

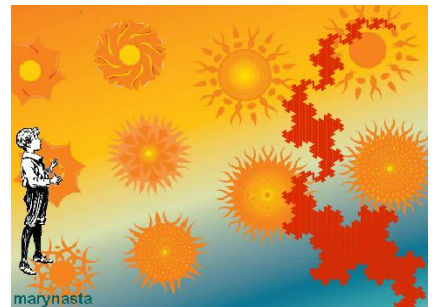


ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΝΑΣΤΑΚΟΥ ΜΑΡΙΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

L-SYSTEMS ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΤΟ SCRATCH



ΕΠΙΒΛΕΠΩΝ: ΛΕΚΤΩΡ ΠΛΑΤΗΣ ΝΙΚΟΛΑΟΣ

2013

Περιεχόμενα

Περίληψη.....	5
Summary.....	7
ΕΙΣΑΓΩΓΗ	9
1. Τι είναι τα L-systems	11
1.1. Γενικά	11
1.2. Γραμματική	11
1.3. Εντός των L-systems	14
1.4. Παραδείγματα L-systems	14
1.4.1. L-systems σε άξονα (πραγματικών αριθμών \mathbb{R}):	14
1.4.2. Καμπύλες πλήρωσης χώρου (Peano's curves, Hilbert curve, Dekking's church, kolams)	15
1.4.3. median space-filling curves (Lévy C curve, Harter-Heighway dragon curve, Davis-Knuth terdragon)	16
1.4.4. Tiling	17
1.4.5. Δέντρα φυτά και τα παρόμοια	18
1.5. Είδη L-systems	20
1.1.1. Πρότυπα L-systems	20
1.1.2. Στοχαστικά L-systems	21
1.1.3. Παραμετρικά L-systems	22
1.1.4. L-systems εξαρτώμενα από τα συμφραζόμενα (context-sensitive)	22
1.1.5. Χρονικά L-systems	23
2. Προγραμματισμός με το scratch	25
2.1. Προγραμματιστικό πολυμεσικό περιβάλλον	25
2.2. Scratched – ιστότοπος	29
2.3. Scratch 2.0	30
3. Υλοποίηση στο scratch	32
3.1. Παραγωγή της επόμενης γενιάς	32
3.2. Σχεδίαση της τρέχουσας γενιάς	33
3.3. Καλοκαίρι (Terdragon curve)	34
3.4. Φθινόπωρο (trees)	34
3.5. Χειμώνας	36

3.5.1.	“Pleasant error”	36
3.5.2.	“The Koch snowflake”	36
3.5.3.	“Snowflake2”	37
3.5.4.	Snowflake3	37
3.6.	Άνοιξη	38
3.7.	Τρίγωνο Sierpinski	38
3.7.1.	Sierpinski_triangle	38
3.7.2.	sierpinski version2	40
3.7.3.	sierpinski_triangle στο scratch 2.0	41
3.8.	“Real” olive-tree	42
3.9.	star_in_star	43
3.10.	tiling-stavros	43
3.11.	Kite and dart tiling	44

Περίληψη

Η έκθεση αυτή περιλαμβάνει τρεις ενότητες. Στη πρώτη ενότητα δίνονται ο ορισμός των L-systems, η γραμματική και παραδείγματα L-systems. Αναλύονται τα είδη των L-systems και οι διαφορές μεταξύ τους με χρήση διαφόρων παραδειγμάτων.

Η δεύτερη ενότητα περιλαμβάνει μια παρουσίαση του προγραμματιστικού περιβάλλοντος του scratch, του εκπαιδευτικού δικτυακού τόπου scratched, του προγραμματιστικού online ιστότοπου scratch2.0, της κοινότητας <http://scratch.mit.edu/> και κυρίως των ιδιαίτερων δυνατοτήτων τους που τα καθιστούν σημαντικά μέσα και εργαλεία διδασκαλίας.

Η τελευταία ενότητα είναι μια παρουσίαση L-systems υλοποιημένων με scratch προγραμματισμό αλλά και η δημοσίευσή τους στη κοινότητα <http://scratch.mit.edu/>. Για τη παρουσίαση χρησιμοποιήθηκαν εικόνες τόσο από τα έργα όσο και των κομματιών κώδικα για την ανάλυσή τους.

Πιστεύω ότι όποιος διαβάσει την εργασία αυτή θα προκληθεί το ενδιαφέρον του να ενασχοληθεί με το ιδιαίτερο αυτό αντικείμενο και όποιος έχει την γνώση να προσπαθήσει να το επεκτείνει.

Summary

This report contains three sections. In the first section the definition of L-systems, the grammar and examples of L-systems is given. All kinds of L-systems are analyzed and the differences between them are shown with examples.

The second section includes , an introduction of scratch programming , the educational site scratched, the online programming scratch 2.0, the community site <http://scratch.mit.edu/> and mainly their special potentials that make them important means and tools of teaching.

The last section is an presentation of L-systems implemented with scratch programming as well as their publishing to the scratch community. Pictures from projects and analyzed stripts have been used for the introduction.

It is my belief that whoever reads this will become interested enough to start studying this particular subject and whoever has the knowledge will attempt to extend it.

ΕΙΣΑΓΩΓΗ

Μια από τις μεγαλύτερες δυνάμεις των μαθηματικών είναι ότι μπορούν να περιγράψουν πράγματα. Γεωμετρικά σχήματα όπως ο κύκλος το τρίγωνο το παραλληλόγραμμο μπορούν να χρησιμοποιηθούν για να περιγράψουν ανθρώπινες κατασκευές με αξιοσημείωτη ακρίβεια.

Όμως τι σχήμα έχει ένα λουλούδι, ένα σύννεφο, τα φτερά των πουλιών, τα φιορδά, οι γαλαξίες; Ακόμα το κουνουπίδι ή το μπρόκολο;
Ποια είναι τα χαρακτηριστικά τους;

Όταν κοιτάζουμε τα φυτά αυτό που βρίσκουμε όμορφο είναι το σχήμα και η μορφή τους. Αλλά υπάρχει κι ένα άλλο επίπεδο ομορφιάς. Πρόκειται για μια κρυμμένη ομορφιά. Δεν είναι αυτό που βλέπουμε. Είναι η ομορφιά της κατανόησης των μηχανισμών οι οποίοι δημιουργούν αυτό το σχήμα.

Ο Prusinkiewicz μελετά την ανάπτυξη των φυτών και στη δουλειά του στρέφεται σε μια νέα γεωμετρία η οποία περιγράφει τους σχηματισμούς της φύσης – τα [fractals](#). Ένα fractal είναι ένα γεωμετρικό σχήμα το οποίο μπορεί να χωριστεί σε τμήματα, καθένα από τα οποία είναι αντίγραφο μικρότερου μεγέθους, του ολόκληρου. Ο Mandelbrot επεσήμανε ότι αυτές οι μορφές έχουν πολλά εντυπωσιακά χαρακτηριστικά όπως αισθητικότητα και πολυπλοκότητα που προκύπτουν όμως από έναν απλό ορισμό.

Η φύση είναι γεμάτη από fractals. Παράδειγμα: Σ' ένα δέντρο τα μικρά κλαδιά φαίνονται όμοια με τα μεγαλύτερα κλαδιά που και αυτά φαίνονται όμοια με ολόκληρο το δέντρο. Παρατηρώντας τα κλαδιά των δέντρων διαπιστώνουμε ότι επεκτείνονται με μια τακτική διαδικασία. Ένα κεντρικό κλαδί μπορεί να διακλαδίζεται σε δύο. Κατόπιν το καθένα σε άλλα δύο κ.λ.π. Αν εξετάσουμε τη διάταξη των φύλλων θα παρατηρήσουμε ότι ακολουθείται ένα οργανωμένο σχέδιο. Είναι σαν κάποιος να αποφασίζει με ορισμένες ακριβείς οδηγίες για το πώς τα φυτά θα μεγαλώσουν. Παρομοίως αν παρατηρήσουμε ένα μικρό κομμάτι από ένα σύννεφο ή ένα μικρό τμήμα της φτέρης διαπιστώνουμε ότι είναι σχεδόν μικρογραφία του συνόλου.

Όπως οι ρώσικες κούκλες ματριούσκι τα fractals είτε είναι φυσικά είτε τεχνητά διατηρούν τη φυσική γεωμετρία τους σε κάθε κλίμακα δηλαδή επαναλαμβάνουν κάθε τόσο τον εαυτό τους. Το χαρακτηριστικό αυτό των fractals όπου τα μικρά τμήματα φαίνονται όμοια με τα μεγαλύτερα και με ολόκληρο το σχηματισμό ονομάζεται **αυτοομοιότητα**.

Μια άλλη ιδιότητα τους είναι ότι δεν είναι δισδιάστατα αλλά έχουν διάσταση λιγότερο από 2 περίπου 1.58 (και ανάμεσα στο 2 και στο 3 όταν πρόκειται για καμπύλη επιφάνεια). Είναι σαν να έχουμε αφαιρέσει διαδοχικά ξεκινώντας από το αρχικό σχήμα, όμοια κομμάτια.

Σύμφωνα με τον Prusinkiewicz φαίνεται ότι η φύση βασίζεται στη γεωμετρία των fractals. Έτσι για να μελετήσει τη fractal φυσική μορφή των φυτών ο Prusinkiewicz στράφηκε σ' ένα μαθηματικό εργαλείο που ονομάζεται **L-system**. [1], <http://marynasta2.blogspot.gr/search/label/FRACTALS>

Η εργασία αυτή πραγματεύεται τα L-systems και παρουσιάζει την υλοποίησή τους σε scratch προγραμματισμό. Δίνεται ο ορισμός τους και οι διάφοροι τύποι. L-systems καθώς και οι διαφορές μεταξύ τους με χρήση διαφόρων παραδειγμάτων.

Παρουσιάζεται επίσης το προγραμματιστικό περιβάλλον του scratch, του εκπαιδευτικού δικτυακού τόπου scratched, του προγραμματιστικού online ιστότοπου scratch2.0 αλλά και της κοινότητας <http://scratch.mit.edu/> καθώς και των ιδιαίτερων δυνατοτήτων τους που τα καθιστούν σημαντικά μέσα-εργαλεία διδασκαλίας.

Ακολουθεί η ανάλυση της υλοποίησης των L-systems με scratch προγραμματισμό αλλά και η παρουσίασή δημοσιευμένων έργων στη κοινότητα <http://scratch.mit.edu/>. Για τη ανάλυση χρησιμοποιήθηκαν εικόνες τόσο από τα έργα όσο και των κομματιών κώδικα για την ανάλυσή τους.

Ευελπιστώ ότι μέσα από την εκπόνηση της εργασίας αυτής πέτυχα σε σημαντικό βαθμό να παρουσιάσω μηχανισμούς ανάπτυξης των σχηματισμών της φύσης, που άλλωστε είναι και ο κεντρικός στόχος των L-systems.

1. Τι είναι τα L-systems

1.1. Γενικά

Τα **L-systems** για πρώτη φορά παρουσιάστηκαν το 1968 από τον Aristid Lindenmayer (1925-1989) σαν μια μαθηματική θεωρία της ανάπτυξης των φυτών, που χρησιμοποιήθηκε για να μοντελοποιήσει

- τη διαδικασία ανάπτυξης των φυτών,
- τη μορφολογία μεγάλης ποικιλίας οργανισμών, αλλά και
- ανόργανες μορφές όπως κρύσταλλοι ή άλλες φυσικές μορφές.[4]

Ο **Aristid Lindenmayer** ήταν βιολόγος και βοτανολόγος, στο Πανεπιστήμιο της Ουτρέχτης. Σαν βιολόγος εργάστηκε με ζυμομύκητες και νηματοειδείς μύκητες και μελέτησε τα μοτίβα ανάπτυξης των διαφόρων τύπων φυκών όπως τα μπλε πράσινα βακτήρια *Anabaena catenula*. Αρχικά τα **L-systems** σχεδιάστηκαν για να παρέχουν μια επίσημη περιγραφή της ανάπτυξης τέτοιων απλών πολυκύτταρων οργανισμών και για να απεικονίσουν τις σχέσεις γειτονίας μεταξύ φυτικών κυττάρων. Αργότερα το σύστημα αυτό επεκτάθηκε για να περιγράψει ανώτερα φυτά και πολύπλοκες δομές διακλάδωσης.[11]

Τα **L-systems** προσέλκυσαν επίσης το ενδιαφέρον και των επιστημόνων υπολογιστών οι οποίοι τα χρησιμοποίησαν σαν ένα εργαλείο για τη δημιουργία εικόνων φυτών σε ηλεκτρονικό υπολογιστή. Ειδικοί σε γραφικά ηλεκτρονικών υπολογιστών, ιδιαίτερα ο Prusinkiewicz, χρησιμοποίησαν τα L-systems για να παράγουν ρεαλιστική εικόνα των δέντρων των θάμνων και των λουλουδιών. Μερικές από τις εικόνες απεικονίζονται στο «The Algorithmic Beauty of Plants» [4][20]

Κεντρική έννοια των **L-systems** είναι η έννοια της **επανεγγραφής**. Γενικά η επανεγγραφή είναι μια τεχνική με βάση την οποία καθορίζονται σύνθετα αντικείμενα με διαδοχική αντικατάσταση μερών από ένα απλό αρχικό αντικείμενο χρησιμοποιώντας ένα σύνολο κανόνων επανεγγραφής ή αναπαραγωγής. Η επανεγγραφή μπορεί να γίνει αναδρομικά.[11]

Από τη στιγμή που τα **L-systems** είναι βασικά αναδρομικές διαδικασίες αποτελούν καλά παραδείγματα αυτοομοιότητας και συχνά θεωρούνται ένα είδος fractals.[4]

Μοντέλα φυτών και φυσικές οργανικές μορφές είναι εύκολο να προσδιοριστούν καθώς με την αύξηση του επιπέδου της αναδρομής η μορφή σιγά σιγά μεγαλώνει και γίνεται όλο και πιο περίπλοκη. Τα **L-systems** μπορούν ακόμα να χρησιμοποιηθούν για τη παραγωγή αυτοπαρόμοιων fractals όπως τα επαναλαμβανόμενα λειτουργικά συστήματα. Είναι επίσης δημοφιλής στη δημιουργία τεχνητής ζωής.[11]

1.2. Γραμματική

Τα L-systems είναι μια εφαρμογή της παραγωγής τυπικών γραμματικών του Chomsky (1957) από την οποία προέκυψε ένα ευρύ ενδιαφέρον για τα συστήματα επανεγγραφής. [4]

Τα **L-systems** είναι τώρα κοινώς γνωστά ως παραμετρικά συστήματα που ορίζονται από μια πλειάδα:

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

όπου:

- V (το αλφάβητο) είναι ένα **σύνολο συμβόλων** που περιέχουν στοιχεία τα οποία μπορούν ν' αντικατασταθούν (μεταβλητές)
- ω (αξίωμα) είναι μια σειρά συμβόλων από το V που ορίζει την **αρχική κατάσταση** του συστήματος.
- P είναι το **σύνολο κανόνων** παραγωγής που καθορίζουν τον τρόπο με τον οποίο οι μεταβλητές μπορούν ν' αντικατασταθούν με συνδυασμό των σταθερών και άλλων μεταβλητών.[11]

Θα μπορούσαμε να πούμε ότι είναι ένα σύνολο **τερματικών (σταθερές-καθορισμένα σύμβολα)** και **μη τερματικών** συμβόλων καθώς και **ορισμένων κανόνων** οι οποίοι καθορίζουν πως τα μη τερματικά σύμβολα παράγουν νέες σειρές των συμβόλων της γλώσσας.

Τα τερματικά σύμβολα παραμένουν πάντα ίδια και συνήθως χρησιμοποιούνται για τη κλιμάκωση, τη περιστροφή και μετακίνηση των στοιχείων του συστήματος.

Στη πραγματικότητα τα **μη τερματικά σύμβολα** συνήθως ερμηνεύονται ως 2D ή 3D αντικείμενα όπως οι εντολές των γραφικών χελώνας της Logo. Θα μπορούσαν επίσης ν' αντιπροσωπεύουν λέξεις και ήχους.[4]

Το **αξίωμα (αρχική λέξη)** είναι μια σειρά από σύμβολα της γλώσσας και χρησιμοποιείται ως σημείο εκκίνησης στη διαδικασία αντικατάστασης. [4],[11]

Οι **κανόνες παραγωγής** εφαρμόζονται επαναληπτικά ξεκινώντας από την αρχική κατάσταση(αξίωμα) και περιγράφουν πώς θα κτιστεί η λέξη. **Ένας κανόνας είναι μια συμβολοσειρά** από οποιαδήποτε σύμβολα του συνόλου της γλώσσας η οποία **θα αντικαταστήσει ένα μη τερματικό σύμβολο**. Είναι δυνατόν να δημιουργήσουμε κανόνες για κάθε μη τερματικό σύμβολο τυχαίους κανόνες , ευαίσθητους κανόνες πλαισίου κ.λ.π.[11]

Όσο το δυνατόν περισσότεροι κανόνες εφαρμόζονται ταυτόχρονα , σε κάθε επανάληψη. Οι λέξεις κτίζονται αναδρομικά εφαρμόζοντας τους κανόνες παραγωγής σε κάθε επανάληψη για να σχηματίσουν μεγαλύτερες και πολυπλοκότερες λέξεις. Είναι μια διαδικασία που εφαρμόζει τους κανόνες για κάθε μη τερματικό σύμβολο του αξιώματος δημιουργώντας έτσι μια νέα και μεγαλύτερη σειρά. Αυτή η νέα σειρά θα επαναληφθεί τόσες φορές όσες εμείς θα ορίσουμε, δημιουργώντας όλο και μεγαλύτερη σειρά. Η τελική σειρά ονομάζεται L-system string και είναι έτοιμη ν' αναλυθεί όπως θα παρουσιαστεί παρακάτω.[4]

Η ταυτόχρονη εφαρμογή των κανόνων είναι χαρακτηριστικό γνώρισμα που διακρίνει τα **L-systems** από μια τυπική γλώσσα που παράγεται από μια τυπική γραμματική. Αν έπρεπε να παραχθεί μόνο ένας κανόνας κάθε φορά, θα δημιουργείτο μια γλώσσα αντί ενός **L-system**. Έτσι τα **L-systems** είναι αυστηρά υποσύνολα γλωσσών. [4]

Ένα απλό **L-system** ονομαζόμενο «**χελώνα**» (ή αντίστοιχα γάτα του scratch, βλ. και κεφ. 3 και 4) παρουσιάζεται για να δείξουμε πως ένα L-system χρησιμοποιείται για να περιγράψουμε κάποιες δομές.

