

**How Can L-Systems Be Used for Games Development?**

Calum Lindsay – 21010093

2025

Submitted in partial fulfilment of the requirements for the BSc (Hons) Computing (Software Development) degree.

Supervisor: Magnus Tullock

College: UHI Shetland

Department: Computer Science

Submitted: 31/07/2025

**Declaration**

I certify that unless otherwise noted in the text, this is all my own work.

Signed: [Your signature here]

Full name: Calum Lindsay

Student number: 21010093

Date: 31/07/2025

**Abstract**

Creating games requires many assets to be created, and mechanics to be designed which costs both time and money. A procedural generation technique called Lindenmayer Systems has seen some use in games development although the majority of this is in the generation of plants. This dissertation highlights other use cases seen in recent literature and develops two use cases of its own. The first is a terrain generation algorithm which works by having a turtle move nodes on a flat plane up or down as it traverses the output of the L-System producing comparable results to other simple generation techniques. The second is a 3D maze generation algorithm that regenerates and renders the maze at a configurable fixed time step which could be used to create an extremely challenging maze that changes shape as you solve it. The algorithm allows multiple L-Systems to be stacked so that the generation can be split into multiple logical steps. Overall, the dissertation accomplishes its goal of showing that L-Systems could be used to generate significant portions of games in ways that are not commonly seen in the industry.

**Acknowledgements**

Thank you to Magnus Tullock, despite my inability to be present and my lack of communication this year you have been prompt, available, and insightful whenever I have reached out for you which I greatly appreciate.

Thank you to Euan Robertson, you have been a brilliant support throughout the last five years and show a tremendous willingness to help and support every student you can. Additionally, I believe your voracious passion to learn, and share deserves significant praise.

Thank you to Jason Spooner, your small act of kindness many years ago helped direct my energies in the right direction allowing me to develop my abilities earlier than I otherwise would have. Today I find your energy, knowledge, and sense of humour to be exquisite company.

Thank you to my oldest friend, Fergus Tonner, …!!!...!!!...!!!

Thank you to my grandmother, …!!!...!!!...!!!

Thank you to my grandfather for beginning my journey all those years ago.

Thank you to Dannii, Chelsea, and Ray for putting up with me and being my reason.

**Table of Contents**

[1 Introduction 9](#_Toc31925259)

[1.1 Background 9](#_Toc31925260)

[1.2 Outline of Dissertation by chapters 9](#_Toc31925261)

[2 Literature review 10](#_Toc31925262)

[1.1 Section heading 11](#_Toc31925263)

[2.1.1 Subsection heading! 11](#_Toc31925264)

[2.1.2 Another subsection 11](#_Toc31925265)

[2.2 Another section heading 11](#_Toc31925266)

[3 Method 12](#_Toc31925267)

[4 Results 13](#_Toc31925268)

[5 Analysis 14](#_Toc31925269)

[6 Conclusions & Future Work 15](#_Toc31925270)

[6.1 Conclusions 15](#_Toc31925271)

[6.2 Summary of Aims/Objectives and or Research Questions 15](#_Toc31925272)

[6.3 Future Work 15](#_Toc31925273)

[7 Critical Reflection & PDP 16](#_Toc31925274)

[8 Bibliography 17](#_Toc31925275)

[Appendices 18](#_Toc31925276)

**Table of Figures**

[Figure. 1. Figure caption below the figure. Inverted pyramid model for the Literature Review 10](#_Toc31925253)

**Table of Tables**

[Table. 1 Table caption goes here 11](#_Toc31925238)

[Table 2.Table caption above. Not totally necessary but you could use it if you wish. It’s here as an example of how to caption a table. 11](#_Toc31925239)

# Introduction

Creating computer games is an expensive endeavour and in fact over time “the most popular commercial games get larger, prettier, more atmospheric, and more detailed” (Hendrikx, Meijer, Veldon, & Iosup, 2013, p. 2) thus, requiring increasingly higher quality content which is a challenge when “manual content production is already expensive and unscalable” (Hendrikx, Meijer, Veldon, & Iosup, 2013, p. 2).

