



UMCS

UNIwersytet Marii Curie-Skłodowskiej
w Lublinie

Wydział Matematyki, Fizyki i Informatyki

Kierunek: **Informatyka**

Specjalność: –

Aleh Hutsko

Nr albumu: 296609

Symulacja wzrostu roślin generowanych przez system Lindenmayera

Simulation of the growth of plants generated by the
Lindenmayer system

Praca licencjacka

napisana w Katedrze oprogramowania systemów informatycznych
pod kierunkiem dr Krzysztof Dmitruk

LUBLIN 2022

Spis treści

Wstęp	3
1 System Lindenmayera	4
1.1 Informacje wstępne	4
1.2 Struktura L systemu	5
1.3 Interpretacja ciągu znaków	7
2 Implementacja	10
2.1 Wykorzystane narzędzia	10
2.1.1 Język i środowisko	10
2.1.2 Biblioteka Proctree	10
2.1.3 Biblioteka nlohmann Json	10
2.1.4 OpenGL	10
2.2 Funkcjonalność aplikacji	10
2.2.1 Planowanie drzew	10
2.2.2 Symulacja wzrostu	10
2.2.3 Ustawianie parametrów	10
2.2.4 Ustawianie tekstur	10
2.2.5 Zapis do pliku	10
2.3 Struktura programu	10
3 Testy i rezultaty	11
3.1 Wydajność	11
3.2 Porównanie z innymi rozwiązaniami	11
4 Podsumowanie	12
Bibliografia	13

Wstęp

Rośliny to rozległa grupa organizmów żywych, występujących na większości lądów na Ziemi, a także w środowisku wodnym. Należą do nich trawy, drzewa, kwiaty, krzewy, paprocie, mchy i wiele innych. Istnieje około 391,000 gatunków roślin, z których zdecydowana większość, około 369,000 (94%), wytwarza nasiona.[1] Rośliny można znaleźć na całym świecie, na wszystkich kontynentach. Rośliny dostarczają znaczną część tlenu na świecie i stanowią podstawę większości ekosystemów na Ziemi. Tak ważna część świata rzeczywistego prędzej czy później wymagała matematycznego opisu i dalszego zastosowania w różnych rodzajach nauki, w szczególności w informatyce. Modelowanie roślin w informatyce jest szeroko stosowane w wielu dziedzinach, takich jak gry, przemysł filmowy, agrokultura i architektura. Rośliny charakteryzują się złożoną, zwykle fraktalną strukturą, która jest trudna do modelowania. Z tego powodu opracowano różne systemy opisywania modeli roślin, aby uporządkować i uprościć pracę z modelowaniem drzew. Jednym z takich systemów jest system Lindenmaiera, który umożliwia opis struktur fraktalnych, w szczególności roślin na poziomie gramatyki formalnej.

Celem tej pracy jest analiza i zapoznanie się z systemem Lindenmayera, możliwościami jego rozbudowy i wykorzystania do generowania roślin, a konkretnie drzew. Ponadto należy opracować oprogramowanie umożliwiające tworzenie trójwymiarowych modeli drzew z możliwością modyfikacji parametrów drzew i symulacji ich wzrostu. Oprogramowanie powinno posiadać następujące funkcje:

- możliwość wyświetlania drzew w przestrzeni trójwymiarowej,
- możliwość modyfikowania drzew przy użyciu różnych parametrów,
- możliwość wyboru tekstur dla pnia drzewa i liści,
- możliwość symulacji wzrostu drzew,
- możliwość zapisywania i wczytywania drzew o określonych parametrach.

