





**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
**ΣΧΟΛΗ ΑΓΡΟΝΟΜΩΝ ΚΑΙ ΤΟΠΟΓΡΑΦΩΝ ΜΗΧΑΝΙΚΩΝ**  
**ΤΟΜΕΑΣ ΤΟΠΟΓΡΑΦΙΑΣ**  
**ΕΡΓΑΣΤΗΡΙΟ ΓΕΝΙΚΗΣ ΓΕΩΔΑΙΣΙΑΣ**

**ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΠΑΡΑΓΩΓΗΣ ΚΑΙ  
ΧΩΡΟΘΕΤΗΣΗΣ ΤΥΧΑΙΩΝ ΣΗΜΕΙΩΝ ΜΕ ΤΗ  
ΧΡΗΣΗ ΜΑΘΗΜΑΤΙΚΩΝ ΜΟΝΤΕΛΩΝ**

**Διπλωματική Εργασία**

του

**Σπυρίδωνος Κονταλή**

**Επιβλέπων:**  
**Γεώργιος Πανταζής**  
**Αναπληρωτής Καθηγητής ΕΜΠ**

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή:

*(Υπογραφή)*

*(Υπογραφή)*

*(Υπογραφή)*

Γεώργιος Πανταζής  
Αν. Καθηγητής ΕΜΠ

Ευαγγελία Λάμπρου  
Αν. Καθηγήτρια ΕΜΠ

Βασίλειος Βεσκούκης  
Επίκουρος Καθηγητής ΕΜΠ

**Αθήνα Οκτώβριος 2017**



(Υπογραφή)

.....

**ΣΠΥΡΙΔΩΝ ΚΟΝΤΑΛΗΣ**

Διπλωματούχος Αγρονόμος και Τοπογράφος Μηχανικός ΕΜΠ

Copyright © Σπυρίδων Κονταλής, 2017

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας διπλωματικής εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν στη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.





## ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα στην αρχή του παρόντος τεύχους να ευχαριστήσω όλους όσους συνέβαλαν, ο καθένας από την πλευρά του, να βρίσκομαι στην ευχάριστη θέση της εκπόνησης και παρουσίασης της διπλωματικής εργασίας.

Ευχαριστώ θερμά τον κύριο Γεώργιο Πανταζή, αναπληρωτή Καθηγητή της Σχολής Αγρονόμων & Τοπογράφων Μηχανικών ΕΜΠ, επιβλέποντα της παρούσας διπλωματικής εργασίας, που μου έδωσε την ευκαιρία να εργασθώ και να εντρυφήσω στο συγκεκριμένο θέμα, με τις συνεχείς οδηγίες παροτρύνσεις επινοήσεις σε όλη τη διάρκεια της εκπόνησης.

Θέρμες ευχαριστίες στην κυρία Ευαγγελία Λάμπρου, αναπληρώτρια Καθηγήτρια της Σχολής Αγρονόμων & Τοπογράφων Μηχανικών ΕΜΠ, για το αμείωτο ενδιαφέρον, τις συμβουλές και υποδείξεις στην εξέλιξη και πρόοδο της διπλωματικής.

Επίσης ευχαριστώ τον κύριο Βασίλειο Βεσκούκη, αναπληρωτή Καθηγητή της Σχολής Αγρονόμων & Τοπογράφων Μηχανικών ΕΜΠ, που με τη συνεργασία του, το ενδιαφέρον και τις ιδέες του συνέβαλε στην ολοκλήρωση αυτής της διπλωματικής.

Ευχαριστώ επίσης την οικογένειά μου και στενούς φίλους για την ηθική συμπαράσταση που είχα μέχρι σήμερα. Ιδιαίτερα δε, θέλω να ευχαριστήσω τον αδερφό μου Παναγιώτη για τη συνεχή και ουσιαστική βοήθεια που μου προσέφερε.

Σπυρίδων Κονταλής

## ΠΡΟΛΟΓΟΣ

Η Σχολή Αγρονόμων & Τοπογράφων Μηχανικών (ΣΑΤΜ) ιδρύθηκε το 1917 με την επωνυμία "Ανωτάτη Σχολή Τοπογράφων Μηχανικών" και στη συνέχεια "Ανωτάτη Σχολή Αγρονόμων & Τοπογράφων Μηχανικών". Στο πέρασμα των 100 χρόνων ο τοπογράφος μηχανικός εξελίχθηκε και μέσα από τις έρευνες και τις μελέτες πέρασε στον ψηφιακό χώρο της γεωπληροφορικής, κάνοντας τους ηλεκτρονικούς υπολογιστές όχι μόνο χρήσιμο αλλά και απαραίτητο εργαλείο στη δούλεψή του. Η εξέλιξη αυτή συνεχίζεται λαμβάνοντας ερεθίσματα από τις ανάγκες σε καθημερινά ζητήματα που απαιτούν λύση.

Ο Αγρονόμος & Τοπογράφος Μηχανικός τόσο με τις εφαρμογές των γνώσεών του στην ύπαιθρο όσο και στα εργαστήρια, σε συνδυασμό και με τον τομέα της γεωπληροφορικής και του προγραμματισμού θεωρείται ο καταλληλότερος επιστήμονας για την εύρεση λύσεων σε θέματα χωροθέτησης και ο πλέον ειδικός να μελετά από όλες τις πλευρές τον χώρο, λαμβάνοντας την ευθύνη της χωροθέτησης, δηλαδή να καθορίζει επακριβώς να εγκρίνει και να οργανώνει τον χώρο αυτόν.

Στην εξέλιξη αυτή βοήθησε τα μέγιστα η πληροφορική που έγινε ένα σημαντικό εργαλείο στα χέρια του Τοπογράφου Μηχανικού. Ο συνδυασμός γνώσεων πληροφορικής και τοπογραφίας έχει μια δυναμική εξέλιξη και λύνει δύσκολα προβλήματα που τις προηγούμενες δεκαετίες φαινόταν άλυτα. Έτσι μπορεί ο τοπογράφος να γίνει ο επιστήμονας που θα δώσει λύσεις και σε παρεμφερείς επιστήμες και σε πάρα πολλούς τομείς του δημόσιου και ιδιωτικού τομέα, όπου άλλοι δε θα μπορούσαν.

Η γρήγορη ανάπτυξη και διάδοση των υπολογιστών και της τεχνολογίας των πληροφοριών δημιούργησε και νέα συστήματα τεχνολογίας, υλικού (hardware) και λογισμικού (software) και την ενσωμάτωσή τους σε νέες τεχνολογίες, φτάνοντας έτσι σε πολύ μεγάλες ταχύτητες επεξεργασίας δεδομένων και στις πιο σύγχρονες γλώσσες προγραμματισμού με πολλές δυνατότητες.

Ένα project έρχεται να ολοκληρωθεί, κάτω από τον συνδυασμό των νέων τεχνολογιών, της σκέψης του μηχανικού, τα μαθηματικά θεωρήματα δημιουργώντας ένα λογισμικό παραγωγής, διαχείρισης και χωροθέτησης τυχαίων σημείων.