Η βασική ιδέα της ερμηνείας της «χελώνας» περιγράφεται από τον Prusinkiewicz και τον Hanan παρακάτω:

Γραφικά χελώνας:

Η χελώνα (η πένα για το scratch) είναι ένα εργαλείο ζωγραφικής για τη γεωμετρική ερμηνεία μιας παραγόμενης συμβολοσειράς (string)

Καθώς διαβάζεται η συμβολοσειρά η χελώνα (πένα) εκτελεί συγκεκριμένες εντολές για ορισμένα σύμβολα και αλλάζει τη θέση-κατάστασή της

Μια χελώνα μπορεί να κινηθεί προς οποιαδήποτε κατεύθυνση είτε προς τα εμπρός, είτε προς τα πίσω, είτε προς τ' αριστερά, είτε προς τα δεξιά. Κάθε μετακίνηση ορίζεται με πέντε θεμελιώδη σύμβολα που είναι τα F, +, -, [,] .

Η θέση-κατάσταση της χελώνας ορίζεται από μια τριπλέτα μεταβλητών (x,y,α) όπου τα x,y είναι οι συντεταγμένες που καθορίζουν τη θέση της στο επίπεδο και η γωνία α τη κατεύθυνσή της.

F σημαίνει «κινήσου προς τα εμπρός». Η θέση-κατάσταση της χελώνας αλλάζει ($x+d \cos \alpha$, $y+d \sin \alpha$, α)

+ σημαίνει «στρίψε προς τα δεξιά κατά μια συγκεκριμένη γωνία »

- σημαίνει «στρίψε προς τα αριστερά κατά μια συγκεκριμένη γωνία »

[σημαίνει «θυμήσου αυτό το σημείο-τοποθεσία»

] σημαίνει «επέστρεψε στο σημείο της μνήμης»

Εκτός από τα θεμελιώδη σύμβολα χρησιμοποιούμε και κάποια άλλα σύμβολα όπως τα παρακάτω:

L σημαίνει «στη θέση σου που βρίσκεσαι αποτύπωσε τον εαυτό σου» με την εντολή «σφραγίδα».

X σημαίνει «μη κάνεις τίποτα» Αυτό είναι απλά μια κράτηση θέσης

f κινήσου προς τα εμπρός ένα βήμα μήκους d χωρίς να γράφεις-ζωγραφίζεις

Στη «βίβλο» των L-systems «The algorithmic Beauty of Plants» ο Prusinkiewicz γράφει:

Τα **L-systems** επεκτείνουν συμβολοσειρές (strings) και η επέκταση αυτή βασίζεται σε ντετερμινιστικούς κανόνες (προβλέψιμο το αποτέλεσμα).

Η διαδικασία της επανεγγραφής αρχίζει από ένα αρχικό string.

Μία παραγωγή είναι γραμμένη ως $a \rightarrow v$ και σημαίνει ότι κάθε εμφάνιση του συμβόλου a στη τρέχουσα συμβολοσειρά αντικαθίσταται από το v.

Εάν δεν υπάρχει παραγωγή για ένα σύμβολο a τότε εφαρμόζεται μια εντελώς όμοια παραγωγή $a \rightarrow a$.

Το σύνολο όλων των συμβολοσειρών που προέρχονται από ένα L-system ονομάζεται formal language (φορμαλιστική γλώσσα) του L-system.

Ένα L-system λέγεται ντετερμινιστικό αν και μόνο αν για κάθε a υπάρχει ένα και μόνο ένα v έτσι ώστε $a \rightarrow v$.

Σε μια επανάληψη όλα τα σύμβολα της τρέχουσας συμβολοσειράς αντικαθίστανται ταυτόχρονα από τη κατάλληλη παραγωγή.

Παράδειγμα:

Μεταβλητές: a, b

Σταθερές : καμμία

Αξίωμα: a

Κανόνες: $a \rightarrow b, b \rightarrow aa$

Τι γίνεται στη συνέχεια

Step#0: a (axiom)

Step#1: $b//a \rightarrow b$

Step#2: $aa//b \rightarrow aa$

Step#3: $bb//a \rightarrow b$

Step#4: $aaaa//b \rightarrow aa$

Επειδή υπάρχει ακριβώς ένας κανόνας επανεγγραφής για κάθε γράμμα, οι εξαγόμενες συμβολοσειρές (strings) θα είναι ντετερμινιστικές (προβλέψιμο το αποτέλεσμα).

Επειδή κάθε γράμμα αντικαταστάθηκε ανεξάρτητα από τους γείτονες που έχει, οι κανόνες επανεγγραφής είναι ανεξάρτητοι συμφραζόμενων (context free).

1.3. Εντός των L-systems

Μια γενική L-system διαδικασία αναπτύσσεται με τις ακόλουθες ενέργειες

Ξεκινά πάντα με το αξίωμα, μια συμβολοσειρά οποιουδήποτε μήκους που περιλαμβάνει σύμβολα από το σύνολο της γλώσσας και γραμματικής L-system

Στη συνέχεια ορισμένοι κανόνες εφαρμόζονται στα μη τερματικά σύμβολα του αξιώματος, έτσι καθένας από αυτούς κάνει το σύστημα να μεγαλώνει, προσθέτοντας νέα σύμβολα στο αξίωμα –συμβολοσειρά

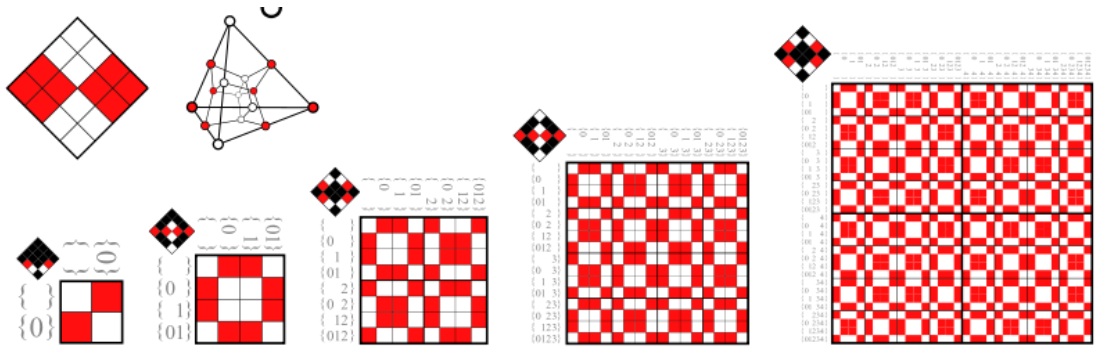
- Αφού δημιουργηθεί η τελική σειρά (εφαρμόζοντας τους κανόνες αναδρομικά) ένας αναλυτής διαβάζει και μεταφράζει τα σύμβολα σε ήχους ή γραφικά στοιχεία.
- Εξάγονται τ' αντικείμενα.

1.4. Παραδείγματα L-systems

Σε αυτή την ενότητα παρουσιάζονται ορισμένα παραδείγματα ανάλογα αν τα L-systems αναπτύσσονται σε μία διάσταση (άξονα) ή στο επίπεδο.

1.4.1. L-systems σε άξονα (πραγματικών αριθμών \mathbb{R}):

Prouhet-Thue-Morse system



Στα μαθηματικά, η Thue-Morse αλληλουχία, ή Prouhet-Thue-Morse αλληλουχία, είναι η δυαδική ακολουθία (μια άπειρη αλληλουχία του 0 και 1) που λαμβάνεται ξεκινώντας με 0 και διαδοχικά προσαρτώντας την Boolean συμπλήρωση της αλληλουχίας που ελήφθη έως τώρα. Αυτή η διαδικασία αποδίδει 0 τότε 01, 0110, 01101001, 0110100110010110, και ούτω καθεξής. Η άπειρη ακολουθία αρχίζει: [7]

0
1

01 10
10

0110 1001
1001

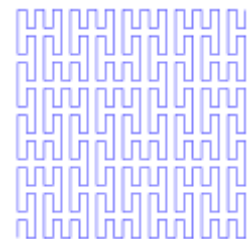
01101001100101101001011001101001

Οποιοδήποτε άλλο διατεταγμένο ζεύγος των συμβόλων μπορεί να χρησιμοποιηθεί αντί του 0 και 1 γιατί η λογική δομή της αλληλουχίας Thue-Morse δεν εξαρτάται από τα σύμβολα που χρησιμοποιούνται.[11]

1.4.2. Καμπύλες πλήρωσης χώρου (Peano's curves, Hilbert curve, Dekking's church, kolams)

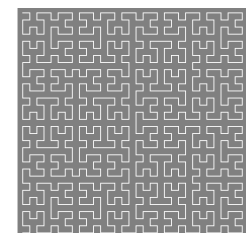
Peano's curves

Στη μαθηματική ανάλυση, η καμπύλη πλήρωσης ενός χώρου είναι μια καμπύλη της οποίας το φάσμα περιέχει ολόκληρο το τετράγωνο των 2-διαστάσεων (ή πιο γενικά ένα n - διάστατο υπερκύβο). Επειδή ο **Giuseppe Peano** (1858-1932) ήταν ο πρώτος που ανακάλυψε μια τέτοια καμπύλη, οι καμπύλες πλήρωσης χώρου 2-διαστάσεων (επίπεδο) μερικές φορές ονομάζονται **Peano** καμπύλες, αλλά η φράση αυτή αναφέρεται επίσης στην καμπύλη Peano, το συγκεκριμένο παράδειγμα μιας καμπύλης πλήρωσης χώρου που βρέθηκε από τον Peano.[12]

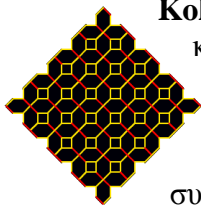


Hilbert curve

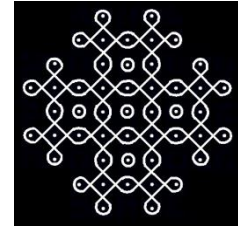
Μια καμπύλη **Hilbert** (επίσης γνωστή ως καμπύλη πλήρωσης χώρου Hilbert) είναι μια συνεχής φράκταλ πλήρωσης χώρου καμπύλη που περιγράφηκε για πρώτη φορά από το Γερμανό μαθηματικό **David Hilbert** το 1891, ως μια παραλλαγή της καμπύλης πλήρωσης χώρου που ανακαλύφθηκε από τον Giuseppe Peano το 1890.[8]



kolams



Kolam είναι μια μορφή ζωγραφικής που αποτελεί καμπύλη πλήρωσης χώρου. Τα υλικά που χρησιμοποιούνται είναι σκόνη ρυζιού, σκόνη κιμωλίας, λευκή σκόνη κ.ά. Κάποιες από τις σκόνες που χρησιμοποιούνται μπορεί να είναι συνθετικές και κάποιες φυσικές. Είναι μια τεχνική που συναντάμε στην Ινδία, στην Ινδονησία, τη Μαλαισία, την Ταϊλάνδη και σε μερικές άλλες χώρες της Ασίας. Ένα Kolam είναι ένα γεωμετρικό σχέδιο που αποτελείται από καμπύλα βρόχων, γύρω από ένα σχέδιο πλέγματος από κουκκίδες. Στη Νότια Ινδία, είναι διαδεδομένο και ασκείται από γυναίκες μέλη της οικογένειας ινδουιστών μπροστά από τα σπίτια τους.[10]

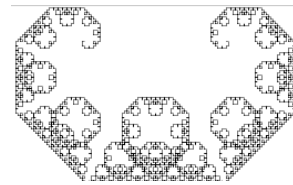


Τα έργα [cycar](#) και [sierpinski triangle](#) στο scratch 2.0 αποτελούν παραδείγματα πλήρωσης χώρου.

1.4.3. median space-filling curves (Lévy C curve, Harter-Heighway dragon curve, Davis-Knuth terdragon)

Lévy C curve

Στα μαθηματικά, η καμπύλη Lévy C είναι ένα αυτοπαρόμοιο φράκταλ που περιγράφηκε για πρώτη φορά, από τον Ernesto Cesàro το 1906 και τον Farber το 1910, αλλά τώρα φέρει το όνομα του Γάλλου μαθηματικού Paul Pierre Lévy, ο οποίος ήταν ο πρώτος που περιέγραψε τις ιδιότητες αυτοομοιότητας, και δημιούργησε μια γεωμετρική κατασκευή ότι έχει παρόμοιες ιδιότητες με την καμπύλη Koch.

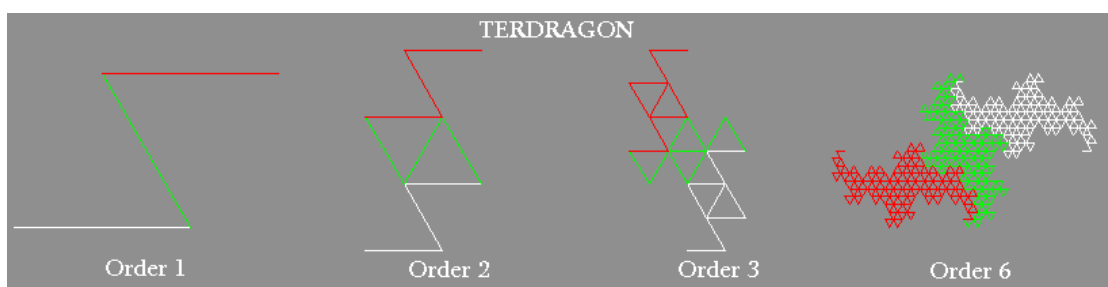


Harter-Heighway dragon curve, Davis-Knuth terdragon

Dragon curve (Heighway dragon, Terdragon, twindragon, Levy dragon) είναι οποιοδήποτε μέλος της οικογένειας αυτοπαρόμοιων fractal curves που μπορούμε να προσεγγίσουμε με αναδρομικές μεθόδους όπως τα L-systems.

Το dragon curve αρχικά εφευρέθηκε από τη NASA και συγκεκριμένα από τους φυσικούς John Heighway, Bruce Banks, και William Harter. Πήρε το όνομα του πρώτου και ονομάστηκε Heighway dragon. Το περιέγραψε ο Martin Gardner στην Επιστημονική Αμερικανική στήλη “Mathematical Games” το 1967. Πολλές από τις ιδιότητές δημοσιεύτηκαν από τους Chandler Davis και Donald Knuth.

Ενώ μελετούσαν το Heighway dragon ο Davis and ο Knuth εφηύραν το δικό τους



spacefilling curve (καμπύλη πλήρωσης χώρου) το οποίο ονόμασαν Terdragon. Το L-system του Terdragon είναι:

1.4.4. Tiling

Περιοδικό και απεριοδικό tiling

Οι tiling πλακοστρώσεις είναι απλά παραδείγματα απεριοδικής πλακόστρωσης ενός επιπέδου. Ένα tiling είναι η κάλυψη ενός επιπέδου με πλακάκια χωρίς επικαλύψεις ή κενά. Τα πλακάκια έχουν συνήθως ένα πεπερασμένο αριθμό σχημάτων που ονομάζονται prototiles και μια σειρά από αυτά εισάγει τη επικάλυψη της επιφάνειας, εάν υπάρχει επικάλυψη της επιφάνειας, με τη χρήση μόνο πλακών σύμφωνες με τα prototiles. Τα tiling μπορούν να εκφραστούν ως L-systems αφού παράγονται με αναδρομικές διαδικασίες.

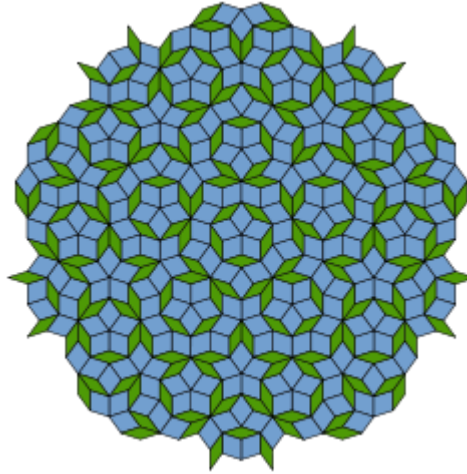
Τα πιο γνωστά tiling (π.χ. με τετράγωνα και τρίγωνα) είναι **περιοδικού τύπου επικαλύψεις**. Ένα τέλειο αντίγραφο του tiling μπορεί να ληφθεί μετατοπίζοντας όλα τα πλακίδια κατά μια σταθερή απόσταση και δεδομένη κατεύθυνση. Μια τέτοια μετατόπιση ονομάζεται **περίοδος του tiling** και πιο ανεπίσημα αυτό σημαίνει ότι μια πεπερασμένη περιοχή του tiling επαναλαμβάνει τον εαυτό της περιοδικά σε τακτά χωρικά διαστήματα. Εάν ένα tiling δεν έχει περίοδο ονομάζεται **απεριοδικό**. Ένα σύνολο από prototiles λέγεται απεριοδικό εάν επικαλύπτει το επίπεδο αλλά μια τέτοια επικάλυψη είναι μη περιοδική από απεριοδικά σύνολα prototiles τα οποία ονομάζονται απεριοδικές επικαλύψεις.

