

A AB ABA

Algoritmit kuvanmuodostajina

Tiivistelmä

Tekijä(t) Jyrkänne, Pilvi	Julkaisun laji Opinnäytetyö, AMK	Valmistumisaika 2022
	Sivumäärä 18	
Työn nimi A AB ABA Algoritmit kuvanmuodostajina		
Tutkinto ja koulutusala Kuvataiteilija (AMK)		
Toimeksiantajan nimi, titteli ja organisaatio (jos opinnäytetyöllä on toimeksiantaja)		
Tiivistelmä Opinnäytteen aiheena oli kuvien muodostaminen L-systeemin lukusääntöä noudattaen. L-systeemi on kasvien ja organismien rakennetta ja kasvua kuvaava matemaattinen malli, jonka toimintaperiaatteena on lausekkeiden eli merkkijonojen uudelleenkirjoitus. Tämä toteutettiin kirjoittamalla ohjelma, joka satunnaislukugeneraattorin avulla luo näitä lausekkeitä, joissa jokaisella merkillä on oma tarkoitus. Ohjelma tulkitsee lausekkeiden merkit käyttäen kilpikonnagrafiikkaa, joka sijainnin, viivan ja suunnan saatuaan piirtää merkkijonon muodostaman kuvan kahdessa ulottuvuudessa. Tavoitteena oli luoda tällä ohjelmalla teossarja, josta tulee esiin L-systeemin muodostamien kuvien monipuolisuus ja luoda niistä sarjallinen kokonaisuus, jonka osat ovat myös itsenäisiä teoksia. Ohjelman muodostamia kuvia varten tehtiin taustoiksi digitaalisia maalauksia. Kuvien yhteyteen lisättiin myös tekstiosioina säännöt, joista kuva muodostuu. Taustat jätettiin yksinkertaisiksi väritaustoiksi korostamaan ohjelman tekemien kuvien viivapiirrosmaisuutta. Väreiksi niille valikoitui perusvärit ja harmaan sävyt. Valmiit teokset vedostettiin laadukkaalle valokuvapaperille 19 kuvan sarjaksi. Vedokset ripustettiin seinästä ulkoneviin ruuveihin magneeteilla, näin saatiin aikaan haluttu syvyysvaikutelma teosten ja seinän välille.		
Asiasanat L-systeemi, ohjelmointi, kuvanmuodostus, algoritmien taide		

Abstract

Author(s) Jyrkänne, Pilvi	Type of Publication Thesis, UAS	Published 2022
	Number of Pages 18	
Title of Publication A AB ABA Generating images with algorithms		
Degree and field of study Bachelor of Culture and Arts, Fine Arts (UAS)		
Name, title and organisation of the client (if the thesis work is commissioned by another party)		
Abstract The subject of this thesis was to form pictures by following the rules of L-system. L-system is a mathematical theory used to model the growth and structure of plants and organisms. This was achieved by writing a program that randomly generates mathematical expressions, in which each symbol has its own function. The program interpreted the symbols into pictures by using turtle graphics, that was also written into the program. Turtle graphics draws in two dimensions, once given the coordinates, a pen (pixels, or line width) and a direction. The goal with this program was to create a series of artworks, that would bring forth the diversity of working with L-system, where the pieces form a connected series, but they also are independent artworks. The backgrounds of the pieces were digitally painted simple colour backgrounds. Each work has mainly one colour and the colours chosen were primary colours and grey. The set of rules was also included in the artwork, so that each piece has both the picture of the L-system and the rules that formed it. The artworks were printed on a high quality photograph paper as a series of 19 prints. They were attached to the wall with magnets to avoid leaving holes or marks in them. There were screws in the wall for the magnets, to create space between the wall and the artworks.		
Keywords L-system, programming, algorithmic art		

Sisällys

1	Johdanto.....	1
2	Algoritmisen taiteen historiaa	2
3	Työvälineet	5
3.1	L-systeemit	5
3.2	Godot Engine ja GDScript-kieli	6
4	Kuvan muodostuminen	7
4.1	Ohjelman kirjoittaminen	7
4.2	Satunnaisuus	9
5	Teosten rakentuminen	12
5.1	Taustat teoksille	12
5.2	Kuvien valitseminen	13
5.3	Kuvat ja tekstit taustoihin	13
5.4	Teosten tulostaminen ja viimeistely	14
6	Yhteenveto ja pohdinta	16
	Lähteet	18

1 Johdanto

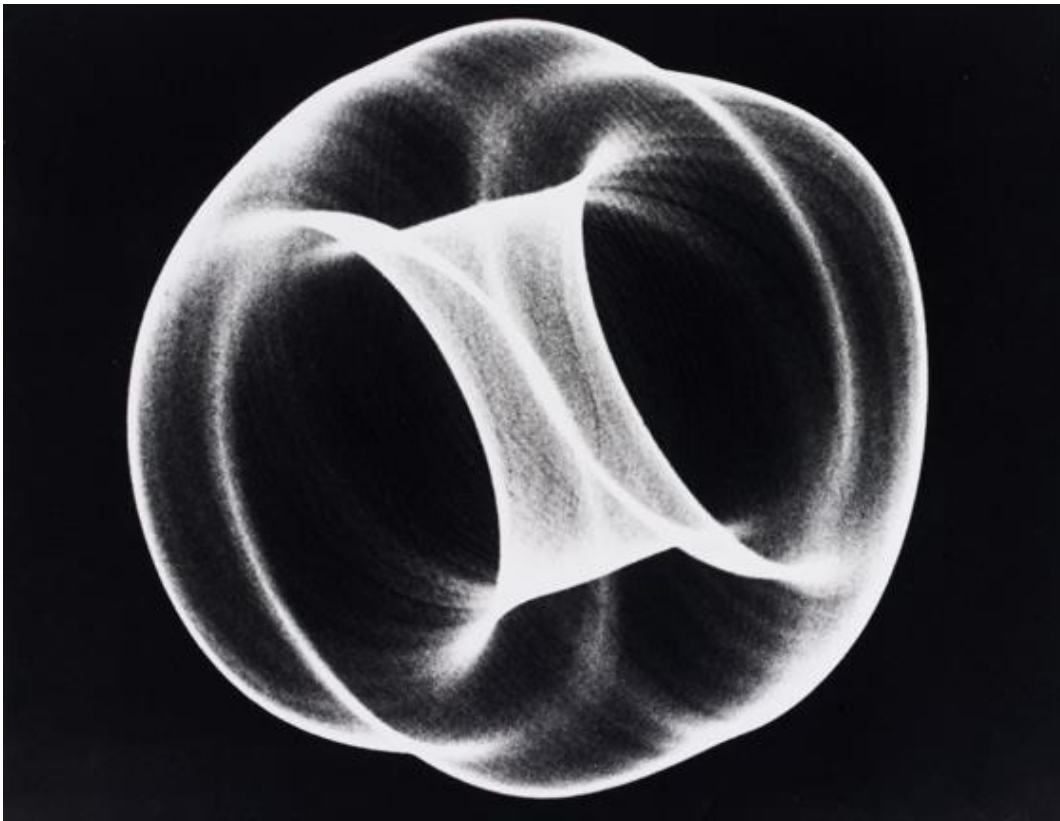
Opinnäytetyön tutkimusaiheena on kuvien muodostaminen L-systeemin ja ohjelmoinnin avulla ja vedostaa näistä kuvista teossarja. L-systeemi on matemaattinen malli, joka on kehitetty kuvaamaan organismien kasvua ja rakennetta ja tavoitteeseen pyritään kirjoittamalla ohjelma, joka L-systeemiä noudattaen satunnaisgeneroi kuvia. Työkaluna ohjelman kirjoittamiselle on pelimoottori Godot Engine ja ohjelmointikielenä sen sisäänrakennettu GDScript-kieli.

