Technical Report for Houdini Garden Builder

Jessica Dong s5803453@bournemouth.ac.uk
Msc CAVE, NCCA
CGI Techniques 25/26
Jian Chang, Phil Spicer
Jan 23, 2026

Introduction

The Houdini Garden Builder is a Houdini Digital Asset (HDA) designed to help artists quickly generate garden assets with abundant customization and predesigned aesthetics. This tool can create various garden shapes, generate roads on painted areas, and scatter customized primitive assets such as flowers and trees. Specifically, it includes a fence builder, road builder, and grass assets. In the process of developing this tool, many technical problems were encountered, requiring research into solutions. This report demonstrates how these challenges were approached by diving deep into Houdini nodes and their underlying methods. Additionally, the report mentions several problems that remained unsolved after various attempts.

Design of the Garden

For this assignment, I approached the design from two perspectives. First, I collected garden aesthetic images to understand the common structures and utility of a garden. Second, I considered the purpose of the garden builder and its essential functions, particularly for procedural workflows such as world-building in games. Technical artist Cody Spahr shared a world designer asset made for games, which I found very insightful (SideFX, n.d.-a). I recognized that the key elements of a world builder are bound shapes, different road levels, road connections, and biological assets. The world builder operates by painting attributes onto a ground plane, a logic similar to landscape tools in game engines. I believe this garden builder tool should easily create a variety of looks for an open-world game, where assets can adjust themselves based on relative relations to save artists time on manual adjustments.

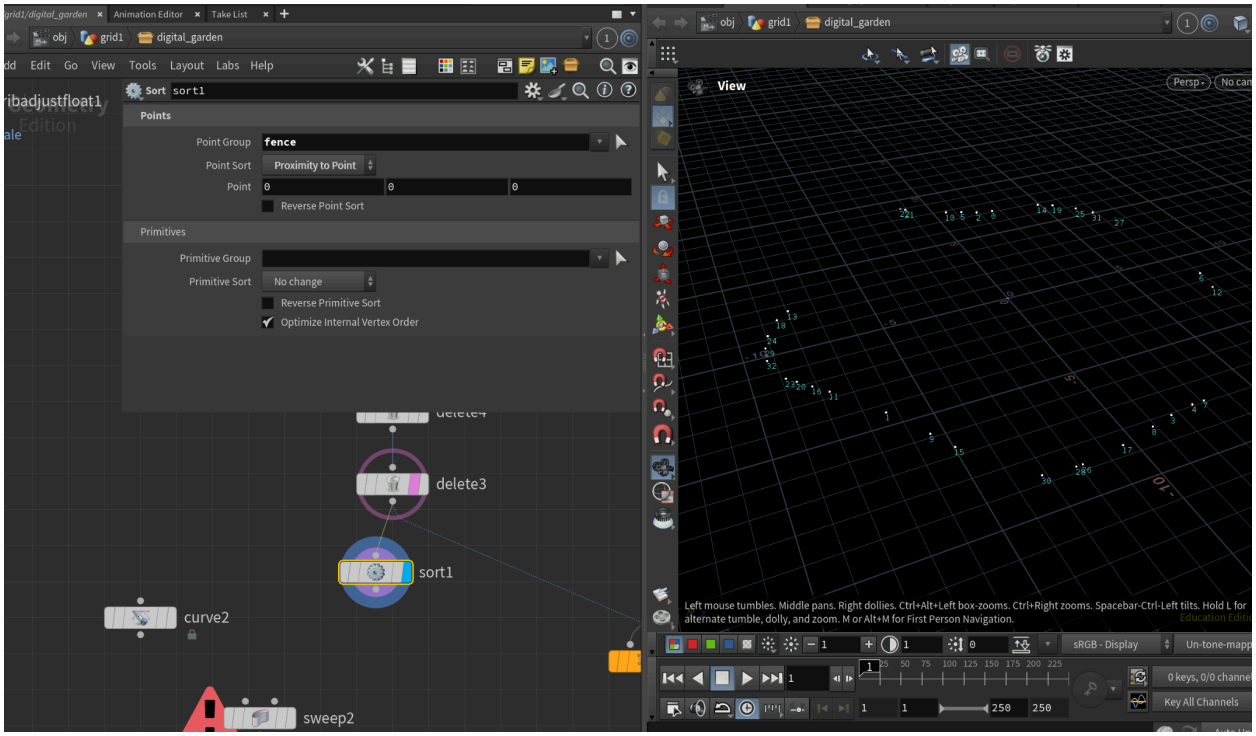
Figure 1 *Final Look of My Garden Builder*

