Automatic Tree Model Generation algorithm for Game Development

CPSC 448 Directed Studies Report

Zihan Zhu

April 27, 2023

1 Introduction

The complexity of scenes in modern video games has led to a growing need for efficient and automatic object generation. Among the various objects required, trees are a common element in game environments. However, manually creating branches to fill the canopy outline of a tree can be a tedious and labor-intensive task. In recognition of this, a game company has identified the need for an automatic tree model generation algorithm that can efficiently and visually reasonably fill in the branches of a tree outline.

The goal of this research project is to create an automatic tree model generation algorithm for game development. This paper presents the design and implementation of the algorithm, which leverages rule-based procedural modeling techniques, such as L-systems and simulated annealing to generate realistic tree models.

The ultimate deliverable of this project is a 2D automatic tree model generation algorithm that takes a canopy contour as input. Future work may involve transitioning from a 2D to a 3D interface, incorporating more advanced algorithms for branch generation, and improving the visual realism of generated tree models.

2 Algorithm

2.1 Tree Abstraction

This 2D tree branch generation algorithm utilizes a rectangle with a fixed width-height ratio to represent a tree branch. This rectangle is characterized by a root, located at the midpoint of the base line, along with a direction, width, and height. Each parent branch can grow several child branches from its top. During the growth process, if a branch encounters a collision with other branches or the canopy outline, it will undergo resizing until either no collision is present or the maximum number of resizing attempts is reached.

2.2 L-System Rule

L-systems have been particularly effective in modeling plant growth due to the presence of self-similar structures in plants (Prusinkiewicz et al., 1996). In many L-system rules for plant growth, a production rule maps a current part (X) to a more complex structure (a combination of multiple X elements with drawing forward, turning left by a certain angle, and turning right by a certain angle). During each iteration, all instances of X are replaced by the more complex structure, resulting in an increasingly intricate branching pattern that mimics natural plant growth.

An L-system rule from Ijiri et al. (2006)'s work is adapted to model the growth of branches. According to the algorithm, a parent branch can have a maximum of three child branches (left, middle, and right) growing from its top. The direction of each child branch is randomly generated within a restricted range. Additionally, each type of child branch exhibits a predefined ratio in relation to its parent branch.

2.3 Rating Function

A rating function is introduced to assess the visual appeal of a generated tree, taking into account its aesthetic qualities and overall appearance. The rating function is composed of two parts. The first part assesses the degree of branch coverage within the tree canopy, while the second part focuses on the ratio between parent and child branches.

For the first part, the ratio between the area not filled by the branches and the total area of the tree canopy is calculated. The Monte Carlo algorithm, a computational method that generates numerous random samples within a predefined space, then computes the desired outcome based on the proportion of samples meeting specific criteria, is utilized to approximate this irregular area. A crucial step of implementing this algorithm is determining whether a generated point lies within the shape or not. To determine this, we first established if the point resides on the shape's edge. If so, the point is considered to be inside the shape. For the remaining points, a ray-casting algorithm is employed. A ray extending in the positive X direction is cast from each point. If the ray intersects the shape an odd number of times, the point is deemed to be outside the shape; otherwise, it is considered to be inside (Sutherland et al., 1974).

In the second part, as mentioned in Section 3.2, the L-system defines the ratios of the left, middle, and right child branches with respect to their parent branches. Additionally, as described in Section 3.1, the generated branch will be resized if a collision occurs. Consequently, the average ratio between the reduction in size and the expected size is calculated to evaluate this aspect.

2.4 Simulated Annealing

A fundamental simulated annealing algorithm is employed in this study (Pincus, 1970). The rating function discussed in Section 3.3 is used to evaluate each tree. In each iteration, only one additional branch is generated based on the current tree. If the newly generated tree has a better rating than the current tree, it is accepted. However, if the new tree has a worse rating, it is still accepted 50 percent of the time, allowing for exploration of the solution space and avoiding local optima.

3 Results

A predefined contour is imported for this study, with the size and position of the main branch at the very bottom being predetermined. The tree growth algorithm adheres to the rules defined in Sections 3.1 and 3.2, utilizing the simulated annealing algorithm in 3.4 to guide the growth process (Figure 1, Figure 2).