Penrose Tiling (επικάλυψη επιφάνειας)

Η επικάλυψη Penrose μιας επιφάνειας (tiling) είναι μια μη περιοδική κάλυψη επιφάνειας που παράγεται από ένα σύνολο πρωτότυπων πλακών.

Τα Penrose tiling πήραν το όνομά τους από το μαθηματικό και φυσικό Roger Penrose ο οποίος ανακάλυψε αυτές τις σειρές στη δεκαετία του 70. **Η απεριοδικότητα των πρωτότυπων πλακών σημαίνει ότι η μετατόπιση ενός αντιγράφου πλακών Penrose ποτέ δεν θα ταίριαζε με το πρωτότυπο** (που σημαίνει ότι **στερείται συμμετρίας μετατόπισης**).

Μια πλακόστρωση μπορεί να κατασκευαστεί έτσι ώστε να παρουσιάζει και τα δύο συμμετρία ανάκλασης και πενταπλή περιστροφική συμμετρία όπως φαίνεται στο σχήμα:



Ένα Penrose tiling εκτός του ότι στερείται συμμετρίας μετατόπισης έχει και άλλες αξιοσημείωτες ιδιότητες όπως

- Είναι **αυτοπαρόμοιο** έτσι ώστε τα ίδια μοτίβα εμφανίζονται σε μεγαλύτερες και μεγαλύτερες κλίμακες Έτσι η πλακόστρωση μπορεί να επιτευχθεί μέσω του πληθωρισμού (ή αποπληθωρισμού) και κάθε πεπερασμένο κομμάτι της πλακόστρωσης επαναλαμβάνεται απείρως πολλές φορές.
- Ένα Penrose tiling είναι ένα **quasicrystal** που υλοποιείται σαν μια φυσική δομή, που παράγει περίθλαση Bragg και το διαθλασιόγραμμά της αποκαλύπτοντας και τα δύο, **τη πενταπλή συμμετρία** αλλά και τη **μεγάλη εμβέλεια** του συστήματος.

1.4.5. Δέντρα φυτά και τα παρόμοια

Φυτά-Διακλάδωση

Τα περισσότερα L-systems (αν όχι όλα) εμφανίζουν δομές διακλάδωσης.

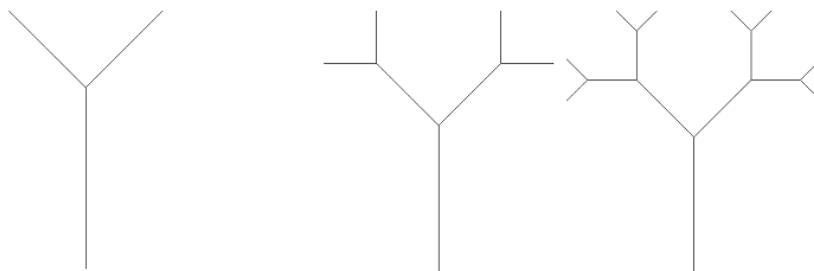
Στη περίπτωση των γραφικών εφαρμογών μια L-system γλώσσα περιλαμβάνει σύμβολα για να οδηγήσει τη χελώνα στην οθόνη (σε κινήσεις, περιστροφές κ.λ.π. όπως και στη γλώσσα Logo) και συνήθως τη δυνατότητα να δημιουργήσει κλαδιά έτσι ώστε να μπορούμε να χρησιμοποιήσουμε τα L-systems για να προσομοιάσουμε για παράδειγμα την ανάπτυξη φυτών και δέντρων .

Για να εκπροσωπήσουν κλάδους με την έννοια ενός string βασισμένο στα L-systems ο Prusinkiewicz παρουσίασε τα bracketed L-systems τα οποία χρησιμοποιούν τις αγκύλες [και] ως σύμβολα οριοθέτησης των κλάδων.

Ένα απλό δυαδικό δέντρο που φαίνεται πιο κάτω παράγεται από ένα αρχικό string F από το κανόνα $F \rightarrow F[+F][-F]$.

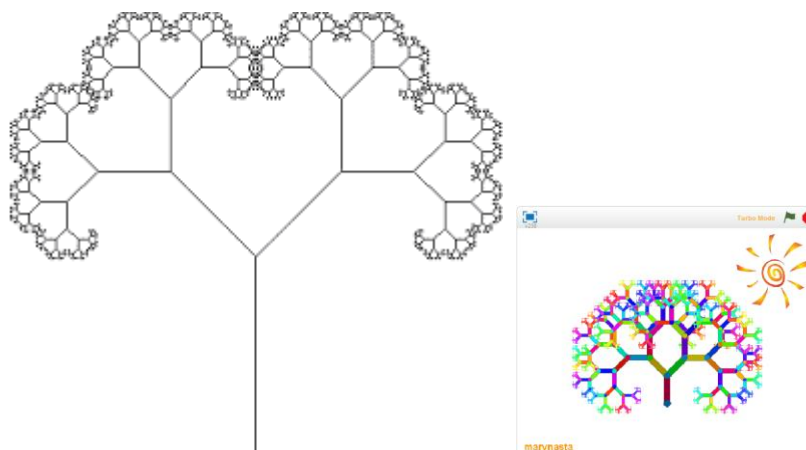
Ένα δέντρο είναι ένας κορμός με δύο κλαδιά... Κάθε κλαδί αποτελεί το κορμό ενός άλλου δέντρου με δυο κλαδιά κ.ο.κ.

Εδώ παρουσιάζεται ο αναδρομικός ορισμός ενός δέντρου (αναδρομικό σημαίνει ότι ορίζεται με βάση τον εαυτό του).



Ένα δέντρο είναι ένας κορμός με δύο πιο κοντά κλαδιά που βγαίνουν από τη κορυφή του κορμού σχηματίζοντας μεταξύ τους ορθή γωνία.

Ένα κλαδί είναι ένα δέντρο. Μετά από αρκετά βήματα θα πάρουμε το παρακάτω σχήμα:



[2], [5]

Ένα παράδειγμα δέντρου είναι το ακόλουθο:



Μεταβλητές (variables) :	X F
Σταθερές (constants) :	+ - []
Αξίωμα (start) :	X
Κανόνες (rules) :	$X \rightarrow F-[[X]+X]+F[+FX]-X$ $F \rightarrow FF$
Γωνία (angle) :	25°

Εδώ το F σημαίνει «κινήσου προς τα εμπρός» Το - σημαίνει «στρίψε αριστερά 25° » Το + σημαίνει «στρίψε δεξιά 25° » Το αξίωμα είναι X. Το X δεν αντιστοιχεί σε κάποια ενέργεια σχεδίασης και χρησιμοποιείται για να ελέγχει την

εξέλιξη της καμπύλης. Το [χρησιμοποιείται για να αποθηκεύει τις τρέχουσες τιμές θέσης και γωνίας, οι οποίες ανακτώνται όταν το] εκτελείται.

Με την ερμηνεία «γραφικά χελώνας» η χελώνα ακολουθεί ένα κλαδί για ένα διάστημα και κατόπιν επιστρέφει στη θέση όπου άρχισε τη διακλάδωση. Μπορούμε να εμφωλεύσουμε ένα κλαδί μέσα στο κλαδί και να δημιουργήσουμε έτσι πολλαπλά κλαδιά αλλά δεν μπορούμε να πηδήξουμε ανάμεσα σε δύο διαφορετικά κλαδιά. Στο τέλος του κλαδιού τοποθετούνται στη στοίβα η τρέχουσα θέση και γωνία και ανακτώνται τα δεδομένα της αρχής του κλαδιού έτσι ώστε η χελώνα μπορεί να πάει πίσω στη ρίζα.

[11],[5],[4]

1.5. Είδη L-systems

Τα L-systems είναι ντετερμινιστικά συστήματα που σημαίνει ότι με οποιαδήποτε γραμματική το αποτέλεσμα είναι πάντα προβλέψιμο. Γι αυτό το λόγο πολλές βελτιώσεις έχουν προστεθεί με αποτέλεσμα διαφορετικά είδη L-systems[5]

1.1.1. Πρότυπα L-systems

Είναι ντετερμινιστικά συστήματα αφού **εφαρμόζεται ένας κανόνας για κάθε μη τερματικό σύμβολο**. Κάθε μη τερματικό σύμβολο αντικαθίσταται από μια συμβολοσειρά που καθορίζεται από τους κανόνες και τα τερματικά σύμβολα.

Στο παρακάτω πρακτικό παράδειγμα θα ορίσουμε πρώτα το αξίωμα και τους κανόνες.

a , b μη τερματικά σύμβολα
+ , - , F τερματικά σύμβολα (τα τερματικά εμφανίζονται μόνο στο δεύτερο μέλος των κανόνων)

a το αξίωμα (μπορεί να είναι οτιδήποτε στο σύνολο της γλώσσας)
a=a+b κανόνας 1
b=aFa-a κανόνας 2

Τα a και b σημαίνουν «ζωγράφισε κάτι» (μια γραμμή, ένα 3D αντικείμενο, οτιδήποτε) Το F σημαίνει μπροστά, Το + στρίψε δεξιά και το - στρίψε αριστερά. Επαναλαμβάνουμε το σύστημα δύο φορές.

Αξίωμα:	a	χρήση του κανόνα 1 :	a=a+b
1η επανάληψη:	a+b	χρήση του κανόνα 1:	a=a+b
μετά + και μετά κανόνας 2:			b=aFa-a
2η επανάληψη:	a+b+afa-a		... κ.λ.π.

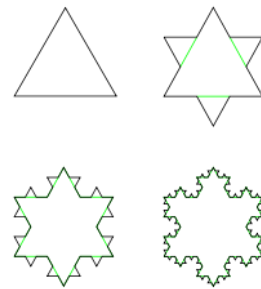
Ο αναλυτής θα διαβάσει τη τελική συμβολοσειρά και θα κινηθεί, θα περιστραφεί και θα ζωγραφίσει σύμφωνα με αυτή. Τα L-systems είναι προσिता γιατί με απλούς κανόνες μπορείς να πάρεις πολύ πολύπλοκα και fractal αντικείμενα.

Ένα χαρακτηριστικό παράδειγμα αποτελεί η καμπύλη Koch

Ο Niels Fabian Helge von Koch (1870-1924) ήταν ένας Σουηδός μαθηματικός που έδωσε το όνομά του στο διάσημο fractal γνωστό ως “Koch snowflake”[9] Η νιφάδα χιονιού Koch γνωστή και ως αστέρι Koch ή νησί Koch είναι μια μαθηματική καμπύλη και μια από τις πρώτες fractal καμπύλες που έχουν περιγραφεί. Βασίζεται στη καμπύλη που εμφανίστηκε το 1904 σε έγγραφο με τίτλο :“On a continuous curve without tangents, constructible from elementary geometry” [9]

Η καμπύλη του «von Koch» γνωστό αντικείμενο fractal και μπορεί εύκολα να παραχθεί με L-systems.(βλ. 3.5.2 “The Koch snowflake”)

variables : F
 constants : + - []
 start : F++F++F
 rules : F → F-F++F-F
 angle : 60°



Μπορούμε να κάνουμε όσες επαναλήψεις θέλουμε αρκεί να προσέξουμε το μέγεθος του συστήματος.

1.1.2. Στοχαστικά L-systems

Οι κανόνες εφαρμόζονται τυχαία έτσι ώστε να καθιστούν τα πρότυπα L-systems λιγότερα ντετερμινιστικά.

Σ’ ένα στοχαστικό σύστημα εμείς θα τροποποιήσουμε τους κανόνες: Αντί για ένα κανόνα για κάθε μη τερματικό σύμβολο μπορούμε να χρησιμοποιήσουμε 2 ή περισσότερους κανόνες οι οποίοι διαλέγονται συνήθως από κάποια παράμετρο βαρύτητας ή τυχαίας αξίας. Να πώς μπορούμε να τροποποιήσουμε την πρώτη γραμματική.[5]

a=a+b	κανόνας 1a	50% πιθανότητα
a=a-bb	κανόνας 1b	50% πιθανότητα
b=aFa-a	κανόνας 2a	30% πιθανότητα
b=aFFba	κανόνας 2b	70% πιθανότητα

Έτσι το a μπορεί να παράγει a+b ή a-bb με την ίδια πιθανότητα. Το b έχει μια μικρή πιθανότητα να παράγει aFa-a και μια μεγάλη πιθανότητα να παράγει aFFba. Αυτή η μέθοδος μπορεί να παράγει πολλούς και διαφορετικούς συνδυασμούς από σύμβολα οπότε και διαφορετικά αντικείμενα. Τα στοχαστικά συστήματα χρησιμοποιούνται για να προσομοιάσουν τη μεγάλη ποικιλία και αταξία των σχηματισμών της φύσης. Για παράδειγμα αν χτίσουμε ένα (απλό) L-system δέντρο θα έχει σαν αποτέλεσμα πάντα στο ίδιο κλάδο παράταξη και στον ίδιο σχηματισμό. Με τους στοχαστικούς κανόνες μπορούμε πάντα να παράγουμε διαφορετικά δέντρα.

1.1.3. Παραμετρικά L-systems

Οι κανόνες που επιλέγονται εξαρτώνται από παραμέτρους των οποίων τα σύμβολα περνούν από τον έναν στον άλλο. Κάθε σύμβολο του αλφάβητου έχει μια λίστα παραμέτρων που συνδέονται μ' αυτό.[5]

Τα παραμετρικά L-systems αποτελούν επέκταση της αρχικής έννοιας των L-systems με τη συμμετοχή αριθμητικών παραμέτρων με τα σύμβολα ν' αντιπροσωπεύουν τα συστατικά των φυτών. Αυτό επιτρέπει την εύκολη ποσοτικοποίηση των γεωμετρικών χαρακτηριστικών του μοντέλου.

Το μοντέλο ανάπτυξης ελέγχεται από ένα μηχανισμό παραγωγής με πληροφορίες που περνούν από το γονέα στο παιδί. Αυτός ο μηχανισμός συνδυάζεται με κάποια άλλα μοντέλα με ενδογενή αλληλεπίδραση όπου οι πληροφορίες ρέουν μέσω μιας αυξανόμενης δομής.

Παράδειγμα:

Κάθε σύμβολο έχει μια λίστα από παραμέτρους που σχετίζονται με αυτό. Ένα σύμβολο σε συνδυασμό με τη λίστα παραμέτρων του ονομάζεται module και μια συμβολοσειρά είναι μια σειρά από modules. (π.χ. $\mathbf{a(0,1)[b(0,0)]a(1,2)}$)

$\mathbf{a(x,y) : x = 0 \rightarrow a(1, y+1)b(2,3)}$ κανόνας παραγωγής (ο κανόνας εφαρμόζεται όταν ικανοποιείται ο όρος $x = 0$)

Στο τμήμα τροποποίησης του κανόνα παραγωγής οι παράμετροι καθώς και ολόκληρα τα modules μπορεί να επηρεαστούν.

Το module $b(x,y)$ προστίθεται στη συμβολοσειρά με αρχικές παραμέτρους (2,3). Επίσης οι παράμετροι του ήδη υπάρχοντος module έχουν τροποποιηθεί σύμφωνα με το κανόνα παραγωγής

Έτσι το $a(0,2)$ παράγει:

$$\mathbf{a(0,2) \rightarrow a(1,3)b(2,3)}$$

1.1.4. L-systems εξαρτώμενα από τα συμφραζόμενα (context-sensitive)

Οι κανόνες επιλέγονται ανάλογα με τους γείτονες του ενεργού συμβόλου είτε από εξωτερικά δεδομένα. Μ' αυτό τον τρόπο τα L-systems μπορεί ν' αντιδράσει σε καιρικές συνθήκες, βαρύτητα κ.λ.π.

Ο καθαρός φορμαλισμός δεν μπορεί ν' αντιμετωπίσει σωστά την εκπροσώπηση μεγάλης ποικιλίας της αρχιτεκτονικής των φυτών και τη συμπεριφορά ανάπτυξής τους. Υπάρχει μια καθολική ευαισθησία στην ανάπτυξη των φυτών που επηρεάζεται:

- από τις κατεύθυνση της βαρύτητας (γεωτροπισμός) και του φωτός (φωτοτροπισμός)
- Από τη πυκνότητα. Τα μέρη των φυτών δεν μπορούν να διαπεράσουν εμπόδια όπως π.χ. άλλα μέρη φυτών ή τοίχους αντ' αυτού θα πρέπει να παρακάμψουν το εμπόδιο με αποκλίνουσες κινήσεις ανάπτυξης.
- η εξάρτηση από το φως. Εκτός από τον προσανατολισμό του φωτός, η απλή διαθεσιμότητά του είναι σημαντική. Αυτό επηρεάζεται από την επισκίαση από άλλα μέρη του φυτού και εμπόδια.