Ohjelman tuottamista kuvista valitaan vedostettavaksi sellaisia, jotka ovat sekä itselleni visuaalisesti mielenkiintoisia, että tuovat kuvan muodostumisen monipuolisuuden esiin. Lopullisiin teoksiin pyrin siis valitsemaan ohjelman tuottamia muotoja, jotka parhaiten kuvaavat sen monipuolisuutta ja asettamaan ne sommittelun ja värivalintojen avulla yhtenäiseksi kokonaisuudeksi. L-systeemin avulla muodostettuja kuvia yhdistän digitaalisesti maalattuihin pohjiin ja vedostan niistä 15 - 20 teoksen kokonaisuuden. Sisällytän lopullisiin teoksiin kuvan lisäksi sen muodostumisen säännöt tekstinä, jolloin vedoksissa on nähtävissä ohjelman tekemä malli sekä teksti- että kuvamuodossa.

Aihe toimii jatkumona aiemmalle taiteelliselle työskentelylleni ja antaa minulle tilaisuuden syventyä L-systeemiin taiteen tekemisen välineenä. Näen loputtomasti potentiaalia sen kyvyssä tuottaa niin esittäviä malleja, kuin täysin abstrakteja muotoja. Koen kiehtovana myös orgaanisessa muodossa, mutta erityisesti hyvin abstraktissa kuvassa sen tiedon, että muodon taustalla on matemaattisia lausekkeita ja tarkkaa sääntöjen noudattamista, joka herkästi jää havaitsematta pelkästään sitä katsomalla.

2 Algoritmisen taiteen historiaa

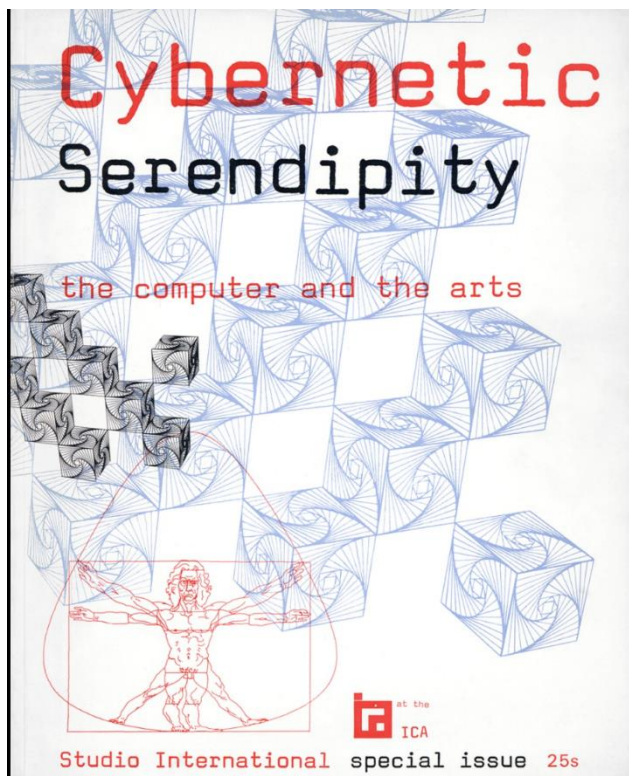
Algoritmisen ja tietokoneella luodun taiteen alku sijoittuu 1950-luvulle. Ensimmäisiä teoksia näillä tekniikoilla tekivät taiteilijoiden sijaan insinöörit, matemaatikot tai tieteilijät, koska tietokoneet eivät olleet kaikkien saatavilla ja niiden käyttäminen vaati koulutusta asiaan ja myös ymmärrystä algoritmeista. Elektronisen ja tietokoneella luodun taiteen alkuaikana tietokoneetkin olivat vielä hyvin kaukana siitä, mitä ne tänä päivänä ovat, ei ollut käyttöjärjestelmiä tai valmiita ohjelmia kuvien tekemiseen, vaan ohjelmat kirjoitettiin itse. Yhdeksi ensimmäisistä teoksista elektronisen taiteen alalla mielletään Ben Laposkyn Oscillon 40 vuodelta 1952 (Kuva 1). Teos toteutettiin oskilloskoopilla ja kuvaamalla tämän syöttämää käyrää kameran pitkällä valotusajalla. Tällaisten teosten katsotaan toimineen alkusysäyksenä tietokoneella ja algoritmien avulla tehdylle taiteelle, vaikka tekomenetelmät ja -välineet eivät vielä määritelmää täyttäneet. (Carlson 2003; Victoria and Albert Museum 2016.)



Kuva 1. Ben Laposky, Oscillon 40, 1952, oskilloskooppi ja kamera (Victoria and Albert museum)

60- ja 70-luvuilla tietokoneella ohjelmoituja teoksia alettiin vedostamaan plotterilla, joka tulostimesta poiketen piirtää viivaa kynällä tai siveltimellä. Plotteri kytkettiin tietokoneeseen ja ohjelmoitiin joko liikuttamaan plotterin kynää tai sivellintä, tai vaihtoehtoisesti se liikutti paperia ja piirtoväline pysyi paikallaan. Myös taiteilijat löysivät tämän uuden tavan tehdä taidetta ja alkoivat tehdä yhteistyötä matemaatikkojen ja insinöörien kanssa. Osa taiteilijoista alkoi myös itsenäisesti opettelemaan ohjelmointia ja pikkuhiljaa taidemaailma kiinnostui uudesta tavasta tehdä taidetta. (Carlson 2003; Victoria and Albert Museum 2016.)

