

Ps5 Source Code: Makefile

```
1 CC= g++
2 CFLAGS= -g -O3 -Wall -Werror -std=c++0x
3 DEPS= -lsfml-system
4
5 all:    ED
6
7 ED: main.o ED.o
8     $(CC) main.o ED.o -o ED $(DEPS)
9
10 main.o: main.cpp ED.hpp
11     $(CC) -c main.cpp ED.hpp $(CFLAGS)
12
13 ED.o:    ED.cpp ED.hpp
14     $(CC) -c ED.cpp ED.hpp $(CFLAGS)
15
16 clean:
17     rm *.o
18     rm *.gch
19     rm ED
```

Ps5 Source Code: PTree.cpp

```
1 // Name: John Simonson
2 // Date: 1/27/2020
3 // Assignment: ps2
4 #include<SFML/Graphics.hpp>
5 #include<iostream>
6 #include<string>
7 #include<cmath>
8 #include<vector>
9 using namespace std;
10 using sf::Color;
11 //argv[1] = L = ("Length of window")
12 //argv[2] = N = ("Depth of tree")
13 //Scale next = size * 1/sqrt2
14
15 void draw(int newDepth, sf::RectangleShape root, sf::RenderWindow &window);
16
17 int length;
18 int width;
19 int size;
20 int depth;
```

```

21 int main(int argc, char* argv[]){
22     string temp = argv[1];
23     string tempDepth = argv[2];
24     depth = stoi(tempDepth);
25     size = stoi(temp);
26     length = size * 6;
27     width = size * 4;
28
29     sf::RenderWindow window(sf::VideoMode(length, width), "PS2");
30     sf::RectangleShape root(sf::Vector2f(size, size));
31     root.setPosition(length/2 - size/2, width-size);
32     root.setFillColor(Color((255/depth),255-(depth*10), + (depth*20)));
33     window.draw(root);
34     draw((depth - 1), root, window);
35
36
37     while (window.isOpen())
38     {
39         sf::Event event;
40         while (window.pollEvent(event))
41         {
42             if (event.type == sf::Event::Closed)
43                 window.close();
44
45
46
47         }
48     }
49     return 0;
50 }
51
52 void draw(int newDepth, sf::RectangleShape root, sf::RenderWindow &window){
53     sf::Event event;
54     while (window.pollEvent(event))
55     {
56         if (event.type == sf::Event::Closed)
57             window.close();
58
59     }
60
61     if(newDepth <= 0){
62         return;
63     }
64     sf::RectangleShape left, right;
65     left.setOrigin(0, root.getSize().y);
66     right.setOrigin(root.getSize().x, root.getSize().y);

```

```
67     left.setSize(root.getSize());
68     right.setSize(root.getSize());
69     left.setScale(root.getScale().x*(1/sqrt(2)),root.getScale().y*(1/sqrt(2)));
70     right.setScale(root.getScale().x*(1/sqrt(2)),root.getScale().y*(1/sqrt(2)));
71     left.setPosition(root.getTransform().transformPoint(root.getPoint(0)));
72     left.setRotation(root.getRotation() - 45);
73     right.setPosition(root.getTransform().transformPoint(root.getPoint(1)));
74     right.setRotation(root.getRotation() + 45);
75
76     left.setFillColor(Color((255/newDepth),255-(newDepth*10), (newDepth*20)));
77     right.setFillColor(Color((255/newDepth),255-(newDepth*10), (newDepth*20)));
78     window.draw(right);
79     window.display();
80     draw((newDepth - 1), right, window);
81     window.draw(left);
82     window.display();
83     draw((newDepth - 1), left, window);
84     return;
85 }
```