- Τοπικές μεταβλητές μπορούν να συνδεθούν με τα στοιχεία της αναπτυσσόμενης δομής. Αυτές μπορεί να εκπροσωπούν τις τοπικές συνθήκες όπως οι θρεπτικές λιμνούλες.

Η παραγωγή ενός κανόνα εφαρμόζεται σ' ένα συγκεκριμένο σύμβολο μόνο εάν αυτό το σύμβολο έχει συγκεκριμένους γείτονες.[5]

Στο παρακάτω παράδειγμα θα εξετάσουμε μόνο το δεξιό γείτονα (> σημαίνει το επόμενο δεξιό σύμβολο) όπου μόνο το a έχει κανόνες εξαρτώμενους από τα συμφραζόμενα:

a , b μη τερματικά σύμβολα
+ , - , F τερματικά σύμβολα

a το αξίωμα

a(>null or [or]) = a+b	κανόνας 1a
a (>+)= a[+b]	κανόνας 1b
a (>F)= ab	κανόνας 1c
a (>b)= null	κανόνας 1d
b=aFa-a	κανόνας 2

Αυτό περιγράφει ένα αντικείμενο με μια περίτεχνη δυναμική Εάν ένα a δεν έχει τίποτα στα δεξιά του ή έχει [ή] τότε δίνει a+b Εάν το a έχει στα δεξιά του b τότε το a εξαφανίζεται κ.λ.π.

1.1.5. Χρονικά L-systems

Οι κανόνες μπορούν να εφαρμοστούν εξαρτώμενοι από μεταβλητές χρόνου (όπως επαναλήψεις , βρόχοι, ηλικία κ.λ.π.)

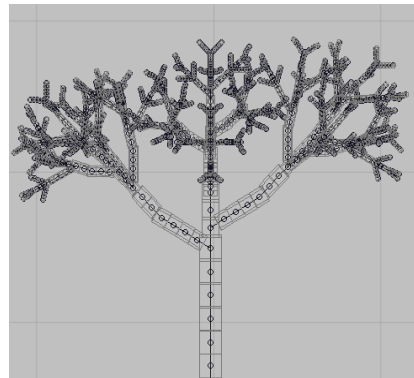
Σ' ένα χρονικό L-system τα σύμβολα μπορεί να έχουν διαφορετικούς κανόνες που εξαρτώνται από την ηλικία, ή οι κανόνες μπορεί ν' αλλάζουν σε κάθε επανάληψη ή το σύμβολο μπορεί να πεθαίνει μετά από κάποιο χρονικό διάστημα.

Παρακάτω θα δούμε ένα γενικό παράδειγμα όπου μόνο το a χρονικούς κανόνες μέσα σ' ένα σύστημα με 4 επαναλήψεις:

a , b μη τερματικά σύμβολα
+ , - , F τερματικά σύμβολα

a	το αξίωμα
a = a+b	κανόνας που ισχύει στη 1η επανάληψη
a = ab	κανόνας που ισχύει στη 2η επανάληψη
a = a[+b]	κανόνας που ισχύει στη 3η επανάληψη
a = ab	κανόνας που ισχύει στη 4η επανάληψη

b=aFb-a κανόνας που ισχύει για όλες τις επαναλήψεις



Γιατί είναι τα L-systems καλά;

Είναι δύσκολο να ελέγξουμε τη πολυπλοκότητα των L-systems που μπορούν να δημιουργηθούν και να ζωγραφίσουμε κάτι που έχουμε στο μυαλό μας χωρίς να χάσουμε τη χαοτική συμπεριφορά που είναι το σημαντικό στη παραγωγή ενός σχεδίου.

Τα L-systems μας ελκύουν λόγω των πολύπλοκων και απρόβλεπτων σχημάτων που μπορούν να δημιουργήσουν. Αλλά αυτή η πολυπλοκότητα είναι ουσιαστικά προβλέψιμη αφού τα L-systems είναι ντετερμινιστικά.

Οι καλλιτέχνες που χρησιμοποιούν στοχαστικά και αισθητικού περιεχομένου L-systems βελτιώνουν το επίπεδο της αισθητικής τους δημιουργικότητας και έρχονται πιο κοντά με τη ποικιλότητα των σχημάτων που θαυμάζουμε στη φύση.

Η γραμματική όμως μπορεί εύκολα να γίνει πολύπλοκη για να τη καταλάβεις και να την ελέγξεις. Γι αυτό εμείς πειραματιζόμαστε με αλληλεπιδραστικά τα L-systems με interfaces και φυσικές υπολογιστικές συσκευές. Αυτό μπορεί να οδηγήσει σε αληθινά παραγωγικά συστήματα όπου αυτό που αναδύεται δεν είναι ψεύτικοι ή τυχαίοι αριθμοί αλλά κάτι που είναι πέρα από τον υπολογιστή και τους χρήστες του.

2. Προγραμματισμός με το scratch

2.1. Προγραμματιστικό πολυμεσικό περιβάλλον



Το **Scratch** είναι μια δυναμική γλώσσα προγραμματισμού που προσφέρει ένα μαθησιακό περιβάλλον για αρχάριους και δίνει αποτελέσματα, χωρίς να απαιτείται πρώτα εκμάθηση σωστής σύνταξης. Όντας δυναμική, επιτρέπει αλλαγές του κώδικα ακόμη και κατά τη διάρκεια της εκτέλεσης των προγραμμάτων. [14],[15]

Δημιουργήθηκε από το **MIT Media Lab** (ερευνητικό εργαστήριο) και στοχεύει να δώσει κίνητρα για περαιτέρω μάθηση μέσω του παιχνιδιού, πειραματισμού και δημιουργίας έργων όπως διαδραστικά κινούμενα σχέδια, παιχνίδια κ.λ.π.

Η πρώτη έκδοση του **Scratch** αναπτύχθηκε το 2006 από την ομάδα του **Lifelong Kindergarten** με επικεφαλής τον Mitchel Resnick στο MIT Media Lab.

Το όνομα **Scratch** προέρχεται από τη τεχνική των **DJ's (scratching)** που βασικό χαρακτηριστικό της είναι η επαναχρησιμοποίηση και ένωση μουσικών κομματιών για να δημιουργηθεί μια μουσική σύνθεση. Στο **Scratch** όλα τα αλληλεπιδραστικά αντικείμενα, γραφικά και ήχοι μπορούν εύκολα να εισαχθούν σε ένα νέο πρόγραμμα και να συνδυαστούν με νέους τρόπους. Έτσι οι αρχάριοι μπορούν να λάβουν γρήγορα αποτελέσματα και αποκτούν κίνητρο να προσπαθήσουν περαιτέρω. [14],[15]

Το **Scratch** είναι ένα προγραμματιστικό πολυμεσικό περιβάλλον. Είναι μια πρακτική μάθησης μέσω σχεδιασμού και υλοποίησης. Ανήκει στη κατηγορία των ανοικτών μικρόκοσμων, εκπαιδευτικών λογισμικών, που θεωρούνται από τα πιο σημαντικά. Οι μικρόκοσμοι αποτελούν ανοικτά περιβάλλοντα στα οποία υπάρχουν μερικές βασικές οντότητες όπως η γάτα scratch (ή η χελώνα lego, ή το ευκλείδειο επίπεδο της Γεωμετρίας) και ο χρήστης μπορεί να δημιουργήσει νέες οντότητες και νέα αντικείμενα. Να δημιουργήσει επίσης σχέσεις μεταξύ των αντικειμένων και να μελετήσει τις αλληλεπιδράσεις τους. Σημαντικό πλεονέκτημα σε σχέση με άλλα εκπαιδευτικά λογισμικά είναι ότι μπορούμε να κατασκευάσουμε όσα κουμπιά και επιλογείς θέλουμε χωρίς περιορισμούς. [19]

Στην ίδια ίσως κατηγορία εντάσσονται και τα προϊόντα της εκπαιδευτικής ρομποτικής ενός κλάδου ο οποίος έχει γνωρίσει μεγάλη ανάπτυξη τα τελευταία χρόνια. Ιδιαίτερα η εκπαιδευτική ρομποτική που συνδυάζει κατασκευή και προγραμματισμό αυτόματων μηχανισμών και ρομπότ εξ αιτίας της φύσης της αποτελεί έναν πολύ ιδιαίτερο κλάδο εφαρμογής των ΤΠΕ στην εκπαίδευση καθώς συνδυάζει πολλές επιστημονικές περιοχές (μαθηματικά, προγραμματισμό, μηχανολογία, φυσική κ.λ.π. [19]

Το **Scratch** χρησιμοποιείται παγκοσμίως σε διάφορα σχολεία και εκπαιδευτικούς οργανισμούς. Ο δικτυακός τόπος του Scratch έχει αναπτύξει μια κοινότητα από αρχόμενους προγραμματιστές, μαθητές, δασκάλους και ερασιτέχνες, που αλληλοπαρακινούνται να αναπτύξουν την δημιουργικότητά τους και τις προγραμματιστικές δεξιότητές τους. Ένα από τα forum στον ιστότοπο του Scratch είναι αφιερωμένο σε συζητήσεις μεταξύ εκπαιδευτικών. [15]

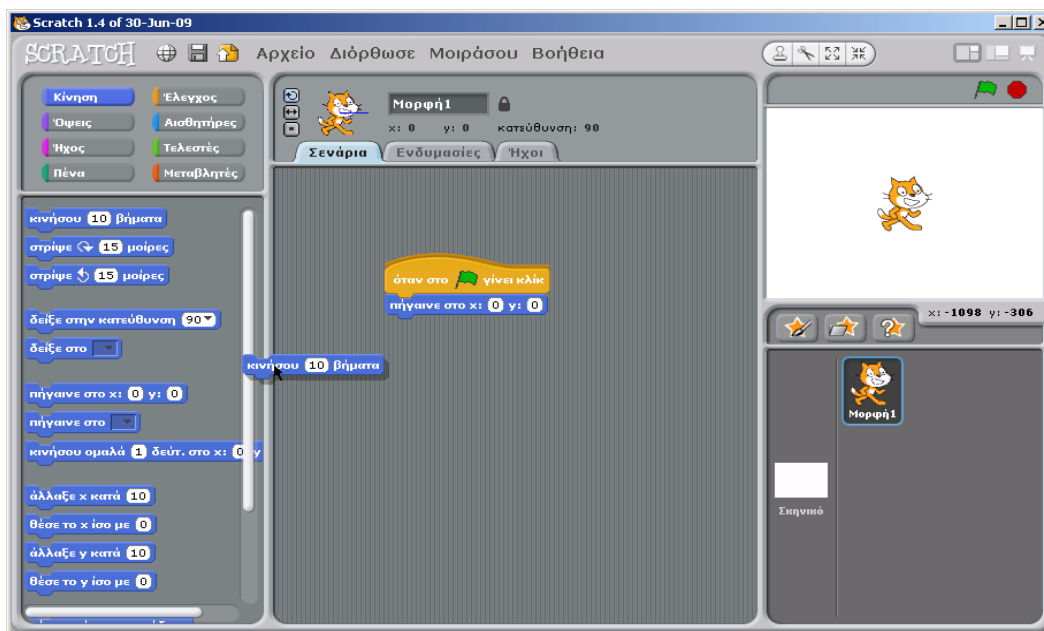
Το σλόγκαν του Scratch είναι "**Φαντάσου · Φτιάξε · Μοιράσου**". Η έμφαση στο μοίρασμα είναι σημαντικό μέρος της παιδαγωγικής για το Scratch: τα προγράμματα δεν θεωρούνται μαύρα κουτιά, αλλά αντικείμενα που μπορούν να αναμιχθούν για τη δημιουργία νέων έργων. Ο μόνος τρόπος να γίνει ένα πρόγραμμα διαθέσιμο για χρήση είναι να δοθεί ο πηγαίος κώδικας του.

Έχει σαν **στόχο** τη διδασκαλία των εννοιών του προγραμματισμού σε παιδιά και εφήβους και να τους επιτρέψει να δημιουργήσουν παιχνίδια, βίντεο και μουσική.[15] Αλλά αποτελεί και σημαντικό εργαλείο που μπορεί να βοηθήσει σημαντικά και στην εκπαίδευση των φυσικών επιστημών αφού βοηθά στη κατανόηση εννοιών όπως π.χ. ο σχηματισμός ειδώλων από [κάτοπτρα](#), αλλά και των μαθηματικών εισάγοντας έννοιες όπως της μεταβλητής, του αλγόριθμου της επανάληψης κ.λ.π. Π.χ. [κανονικά πολύγωνα](#)

Υποστηρίζεται η άποψη ότι άτομα που χρησιμοποιούν το scratch σε νεαρή ηλικία αναπτύσσουν γερό θεμέλιο γνώσης που μπορεί να βοηθήσει στη προετοιμασία για τη χρήση των γλωσσών προγραμματισμού σε υψηλότερο επίπεδο.

Το **scratch** επιτρέπει τη κατασκευή και τη δοκιμή ως επί το πλείστον μέσω της διαδικασίας της αφήs καθώς η πρώτη προτεραιότητα των δημιουργών ήταν να γίνει η γλώσσα και το περιβάλλον αυτονόητο και να γίνεται εύκολη εκμάθησή τους από παιδιά που δεν είχαν προηγούμενη προγραμματιστική εμπειρία [15]

Το οπτικό περιβάλλον και περιβάλλον αφήs του scratch επιτρέπει στα παιδιά να εξερευνήσουν μέσω της διαδικασίας «σύρε και άφησε» **blocks** εντολών, συνθηκών (με παραμέτρους) και συνεπειών (δράσεις) σε επιλεγμένους παράγοντες (πράκτορες) που ονομάζονται **sprites** και υπόβαθρα που ονομάζονται **stages**. Μας δίνει τη δυνατότητα να συμπεριλάβουμε παιδιά που δεν είναι σε θέση να γράψουν συντακτικά σωστές δομές ούτε να τις διαβάσουν. Ακόμα οπτικά ομαδοποιημένα blocks μπορούν να ελεγχθούν με ένα κλικ και μπορούν εύκολα να αντικατασταθούν με διαφορετικά blocks – εντολές με σκοπό να τροποποιήσουν ή να δημιουργήσουν νέες εκδόσεις έργων.[15]



Το περιβάλλον διεπαφής του scratch με το χρήστη, είναι χωρισμένο σε αρκετά τμήματα. Στα αριστερά βρίσκεται η παλέτα με τα blocks εντολών, στο μέσον το

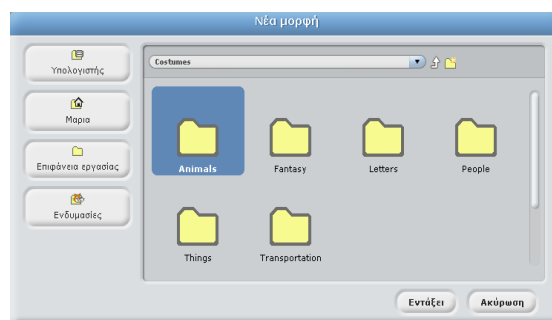
τρέχον sprite και η περιοχή σύνθεσης του προγράμματος-σεναρίου και δεξιά η οθόνη (υπόβαθρα) και η λίστα των sprites. [15]

Το **Scratch** προσφέρει ευκαιρίες **συνεργατικής μάθησης**. Για παράδειγμα μικροί μαθητές μπορούν να δημιουργήσουν έργα με τους γονείς τους, με τα μεγαλύτερα αδέρφια τους, ή με τους συμμαθητές τους αλλά και φοιτητές χρησιμοποιούν το Scratch σε ορισμένες εισαγωγικές τάξεις της επιστήμης των υπολογιστών συμπεριλαμβανομένων των ηλεκτρονικών υπολογιστών εισαγωγικής τάξης του Harvard. [14]



Το καινοτόμο στη γλώσσα **Scratch** είναι ότι δεν πληκτρολογούμε το κώδικα. Η παλέτα με τα blocks έχει κομμάτια κώδικα (τα blocks ή εντολές) που μπορούν να συρθούν. Η γλώσσα δομείται όπως ένα παζλ σύροντας, στην περιοχή **scripts**, τα blocks για να γίνει η σύνθεση του προγράμματος-σεναρίου.

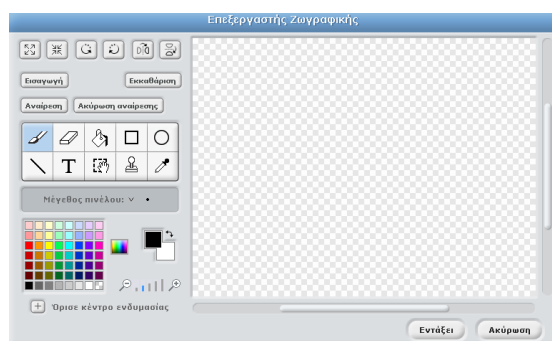
Τα blocks ή εντολές έχουν σχεδιαστεί με τέτοιο τρόπο ώστε να συνδέονται μεταξύ τους μόνο όταν ο συνδυασμός τους είναι έγκυρος και έχει νόημα. Τοποθετούνται σε στοιβές, όπως φαίνεται στην εικόνα, για τη σύνθεση ενός σεναρίου. Τα blocks της παλέτας είναι οργανωμένα σε οκτώ ομάδες : κίνηση, όψεις, ήχος, πένα, έλεγχος, αισθητήρες, τελεστές και μεταβλητές. Διαφορετικά είδη των blocks έχουν διαφορετικά χρώματα και σχήματα. Σημαντικό πλεονέκτημα του scratch είναι η δυνατότητα που μας δίνει να εξετάσουμε το αποτέλεσμα οποιασδήποτε εντολής αρκεί να πατήσουμε διπλό κλικ επάνω σε αυτήν.