Vuonna 1968 Lontoon Institute of Contemporary Arts -nykytaidemuseossa järjestettiin Cybernetic Serendipity -näyttely, joka esitteli elektronista, tietokoneella ja ohjelmoimalla tehtyä taidetta, runoutta ja musiikkia (Kuva 2). Näyttely ei ollut ensimmäinen tietokoneella luodun taiteen näyttely, mutta sitä pidetään merkittävänä tapahtumana tällä taiteen alalla ja oli tähän mennessä suurin tietokonetaidetta esittelevä näyttely. Näyttelyyn osallistui 325 tekijää, mm. taiteilijoita, insinöörejä ja matemaatikkoja. Cybernetic Serendipity kiersi Lontoon jälkeen Washingtonissa ja San Franciscossa ja auttoi tietokoneella luotua taidetta pääsemään hyväksytyksi taiteenlajiksi. (Carlson 2003; Norman.)



Kuva 2. Jasia Reichardt, Cybernetic Serendipity: the computer and the arts, 1968, (Studio International Foundation)

70-luvun loppupuolella perustettiin Apple ja Microsoft ja 80-luvulle siirryttäessä henkilökohtaiset tietokoneet alkoivat yleistymään ja olivat suuremman yleisön saavutettavissa. Tämä toimi tietokonetaiteen hyväksi nostaen sen suosiota, kun tietokoneen pystyi kohtuulliseen hintaan ostamaan kotiin. Myös tulostimien kehittyminen auttoi nostamaan taiteenlajin suosiota, kun tietokoneella luotuja kuvia pystyi nyt edullisemmin tulostamaan fyysisiksi vedoksiksi. Lopullisen läpimurron kotiin ostettavat tietokoneet kokivat 90-luvun puolivälin jälkeen. (Carlson 2003; Victoria and Albert Museum 2016.)

Tänä päivänä tietokoneella taiteen luomiseen on lukemattomia keinoja, voi ohjelmoida alusta alkaen itse, maalata digitaalisesti, luoda liikkuvaa kuvaa, tehdä kolmiulotteista taidetta tai manipuloida jo jotain olemassa olevaa. Koen, että kirjoittamalla itse ohjelmani, joka muodostaa kuvat, palaan algoritmisen taiteen juurille, kun ei ollut valmiita ohjelmia, joilla lähteä piirtämään tai luomaan kuvia.

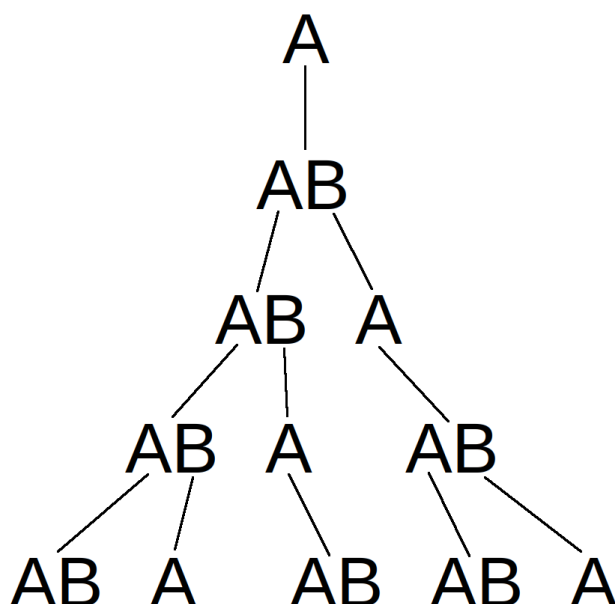
3 Työvälineet

3.1 L-systeemit

Kuvan muodostamisen lukusääntönä käytetty L-systeemi on unkarilaisen teoreettisen biologin Aristid Lindenmayerin 1960-luvulla kehittämä organismien kasvua ja rakennetta kuvaava matemaattinen teoria. Sen toimintaperiaatteena on lausekkeiden eli merkkijonojen uudelleenkirjoitus. Näissä lausekkeissa jokaisella merkillä on oma arvo, joille uudelleenkirjoituksen säännöillä voidaan antaa uusi arvo seuraavaan lausekkeeseen. (Prusinkiewicz & Lindenmayer 2004, 13) Olen aiemmassakin työskentelyssäni hyödyntänyt L-systeemiä ja käyttänyt sen avulla muodostettuja kuvia lähtökohtana teoksissa, mutta vasta tämän opin-
näytteen myötä syvennyin tarkemmin sen toimintaan ja logiikkaan.

Esimerkki uudelleenkirjoituksesta ja L-systeemistä, Algae

Algae-L-systeemi on Lindenmayerin alkuperäinen matemaattinen malli levästä. Alla havainnointikuva sen rakenteesta ja kasvusta (Kuva 3). Sen alkuluku eli aksiooma on tässä esimerkissä A. Ensimmäinen sääntö sille on $A \rightarrow AB$. Tämä tarkoittaa, että se muuttaa alkuluvun seuraavaan lausekkeeseen AB:ksi. Toinen sääntö sille on $B \rightarrow A$. Tämä muuttaa nykyisestä lausekkeesta löydetty B-arvot A:ksi.