## EΙΚΟΝΕΣ

Εικόνα 1.1: Δασική έκταση με τα καταφύγια να σημειώνονται με κόκκινο .....	13
Εικόνα 1.2: Απλοποιημένη αναπαράσταση των καταφυγίων .....	13
Εικόνα 1.3: Η μεσοκάθετος του AB διαχωρίζει το επίπεδο σε σημεία πλησιέστερα στο A (μπλέ) και σε σημεία πλησιέστερα στο B.....	14
Εικόνα 2.1: Τυπική μορφή του γραφικού περιβάλλοντος της MATLAB .....	28
Εικόνα 2.2: Τυπική μορφή του γραφικού περιβάλλοντος της WPF .....	30
Εικόνα 2.3: Τυπική μορφή του γραφικού περιβάλλοντος της C++ .....	32
Εικόνα 2.4: Τυπική μορφή του γραφικού περιβάλλοντος της Java .....	34
Εικόνα 3.1: Εκτέλεση της συνάρτησης System.Math.Rnd (για 10000 νούμερα).....	44
Εικόνα 3.2: Εκτέλεση της κλάσης System.Random (για 10000 νούμερα).....	44
Εικόνα 3.3: Εκτέλεση της κλάσης Security.Cryptographic (για 10000 νούμερα).....	45
Εικόνα 3.4: Διάγραμμα χρόνου-πλήθους αριθμών.....	46
Εικόνα 3.5: Παράδειγμα παραγωγής τυχαίων συντεταγμένων .....	47
Εικόνα 3.6: Παράδειγμα παραγωγής συντεταγμένων σταθερής απόστασης .....	49
Εικόνα 3.7: Παράδειγμα παραγωγής τυχαίων συντεταγμένων .....	50
Εικόνα 3.8: Παράδειγμα δημιουργίας τυχαίων κέντρων.....	51
Εικόνα 3.9: Παράδειγμα δημιουργίας νέων τυχαίων κέντρων.....	52
Εικόνα 3.10: Παράδειγμα τριγωνισμού Delaunay .....	53
Εικόνα 3.11: Παράδειγμα τριγωνισμού Delaunay στον κάρναβο.....	53
Εικόνα 3.12: Παράδειγμα δημιουργίας πολυγώνων Voronoi .....	54
Εικόνα 3.13: Παράδειγμα δημιουργίας πολυγώνων Voronoi στον κάρναβο .....	54

## ΠΙΝΑΚΕΣ

Πίνακας 2.1: Συνοπτική αξιολόγηση γλωσσών προγραμματισμού .....	36
Πίνακας 2.2: Αξιολόγηση γλωσσών προγραμματισμού σε ραβδόγραμμα .....	37
Πίνακας 3.1: Βοηθητικές εξωτερικές βιβλιοθήκες που εφαρμόστηκαν ..	39

## ΣΧΗΜΑΤΑ

Σχήμα 1.1: Ο αντίστοιχος τριγωνισμός Delaunay .....	9
Σχήμα 1.2: Διάγραμμα Voronoi .....	10
Σχήμα 1.3: Προσθήκη σημείου $\Gamma$ σε υπάρχον διάγραμμα. οι μεσοκάθετες εμφανίζονται ως ευθείες .....	14
Σχήμα 1.4: Προσθήκη σημείου $\Gamma$ σε υπάρχον διάγραμμα. οι μεσοκάθετες εμφανίζονται ως ημιευθείες.....	14
Σχήμα 1.5: Αλληλουχία βημάτων για την προσθήκη ενός νέου σημείου σε υπάρχον διάγραμμα.....	15
Σχήμα 1.6: Απόδειξη πως τα σημεία αριστερά της μεσοκαθέτου είναι πλησιέστερα στο σημείο $A$ .....	17
Σχήμα 1.7: Απεικόνιση των βασικών ιδιοτήτων ενός διαγράμματος Voronoi.....	18
Σχήμα 1.8: Διάγραμμα Voronoi (συνεχής γραμμές) και η αντίστοιχη Τριγωνοποίηση Delaunay (διακεκομμένες γραμμές). .....	20
Σχήμα 1.9: Τριγωνοποίηση Delaunay με περιορισμό .....	20
Σχήμα 2.1: διάγραμμα ροής, το οποίο απεικονίζει τον αλγόριθμο ελέγχου και επισκευής μιας λάμπας η οποία δεν λειτουργεί. ....	21
Σχήμα 3.1: Διάγραμμα ροής γενικής ακολουθίας 1ης φάσης του προγράμματος.....	40
Σχήμα 3.2: Διάγραμμα ροής γενικής ακολουθίας 2ης φάσης του προγράμματος.....	41
Σχήμα 3.3: Διάγραμμα ροής γενικής ακολουθίας 3ης φάσης του προγράμματος.....	42
Σχήμα 3.4: Στάδια βελτιστοποίησης πολυγώνων Voronoi .....	55
Σχήμα 4.1: Εφαρμογή αλγορίθμου για 1000 σημεία 20 κέντρα .....	58
Σχήμα 4.2: Εφαρμογή αλγορίθμου για 1000 σημεία 200 κέντρα .....	58
Σχήμα 4.3: Εφαρμογή αλγορίθμου για 1000 σημεία 50 κέντρα “no actual point” .....	59
Σχήμα 4.4: Εφαρμογή αλγορίθμου για 1000 σημεία 100 κέντρα “no actual point” .....	59

## ΠΕΡΙΛΗΨΗ

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός λογισμικού για την παραγωγή και χωροθέτηση τυχαίων σημείων με τη χρήση μαθηματικών μοντέλων. Το λογισμικό αυτό αποτελεί την υποδομή για την εξέλιξη της αυτοματοποιημένης χωροθέτησης.

Η εργασία αποσκοπεί στη δημιουργία μιας ρουτίνας κατά την οποία ο χρήστης θα μπορεί να κάνει τον έλεγχο της τυχειότητας και του χρόνου παραγωγής τυχαίων αριθμών και συντεταγμένων. Ακολουθώς γίνεται η χωροθέτησή τους με τη χρήση των μαθηματικών μοντέλων των Delaunay και Voronoi.

Αναλυτικότερα στο πρώτο κεφάλαιο παρουσιάζεται ο κλάδος της Υπολογιστικής Γεωμετρίας και οι αντίστοιχες εφαρμογές της, τα μαθηματικά μοντέλα του τριγωνισμού Delaunay και του διαγράμματος Voronoi και έπειτα η έννοια της χωροθέτησης και η χρησιμότητα της. Τέλος γίνεται η εισαγωγή στο πρόβλημα που καλείται η παρούσα διπλωματική εργασία να επιλύσει.

Στο δεύτερο κεφάλαιο, μετά την εισαγωγή των βασικών εννοιών (αλγόριθμος και γλώσσα προγραμματισμού), μελετιούνται τέσσερις διαφορετικές γλώσσες προγραμματισμού, η MatLab, η C++, η Java και η WPF (Visual Basic. NET + XAML). Συγκρίνονται μεταξύ τους εφαρμόζοντας πολυκριτηριακή ανάλυση, χρησιμοποιώντας βασικά κριτήρια μιας γλώσσας προγραμματισμού.

Στο τρίτο κεφάλαιο, μετά την επιλογή της γλώσσας προγραμματισμού WPF ως καταλληλότερη, αναλύονται τα τρία στάδια του λογισμικού. Στο πρώτο γίνεται ο έλεγχος (τυχειότητας και χρόνου) τριών διαφορετικών αλγορίθμων παραγωγής τυχαίων αριθμών. Στο δεύτερο στάδιο γίνεται η παραγωγή τυχαίων σημείων, με την επιλεγμένη  $\sigma$ , με επιλογές και περιορισμούς από το χρήστη ή η απευθείας εισαγωγή αρχείου με γνωστές συντεταγμένες. Στο τρίτο στάδιο πραγματοποιείται η εφαρμογή των μαθηματικών μοντέλων του Delaunay και του Voronoi και εμφανίζονται αντίστοιχα η τριγωνοποίηση και το διάγραμμα, όπου με τις συνεχείς επαναλήψεις το αποτέλεσμα βελτιστοποιείται.

