

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 491

GENERIRANJE MODELA DRVEĆA

Marina Tajić

Zagreb, lipanj 2009

Sadržaj

1. UVOD.....	1
2. LINDENMAYEROVI SUSTAVI	2
2.1. POVIJEST LINDENMAYEROVIH SUSTAVA	2
2.2. DEFINICIJA	2
2.3. INTERPRETACIJA ZNAKOVNIH NIZOVA TURTLE GRAFIKOM	5
2.3.1. <i>Turtle grafika u 3D prostoru.....</i>	<i>6</i>
3. KLASIFIKACIJA I MORFOLOGIJA DRVEĆA	10
3.1. STRUKTURE GRANANJA	11
4. PROGRAMSKA IMPLEMENTACIJA	14
4.1. KORIŠTENE TEHNOLOGIJE	14
4.2. GRAFIČKO SUČELJE	15
4.3. IMPLEMENTACIJA L-SUSTAVA	17
4.3.1. <i>Primjer metode za crtanje</i>	<i>17</i>
5. REZULTATI	21
6. ZAKLJUČAK	24
7. POPIS SLIKA.....	25
8. POPIS TABLICA.....	26
9. LITERATURA	27
10. SAŽETAK	28

1. Uvod

Od početka razvoja računalne grafike ljudi su nastojali prenijeti elemente svijeta koji ih okružuje u novi, virtualni svijet. Na početku to su bile dvodimenzionalne jednostavne reprezentacije koje su se tijekom godina razvile u današnje vrlo realistične i detaljne trodimenzionalne kopije stvarnih objekata. Usporedno s pokušajem prenošenja objekata pokušavalo se i što realnije prikazati svijet iz kojeg oni potječu. Danas, gotovo svaka aplikacija koja prikazuje realni svijet ima barem jedno drvo. Ona su manje ili više detaljna, ali su sva vjerne reprezentacije drveća koje vidimo svaki dan u šetnji, na putu na posao ...

Tema ovoga rada je upravo modeliranje drveća jednom od najčešćih metoda, L-sustavima ili Lindenmayerovim sustavima. U nastavku rada prvo će biti objašnjeni Lindenmayerovi sustavi općenito i njihova implementacija *turtle* grafikom. Zatim će se obraditi morfologija i klasifikacija drveća. Treći dio rada opisat će ostvarenu programsku implementaciju, dok su u četvrtom dijelu prezentirani i diskutirani rezultati.

2. Lindenmayerovi sustavi

2.1. Povijest Lindenmayerovih sustava

Lindenmayerove sustave ili skraćeno L-sustave uveo je i razvio 1968. godine mađarski teoretski biolog i botaničar Aristid Lindenmayer. Zamišljeni su kao matematička teorija za razvoj biljaka. U samome početku nisu bili dovoljno detaljni za kompleksno modeliranje viši oblika biljaka, već su bili namijenjeni za topologiju biljaka, tj. odnose između stanica i većih dijelova biljaka. Detaljnije proučavanje otkrilo je veliki potencijal L-sustava i oni su se počeli primjenjivati i za detaljno modeliranje kompleksnih vrsta biljaka, npr. drveća i cvijeća.

2.2. Definicija

Lindenmayerovi sustavi su formalne gramatike koje neki početni niz znakova putem skupa produkcija pretvaraju u kompleksniji niz znakova iste gramatike. Definiiraju se pomoću n-torke:

$$\mathbf{G} = \{V, S, \omega, P\},$$

gdje je:

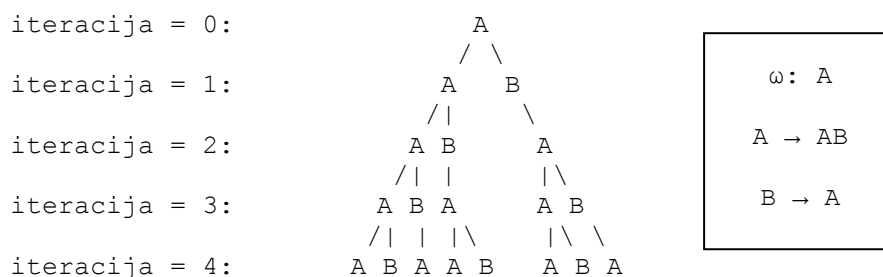
V skup nezavršnih znakova,

S skup završnih znakova,

ω niz simbola skupa V koji definiraju početno stanje sustava,

P skup produkcija ili pravila zamjene koji definiraju na koji način se nezavršni znakovi zamjenjuju nekom kombinacijom završnih i nezavršnih znakova.

Ideja L-sustava je iterativna zamjena znakova znakovnog niza tako da se u svakoj iteraciji primjeni što je moguće više produkcija. Na slici je prikazan originalni Lindenmayerov sustav za simuliranje rasta algi (Slika 1.)



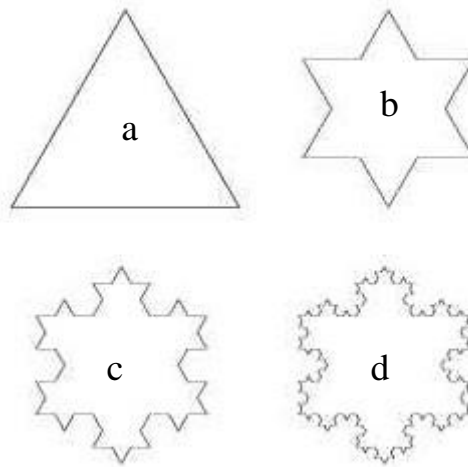
Slika 1. Princip zamjene u L-sustavu.

Upravo nastojanje da se što više zamjena napravi u jednoj iteraciji razlikuje L-sustave od jezika generiranih formalnim gramatikama i čini ih strogim podskupom generiranih jezika.