The Application of Arctangent

Arctangent can be used for calculating angles in radians or polar coordinates. The four-quadrant inverse tangent (atan2) is widely used in computer graphics because it returns values from -PI to and is numerically stable ("Inverse Trigonometric Functions," 2026). It is frequently mentioned in robotic dynamics research. In *Motion Planning in a Robot Soccer System*, atan2 is used for robot posing (Dierssen, 2003). Similarly, Yan et al. (2020) developed a method for multiple robots to encircle a target, using atan2 to calculate agent positions relative to the target and compare them with neighbors to average the gaps in the circle.

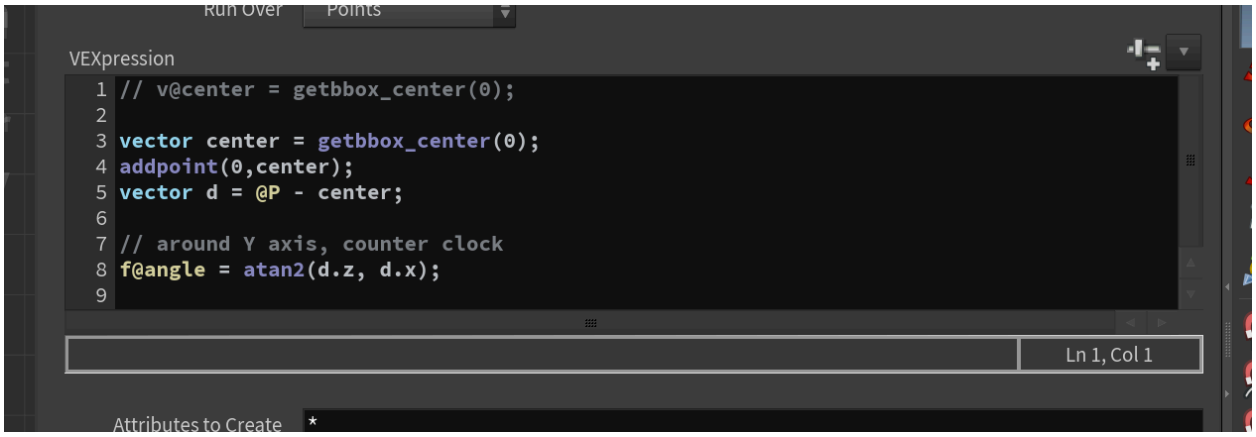
When building the garden fences, I wanted them to follow the land edge but break where a road was present. A significant problem occurred because I could not connect the edge loop in order; the point numbers had large gaps due to the complexity of the geometry.