Rozdział 1

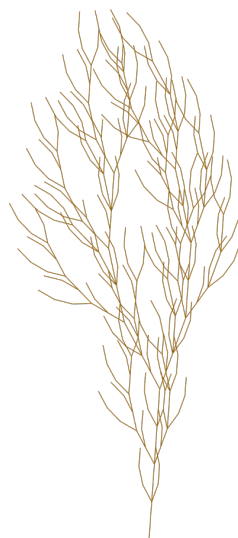
System Lindenmayera

1.1 Informacje wstępne

Systemy Lindenmayera (L-Systemy) zostały wprowadzone i rozwinięte w 1968 roku przez Aristida Lindenmayera, węgierskiego biologa teoretycznego i botanika z Uniwersytetu w Utrechcie. Lindenmayer wykorzystał L-systemy do opisu zachowania komórek roślinnych i modelowania procesów wzrostu w rozwoju roślin.

Reguły L-systemu reprezentują rekurencję. Prowadzi to do samopodobieństwa, a więc formy fraktalne można łatwo opisać za pomocą L-Systemu. Modele roślin, komórek i innych form organicznych naturalnie występujących gatunków można łatwo zdefiniować za pomocą L-systemu, ponieważ wraz ze wzrostem poziomu rekurencji forma powoli "rośnie" i staje się coraz bardziej złożona. Systemy Lindenmayera są również popularne w symulowaniu sztucznego życia.

Na rysunku 1.1 jest przykład zastosowania L-systemu dla stworzenia fraktalnej struktury, która przypomina drzewo.



Rysunek 1.1: Przykład stworzonej struktury za pomocą L-systemu

1.2 Struktura L systemu

L-systemy są obecnie powszechnie nazywane parametrycznymi L-systemami, definiowanymi jako krotka:

$$G = (V, \omega, P), \quad (1.1)$$

gdzie

- V (alfabet) – to zbiór symboli zawierający zarówno elementy, które można zastąpić (zmienne), jak i te, których nie można zastąpić ("stałe" lub "terminale")
- ω (początek, aksjomat lub inicjator) – to ciąg symboli z V , który określa stan początkowy systemu.
- P – to zbiór reguł produkcji lub produktów określających sposób zastępowania zmiennych przez kombinacje stałych i innych zmiennych. Produkcja składa się z dwóch ciągów: poprzednika i następnika. Dla każdego symbolu A , który jest członkiem zbioru V i nie występuje po lewej stronie żadnego iloczynu w P , zakłada się tożsamość iloczynu $A \rightarrow A$; symbole te nazywamy stałymi lub terminalnymi.

W standardowej wersji L-systemów reguły wnioskowania są następujące:

$$v \rightarrow \omega, \quad (1.2)$$

gdzie v jest znakiem danego alfabetu V , $\omega \in V^*$ jest łańcuchem znaki (ewentualnie puste) w tym samym alfabecie. Każdą regułę można więc interpretować jako podział komórki ($|\omega| > 1$), lub jej modyfikację ($|\omega| = 1$), lub jako jej śmierć ($|\omega| = 0$).

Na tabeli 1.1 przedstawiono przykład L-systemu.

Tabela 1.1: Przykład stworzonej struktury za pomocą L-systemu

Alfabet	Aksjomat	Reguły
$\{ \ulcorner A \urcorner, \ulcorner B \urcorner, \ulcorner F \urcorner, \ulcorner H \urcorner, \ulcorner J \urcorner, \ulcorner + \urcorner, \ulcorner - \urcorner \}$	$\ulcorner FB \urcorner$	$\ulcorner A \urcorner \rightarrow \ulcorner FBFA + HFA + FB - FA \urcorner$ $\ulcorner B \urcorner \rightarrow \ulcorner FB + FA - FB - JFBFA \urcorner$ $\ulcorner F \urcorner \rightarrow \ulcorner \urcorner$ $\ulcorner H \urcorner \rightarrow \ulcorner - \urcorner$ $\ulcorner J \urcorner \rightarrow \ulcorner + \urcorner$