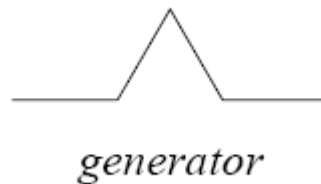
L-sustavi mogu biti kontekstno ovisni i kontekstno neovisni tj. produkcijska pravila se mogu odnositi na znak u ovisnosti o njegovim susjednim znakovima ili samo na znak, bez obzira koji znakovi su mu susjedni. Također, L-sustavi mogu biti deterministički ili stohastički. Deterministički su onda kada za jedan nezavršni znak imaju točno jednu produkciju i tada se popularno nazivaju *DOL-sustavi* (*deterministički i kontekstno neovisni L-sustavi*). Ako postoji nekoliko produkcija za isti nezavršni znak i izabiru se prema pripadnim vjerojatnostima L-sustav je stohastički. Stohastički mogu biti i takvi da u produkcijskim pravilima postoje parametri (npr. kut rotacije) koji se mijenjaju o određenom rasponu pod utjecajem slučajne veličine.

Kako se L-sustavi koriste za generiranje grafičkih objekata potrebno je da znakovi znakovnog niza predstavljaju elemente slike na grafičkom zaslonu. Tako se znakovni nizovi L-sustava najčešće sastoje od znakova koji su naredbe *turtle* grafike koja je slična onoj u programskom jeziku Logo. Tipičan primjer L-sustava je krivulja snježne pahuljice ili Kochova krivulja (Slika 2.). Inicijalni oblik je

jednakostraničan trokut (Slika 2.a) koji se zatim u daljnjim iteracijama zamjenjuje drugim objektom koji se naziva *generator* (Slika 3.).



Slika 2. Kochova krivulja.



Slika 3. Generator Kochove krivulje.

U prvoj iteraciji zamjenom svake stranice početnog trokuta generatorom dobijem oblik zvijezde (Slika 2.b). U drugoj iteraciji svaku stranicu zvijezde zamijenimo generatorom i dobijem kompleksniji oblik koji već nalikuje na pahuljicu (Slika 2.c). Pahuljica se dobije u n -toj iteraciji zamjena (Slika 2.d).

2.3. Interpretacija znakovnih nizova turtle grafikom

Turtle grafika je pojam za metodu crtanja vektorske grafike u računalnoj grafici pomoću pokazivača položaja (*turtle*) u Kartezijevom sustavu. Stanje *turtle* pokazivača se definira trojkom:

$$(x, y, \alpha)$$

gdje su:

x i **y** koordinate Kartezijevog koordinatnog sustava,

α kut tj. usmjerenje *turtle* pokazivača.

Kretanje *turtle* pokazivača može se definirati sljedećim znakovima, uz uvjet da su duljina segmenta koji se iscrtava d i inkrement kuta δ definirani prije početka crtanja:

F pomaknu se naprijed za duljinu d i pri tome iscrtaj segment

f pomakni se naprijed za duljinu d i pri tome nemoj iscrtati segment

+ okreni se u smjeru kazaljke na satu za kut δ

- okreni se suprotno od smjera kazaljke na satu za kut δ

Primjenom ovih simbola mogu se interpretirati nizovi znakova generirani L-sustavima. Tako je Kochova pahuljica interpretacija *turtle* grafikom sljedećeg L-sustava:

$$\omega: F + + F + + F$$

$$F \rightarrow F - F + + F - F$$

Za prve dvije iteracije dobiju su ovi nizovi znakova:

Nulta iteracija: $F + + F + + F$

Prva iteracija: $F - F + + F - F + + F - F + + F - F + + F - F + + F - F + + F - F$

Druga iteracija: F - F + + F - F - F - F + + F - F + + F - F + + F
 - F - F - F + + F - F + + F - F + + F - F - F - F + + F - F +
 + F - F + + F - F - F - F + + F - F + + F - F + + F - F - F -
 F + + F - F + + F - F + + F - F - F - F + + F - F

Duljina segmenta d Kochove pahuljice je proizvoljna, dok inkrement kuta δ iznosi 60° .

2.3.1. Turtle grafika u 3D prostoru

Da bi interpretaciju L-sustava *turtle* grafikom proširili na 3D prostor orijentaciju *turtle* pokazivača definiramo pomoći tri vektora $\vec{H}, \vec{L}, \vec{U}$ (Slika 4.) prema ideji iznesenoj u [1]. Vektori $\vec{H}, \vec{L}, \vec{U}$ određuju smjer *turtle* pokazivača, smjer lijevo od *turtle* pokazivača i smjer prema gore, respektivno. Definirani vektori su međusobno položeni pod pravim kutom i zadovoljavaju jednadžbu:

$$\vec{H} \times \vec{L} = \vec{U}$$

Da bi se dobio novi položaj *turtle* pokazivača potrebno je vektore pomnožiti jednom od rotacijskih matrica definiranih za vrijednost kuta α oko jednog od vektora:

$$[\vec{H}' \quad \vec{L}' \quad \vec{U}'] = R \cdot [\vec{H} \quad \vec{L} \quad \vec{U}]$$

gdje je R jedna od matrica:

$$R_U(\alpha) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_L(\alpha) = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix}$$

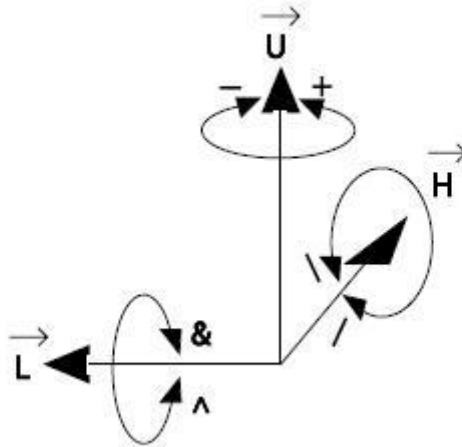
$$R_H(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}$$

