

Modeling Trees with a Space Colonization Algorithm

Marc Morell Higuera

Memòria del Treball Final de Grau

a la

Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

Director: Antonio Chica Calaf

Data de la defensa
26 de juny del 2014

Índex

Índex de figures	5
Índex de taules	7
Índex d'algorismes	8
0 Resum/Resumen/Abstract	9
0.1 Català	9
0.2 Castellano	9
0.3 English	10
1 Introducció	11
2 Competències tècniques	12
2.1 Disseny i implementació d'aplicacions gràfiques	12
2.2 Implementació de codi crític	12
2.3 Avaluació de la complexitat del problema i ús d'algorismes	13
3 Formulació del problema	14
3.1 Descripció del problema	14
3.2 Estat de l'art	14
3.3 Resolució del problema	16
4 Planificació	17
4.1 Compromisos inicials	17
4.1.1 Objectius del projecte	17
4.1.2 Planificació	17
4.1.3 Pressupost	19
4.2 Compromisos adquirits	19
4.2.1 Objectius del projecte	19
4.2.2 Planificació	20
4.2.3 Pressupost	20
4.3 Canvis en la planificació	21

4.4	Quantificació del treball realitzat	22
4.4.1	Quantificació temporal	22
4.4.2	Quantificació econòmica	22
4.5	Identificació de lleis i regulacions	23
5	Metodologia i rigor	26
5.1	Procediment de desenvolupament del problema	26
5.1.1	Generació de l'estructura de l'arbre	26
5.1.2	Modelització de l'arbre	27
5.1.3	Implementació de la interfície	27
5.1.4	Exportació del model	27
5.2	Validació dels resultats	27
5.2.1	Generació de l'estructura de l'arbre	28
5.2.2	Modelització de l'arbre	28
5.2.3	Implementació de la interfície	28
5.2.4	Exportació del model	29
6	Sostenibilitat i compromís social	30
6.1	Impacte social i econòmic	30
6.2	Impacte ambiental	30
7	Implementació	32
7.1	Terminologia	32
7.1.1	Arbre	32
7.1.2	Algorisme de colonització espacial	32
7.1.3	Esquelet de l'arbre	33
7.1.4	Punt d'atracció	33
7.1.5	Node de l'arbre	33
7.1.6	Distància del nou node de l'arbre	33
7.1.7	Distància d'influència	34
7.1.8	Distància de mort	34
7.1.9	Triangulació de Delaunay	34
7.1.10	Diagrama de Voronoi	34
7.1.11	Disc	36
7.2	Implementació de l'arbre	36
7.2.1	Copa de l'arbre	37
7.2.2	Estructura de l'arbre	38
7.2.3	Generació dels nous nodes i l'esquelet	38
7.2.4	Generació del model	42
7.3	Implementació de la interfície	45
7.3.1	Càmera	45
7.3.2	Visor	45
7.4	Exportació del model	46

8	Treball futur	48
8.1	Millores morfològiques	48
8.1.1	Curvatura dels segments	48
8.1.2	Estructura de les bifurcacions	48
8.1.3	Efecte de les forces externes	49
8.1.4	Ús de models naturals addicionals	49
8.1.5	Millora de la modelització de les branques i tronc	50
8.2	Millores gràfiques	50
8.2.1	Addició de fulles, flors i fruits	51
8.2.2	Ús de textures	51
8.2.3	Ús de mapejat topològic (<i>bump maps</i>)	52
9	Conclusions	54
	Bibliografia	55

Índex de figures

3.1	Estructura creada per l'algorisme de generació procedural.	15
3.2	Model de l'arbre resultant vist en el visor.	16
4.1	Diagrama de Gantt de la planificació temporal del projecte.	25
7.1	Triangulació de Delaunay bidimensional que mostra les circumferències circumscrites.	35
7.2	Diagrama de Voronoi bidimensional amb les seves llavors i les regions de cadascuna, utilitzant una distància Euclídia.	35
7.3	Relació entre la triangulació de Delaunay i el diagrama de Voronoi.	36
7.4	Representació del disc i les connexions entre discs.	37
7.5	Efecte d'alterar el nombre de punts d'atracció. A l'esquerra, una estructura formada per 1000 punts d'atracció, i a la dreta una estructura formada per 100 punts d'atracció. Ambdues copes de l'arbre són esfèriques.	41
7.6	Efecte d'alterar la distància del nou node de l'arbre. A l'esquerra, una estructura on la distància del nou node és D , i a la dreta una estructura on aquesta distància és $2D$. Ambdues copes de l'arbre són esfèriques.	41
7.7	Efecte d'alterar la distància d'influència. A l'esquerra, una estructura on la distància d'influència dels punts d'atracció és $20D$, i a la dreta una estructura on aquesta distància és $5D$. Ambdues copes de l'arbre són esfèriques.	42

7.8	Efecte d'alterar la distància de mort. A l'esquerra, una estructura on la distància de mort dels punts d'atracció és $2D$, i a la dreta una estructura on aquesta distància és $4D$. Ambdues copes de l'arbre són esfèriques.	42
7.9	Càlcul del disc d'un node de l'arbre. El segment verd mostra la direcció mitjana dels seus fills, i el segment blau mostra el vector intermedi entre el verd i el seu pare. Aquest vector blau el rotem per a determinar els vuit punts que defineixen el disc.	44
7.10	Model de l'arbre tal i com el veiem en el visor.	46
8.1	A l'esquerra, un arbre format per segments rectes, i a la dreta, un arbre amb els segments interpolats per a donar una estructura corba més realista.	49
8.2	Estructura per a les bifurcacions de les branques. Es representen les corbes que formen els seus eixos centrals i l'estructura envoltant.	49
8.3	Arbre amb un tropisme cap avall cap a les puntes de les branques.	50
8.4	Model de propagació d'ombres.	50
8.5	Simulació de la irregularitat de la base del tronc.	51
8.6	Arbre amb flors.	52
8.7	Ús d'impostors per a donar realisme a un arbre.	52
8.8	Secció d'un arbre amb diverses millores de les que hem presentat. Veiem la curvatura dels segments, l'estructura de les bifurcacions, la irregularitat de la base del tronc, l'aplicació de textures i l'ús del mapejat topològic.	53

Índex de taules

4.1	Costs de recursos humans.	22
4.2	Costs de material físic.	23
4.3	Costs de software i serveis.	23
4.4	Costs totals.	24

Índex d'algorismes

7.1	Generació dels punts d'atracció a la copa	37
7.2	Algorisme de colonització espacial	40
7.3	Algorisme de dibuix a pantalla de l'arbre.	45

Resum/Resumen/Abstract

0.1 Català

Aquest Treball Final de Grau tracta sobre la implementació d'un algorisme de generació procedural que construeixi una estructura reminiscent a la d'un arbre de clima temperat i també la implementació del pas d'aquesta estructura a un model tridimensional, acompanyat de l'eina per a visualitzar el resultat i fer-ne la seva exportació per a l'ús a aplicacions diferents.

Aquest algorisme, contrari a una bona part del treball existent en l'àmbit, no utilitzarà una definició recursiva de l'estructura de l'arbre, sinó que prendrem la suposició que el desenvolupament de la forma d'un arbre està governat per un procés auto-organitzat de competició per l'espai i la llum entre les diferents branques, amb la finalitat d'intentar emular un comportament més biològic i natural com el que trobem al nostre voltant.

El software que implementarem estarà dividit en dues parts principals. Per una part tindrem la implementació de l'algorisme que hem proposat i la modelització de l'estructura resultant, i per altra banda tindrem la implementació d'un visor d'escenes tridimensionals per a poder visualitzar el resultat que hem obtingut i exportar-lo.

0.2 Castellano

Este Trabajo Final de Grado trata sobre la implementación de un algoritmo de generación procedural que construya una estructura reminiscente a la de un árbol de clima temperado y también la implementación del paso de esta estructura a un modelo tridimensional, acompañado de la herramienta para visualizar el resultado y hacer su exportación para su uso en aplicaciones distintas.

Este algoritmo, contrario a una buena parte del trabajo existente en el ámbito, no usará una definición recursiva de la estructura del árbol, sino que tomaremos la suposición de que el desarrollo de la forma de un árbol está gobernado por un proceso auto-organizado de competición por el espacio y la luz entre las diferentes ramas, con la finalidad de intentar emular un comportamiento más biológico y natural como el que encontramos a nuestro alrededor.

El software que implementaremos estará dividido en dos partes principales. Por una parte tendremos la implementación del algoritmo que hemos propuesto y la modelización de la estructura resultante, y por otro lado tendremos la implementación de un visor de escenas tridimensionales para poder visualizar el resultado que hemos obtenido y exportarlo.

0.3 English

This Degree Final Project consists of the implementation of a procedural generation algorithm that builds a structure reminiscent to that of a temperate climate tree and also consists of the implementation of the translation from this structure to a tridimensional model, together with a tool to visualize the resulting model and export it for external use.

This algorithm, contrary to a fairly big portion of the existing work in the field, will not use a recursive definition of the tree's structure, instead we will suppose that the development of the form of the tree is governed by an auto-organized process of competition for space and light between the different branches, with the aim of trying to emulate a more biological and natural behaviour like the one we find around us.

The software that we will implement will be divided in two main parts. On one side we will have the implementation of the algorithm we proposed and the modelization of the resulting structure, and on the other side we will have the implementation of a tridimensional scene viewer in order to be able to visualize the result that we have obtained and export it.

Introducció

Aquest és un projecte dirigit per Antonio Chica Calaf i desenvolupat per Marc Morell Higuera. Aquest projecte és un Treball Final de Grau de la Facultat d'Informàtica de Barcelona, a la Universitat Politècnica de Catalunya.

La generació de contingut artístic digital de totes menes, ja sigui contingut visual o fins i tot musical, ha estat històricament a càrrec d'artistes, i amb l'ús de software normalment força específic. La generació procedural de contingut artístic digital es basa en l'ús de tècniques automatitzades (algorítmiques) per a la creació d'aquests recursos, amb una interacció humana molt variada: des de l'especificació d'un gran nombre de variables fins a cap interacció.

L'interès per a la generació procedural té diverses motivacions, des de l'interès purament acadèmic per a la comprensió del funcionament de la nostra realitat natural, fins a la reducció de costos o estalvi de recursos en la generació d'aquell contingut que vulguem obtenir, sovintment aplicat al món dels videojocs o la indústria filmogràfica. Els recursos generats proceduralment s'utilitzen com a elements de diversos graus d'importància: elements primaris com el terreny complet d'una escena, o elements secundaris com decoracions que serveixen per a contextualitzar o enriquir una escena donant només un cop d'ull. La generació procedural també té requeriments molt variats, en el cas de la producció cinematogràfica no és necessari tenir sistemes de generació en temps real, i podem permetre'ns temps relativament elevats per a la generació de cada model, fet que ens permet també obtenir un major grau de realisme. En canvi, a la indústria dels videojocs trobem casos en què aquesta generació és possible que la necessitem en un temps menor als 10 segons, o fins i tot a temps real.

Aquest Treball Final de Grau encararà un problema relacionat amb la generació procedural. Concretament, el de la generació procedural d'arbres i vegetació.

Competències tècniques

Seguidament exposem de quina forma les disciplines treballades al llarg dels estudis de grau han tingut efecte en les diferents competències tècniques que tracta aquest Treball Final de Grau.

2.1 Disseny i implementació d'aplicacions gràfiques

Els coneixements de OpenGL i Qt adquirits a les assignatures de Interacció i Disseny d'Interfícies (IDI), Gràfics (G) i Videojocs (VJ) han estat clau per al desenvolupament de la part gràfica de l'aplicació.

D'Interacció i Disseny d'Interfícies hem aprofitat sobretot els coneixements de l'ús de Qt Creator per a dissenyar i implementar la interfície, amb l'interacció de botons, menús i els demés *widgets*.

Pel que fa Gràfics, gran part de l'estructura del nostre motor gràfic està basada en el projecte que es duu a terme a aquesta assignatura, i també hem utilitzat part dels coneixements més avançats d'OpenGL que s'imparteixen per a implementar el visor, i tot i que vam considerar l'ús de *shaders* que s'imparteix a aquesta assignatura, les restriccions temporals a les que estàvem subjectes no ens han permès explorar més en profunditat la possibilitat d'afegir-los.

De Videojocs hem utilitzat de nou els coneixements d'OpenGL que vam adquirir durant les implementacions del joc 2D i del joc 3D que es duen a terme a l'assignatura. També vam explorar la possibilitat de millorar els models afegint fulles, flors i fruits pels que podríem haver aprofitat els coneixements que vam adquirir a l'assignatura. Malauradament, no ho hem pogut implementar degut a les restriccions temporals.