Po zdefiniowaniu L-systemu, zaczyna ona ewoluować zgodnie ze swoimi zasadami. Stanem początkowym L-systemu jest jego aksjomat. Wraz z dalszym rozwojem ta linia

opisująca stan ulegnie zmianie. Rozwój L-systemu odbywa się cyklicznie. W każdym cyklu rozwoju ciąg jest oglądany od początku do końca, symbol po symbolu. Dla każdego znaku wyszukiwana jest reguła, dla której ten znak jest poprzednikiem. Jeśli taka reguła nie zostanie znaleziona, znak jest pozostawiony bez zmian. Innymi słowy, dla tych znaków $\lceil X \rceil$, dla których nie istnieje reguła jawna, obowiązuje reguła domyślna: $\lceil X \rceil \rightarrow \lceil X \rceil$. Jeśli zostanie znaleziona pasująca reguła, znak poprzednika jest zastępowany przez łańcuch następnika z tej reguły.

Dla ilustracji rozważmy następujący L-system (nazywamy go glon (łat. *Algæ*), ponieważ jego rozwój modeluje wzrost pewnego gatunku alg) w tabeli 1.2:

Tabela 1.2: Przykład stworzonej struktury za pomocą L-systemu

Aksjomat	Reguły
$\lceil A \rceil$	$\lceil A \rceil \rightarrow \lceil B \rceil$ $\lceil B \rceil \rightarrow \lceil AB \rceil$

W tabeli 1.4 przedstawiono stany tego L-systemu odpowiadające pierwszym dziesięciu cyklom rozwoju systemu.

Tabela 1.3: Przykład stworzonej struktury za pomocą L-systemu

Generacja	Stan
0	$\lceil A \rceil$
1	$\lceil B \rceil$
2	$\lceil AB \rceil$
3	$\lceil BAB \rceil$
4	$\lceil ABBAB \rceil$
5	$\lceil BABABBAB \rceil$
6	$\lceil ABBABBABABBAB \rceil$
7	$\lceil BABABBABABBABBABABBAB \rceil$
8	$\lceil ABBABBABABBABBABABBABABBABABBABABBAB \rceil$

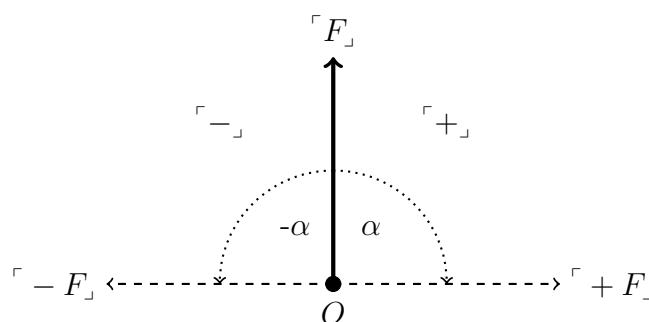
Można zauważyć, że długości ciągów kodujących stan takiego L-systemu tworzą ciąg liczb Fibonacciego, czyli ciąg liczbowy, w którym każda liczba jest równa sumie dwóch poprzednich. Ciągami Fibonacciego będą także numery znaków A i B w tych ciągach. Bardziej zaskakujący jest fakt, że ciąg ciągów ma taką samą prawidłowość jak ciąg liczb Fibonacciego: każdy ciąg jest sumą (konkatenacją) dwóch poprzednich.

1.3 Interpretacja ciągu znaków

W celu dalszej graficznej interpretacji otrzymanych ciągów należy wprowadzić pojęcie grafiki żółwia. Grafika żółwia to zasada organizacji graficznej biblioteki wyjściowej oparta na metaforze żółwia, wyimaginowanego (a w niektórych eksperymentach rzeczywistego) urządzenia przypominającego robota, które porusza się po ekranie lub papierze i obraca w zadanym kierunku, pozostawiając za sobą (lub opcjonalnie nie pozostawiając) narysowaną linię o zadanym kolorze i szerokości.