Procedural generation is fast, cheap and can produce many variations of an asset with similar characteristics which allows the fast creation of large amounts of content with smaller development teams than if the content was manually generated.

Lindenmayer Systems are a specific type of procedural generation, mostly being used “for botanic modelling and generation” (Fornander, 2013, p. 8) with some famous examples of games that use them through a tool called SpeedTree being: Battlefield 3, The Witcher 2, Gears of War 3 (Fornander, 2013, p. 8), Elden Ring and Ghostwire: Tokyo (SpeedTree - Wikipedia, 2024), however, their usage has been fairly limited in other areas such as game mechanics, terrain generation and text generation.

This dissertation builds a picture of the literature both recent and historic and explores the problems and solutions presented within. An emphasis is put on use cases that are not botanical in nature as this area has been well researched. Hypothesises for both further development of discovered solutions as well as entirely original solutions are developed. These Hypothesises are tested by attempting to develop them in to prototype algorithms. The final selection of successful prototypes are compiled into an application for demonstration purposes, tested, and compared to existing solutions.

## Background

### Procedural Generation

Procedural generation in games development is a technique which uses an algorithm to generate content. This could be, for example, using a combination of random noise and mathematics to generate the height of the ground at each point in a game world rather than having an artist do this manually one point at a time. Procedural generation is extensively used due to its wide array of benefits including:

* Reducing development time as designers need to spend less time creating assets which can be created either completely or partially by an algorithm.
* Allowing for almost limitless variations to be generated without consuming any more physical space.
* They can create unexpected and unique gameplay experiences, commonly called “emergent gameplay” when systems interact in unanticipated ways.

### Lindenmayer Systems

Lindenmayer Systems are parallel self-rewriting string generation algorithms at heart and were first proposed as a way of modelling the development of plant like structures by Aristid Lindenmayer in 1968 with later work expanding on this and importantly introducing the concept of a turtle renderer to produce rendered images, culminating in a fascinating book: The Algorithmic Beauty of Plants (Prusinkiewicz & Lindenmayer, 1990).

Lindenmayer Systems consist of two main components:

* An axiom – a string used as the initial state for the generator.
* A set of re-write rules – each rule consists of a symbol to find and a set of symbols to replace it with.

The generator is initialized with the axiom and then on each iteration of the algorithm the rewrite rules are applied.

A white background with black text

AI-generated content may be incorrect.

### Rendering of L-Systems

Rendering is achieved by using a turtle renderer which is an algorithm where a virtual entity named a turtle reads the output of an L-System as a set of instructions to traverse and interact with a virtual environment. The mapping of symbols to actions performed by the turtle is implementation specific although a common configuration for a turtle that can draw edges in a 2D environment and an example output is shown:

A computer screen shot of a tree

AI-generated content may be incorrect.

### Game Assets

The term game asset typically refers to a data file or discrete element that can be used on its own or composed with other assets in order to create logical objects within a game world that can be sensed by the player. This includes but is not limited to:

* Environmental Models – Terrain, boulders, lakes, rivers and plant life.
* Architectural Models – Buildings, roads and bridges.
* Images – Textures, user interface components and maps.
* Text – Dialogues, descriptions and objectives.
* Audio – Voice acting, music and sound effects.

### Game Mechanics

Game mechanics are the behaviours of objects and the rules or systems within the virtual world rather than the objects themselves. Examples of these are:

* Day/ Night Cycles
* Quest Systems
* Patrolling Behaviours
* Skill Systems
* Level Scaling

## Purpose

The purpose of this dissertation is to develop new solutions using L-Systems to existing problems as well as further develop existing solutions that are uncovered through research.

### Hypothesis

That there exists solutions using L-Systems to problems that are traditionally solved using other methods, that there are further developments that could be made to existing solutions using L-Systems, and that exploration of these solutions would prove beneficial for games developers.

### Aims

* Creation and implementation of original solutions in domains with little to no prior work.
* Expansion of existing solutions.
* Combining of elements from multiple solutions to produce new solutions.