2.2 Implementació de codi crític

Aquesta competència va molt lligada a la gran majoria d'assignatures de programació del grau.

Des de Programació 1 (PRO1) fins Algorísmia (A) i Aplicació d'Algorísmia (AA) es té en compte la implementació de codi crític, i s'estableixen els criteris que ha de seguir un bon codi, com són el

temps d'execució, l'eficiència, l'estil i la seguretat.

Aquests criteris s'apliquen activament en tot el codi que forma el nostre projecte, utilitzant l'estil que s'imparteix durant totes les assignatures de programació del grau, però especialment a les inicials com PRO1 i PRO2.

També hem fet anàlisis dels nostres algorismes per a determinar el nombre de recorreguts que fèiem per les nostres estructures i per a evitar fer operacions o procediments innecessaris, fet que també va lligat amb la propera competència tècnica que hem treballat.

2.3 Avaluació de la complexitat del problema i ús d'algorismes

La complexitat computacional també és un factor clau de la nostra aplicació. Durant tota la fase de desenvolupament hem tractat de trobar les millors solucions mitjançant l'ús d'algorismes i d'estructures de dades que hem vist durant els estudis i també que no hem vist però hem après autònomament.

Les assignatures que més hem tingut presents per a satisfer aquesta competència són Programació 1 i 2 (PRO1 i PRO2), Estructures de Dades i Algorismes (EDA), Algorísmia i Amplicació d'Algorísmia (A i AA), i finalment Teoria de la Computació (TC).

Formulació del problema

En aquest capítol analitzarem el problema que es va plantejar inicialment i sobre el que basem aquest Treball Final de Grau, i com a data de la presentació de la memòria l'hem encarat i solucionat.

3.1 Descripció del problema

El problema que encarem en aquest projecte és el de la generació procedural de vegetació de climes temperats. Volem implementar un algorisme que generi automàticament l'estructura que representa les ramificacions que formen un arbre o arbust d'acord a una sèrie de paràmetres definits per l'usuari tal i com explicarem al capítol 7.

Més concretament, utilitzarem el mètode presentat a l'article homònim *Modeling Trees with a Space Colonization Algorithm*[1], que descriu aquest procediment sota la suposició que el desenvolupament de la forma d'un arbre està governat per un procés auto-organitzat de competició per l'espai i la llum entre les diferents branques.

Amb aquest algorisme obtindrem una estructura de punts units entre ells com la que veiem a la figura 3.1 que ens servirà com a guia per a la seva posterior modelització tridimensional, amb la finalitat d'obtenir un model que s'assembli a un arbre o arbust real.

Adicionalment a l'algorisme de generació procedural i el procediment de modelització de l'estructura, també implementarem una interfície gràfica que ens faciliti l'entrada de dades, i la verificació i validació del resultat obtingut, tal i com hem vist a la figura 3.1.

Finalment, donat que volem poder utilitzar aquest model fora del nostre propi programa, també implementarem la possibilitat d'exportar el model generat.

3.2 Estat de l'art

Seguidament exposem els estudis i investigació del problema que tractem i les fonts d'informació rellevants pel desenvolupament del projecte i el rol de cadascuna.

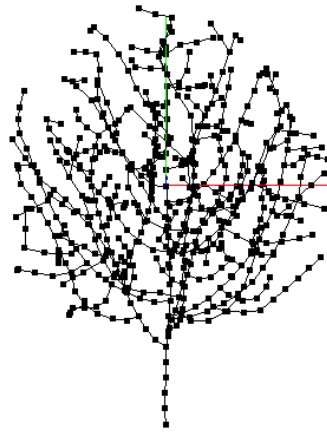


Figura 3.1: Estructura creada per l'algorisme de generació procedural.

Font: Execució del nostre programa.

La font principal pel desenvolupament del projecte, tal i com hem comentat a la secció 3.1, és l'article *Modeling Trees with a Space Colonization Algorithm*[1] que contextualitza i descriu la base teòrica que implementarem. Aquest article conté una contextualització i explicació de l'algorisme principal de generació procedural, que també anomenarem algorisme de colonització espacial, i algunes guies sobre la seva implementació i també algunes referències per a la seva traducció a model tridimensional com les que es descriuen a *Modeling the Mighty Maple*[2]. Aquest article és una extensió d'un altre article amb el títol *Modeling and visualization of leaf venation patterns*[3] que estudia l'ús d'aquests algorismes però per a la generació de venacions de fulles.

A banda d'això, hi ha un conjunt de tècniques diferents per a la generació procedural d'arbres amb diferents suposicions. Recordem que l'algorisme principal que tracta aquest projecte es basa en assumir que el desenvolupament de la forma d'un arbre està governat per un procés de competició per l'espai i la llum entre les diferents branques. No partim la construcció de l'arbre des del que seria l'estructura tronc d'una forma recursiva, sinó que partim des d'un conjunt de punts a l'espai que guien el creixement de l'arbre d'una forma que intenta emular el comportament de la pròpia naturalesa.

Altres punts de vista cap al tractament de la generació procedural de vegetació parteixen de bases com que la seva estructura és un cos del que recursivament creixen branques, com el que es va començar a estudiar a *Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body*[4] i després a implementar a *Botanical tree image generation*[5], seguit de les extensions d'altres[2][6][7][8][9] que inspiren la tècnica que s'implementa a l'article principal que seguim.

Així doncs, el nostre projecte es tracta de la implementació i anàlisi de la tècnica de generació procedural presentada a l'article *Modeling Trees with a Space Colonization Algorithm*[1], donat que a data de redacció de la memòria no hi ha cap implementació de domini públic. A més a més, pel que fa a la implementació de la modelització seguirem parts de la tècnica explicada a *Modeling*

the Mighty Maple[2] i *A quantitative analysis of plant form — the pipe model theory. I. Basic analyses*[10] per a obtenir la malla tridimensional que representarà el model final.

Finalment, a l'article *Self-organizing tree models for image synthesis*[11] trobem tècniques més avançades que pretenen guiar el creixement de l'estructura de l'arbre cap als focus de llum i evitant les ombres, que es coneix amb el nom de model de propagació d'ombra. Aquesta tècnica està explicada a fons al llibre *Fuzzy Plant Modeling with OpenGL — Novel Approaches in Simulating Phototropism and Environmental Conditions*[12]. Aquest article també proposa la gravetat com a variable a tractar en la generació procedural, entre d'altres. Totes aquestes millores es podrien aplicar al model que nosaltres presentarem en la implementació d'aquest projecte, però que s'han deixat com a treball futur degut a restriccions temporals tal i com veurem al capítol 8.

3.3 Resolució del problema

El software que hem implementat ha aconseguit solucionar satisfactòriament els problemes que hem descrit anteriorment tal i com descriurem extensament al capítol 7.

Hem implementat l'algorisme de colonització espacial, que com hem vist a la figura 3.1 genera una estructura reminiscent a la d'un arbre, també hem elaborat l'algorisme que tradueix aquesta estructura a un model tridimensional de forma satisfactòria tal i com podem veure a la figura 3.2.

Finalment, també hem fet l'exportació del model per a que es pugui utilitzar en altres entorns.

Les millores que hem aplicat han estat majoritàriament a l'execució i a la navegabilitat del visor, i com veurem al capítol 8, hem deixat altres millores com a treball futur.



Figura 3.2: Model de l'arbre resultant vist en el visor.

Font: Execució del nostre programa.

Planificació

En aquest capítol procedirem a revisar els compromisos de les fites anteriors, farem un anàlisi i justificació dels canvis, i finalment presentarem una quantificació del treball realitzat i una valoració econòmica del projecte.

4.1 Compromisos inicials

A la fita inicial vam establir els següents compromisos, que separem en objectius, compromisos de planificació i compromisos de pressupost.

4.1.1 Objectius del projecte

L'objectiu del projecte és començar i finalitzar el desenvolupament d'un programa informàtic que generi models tridimensionals d'arbres de forma automàtica, seguint l'especificació d'un cert nombre de paràmetres.

Amb *programa informàtic* volem dir el següent: la funcionalitat que acabem de descriure (ampliada anteriorment al capítol 3), la interfície gràfica que permet usar-lo i l'eina d'exportació del model.

4.1.2 Planificació

Pel que fa a la planificació, el projecte ha d'estar tancat i defensat el dia 26 de juny del 2014 a les 12:00 hores.

Hem dividit el projecte en quatre tasques: la fita inicial, el desenvolupament, la fita de seguiment i la memòria. Es preveu que la duració total del projecte sigui des del 17 de febrer del 2014 fins el 26 de juny del 2014 quan serà defensat.

Fita inicial

La fita inicial va constar de la definició de l'abast, la planificació, el pressupost i la sostenibilitat, la presentació preliminar, el context i la biografia, el plec de condicions i la presentació oral i el document final.

De l'elaboració dels 6 primers documents en vam treure l'últim, i aquest últim el vam utilitzar de nou per a l'elaboració de la fita de seguiment i de la memòria.

Aquestes tasques tenien l'objectiu d'encarrilar l'enfocament del projecte i constaven de l'elaboració de 7 documents i la preparació de dues presentacions orals. Els terminis de presentació de cada document i de les presentacions orals ja estaven sòlidament determinats i per tant no hi va haver endarreriments notables. Aquesta fase va acabar el 14 de març del 2014.

Els recursos humans que van involucrar aquesta fase van ser el propi desenvolupador del projecte com a autor dels documents i les presentacions, i el director del projecte com a autoritat de consulta de les especificacions del propi projecte.

Els recursos materials que es van utilitzar durant aquesta fase van ser l'ordinador de treball per a generar els documents i una càmera digital per a gravar la primera presentació oral. El software i serveis emprats van ser [writeL^AT_EX](#) i Microsoft Project 2013.

Desenvolupament

A grans trets, el desenvolupament del projecte s'ha dividit en quatre grans tasques: el desenvolupament de l'algorisme de generació procedural, la generació del model tridimensional a partir dels resultats obtinguts, el desenvolupament de les interfícies i el desenvolupament de l'exportació.

La descripció a fons d'aquestes tasques la podem trobar al capítol 7.

Aquestes quatre subfases han compartit els mateixos recursos, que enumerem seguidament.

Els recursos humans han estat el propi desenvolupador del projecte com a implementador del software i el director del projecte com a autoritat de consulta.

Els recursos materials han estat el propi ordinador de desenvolupament. En aquesta fase hem utilitzat Microsoft Visual Studio 2013 i Qt Creator com a entorns de programació.

Aquesta fase s'ha prolongat fins dues setmanes abans de l'entrega de la memòria.

Fita de seguiment

La fita de seguiment va constar d'un document que es va presentar al director del projecte i contenia les especificacions necessàries per a dur a terme el projecte, de forma similar a la fita inicial, i el progrés fins al moment en el desenvolupament del projecte.

Els recursos humans que van participar en aquesta fase van ser únicament el propi desenvolupador com a autor del document.

Els recursos materials van l'ordinador personal per a generar el propi document. Detalladament, el software que es va utilitzar va ser [writeL^AT_EX](#) i Microsoft Project 2013.

La fita de seguiment es va entregar el 19 de maig del 2014.

Memòria

Aquesta memòria s'ha realitzat i refinat en paral·lel al desenvolupament, documentant breument les decisions després de cada tasca de la implementació, i finalment expandint aquestes decisions i explicacions un cop hem satisfet els objectius del programari que hem implementat.

La memòria ha estat redactada pel desenvolupador del projecte, i el software que s'ha utilitzat ha estat el programa que s'implementa en aquest projecte, [ShareL^AT_EX](#) i Microsoft Project 2013, acompanyat de Adobe Photoshop CS6 per a fer algunes de les figures que trobem en aquesta memòria.

La defensa de la memòria és el dia 26 de juny de 2014.

4.1.3 Pressupost

A la fita inicial vam dividir els costs del projecte en dos tipus, costs d'activitat i costs d'estructura.

Dins dels costs d'activitat no trobem costs variables, ja que la nostra producció consta d'un únic software que no es planeja distribuir físicament, i si es decideix fer-ne la distribució, nosaltres no serem els encarregats de fer-la a petita, mitjana o gran escala.

Per altra banda, el costs fixes són la part amb més pes del pressupost del projecte. Hi trobem els salaris (o més aviat una estimació donada la naturalesa docent del projecte), l'adquisició del material de treball com l'ordinador, els seus perifèrics i les llicències software i de serveis web que utilitzem, i el manteniment d'aquests.

Pel que fa als costs d'estructura, per la part de costs indirectes haurem de tenir en compte la despesa en llum, la mobilitat per a assistir a les reunions de seguiment i el salari del director. Com ja hem dit, donat que aquest projecte consta només d'una unitat a produir, els costs directes de mà d'obra seran els costs totals de mà d'obra.