Figure 2 *Edge Point Number Unsorted*



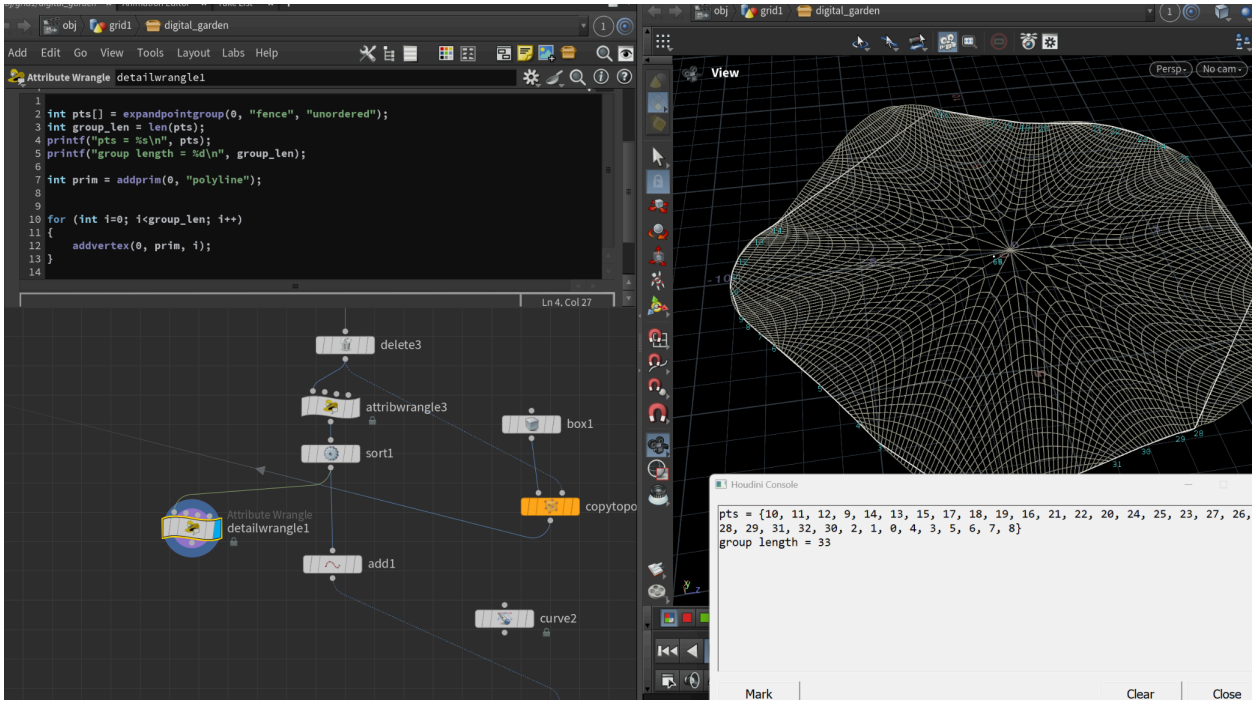
To use the Sweep node to generate the fence body, an ordered point list is required to generate a curve around the landscape. The solution involved calculating a center point and using atan2 to calculate radians, resulting in an ordered list.