Τέλος στο τέταρτο κεφάλαιο, αναφέρονται ενδεικτικά κάποια από τα αποτελέσματα της εφαρμογής του αλγορίθμου. Ενεργοποιώντας τους περιορισμούς κατά την παραγωγή των συντεταγμένων, παρατηρήθηκε ότι η εφαρμογή καθυστερεί, καθώς παράλληλα με τους ελέγχους, δημιουργεί και τη γραφική αναπαράσταση του κάθε σημείου. Επίσης με διαφορετική επιλογή αρχικών κέντρων των ίδιων σημείων, παράγονται διαφορετικά διαγράμματα Voronoi μετά το τέλος της βελτιστοποίησης καθώς επίσης και όταν δεν είναι επιλεγμένο το πεδίο ενεργοποίησης του πραγματικού σημείου, το διάγραμμα Voronoi βελτιστοποιείται πλήρως.

Στη συνέχεια παρουσιάζονται κάποιες εφαρμογές του λογισμικού και γίνονται προτάσεις για μελλοντικές βελτιώσεις.

Στο παράρτημα της εργασίας, επισυνάπτεται ο πηγαίος κώδικας γραφιστικής απεικόνισης της εφαρμογής καθώς και ο πηγαίος κώδικας του αλγορίθμου της.

## **Software development for production and positioning random points using mathematic models**

### **ABSTRACT**

The object of this thesis is the development of a software for the production and placement of random points (coordinates), using mathematical models. This software is the base for the development of the automated placement.

This project aims to create a routine in which the user can compare the randomness and the duration of producing random numbers and coordinates. Subsequently, their placement is made using the mathematical models of Delaunay and Voronoi.

The first chapter presents the Computational Geometry Theory and its corresponding applications, the mathematical models of the Delaunay triangulation and the Voronoi tessellation. Also presented the concept of spatial model and its use. At the end of this chapter, is introduced the task which this diploma thesis is called to resolve.

In the second chapter, introduced the basic concepts (algorithm and programming language) and the differences of four programming languages which have been studied (MatLab, C ++, Java and WPF (Visual Basic .NET + XAML). These are compared to each other by applying multi-criteria analysis using basic programming language criteria.

As the WPF language selected as more appropriate, the third chapter, covers the three stages of the software. The first stage (randomness and time) demonstrates the three random generation algorithms. In the second stage, user can either create random points with the selected random generation algorithm of the previous stage with limitations, or he direct import a file with known coordinates. In the third stage, the mathematical models of Delaunay and Voronoi applied to the coordinates (points) to create the placement and optimize the result.

In the last chapter presented the software execution conclusions. By activating the limitations in the coordinates production, the application was delayed, as it makes the graphical representation of each point in parallel with the coordinates audit. Also, it is noticed that with alternative selection of the initial Voronoi centers, the optimized result returns alternative Voronoi diagram. Additionally an improved visual result displayed when the actual point parameter is enabled. In the end of this chapter presented software applications and suggestions for future improvements.



In the appendix of the paper is attached, the source code of the graphic user interface of the software and the source code of its algorithm.

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΥΧΑΡΙΣΤΙΕΣ.....	I
ΠΡΟΛΟΓΟΣ.....	II
ΕΙΚΟΝΕΣ.....	III
ΠΙΝΑΚΕΣ.....	III
ΣΧΗΜΑΤΑ.....	IV
ΠΕΡΙΛΗΨΗ.....	V
Abstract.....	VII
ΕΙΣΑΓΩΓΗ.....	3
Κεφάλαιο 1. Θεωρία Υπολογιστικής Γεωμετρίας.....	5
1.1 Γενικά.....	5
1.2 Ο κλάδος της Υπολογιστικής Γεωμετρίας.....	5
1.3 Διαγράμματα Voronoi.....	8
1.4 Τριγωνισμός Delaunay.....	9
1.5 Χωροθέτηση.....	11
1.5.1 Το πρόβλημα.....	13
Κεφάλαιο 2. Αλγόριθμος – Γλώσσα προγραμματισμού.....	21
2.1 Αλγόριθμος.....	21
2.2 Γλώσσα προγραμματισμού.....	24
2.2.1 Κατηγοριοποίηση γλωσσών προγραμματισμού.....	24
2.2.2 MatLab.....	27
2.2.3 WPF (Visual Basic. NET + XAML).....	29
2.2.4 C++.....	31
2.2.5 Java.....	33
2.3 Επιλογή γλώσσας προγραμματισμού.....	35
Κεφάλαιο 3. Αναλυση ΛΟΓΙΣΜΙΚΟΥ.....	39
3.1 Γενικά.....	39
3.2 Έλεγχος – Επιλογή συνάρτησης τυχαίων σημείων.....	43
3.2.1 Έλεγχος συναρτήσεων-κλάσεων.....	43
3.2.2 Δημιουργία – Εισαγωγή Σημείων.....	47

3.2.2.1 Δημιουργία τυχαίων συντεταγμένων.....	47
3.2.2.2 Δημιουργία συντεταγμένων σταθερής απόστασης (Grid) ..	49
3.2.2.3 Εισαγωγή Σημείων.....	50
3.2.3 Παραγωγή Διαγράμματος Voronoi .....	50
Κεφάλαιο 4. Συμπερασματα – Προτασεις.....	57
4.1 Συμπεράσματα .....	57
Βιβλιογραφια .....	61
Ιστογραφια.....	63
Παράρτημα I (Πηγαίος κώδικας XAML).....	65
Παράρτημα II (Πηγαίος κώδικας VB.NET).....	79
Παράρτημα III “ΕΓΧΕΙΡΙΔΙΟ .....	115
A. Παραγωγή Τυχαίων Σημείων .....	117
B. Δημιουργία Πλέγματος .....	118
B. Εισαγωγή σημείων .....	119

## ΕΙΣΑΓΩΓΗ

Στο επιχειρηματικό περιβάλλον, ένα ιδιαίτερα σημαντικό και συνηθισμένο πρόβλημα που αντιμετωπίζεται στη διαδικασία λήψης αποφάσεων, είναι η χωροθέτηση των εγκαταστάσεων. Η επιλογή δηλαδή μίας ή και περισσοτέρων εγκαταστάσεων εξυπηρέτησης, που μας δείχνει την αποτελεσματικότητα του στρατηγικού σχεδιασμού.

Στον τομέα της χωροταξίας περιλαμβάνονται ένα πλήθος εργασιών που αφορούν τη σχεδιαστική διαδικασία, οι οποίες κυμαίνονται από τις παρυφές του φυσικού σχεδιασμού έως τις παρυφές της περιφερειακής πολιτικής. Σημειώνεται ότι οι συγκεκριμένες μορφές σχεδιασμού είναι ανομοιογενείς, ενώ διακρίνονται από ορισμένες κοινές παραμέτρους, όπως η σχετικά μεγάλη κλίμακα της περιοχής αναφοράς, η χωρική προσέγγιση, ο επιτελικός χαρακτήρας και η ύπαρξη ρυθμίσεων κατευθυντήριου και τοπολογικού τύπου, παρά κανονιστικού και χωροθετικά απόλυτα ακριβούς χαρακτήρα.