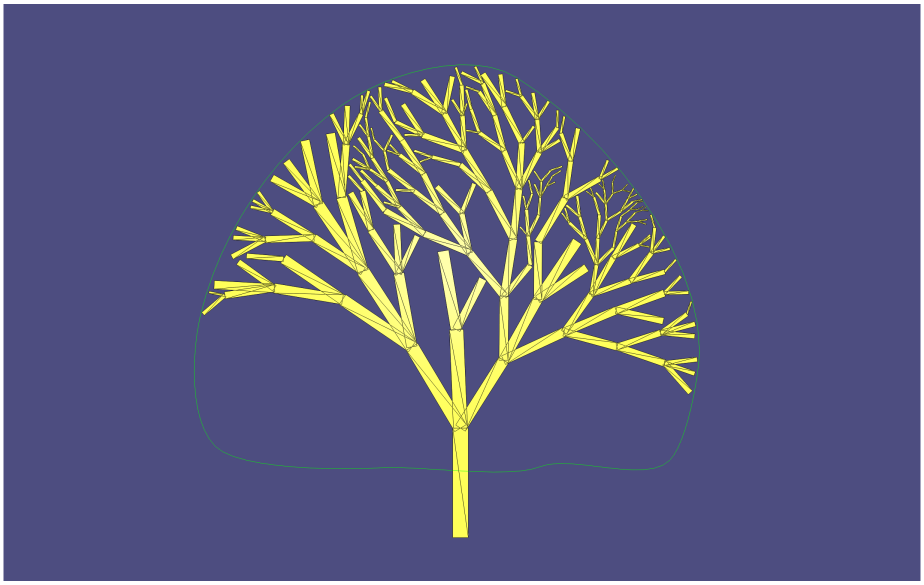

Figure 1: Generated Tree 1

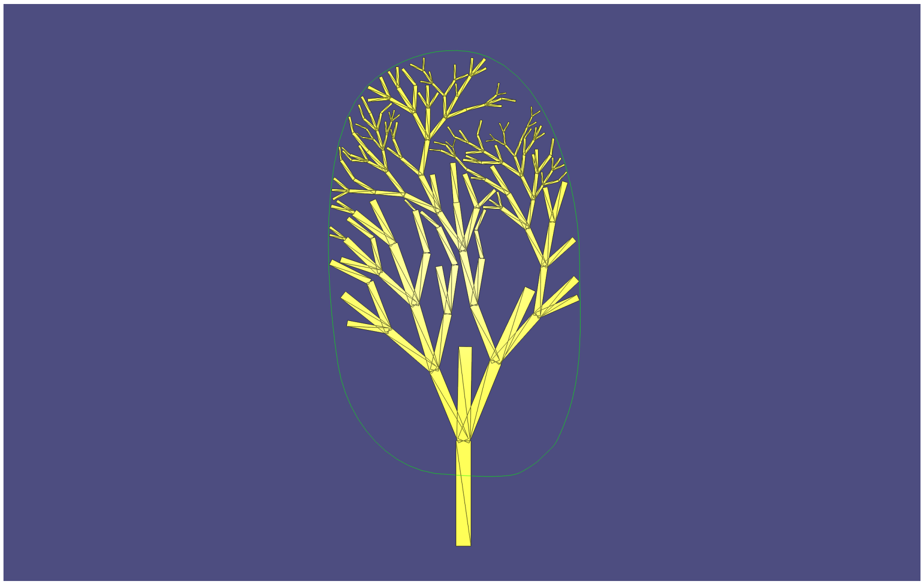

Figure 2: Generated Tree 2

4 Discussion

There are three recommendations for future work:

1. Transition the algorithm from 2D to 3D.

2. Currently, the starting branch's size and position, as well as the L-system rule, are manually defined based on the input contour. Future work could involve analyzing the input contour to automatically generate the starting branch and corresponding rule.

3. In the area calculation process, a Monte Carlo algorithm is utilized. To ensure a certain level of precision, a large number of points must be generated, which can be time-consuming. Additionally, collision checking can also take a significant amount of time. Future work could focus on accelerating this algorithm to improve its efficiency.

References

Ijiri, T., Owada, S., and Igarashi, T. (2006). The sketch l-system: Global control of tree modeling using free-form strokes. *Smart Graphics*, page 138–146.

Pincus, M. (1970). Letter to the editor—a monte carlo method for the approximate solution of certain types of constrained optimization problems. *Operations Research, 18(6)*, page 1225–1228.

Prusinkiewicz, P., Hammel, M., J., H., and Měch, R. (1996). L-systems: from the theory to visual models of plants. *M. T. Michalewicz (Ed.): Plants to Ecosystems. Advances in Computational Life Sciences, CSIRO, Collingwood, Australia*, pages 1–27.

Sutherland, I. E., Sproull, R. F., and Schumacker, R. A. (1974). A characterization of ten hidden-surface algorithms. *ACM Computing Surveys, 6(1)*, page 1–55.