Figure 3 *VEXpression for Calculating Radians*



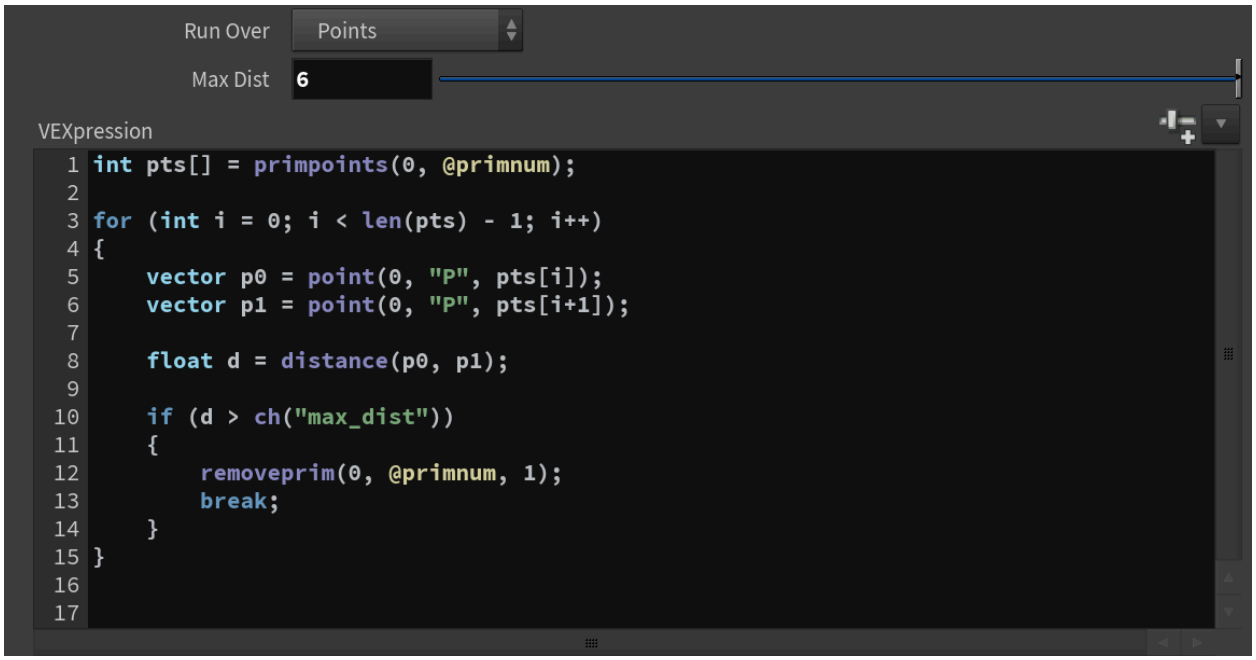
After creating a point attribute for the radians, I added a Sort node to organize the points by attribute. However, I found that even when sorted, the points would not connect automatically using the Connect Adjacent Pieces node (or Connect Line), as it still referenced the original point order. I resolved this by adding another Attribute Wrangle to create a sort list and looping through that list to connect the lines.

Figure 4 *Result of Sorted List and Primline*



Finally, I calculated the distance between each point to decide where to break the line.

Figure 5 *VEXpression for Breakline*



The L-System

An L-System is "a parallel rewriting system and a type of formal grammar" containing rules written with alphabet symbols ("L-system," 2026). L-Systems are often used in plant growth simulation and fractal pattern generation. They can also generate structures and cities for

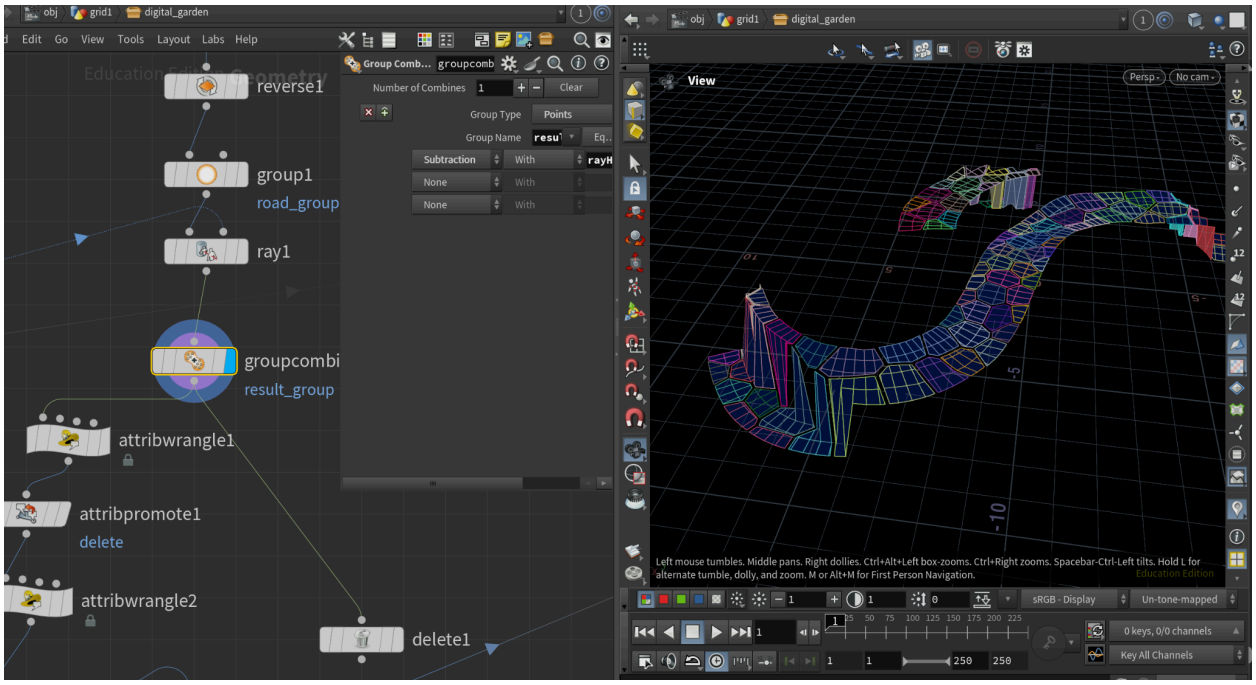
games to increase realism. In the game *Elite Dangerous*, L-Systems were used during the generation of alien civilization scenery (Montserrat, 2025).

