void original::draw(sf::RenderTarget &target, sf::RenderStates states) const
117: {
118:     for(int i = 0; i < square_vector.size(); i++)
119:     {
120:         target.draw(square_vector.at(i), states);
121:     }
122: }
123:
124:
125:
126:
```