A la secció [4.4](#) trobarem les quantificacions que hem estimat pel nostre projecte.

4.2 Compromisos adquirits

Seguidament analitzarem de quina manera hem satisfet els compromisos que hem descrit a la secció [4.1](#).

4.2.1 Objectius del projecte

Al capítol anterior hem descrit com hem satisfet els compromisos dels objectius i l'abast del projecte.

Hem implementat satisfactòriament els mòduls del programari que havíem establert inicialment com a objectius.

4.2.2 Planificació

Fita inicial

Els compromisos de la fita inicial es van satisfer mitjançant el seguiment de la planificació proposada a l'assignatura de Gestió de Projectes (GEP), i vam utilitzar la retroalimentació rebuda per a aclarar i millorar la qualitat dels documents posteriors.

Tal i com hem comentat, les entregues van consistir de 7 documents que tractaven les diferents consideracions a tenir en compte per al desenvolupament del projecte, que ens han estat de gran ajuda.

Desenvolupament

El desenvolupament s'ha fet seguint les pautes establertes a la planificació de la fita inicial i amb els canvis que es van explicar a la fita de seguiment. A la secció 4.3 veurem amb més profunditat els canvis que hem decidit fer i la seva justificació.

La divisió en quatre parts del desenvolupament, tal i com hem explicat a la secció 4.1.2, ens ha ajudat a poder fer una validació per separat de cadascun dels components que integren el nostre programa final.

Fita de seguiment

La fita de seguiment es va elaborar d'acord amb les pautes que estaven presents al Racó de la Facultat, i seguint l'etapa de desenvolupament en la que es trobava el projecte. D'aquesta fita en va sorgir una retroalimentació per part del director del projecte que hem utilitzat per a l'elaboració d'aquesta memòria i del desenvolupament del projecte posterior a l'entrega de la fita de seguiment.

Memòria

La memòria s'ha redactat durant les etapes finals del projecte, seguint les pautes que podem trobar al Racó de la Facultat, i utilitzant L^AT_EX, que és l'estàndard de les publicacions científiques.

Durant el desenvolupament del projecte hem pres apunts d'aquelles decisions que volíem destacar durant la redacció a fons de la memòria. També hem utilitzat la fita inicial i la fita de seguiment amb les seves retroalimentacions com a pautes per a la redacció de gran part de les seccions que trobem en aquest document, intentant cobrir i satisfer els continguts de les rúbriques d'avaluació, i amb la informació que també hem trobat rellevant.

4.2.3 Pressupost

Inicialment, a la fita inicial, es va proposar una estimació del pressupost que tenia en compte les hores de treball, el cost del material, i les diferents llicències de programaris i serveis que

utilitzaríem durant el desenvolupament d'aquest projecte.

Degut a la nostra experiència anterior amb el material i el programari, ens hem pogut aferrar fortament a les estimacions inicials. A la secció 4.4 podem trobar la quantificació econòmica resultant de l'elaboració d'aquest projecte, que recordem, ha estat un projecte de caire docent i per tant l'anàlisi dels costos és més aviat aproximat.

4.3 Canvis en la planificació

A la fita inicial vam proposar un pla d'acció i una distribució del projecte flexible, que ens permetia, si ho vèiem necessari, fer canvis en l'ordre de la implementació dels diferents mòduls que s'incorporen en el nostre programa.

Durant el desenvolupament ens hem donat compte que algunes tasques podien ser implementades en desordre per tal de facilitar la validació de les demés, i per a establir un flux de treball que hem trobat més natural. Addicionalment també hem trobat la necessitat de dedicar més temps del previst a algunes tasques degut a la seva complexitat d'implementació i l'existència de tècniques més eficients que hem volgut explorar. Seguidament les exposem amb més detall.

Primerament, hem eliminat la versió textual de l'algorisme de colonització, principalment per dos motius: El primer és que no vam trobar un format adequat per a representar l'estructura de forma visible i fàcilment analitzable, i el segon motiu és que les estructures que utilitzem i el sistema gràfic que hem desenvolupat inicialment ja són molt compatibles i integrables, i per tant podíem passar a una versió gràfica de l'algorisme de colonització immediatament. Els costos monetaris derivats d'aquests canvis són nuls, i el cost temporal de les seccions afectades queda inalterat.

Addicionalment, donat que l'algorisme de colonització requereix l'ús intensiu de tests de distància entre punts, hem volgut utilitzar una tècnica recomanada a *Modeling Trees with a Space Colonization Algorithm*[1] per a optimitzar aquest procés. Es tracta de l'ús d'una triangulació de Delaunay i el seu corresponent diagrama de Voronoi per a tenir una estructura de dades que faciliti i disminueixi el cost computacional dels tests de distància. Per a fer això utilitzem la llibreria *Computational Geometry Algorithms Library (CGAL)*. Explicarem aquestes estructures al capítol 7.

L'ús d'aquesta llibreria i aquestes tècniques han suposat un temps de dedicació que no havíem previst, ja que no teníem coneixements previs de les estructures i les tècniques, ni tampoc havíem utilitzat mai la llibreria. El cost monetari queda de nou inalterat, però el cost temporal incrementa en aproximadament 16 hores útils, derivades de l'estudi dels conceptes nous que implementem i de la instal·lació, estudi i familiarització amb la llibreria. Estem molt satisfest de la decisió d'haver fet servir aquesta llibreria, ja que s'ocupa de dur a terme una part de la implementació que podria haver estat molt complicada d'implementar, i que ens podria haver suposat un temps addicional de feina imprevist força elevat.

4.4 Quantificació del treball realitzat

Seguidament presentem les dues quantificacions del treball realitzat que hem fet. Per una banda presentarem una quantificació temporal amb el temps que hem dedicat a l'elaboració d'aquest projecte i per altra banda exposarem la quantificació econòmica dels costos derivats de la realització del treball.

4.4.1 Quantificació temporal

A la figura 4.1 trobem el diagrama de Gantt que conté el desenvolupament del projecte. Podem veure com l'elaboració del projecte s'ha extès durant tota la durada del quadrimestre, i també veiem la importància dels algorismes que hem implementat en el temps total dedicat al projecte. El temps addicional per a familiaritzar-nos amb la llibreria CGAL ha estat afegit al temps de la implementació de l'algorisme de colonització espacial, que ha vist el seu temps incrementat respecte el Gantt que vam presentar originalment.

En total, valorem el temps de treball en el projecte amb aproximadament 400 hores.

4.4.2 Quantificació econòmica

Seguidament explorarem la quantificació econòmica dels costos associats al projecte.

Cal tenir en compte que en el nostre cas concret disposem de llicències software facilitades per la institució docent donada la naturalesa acadèmica del projecte, però els detallarem de totes maneres.

A la taula 4.1 veiem els costos estimats dels recursos humans. Donat que el nostre projecte és acadèmic, les dades les hem extretes d'altres projectes d'una complexitat similar al nostre. Els costos de mobilitat i impostos estan inclosos al salari.

Càrrec	Hores	Sou/hora	Cost total
Director del projecte	100	50	5000€
Desenvolupador del projecte	400	30	12000€
Total			17000€

Taula 4.1: Costos de recursos humans.

A la taula 4.2 trobem els costos de material físic. En el nostre cas només hem utilitzat un ordinador de sobretaula i també un portàtil per tal de tenir dues plataformes diferents on provar que el nostre producte funcioni correctament. Mostrarem el cost amortitzat que correspon al desenvolupament d'aquest projecte, ja que la vida útil del hardware s'estén més enllà d'aquest projecte.

Finalment, a la taula 4.3 trobem els costos del software i els serveis que utilitzarem per al desenvolupament del projecte. La majoria de llicències s'ofereixen amb un cost anual en comptes d'un cost vitalici, però degut a la durada del projecte només inclourem un cost, que serà el d'una llicència d'un any o de quatre mesos quan sigui aplicable. Els sistemes operatius d'ambdós ordinadors no s'inclouen donat que ja vénen inclosos al preu de la compra dels ordinadors.

Material	Cost amortitzat
Ordinador Intel Core i7	180€
Sony VAIO Series Z	100€
Perifèrics (ratolí, teclat, etc.)	30€
Total	310€

Taula 4.2: Costs de material físic.

A fi justificatiu exposem la finalitat de cada servei i software que incloïm a la taula:

- Microsoft Office 365 Home Premium: Ens proporciona accés a utilitats d'oficina com un entorn d'edició de documents professional, fulls de càlcul que podem utilitzar per a expressar gràficament resultats i explorar millores i una eina per a generar les diapositives de la presentació del projecte, entre d'altres.
- Project Pro for Office 365: És el gestor del projecte que estem utilitzant. Ens mostra la planificació, ens permet generar el diagrama de Gantt i calcular camins crítics.
- Visual Studio Professional with MSDN: L'entorn de programació que hem utilitzat. La versió professional ens proporciona l'habilitat de mantenir únicament un projecte en tots els ordinadors, i millor suport a diverses plataformes.
- Qt Creator: L'entorn de disseny de la interfície que hem utilitzat. Aquest software facilita el disseny d'interfícies Qt, que són les que s'han fet servir en aquest projecte.
- Llicència Micro de GitHub: Ens proporciona l'habilitat de tenir repositoris privats per tal que el nostre codi no sigui d'accés públic. També ens proporciona un sistema de control de canvis molt útil i ens assegura que no perdrem cap informació.

Software o servei	Cost total
Microsoft Office 365 Home Premium	40€
Project Pro for Office 365	100€
Visual Studio Professional with MSDN	1283.65€
Qt Creator	0€
Llicència Micro de GitHub	20.42€
Total	1444.07€

Taula 4.3: Costs de software i serveis.

Seguidament presentem a la taula 4.4 un resum de l'estimació de costs incloent els costs de factures de manteniment:

4.5 Identificació de lleis i regulacions

Finalment, també volem fer referència a la consideració de la diferent normativa que afecta el nostre projecte.

Tipus de recurs	Cost total
Recursos humans	17000€
Hardware	310€
Software	1444.07€
Costs passius	300€
Total	19054.07€

Taula 4.4: Costs totals.

Donada la naturalesa no lucrativa d'aquest projecte, ens trobem amb les següents consideracions respecte lleis i regulacions. Noteu que podeu consultar les llicències accedint als vincles que ressalten cada recurs que utilitzem en els següents paràgrafs.

Pel que fa les eines de programació, tenim permès l'ús de [Qt Creator](#) i [Microsoft Visual Studio 2013](#).

Si ens fixem ara en les llibreries que utilitzem, l'ús d'[OpenGL](#) i [CGAL](#) estan permesos en el nostre context.

Només ens resten els recursos o *assets* que utilitzem. Per a la interfície utilitzem els mateixos *widgets* de Qt complementats amb els nostres propis quan cal, i amb el tema visual de Windows, i podem utilitzar tots dintre del marc legal en què ens trobem.