Σε κάθε περίπτωση θα πρέπει να σημειωθεί ότι το επάγγελμα του τοπογράφου στον τομέα της χωροταξίας, διακρίνεται για το διεπιστημονικό χαρακτήρα του, ο οποίος συνδυάζει τις τεχνικές επιστήμες με την παροχή ουσιαστικών γνώσεων σε θέματα οικονομίας και ανάπτυξης, περιβάλλοντος, υποδομών, διοίκησης, καθώς και σε προηγμένα εργαλεία, μεθόδους και τεχνικές που βασίζονται στη χρήση ηλεκτρονικών υπολογιστών. Ειδικότερα όμως, ένας τοπογράφος είναι καλός γνώστης των θεματικών πεδίων των μαθηματικών και της πληροφορικής. Των ζητημάτων αστικής γεωγραφίας και περιβάλλοντος, της κοινωνιολογίας του αστικού χώρου. Των θεμάτων οικονομικής και περιφερειακής ανάπτυξης, της πολεοδομίας.

Αντικείμενο της διπλωματικής εργασίας, είναι η υλοποίηση ενός λογισμικού, το οποίο θα χρησιμοποιεί τα μαθηματικά μοντέλα του Delaunay και του Voronoi, με μια πρόσθετη μεθοδολογία, αυτή της αλλαγής των κέντρων πολυγώνων Voronoi για τη βέλτιστη χωροθέτηση σημείων.



# ΚΕΦΑΛΑΙΟ 1

## ΘΕΩΡΙΑ ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΓΕΩΜΕΤΡΙΑΣ

### 1.1 Γενικά

Η υπολογιστική γεωμετρία προέκυψε από το πεδίο του σχεδιασμού και της ανάλυσης αλγορίθμων στα τέλη της δεκαετίας του 1970. Έχει γίνει μια αναγνωρισμένη επιστημονική περιοχή, με δικά της περιοδικά, συνέδρια και μια μεγάλη κοινότητα ενεργών ερευνητών.

Η επιτυχία του πεδίου ως επιστημονικής περιοχής μπορεί να εξηγηθεί αφενός από την ομοιομορφία των προβλημάτων που μελετήθηκαν και από τις λύσεις που ελήφθησαν και, αφετέρου, από τους πολλούς τομείς εφαρμογών. Για πολλά γεωμετρικά προβλήματα οι αρχικές αλγοριθμικές λύσεις ήταν είτε αργές είτε δύσκολο να κατανοηθούν και να εφαρμοστούν. Τα τελευταία χρόνια έχουν αναπτυχθεί πολλές νέες αλγοριθμικές τεχνικές που βελτίωσαν, έκαναν ταχύτερες και απλοποίησαν πολλές από τις προηγούμενες προσεγγίσεις.

Ξεκινώντας το κεφάλαιο, αναφέρονται σημαντικές εισαγωγικές έννοιες που προσδιορίζουν το αντικείμενο της διπλωματικής εργασίας. Στην πρώτη ενότητα παρουσιάζονται διάφορα γεωμετρικά προβλήματα και εφαρμογές, ενδεικτικά του εύρους της Υπολογιστικής Γεωμετρίας. Οι δύο επόμενες ενότητες παρέχουν ορισμένα βασικά αλγοριθμικά εργαλεία, χρήσιμα στη σχεδίαση και στην ανάλυση γεωμετρικών αλγορίθμων.

### 1.2 Ο κλάδος της Υπολογιστικής Γεωμετρίας

*Υπολογιστική Γεωμετρία* καλείται η επιστημονική περιοχή της Πληροφορικής που μελετά γεωμετρικά προβλήματα και τις αλγοριθμικές τους λύσεις. Περιλαμβάνεται, συνεπώς, ο σχεδιασμός μεθόδων για προβλήματα γεωμετρίας, αλλά και τεχνικές για την αποτελεσματική και σθεναρή υλοποίησή τους. Μία εναλλακτική ονομασία θα ήταν γεωμετρικοί αλγόριθμοι ή αλγοριθμική γεωμετρία. Τα γεωμετρικά αντικείμενα που ενδιαφέρουν είναι κυρίως γραμμικά, δηλ. πολυεδρικά αντικείμενα, τα οποία συνηθέστερα μελετώνται στο επίπεδο. Σε δεύτερο επίπεδο η μελέτη επεκτείνεται σε τρισδιάστατα γραμμικά αντικείμενα και, σπανιότερα, σε περισσότερες από 3 διαστάσεις. Ειδικότερα

εξετάζονται σημεία, ευθύγραμμα τμήματα και ευθείες, πολύγωνα και σύνολα πολυγώνων (πχ. τριγώνων), καθώς και πολύεδρα σε 3 ή περισσότερες διαστάσεις.

Η Υπολογιστική Γεωμετρία εστιάζει σε προβλήματα, όπου η διάσταση του γεωμετρικού χώρου μπορεί να θεωρηθεί σταθερή κι αυτό διότι συνήθως ενδιαφέρουν προβλήματα σε 2 και 3 διαστάσεις. Γι' αυτό το λόγο, η διάσταση του χώρου συνήθως δεν αποτελεί παράμετρο στην ανάλυση της χρονικής και χωρικής πολυπλοκότητας (παρόλο που μερικές φορές η διάσταση μπορεί να είναι αισθητά μεγαλύτερη από 3).

Οι επιστημονικές περιοχές που καλύπτουν προβλήματα γενικής διάστασης είναι η Διακριτή Γεωμετρία για προβλήματα με έντονο διακριτό χαρακτήρα, η Κυρτή Γεωμετρία και η γραμμική Βελτιστοποίηση εφόσον πρόκειται για πολυεδρικά αντικείμενα, και η Αλγεβρική Γεωμετρία.

Αναφέρθηκε πως ο κύριος όγκος δουλειάς μέχρι σήμερα στην Υπολογιστική Γεωμετρία αφορά σε γεωμετρικά αντικείμενα που ορίζονται αλγεβρικά από γραμμικές εξισώσεις. Τα καμπύλα αντικείμενα αποτελούν μία σύγχρονη ερευνητική κατεύθυνση που επεκτείνει τα παλαιότερα προβλήματα ενώ προσθέτει και νέα. Προς αυτή την κατεύθυνση, η Υπολογιστική Γεωμετρία συναντά τον Σχεδιασμό με Υπολογιστή (CAD) και τη γεωμετρική μοντελοποίηση.

Τα αλγοριθμικά προβλήματα αποτελούν, παραδοσιακά, τον κύριο κορμό της Υπολογιστικής Γεωμετρίας. Συγχρόνως, όμως, προκύπτουν και προβλήματα υλοποίησης, που αποτελούν πλέον ένα αναπόσπαστο θέμα έρευνας. Τα τελευταία χρόνια, έχει γίνει μεγάλη προσπάθεια υλοποίησης πολλών αλγορίθμων και συγκέντρωσής τους σε βιβλιοθήκες γεωμετρικού λογισμικού. Οι κυριότερες τέτοιες προσπάθειες είναι οι βιβλιοθήκες CGAL και LEDA.

Στη συνέχεια δίνονται ενδεικτικά ορισμένα παραδείγματα εφαρμογών της υπολογιστικής γεωμετρίας, προκειμένου να γίνει περισσότερο κατανοητή η σημασία της. Μερικές από τις τεχνικές και τα εργαλεία που περιγράφονται παρακάτω, όπως τα πολύγωνα Voronoi, ο τριγωνισμός Delaunay, τα κυρτά (και μη κυρτά) πολύγωνα και η εύρεση του πλησιέστερου γείτονα χρησιμοποιούνται στο σχεδιασμό και στην επίλυση του προβλήματος της χωροθέτησης.