In this project, the tree generator uses a set of tools built by SideFX that includes functions for generating branches and leaves. The generation of tree branches is often based on L-Systems, and Houdini provides an L-System node to "simulate complex organic structures such as trees, lightning, snowflakes, flowers, and other branching phenomena" (SideFX, n.d.-b). I also planned to use L-Systems to generate flower plants and explore recursive patterns for designs like a garden maze.

Using Detail Attributes to Transfer Values

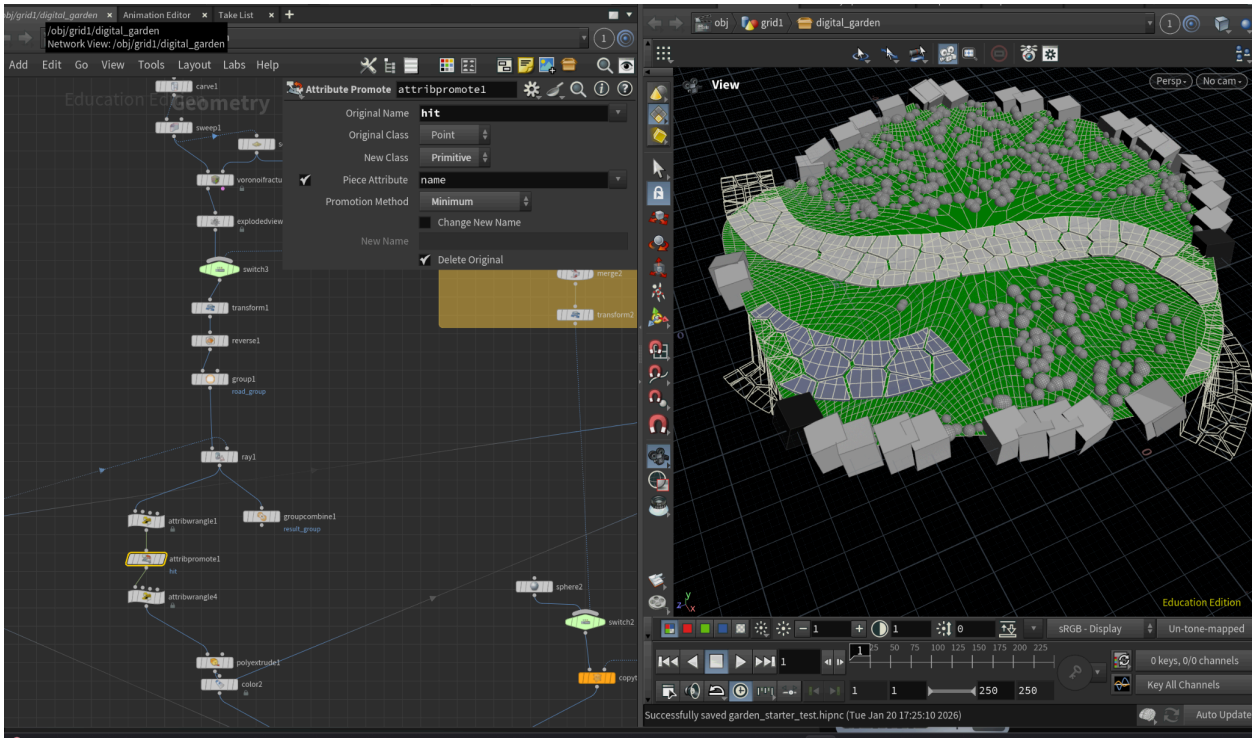
In Houdini, there are four levels of attributes: vertex, point, primitive, and detail (SideFX, 2026). A problem encountered in this project involved the deletion of geometry pieces. To adjust the road to the landscape, a Ray node was used to detect hitting areas and return a hit point group. As shown in Figure 6, some geometry pieces fell outside the boundary. Simply deleting by points resulted in sharp primitive edges and extra points, which appeared unnatural.