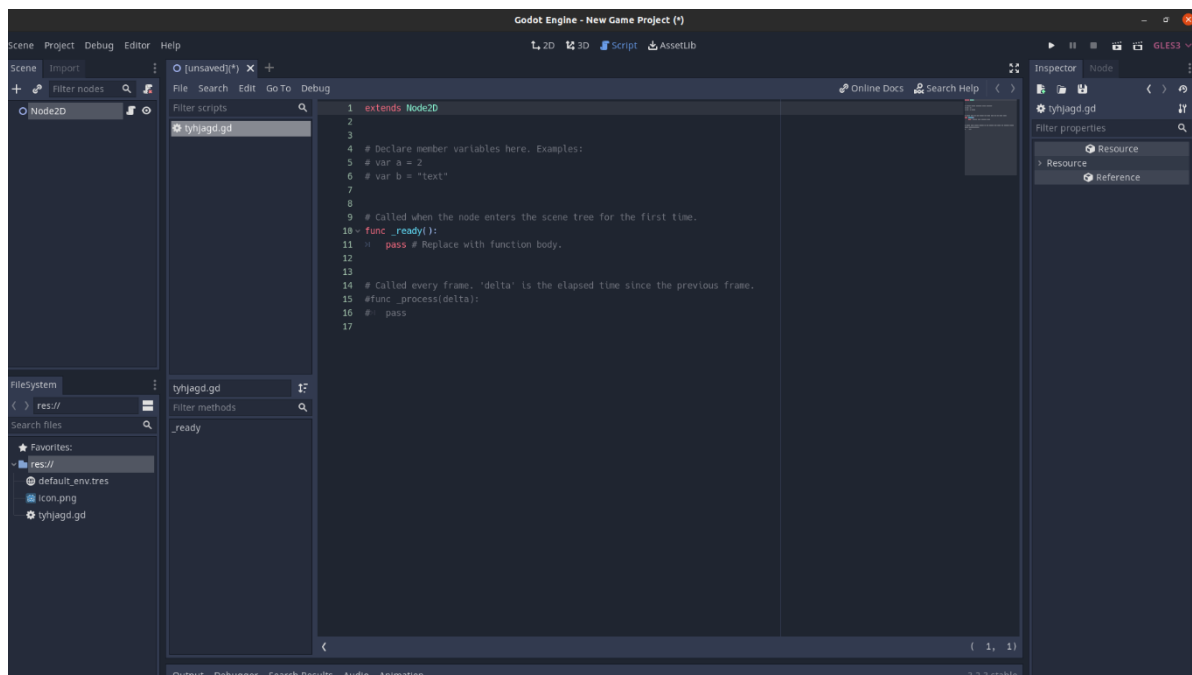
Kuva 3. Havainnointikuva algae-mallista

3.2 Godot Engine ja GDScript-kieli

Käyttämäni Godot Engine on pelimoottori, jolla sekä kirjoitan että visualisoin teokset. Valitsin Godot Enginen opinnäytetyöni tekoon, koska se on avoimen lähdekoodin moottori. Avoimen lähdekoodin ohjelmistot ovat käyttäjille ilmaisia, niillä on usein laaja kehittäjäkunta, käyttäjät ovat vapaita muokkaamaan niitä ja käyttämään niillä tekemiään asioita vapaasti.

Pelimoottoreita useimmiten nimensä mukaisesti käytetään videopelien tekoon, mutta niillä voi tehdä myös esimerkiksi kuvaa tai animaatiota. Godot Enginellä pystyy tekemään sekä kaksi- että kolmiulotteisesta mediaa, mutta teokseni ovat kaksiulotteisia kuvia. Olen aiemmassa työskentelyssäni käyttänyt Godot Engineä soluautomaation tekoon, joten se on entuudestaan melko tuttu.

Käytän ohjelmaa kirjoittaessani Godot Enginen sisäänrakennettua GDScript-ohjelmointikieltä. Ohjelman kirjoittamista varten opiskelin Python-ohjelmointikieltä, joka on komennoitua ja logiikaltaan samankaltainen tässä työssä käyttämäni GDScript-kielen kanssa ja näin nopeutti työskentelyäni Godot Enginellä. Alla kuva pelimoottorin käyttöliittymästä (Kuva 4).