Τοπογραφία: Η αναπαράσταση του αναγλύφου του εδάφους, απασχολούσε ανέκαθεν όσους ασχολούνταν με μετρήσεις της φυσικής γήινης επιφάνειας (ΦΓΕ). Ένας τρόπος είναι η απεικόνιση με ισοϋψείς καμπύλες. Παλαιότερα η σχεδίαση των ισοϋψών γινόταν με τυχαίο

τριγωνισμό των μετρημένων σημείων. Σήμερα πολλά λογισμικά βασίζονται στον τριγωνισμό Delaunay, ο οποίος όπως αποδεικνύεται παρακάτω αποτελεί τον βέλτιστο τριγωνισμό σημείων του χώρου.

Γεωγραφικά Συστήματα Πληροφοριών (GIS): Πολλά προβλήματα τα οποία επιλύονται με τη βοήθεια λογισμικών GIS, ανάγονται σε προβλήματα υπολογιστικής γεωμετρίας. Ενδεικτικά αναφέρονται η αλληλεπίθεση διαφορετικών χαρτογραφικών υποβάθρων, η εύρεση σημείων που βρίσκονται εντός ή εκτός συγκεκριμένης περιοχής, προβλήματα βέλτιστης διαδρομής, χαρτογραφικής γενίκευσης, δόμηση τοπολογίας, χωρική ανάλυση κ.α.

Ρομποτική: Ένα θεμελιώδες πρόβλημα της ρομποτικής αφορά στην πλοήγηση ανάμεσα σε δεδομένα εμπόδια. Έτσι η διαδρομή ενός ρομπότ (ή μη επανδρωμένου οχήματος) από ένα δεδομένο αρχικό σημείο προς ένα τελικό σημείο, εξαρτάται από τα ενδιάμεσα εμπόδια. Αρχικά πρέπει να βρεθεί αν είναι εφικτή η μετακίνηση και έπειτα να υπολογισθεί η βέλτιστη διαδρομή. Η πλοήγηση ανάμεσα σε πολυγωνικά εμπόδια, αντιμετωπίζεται μέσω της διάταξης ευθυγράμμων τμημάτων και μπορεί να προσεγγισθεί και μέσω ενός διαγράμματος Voronoi.

Ορατότητα και δίκτυα: Πολλά προβλήματα μοντελοποιούνται ως προβλήματα ορατότητας ανάμεσα σε εμπόδια. Το γνωστότερο, ίσως, τέτοιο πρόβλημα είναι το πρόβλημα φύλαξης ενός μουσείου, όπου δεδομένων των εμποδίων (τοιχών) ζητείται ο ελάχιστων αριθμός φρουρών (ή αισθητήρων) για την πλήρη οπτική κάλυψη του χώρου.

Το πρόβλημα μπορεί να γενικευθεί και σε θέματα που σχετίζονται με την ασύρματη τηλεπικοινωνιακή κάλυψη μιας περιοχής, καθώς μια περιοχή μπορεί να μοντελοποιηθεί από ένα σύνολο πολυγωνικών γραμμών και να υποδιαιρεθεί σε μικρότερες με τεχνικές δέντρων. Το μοντέλο προϋποθέτει την χωρίς εμπόδια επικοινωνία μεταξύ των σταθμών. Στις περισσότερες περιπτώσεις η πλήρης κάλυψη είναι πρακτικά αδύνατη ή οικονομικά ασύμφορη, συνεπώς το πρόβλημα μετατοπίζεται στην εύρεση των κατάλληλων θέσεων, ώστε η κάλυψη να είναι η μέγιστη δυνατή. Εφαρμογές των προσεγγιστικών μεθόδων, δείχνουν ότι για τις περισσότερες πρακτικές περιπτώσεις, οι προσεγγιστικές λύσεις, είναι ταυτόχρονα και οι βέλτιστες. Σημαντικό πρόβλημα στον τομέα, παραμένει η διατύπωση οποιασδήποτε προσεγγιστικής λύσης για τις περιπτώσεις ελαχιστοποίησης του αριθμού των σταθμών. [Αθανάσιος Ε. Ηλιοδρομίτης, Ανάπτυξη μεθοδολογιών χωροθέτησης & αξιολόγησης ασυρμάτων δικτύων γεωαισθητήρων με αλγορίθμους υπολογιστικής γεωμετρίας].



### 1.3 Διαγράμματα Voronoi

Στην καρδιά των δημοφιλέστερων ερευνητικών θεμάτων της Υπολογιστικής Γεωμετρίας βρίσκεται η θεωρία των διαγραμμάτων Voronoi. Τα διαγράμματα Voronoi παρουσιάζουν μια πληθώρα εφαρμογών που εκτείνονται από τη θεωρία βάσεων δεδομένων μέχρι και τη μοριακή βιολογία, καθώς έχει παρατηρηθεί ότι κάποιοι οργανισμοί στο εμβρυϊκό στάδιο παρουσιάζουν κυτταρική δομή παρόμοια με αυτή του διαγράμματος Voronoi βεβαρυμμένων σημείων.

Τα διαγράμματα Voronoi λόγω της φύσης τους βρίσκουν μεγάλη εφαρμογή σε προβλήματα που απαιτούν κατασκευή καμπύλης, που να περιβάλλει κυρτά πολυγωνικά αντικείμενα, σε βιβλιοθήκες γεωμετρικών εφαρμογών αλλά και σε πολλές διαφορετικές επιστήμες και επιστημονικούς τομείς όπως αρχαιολογία, κρυσταλλογραφία και κρυπτογραφία.

Στη συνέχεια αναφέρονται ορισμένοι από τους κυριότερους τομείς εφαρμογής και ενδεικτικά παραδείγματα από τη διεθνή βιβλιογραφία, όπως είναι:

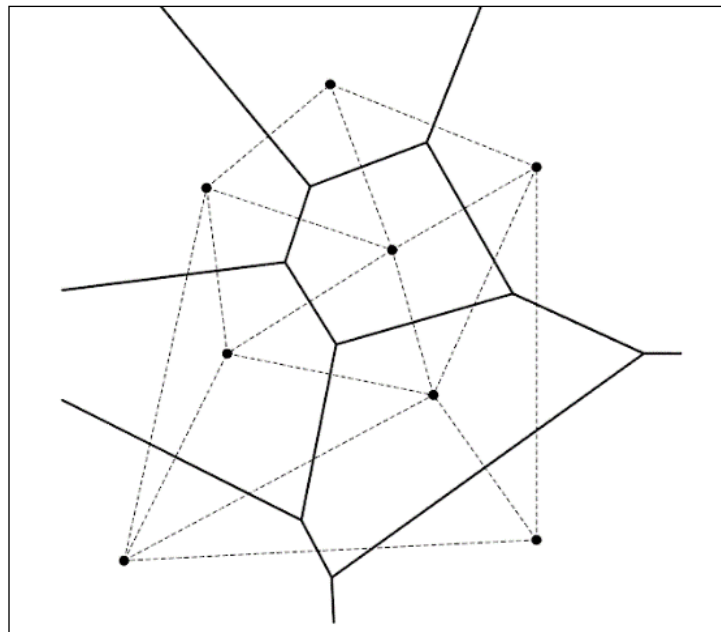
- Εντοπισμός σημείου μέσα σε συγκεκριμένη ακτίνα/κοντινότερου σημείου-γείτονα (Wang M. T., 2016)
- Προβλήματα χωροθέτησης και κάλυψης, πχ. τοποθέτηση μηχανημάτων ATM (Kisore, N. R., & Koteswaraiah, C. B., 2017).
- Έλεγχος ταιριάσματος και αναγνώρισης προτύπων, σε συνδυασμό με τη τριγωνοποίηση Delaunay π.χ. αναγνώριση δακτυλικού αποτυπώματος (Khodadoust, J., & Khodadoust, A. M. 2017).
- Τμηματοποίηση εικόνας (Zhao, Q. H. et. al., 2017)
- Βελτίωση στην απόδοση των καρτών γραφικών (Yuan Z. et al., 2011) (ομοίως με Delaunay)
- Επικύρωση ακρίβειας, πιστότητας λειτουργίας μηχανημάτων, πχ. καρδιακών αξονικών τομογράφων (Ide S. et. al., 2017)
- Προγραμματισμός διαδρομών, κυρίως σε εσωτερικούς χώρους ή σε χώρους με ασύρματα δίκτυα αισθητήρων, για αποφυγή εμποδίων (Chang C. Y et al, 2006), προγραμματισμός διαφόρων ειδών διαδρομών πχ. θαλάσσια σκάφη (Candeloro M. et. al., 2017)