Figure 6 *Raycast Road to Ground*



The aim was to transfer deletion information from the "hit-miss" points to the entire geometry piece. The solution used an Attribute Promote node to promote the point attribute to primitives and toggled "Piece Attributes" so the attribute would be promoted for all primitives sharing the same "name" attribute. By setting the promotion method to "Minimum," the system only reads the minimum value when a piece has multiple points. As seen in Figure 7, the road pieces are clearly separated.

Figure 7 *Attribute Promote*



Critical Evaluation and Future Work

I successfully realized several intended functions, such as the self-adjusting fences and roads, but I did not have the opportunity to create more assets to make the garden fully functional. I considered generating procedural patterns to map bushes or using procedural texturing to make the result look more complete.

Although I separately generated assets and managed attributes for each part, I still do not fully understand when and how to optimize for performance. I need more practical experience to understand performance saving. When designing the interface parameters, it was difficult to create intuitive names and decide which parameters to expose. There are many parameters to manipulate, but determining the most effective ones is a challenge. I also explored L-Systems for building procedural plants and sub-roads; I find L-Systems very interesting and intend to implement them further in the future.

The current tool has not been user-tested for potential bugs, and it needs to be assessed for larger-scale generation to test the performance. Some functions may have better solutions; for example, drawing roads using splines is not very easy in a 3D context. There could also be some interactive guides built into the tool.

References

Dierssen, W. D. J. (2003). *Motion planning in a robot soccer system* [Master's thesis, University of Twente].

Inverse trigonometric functions. (2026, January 10). In *Wikipedia*. https://en.wikipedia.org/wiki/Inverse_trigonometric_functions

L-system. (2026, January 15). In *Wikipedia*. https://en.wikipedia.org/wiki/L-system

Montserrat, R. (2025, March 4). *Procedural content generation for video games: A friendly approach*. Level Up Game Dev Hub. https://www.levelup-gamedevhub.com/en/news/procedural-content-generation-for-video-games-a-friendly-approach/

SideFX. (n.d.-a). *World designer for MOBA*. https://www.sidefx.com/contentlibrary/world-designer-for-moba/

SideFX. (n.d.-b). *L-System SOP*. https://www.sidefx.com/docs/houdini/nodes/sop/lsystem.html

SideFX. (n.d.-c). *Tree generator*. https://www.sidefx.com/tutorials/tree-generator/

SideFX. (2026). *Geometry attributes*. https://www.sidefx.com/docs/houdini/model/attributes.html

SideFX. (2026). *VEX functions: atan2*. https://www.sidefx.com/docs/houdini/expressions/atan2.html

Yan, P., Fan, Y., Liu, R., & Wang, M. (2020). Distributed optimal deployment on a circle for cooperative encirclement of autonomous mobile multi-agents. *IEEE Access*, *8*, 58337-58344. https://doi.org/10.1109/ACCESS.2020.2982581