Nakon definiranja pozicije *turtle* pokazivača i operacija za promjenu smjera u prostoru potrebno je definirati i simbole L-sustava koji će odgovarati promjenama smjera. Simboli i slika 4 preuzeti su iz [2] i glase:

- + skreni lijevo za kut δ ($R_U(\delta)$)
- skreni desno za kut δ ($R_U(-\delta)$)
- & okreni se u dubinu za kut δ ($R_L(\delta)$)
- ^ okreni se u visinu za kut δ ($R_L(-\delta)$)
- \ povećaj nagib na lijevu stranu za kut δ ($R_U(\delta)$)
- / povećaj nagib na desnu stranu za kut δ ($R_U(-\delta)$)
- | okreni se u suprotni smjer ($R_U(180^\circ)$)

Da bi se mogle definirati grane drveta, skup simbola L-sustava proširen je za dva znaka:

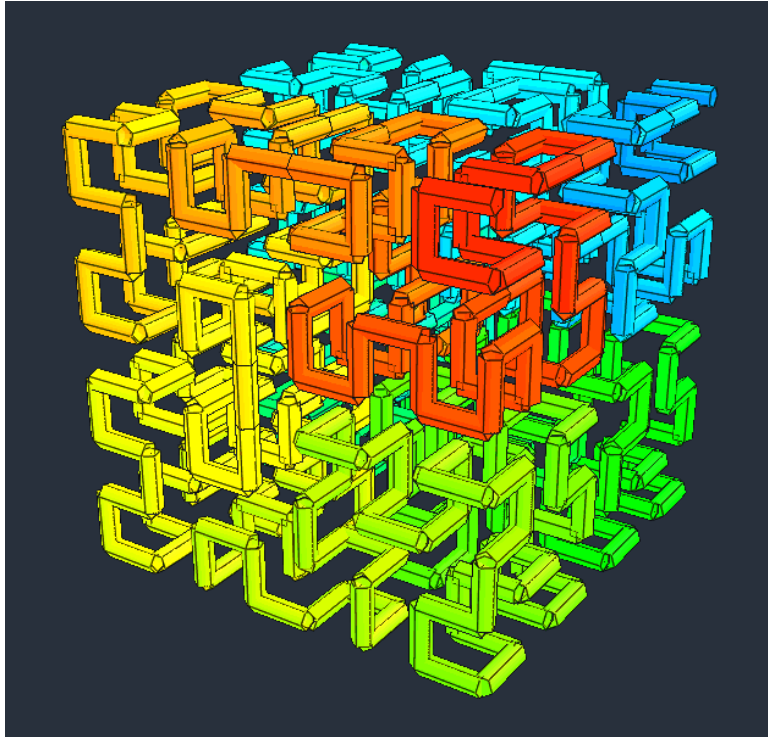
- [stavi trenutnu poziciju *turtle* pokazivača na stog
-] skini poziciju *turtle* pokazivača sa stoga



Slika 4. Vektori smjera *turtle* pokazivača.

Slika 5 prikazuje Hilbertovu krivulju u 3D prostoru generiranu pomoću slijedećeg L-sustava [2] za dvije iteracije primjena produkcijskih pravila i kut skretanja od $\delta = 90^\circ$:

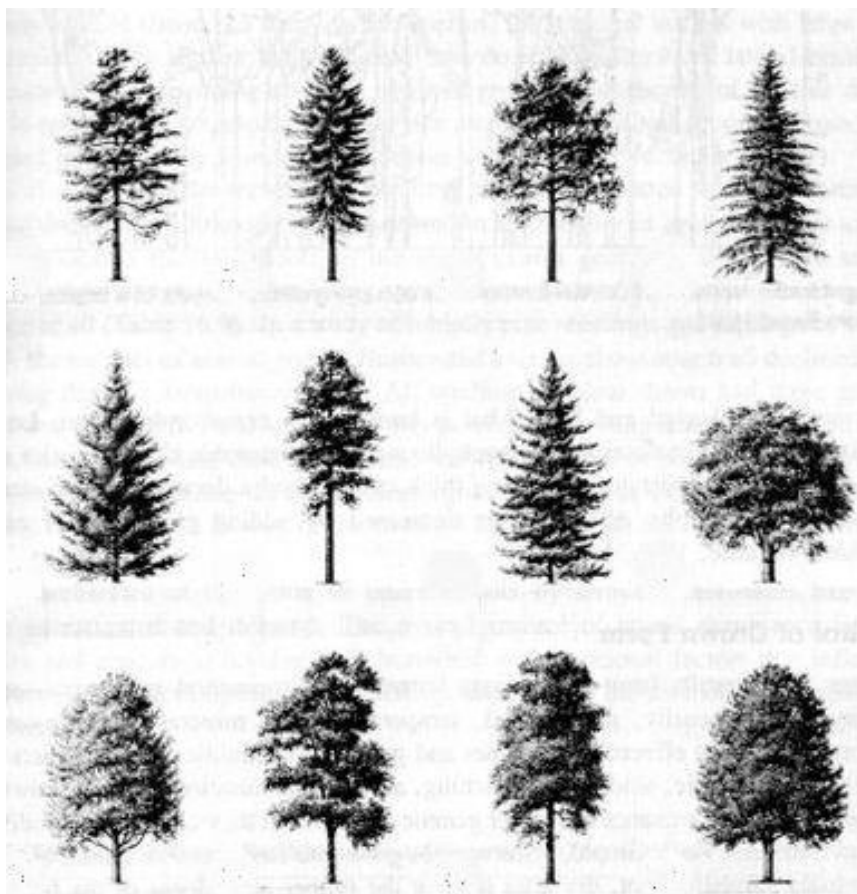
- A
- A \rightarrow B-F+CFC+F-D&F^D-F+&&CFC+F+B//
- B \rightarrow A&F^CFB^F^D^F-D^|F^B|FC^F^A//
- C \rightarrow |D^|F^B-F+C^F^A&&FA&F^C+F+B^F^D//
- D \rightarrow |CFB-F+B|FA&F^A&&FB-F+B|FC//



Slika 5. Hilbertova krivulja u 3D prostoru.