## 1.4 Τριγωνισμός Delaunay

Στο κεφάλαιο αυτό, γίνεται αναφορά στον Delaunay τριγωνισμό ή τριγωνοποίηση. Η τριγωνοποίηση Delaunay είναι ένα από τα θεμελιώδη θέματα στην υπολογιστική γεωμετρία και χρησιμοποιείται σε πολλούς τομείς, όπως:

- η μοντελοποίηση εδάφους (GIS)
- στην επιστημονική οπτικοποίηση δεδομένων και παρεμβολής
- στη ρομποτική
- στην αναγνώριση προτύπων
- στη δημιουργία πλέγματος (Mesh) για μεθόδους πεπερασμένων στοιχείων (finite element methods FEM)
- σε γραφικά υπολογιστών και των πολυμέσων, κλπ.

Η τριγωνοποίηση Delaunay (σχήμα 1.1) είναι τόσο δημοφιλής, διότι τα τρίγωνα που παράγει είναι πολύ υψηλής ποιότητας. Πλησιάζουν κατά βάση το ισόπλευρο τρίγωνο, ενώ δεν εμφανίζονται σκαλινά τρίγωνα. Από μια άποψη, η τριγωνοποίηση Delaunay αποτελεί όπως θα φανεί και παρακάτω την καλύτερη μέθοδο διαμέρισης του χώρου.



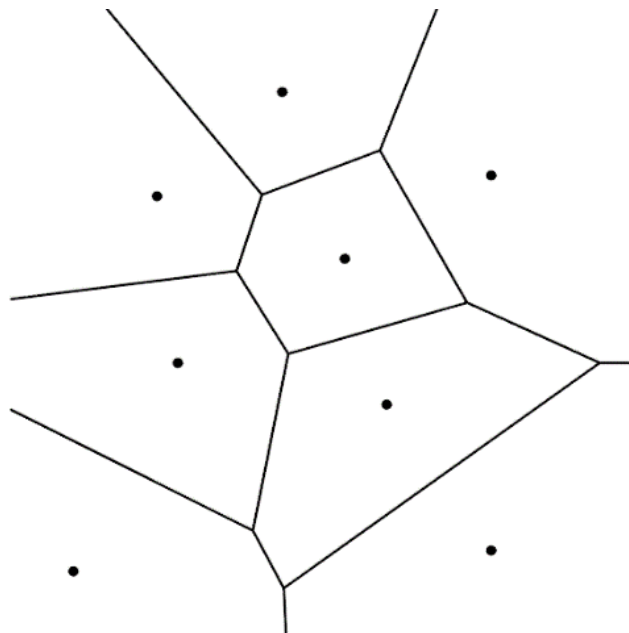
Σχήμα 1.1: Ο αντίστοιχος τριγωνισμός Delaunay

Ο Delaunay τριγωνισμός ενός συνόλου  $P$  σημείων του επιπέδου είναι ένας τριγωνισμός  $DT(P)$  έτσι ώστε κανένα σημείο του  $P$  να μην ανήκει στον περιγεγραμμένο κύκλο οποιουδήποτε τριγώνου του  $DT(P)$ . Η μέθοδος αυτή μεγιστοποιεί την ελάχιστη γωνία όλων των γωνιών των

τριγώνων του τριγωνισμού, αποφεύγοντας την ύπαρξη λεπτών τριγώνων. Ο τριγωνισμός Delaunay αποτελεί μια ειδική περίπτωση τριγωνισμού και είναι το δυικό πρόβλημα των πολυγώνων Voronoi. Πήρε το όνομα του από τον Ρώσο μαθηματικό Boris Nikolaevich Delaunay, μετά την έρευνα του στο αντικείμενο το 1934 [de Berg, Cheong, van Kreveld, & Overmars, 2008].

Σύμφωνα με τον ορισμό των DT, ένα τριγωνικό πλέγμα είναι Delaunay τριγωνισμός, αν όλοι οι περιγεγραμμένοι κύκλοι όλων των τριγώνων είναι άδειοι (δηλαδή δεν υπάρχει κάποιο σημείο του  $P$  στο εσωτερικό τους). Αυτός είναι ο ορισμός του Delaunay τριγωνισμού στον 2Δ χώρο. Στον 3Δ χώρο, ο ορισμός παραμένει ο ίδιος, με εναλλαγή του κύκλου με σφαίρα.

Το αποτέλεσμα του Delaunay τριγωνισμού σχετίζεται με το διάγραμμα Voronoi. Συγκεκριμένα, αν ενωθούν τα κέντρα των κύκλων των περιγεγραμμένων τριγώνων DT ( $P$ ) για ένα διακριτό σύνολο  $P$  παράγεται το διάγραμμα Voronoi.



Σχήμα 1.2: Διάγραμμα Voronoi

Πολλοί αλγόριθμοι έχουν προταθεί για τον υπολογισμό του Delaunay τριγωνικού πλέγματος από σύνολο σημείων στο χώρο. Σε αυτούς περιλαμβάνονται flip αλγόριθμοι, incremental αλγόριθμοι και αλγόριθμοι διαίρει και βασίλευε.

Ο Delaunay τριγωνισμός έχει χρησιμοποιηθεί εκτενώς σε αλγορίθμους που άπτονται πολλών και διαφορετικών επιστημών και

ειδικοτήτων. Στη συνέχεια αναφέρονται ορισμένοι από τους κυριότερους τομείς εφαρμογής και ενδεικτικά παραδείγματα από τη διεθνή βιβλιογραφία, όπως:

- Προβλήματα χωροθέτησης
- Δίκτυα ασύρματων αισθητήρων (Wireless Sensors Network) (Vu, C. T., & Li, Y., 2009)
- Τοποθέτηση καμερών για καταγραφή 3D εικόνας (Rahimian, P., & Kearney, J. K., 2017)
- Αναπαράσταση χαρτών σε τρεις διαστάσεις (3D) για θέαση γεωγραφικών πληροφοριών (De Kok et.al., 2007), δημιουργία επιφανειών σε τρεις διαστάσεις από νέφος σημείων σε τρεις διαστάσεις, πχ. ψηφιακή αναπαράσταση του ανθρώπινου σώματος, μετατροπή εικόνων σε τρεις διαστάσεις (image to mesh) (Feng D. et al., 2015) και διαφόρων ειδών αναπαραστάσεις.
- Ανάλυση εικόνας για συστήματα λήψης απόφασης, πχ. αναγνώριση δακτυλικού αποτυπώματος (Khodadoust, J., & Khodadoust, A. M. 2017).
- Βελτίωση στην απόδοση των καρτών γραφικών (Yuan Z. et al., 2011).