Όσον αφορά τα sprites μπορούμε να χρησιμοποιήσουμε τις μορφές που υπάρχουν στο ενσωματωμένο αρχείο του **Scratch** και εμφανίζονται όπως στο διπλανό παράθυρο όταν επιλέξουμε το μεσαίο κουμπί που βρίσκεται κάτω από την οθόνη του περιβάλλοντος **Scratch**. Αν επιλέξουμε το τρίτο κουμπί θα εμφανιστεί αυτόματα μια μορφή «έκκληξη». Πολύ σημαντικό



είναι επίσης ότι μπορούμε επιλέγοντας το πρώτο κουμπί, θα εμφανιστεί το παράθυρο που φαίνεται στην εικόνα, ένα περιβάλλον ζωγραφικής ή επεξεργασίας εικόνας-φωτογραφίας και εκεί μπορούμε να δημιουργήσουμε - ζωγραφίσουμε το δικό μας sprite[16].



Η καινούρια αυτή γλώσσα καθιστά εύκολη τη κατασκευή αλληλεπιδραστικών ιστοριών, παιχνιδιών, animations (κινούμενα σχέδια), simulations (προσομοιώσεις), παιχνιδιών, τέστ αξιολόγησης ακόμα και ψηφιακής τέχνης χρησιμοποιώντας μαθηματικές εξισώσεις. Μας δίνει ακόμα τη δυνατότητα να ζωγραφίσουμε ή να

συνθέσουμε μουσική συνδυάζοντας κατάλληλα τις εντολές ήχων. Να γίνουμε δημιουργοί πολλών και διαφορετικών έργων.

Για να δημιουργήσουμε ένα πρόγραμμα που θα επιλύει ένα πρόβλημα θα πρέπει αφού πρώτα το μελετήσουμε και γνωρίσουμε τους νόμους ή κανόνες στους οποίους υπακούει, θα διασπάσουμε το πρόβλημα σε μικρότερα υποπροβλήματα τα οποία είναι πιο εύκολο να επιλύσουμε.

Στη συνέχεια

- Θα επιλέξουμε το υπόβαθρο (background) που διαδραματίζεται το φαινόμενο ή η ιστορία μας ή το παιχνίδι που θα δημιουργήσουμε.
- Θα σχεδιάσουμε τα sprites του έργου, ή θα χρησιμοποιήσουμε κάποια ήδη υπάρχοντα.
- Θα προσδιορίσουμε τη συμπεριφορά του κάθε sprite (αν θα είναι στατικό ή αν θα κινείται βάση κάποιων νόμων-κανόνων) και
- Ποιες θα είναι οι αλληλεπιδράσεις μεταξύ των sprites και του χρήστη.

Αφού γράψουμε το πρόγραμμά μας για καθένα από τα υποπροβλήματα τα τρέχουμε –εκτελούμε για να ελέγξουμε αν το αποτέλεσμα είναι το επιθυμητό με βάση τον αρχικό σχεδιασμό. Αφού συνθέσουμε τα επιμέρους κομμάτια του κώδικα ελέγχουμε και πάλι, αυτή τη φορά ολόκληρο το πρόβλημα και κάνουμε τις σχετικές διορθώσεις.

Τα προγράμματα **Scratch** μπορούν να φορτωθούν αυτόματα από το περιβάλλον ανάπτυξης σε προσωπικές σελίδες στον ιστότοπο του Scratch, όπου άλλα μέλη της κοινότητας του Scratch μπορούν να τα μεταφορτώσουν (συμπεριλαμβανομένου του πλήρους πηγαίου κώδικα) για μάθηση ή ανάμιξη σε νέα έργα. Ο ιστότοπος, παρέχει επίσης δυνατότητα στα μέλη της κοινότητας να σχολιάσουν έργα, πέραν της παροχής γενικών χώρων συζητήσεων και χώρων επίδειξης έργων. Προγράμματα που έχουν αναπτυχθεί σε Scratch μπορούν να εκτελούνται είτε στο περιβάλλον ανάπτυξης ή μέσω μιας μικροεφαρμογής Java γνωστής ως Scratch Player. Ο Scratch Player επιτρέπει σε προγράμματα Scratch να εκτελεστούν από σχεδόν οποιαδήποτε εφαρμογή εμφάνισης ιστοσελίδων ή [blogs](#). [15]

Εκτός από την εύκολη δημοσίευση και διαμοιρασμό των έργων στο διαδίκτυο, είναι εύκολη και η δημιουργία συλλογής έργων όπως:

- [Best of Geometry](#)
- [Math projects](#)
- [Physics Lab at Scratch](#)
- [Optics for Education](#)
- [Cellular Automaton and L-System](#)
- [geography](#)

κ.λ.π.

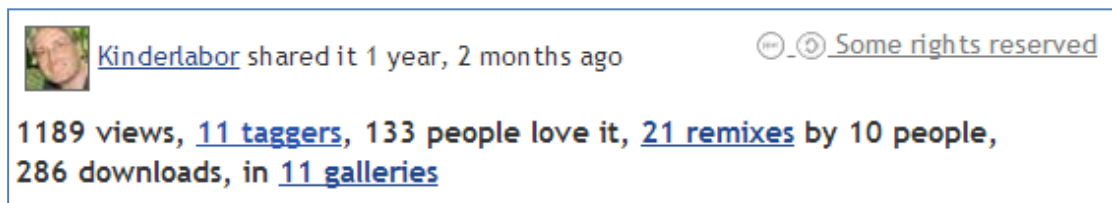
Η κοινότητα **Scratch** δίνει επίσης τη δυνατότητα για remix. Οι δημιουργοί έργων scratch μπορούν να επεκτείνουν τα projects κάνοντας κάποιες τροποποιήσεις του αρχικού κώδικα. Το έργο μας αυτό θα φαίνεται ότι βασίστηκε στο έργο κάποιου άλλου μέλους και θα φαίνεται κάτω από το δικό μας έργο και το όνομα και το έργο του μέλους αυτού. Π.χ.



[marynasta](#) shared it 7 months ago

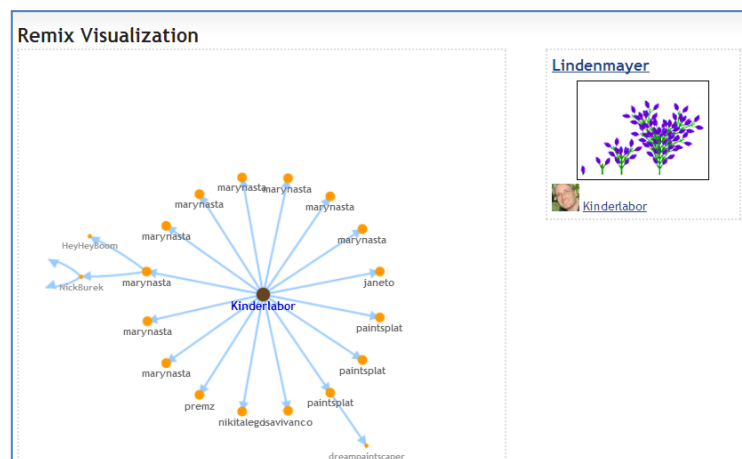
Based on [Kindertabor's project](#)

Στο παράδειγμα αυτό φαίνεται ότι το πιο πάνω [έργο](#) μου βασίστηκε στο έργο του Kinderlabor και αν εμφανίσουμε το έργο του κάνοντας κλικ στο «[project](#)» τότε κάτω από το έργο του αυτό θα δούμε πόσα remixes έχουν γίνει όπως φαίνεται στη



παρακάτω εικόνα.

Στη περίπτωση που θέλουμε να δούμε από ποιους έγιναν οι τροποποιήσεις με κλικ στο «[21remixes](#)» εμφανίζεται η παρακάτω οπτικοποίηση των τροποποιήσεων που φαίνεται από ποιους έγινε και ποια είναι τα remix-έργα.



Το λογισμικό είναι διαθέσιμο για διαφορετικά λειτουργικά συστήματα windows Mac OS X ή Linux και η εγκατάστασή του είναι πολύ απλή. [14]

Το περιβάλλον ανάπτυξης του Scratch μπορεί να μεταφορτωθεί δωρεάν:

Download Scratch: http://info.scratch.mit.edu/Scratch_1.4_Download

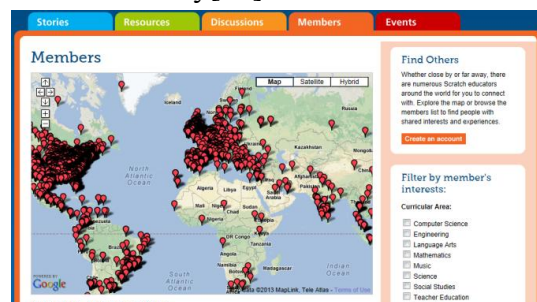
2.2. Scratched - ιστότοπος



Εκπαιδευτικοί πολλών ειδικοτήτων έχουν υποστηρίξει τους δημιουργούς του scratch τόσο σε επίσημα όσο και σε ανεπίσημα εκπαιδευτικά περιβάλλοντα μαθησης:

- Ένας δάσκαλος που θέλει να μοιραστεί ιστορίες για το scratch αλλά και για τη διαθεματική

Το [Scratched](#) είναι διαδικτυακή κοινότητα που έχουν δημιουργήσει οι άνθρωποι του Scratch ειδικά για τους εκπαιδευτικούς.[16]



προσέγγιση της διδασκαλίας.

- Ένας δάσκαλος που θέλει ανατροφοδότηση σχετικά με το υλικό που αναπτύχθηκε για την εξερεύνηση του Scratch στα πλαίσια της συνεργατικής μάθησης. [16]
- Ένας δάσκαλος που χρειάζεται συμβουλές για το πως θα εισαγάγει το Scratch σε ένα γυμνάσιο.
- Υπεύθυνος εκπαιδευτικός ενός project στο scratch που θέλει να συνδεθεί και να συνεργαστεί με άλλα σχολεία που έχουν εισάγει το project στο **Scratch**. [16]

Σε απάντηση αυτής της αναπτυσσόμενης κοινότητας των εκπαιδευτικών που εργάζεται με scratch, έχει αναπτυχθεί το Scratched. Ξεκίνησε τον Ιούλιο του 2009 και είναι μια νέα διαδικτυακή κοινότητα όπου οι εκπαιδευτές του **Scratch** μοιράζονται ιστορίες ανταλλάσσουν πόρους και εμπειρίες στο χώρο συζητήσεων (forum) με συναδέλφους και μπορούν να εντοπίσουν και άλλους εκπαιδευτικούς που χρησιμοποιούν το scratch από τη περιοχή τους αλλά και από όλο τον κόσμο. [16]

Μέσω του Scratched οι εκπαιδευτικοί ανακαλύπτουν νέες μεθόδους διδασκαλίας και κατευθύνουν τους μαθητές στη συνεργασία για τη δημιουργία έργων.

2.3. Scratch 2.0

Η ομάδα του **Scratch** στο Mit Media Lab, τα τελευταία 3 χρόνια, εργάζεται σε μια νέα γενιά του scratch, το scratch 2.0 [17]

Ο κύριος στόχος της είναι να ενισχύσει τη δημιουργικότητα και τη συνεργασία. Στο **Scratch 2.0** μπορούμε να φτιάξουμε νέους τύπους έργων, να συνεργαστούμε και να μάθουμε νέα πράγματα κατά τη διαδικασία. [17]

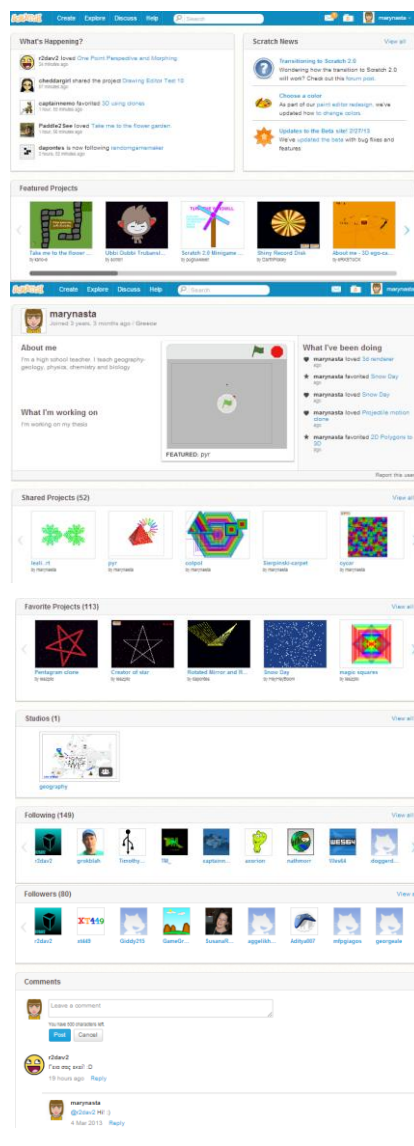
Ας δούμε μερικά νέα χαρακτηριστικά του **Scratch 2.0**. [17]

Στην αρχική σελίδα και στην αριστερή στήλη μπορούμε να δούμε τις τελευταίες δραστηριότητες των μελών. Αν π.χ. ακολουθεί κάποιος ένα άλλο μέλος γιατί είδε τα έργα του και του φάνηκαν ενδιαφέροντα, αν πρόσθεσε κάποιο έργο στα αγαπημένα του, αν του άρεσε κάποιο έργο ή αν μοιράστηκε κάποιο έργο δικό του.

Ακριβώς από κάτω και κινώντας τη μπάρα μπορείς να δεις όλα τα νέα έργα που δημιουργήθηκαν και μοιράστηκαν με τα υπόλοιπα μέλη.

Μπορείς να εκτελέσεις ένα έργο σύμφωνα με τις οδηγίες που δίνει ο δημιουργός. Με κλικ στο αστεράκι και στη καρδούλα να το βάλεις στα αγαπημένα σου έργα ή να δείξεις ότι σ'αρέσει.

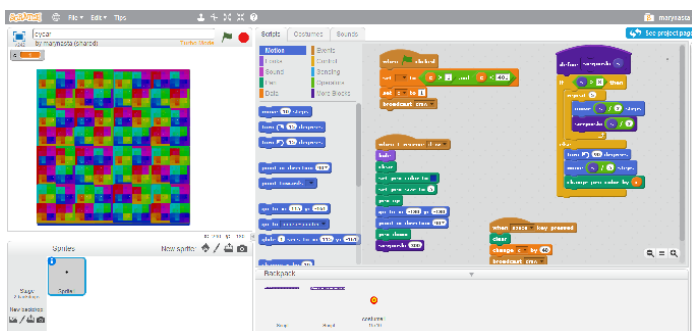
Στη προσωπική σελίδα σου στο «profile» σου μπορείς να εμφανίσεις ένα από τα έργα σου το οποίο έχεις επιλέξει ως χαρακτηριστικό. Να παρουσιάσεις



τον εαυτό σου στα υπόλοιπα μέρη της κοινότητας γράφοντας για τα ενδιαφέροντά σου αλλά και με το τι ασχολείται αυτή την εποχή.

Στη σελίδα σου μπορούν όλοι να δουν τα έργα σου και τις συλλογές σου, καθώς και τα μέλη που παρακολουθείς αλλά και αυτά που σε ακολουθούν. Στο κάτω μέρος επίσης υπάρχει χώρος για σχόλια.

Και σ' αυτή την έκδοση του scratch τα μέλη κάτω από το έργο μπορούν να γράψουν σχόλια που μερικές φορές σε βοηθούν να κάνεις το έργο καλύτερο.

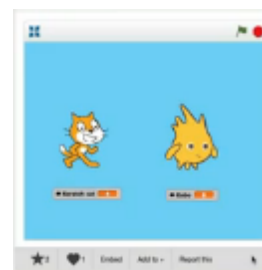


Το **Scratch 2.0** μπορείς να κάνεις πολύ περισσότερα. Μπορείς να δημιουργήσεις νέα blocks-εντολές. Επιπλέον και σημαντικό είναι ότι μπορείς να δεις εσωτερικά το έργο κάνοντας κλικ στο «see inside» να δεις δηλαδή πως φτιάχτηκε το έργο και να πειραματιστείς με το κώδικα.

(Ένα παράδειγμα έργου με χρήση νέου block-εντολής είναι το [lealibh](#))

Από τη καρτέλα «costumes» μπορείς ν' αποθηκεύσεις sprites στον αποθηκευτικό χώρο που υπάρχει στο κάτω μέρος στο «**backup**» και να τα χρησιμοποιήσεις σε ένα δικό σου έργο. Τα νέα blocks-εντολές που θα θεωρήσεις αξιόλογα, ή που θα σε εντυπωσιάσουν, σε ένα άλλο έργο, μπορείς και αυτά να τα αποθηκεύσεις στο χώρο αυτό «backup».

Ένα άλλο νέο χαρακτηριστικό του **Scratch 2.0** είναι το **cloud data** που με τη χρήση μεταβλητών μπορείς να δημιουργήσεις δεδομένα στα οποία να έχουν πρόσβαση όλα τα μέλη. Όπως ένα ερωτηματολόγιο. Ένα απλό παράδειγμα που φαίνεται στην εικόνα με μόνο μια ερώτηση: «Ποιο είναι το αγαπημένο σου sprite;» και «ψήφισε με ένα κλικ». Κάθε φορά που ένα μέλος ψηφίζει αυξάνεται το σκορ online για το αντίστοιχο sprite.