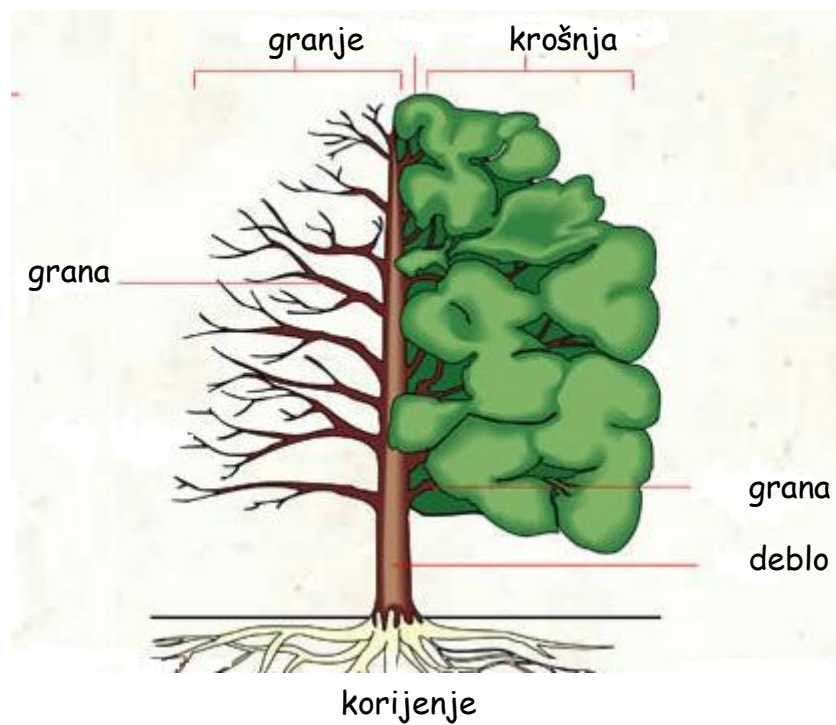
3. Klasifikacija i morfologija drveća

Stablo je trajnica koja se najčešće opisuje kao biljka koja ima jedan glavni izdanak ili deblo i sekundarne grane koje izrastaju iz tog debela i ne dotiču zemlju. Drveće se klasificira u mnogo redova i porodica biljaka, a osnovna podjela im je na listopadne i zimzelene. Procijenjeno je da postoji oko 100 000 vrsta drveća. Dolaze u svim oblicima i veličinama i pokazuju puno različitih struktura rasta (Slika 6.), oblika i veličine listova, reproduktivnih organa i strukture i izgleda kore.



Slika 6. Neke strukture rasta drveća.

Osnovni dijelovi drveta su korijen (nevidljivi dio), deblo (vidljivi drveni dio), te grane i lišće koji čine krošnju drveta (Slika 7.).

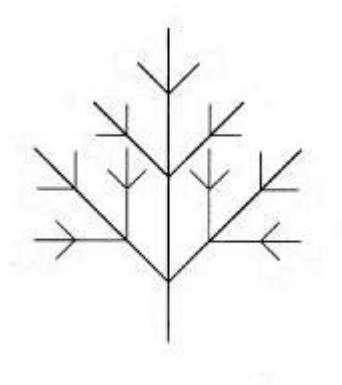


Slika 7. Struktura drveta

3.1. Strukture grananja

Postoje dvije osnovne struktura grananja drveća, monopodijalno i simpodijalno. Razlikuju se po tome da li je deblo dominantno ili su grane dominantne.

Kod monopodijalnog grananja drvo ima jednu glavnu os (deblo) koja vuče rast u visinu dok bočne osi (grane) rastu iz te osovine i zaostaju u rastu (Slika 8.). Drvo ima oblik stošca, a primjer su jasen, hrast (Slika 9. a) i četinjače (Slika 9. b, smreka).

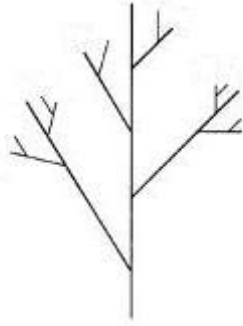


Slika 8. Princip monopodijalnog grananja.



Slika 9. Primjer monopodijalnog grananja u prirodi.

Druga vrsta grananja, simpodijalno grananje, može se prepoznati po tome što glavna os (deblo) ne raste konstantno prema vrhu već se njezin rast prekida, a jedna od bočnih osi (grana) preuzima pravac rasta glavne osi (Slika 10.). Preuzimanje rasta se ponavlja tijekom rasta drveta. Primjer simpodijalnog rasta drveta prikazan je na slici 11.



Slika 10. Princip simpodijalnog grananja.



Slika 11. Primjer simpodijalnog grananja u prirodi.

4. Programska implementacija

4.1. Korištene tehnologije

Programsko rješenje implementirano je u programskom jeziku C# 3.0 i prevedeno koristeći Microsoft Visual Studio 2008. Za crtanje korišten je OpenGL API (*Application Programming Interface*), GLUT i GLU kojima se pristupa preko Tao Frameworka.