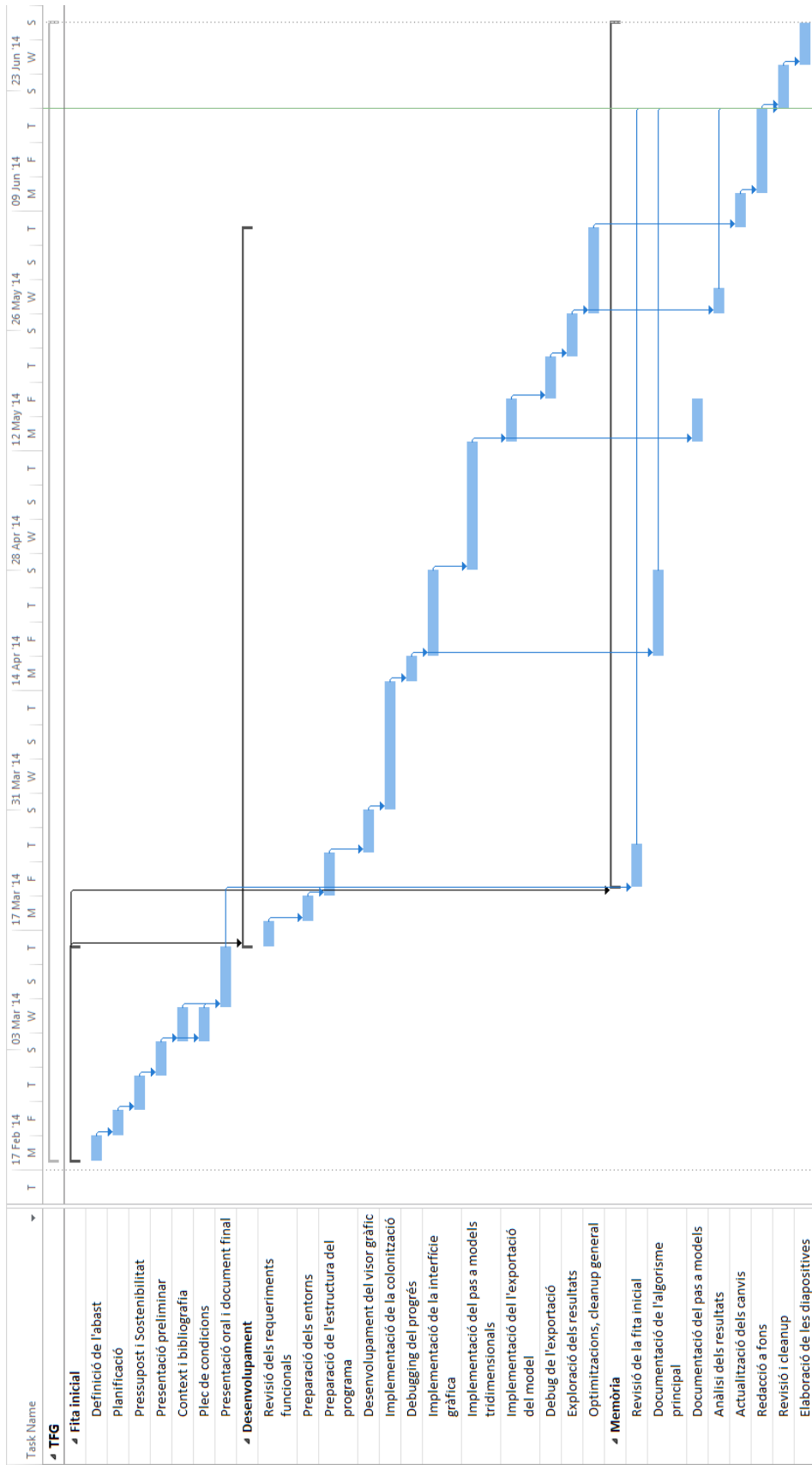


Figura 4.1: Diagrama de Gantt de la planificació temporal del projecte.

Metodologia i rigor

Seguidament exposarem els procediments que hem seguit per a arribar a la solució proposada i els mètodes d'avaluació i validació dels resultats obtinguts.

5.1 Procediment de desenvolupament del problema

Les primeres passes cap a la resolució del problema es van trobar en les reunions inicials amb el director del projecte, en les que es va estudiar el problema i es va fer un primer anàlisi del treball a realitzar.

Seguidament, amb l'ajuda de l'assignatura de Gestió de Projectes vam determinar quins factors havíem d'encarar i ens va guiar cap a l'especificació de la definició del problema i la seva resolució.

Per a analitzar el procediment per a arribar a la solució que hem proposat al capítol 3 cal que mirem els objectius del projecte. Tal i com hem esmentat diverses vegades durant la redacció d'aquesta memòria, el nostre objectiu és generar proceduralment un model d'un arbre o arbust de clima temperat utilitzant l'algorisme que es descriu a *Modeling Trees with a Space Colonization Algorithm*[1], que posteriorment podrem exportar. Amb aquest objectiu vam identificar una divisió de la feina en quatre tasques a implementar que hem descrit al capítol 7, que van encarades cap a la resolució del problema i l'assoliment dels objectius, i les descrivim a continuació.

5.1.1 Generació de l'estructura de l'arbre

A *Modeling Trees with a Space Colonization Algorithm*[1] es descriu l'algorisme de generació procedural amb el que generarem un conjunt de punts units entre ells per a formar una estructura semblant a la d'un arbre, tal i com vèiem a la figura 3.1.

Aquesta tasca no acaba de complir cap dels objectius que havíem establert, únicament ens proporciona l'estructura que hem anomenat esquelet, que haurem de tractar per a generar el model de l'arbre, però és un prerequisite per a poder arribar als objectius establerts.

De totes maneres, aquesta és la tasca que hem considerat principal en el nostre projecte, ja que és la

que la caracteritza i la diferencia d'altres projectes de generació procedural de vegetació, tal i com hem descrit al capítol 3.

També és una tasca important perquè és la que defineix l'estructura que modelarem, i per tant és el principal agent que afecta la morfologia de l'arbre.

5.1.2 Modelització de l'arbre

Aquesta tasca tracta el pas de l'estructura que hem generat anteriorment a una malla tridimensional que representarà l'arbre, tal i com hem vist a la figura 3.2.

Amb aquesta tasca haurem satisfet l'objectiu de la creació del model tridimensional de l'arbre.

5.1.3 Implementació de la interfície

Aquesta tasca no es deriva explícitament de la generació procedural de l'arbre, però ens permet tenir una eina per a visualitzar i validar el progrés i resultats als que arribem durant el mateix desenvolupament de la resta de tasques sense haver de dependre en eines externes a les que hauríem d'adaptar el nostre projecte.

D'aquesta manera, construïm un entorn propi que actua com un únic agent, i amb la flexibilitat de poder-lo adaptar a les nostres necessitats.

Evidentment, això afegeix un temps de dedicació addicional a la planificació temporal del projecte, però com hem pogut observar al diagrama de Gantt de la figura 4.1, l'hem pogut adaptar i definir com a objectiu del nostre projecte.

Amb aquesta tasca haurem satisfet l'objectiu de la implementació del visor del model tridimensional.

5.1.4 Exportació del model

L'exportació del model tampoc és un objectiu derivat directament de la generació del model procedural, però vam trobar convenient afegir-la com a objectiu del projecte per a poder donar un ús extern als models que generem.

Amb aquesta tasca haurem satisfet l'últim objectiu del projecte, l'exportació del model generat.

5.2 Validació dels resultats

Degut a que el nostre projecte té quatre tasques que el defineixen, tal i com hem vist a la secció anterior, la validació també s'ha dut a terme independentment per a cadascuna de les tasques.

També, ja que aquest projecte està estretament relacionat amb l'observació natural, veurem que el procés de validació està fortament lligat al de la nostra pròpia experiència i utilitzarà un anàlisi majoritàriament qualitatiu dels resultats que pot ser interpretat de formes molt diferents segons qui l'avalui.

A continuació descrivim el procediment per a cadascuna.

5.2.1 Generació de l'estructura de l'arbre

Per a validar el procés de la generació de l'estructura de l'arbre hem utilitzat el visor per a visualitzar les estructures resultants de l'execució de l'algorisme i fer-ne un anàlisi qualitatiu.

Aquest anàlisi qualitatiu consisteix en identificar que la morfologia de l'esquelet resultant comparteix una similitud amb el d'un arbre. D'aquesta manera hem volgut identificar aquells trets que defineixen una estructura d'arbre, com per exemple la presència d'un punt inferior que s'extén en un sentit més o menys vertical en forma troncal, i que d'aquest creixen diferents segments en sentits que intenten ocupar l'espai establert amb un creixement que tendeix a ser vertical, i que d'aquests també creixen segments que en general cada cop són més curts i que també intenten ocupar l'espai lliure.

Amb aquest procediment haurem validat la generació de l'estructura de l'arbre.

5.2.2 Modelització de l'arbre

La validació de la modelització de l'arbre segueix un procediment semblant a la validació de la generació de l'estructura de l'arbre. En aquest cas utilitzarem el visor de nou per a visualitzar el model tridimensional obtingut de la realització d'aquesta tasca.

El procés de validació consistirà en un anàlisi qualitatiu del model obtingut, en el que buscarem que s'assembli a un arbre natural, igual que en la tasca anterior, però amb trets addicionals. Així doncs, voldrem identificar els trets que defineixen el que coneixem com a arbre, com l'estructura troncal i les branques similarment a la tasca anterior, però a més a més també mirarem que aquesta estructura troncal sigui la més ampla i que a mida que l'estructura es va bifurcant en branques, aquestes siguin més estretes, com les d'un arbre natural. També mirarem que les branques tinguin una forma circular, i que el punt d'embranchament ens recordi a un embranchament natural.

Amb aquest procediment haurem valorat la modelització de l'arbre.

5.2.3 Implementació de la interfície

La validació de la implementació de la interfície s'ha dut a terme incrementalment, comprovant que les funcionalitats que afegíem tenien el comportament esperat.

Per a fer la validació del visor hem de prendre la suposició que el model és correcte, ja que el mètode per a validar-lo tracta en comprovar si veiem el model tal i com esperàvem. Evidentment, és molt poc probable que veiem un model correctament que no ho és, i més quan a cada execució el model canvia.

Una altra solució per a dur a terme la validació del visor és exportar el model per a visualitzar-lo en una aplicació externa i comprovar si obtenim els mateixos resultats. En cas que no sigui així,

normalment suposarem que aquesta aplicació externa està correctament implementada i el visor no passarà la prova de validació.

5.2.4 Exportació del model

Finalment, la validació de l'exportació del model es farà de forma similar a la del visor, ja que hem vist que una tasca pot validar l'altra.

D'aquesta manera, per a validar l'exportació del model farem dues coses. La primera serà comprovar que l'arxiu que s'exporta estigui correctament escrit en text pla, i amb un format com el que esperem i que veurem al capítol 7, indicant una llista de vèrtexs, cares i normals. El segon pas d'aquesta validació serà carregar el model amb un software extern que suposem que funciona correctament i observar si els resultats de la visualització en aquest software són els mateixos que en el nostre.

Sostenibilitat i compromís social

En aquest capítol analitzem de quina manera el nostre projecte té un impacte social, ambiental i econòmic, i de quina manera això ha influenciat el desenvolupament i la presa de decisions en la planificació i elaboració del treball.

6.1 Impacte social i econòmic

El producte final que hem realitzat és una utilitat enfocada al públic científic, docent, i desenvolupador d'entorns o escenes tridimensionals. L'impacte que, una versió comercial del nostre projecte pretendria tenir, seria la reducció de despeses dels usuaris, una rebaixa considerable del temps de desenvolupament dels entorns amb vegetació, i en general una millora en el rendiment laboral.

Però donada la naturalesa no lucrativa del projecte, l'impacte social quedarà reservat a una naturalesa docent de l'implementador del projecte i de futurs estudis a l'àmbit de la generació procedural de vegetació.

6.2 Impacte ambiental

Pel que fa a l'impacte ambiental, un projecte de software té un impacte ambiental derivat dels costos de producció, com els de la fabricació dels ordinadors o l'obtenció de l'energia elèctrica per fer-los funcionar. Aquest projecte però, no té un impacte ambiental comparable amb el de projectes d'àrees més directament involucrades amb els trets que hem exposat, ja que es tracta d'un projecte a petita escala, on els factors que afecten el mediambient normalment estan fora de la nostra possibilitat de presa de decisions.

Així doncs, no explorarem alternatives possiblement més sostenibles amb les que no estem familiaritzats i que potencialment introduirien riscos amb els que hauríem de tractar, com bé seria fer un anàlisi de les instal·lacions en les que treballem, i del rendiment energètic del hardware que utilitzem, que se surten de l'abast d'aquest projecte.

De totes maneres, i de caràcter més aviat informal, els productes hardware que hem utilitzat

compleixen l'estàndard [Energy Star](#) d'eficiència energètica, i les commutacions que hem fet degut a l'elaboració d'aquest projecte sempre han estat en transport públic.

Implementació

Les següents seccions expliquen totes les particularitats i especificacions de la implementació del nostre projecte.

Primer de tot definirem els termes que creiem convenients i rellevants per a les explicacions de la implementació, i després explicarem la implementació de la generació de l'arbre, amb tots els passos que involucra, i finalment explicarem la implementació del visor que utilitzem per a poder veure els resultats de la generació i l'eina d'exportació.

7.1 Terminologia

Durant l'explicació de la implementació dels diferents mòduls, funcionalitats i algorismes del nostre programa utilitzarem uns termes concrets per a referir-nos als diferents components rellevants de les seves corresponents seccions.

A continuació presentem una llista amb els termes que hem considerat concrets del nostre projecte i una breu explicació per a cadascun. És possible que aquestes explicacions no deixin clara la finalitat de cada terme, però tots quedaran contextualitzats mitjançant els seus usos en les seves corresponents explicacions de les properes seccions.

7.1.1 Arbre

L'arbre és l'estructura i el corresponent model tridimensional que pretenem generar amb el nostre programa.

7.1.2 Algorisme de colonització espacial

Aquest és l'algorisme principal sobre el que basem aquest projecte i que guia el creixement de l'arbre. En termes generals, l'algorisme consisteix en, partint d'un node inicial, utilitzar un conjunt de punts per a determinar la posició d'un nou node, i seguir iterant per a determinar nous nodes fins que arribem a un punt que determina el final de l'execució.