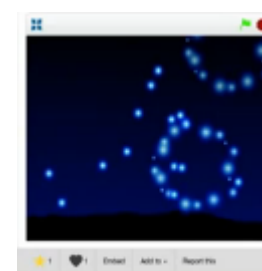


Ένα άλλο παράδειγμα είναι η δημιουργία ενός παιχνιδιού που θα καταγράφεται σε μια λίστα το σκορ κάθε μέλους που έπαιξε και να αποθηκεύεις τα υψηλότερα σκορ οποιουδήποτε και όποτεδήποτε έπαιξε το παιχνίδι.

Σημαντικό επίσης χαρακτηριστικό του scratch 2.0 είναι η **κλωνοποίηση** που σου δίνει τη δυνατότητα να δημιουργείς πολλά αντίγραφα ενός και μόνο script χρησιμοποιώντας την εντολή



(Παράδειγμα έργου που χρησιμοποιώ τη παραπάνω εντολή είναι το [pyr](#))



Μπορείς τέλος να χρησιμοποιήσεις την **web-camera** στο κομπιούτερ σου για να ανιχνεύσεις κινήσεις σώματος. Στην εικόνα φαίνεται ένα έργο όπου μπορείς να σπάσεις μπαλόνια απλά κινώντας τα χέρια σου.



3. Υλοποίηση στο scratch

Το <http://scratch.mit.edu/projects/marynasta/2708029> έργο μου (The painter) για την υλοποίηση των L-systems στο scratch βασίστηκε στο έργο του [Kinderlabor](#) από το οποίο χρησιμοποίησα ένα κομμάτι του κώδικα το οποίο αναλύω πιο κάτω. Βασίστηκε επίσης σε μια μικρή φανταστική ιστορία την οποία συνέθεσα ώστε ο συνδυασμός των γραφικών (L-systems) και του παραμυθιού αυτού να προσελκύει τον επισκέπτη να παρακολουθήσει όλα τα γραφικά μέχρι τέλους της ιστορίας.

Και η φανταστική ιστοριούλα έχει ως εξής:

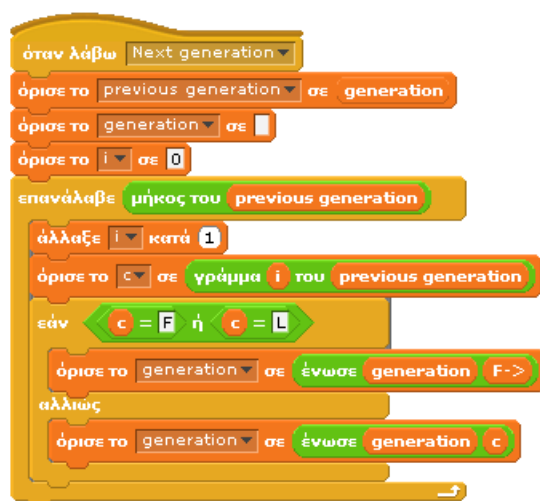
Υπήρχε ένα μικρό αγόρι που ονομαζόταν Ιερεμίας. Αυτός ζούσε σ'ένα μικρό σπίτι κοντά στο δάσος.Κάθε μέρα περιπλανιόταν στο δάσος. Η μητέρα του συνέχεια του έλεγε ότι δεν θα μάθαινε τίποτα ποτέ τίποτα για τον κόσμο.Αυτός συνέχεια γελούσε και της έδινε πάντα την ίδια απάντηση. «Εγώ μπορώ να δώ ένα τόσο όμορφο κόσμο που οι περισσότεροι άνθρωποι δεν μπορούν να δουν, αν και δεν είναι τυφλοί.»

Αυτός συνήθιζε να περπατά στο δάσος και να παρατηρεί όλα γύρω του. Μπορούσε να δει τα διαφορετικά χρώματα που εμφανίζονταν από τις ακτίνες του ήλιου. Μπορούσε ν'ακούσει τη «μουσική» από το θρόισμα των φύλλων που κανείς άλλος δεν μπορούσε ν'ακούσει.Παρατηρούσε τον ουρανό και ταξίδευε μέσα από τα σύννεφα. Επέστρεφε στο σπίτι του κάθε βράδυ.Επαναλάμβανε τον περίπατό του κάθε μέρα και το κεφάλι του γέμιζε με εικόνες, χρώματα και ήχους.

Ένα πρωί μερικά χρόνια αργότερα, ζύπνησε με τρομερό πονοκέφαλο. Ένωσε το κεφάλι του βαρύ σαν πέτρα. Δεν μπορούσε να κάνει ούτε ένα βήμα. Την επόμενη μέρα ένωθε το ίδιο. Άρχισε να ψάχνει το ντουλάπι του για να βρει το κουτί που του είχε δώσει η νουνά του όταν ήταν πέντε χρονών.Ένα πινέλο, χρώματα και χαρτιά ήταν μέσα στο κουτί. Άρχισε να ζωγραφίζει και ένα θαύμα συνέβη. Είκονες και ήχοι άρχισαν να βγαίνουν από το μυαλό του. Δεν είχε πια πονοκέφαλοκαι ένωθε το κεφάλι λιγότερο βαρύ. Συνέχιζε να ζωγραφίζει για μέρες, για χρόνια...Το κεφάλι του άδειασε από τις σκέψεις του σαν να περπατούσε μέσα στο δάσος πάλι...

3.1. Παραγωγή της επόμενης γενιάς

Αυτός ο κώδικας δεδομένης της τρέχουσας γενιάς ενός L-system εφαρμόζει τους κανόνες και παράγει την επόμενη γενιά.



Χρησιμοποιούμε ένα δείκτη τον i , που δείχνει στα στοιχεία της προηγούμενης γενιάς.

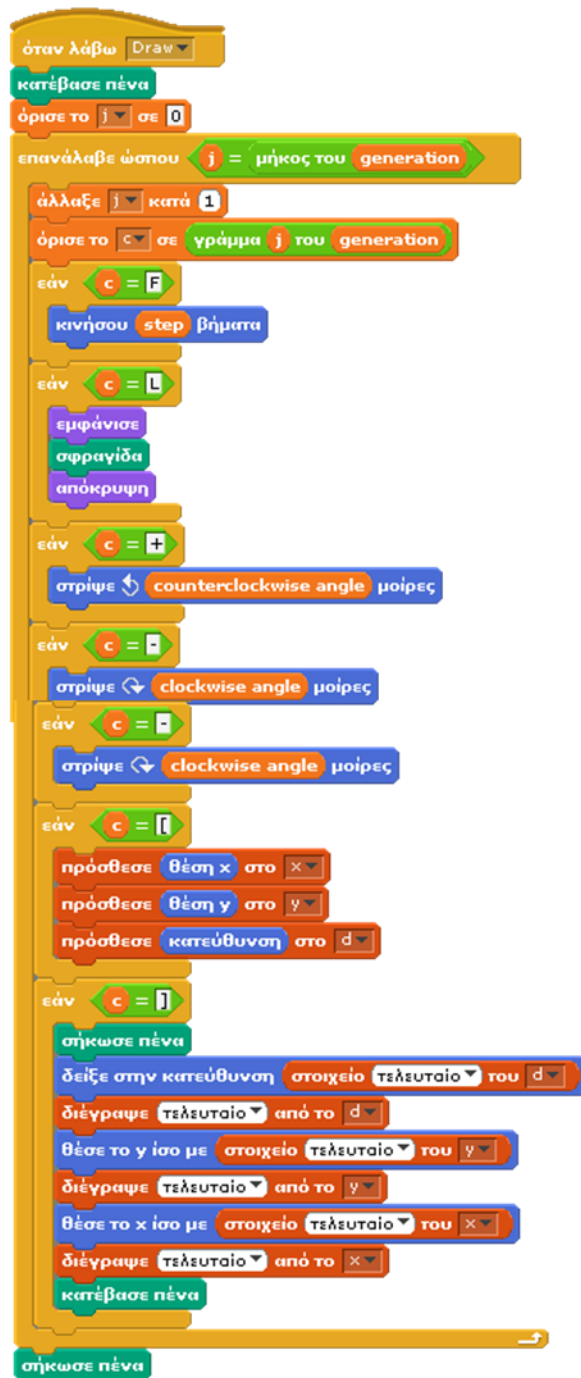
Για να δημιουργηθεί η επόμενη γενιά ορίζουμε σαν generation (γενιά) την προηγούμενη γενιά (previous generation). Η επόμενη γενιά κτίζεται με βάση τη προηγούμενη γενιά. Χρησιμοποιούμε ένα δείκτη τον i που δείχνει στη συμβολοσειρά από την οποία καθορίζεται η προηγούμενη γενιά.

Ξεκινώντας, το i δείχνει το πρώτο γράμμα της συμβολοσειράς της προηγούμενης γενιάς και ορίζουμε το c να είναι το γράμμα που δείχνει το i .

Αν το γράμμα αυτό είναι το F ή το L τότε ορίζεται ως generation η ένωση του generation με το F → Διαφορετικά ορίζεται ως generation η ένωση του generation με το c (που θα αντιστοιχεί σε κάποια από τις σταθερές (+, -, [,]).

Στη συνέχεια το i δείχνει στο δεύτερο γράμμα της συμβολοσειράς Η διαδικασία επαναλαμβάνεται μέχρι να ολοκληρωθεί το μήκος της προηγούμενης γενιάς δηλαδή ολόκληρη η συμβολοσειρά.

3.2. Σχεδίαση της τρέχουσας γενιάς



Επιλέγουμε το γάτο scratch ή κάποιο άλλο αντικείμενο για να εκτελέσει συγκεκριμένες εντολές (να κινηθεί, να γράψει, να στραφεί, σφραγίδα κ.λ.π.) για ορισμένα σύμβολα καθώς θα διαβάζεται η συμβολοσειρά.

Δημιουργούμε τρεις λίστες το d (κατεύθυνση πέννας) τη x (τεταγμένη) και τη y (τεταγμένη) για κάθε θέση του αντικειμένου που γράφει-ζωγραφίζει

Χρησιμοποιούμε ένα δείκτη τον j που δείχνει στη συμβολοσειρά από την οποία καθορίζεται η τρέχουσα γενιά.

Ξεκινώντας, το j δείχνει το πρώτο γράμμα της συμβολοσειράς της τρέχουσας γενιάς και ορίζουμε το c να είναι το γράμμα που δείχνει το j.

Αν το γράμμα αυτό είναι το F τότε η πένα θα γράψει γραμμή μήκους step

Αν το γράμμα αυτό είναι το L τότε το αντικείμενο που γράφει θα αποτυπωθεί στη θέση αυτή με τη σφραγίδα

Αν είναι + τότε το αντικείμενο που γράφει θα στρίψει κατά counterclock angle μοίρες προς τ' αριστερά

Αν το γράμμα αυτό είναι - τότε το αντικείμενο που γράφει θα στρίψει κατά clockwise angle μοίρες προς τ' δεξιά

Αν το γράμμα αυτό είναι [τότε θα προστεθεί στη λίστα x η τεταγμένη του αντικειμένου που

ζωγραφίζει, στη λίστα y η τεταγμένη του και στη λίστα d η κατεύθυνσή του.

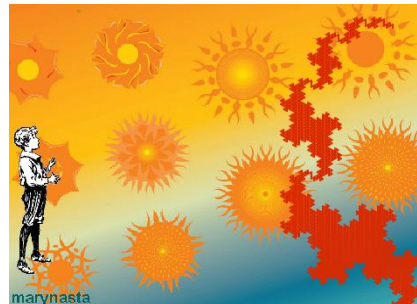
Αν είναι το] θα ανακαλέσει τη κατεύθυνση και τις συντεταγμένες που είχε αποθηκεύσει με το σύμβολο [(οπότε και η χελώνα-πένα θα πάρει αυτή τη θέση και κατεύθυνση) και μετά θα τα σβήσει από τις λίστες

3.3. Καλοκαίρι (Terdragon curve)

...Ζωγράφισε τον ήλιο του καλοκαιριού, χρησιμοποιώντας κόκκινο, κίτρινο και πορτοκαλί με πολλά όμορφα σχήματα και τη θερμότητα σαν κόκκινους δράκους που έβγαιναν από τον ήλιο και έρχονταν κάτω στη γη...

```

όταν λάβω sum
όρισε το μέγεθος σε 10 %
όρισε το F-> σε F+F-F
όρισε το step σε 4
όρισε το counterclockwise angle σε 120
όρισε το clockwise angle σε 120
όρισε το generation σε F+F-F
όρισε το χρώμα πέννας σε
όρισε το μέγεθος πέννας σε 2
πήγαινε στο x: 200 y: 150
δείξε στην κατεύθυνση 0
επανάλαβε 3
  μετάδωσε Next generation
  μετάδωσε Draw και περίμενε
  μετάδωσε Next generation
  μετάδωσε Draw και περίμενε
  περίμενε 0.5 δευτερόλεπτα
περίμενε 5 δευτερόλεπτα
μετάδωσε Clean up και περίμενε
μετάδωσε fall και περίμενε
  
```



variables : F
 constants : + - []
 start : F+F-F
 rules : F → F+F-F
 angle : 120°

3.4. Φθινόπωρο (trees)

...Ζωγράφισε το Φθινόπωρο με δέντρα χωρίς φύλλα και νεκρά φύλλα που κοίτονταν στο έδαφος...

Το πρώτο δέντρο του φθινοπώρου είναι ένα απλό δυαδικό δέντρο, με γωνία διακλάδωσης 30°. Παράγεται με τέσσερις επαναλήψεις που σε κάθε μία καλείται η επόμενη γενιά.

Το δεύτερο δέντρο (Ken's tree), είναι ένα πιο σύνθετο δέντρο. Σχεδιάζεται πρώτα ένας διπλός σε μήκος κορμός και στη συνέχεια ένα κλαδί προς τ' αριστερά με γωνία διακλάδωσης 22.5° και ένα κλαδί προς τα δεξιά με την ίδια γωνία διακλάδωσης. Κάθε κλαδί αποτελείται από επί μέρους κλαδιά.

Το προς τα δεξιά κλαδί περιγράφεται ως εξής: στροφή δεξιά-κλαδί, στροφή αριστερά-κλαδί, στροφή αριστερά-κλαδί.

Το προς τ' αριστερά κλαδί περιγράφεται ως εξής: στροφή αριστερά-κλαδί, στροφή δεξιά-κλαδί, στροφή δεξιά-κλαδί.

```

όταν λάβω Fall
καθάρισε
μετάδωσε Clean up
περίμενε 5 δευτερόλεπτα
όρισε το μέγεθος σε 10 %
όρισε το F-> σε F[+F][-F]
όρισε το step σε 50
όρισε το counterclockwise angle σε 15
όρισε το clockwise angle σε 15
όρισε το generation σε F
όρισε το μέγεθος πέννας σε 7
όρισε το χρώμα πέννας σε
πήγαινε στο x: 0 y: -100
δείξε στην κατεύθυνση 0
επανάλαβε 4
  μετάδωσε Next generation
  μετάδωσε Draw και περίμενε
  περίμενε 0.5 δευτερόλεπτα
  όρισε το χρώμα πέννας σε
  όρισε το F-> σε FF[-F+F+F][+F-F-F]
  όρισε το step σε 10
  όρισε το counterclockwise angle σε 22.5
  όρισε το clockwise angle σε 22.5
  όρισε το generation σε F
  όρισε το μέγεθος πέννας σε 2
  πήγαινε στο x: 0 y: -100
  επανάλαβε 3
    όρισε το χρώμα πέννας σε
    μετάδωσε Next generation
    μετάδωσε Draw και περίμενε
    όρισε το χρώμα πέννας σε
    περίμενε 0.5 δευτερόλεπτα
  όρισε το F-> σε F[-F]F[+F][F]
  όρισε το step σε 30
  όρισε το counterclockwise angle σε 30
  όρισε το clockwise angle σε 30
  όρισε το generation σε F
  όρισε το μέγεθος πέννας σε 2
  πήγαινε στο x: 0 y: -100
  επανάλαβε 3
    όρισε το χρώμα πέννας σε
    μετάδωσε Next generation
    μετάδωσε Draw και περίμενε
    περίμενε 0.5 δευτερόλεπτα

```



variables : F
 constants : + - []
 start : F
 rules : $F \rightarrow F[+F][-F]$
 angle : 30°



variables : F
 constants : + - []
 start : F
 rules : $F \rightarrow FF[-F+F+F][+F-F-F]$
 angle : 22.5°



variables : F
 constants : + - []
 start : F
 rules : $F \rightarrow F[-F]F[+F][F]$
 angle : 30°

Στο τρίτο δέντρο σχεδιάζεται πρώτα ο κορμός και ένα κλαδί προς τα δεξιά με γωνία διακλάδωσης 30°. Το αντικείμενο που σχεδιάζει επιστρέφει στο σημείο διακλάδωσης και επεκτείνει το κορμό. Σχεδιάζει κλαδί προς τ' αριστερά και επιστρέφει στην αρχή για να επεκτείνει το κορμό και πάλι. Το αντικείμενο επιστρέφει στην αρχή του κορμού

που σχεδίασε. Παράγεται με τέσσερις επαναλήψεις που σε κάθε μία καλείται η επόμενη γενιά.