OpenGL (Open Graphics Library) je višeplatformski i višejezični API za pisanje aplikacija koje renderiraju 2D i 3D grafiku, a sastoji se od više od 250 različitih funkcija kojima se mogu crtati kompleksne 2D i 3D scene od grafičkih primitiva. OpenGL razvio je Silicon Graphics Inc. 1992. godine, a koristi se u virtualnoj stvarnosti, simulacijama letenja, vizualizaciji podataka, CAD-u (eng. *Computer-aided design*) i dr.

GLUT ili The OpenGL Utility Toolkit je dodatna datoteka za OpenGL koja prvenstveno služi za komunikaciju s operativnim sustavom kao npr. stvaranje i kontrola prozora za crtanje, funkcije za prihvaćanje ulaza s miša i tipkovnice i sl. Stvoren je s namjenom da olakša programiranje i omogući pisanje višeplatformskog koda.

OpenGL Utility Library ili GLU je biblioteka koja uobičajeno dolazi s osnovnim paketom OpenGL biblioteke, a sadrži metode za crtanje primitiva više razine nego što to omogućuje sam OpenGL. U ovom radu koristi se za crtanje valjaka koji su segmenti generiranog drveća.

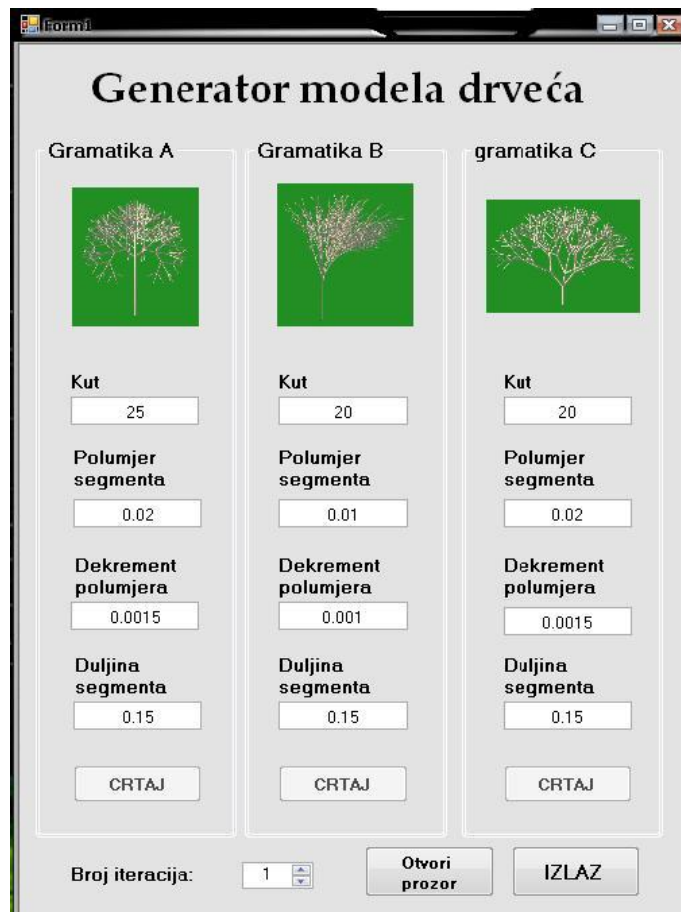
Tao Framework je biblioteka (eng. *library*) za programski jezik C# koji omogućuje .NET developerima pristup OpenGL i GLUT bibliotekama. Da bi se mogao koristiti potrebno je u C# projekt pod reference dodati Tao.FreeGlut i Tao.OpenGl zaglavlja. Nakon toga u svaku klasu koja koristi OpenGL ili GLUT naredbe potrebno je dodati dvije *using* direktive:

```
using Tao.FreeGlut;  
using Tao.OpenGl;
```

Nakon što je to podešeno sve je spremno za korištenje OpenGL, GLUT i GLU naredbi koje u C#-u počinju sufiksom `Gl.`, `Glut.` ili `Glu.` ovisno o kojoj se biblioteci radi.

4.2. Grafičko sučelje

Grafičko sučelje (Slika 12.) rađeno je u *windows forms application* projektu. Sučelje sadrži tri grupe parametara, za svaki L-sustav po jednu grupu. Parametri koje korisnik može unijeti su sljedeći: kut rotacije, polumjer segmenta, dekrement polumjera segmenta i duljina segmenta. Dubina rekurzije odabire se na samo jednome mjestu i trenutno odabrana vrijednost vrijedi za sve L-sustave. Za svaku grupu prikazana je slika modela koji se iscrta za korisniku predložene parametre.



Slika 12. Grafičko sučelje.

Grafičko sučelje sadržava gumb (*Otvori prozor*) pomoću kojega je potrebno prvo otvoriti OpenGL prozor u kojem će se crtati modeli drveća. Nakon što se otvori prozor za crtanje korisnik može unijeti parametre za neki od L-sustava ili može iskoristiti predložene vrijednosti. Kada su vrijednosti parametara unesene i odabrana je željena dubina rekurzije potrebno je pomoću gumba *Crtaj* pokrenuti iscrtavanje modela u prozoru za crtanje. Za svaki L-sustav postoji po jedan gumb za crtanje pa je potrebno aktivirati onaj koji pokreće metodu za crtanje željenog L-sustava. Nakon što je željeni model iscrtan moguće je iscrtati drugi model ponovnim odabirom parametara i dubine rekurzije te pritiskom ne jedan od tri gumba za crtanje L-sustava. Za izlazak iz generatora postoji gumb *IZLAZ*.