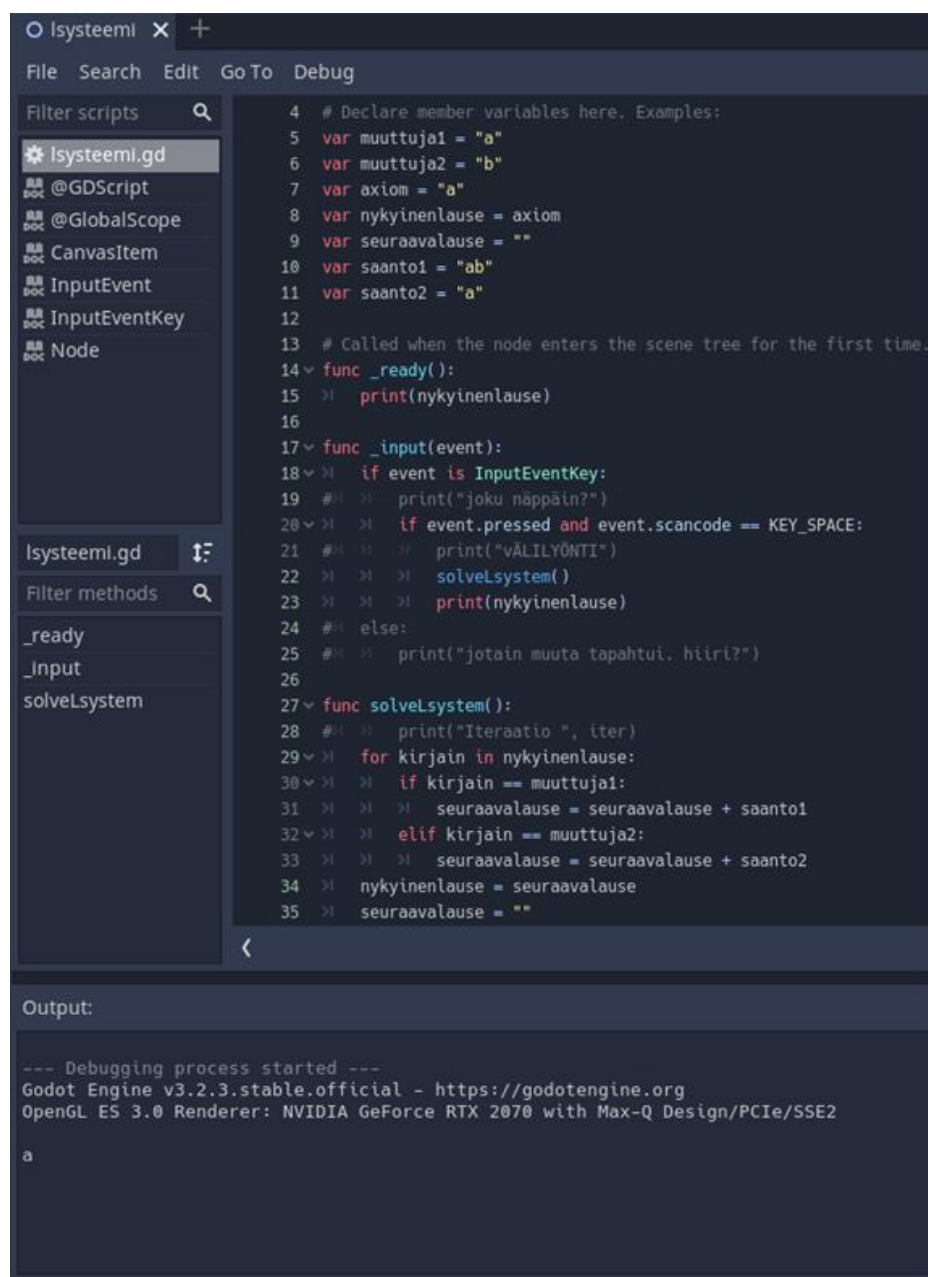


Kuva 4. Ohjelmointinäkökulma Godot Enginen käyttöliittymässä

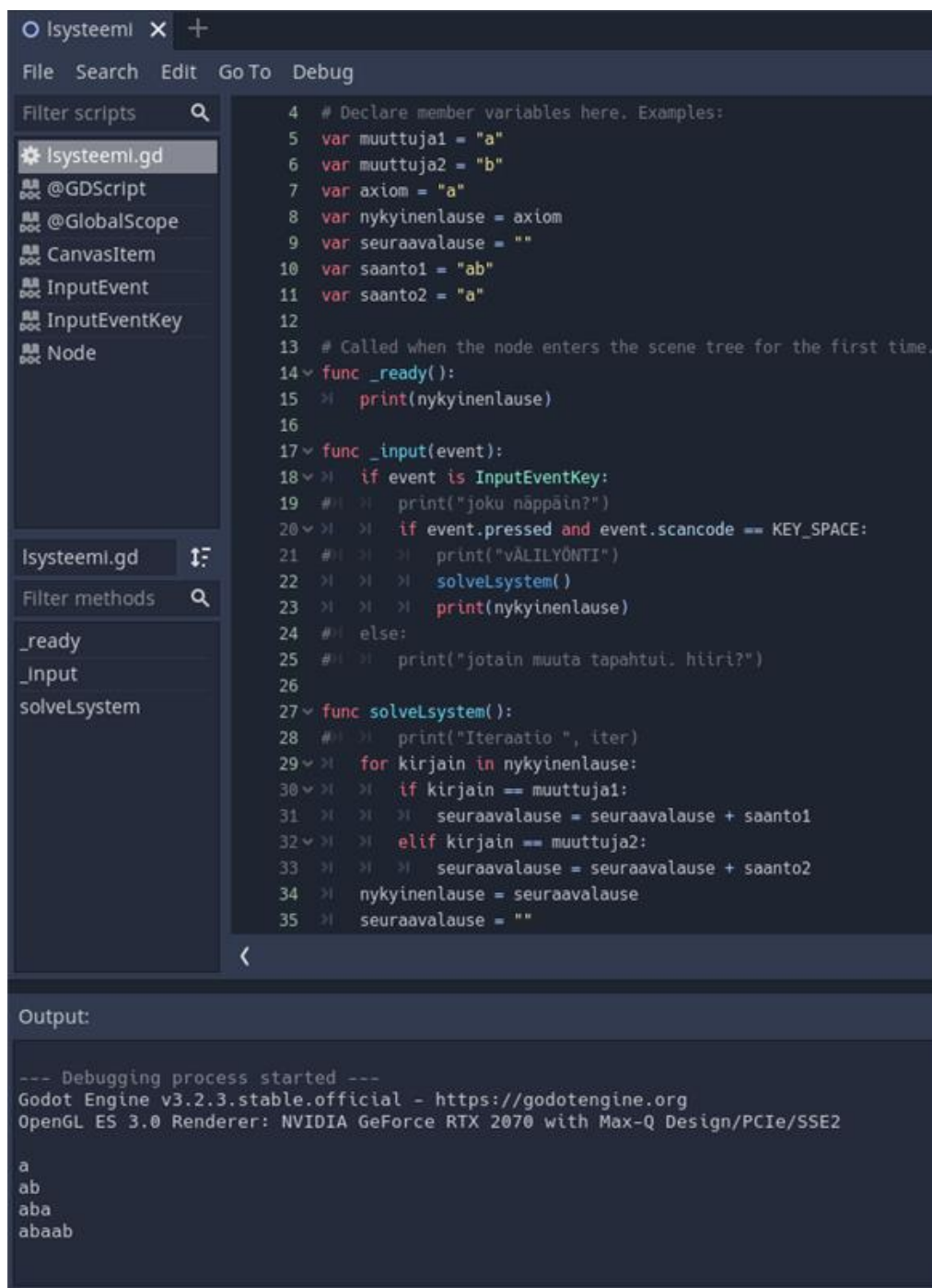
4 Kuvan muodostuminen

4.1 Ohjelman kirjoittaminen

Aloitin ohjelman tekemisen kirjoittamalla kappaleessa 2.1 mainitsemani yksinkertaisen Algae-mallin tekstinä Godot Enginellä, saadakseni L-systeemin logiikalle rungon. Ensimmäisessä kuvassa (Kuva 5) iteraatioita on 0, joten ohjelma kirjoittaa vain alkuluvun, eli aksiooman "a". Toisessa kuvassa (Kuva 6) iteraatioita on jo 3, eli ohjelma on lukenut ja kirjoittanut uudelleen lausekkeet 3 kertaa ja muodostanut näin lausekkeen abaab.



Kuva 5. Algae Godot Enginellä kirjoitettuna, 0 iteraatiota



Kuva 6. Algae Godot Enginellä kirjoitettuna, 3 iteraatiota

Tämän rungon päälle aloin rakentamaan ohjelmalle haluamiani ominaisuuksia. Kirjoitin turtle-funktion (Kuva 7), joka tarvitaan kuvan muodostumiseen. Turtle graphics on tietokoneen käyttämää vektorigrafiikkaa, joka piirtää kaksiulotteista kuvaa x- ja y-akseleilla.

Piirtäkseen se tarvitsee alkusijainnin, suunnan ja viivan pituuden (tai vaihtoehtoisesti loppusijainnin).

```
func turtle(ratkaisu):
#> print(ratkaisu)
> var sijainti = Vector2(0,0)
> var suunta = 270
> var viivat = []
> var viivanpituus = 100#px
> var risteykset = []
> ympyrat.clear()
> for merkki in ratkaisu:
> > if merkki == "X":
> > > pass
> > elif merkki == "C":
> > > ympyrat.append(sijainti)
> > elif merkki == "F":
> > > var uusisijainti = sijainti + Vector2(cos(deg2rad(suunta)) * viivanpituus, sin(deg2rad(suunta)) * viivanpituus)
> > > viivat.append([sijainti, uusisijainti])
> > > sijainti = uusisijainti
> > elif merkki == "+":
> > > suunta = suunta - kaantokulma
> > elif merkki == "-":
> > > suunta = suunta + kaantokulma
> > elif merkki == "[":
> > > risteykset.append([sijainti, suunta])
> > elif merkki == "]":
> > > var risteys = risteykset.pop_back()
> > > sijainti = risteys[0]
> > > suunta = risteys[1]
> return viivat
```

Kuva 7. Turtle-funktion lähdekoodi

Tähän funktioon määritin myös lausekkeissa käytettävien merkkien arvot: plus-merkin (+) oikealle kääntymisen ja valitun kulman muuttujaksi, miinus-merkin (-) vasemmalle kääntymisen ja valitun kulman muuttujaksi, X-arvo on tyhjä, F-muuttuja tekee viivan ja C-muuttuja piirtää ympyrän. Hakasulje "]" tallentaa sijainnin ja kulman, sen vastapari "[" palauttaa ne.