### Objectives

* Identify existing solutions and how developed they are.
* Compare developed solutions to existing solutions.
* An application for demonstrating created solutions.
* An interface for tweaking the parameters of the created solutions.
* Reasonable performance of developed solutions.
* Modularity of final solution to allow composition of complex generation systems.

## Structure

**Chapter 2** will present a comprehensive review of the currently known literature.

**Chapter 3** will present the methodology used to create and test the project.

**Chapter 4** will present the results produced during the project.

**Chapter 5** will critically evaluate the project, and the results obtained.

**Chapter 6** will present conclusions drawn and future work.

**Chapter 7** will discuss challenges faced, lessons learned and future goals.

# Literature review

This is your lit review. Introduce and outline what is being discussed here. Start with an inverted pyramid model

**Referencing examples. Please use the different types of layout for in text citations as it makes the finished dissertation look more professional. Use RefWorks rather than Word for referencing.**

**Quoting directly:**

Aitken (2006: 48) argues that land fill sites are “not cost efficient”….

**Paraphrasing:**

Aitken (2006: 48) argues that land fill sites are expensive and inefficient

**Summarising an entire book or article:**

A recent study reveals new information about child poverty in Scotland (Weir 2007).

**Summarising a point made on two consecutive pages of a book or article:**

The book provides examples of how the eating habits of parents directly influence children (Taylor 2006: 19-20).

A close up of a map

Description automatically generated

Figure. 1. Figure caption below the figure. Inverted pyramid model for the Literature Review

If you use a Table then this caption goes above… To create a caption you use menu Insert> Caption then choose the type. If you do this for each table and figure then you can create a Table of Tables and a Table of Figures in the front matter of this document.

Table. 1 Table caption goes here

|  |  |  |
| --- | --- | --- |
| **Subject** | **Author** | **Year** |
|  |  |  |
|  |  |  |

## The Problem

What is the problem that we are trying to solve? Basically time and cost of artists and game designers.

You may choose to divide this into sections, perhaps looking ahead to the method chapter.

Table 2.Table caption above. Not totally necessary but you could use it if you wish. It’s here as an example of how to caption a table.

|  |  |
| --- | --- |
| **Type of Literature** | **Authors** |
| Machine Learning | Bleh et al. 2019 |
| Deep Learning | Mushroom et al. 2014 |

### Subsection heading!

### Another subsection

#### Another subsection

Only go down to four levels of sub-sections. If you need another level just use:

**Another section formatting**

More text.

## Previous Research

### L-Systems

### Botanical Modelling

### Cities

### River Deltas

Extensive literature exists for solutions that use L-Systems to produce plants and trees. The Algorithmic Beauty of Plants (Prusinkiewicz & Lindenmayer, 1990) introduces the concept of L-Systems as a parallel string rewriting algorithm, presenting many examples of how they can be used with the combination of a turtle renderer to produce beautiful visuals of plants and trees. Further it introduces multiple types of L-System including parametric L-Systems where parameters or variables can be associated with the symbols within the string being operated on and thus arithmetic operations can be performed upon these values in rewriting rules, and stochastic L-Systems where a symbol is added to the alphabet to represent a random value which can be used to add variation to generated plants. Another paper uses genetic algorithms alongside L-Systems to produce plants for use in games. The initial parameters for each L-System are generated and then the L-Systems are allowed to develop with a user defined fitness function being used to determine which plants will pass their genome on to the next generation. This continues until a user-defined number of generations is simulated (Fitch, Parslow, & Lundqvist , 2018). Other papers have implemented skeletal-like animation of 2D L-Systems which can produce animations to mimic gravity, wind and other physically-driven effects (Hampshire, 2009), and the semantic death of plants using L-Systems by assigning a parameter to each symbol that tracks age and defining a maximum age which, when symbols reach they are replaced with symbols that represent a dead part of the plant with its own parameter that causes the part to slowly wilt and die (Castellanos, Ramos, & Ramos, 2014).