4.3. Implementacija L-sustava

Implementirani L-sustavi preuzeti su s Internet stranice [3] i ima ih ukupno 3 te su im dodijeljeni nazivi A, B i C. Dorađeni su u skladu sa znakovljem preuzetim iz [2], te malo izmijenjeni za potrebe ovoga rada. Metode za crtanje osmišljene su rekurzivno čime se simulira primjena produkcija na niz znakova. Svaki od tri implementirana L-sustava ima zasebnu klasu u kojoj su implementirane metode za svaku od produkcija. Na primjeru L-sustava A objašnjen je princip crtanja drveta.

4.3.1. Primjer metode za crtanje

Metoda za crtanje L-sustava A izabrana je kao primjer zato što L-sustav A ima samo jednu produkciju pa time ima i samo jednu metodu za crtanje. Metoda se unutar klase L-sustava A prvi put poziva na slijedeći način:

```
CrtanjeA(kut, polumjer, dekrement, duljina, brojIteracija);
```

Argumente metode za crtanje unosi korisnik preko grafičkog sučelja i znače slijedeće:

kut	kut rotacije oko koordinatnih osi Kartezijevog koordinatnog sustava
polumjer	polumjer segmenta stabla
dekrement	veličina za koju se smanjuje polumjer segmenta u svakoj razini rekurzije
brojIteracija	broj iteracija rekurzije koji se smanjuje kako se dubina rekurzije povećava

Za svaki idući poziv metoda se poziva za isti kut, dekrement i duljinu segmenta, dok se polumjer segmenta i brojiteraciju smanjuju. Nakon što se na početku metode argumenti učitaju u lokalne varijable, provjerava se da li je *brojiteracije* jednak korisnički zadanom broju iteracija rekurzije. Ako je jednak, znači da treba iscrtati početno stanje sustava definirano L-sustavom. L-sustava A glasi:

ω : fffffA

f crtaj segment

A \rightarrow f[\\A][//A]---A

Za početno stanje će se iscrtati pet segmenata drveta koji čine deblo drveta. Nakon provjere treba li iscrtati početno stanje, iscrta se segment drveta pomoću funkcije:

```
Glu.gluCylinder(valjak, (float)polumjer, (float)polumjer, (float)duljina,
10, 10);,
```

Prvi argument funkcije je objekt tipa `GLUquadric`, zatim slijede polumjeri kružnica na bazi i na vrhu valjka, visina valjka, broj podjela kružnice oko z koordinatne osi i broj podjela valjka duž z koordinatne osi. Kada se stvori novi valjak nalazi se duž z koordinatne osi u pozitivnom smjeru sa središtem baze u točki (0, 0, 0) pa ga je potrebno rotirati za 90° oko x koordinatne osi.

Nakon prve provjere slijedi provjera da li je dubina rekurzije dosegla nulu. Kada dosegne dubinu nula, iscrtava se segment drveta i vraća se na jednu razinu više u rekurziji i nastavlja se s rekurzivnim pozivima. Ako trenutna dubina rekurzije nije nula izvršava se slijedeći dio koda koji odgovara produkciji L-sustava nakon iscrtavanja segmenta na početku produkcije:

```
Gl.glTranslatef(0.0f, 0.0f, (float)duljina);
Gl.glPushMatrix();

Gl.glRotatef((float)kut, 1.0f, 0.0f, 0.0f);
Gl.glRotatef((float)kut, 1.0f, 0.0f, 0.0f);

CrtanjeA(kut, polumjer - dekrement,
        dekrement, duljina, brojIteracija - 1);
```

```

Gl.glPopMatrix();

Gl.glPushMatrix();

Gl.glRotatef(-(float)kut, 1.0f, 0.0f, 0.0f);
Gl.glRotatef(-(float)kut, 1.0f, 0.0f, 0.0f);

CrtanjeA(kut, polumjer - dekrement,
         dekrement, duljina, brojIteracija - 1);

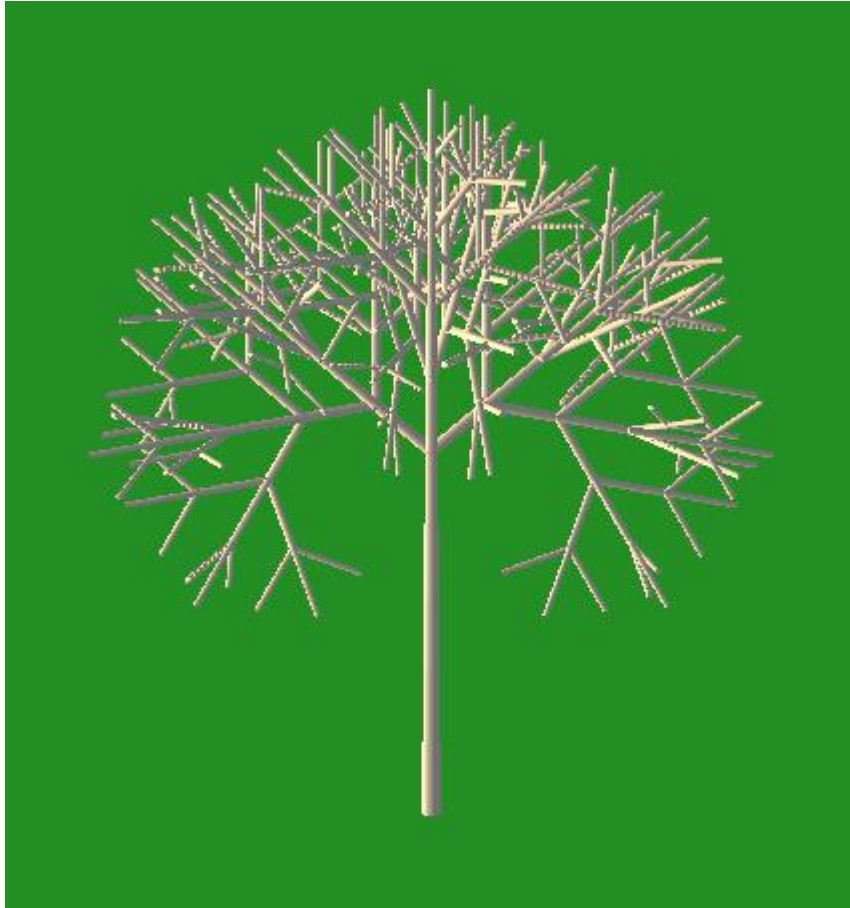
Gl.glPopMatrix();

Gl.glRotatef(-(float)kut, 0.0f, 0.0f, 1.0f);
Gl.glRotatef(-(float)kut, 0.0f, 0.0f, 1.0f);
Gl.glRotatef(-(float)kut, 0.0f, 0.0f, 1.0f);

CrtanjeA(kut, polumjer - dekrement,
         dekrement, duljina, brojIteracija - 1);