3.5. Χειμώνας

...Ζωγράφισε το χειμώνα με άσπρο, ρόζ, μώβ και γκρι χρώματα, απεικονίζοντας τις νιφάδες του χιονιού με όμορφα γεωμετρικά σχήματα...

```

όταν λάβω snowflake1
  σήκωσε πένα
  όρισε το μέγεθος σε 10 %
  όρισε το F-> σε F-F++F-F-F-F
  όρισε το step σε 4
  όρισε το counterclockwise angle σε 72
  όρισε το clockwise angle σε 72
  όρισε το generation σε F-F-F-F-F
  όρισε το χρώμα πέννας σε
  όρισε το μέγεθος πέννας σε 2
  πήγαινε στο x: -50 y: 160
  δείξε στην κατεύθυνση 0
  κατέβασε πένα
  επανάλαβε 3
    μετάδωσε Draw και περίμενε
    περίμενε 0.5 δευτερόλεπτα
    μετάδωσε Next generation
    άλλαξε y κατά -50
    περίμενε 3 δευτερόλεπτα
    καθάρισε
  μετάδωσε Draw και περίμενε
  περίμενε 0.5 δευτερόλεπτα
  μετάδωσε Next generation
  όρισε το F-> σε F-F++F-F
  όρισε το step σε 5
  όρισε το counterclockwise angle σε 60
  όρισε το clockwise angle σε 60
  όρισε το generation σε F++F++F
  όρισε το χρώμα πέννας σε
  όρισε το μέγεθος πέννας σε 2
  πήγαινε στο x: -14 y: -2
  δείξε στην κατεύθυνση 0
  μετάδωσε Next generation
  περίμενε 0.5 δευτερόλεπτα
  μετάδωσε Draw και περίμενε
  πήγαινε στο x: -6 y: -16
  δείξε στην κατεύθυνση 0
  μετάδωσε Next generation
  περίμενε 0.5 δευτερόλεπτα
  μετάδωσε Draw και περίμενε
  πήγαινε στο x: 20 y: -60
  δείξε στην κατεύθυνση 0
  μετάδωσε Next generation
  περίμενε 0.5 δευτερόλεπτα
  μετάδωσε Draw και περίμενε
  περίμενε 10 δευτερόλεπτα
  μετάδωσε Clean up και περίμενε
  μετάδωσε snowflake2 και περίμενε
  
```

3.5.1. "Pleasant error".

Πρόκειται για ένα "ευχάριστο λάθος" μιας αποτυχημένης αναζήτησης ενός νέου spacefilling curve (καμπύλη πλήρωσης χώρου).[3]



variables : F
 constants : + -
 start : F+F-F
 rules : F → F-F++F-F-F
 angle : 30°

3.5.2. "The Koch snowflake"



variables : F
 constants : + -
 start : F++F++F
 rules : F → F-F++F-F
 angle : 60°

3.5.3. “Snowflake2”

Σε αυτή τη νιφάδα χιονιού η πένα ζωγραφίζει μια φορά την επόμενη γενιά. Στη συνέχεια η πένα πηγαίνει στην αρχή (0,0) και αφού στραφεί κατά γωνία= 60° επαναλαμβάνει την ίδια διαδικασία δηλαδή ζωγραφίζει μέχρι τη δεύτερη γενιά. Οι επαναλήψεις είναι έξι (6) σε αριθμό οπότε προκύπτει μια νιφάδα εξαγωνικής μορφής.

```

όταν λάβω snowflake2
  περίμενε 3 δευτερόλεπτα
  όρισε το μέγεθος σε 10 %
  όρισε το F-> σε F[+FF][-FF]F
  όρισε το step σε 10
  όρισε το counterclockwise angle σε 30
  όρισε το clockwise angle σε 30
  όρισε το generation σε F[+FF][-FF]F
  όρισε το χρώμα πέννας σε 
  όρισε το μέγεθος πέννας σε 4
  πήγαινε στο x: 0 y: 0
  δείξε στην κατεύθυνση 0
  μετάδωσε Next generation
  επανάλαβε 6
    περίμενε 0.5 δευτερόλεπτα
    μετάδωσε Draw και περίμενε
    πήγαινε στο x: 0 y: 0
    στρίψε 60 μοίρες
  
```



variables : F
 constants : + - []
 start : F[+FF][-FF]F
 rules : F → F[+FF][-FF]F
 angle : 30°

3.5.4. Snowflake3

Το snowflake3 υλοποιείται με τον ίδιο τρόπο όπως και το snowflake2

```

όταν λάβω snowflake3
  όρισε το μέγεθος σε 10 %
  όρισε το F-> σε F[+F][F][-F][F]F[+F][-F]
  όρισε το step σε 10
  όρισε το counterclockwise angle σε 30
  όρισε το clockwise angle σε 30
  όρισε το generation σε F[+F][F][-F]F
  όρισε το χρώμα πέννας σε 
  όρισε το μέγεθος πέννας σε 4
  πήγαινε στο x: 0 y: 0
  δείξε στην κατεύθυνση 0
  μετάδωσε Next generation
  επανάλαβε 6
    όρισε το χρώμα πέννας σε 
    περίμενε 0.5 δευτερόλεπτα
    μετάδωσε Draw και περίμενε
    πήγαινε στο x: 0 y: 0
    στρίψε 60 μοίρες
    άλλαξε μέγεθος πέννας κατά 0.5
  
```

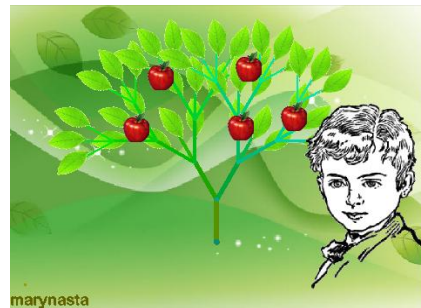


variables : F
 constants : + - []
 start : F[+F][F][-F]F
 rules : F → F[+F][F][-F][F]F[+F][-F]
 angle : 30°

3.6. Άνοιξη

...Ζωγράφισε την άνοιξη χρησιμοποιώντας πράσινο , κίτρινο μπλε και κόκκινο δείχνοντας τη ζωτικότητα και την αναγέννηση....

Το δέντρο της άνοιξης είναι ένα δυαδικό δέντρο όπως και το πρώτο του φθινοπώρου. Διαφέρει στη γωνία διακλάδωσης και για τη τελευταία γενιά (χρησιμοποίησα τον τύπο με τύπο $F=F[+FL][-FL][FL]$ που σημαίνει ένας κορμός , ένας κλάδος προς τα δεξιά, ένας κλάδος προς τ' αριστερά και ένας κλάδος σαν συνέχεια του κορμού. Στο τέλος κάθε διακλάδωσης αποτυπώνεται (με την εντολή σφραγίδα) ένα φύλλο (L). Έτσι το δέντρο φαίνεται πιο φουντωτό.



variables :	F
constants :	+ - []
start :	[F]
rules :	$F \rightarrow F[+F][-F]$, $F \rightarrow F[+FL][-FL][FL]$
angle :	30°

...το μικρό αγόρι έγινε ζωγράφος.

3.7. Τρίγωνο Sierpinski

3.7.1. Sierpinski_triangle

Το τρίγωνο Sierpinski (<http://scratch.mit.edu/projects/marynasta/2808241>) που ονομάζεται επίσης και κόσκινο Sierpinski είναι ένα φράκταλ που πήρε τ' όνομά του από τον μαθηματικό Waclaw Sierpinski ο οποίος το παρουσίασε το 1915. Ωστόσο παρόμοια μοτίβα είχαν εμφανιστεί ήδη από το 13ο αιώνα τα ονομαζόμενα μωσαϊκά Cosmati στον καθεδρικό ναό του Anagni στην Ιταλία αλλά και σε άλλα μέρη όπως στο ναό της Ρωμαϊκής Βασιλικής της Santa Maria στο Cosmedin. Είναι από τα βασικά παραδείγματα των συνόλων αυτοομοιότητας. Είναι ένα μοτίβο που παράγεται

με μαθηματικό τρόπο και μπορεί ν' αναπαραχθεί σε οποιαδήποτε μεγέθυνση ή σμίκρυνση.[18]



Θα εξετάσουμε πώς μεταβάλλεται η περίμετρος και το εμβαδόν του τριγώνου σε κάθε βήμα:

Βήμα 1ο: Θεωρούμε το μήκος κάθε πλευράς 1 και το αρχικό εμβαδόν μοναδιαίο (E_1)

Περίμετρος 3

Εμβαδόν E_1

Βήμα 2ο: Περίμετρος $9/2$ (είναι 9 ευθύγραμμα τμήματα που το καθένα έχει το μισό μήκος του αρχικού).

Εμβαδόν $3/4$ (του E_1)

Βήμα 3ο: Περίμετρος $27/4$

Εμβαδόν $9/16$ (του E_1)

Βήμα 4ο: Περίμετρος $81/8$

Εμβαδόν $27/64$ (του E_1)

Βήμα n : Περίμετρος $3^{n+1}/2^n$

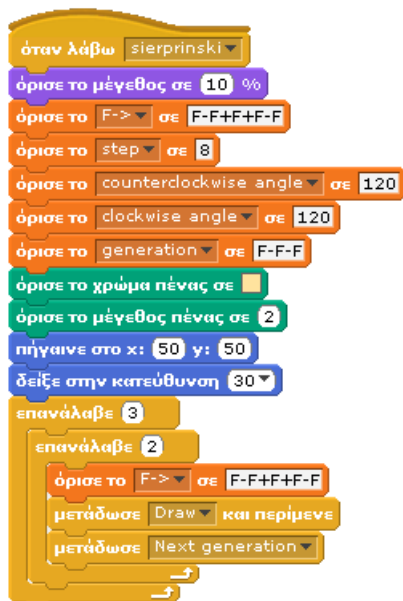
Εμβαδόν $3^n / 4^n$ (του E_1)

$$\lim_{n \rightarrow \infty} 3^{n+1}/2^n = \lim 3 (3/2)^n = \infty$$

$$\lim_{n \rightarrow \infty} 3^n / 4^n = \lim (3/4)^n = 0$$

Είναι ενδιαφέρον να παρατηρήσουμε ότι μετά από n βήματα η περίμετρος τείνει στο άπειρο ενώ το εμβαδόν τείνει στο μηδέν. [6]

Η υλοποίηση Sierpinski στο scratch παρουσιάζεται παρακάτω:



variables :	F
constants :	+ -
start :	F-F-F
rules :	F → F-F+F+F-F
angle :	120°

3.7.2. sierpinski version2

L-system με δύο κανόνες

Η πρώτη γενιά καθορίζεται από το F-F-F όπως και στη προηγούμενη περίπτωση των πυραμίδων. Εδώ η επόμενη γενιά παράγεται εναλλάξ από δύο κανόνες τον F-> και τον X-> και ενώνονται μεταξύ τους και με τη προηγούμενη σχηματισμένη μέχρι εκείνη τη στιγμή όπως φαίνεται στο Next generation.

(<http://scratch.mit.edu/projects/marynasta/2829785>) (sierpinski version2)

```

όταν λάβω sierprinski
  περίμενε 5 δευτερόλεπτα
  όρισε το μέγεθος σε 10 %
  όρισε το F-> σε F-F+F+F-F
  όρισε το step σε 4
  όρισε το counterclockwise angle σε 120
  όρισε το clockwise angle σε 120
  όρισε το generation σε F-F-F
  όρισε το χρώμα πέννας σε 
  όρισε το μέγεθος πέννας σε 2
  πήγαινε στο x: -80 y: -50
  δείξε στην κατεύθυνση 30
  επανάλαβε 2
    όρισε το F-> σε F-F+F+F-F
    μετάδωσε Draw και περίμενε
    μετάδωσε Next generation
    όρισε το X-> σε FF
    μετάδωσε Draw και περίμενε
    μετάδωσε Next generation
  
```



variables :	F,X
constants :	+ -
start :	F-F-F
rules :	F → F-F+F+F-F, X → FF
angle :	120°

Στο διπλανό κομμάτι του κώδικα φαίνεται ο ορισμός της επόμενης γενιάς. Όταν ο δείκτης j δείχνει σε γράμμα της συμβολοσειράς που είναι το F τότε η επόμενη γενιά προκύπτει από την ένωση της προηγούμενης με την F-> και τη X-> Ενώ όταν ο δείκτης j είναι οποιοδήποτε από τα γράμματα +,-,[,] τότε η επόμενη γενιά προκύπτει από την ενωσή του με τη προηγούμενη γενιά.

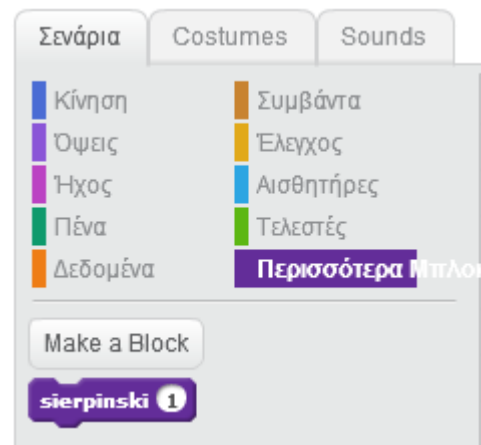
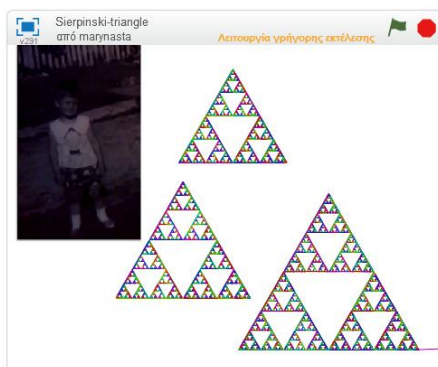
```

εάν c = F
  όρισε το generation σε ένωση generation ένωση F-> X->
αλλιώς
  όρισε το generation σε ένωση generation c
  
```


3.7.3. sierpinski_triangle στο scratch 2.0

Η παραγωγή του [sierpinski triangle](http://beta.scratch.mit.edu/projects/10040902/) στο scratch 2.0 γίνεται πολύ πιο γρήγορα και εύκολα με τη χρήση κατάλληλου νέου block-εντολής.

Η εντολή όπως φαίνεται στην εικόνα είναι η sierpinski μια εντολή αναδρομής που ορίζεται ως εξής: Εάν η απόσταση s (η πλευρά του τριγώνου) προς τη κατεύθυνση κίνησης του αντικειμένου είναι μεγαλύτερη από 3 βήματα τότε θα γράψει τη μισή απόσταση και για κάθε s θα επαναλάβει τη διαδικασία 4 φορές. Διαφορετικά (αν δηλαδή η απόσταση είναι ίση με 3 βήματα) τότε θα κινηθεί ένα βήμα και θα στραφεί κατά 120° δημιουργώντας ένα μικρό τρίγωνο.



Εδώ σχεδιάζονται τρία τρίγωνα με πλευρές που ορίζονται στο κώδικα 200 το πρώτο, 150 το δεύτερο και 120 το τρίτο.

```

όταν στο πρόγραμμα γίνει κλικ
κρύψου
καθάρσε
πήγαινε στη θέση x: 200 y: -200
καθάρσε
όρισε το χρώμα πέννας σε κόκκινο
όρισε το μέγεθος πέννας σε 1
σήκωσε την πένα
πήγαινε στη θέση x: 200 y: -200
δείξε προς την κατεύθυνση 90
κατέβασε την πένα
sierpinski 200
σήκωσε την πένα
πήγαινε στη θέση x: -120 y: -100
δείξε προς την κατεύθυνση 90
κατέβασε την πένα
sierpinski 150
σήκωσε την πένα
πήγαινε στη θέση x: -50 y: 50
δείξε προς την κατεύθυνση 90
κατέβασε την πένα
sierpinski 120
σήκωσε την πένα
  
```

```

ορισμός του sierpinski s
εάν s > 3 τότε
  επανάλαβε 4
    κινήσου s / 2 βήματα
    sierpinski s / 2
  αλλιώς
    κινήσου s / 3 βήματα
    στρίψε 120 μοίρες
    άλλαξε το χρώμα της πέννας κατά 20
  τέλος
τέλος
  
```

3.8. “Real” olive-tree

Είναι ένα χρονικό L-system. Ο κανόνας που εφαρμόζεται είναι εξαρτώμενος από τις επαναλήψεις που είναι μια μεταβλητή. Αυτό σημαίνει ότι ο κανόνας αλλάζει σε κάθε επανάληψη. Η παραγωγή του δέντρου ελιάς (Rtree_ls) σε μορφή που να φαίνεται «αληθινή» επιτυγχάνεται αλλάζοντας κάθε φορά, στην επόμενη γενιά τη γωνία διακλάδωσης και το step όπως φαίνεται στο παρακάτω κομμάτι του κώδικα.