Plant generation has been the focus for L-System research, however, there has been some exploration into other areas:

* Using L-Systems to dynamically generate unique levels for each playthrough within a virtual reality escape room game (Yaswinski, Chelladurai, & Barot, 2024).
* Using L-Systems to model cities, generating road networks as well as buildings. The approach coins the term ‘Extended L-systems’ where the parametric L-System produces a generic template and then a second step is performed where the parameters of the L-System are externally modified by functions to bring them in accordance with specific goals, thus allowing the output of the L-System to be modified without changing its rule set (Parish & Müller, 2001).
* Using L-Systems to generate river deltas. This is a preliminary paper that presents a method using the simplest of deterministic L-Systems without stochastic operators or parameters to generate skeletons which could be further processed to produce realistic river deltas in virtual 3D environments (Valencia-Rosado & Starostenko, 2019).

Using L-Systems as a game mechanic. In this case L-Systems generate trees that would have a random number of poisonous branches releasing toxic fumes to damage a player who would have to use these branches to navigate an environment. The player was able to control the growth of the L-Systems and branches from a system could fall off and continue to grow on their own. The concept is that players must carefully plan when to grow L-Systems without full knowledge due to the randomness of the branches to navigate a virtual environment (Fornander, 2013).

## Research Gap

What doesn’t exist or is underdeveloped? I can’t say it doesn’t exist though maybe just not prevalent.

## Conclusion?

How do I use this going forwards.

# Method

This chapter should explain in detail how you carried out your project. This includes several aspects, although not all projects will need all of these. Use your judgement to decide which of these are appropriate to your project. Add anything I may have left out, if you think it’s necessary.

1. the development methodology you have chosen (waterfall, agile,…)
2. testing method
3. how you gathered requirements
4. the tools (languages, software, platforms, libraries etc.) you chose
5. research methods used, if there is a “research” aspect to your project, such as conducting a usability study of your program or web site, checking the effectiveness of educational tools or environments etc.

As well as explaining the method/tool etc., you should also provide some argument justifying your choice. In the best dissertations, this will include consideration of the alternatives.

## Development Methodology

The project does not have a clearly defined destination and the timescales of individual tasks was not predictable. Therefore the simple waterfall method where timelines and tasks are planned then plotted out ahead of time doesn’t work particularly well. Instead, an iterative and incremental development methodology will be used as these provide more flexibility and are “associated with many successful large projects and recommended by standards boards” (Larman & Basili, 2003).

A specific named methodology will not be used as these are typically designed for teams creating commercial software instead of lone developers without involved clients. The chosen methodology will use a Kanban board to track what tasks are pending, in progress and complete, and Scrum inspired sprints will be used where tasks are chosen each week based on a ranking of their estimated value. The pending column of the Kanban board will serve as this ranking and will be updated each week reconsidering the value of all the tasks in the list as well as uncompleted tasks from the previous sprint which may be moved to pending if their estimated value drops below other tasks. New tasks will be added to the pending column as they are formulated. I have used this methodology in past projects to good effect and so am confident it will be effective here.

## Design Methodology

Find concepts in papers or come up with concepts, hypothesize how these concepts could work, test hypothesis, repeat until working prototype is created or an impasse is reached.

## Testing Methodology

Since not product just for exploring use cases no usability just developer regression testing as features are created to ensure nothing is accidentally broken as new features are implemented.

## Tools

Options, why one was selected Unreal, Unity, roll your own..

### Unreal Engine

Using a game engine will significantly reduce the workload while producing better results compared to learning about and then writing all the rendering code from scratch which will allow more focus to be given to the main task of researching and developing L-System based algorithms.

Unreal engine 5 has been chosen due to its extensive use in the industry with Epic Games’ own titles such as Fortnite, Rocket League and Fall Guys as well as big hitters like Kingdom Hearts III (Kayser, 2020) and Hogwarts Legacy (Crecente, 2023).

There are other options such as Unity and Godot. In this case it would not make much difference which of these options was used due to the student having no experience with any of them, however, they do have extensive experience with C++ which is what Unreal Engine is written in and a pre-existing desire to learn Unreal Engine.