Interpretacja znaków polega na zdefiniowaniu operacji dla symboli (nie jest konieczne dla wszystkich) w alfabecie. Czynności, podobnie jak symbole, są z kolei definiowane przez autora systemu. Rysunek ref przedstawia przykład interpretacji symbolu (z kątem α równym 90 stopni) w następujący sposób:

- $\text{「}F\text{」}$ oznacza przejście do przodu i narysuj linię
- $\text{「}-\text{」}$ oznacza obrót w kierunku przeciwnym do ruchu wskazówek zegara na kąt o mierze α°
- $\text{「}+\text{」}$ oznacza obrót zgodnie z ruchem wskazówek zegara o α°



Rysunek 1.2: Przykładowa interpretacja symboli



Rysunek 1.3: Wynik L-systemu



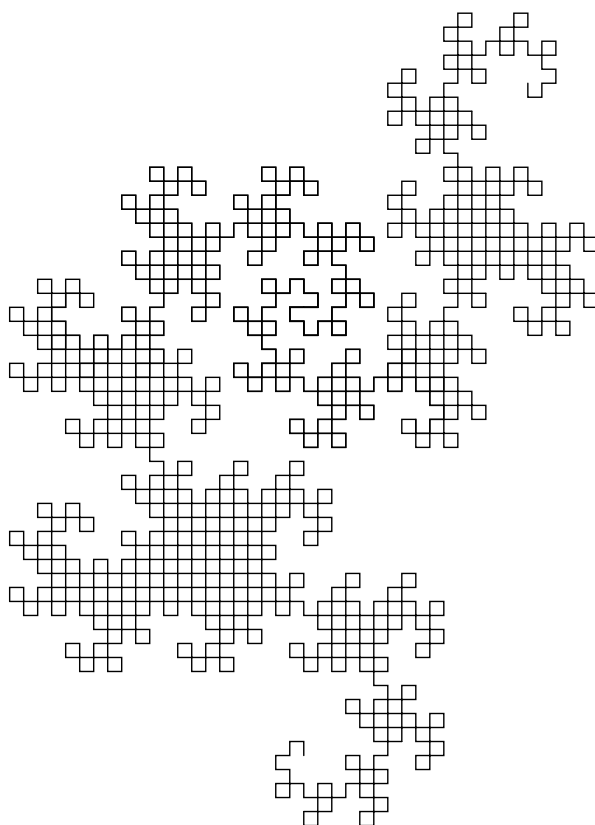
Rysunek 1.4: Wynik L-systemu

Tabela 1.4: Przykład stworzonej struktury za pomocą L-systemu

Generacja	Stan
1	┐
2	┐ └┐
3	┐ └┐ └┐
4	┐ └┐ └┐ └┐
5	┐ └┐ └┐ └┐ └┐ └┐
6	┐ └┐ └┐ └┐ └┐ └┐ └┐ └┐
7	┐ └┐ └┐ └┐ └┐ └┐ └┐ └┐ └┐ └┐



Rysunek 1.5: Wynik L-systemu



Rysunek 1.6: Wynik L-systemu

Rozdział 2

Implementacja

2.1 Wykorzystane narzędzia

2.1.1 Język i środowisko

2.1.2 Biblioteka Proctree

2.1.3 Biblioteka nlohmann Json

2.1.4 OpenGL

2.2 Funkcjonalność aplikacji

2.2.1 Planowanie drzew

2.2.2 Symulacja wzrostu

2.2.3 Ustawianie parametrów

2.2.4 Ustawianie tekstur

2.2.5 Zapis do pliku

2.3 Struktura programu

Rozdział 3

Testy i rezultaty

3.1 Wydajność

3.2 Porównanie z innymi rozwiązaniami

Rozdział 4

Podsumowanie

Bibliografia

- [1] S. Dasgupta, “How many plant species are there in the world? scientists now have an answer,” *Mongabay*, 2016. [Online; dostęp 19 czerwca 2022].