```
    1: // Copyright 2023 Thomas O'Connor
    2: #ifndef PTREE_HPP
    3: #define PTREE_HPP
    4:
    5: #include <cmath>
    6: #include <iostream>
    7: #include <utility>
    8: #include <SFML/Graphics.hpp>
    9: #include <SFML/System.hpp>
   10: #include <SFML/Window.hpp>
   11:
   12: using std::cout;
   13: using std::endl;
   14: using std::pair;
   15: using sf::RectangleShape;
   16: using sf::Vector2f;
   17:
   18: class PTree : public sf::Drawable, sf::Transformable {
   19:  public:
   20:     // Constructors
   21:     PTree() : _L(0), _N(0) {}
   22:     PTree(double L, int N);
   23:
   24:     // Getters
   25:     int getDepthN() const { return _N; }
   26:     double getLengthL() const { return _L; }
   27:
   28:  private:
   29:     // Draw game in SFML
   30:     void draw(sf::RenderTarget& target, sf::RenderStates states) const ov
erride;
   31:     // Recursive draw function
   32:     void pTree(sf::RenderTarget& target,
   33:       pair<Vector2f, Vector2f> newPoints, int depthN,
   34:       float angleR) const;
   35:  private:
   36:     double _L;
   37:     int _N;
   38: };
   39:
   40: // Debugger functions
   41: void drawIndicators(sf::RenderTarget& target, pair<Vector2f, Vector2f> ne
wPoints);
   42: void drawIndicator(sf::RenderTarget& target, Vector2f point);
   43:
   44: #endif
```