C++ will be used to create the procedural generation code as it easily integrates with Unreal Engine, and it is a high-performance low-level language well-suited to producing code that may run under performance critical conditions such as during initialization or within tight game loops.

# Results

This chapter is where you present the results of your project and explain what you achieved.

If you were collecting data for some reason then this is where you could present it. If you built something then this is where you would describe it, including a discussion of how far you got with the original plan.

If you built something and tested it (and you should have…) then this is where you would present your test results.

Actual data u (at least if it’s large) and code listings should be placed in an appendix and referred to from here. This chapter would then discuss those maybe using descriptive statistics.

## Walkthrough of Solution

## Testing Tables

## Summary

No interpreting or conclusions, mention unexplored or halted directions (Dialogue / Quests).

# Analysis

This chapter is for a critical evaluation of the results obtained and of the project in general.

If you collected data, this is where you should interpret it.

This is also the place for an evaluation of the project, of its methods and outputs. (Product **and** process.) Markers will expect a high level of reflection, analysis, thought, perception, insight… This is where you are expected to demonstrate high level academic skills. Going back over your project log book (you remembered to keep one, right?) will be helpful here.

## Fulfilment of Hypothesis

## Versus Existing Literature

## Compared to Other Techniques

## Implications

Shows that there is room for further development of L-Systems and probably untapped potential. Used as an aid to artists is fairly interesting with the way it can be split into stages and the concept of visual rules rather than text.

# Conclusions & Future Work

What it says on the tin.

You might want to talk about interesting/worthwhile/possible future directions for this project. Perhaps put that in a section called “Further work”.

## Research Question / Hypothesis

## Conclusions

Maybe list your original objective(s) and aims then add a sentence or two explaining how (if!) you met them.

## Future Work

* Dialogue, Quests, Textures
* Visual Rules – this shape replaces that shape.
* Terrain that allows deforming in 3D to create caves rather than a simple heightmap
* Maze turned into a full game where something is done about the player being inside the maze when it regenerates.
* Direct interpretation of the L-System output as a grid of values or equivalent.
* Integration of more L-System types (parametric, etc) and swap-in / out functionality for both turtles and l-system types.

# Critical Reflection & PDP

Write about how the project has helped you develop both personally and professionally. Include any ethical issues that have arisen and discuss how they have been resolved. For example, maybe before the course you were awful at planning and management and that has changed. Or maybe you have experienced and overcome adversity during the course? Please don't put any private/personal details in here but loosely refer to the issue and how you managed to overcome it to help you develop.

Provide a convincing discussion of personal and professional learning that has taken place and how this learning will influence your future direction. This should also include personal development that has taken place, with examples to back up your discussion wherever possible.

Where appropriate, clearly and convincingly discuss how ethical and professional issues have influenced decisions taken throughout the project.

## Challenges

Unreal Engine

Time – self doubt (surpass your limits)

Limited relevant research

## Skills Acquired!?

Unreal Engine is not a complete unknown. Research skills?

## Future Goals

Reaffirmed interest in game development. L-systems write and publish a paper exploring more in depth in attempt to convince more people to research and consider it an option.

## Lessons Learned (Conclusion)

Unreal

# Bibliography

All your references should be listed here, following the UHI Harvard Referencing standard. You’ve been doing this for a while now, so there is no excuse for doing it wrong. Remember to list the references in alphabetical order of the first author’s last name. **Refworks** may be a good solution here.

# Appendices

## Appendix A: Source Code Listing

## Appendix B: Exemplars

## Appendix C: Test Log

These are like chapters, but the numbering restarts: Appendix A, Appendix B,...

Each should start on a new page.

[I don’t know how to restart the numbering and get these included in the TOC, but you’re all expert IT users so it shouldn’t be a problem for you. If you don’t know how to do it, you could always Google it! I’m sure we’re not the first people to want to create a dissertation with appendices in Microsoft Word. Headings for appendices should be like this: Appendix A: Source code listing, Appendix B: Test Plan, Appendix C: Test Log, etc.]