A les properes seccions especificarem totes les variables i decisions que són rellevants per a l'execució d'aquest algorisme.

7.1.3 Esquelet de l'arbre

El nostre arbre està format per un conjunt de punts i segments que pretenen simular els eixos pels quals creix l'arbre.

L'esquelet, per tant, és la guia principal que utilitzarem per a generar el model tridimensional.

7.1.4 Punt d'atracció

Un punt d'atracció és un punt a l'espai que potencialment afectarà el creixement de l'arbre durant l'execució de l'algorisme de colonització espacial.

Els punts d'atracció no s'han de confondre amb potencials punts que formaran part de l'estructura de l'arbre, ja que com hem dit, la seva finalitat és afectar el creixement de l'arbre guiant-lo cap a la direcció on es troben aquests punts.

Tal i com veurem més endavant, aquests punts es crearan a l'inici i s'aniran eliminant.

7.1.5 Node de l'arbre

Un node de l'arbre és un punt a l'espai que forma part de l'estructura de l'arbre que generem.

Els nodes de l'arbre estaran connectats entre ells, i formaran l'estructura que anomenem esquelet de l'arbre.

Inicialment tindrem un sol node que anomenem arrel, aquest node té la particularitat que és l'únic node de l'estructura que no té un predecessor, tota la resta de nodes tindran un únic predecessor.

Adicionalment, els nodes de l'arbre podran tenir cap, un, o diversos fills. Aquells nodes que no tinguin fills els anomenarem finals de branques. També és important destacar que la nostra estructura no contindrà cicles, i aprofitarem aquesta propietat per a fer diverses implementacions recursives dels diferents algorismes.

7.1.6 Distància del nou node de l'arbre

Aquest terme es refereix a la distància entre dos nodes de l'arbre.

Durant l'execució de l'algorisme de colonització espacial determinarem una direcció cap on es trobarà un nou node de l'arbre, aquest terme fa referència a quina distància el posarem del seu predecessor.

La distància del nou node de l'arbre té una importància força notable en la qualitat del model que generem. Amb una distància del nou node menor tindrem una estructura que a simple vista

contindrà menys línies rectes, però amb l'inconvenient que complicarà més la geometria d'aquest, que es tradueix en un increment del temps d'execució i de la mida del model.

7.1.7 Distància d'influència

La distància d'influència fa referència a la distància màxima entre un punt d'atracció i un node de l'arbre que tingui un efecte en el creixement de l'arbre.

És a dir, si un punt d'atracció està a una distància major que la distància d'influència d'un node de l'arbre, no afectarà en el creixement d'aquest.

7.1.8 Distància de mort

Per tal de frenar el creixement de l'arbre utilitzem la distància de mort, que és la distància entre un punt d'atracció i un node de l'arbre a partir de la qual eliminem aquest punt d'atracció del conjunt de punts d'atracció per a totes les properes iteracions de l'algorisme de colonització espacial.

Com és evident, si el distància de mort és major a la distància d'influència, l'arbre no creixerà, així que cal anar amb compte.

7.1.9 Triangulació de Delaunay

La triangulació de Delaunay és l'estructura que utilitzem per a assignar els punts d'atracció als nodes de l'arbre que afecten durant les iteracions de l'algorisme de colonització espacial.

En el nostre cas, una triangulació és una subdivisió de l'espai en tetràedres.

La particularitat de la triangulació de Delaunay és que no trobem cap vèrtex a l'interior de les circumferències circumscrites de cada tripleta de punts connectats entre si. Recordem que una circumferència circumscrita d'un polígon és una circumferència que passa per tots els seus vèrtexs.

La figura 7.1 mostra una triangulació de Delaunay en un pla. Només cal estendre aquesta representació a un volum per a obtenir l'estructura amb la que treballem nosaltres.

7.1.10 Diagrama de Voronoi

El diagrama de Voronoi és una altra forma de dividir l'espai, molt estretament relacionada a la triangulació de Delaunay.

Aquest diagrama treballa sobre un conjunt de punts, que anomenarem llavors per claredat, i defineix una subdivisió de l'espai per a cada llavor, amb la particularitat que tots els punts de l'espai d'aquesta subdivisió són els que són més propers a la llavor a la que s'assigna la subdivisió.

A la figura 7.2 veiem clarament què vol dir això, utilitzant un exemple bidimensional, que de nou, és fàcil generalitzar per a una representació tridimensional com la que utilitzarem en el nostre programa.

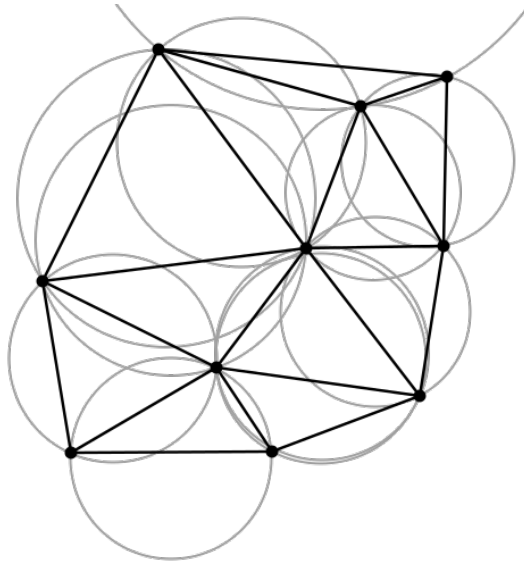


Figura 7.1: Triangulació de Delaunay bidimensional que mostra les circumferències circumscrites.

Font: [Wikipedia. Delaunay Triangulation.](#)

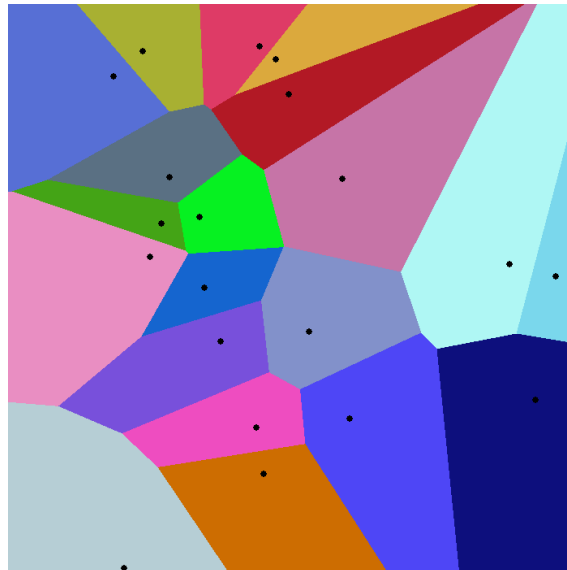


Figura 7.2: Diagrama de Voronoi bidimensional amb les seves llavors i les regions de cadascuna, utilitzant una distància Euclídia.

Font: [Wikipedia. Voronoi Diagram](#)

Tal i com hem dit, el diagrama de Voronoi i la triangulació de Delaunay van estretament lligades. De fet, matemàticament parlant, són duals. La figura 7.3 mostra la relació entre ambdues estructures. Tal i com podem observar, connectar els centres de les circumferències circumscrites dels triangles que comparteixen dos vèrtexs de la triangulació de Delaunay forma el diagrama de Voronoi dels vèrtexs que defineixen la triangulació de Delaunay.

Aquestes dues estructures són una part clau del nostre projecte, ja que voldrem construir dinàmicament una triangulació de Delaunay utilitzant els nodes de l'arbre, que utilitzarem per a generar el corresponent diagrama de Voronoi i consultar les sub-regions d'aquest diagrama per al desenvolupament

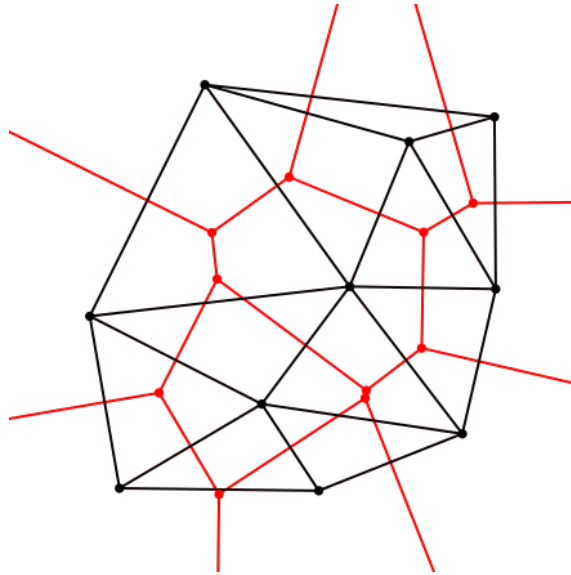


Figura 7.3: Relació entre la triangulació de Delaunay i el diagrama de Voronoi.

Font: [Wikipedia. Delaunay Triangulation](#)

lupament de l'algorisme de colonització espacial. A les properes seccions veurem amb detall de quina manera ho farem, però és molt important ressaltar la importància d'aquestes dues estructures, que donada la seva dualitat, estarà representada com una única.

7.1.11 Disc

Anomenarem disc a la secció que envolta un node de l'arbre en el model tridimensional. En un arbre tindrem tants discs com nodes de l'arbre hi hagi, un per a cadascun.

El disc l'utilitzarem per a fer les unions entre nodes en el model tridimensional, per tal de poder visualitzar una estructura contínua.

Un disc estarà definit per un nombre de vectors i una secció s . Aquests vectors s'originen al centre del disc i defineixen el sentit cap a on trobem un vèrtex del model tridimensional a distància s . En el nostre model, actualment utilitzem vuit vectors, orientats en el sentit dels eixos \vec{x} i \vec{z} positius i negatius amb una \vec{y} que depèn de la direcció de la branca, i els altres quatre vectors són els intermedis entre aquests quatre inicials.

A la figura 7.4 s'il·lustra el que volem representar. El diagrama de baix a l'esquerra representa el disc, amb els dos vectors que generen el pla sobre el que es troba, i el vector amb la seva normal.

7.2 Implementació de l'arbre

A continuació exposem el procediment de la implementació de la generació de l'arbre. Aquesta generació l'hem dividida en quatre subseccions, la implementació de la copa de l'arbre, l'especificació de l'estructura de l'arbre, la generació dels nous nodes i de l'esquelet, i finalment la generació del model tridimensional.

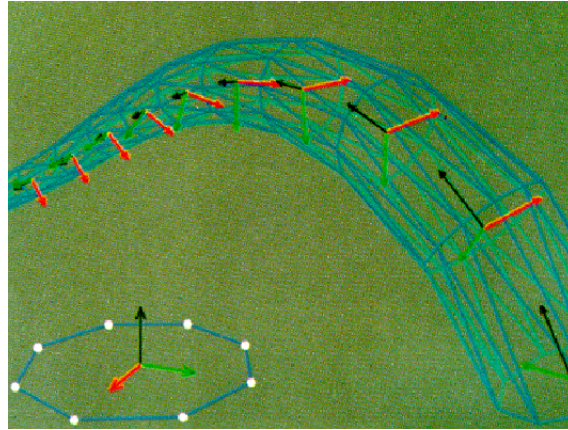


Figura 7.4: Representació del disc i les connexions entre discs.

Font: *Modeling the Mighty Maple*[2].

Cada subsecció contindrà l'explicació de les decisions preses, l'efecte d'aquestes, i les alternatives que hem considerat sempre que sigui aplicable.

7.2.1 Copa de l'arbre

La copa de l'arbre és l'estructura que dona forma general de l'arbre. A la copa de l'arbre trobarem tots els punts d'atracció que conduiran el creixement de l'execució de l'algorisme de colonització espacial.

La copa de l'arbre es defineix amb una caixa envolupant i una funció que ens retorna si un punt donat està a l'interior de la copa de l'arbre o no. Amb aquests dos components podem definir tota mena de formes de la copa de l'arbre.

Inicialment, prenent un nombre N determinat per l'usuari, generem N punts d'atracció dins de la copa de l'arbre utilitzant l'algorisme 7.1.

```

Data: nombre  $N$ , i la copa de l'arbre copa
Result: copa conté  $N$  punts d'atracció
for  $i \leftarrow 1$  to  $N$  do
    generem un punt  $P(x, y, z)$  aleatori dins la caixa envolupant de copa;
    if  $P$  està a l'interior copa then
        afegim  $P$  als punts d'atracció de copa;
         $i++$ ;
    end
end

```

Algorisme 7.1: Generació dels punts d'atracció a la copa

L'algorisme 7.1 té una particularitat que ens agradaria destacar, que és la distribució dels nombres aleatoris que utilitzem per a generar els punts P . Com bé sabem, no totes les distribucions aleatòries són uniformes, i podem generar punts, per exemple, que tendixin cap als extrems de la meitat superior. Una distribució com la que acabem de definir, tindria un efecte de poca ramificació de

l'arbre fins que el creixement arribi als extrems, ja que el conjunt de punts d'atracció que afectarien el creixement d'aquest sempre seria el mateix fins a les parts finals de l'execució del creixement.