(<http://scratch.mit.edu/projects/marynasta/2818889>)

Η διαφορετική γωνία δίνει την εντύπωση ότι άλλα κλαδιά φυτρώνουν προς τα πίσω και άλλα κλαδιά προς τα πλάγια ή εμπρός. Χρησιμοποίησα ένα μικρό κλαδί με δύο φύλλα ελιάς, σε δύο «ενδυμασίες»(με αντίθετη κατεύθυνση). Έτσι με τη σφραγίδα (L) γίνεται κάθε φορά αποτύπωσή του και στη δεξιά διακλάδωση τα φύλλα έχουν κατεύθυνση προς τα δεξιά και στην αριστερή διακλάδωση τα φύλλα έχουν κατεύθυνση προς τα αριστερά (άλλη ενδυμασία) ώστε το δέντρο να φαίνεται πιο αληθινό.

```

όταν λάβω Tree
  περιμένε 5 δευτερόλεπτα
  αλλαγή σε ενδυμασία Leaf1
  όρισε το μέγεθος σε 10 %
  όρισε το F-> σε F[+FL][-FL]
  όρισε το step σε 50
  όρισε το counterclockwise angle σε 10
  όρισε το clockwise angle σε 10
  όρισε το generation σε [F]
  όρισε το χρώμα πένας σε
  όρισε το μέγεθος πένας σε 7
  πήγαινε στο x: 0 y: -140
  δείξε στην κατεύθυνση 0
  μετάδωσε Draw και περιμένε
  μετάδωσε Next generation
  περιμένε 0.5 δευτερόλεπτα
  άλλαξε clockwise angle κατά 5
  άλλαξε step κατά -2
  άλλαξε counterclockwise angle κατά 5
  άλλαξε step κατά -2
  μετάδωσε Draw και περιμένε
  μετάδωσε Next generation
  περιμένε 0.5 δευτερόλεπτα
  μετάδωσε Draw και περιμένε
  μετάδωσε Next generation
  περιμένε 0.5 δευτερόλεπτα
  αλλαγή σε ενδυμασία Leaf2
  άλλαξε clockwise angle κατά 10
  άλλαξε step κατά -2
  άλλαξε counterclockwise angle κατά 10
  άλλαξε step κατά -2
  μετάδωσε Draw και περιμένε
  μετάδωσε Next generation
  περιμένε 0.5 δευτερόλεπτα
  
```



variables : F
 constants : + , - , [,]
 start : [F]
 rules : F → F[+FL][-FL]
 angle : μεταβλητή

```

εάν c = +
  αλλαγή σε ενδυμασία Leaf1
  στρίψε counterclockwise angle μοίρες
  άλλαξε μέγεθος πένας κατά -1
εάν c = -
  αλλαγή σε ενδυμασία Leaf2
  στρίψε clockwise angle μοίρες
  άλλαξε μέγεθος πένας κατά 1
  
```

Επίσης στο “draw” όταν το σύμβολο της συμβολοσειρά είναι το “+” δηλαδή όταν στρίβει προς τ’ αριστερά, αλλάζω το μέγεθος της πένας κατά -1 ενώ όταν είναι “-“

δηλαδή όταν στρίβει προς τα δεξιά τότε αλλάζω το μέγεθος της πέννας κατά 1. Έτσι τα κλαδιά που φυτρώνουν προς τα δεξιά να φαίνονται πιο παχιά από τα κλαδιά που φυτρώνουν προς τ' αριστερά. Έτσι το δέντρο μου φαίνεται πιο πραγματικό.

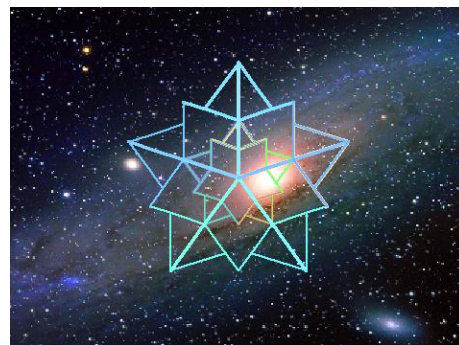
3.9. star_in_star

Το έργο [star_in_star](http://scratch.mit.edu/projects/marynasta/3179734) (<http://scratch.mit.edu/projects/marynasta/3179734>)

Παρουσιάζει συμμετρία ανάκλασης και πενταπλή συμμετρία. Είναι αυτοπαρόμοιο, δηλαδή τα ίδια μοτίβα εμφανίζονται σε μεγαλύτερες και μεγαλύτερες κλίμακες.

```

όταν λάβω starinstar
  περιμένε 3 δευτερόλεπτα
  όρισε το μέγεθος σε 10 %
  όρισε το F-> σε [F[+F][-F]F[++F][--F]]
  όρισε το counterclockwise angle σε 72
  όρισε το clockwise angle σε 72
  όρισε το generation σε F
  όρισε το χρώμα πέννας σε
  όρισε το μέγεθος πέννας σε 5
  πήγαινε στο x: 0 y: 0
  δείξε στην κατεύθυνση 0
  επανάλαβε 3
    όρισε το F-> σε [F[+F][-F]F[++F][--F]]
    μετάδωσε Next generation
    περιμένε 0.5 δευτερόλεπτα
    μετάδωσε Draw και περιμένε
    στρίψε 72 μοίρες
    όρισε το F-> σε FF
    μετάδωσε Next generation
    περιμένε 0.5 δευτερόλεπτα
    μετάδωσε Draw και περιμένε
    στρίψε 72 μοίρες
  
```



variables :	F
constants :	+ , - , [,]
start :	F
rules :	F → [F[+F][-F]F[++F][--F]], F → FF
Angle :	90°

Χρησιμοποιούνται δύο κανόνες για ένα μη τερματικό σύμβολο. Έτσι το F παράγει [F[+F][-F]F[++F][--F]] ή FF εναλλάξ για κάθε γενιά.

3.10. tiling-stavros

Όπως έχει αναφερθεί σε προηγούμενη ενότητα (βλ.1.4.4 Περιοδικό και απεριοδικό tiling) το **tiling** είναι η επικάλυψη μιας επιφάνειας χωρίς επικαλύψεις και κενά. Ο αριθμός των σχημάτων (πεπερασμένος), τα protoliles, που μια σειρά από αυτά εισάγει την επικάλυψη της επιφάνειας.

Στη προκειμένη περίπτωση στο έργο [tiling-stavros](http://scratch.mit.edu/projects/marynasta/2862587) (<http://scratch.mit.edu/projects/marynasta/2862587>) τα protoliles είναι δύο σταυροί (με ορισμένη απόσταση και κατεύθυνση) που είναι και η



περίοδος του **tiling** που σημαίνει ότι επαναλαμβάνοντας τον εαυτό της περιοδικά σε τακτά χωρικά διαστήματα μπορεί να επικαλύψει το επίπεδο. Είναι δηλαδή ένα περιοδικό **tiling**

```

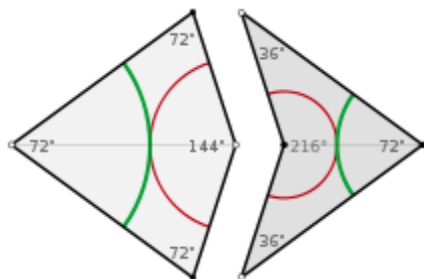
όρισε το μέγεθος σε 10 %
όρισε το F-> σε F+F-F-F+F-F
όρισε το step σε 6
όρισε το counterclockwise angle σε 90
όρισε το clockwise angle σε 90
όρισε το generation σε -F-
όρισε το χρώμα πένας σε 
όρισε το μέγεθος πένας σε 2
πήγαινε στο x: 30 y: 50
δείξε στην κατεύθυνση 72
μετάδωσε Draw και περίμενε
μετάδωσε Next generation
μετάδωσε Draw και περίμενε
μετάδωσε Next generation
καθάρισε
επανάλαβε 5
  μετάδωσε Draw και περίμενε
  μετάδωσε Next generation
  μετάδωσε Draw και περίμενε
  μετάδωσε Next generation
  
```



variables : F
 constants : + , - , [,]
 start : - F-
 rules : $F \rightarrow F+F-F-F+F-F$
 Angle : 90°
 Direction : 72°

3.11. Kite and dart tiling

Αυτός ο τύπος επικάλυψης χρησιμοποιεί τετράπλευρα που ονομάζονται kite (χαρταετός) και dart (βέλος) τα οποία όταν συνδεθούν σχηματίζουν ρόμβους. Και τα δύο είναι αποτέλεσμα της σύνθεσης δύο τριγώνων που ονομάζονται τρίγωνα Robinson.



Ο χαρταετός είναι ένα τετράπλευρο του οποίου οι τέσσερις εσωτερικές γωνίες είναι 72° , 72° , 72° και 144° . Ο αετός μπορεί να διχοτομείται κατά μήκος του άξονα συμμετρίας για να σχηματίσει ένα ζεύγος από οξυγώνια τρίγωνα Robinson (με γωνίες 36° , 36° και 72° και 216°).

Το βέλος είναι ένα μη κυρτό τετράπλευρο του οποίου οι τέσσερις εσωτερικές γωνίες είναι 36° , 72° , 36° και 216° . Το βέλος μπορεί να διχοτομείται κατά μήκος του άξονα συμμετρίας για να σχηματίσει ένα ζεύγος από αμβλειγώνια τρίγωνα Robinson (με γωνίες των 36, 36 και 108 μοίρες) [13]

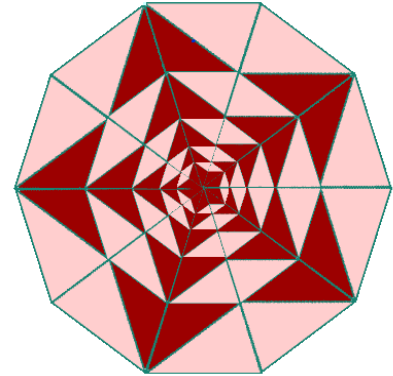
Ένα απεριοδικό dart tiling είναι το έργο [dart-til](http://scratch.mit.edu/projects/marynasta/2959475) που με τη κατάλληλη σύνθεση των τριγώνων Robinson σχηματίζονται ρόμβοι και βέλη. (<http://scratch.mit.edu/projects/marynasta/2959475>)

```

όταν στο  γίνει κλικ
  απόκρυψη
  καθάρισε
  όρισε το μέγεθος σε 80 %
  πήγαινε στο x: 0 y: 0
  αλλαγή σε ενδυμασία 11
  εμφάνισε
  σφραγίδα
  περίμενε 0.2 δευτερόλεπτα
  επανάλαβε 5
    αλλαγή σε ενδυμασία 11
    στρίψε 36 μοίρες
    σφραγίδα
    περίμενε 0.4 δευτερόλεπτα
    αλλαγή σε ενδυμασία 21
    στρίψε 36 μοίρες
    σφραγίδα
    περίμενε 0.4 δευτερόλεπτα
  περίμενε 1 δευτερόλεπτα
  επανάλαβε 5
    μετάδωσε giro
    περίμενε 1 δευτερόλεπτα
  περίμενε 1 δευτερόλεπτα
  επανάλαβε 5
    όρισε το μέγεθος σε 50 %
    μετάδωσε giro
    περίμενε 1 δευτερόλεπτα
  σφραγίδα
  περίμενε 1 δευτερόλεπτα
  επανάλαβε 5
    όρισε το μέγεθος σε 30 %
    μετάδωσε giro
    περίμενε 1 δευτερόλεπτα
  σφραγίδα
  περίμενε 1 δευτερόλεπτα
  επανάλαβε 5
    όρισε το μέγεθος σε 18 %
    μετάδωσε giro
    περίμενε 1 δευτερόλεπτα
  σφραγίδα
  περίμενε 1 δευτερόλεπτα
  επανάλαβε 8
    όρισε το μέγεθος σε 11 %
    μετάδωσε giro
    περίμενε 1 δευτερόλεπτα
  σφραγίδα
  περίμενε 3 δευτερόλεπτα
  
```

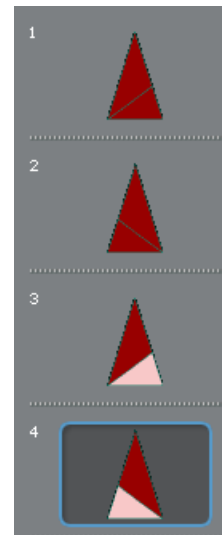
```

όταν λάβω giro
  αλλαγή σε ενδυμασία 12
  στρίψε 36 μοίρες
  σφραγίδα
  περίμενε 0.4 δευτερόλεπτα
  αλλαγή σε ενδυμασία 22
  στρίψε 36 μοίρες
  σφραγίδα
  περίμενε 0.4 δευτερόλεπτα
  
```



Για τη παραγωγή αυτού του tiling, χρησιμοποίησα ένα αντικείμενο με 4 διαφορετικές ενδυμασίες. Το αντικείμενο αυτό είναι ένα τρίγωνο Robinson που αποτελείται από δύο σχηματισμένα μικρότερα τρίγωνα Robinson.

Τοποθετώ τη κορυφή του τριγώνου στο σημείο (0,0) και με την εντολή σφραγίδα αποτυπώνεται το τρίγωνο στο επίπεδο. Στη συνέχεια το τρίγωνο στρέφεται κατά 36° και αποτυπώνεται και πάλι. Επαναλαμβάνεται η διαδικασία 5 φορές εναλλάσσοντας τις δύο πρώτες ενδυμασίες ώστε να σχηματιστεί ένα δεκάγωνο. Στην επόμενη επανάληψη εναλλάσσονται οι δύο επόμενες ενδυμασίες.



Η διαδικασία συνεχίζεται αλλάζοντας το μέγεθος το τριγώνου. Η κορυφή του τριγώνου βρίσκεται πάντα στο σημείο (0,0).

Παρουσιάζει συμμετρία ανάκλασης και πενταπλή συμμετρία. Είναι αυτοπαρόμοιο, δηλαδή τα ίδια μοτίβα εμφανίζονται σε όλο και μικρότερες κλίμακες.

Βιβλιογραφία-πηγές

1. Rick Diefenderfer , *Applied to the Structure & Strategy of a Cell-Based Church System* , Creating Christian Communities 2009.ppt
2. Christopher G. Jennings, *The Tickle Trunk Lindenmayer Systems*, Graphics, Visualization, and Usability Lab Department of Computing Science Simon Fraser University (<http://cgjennings.ca/toybox/lsystems/index.html>)
3. William McWorter, *Fractint L-System Spacefilling Curves*, mcworter@midohio.net version 1.4 January 1997,
<http://www.nahee.com/spanky/www/fractint/lsys/truefractal.html>
4. Gabriela Ochoa, *An Introduction to Lindenmayer Systems*, School of Cognitive and Computing Sciences The University of Sussex http://www.biologie.uni-hamburg.de/b-online/e28_3/lsys.html
5. Umberto Roncoroni. *LSystems Tutorial For Artists*, First Published: August 2007. Updated: March, 2008
6. " Area and Perimeter of a Sierpinski triangle " youtube , *Area and Perimeter of a Sierpinski triangle* (ανάκτηση 23 Οκτωβρίου 2012),
<https://www.youtube.com/watch?v=5pLxMnbtAw>
7. "Dragon curve", *Wikipedia, The Free Encyclopedia*, (ανάκτηση 20 Οκτωβρίου 2012), http://en.wikipedia.org/wiki/Dragon_curve
8. "Hilbert", *Wikipedia, The Free Encyclopedia*, (ανάκτηση 11 Μαρτίου 2013), http://en.wikipedia.org/wiki/Hilbert_curve
9. "Koch snowflake", *Wikipedia, The Free Encyclopedia*, (ανάκτηση 20 Οκτωβρίου 2012), http://en.wikipedia.org/wiki/Koch_snowflake
10. "Kolam", *Wikipedia, The Free Encyclopedia*, (ανάκτηση 11 Μαρτίου 2013), <http://en.wikipedia.org/wiki/Kolam>
11. "L-system", *Wikipedia, The Free Encyclopedia*, (ανάκτηση 5 Νοεμβρίου 2012), <http://en.wikipedia.org/wiki/L-system>
12. "Peano curve", *Wikipedia, The Free Encyclopedia*, (ανάκτηση 11 Μαρτίου 2013), http://en.wikipedia.org/wiki/Peano%27s_curves
13. "Penrose tiling", *Wikipedia, The Free Encyclopedia*, (ανάκτηση 20 Οκτωβρίου 2012), http://en.wikipedia.org/wiki/Penrose_tiling
14. "Scratch (programming language)" *Wikipedia, The Free Encyclopedia*, (ανάκτηση 5 Μαρτίου 2013), [http://en.wikipedia.org/wiki/Scratch_\(programming_language\)](http://en.wikipedia.org/wiki/Scratch_(programming_language))

15. "Scratch" *Wikipedia, The Free Encyclopedia*, (ανάκτηση 5 Μαρτίου 2013), <http://goo.gl/BaonK>
16. "Scratched" *youtube*, *δικτυακός τόπος* (ανάκτηση 8 Μαρτίου 2013), <http://scratched.media.mit.edu/about>
17. "Scratch 2.0" *youtube*, *Scratch 2.0 Preview* (ανάκτηση 8 Μαρτίου 2012), <https://www.youtube.com/watch?v=z4UIf7gDojg>
18. "Sierpinski_triangle" *Wikipedia, The Free Encyclopedia*, (ανάκτηση 5 Μαρτίου 2013), http://en.wikipedia.org/wiki/Sierpinski_triangle
19. "Επιμορφωτικό υλικό για την επιμόρφωση των εκπαιδευτικών στα ΚΣΕ " *Ινστιτούτο Τεχνολογίας Υπολογιστών*, Τομέας Επιμόρφωσης και Κατάρτισης, Πάτρα, Δεκέμβριος 2010
20. Przemyslaw Prusinkiewicz, Aristid Lindenmayer. *The Algorithmic Beauty of Plants*, published by Springer-Verlag, New York, in 1990
<http://algorithmicbotany.org/papers/abop/abop.pdf>