Draw-funktiolla (piirto) määritin piirrettävän viivan ominaisuudet. Väriksi valitsin mustan ja viivan paksuutta vaihtelin 1 - 10 pikselin välillä. Lisäsin tähän funktioon myös ympyrän ominaisuudet (C-muuttuja), jonka ohjelma voi satunnaisesti valita mukaan lausekkeisiin.

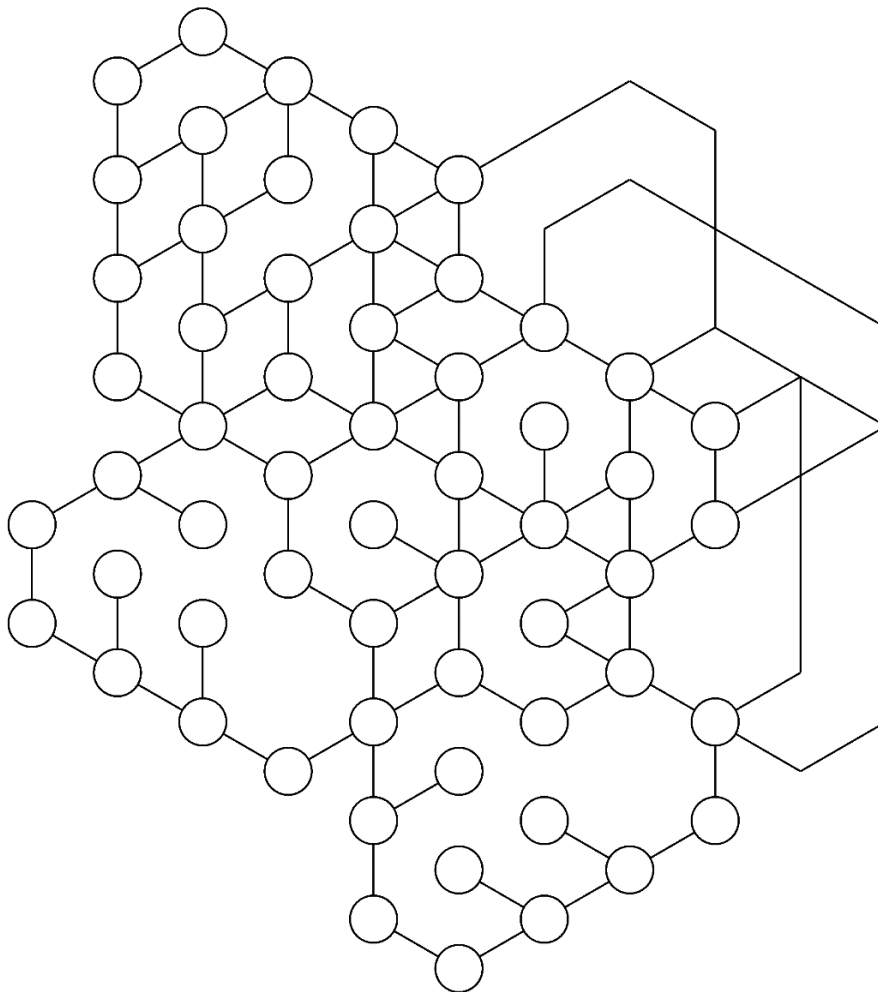
4.2 Satunnaisuus

Tein ohjelmalla kokeiluja kirjoittamalla itse lausekkeita ja sääntöjä, eli syöttämällä itse sijainnit, suunnan, viivan pituuden, käännökset ja kulman ja antamalla niille sääntöjä. Merkkien ja asteiden muuttaminen manuaalisesti oli kuitenkin hyvin hidasta eikä mielestäni tukenut tarpeeksi ajatustani ohjelmasta muodostamassa kuvia.

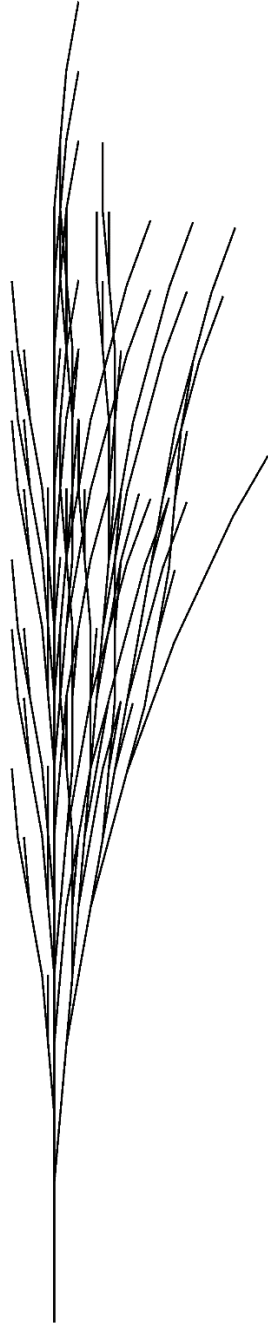
Tämän vuoksi lisäsin siihen satunnaisuuden, jonka mukaan tuominen antoi kuvien muodostumiseen huomattavasti vaihtelua ja nopeutta. Näin ohjelma myös itse valitsee, minkälaisia lausekkeita ja kuvia muodostaa. Ohjelma muodostaa lausekkeita todella nopeasti ihmiseen

verrattuna, ja ajallisen hyödyn lisäksi se myös todennäköisemmin kokeilee niistä monipuolisempia variaatioita.