Per a entendre completament aquest efecte cal conèixer l'algorisme de colonització espacial, però exposem la idea ara com a part rellevant que afectarà la forma del model final.

El nostre programa en particular utilitza una distribució uniforme dels punts, i queda com a treball futur la possibilitat d'afegir altres distribucions aleatòries.

7.2.2 Estructura de l'arbre

L'arbre està internament definit per la seva copa, que conté els punts d'atracció, i una llista de nodes de l'arbre, que inicialment conté l'únic node arrel a una distància D_{root} per sota de la copa.

Aquesta estructura també contindrà els valors que especifiquen les diferents propietats que condicionaran el creixement de l'arbre. Aquests són la distància del nou node de l'arbre (7.1.6), la distància d'influència dels punts d'atracció (7.1.7) i la distància de mort (7.1.8).

Adicionalment, també contindrà l'estructura que forma la triangulació de Delaunay (7.1.9 i 7.1.10), que inicialment només contindrà el node arrel de l'arbre, i que anirem construint paral·lelament a la generació, tal i com veurem a la següent secció.

7.2.3 Generació dels nous nodes i l'esquelet

Seguidament explicarem el procediment per a generar els nous nodes que definiran la forma i l'estructura de l'arbre que més tard modelarem.

Tal i com hem esmentat en diverses ocasions, a diferència de prendre un algorisme de creixement basat en la recursivitat de l'estructura de branques, experimentem amb la modelació mitjançant un algorisme de colonització espacial.

Per a generar l'arbre seguirem una construcció iterativa. A cada iteració, cada punt d'atracció del conjunt de punts d'atracció A estarà assignat a un únic node de l'arbre, el més proper, mitjançant una consulta al diagrama de Voronoi dels nodes de l'arbre. Si ho mirem des de la perspectiva del conjunt de nodes de l'arbre S , cada node de l'arbre tindrà assignat un conjunt de punts d'atracció que potencialment afectaran el seu creixement, depenent de si el node de l'arbre està dins de la seva distància d'influència D_i . Adicionalment a la comprovació de la distància d'influència, fem també la comprovació de la distància de mort D_k , que en cas de donar positiva eliminarem el punt d'atracció que estiguem comprovant. Observem que per a cada punt d'atracció només cal comprovar la distància de mort des d'un únic node de l'arbre, ja que la mateixa construcció del diagrama de Voronoi ens assegura que aquest node de l'arbre és el més proper al punt d'atracció donat, i per tant, si aquest no passa la comprovació de la d_k , cap altre node la passarà per al punt d'atracció donat.

Un cop hem determinat quins punts d'atracció afecten a cada node de l'arbre, si aquest conjunt C

no és buit es generarà un nou node v' que sortirà de l'original v , a una distància D (distància del nou node), i amb un sentit \vec{n} respecte el node de l'arbre original que serà el sentit mitjà no ponderat de tots els punts d'atracció de C respecte el node de l'arbre en qüestió.

$$v' = v + D \frac{\vec{n}}{\|\vec{n}\|} \quad (7.1)$$

$$\vec{n} = \sum_{s \in C} (v - s) \quad (7.2)$$

Per a cada node nou que afegim, relacionarem pare amb fill, i fill amb pare per a tenir la construcció de l'esquelet i poder recórrer-la en qualsevol sentit. Veurem que serà important tenir aquesta bidireccionalitat per a generar el model posteriorment. També és important esmentar que un node pot tenir diversos fills, que sorgeixin a iteracions diferents, però tots els nodes (excepte l'arrel) tindran únicament un pare.

També, cada node nou que afegim actualitzarà la triangulació de Delaunay T de l'arbre, que requerirà tenir la nova assignació entre nodes de l'arbre i punts d'atracció. Amb aquesta finalitat hem fet servir la llibreria *Computational Geometry Algorithms Library (CGAL)*, que inclou un mòdul per a construir triangulacions de Delaunay dinàmicament i per a fer consultes de proximitat ràpidament.

Podem concloure d'aquesta explicació que cada iteració com a màxim duplicarà el nombre de nodes, i també és possible que no se n'afegeixi cap.

A l'algorisme 7.2 descrivim en pseudo-codi l'algorisme de colonització espacial.

De l'algorisme 7.2 volem destacar primerament la condició d'aturada. Aquesta condició pot definir-se de diferents maneres, segons els interessos de l'usuari. Per exemple, l'execució es pot aturar en un nombre determinat d'iteracions, o també es pot aturar quan no queda cap punt d'atracció (mala idea teòricament, ja que si un punt d'atracció mai és prou proper a un node de l'arbre degut a la seva distància d'influència, mai s'eliminarà), o si no quan no tinguem més nodes de l'arbre a afegir, és a dir, quan *afegir* sigui buit. A la nostra implementació hem utilitzat aquesta última opció.

Un altre detall que hem afegit a la nostra implementació és que per a obtenir sempre models complets, en el cas que el node arrel no tingui cap punt d'atracció dins de la seva distància d'influència, anirem fent créixer l'arbre cap al centre de la copa fins que l'algorisme s'executi de forma natural. Així podem tenir troncs llargs amb copes de l'arbre petites sense haver de recórrer a utilitzar distàncies d'influència més grans del que volem.

Finalment, també destaquem la importància de l'eficiència de la implementació de les consultes a la triangulació, ja que és la operació amb més pes de l'algorisme de colonització espacial. Estem molt satisfets amb el rendiment de la implementació d'aquesta a la llibreria CGAL, que fa que la

```

Data:  $S, A, D, D_i, D_k, T$ 
Result:  $S$  té els nodes de l'arbre connectats que formen el seu esquelet, i  $C$  està actualitzat amb la
potencial pèrdua de punts d'atracció

while !condició d'aturada do
     $aAfegir \leftarrow \emptyset$ ;
    foreach  $c$  in  $A$  do
        Consultem a  $T$  el node de l'arbre  $s$  més proper;
        if  $Distància(s,c) \leq D_i$  then
            if  $Distància(s,c) \leq D_k$  then
                Eliminem  $c$  de  $A$ ;
            else
                 $aAfegir[s] += c$ ;
            end
        end
    end
    foreach  $s$  in  $S$  do
        if  $aAfegir[s] \neq \emptyset$  then
            Calculem el nou node  $s'$  amb la informació de  $aAfegir[s]$  i  $D$ ;
            Afegim  $s'$  a  $S$ ;
            Assignem  $s$  a  $s'$  i  $s'$  a  $s$ ;
            Afegim  $s'$  a  $T$ ;
        end
    end
    Actualització de la condició d'aturada;
end

```

Algorisme 7.2: Algorisme de colonització espacial

colonització s'executi amb un rendiment major al que esperàvem, de l'ordre d'uns pocs segons (menys de 5) per a 2000 punts d'atracció sobre una màquina amb un core i7.

Seguidament volem analitzar l'efecte de les diferents variables sobre la morfologia de l'esquelet de l'arbre resultant.

Quantitat de punts d'atracció

Incrementar el nombre de punts d'atracció mantenint la distància del nou node, la distància d'influència i la distància de mort generalment resultarà en una estructura de l'arbre amb un nombre de branques més elevat, i si fem servir una distribució aleatòria uniforme dels punts d'atracció, la seva forma general s'ajustarà més a la forma de la copa de l'arbre.

A la figura 7.5 observem l'efecte que acabem de descriure, veiem que la figura de l'esquerra és molt més complexa i s'ajusta més a la forma de la copa esfèrica que definia l'arbre.

Distància del nou node de l'arbre

La distància del nou node D afectarà a la quantitat de branques, i també a la qualitat del model final. Amb una distància del nou node major, veurem que hi haurà menys nodes de l'arbre dels que

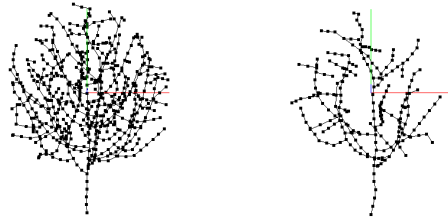


Figura 7.5: Efecte d'alterar el nombre de punts d'atracció. A l'esquerra, una estructura formada per 1000 punts d'atracció, i a la dreta una estructura formada per 100 punts d'atracció. Ambdues copes de l'arbre són esfèriques.

Font: Execució del nostre programa.

puguin créixer nous nodes, i per tant la quantitat de branques noves normalment serà menor. Pel que fa a la qualitat del model final, donat que utilitzarem les connexions entre nodes com eixos de les branques, veurem que tindrem canvis més bruscs de direcció i una estructura generalment més rectilínia.

A la figura 7.6 veiem la diferència entre dos arbres, on l'arbre de l'esquerra té una distància del nou node de $2D$. S'observa clarament l'efecte que hem definit anteriorment.



Figura 7.6: Efecte d'alterar la distància del nou node de l'arbre. A l'esquerra, una estructura on la distància del nou node és D , i a la dreta una estructura on aquesta distància és $2D$. Ambdues copes de l'arbre són esfèriques.

Font: Execució del nostre programa.

Distància d'influència

La distància d'influència D_i afectarà principalment la forma de les branques. Amb una distància d'influència més gran, veurem que més punts d'atracció afectaran el creixement de cada node de l'arbre, i per tant les branques acostumaran a créixer d'una forma més rectilínia (cap al centre del conjunt de nodes d'atracció que l'afecten) que si tenim una distància d'influència més petita.

Podem observar aquest efecte a la figura 7.7, que l'arbre de l'esquerra creix amb unes branques que generalment tendeixen a dirigir-se cap a les zones superiors d'una forma més rectilínia que a l'arbre de la dreta, on trobem més variacions en els sentits de les branques.

Distància de mort

La distància de mort D_k , similarment a la quantitat de punts d'atracció, afectarà principalment al nombre de branques de l'estructura resultant. Amb una distància de mort major, els punts d'atracció seran eliminats més d'hora, i l'efecte d'això serà que naixeran menys nous nodes de l'arbre.

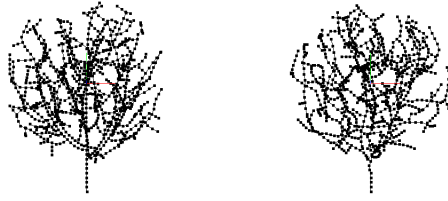


Figura 7.7: Efecte d'alterar la distància d'influència. A l'esquerra, una estructura on la distància d'influència dels punts d'atracció és $20D$, i a la dreta una estructura on aquesta distància és $5D$. Ambdues copes de l'arbre són esfèriques.

Font: Execució del nostre programa.

A la figura 7.8 veiem aquest efecte. Es veu clarament com l'arbre de l'esquerra té moltes més branques que l'arbre de la dreta. Tot i que la diferència entre distàncies de mort és només $2D$, cal tenir en compte que aquests valors representen una part important del radi de la copa de l'arbre (que és $16.6D$), i per tant tot i que pot semblar un canvi molt brusc en el resultat per a un canvi relativament petit de Kd , això no seria tan cert si el radi de la copa fos major.

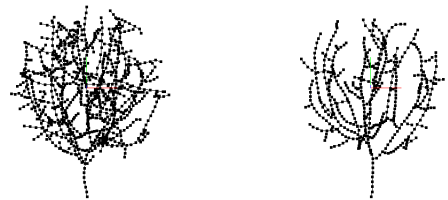


Figura 7.8: Efecte d'alterar la distància de mort. A l'esquerra, una estructura on la distància de mort dels punts d'atracció és $2D$, i a la dreta una estructura on aquesta distància és $4D$. Ambdues copes de l'arbre són esfèriques.

Font: Execució del nostre programa.

7.2.4 Generació del model

Un cop tenim l'esquelet de l'arbre, ja podem procedir a la seva modelització tridimensional. Utilitzarem tots els segments entre nodes de l'arbre com a eixos del tronc i les branques que formaran el model final.

Secció

Hi ha diverses consideracions que hem de prendre quan volem passar de l'esquelet al model. Una de les més importants és determinar l'amplada de cada branca, que anomenarem secció, per tal d'obtenir una representació realista de l'arbre. Hi ha diversos estudis que fan referència a aquest fet, més notablement destaquem el de *A quantitative analysis of plant form — the pipe model theory. I. Basic analyses*[10], que estableix la secció d'un punt de bifurcació en termes de les seccions r_i de les branques superiors incidents. D'aquesta manera, travessa l'arbre de puntes a arrel per a determinar la secció de cada component de l'arbre. Estableix aquesta relació amb la següent fórmula:

$$r^n = r_1^n + r_2^n \quad (7.3)$$

on n és un paràmetre donat, normalment entre 2 i 3, tal i com es discuteix a *Trees and networks in biological models*[13].