```

`Gl.glPushMatrix();` odgovara simbolu [iz skupa simbola L-sustava i sprema trenutnu matricu transformacije na stog, dok `Gl.glPopMatrix();` odgovara simbolu] i skida matricu transformacije sa stoga te ju koristi kao trenutnu matricu transformacije. `Gl.glRotatef((float)kut, 1.0f, 0.0f, 0.0f);` odgovara rotaciji za neki kut oko željene koordinatne osi, a u ovom slučaju za negativnu vrijednost kuta oko koordinatne osi x. `CrtanjeA(kut, polumjer - dekrement, dekrement, duljina, brojIteracija - 1);` je rekurzivni poziv funkcije za crtanje i zapravo predstavlja primjenu produkcijskog pravila na niz znakova. Funkcija se zaustavlja kada se rekurzija za zadnji znak za koji se može primijeniti produkcija izvrši do kraja. Za dubinu rekurzije pet, funkcija za crtanje L-sustava A pozvan s vrijednostima 25° za kut, 0.02 za polumjer segmenta, 0.0015 za dekrement polumjera i 0.15 za duljinu segmenta daje model prikazan na slici 12.



Slika 13. Model stabla L-sustava A.

5. Rezultati

Dobiveni rezultati prikazani su na primjeru L-sustava C (Slika 13) koji glasi:

$$\omega: fA$$

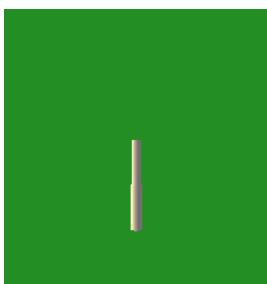
$$A \rightarrow f-B---B-----B$$

$$B \rightarrow f[/f-----A]$$

za vrijednosti parametara u tablici 1:

Tablica 1. Parametri za primjer L-sustava C

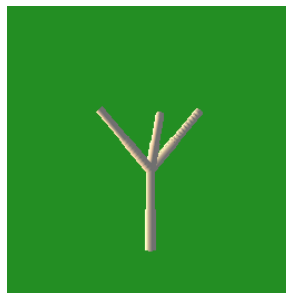
Kut	Polumjer segmenta	Dekrement polumjera	Duljina segmenta	Dubina rekurzije
20°	0.02	0.0015	0.15	10



dubina = 1



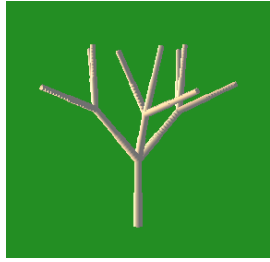
dubina = 2



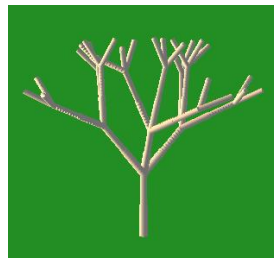
dubina = 3



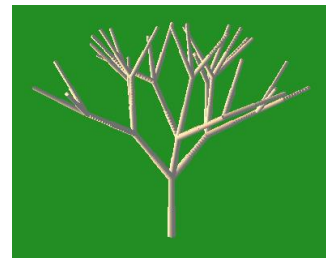
dubina = 4



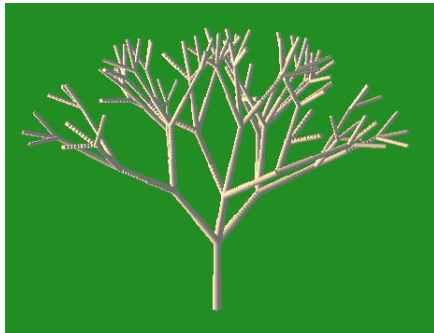
dubina = 5



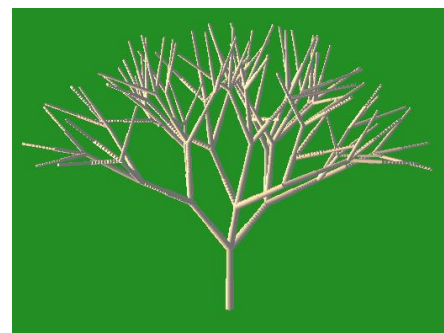
dubina = 6



dubina = 7



dubina = 8



dubina = 9



dubina = 10

Slika 14. Prikaz rezultata L-sustava C za različite dubine rekurzije.

Tri implementirana L-sustava su deterministička i kontekstno neovisna. Unatoč tim ograničenjima daju zadovoljavajuće rezultate. Za manje dubine rekurzija slika se za sva tri L-sustava iscrtava trenutačno. Pri većim dubinama vrijeme crtanja raste i za svaki L-sustav je drugačije, npr. za L-sustav B pri dubini rekurzije 10 potrebno je 2-3 sekunde za iscrtavanje dok za L-sustav C se slika za istu dubinu iscrtava trenutno, ali za dubinu rekurzije 15 njeno vrijeme iscrtavanja raste na približno jednu sekundu (Tablica 2.). Procjena je rađena na konfiguraciji koja sadrži NV 6600 GT 256 MB GPU, Athlon64 3500+, 1 GB RAM i HDD od 160 GB, 7.2k RPM.