Määritin turtle-funktiossa (Kuva 7) valmiiksi erilaisia arvoja, joista ohjelma valitsee satunnaisesti, mitä kulloinkin käyttää lausekkeiden muodostuksessa ja piirtää ne L-systeemin lukusääntöä noudattaen. Kääntökulman ohjelma valitsee luettelosta, jossa on valittavana eri asteita 5:n ja 160:n asteen väliltä. Yksittäiseen kuvaan se voi valita vain yhden kulman, jota noudattaa, se ei voi muuttua kesken teoksen. Alla on ohjelman satunnaisesti muodostamia kuvia (Kuvat 8 ja 9)



Kuva 8. Esimerkki ohjelman tuottamasta geometrisestä muodosta



Kuva 9. Esimerkki ohjelman tuottamasta orgaanisemmasta muodosta

5 Teosten rakentuminen

5.1 Taustat teoksille

Maalasin teoksille digitaalisesti iPadin Procreate-piirustusohjelmalla erilaisia taustoja. Aloitin tekemällä yksinkertaisia tasavärisiä pohjia ja väripaletiksi valitsin perusvärit, keltaisen, sinisen ja punaisen. Olen työskentelyssäni usein päätenyt käyttämään perusvärejä, erityisesti keltaista tai sinistä, joten koin luontevaksi tuoda myös itselleni tutun ja esteettisesti miellyttävän väripaletin opinnäytteeseen. Halusin hieman rauhoittaa värimaailmaa, joten tein mukaan myös harmaansävyisiä taustoja.

Tasaiset yksiväriset pohjat eivät kuitenkaan toimineet haluamallani tavalla ohjelman muodostamien kuvien kanssa. Yleisilme jäi mielestäni liian graafiseksi yhdistettyäni ohjelman tekemiä viivapiirrosmaisia kuvia tasaväriseen pohjaan. Kokeiltuani erilaisia taustoja, päädyin pitämään eläväisempää pohjaa parhaana komplementtina ohjelman piirtämälle viivalle. Taustat ovat näennäisen yksivärisiä, mutta tein niihin sävy- ja rakennevaihtelua, jotta niissä olisi orgaanisempaa tunnetta (Kuva 10).

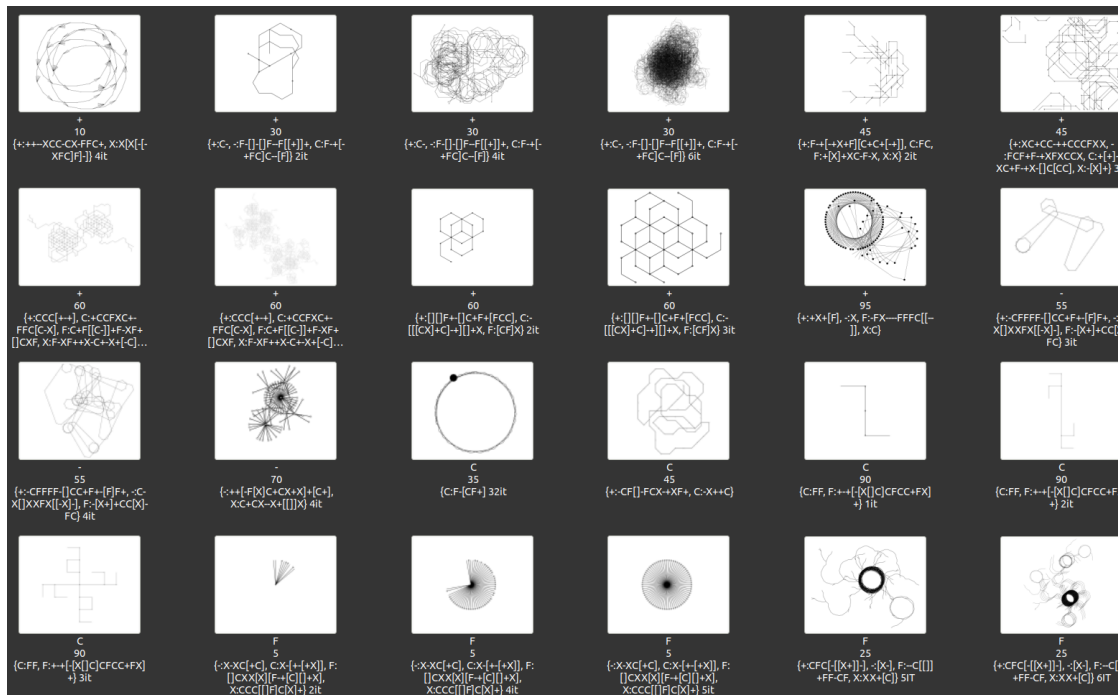


Kuva 10. Sininen tausta

5.2 Kuvien valitseminen

Ohjelman muodostamista kuvista valikoin muotoja, jotka olisivat mahdollisimman monipuolisia toisiinsa verrattuina, mutta myös yksilöinä mielenkiintoisia. Tuhansista ohjelman muodostamista kuvista ja säännöistä tallensin valikoitavaksi noin sata potentiaalista. Tämän jälkeen karsin niitä kokonaisuutta miettien, valikoiden niistä 20, jotka antaisivat monipuolisen näytteen L-systeemin kyvystä muodostaa kuvia. Halusin lopulliseen teossarjaan mukaan sekä pehmeää, orgaanista jälkeä, että geometrisiä, mekaanisempia kuvia.

Valintaprosessissa huomasin herkästi sokeutuvani yksittäisten kuvien visuaalisille ominaisuuksille. Pystyin paremmin arvioimaan teosten muodostamaa kokonaisuutta katselemalla useampia kuvia pieninä kuvina samanaikaisesti ruudulla (Kuva 11). Tällä menetelmällä yksittäiset teokset vakiinnuttivat paikkansa kokonaisuudessa.



Kuva 11. Laajempi kuva ohjelman tuottamista L-systeemeistä

5.3 Kuvat ja tekstit taustoihin

Ohjelman tekemien kuvien lisäksi liitin jokaiseen teokseen tekstiosion, joista tulee ilmi, mistä kuva on muodostunut. Tekstiosioihin lisäsin kuvan alkuluvun, kääntökulman, iteraatiot ja säännöt, eli kaavan, jolla kuva on muodostunut ja jotka syöttämällä ohjelmaan se tekee saman kuvan uudestaan (Kuva 12).