El nostre programa utilitza un enfocament diferent que també ha donat bons resultats. Similarment al mètode descrit anteriorment, recorrem l'arbre de puntes a arrel, i a cada bifurcació utilitzem la següent fórmula per a determinar la secció resultant:

$$r = \max(r_i) + 0.07 * \lceil \frac{\# \text{ branques incidents}}{2} \rceil \quad (7.4)$$

que ens dóna uns resultats satisfactoris a la nostra implementació i escala dels nostres models. Com és evident, la constant podria haver de variar si s'utilitzen altres escales per al model. Cal notar també que la divisió és entera, i per tant la secció només creixerà quan hi hagi més d'una branca incident.

El recorregut que determina la secció de cada branca es fa, per a estalviar temps d'execució, durant el mateix procés de dibuix a pantalla del model. Això ens permet fer un únic recorregut de l'estructura, i també ens permet poder afegir nous nodes i actualitzar les seccions dinàmicament.

Model

A banda de la secció, per a definir el model de l'arbre caldrà que definim una sèrie de vèrtexs i les seves unions en forma de triangles. Addicionalment, també definirem els vectors normals de cada cara per a poder afegir il·luminació. Al capítol 8 explorarem quins altres atributs que s'han escapat de l'abast d'aquest projecte podríem afegir al model per tal de donar-li més fotorealisme.

El procediment per a determinar la localització dels vèrtexs que formaran el model és el que expliquem a continuació.

Tal i com hem explicat a la secció 7.1.11, utilitzarem una estructura formada per discs units entre ells per a modelar l'arbre final. Durant la creació de l'arbre amb l'algorisme de colonització espacial també calcularem dinàmicament el vector mitjà dels fills de cada node, que utilitzarem en el procés de dibuix a pantalla per a orientar el disc d'una forma més natural. Amb aquest vector, i el vector que indica el sentit del node pare al node amb el que estem treballant determinarem un sentit intermedi per a orientar el disc resultant. Un cop tenim l'orientació del disc podem calcular les vuit orientacions que descriuen els vuit vèrtexs que formaran el disc.

A la figura 7.9 il·lustrem aquesta explicació. El segment verd \vec{v} representa la direcció mitjana dels fills del node de l'arbre s del que volem determinar el disc, i el segment blau \vec{b} és el vector intermedi entre aquest verd i la direcció \vec{p} en la que es troba el pare del node. Rotant sobre el producte vectorial entre aquests dos vectors determinem els punts que defineixen el disc.

Per a trobar aquests vuit punts simplement assignem una de les coordenades \vec{x} o \vec{z} a 0, i intercanviem les altres dues, una d'elles amb un canvi de signe. Així trobem els quatre vectors normals sobre l'eix de rotació, i els altres quatre els trobem fent la suma dels grups de dos d'aquests últims quatre vectors que no tinguin direccions oposades i normalitzant.

En el cas extrem en què els vectors \vec{v} i \vec{b} són colineals, utilitzarem qualsevol d'aquests dos com a normal del disc i seguirem el mateix procediment que hem explicat anteriorment per a determinar els vèrtexs.

$$\vec{v} = \sum_{f \text{ in fills}(s)} \frac{f - s}{\|f - s\|} \text{ (normalitzat)} \quad (7.5)$$

$$\vec{b} = \frac{\vec{p}}{\|\vec{p}\|} + \vec{v} \text{ (normalitzat)} \quad (7.6)$$

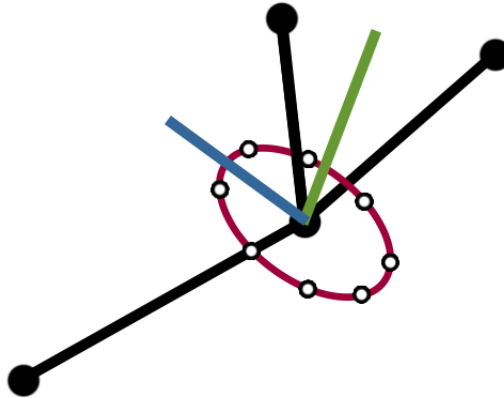


Figura 7.9: Càlcul del disc d'un node de l'arbre. El segment verd mostra la direcció mitjana dels seus fills, i el segment blau mostra el vector intermediari entre el verd i el seu pare. Aquest vector blau el rotem per a determinar els vuit punts que defineixen el disc.

Font: Pròpia.

La decisió d'utilitzar vuit vèrtexs per a l'especificació de cada disc ens ha permès elaborar un codi específic que no utilitza funcions trigonomètriques com sinus i cosinus tal i com hem descrit, i també tenir un nombre de cares raonable per a definir la malla del model. El desavantatge d'haver fet això és que no es pot fer una transformació immediata a un nombre de cares diferent amb un simple canvi de variable. De totes maneres, en el visor hem utilitzat interpolació de la il·luminació de les cares per a donar un efecte cilíndric a l'estructura, que tal i com veurem, queda molt convincent, així que tampoc és estrictament necessari utilitzar un nombre de cares major.

Així doncs, travessarem l'arbre recursivament determinant la secció i el disc de cada node, i amb aquesta i el procediment explicat anteriorment determinem el conjunt de vèrtexs que acaben formant el model de l'arbre.

Seguidament indiquem en pseudo-codi el procediment per a dibuixar l'arbre a pantalla, a l'algorisme 7.3.

```
Data: Arbre, node S  
Result: Es dibuixa el model de l'arbre. Es retorna la secció del node actual.  
if S no té fills then  
    | Dibuíxem el disc del node pare de S i la punta de la branca;  
    | return section0;  
else  
    | foreach fill F de S do  
    |   | secc += Render(Arbre, F)  
    | end  
    | Calculem la nostra secció section amb el max(secc);  
    | if S no és arrel then  
    |   | Dibuíxem el disc del node pare de S i l'unim a S;  
    |   | end  
    |   | return section;  
end
```

Algorisme 7.3: Algorisme de dibuix a pantalla de l'arbre.

Tal i com veiem, no cal dibuixar l'arrel ja que el segon node dibuixa el seu disc. Amb l'especificació del disc assegurem que encara que les cares de cada branca siguin 8, les unions sempre estan correctament encaixades ja que els vèrtexs dels discs estan orientats segons els eixos de coordenades.

7.3 Implementació de la interfície

Ja hem vist com implementem l'arbre, des del mètode que utilitzem per a generar els nodes que el formen, fins a la seva renderització.

Ara veurem com hem implementat la interfície amb el visor OpenGL que ens permet visualitzar i navegar el model que hem generat per a concloure amb la implementació del projecte. Hem separat el visor en el propi visor, que controla el dibuix a pantalla i la càmera que controla la navegació.

7.3.1 Càmera

Tal i com hem dit, la càmera s'ocupa de controlar la part de l'escena que veiem. Inicialment, utilitza el punt més elevat de la copa de l'arbre i les coordenades del node arrel per a centrar la visió al mig de l'arbre i per a poder incloure tot el model.

La càmera també inclou els procediments de navegació, com el moviment en qualsevol direcció (incloent zoom) i la rotació del model. Aquests procediments els invoca el visor quan rep la senyalització corresponent.

7.3.2 Visor

El visor implementa les funcions que tradueixen l'estructura de l'arbre en una escena tridimensional que podem observar per pantalla. El visor és una eina clau per a la verificació la validació del

nostre progrés que vam implementar inicialment i hem anat ampliant amb funcionalitats i millores gràfiques per a satisfer les nostres necessitats i per a arribar als objectius que havíem establert.

Donat que la interfície gràfica que hem decidit utilitzar està basada sobre Qt, utilitzem el mòdul de `QGLWidget` per a combinar Qt i OpenGL a la nostra aplicació. Això ens proporciona les funcionalitats per a incorporar i connectar el nostre visor en un entorn gràfic, i ens permet escoltar els esdeveniments que vénen dels perifèrics per a actualitzar la càmera d'acord amb l'esdeveniment que s'hagi produït.

Adicionalment, hem inclòs la possibilitat de veure també els punts d'atracció i l'esquelet de l'arbre generat per l'algorisme de colonització espacial amb una finalitat instructiva, informativa i de suport al desenvolupament del projecte.

Amb totes aquestes funcionalitats implementades, ja estem llestos per a veure i analitzar el model de l'arbre que ha generat el nostre programa.



Figura 7.10: Model de l'arbre tal i com el veiem en el visor.

Font: Execució del nostre programa.

7.4 Exportació del model

Aquests vèrtexs i normals de les cares que hem calculat anteriorment les podem utilitzar per a exportar un model que ens permetrà utilitzar el nostre resultat en altres entorns software i també afegir-lo a altres projectes per a poder fer-ne ús.

Existeixen un gran nombre de formats per a la representació de models, nosaltres en particular hem decidit utilitzar el format [OBJ](#) ja que ens permet especificar totes les propietats que formen el

nostre model, i a més a més, és un format obert que podem utilitzar sense llicència.

El format OBJ es guarda en un fitxer de text pla (en anglès *plain text*), i conté un llistat de les coordenades dels vèrtexs, les seves coordenades de textura, les normals i les cares quan s'escau. També es poden especificar altres components, com per exemple materials, però no ho tractarem en aquest projecte.

Per a determinar els vèrtexs, les normals i les cares podem utilitzar un recorregut com el que definim a l'algorisme de dibuix a pantalla [7.3](#) que en comptes de pintar a pantalla els discs, guardi els vèrtexs, les normals i les cares a una estructura que recorrerem per a fer l'escriptura del fitxer OBJ resultant.

Treball futur

Degut a restriccions de l'abast del projecte, de complexitat i temporals, hi ha funcionalitats i millores que hem considerat però no hem explorat ni implementat en el programa final a data de la presentació de la memòria.

El disseny del projecte està preparat per a adoptar les millores i altres addicions que hem considerat, que seguidament exposem.

8.1 Millores morfològiques

Les millores morfològiques inclouen totes aquelles millores o alternatives que afecten la morfologia del model, és a dir, que afecten la forma en què es calcula el creixement de l'arbre o que inclouen enfocaments més realistes cap a la seva modelització (pel que fa la seva forma).

8.1.1 Curvatura dels segments

A *Modeling the Mighty Maple*[2] es descriu un procediment per a donar realisme a l'estructura de l'arbre substituint els segments rectes que utilitzem per corbes, com veiem a la figura 8.1.

Tot i que aquest canvi pot semblar inicialment subtil, ofereix una representació molt més natural d'una estructura com la d'un arbre, que creix naturalment de tal forma que les branques tenen una curvatura suau cap al punt de suport.

8.1.2 Estructura de les bifurcacions

A *Modeling the Mighty Maple*[2] també es tracta una estructura més reminiscent a la d'un arbre natural en les bifurcacions de les branques. A la figura 8.2 s'il·lustra aquesta estructura.

Aquesta millora també ofereix un resultat més realista morfològicament, però introduïa força complexitat a la modelització, com per exemple quan tenim una separació de més de dues branques des d'un mateix node, i el procediment per a la seva construcció no està explicat a fons a l'article que citem.

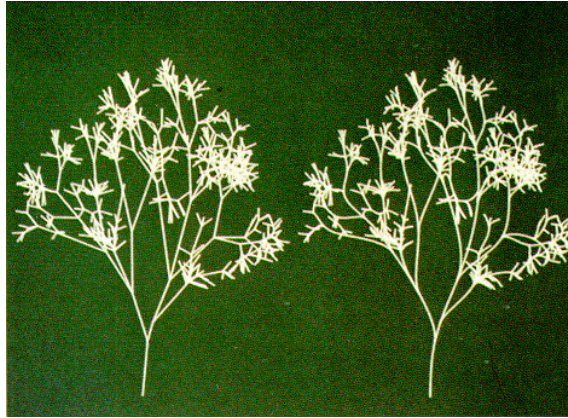


Figura 8.1: A l'esquerra, un arbre format per segments rectes, i a la dreta, un arbre amb els segments interpolats per a donar una estructura corba més realista.

Font: *Modeling the Mighty Maple*[2].