## 1.5 Χωροθέτηση

Αν και αρχικά η ανάλυση της χωροθέτησης αναπτύχθηκε σε ανεξάρτητες εφαρμογές, με μη ενοποιημένη θεωρία (πχ χωροθέτηση πυροσβεστικών σχημάτων (Valinsky, 1955), του σιδηροδρομικού δικτύου (Mansfield και Wein, 1958), κ.α), σήμερα συναντάται στους περισσότερους κλάδους. Αναπτύσσεται τόσο στον ιδιωτικό όσο και στον δημόσιο τομέα. Εφαρμόζεται τόσο σε προβλήματα με πραγματικά δεδομένα όσο και σε προβλήματα που τελικά δεν είναι εφαρμόσιμη η ανάλυση. Μερικά από αυτά είναι η τοποθέτηση αποθηκών, εργοστασίων, νοσοκομείων, σχολείων, λιανική αγορά αγαθών, τοποθέτηση σειρήνων συναγερμού, κεραιών ραντάρ κά. Αυτά ονομάζονται κέντρα παροχής υπηρεσιών, "facilities" και σκοπός είναι να βρεθεί η βέλτιστη θέση για κάθε ένα απ' αυτά. Γενικά η έρευνα των προβλημάτων χωροθέτησης κέντρων παροχής υπηρεσιών συνδέει πολλά ερευνητικά και επιστημονικά πεδία όπως είναι η Τοπογραφία, η Γεωγραφία, τα Μαθηματικά, τα Οικονομικά, το Marketing, η Μηχανολογία, η Επιχειρησιακή Έρευνα, η Πληροφορική και άλλα.

Η ανάλυση χωροθέτησης, "location analysis", εμφανίζεται σε μια

ποικιλία προβλημάτων του ιδιωτικού και το δημόσιου τομέα. Μπορεί δηλαδή να εφαρμοστεί για τη λειτουργία ενός αστυνομικού τμήματος ή ενός πυροσβεστικού σταθμού (μείωση των πιθανοτήτων καταστροφής περιουσίας και απώλειας ζωής) όσον αφορά το δημόσιο τομέα. Από την άλλη στον ιδιωτικό τομέα οι επιχειρήσεις, βιομηχανίες, βιοτεχνίες θα πρέπει να χωροθετήσουν γραφεία, εργοστάσια – κέντρα παραγωγής, αποθήκες, κέντρα διανομής και καταστήματα πώλησης. Στην περίπτωση αυτή η ανάλυση χωροθέτησης βασίζεται σε οικονομικά κριτήρια, κυρίως σε μείωση επενδυτικού και λειτουργικού κόστους και στην αύξηση της ανταγωνιστικότητας αυτής.

Επομένως ο στρατηγικός χωροθετικός σχεδιασμός καθορίζει την επιτυχία ή την αποτυχία των λειτουργιών. Εφαρμόζεται σε πολλά πεδία και με διαφορετικά κριτήρια ανάλογα με την κάθε περίπτωση. Ορισμένα από τα κριτήρια που λαμβάνονται υπόψη είναι:

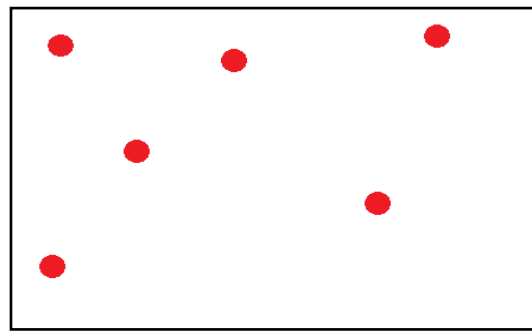
- *Κόστος εγκατάστασης*: το κόστος της επένδυσης για την εγκατάσταση της νέας μονάδας, δηλαδή το κόστος αγοράς ιδιοκτησίας-γης, το κόστος του απαιτούμενου εξοπλισμού, το κόστος οικοδόμησης κ.τ.λ.
- *Κόστος λειτουργίας*: δηλαδή το κόστος ενοικίου ή αποπληρωμής δόσεων δανείου, αμοιβές προσωπικού, συντήρηση εξοπλισμού, παροχές ΔΕΗ, νερό κ.τ.λ.
- *Ζήτηση*: δηλαδή τις προτιμήσεις των καταναλωτών, την ποιότητα, την τιμή του προϊόντος ή της υπηρεσίας, την συχνότητα αγοράς, την αξία για τον πελάτη κ.τ.λ.
- *Ανταγωνισμός*: δηλαδή τα μερίδια της αγοράς, την ένταση του ανταγωνισμού, τις τιμές των ανταγωνιστών, τις στρατηγικές κ.τ.λ.
- *Δυνατότητα πρόσβασης*: δηλαδή την εύκολη πρόσβαση από τους πελάτες και τους προμηθευτές για φόρτωση – εκφόρτωση, χρόνος μετακίνησης των πελατών μέχρι τη μονάδα, τρόπος μεταφοράς των πελατών στη μονάδα, οδικό δίκτυο, διαθέσιμα μέσα μαζικής μεταφοράς κ.τ.λ.

### 1.5.1 Το πρόβλημα

Ας υποθεθεί πως σε ένα δάσος υπάρχουν χτισμένα κάποια καταφύγια για τους επισκέπτες και τους εκδρομείς. Τα καταφύγια αυτά έχουν ως σκοπό να προσφέρουν στέγη και τροφή σε κάποιον ο οποίος θα βρεθεί σε έκτακτη ανάγκη κατά την επίσκεψή του στο δάσος. Η λύση στο πρόβλημα βρίσκεται στην κατασκευή ενός χάρτη ο οποίος θα υποδεικνύει στους εκδρομείς το πλησιέστερο καταφύγιο που μπορούν να προσφύγουν στην περίπτωση αυτή. Στην εικόνα 1 παρουσιάζεται ο χάρτης του δάσους με τα καταφύγια (αριστερά) καθώς και μία απλοποιημένη μορφή του (δεξιά).



*Εικόνα 1.1: Δασική έκταση με τα καταφύγια να σημειώνονται με κόκκινο*

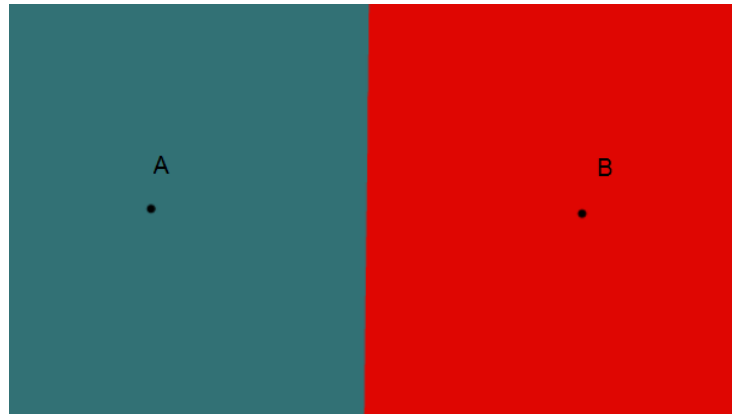


*Εικόνα 1.2: Απλοποιημένη αναπαράσταση των καταφυγίων*

Προς τη λύση του προβλήματος

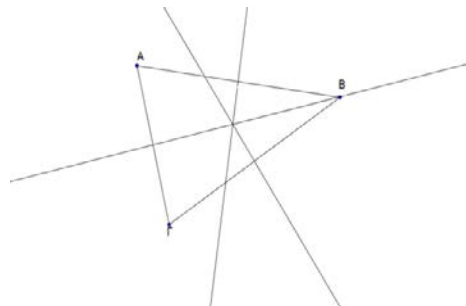
Για να λυθεί το πρόβλημα, θα πρέπει να εντοπιστεί για κάθε σημείο του δάσους ποιο είναι το πλησιέστερο καταφύγιο. Με άλλα λόγια, θα πρέπει να εντοπιστεί για κάθε καταφύγιο η "περιοχή ευθύνης" του, η οποία θα περιλαμβάνει όλα εκείνα τα σημεία που θα απέχουν απόσταση μικρότερη από κάθε άλλο καταφύγιο.