Tablica 2. Približno vrijeme iscrtavanja za neke dubine rekurzije.

Dubina rekurzije	Vrijeme iscrtavanja (~sekunde)		
	A	B	C
6	0	0	0
9	1	1.5	0
10	1	2	0
12	2	3	0
14	7	10	1
15	13	25	1

Generator modela drveća mogao bi se dodatno poboljšati implementiranjem stohastičkih i kontekstno ovisnih L-sustava čime bi se postiglo da drveće ne izgleda "pravilno" i za zadane parametre uvijek identično. Dodatno poboljšanje generatora bilo bi mogućnost crtanje lišća na drveću bilo pomoću dodatnih L-sustava ili gotovih objekata listova.

6. Zaključak

Modeli drveća potrebni su u mnogim strukama što njihovu lakšu i jednostavniju izradu čini vrlo zanimljivom i izazovnom sa stanovišta računalne grafike. Do sada uvjerljivo najmoćniji sustavi za izradu modela drveća pokazali su se Lindenmayerovi sustavi ili L-sustavi. Već za najjednostavnije determinističke i kontekstno neovisne L-sustave generiraju se realni modeli drveća i to za malen broj primjena produkcijskih pravila. Da bi modeli izgubili svoju "pravilnost" koriste se nešto malo složeniji L-sustavi koji mogu i ne moraju bit stohastički ili kontekstno ovisni.

Cilj ovoga rada bio je programski izraditi generator modela drveća s parametarskom kontrolom generiranih modela. Generator je implementiran u programskom jeziku C# pomoću OpenGL API-a i pokazao je na primjerima tri jednostavna L-sustava zadovoljavajuće rezultate koji se mogu dodatno poboljšati daljnjom doradom generatora i razvojem implementiranih L-sustava ili implementacijom novih.

7. Popis slika

Slika 1. Princip zamjene u L-sustavu.

Slika 2. Kochova krivulja.

Slika 3. Generator Kochove krivulje.

Slika 4. Vektori smjera *turtle* pokazivača.

Slika 5. Hilbertova krivulja u 3D prostoru.

Slika 6. Neke strukture rasta drveća.

Slika 7. Struktura drveta.

Slika 9. Primjer monopodijalnog grananja u prirodi.

Slika 10. Princip simpodijalnog grananja.

Slika 11. Primjer simpodijalnog grananja u prirodi.

Slika 12. Grafičko sučelje.

Slika 13. Model stabla L-sustava A.

Slika 14. Prikaz rezultata L-sustava C za različite dubine rekurzije.

8. Popis tablica

Tablica 1. Parametri za primjer L-sustava C.

Tablica 2. Približno vrijeme iscrtavanja za neke dubine rekurzije.

9. Literatura

- [1] H. Abelson, A. A. diSessa, *Turtle geometry*, Cambridge, M.I.T. Press, 1982.
- [2] P. Prusinkiewicz, A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, 1990.
- [3] <http://www.geekyblogger.com/2008/04/tree-and-l-system.html>, lipanj 2009.
- [4] L-system, http://en.wikipedia.org/wiki/Lindenmayer_system, lipanj 2009.
- [5] Branching: L-system Tree, <http://www.mizuno.org/applet/branching/#src>, lipanj 2009.
- [6] L-System Plant Geometry Generator,
<http://www.nbb.cornell.edu/neurobio/land/OldStudentProjects/cs490-94to95/hwchen/>, lipanj 2009.
- [7] An Introduction to Lindenmayer Systems,
http://www.biologie.uni-hamburg.de/b-online/e28_3/lsys.html, lipanj 2009.
- [8] Modeling with GLU and GLUT,
<http://www.cs.csustan.edu/~rsc/SDSU/Modeling.GLU.GLUT.pdf>, lipanj 2009.
- [9] http://www.infovisual.info/01/002_en.html, lipanj 2009.
- [10] Tree, <http://en.wikipedia.org/wiki/Tree>, lipanj 2009.

10. Sažetak

Generiranje modela drveća

Najmoćnija i najjednostavnija metoda za generiranje modela drveća su Lindenmayerovi sustavi ili L-sustavi. Oni su podskup formalnih gramatika i zasnivaju se na paralelnoj primjeni što je više moguće produkcijskih pravila odjednom. Izmislio ih je Aristid Lindenmayer i do danas su se razvili u vrlo jak alat za generiranje modela biljaka i simulaciju njihova rasta. U ovom radu načinjena je programska implementacija generatora modela drveća koji koristi jednostavne determinističke i kontekstno neovisne L-sustave za generiranje modela drveća. Umjesto direktnog rada na nizovima znakova L-sustava generator je implementiran rekurzivnim pozivima metode za crtanje.

Ključne riječi: Lindenmayerovi sustavi, L-sustavi, generiranje modela drveća, modeliranje

Abstract

Generation of tree models

The most powerful and easiest method for generating 3D models of trees are Lindenmayer systems or L-systems. They are a variant of formal grammar and are based on parallel string rewriting system. They were invented by Aristid Lindenmayer and have since grown into a powerful tool for generating plant models and simulating their growth. The implementation of L-systems in a tree model generator described in this thesis generates and displays a 3D model generated from one of the three deterministic and context-free L-systems that were implemented. Instead of implementing a string rewriting system, the model generator implements recursive calls of the method for drawing models.

Key words: Lindenmayer systems, L-systems, generating tree models, modeling