Tekstin asettelulla kuvaan loin teosten välille sommitelmallista yhtenäisyyttä, käytin jokaisen teoksen tekstiosioissa samoja raameja, fonttia ja fonttikokoa. Tekstiosioden rakentamiseen otin vaikutteita teknisessä piirustuksessa käytetyistä otsaketaulukoista. Teknisen piirustuksen taulukoiden funktionaalinen muotokieli tukee mielestäni hyvin ajatusta teosteni koneellisesta muodostumisesta.

a x i o m	F
a n g l e	7 ∅
i t e r a t i o n s	4
r u l e 1	C : + [F X] F F - C X - F
r u l e 2	F : + [F - X [] X F + + F [+ C F + -]]

Kuva 12. Esimerkki teosten tekstiosioista

Kuvien sommittelussa taustoihin vältin keskeissommittelua, jotta tekstiosioille jäisi riittävästi tilaa. Tekstilaatikoiden ja kuvien visuaalisesta erosta huolimatta tahdoin pitää niiden keskinäisen suhteen kussakin teoksessa avoimena niin, etteivät ne häiritse toistensa huomioarvoa vaan muodostavat tasapainoisen parin. Pyrin pitämään sommittelun mahdollisimman staattisena, joskin tämä osoittautui haastavaksi monien kuvien kanssa, niissä itsessään olevan liikkeen tunnun vuoksi. Kuvien graafisen vaikutelman hälventämiseksi lisäsin muodoille varjot ja huomasin tämän myös tuovan kuvan lähemmäs katsojaa.

5.4 Teosten tulostaminen ja viimeistely

Tulostin teokset A2-kokoiselle mattapintaiselle valokuvapaperille ja leikkasin papereihin n. 0,5 senttimetrin marginaalit. Tein myös yhdelle teokselle kokeilun leikkaamalla marginaalit pois, mutta koin sen välittömästi hieman julistemaiseksi, joten päädyin käyttämään lopullisissa töissä marginaaleja.

Valitsin pintarakenteeltaan mahdollisimman karhean paperin tuodakseni lisää tekstuurin tuntua lopullisiin vedoksiin. Valitsemani paperi toimi teoksissa hyvin, se lisäsi hieman toivo-
maani pintarakennetta niihin, mutta rakennevaihtelu ei ollut kuitenkaan liian suurta vaikut-
taakseen pienempien yksityiskohtien tarkkuuteen tai tekstin luettavuuteen. Alla on kuva val-
miista teoksesta (Kuva 13).



Kuva 13. Valmis teos sarjasta

6 Yhteenveto ja pohdinta

Opinnäytteen tavoitteena oli tehdä kuvia muodostava ohjelma ja koostaa sen tuotoksista teossarja, joka tuo esiin kuvan muodostamisen monipuolisuuden L-systeemin avulla. Kasvien muodostumista ja rakennetta kuvaavana teoriana L-systeemillä usein on käytössään huomattava määrä arvoja lausekkeiden muodostumiseen ja iteraatiot voivat kasvaa todella suuriksi. Ohjelmallani kuitenkin rajoitin huomattavasti sen toimintakykyä asettamalla sille piirrettäväksi vain viivan ja ympyrän, rajoittamalla sen merkkien ja sääntöjen maksimimäärää ja antamalla sille vain tietyjä asteita valittavaksi käänntökulmia varten. Tämä huomioon otettaessa, mielestäni opinnäytteen lopputulos antaa monipuolisen näytteen L-systeemin sääntöjä noudattavista kuvista ja jo prosessin aikana olin tyytyväinen ohjelman tuottamien muotojen vaihtelevuuteen, enkä nähnyt tarpeelliseksi antaa sille laajempia vapauksia kuvan muodostamiseen.

Töiden ripustuksessa jouduin karsimaan teoksia kokonaisuudesta tilallisista ja sommitelmallisista syistä. Teosten installoinnissa täytyi tilan lisäksi miettiä niiden suhdetta toisiinsa, miten värit sijoitellaan ja kuinka yhden teoksen muodot vaikuttavat seuraavaan teokseen. Teokset pyrittiin sommittelemaan niin, että sekä yksittäisten vedosten että kokonaiskuvan liikesuunnat ohjaavat katsojan silmää luonnollisesti vasemmasta yläkulmasta oikeaan alakulmaan. Teosten installointi toteutettiin kiinnittämällä seinään ruuveja, joiden kantojen päälle teokset asetettiin magneetteja käyttäen. Ruuvit jätettiin tarkoituksenmukaisesti n. 2 senttimetriä ulos seinän pinnasta. Näin teosten ja seinän väliin saatiin tila, jolla korostettiin jo kuvioiden varjoilla haettua tilallisuuden tuntua. (Kuvat 14 ja 15)



Kuva 14. Teossarja ripustettuna Imatran Taidemuseolla



Kuva 15. Töitä ripustettuna Imatran Taidemuseolla

Kiinnostukseni L-systeemiä ja muita tieteenaloja kohtaan taiteen välineenä on lisääntynyt opiskelujeni aikana ja koen edelleen kasvavaa innostusta niitä kohtaan. Ajatus tieteen, teknologian ja taiteen yhdistämisestä tuntuu oikealta suunnalta omalla taiteellisella uralla. Opinnäytetyötä varten tekemääni ohjelmaa aion käyttää jatkossakin välineenä taiteen tekemiseen, mutta haluan myös kehittää ajatusta pidemmälle ja tutkia sen potentiaalia täysin digitaalisen taiteen tekemiseen, esimerkiksi animoinnin muodossa. Laajuudestaan huolimatta koen tämän opinnäytteen olleen minulle vasta pintakosketus L-systeemin ja ohjelmoinnin käyttämiseen taiteen teossa

Lähteet

- Carlson, W. 2003. A critical history of computer graphics and animation. The Ohio State University. Viitattu 21.3.2022. Saatavissa <https://web.archive.org/web/20160703101038/http://design.osu.edu/carlson/history/lesson9.html>
- Laposky, B. 1952. Oscillon 40. Viitattu 21.3.2022. Saatavissa <http://www.vam.ac.uk/content/articles/a/computer-art-history/>
- Norman, J. "Cybernetic Serendipity": The First Widely-Attended International Exhibition of Computer Art. Viitattu 21.3.2022. Saatavissa <https://www.historyofinformation.com/detail.php?entryid=1089>
- Prusinkiewicz, P., Lindenmayer, A. 2004. The Algorithmic Beauty of Plants. Viitattu 6.3.2022. Saatavissa <https://www.semanticscholar.org/paper/The-Algorithmic-Beauty-of-Plants-Prusinkiewicz-Lindenmayer/257c08b0a31deee145a7ec85890f11ec96eefbcc>
- Reichardt, J. 1968. Cybernetic Serendipity: the computer and the arts. Viitattu 21.3.2022. Saatavissa <https://www.studiointernational.com/index.php/cybernetic-serendipity-the-computer-and-the-arts>
- Victoria and Albert Museum. 2016. A History of Computer Art. Viitattu 21.3.2022. Saatavissa <http://www.vam.ac.uk/content/articles/a/computer-art-history/>