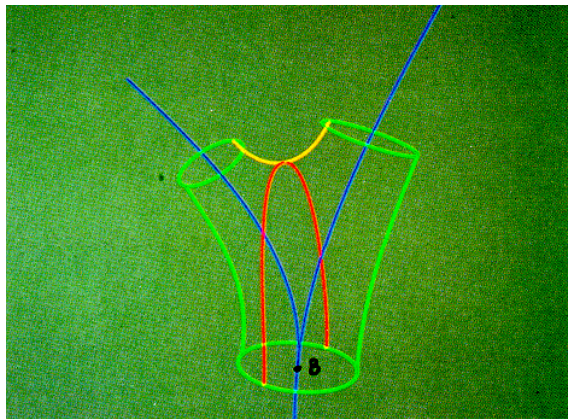


Figura 8.2: Estructura per a les bifurcacions de les branques. Es representen les corbes que formen els seus eixos centrals i l'estructura envoltant.

Font: *Modeling the Mighty Maple*[2].

8.1.3 Efecte de les forces externes

A *Self-organizing tree models for image synthesis*[11] s'estudia, entre d'altres, l'efecte de l'aplicació de forces externes com la gravetat en la construcció dels arbres, de formes elaborades com les que mostrem a la figura 8.3 que introdueix tropismes cap avall en les fases avançades de la simulació.

Aquests efectes donen un altre grau de realisme al model, ja que simulen més precisament el comportament natural del creixement dels arbres.

8.1.4 Ús de models naturals addicionals

A *Self-organizing tree models for image synthesis*[11] s'exposa també l'ús del model de propagacions d'ombres (que podem veure a la figura 8.4) per a afectar el creixement dels arbres.

Aquest model s'ajusta a la realitat ja que el creixement natural dels arbres involucra una part de competició per la llum que aquest model tracta.



Figura 8.3: Arbre amb un tropisme cap avall cap a les puntes de les branques.

Font: *Self-organizing tree models for image synthesis*[11].

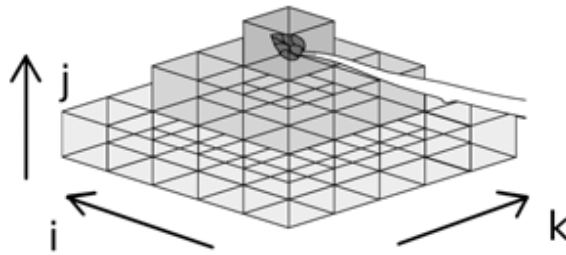


Figura 8.4: Model de propagació d'ombres.

Font: *Self-organizing tree models for image synthesis*[11].

8.1.5 Millora de la modelització de les branques i tronc

Una altra millora que es podria incloure és la introducció de superfícies no planes i irregulars per a modelitzar les branques i el tronc. Aquesta millora és arriscada d'implementar en la morfologia del model, i proposarem una alternativa a la següent secció per a no córrer riscos de casos extrems on aquestes irregularitats poden provocar que el model quedi d'una forma no desitjada.

A més a més, aplicar aquesta millora a la morfologia del model potencialment complica molt la seva geometria, i això pot tenir un efecte negatiu en el rendiment del model, sobretot comparat amb l'alternativa que presentarem.

Adicionalment, una millora que sí que seria viable afegir és la simulació d'irregularitat a la base del tronc, tal i com veiem a la figura 8.5.

8.2 Millores gràfiques

Les millores gràfiques inclouen aquelles millores que podem aplicar al model o al visor per tal d'obtenir un resultat més realista sense haver d'alterar la morfologia del model.

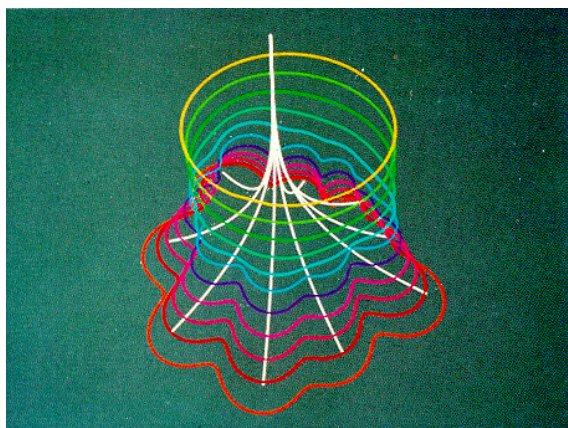


Figura 8.5: Simulació de la irregularitat de la base del tronc.

Font: *Self-organizing tree models for image synthesis*[11].

8.2.1 Addició de fulles, flors i fruits

La principal millora que havíem estudiat i que malauradament no hem pogut incloure degut a restriccions temporals és l'addició de fulles, flors i fruits al model. A la figura 8.6 es pot observar l'efecte d'afegir aquests elements al model.

Es va estudiar la inclusió d'aquesta millora de diverses formes, que les esmentem a continuació:

- Ús de *quads* texturats. La primera opció que vam plantejar va ser dibuixar quadrats transparents que continguessin la textura d'una fulla, flor o fruit i posicionar-los perpendicularment a les branques per a donar l'efecte que sorgien d'aquests.
- Ús d'impostors¹: Es va plantejar també l'ús d'impostors que continguessin una multitud de fulles, flors i fruits per tal d'incloure aquesta millora gràfica. A la figura 8.7 veiem de quina manera s'utilitzen els impostors per a donar una sensació de tridimensionalitat.
- Ús de models: Finalment, també ens vam plantejar l'ús de models de forma similar als *quads* texturats per a donar un realisme tridimensional.

8.2.2 Ús de textures

L'altra millora que vam estudiar i no vam poder incloure degut a les restriccions temporals ha estat la de l'ús de textures.

En comptes d'utilitzar un color sòlid com fem ara mateix, es poden assignar coordenades de textura a les cares del model per tal de poder aplicar-lis texturació. Amb això, podríem haver utilitzat una textura d'escorça per a cobrir les cares del nostre model.

¹Un impostor és una imatge bidimensional que normalment es troba conjuntament a altres impostors amb la finalitat d'intentar fer que l'usuari vegi una figura tridimensional. Normalment els trobem a videojocs.



Figura 8.6: Arbre amb flors.

Font: *Modeling Trees with a Space Colonization Algorithm*[1].

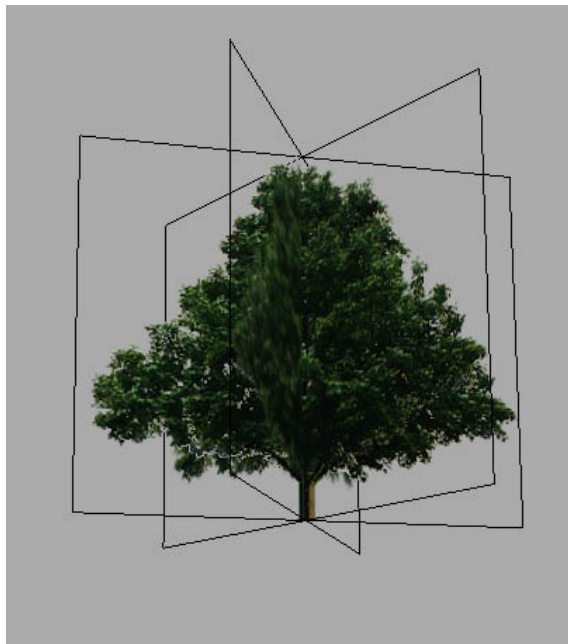


Figura 8.7: Ús d'impostors per a donar realisme a un arbre.

Font: Captura de pantalla del programa Unity3D.

8.2.3 Ús de mapejat topològic (*bump maps*)

Aquesta millora, molt relacionada amb l'anterior, proposa l'ús de *bump maps*, que és una tècnica per a donar un aspecte rugós a les superfícies sense alterar la geometria. Amb aquesta tècnica variem les normals de la superfície i per tant afectem la il·luminació d'aquesta, que resulta en un efecte de rugositat molt convincent.

A la figura 8.8 veiem l'efecte d'aplicar texturació i mapejat topològic a l'estructura, juntament amb altres millores que hem presentat.

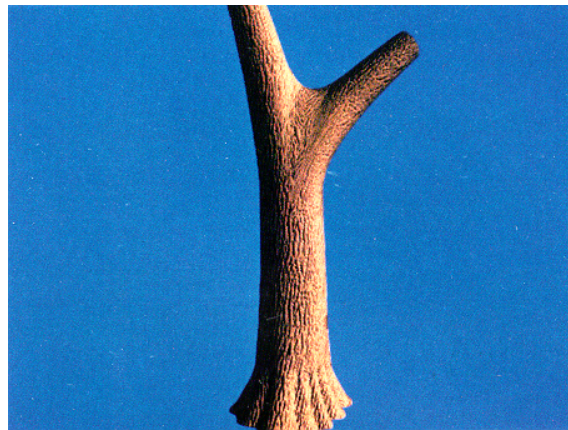


Figura 8.8: Secció d'un arbre amb diverses millores de les que hem presentat. Veiem la curvatura dels segments, l'estructura de les bifurcacions, la irregularitat de la base del tronc, l'aplicació de textures i l'ús del mapejat topològic.

Font: *Self-organizing tree models for image synthesis*[11].

Conclusions

Com a cloenda d'aquest Treball Final de Grau volem fer un breu resum dels resultats obtinguts i de l'experiència de l'elaboració d'aquest projecte.

Hem implementat satisfactòriament totes les tasques que es descriuen en els objectius del treball i hem pogut comprovar com l'algorisme de colonització espacial és un mètode bo per a la descripció del creixement natural de vegetació, que resulta en estructures que defineixen models molt semblants als arbres reals.

Aquest projecte ens ha afrontat per primera vegada amb la responsabilitat d'encarar i solucionar un problema que es discuteix i s'estudia activament a la comunitat científica, com és el de la generació procedural, i del que n'extraïem coneixements nous i metodologies noves que esperem poder aplicar en un futur per a encarar altres projectes.

Tal i com hem vist, el nostre projecte és una pinzellada del món de possibilitats que és el de la generació procedural, i el problema està encara obert a revisions, ampliacions i millores que no hem pogut implementar degut a les nostres restriccions d'abast, però que hem exposat per tal de guiar el seu desenvolupament.

També podem extreure que la generació procedural és una eina molt potent i amb grans possibilitats i creiem que guanyarà força en el futur, amb hardware millor i capaç de fer simulacions molt fidels a les dels entorns reals.

Bibliografia

- [1] Runions A., Lane B., Prusinkiewicz P.: *Modeling Trees with a Space Colonization Algorithm*. Eurographics Workshop on Natural Phenomena (2007), pp. 63-70
- [2] Bloomenthal J.: *Modeling the Mighty Maple*. Computer Graphics (SIGGRAPH '85 Conference Proceedings) 19, 3 (Juliol 1985), pp. 305-311
- [3] Runions A., Fuhrer M., Lane B., Federl P., Rolland-Lagan A.-G., Prusinkiewicz P.: *Modeling and visualization of leaf venation patterns*. ACM Transactions on Graphics 24, 3 (2005), pp. 702-711
- [4] Honda H.: *Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body*. Journal of Theoretical Biology 31 (1971), pp. 331-338
- [5] Aono M., Kunii T.L.: *Botanical tree image generation*. IEEE Computer Graphics and Applications 4, 5 (1984), pp. 10-34
- [6] Oppenheimer P.: *Real time design and animation of fractal plants and trees*. Computer Graphics (SIGGRAPH '86 Conference Proceedings) 20, 4 (1986), pp. 55-64
- [7] Reeves W.T., Blau R.: *Approximate and probabilistic algorithms for shading and rendering structured particle systems*. Computer Graphics (SIGGRAPH '85 Conference Proceedings) 19, 3 (Juliol 1985), pp. 313-322
- [8] Weber J., Penn J.: *Creation and rendering of realistic trees*. Computer Graphics (SIGGRAPH '95 Conference Proceedings) 29 (1996), pp. 119-128
- [9] Prusinkiewicz P., Mündermann L., Karwowski R., Lane B.: *Creation and rendering of realistic trees*. Computer Graphi
- [10] Shinozaki K., Yoda K., Hozumi K., Kira T.: *A quantitative analysis of plant form — the pipe model theory. I. Basic analyses*. Japanese Journal of Ecology 14, 3 (1964), pp. 97-104
- [11] Palubicki W., Horel K., Longay S., Runions A., Lane B., Měch R., Prusinkiewicz P.: *Self-organizing tree models for image synthesis*. ACM SIGGRAPH '09 papers (2009), pp. 1-10

- [12] Palubicki W.: *Fuzzy Plant Modeling with OpenGL — Novel Approaches in Simulating Phototropism and Environmental Conditions*. VDM Verlag, Saarbrücken (2007)
- [13] MacDonald N.: *Trees and networks in biological models*. J. Wiley & Sons, New York, 1983, pp. 131-135