Θα γίνει προσπάθεια να επιλυθεί το πρόβλημα ξεκινώντας από την πιο απλή εκδοχή του. Αυτή είναι όταν στο δάσος υπάρχουν μόνο δύο καταφύγια, έστω A και B (εικόνα 1.3). Σε αυτή την περίπτωση, για να βρεθεί ο γεωμετρικός τόπος των σημείων που απέχουν λιγότερο στο A από ότι στο B αρκεί να σχεδιαστεί η μεσοκάθετος του ευθύγραμμου τμήματος AB. Είναι γνωστό πως τα σημεία της μεσοκαθέτου ισαπέχουν από τα A και B. Όλα τα σημεία που βρίσκονται εκτός της μεσοκαθέτου και προς το A απέχουν λιγότερο από το A παρά από το B και αντίστοιχα τα υπόλοιπα σημεία απέχουν λιγότερο από το B παρά από το A.

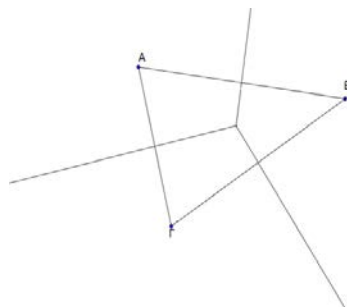


Εικόνα 1.3: Η μεσοκάθετος του  $AB$  διαχωρίζει το επίπεδο σε σημεία πλησιέστερα στο  $A$  (μπλέ) και σε σημεία πλησιέστερα στο  $B$

Στο επόμενο βήμα προστίθεται ένα ακόμα καταφύγιο, έστω  $\Gamma$ . Για να ανανεωθεί ο χάρτης, ακολουθείται η εξής διαδικασία: Εντοπίζεται σε ποια περιοχή ανήκει το  $\Gamma$ . Χωρίς βλάβη της γενικότητας, γίνεται η υπόθεση ότι ανήκει στην περιοχή του καταφυγίου  $A$ . Τότε άγεται η μεσοκάθετος του  $A\Gamma$  χωρίς να επεκταθεί πέρα της μεσοκαθέτου του  $AB$ . Στη συνέχεια άγεται η μεσοκάθετος του τμήματος  $B\Gamma$ , αντίστοιχα χωρίς να επεκταθεί πέρα της μεσοκαθέτου του  $A\Gamma$ . Η διαδικασία και το τελικό σχήμα φαίνονται στο σχήμα 1.3.

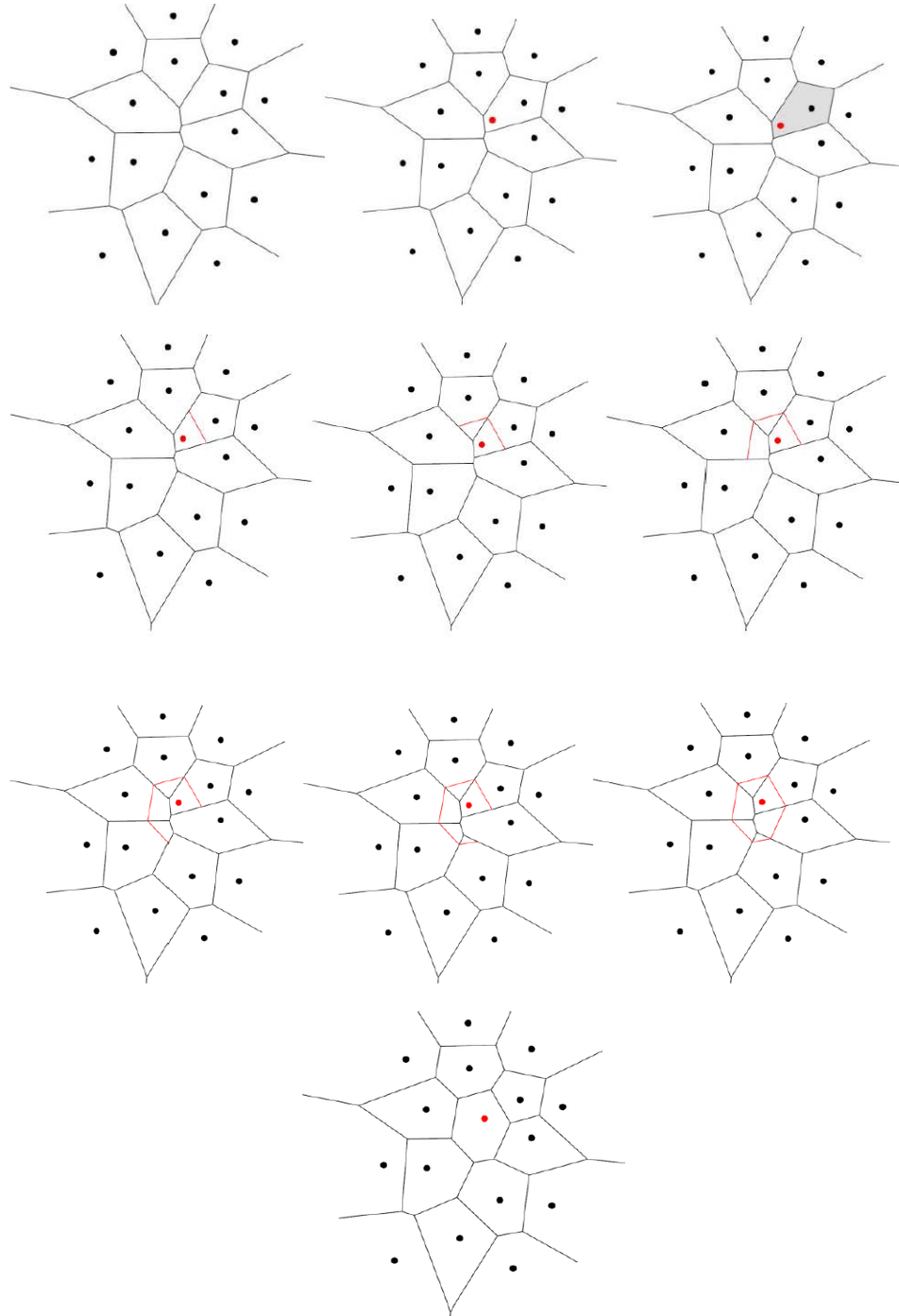


Σχήμα 1.3: Προσθήκη σημείου  $\Gamma$  σε υπάρχον διάγραμμα. οι μεσοκάθετες εμφανίζονται ως ευθείες



Σχήμα 1.4: Προσθήκη σημείου  $\Gamma$  σε υπάρχον διάγραμμα. οι μεσοκάθετες εμφανίζονται ως ημιευθείες

Ακολουθώντας την ίδια λογική, μπορούν να προστεθούν ένα ένα τα καταφύγια και να ανανεώνεται συνεχώς ο χάρτης. Στην παρακάτω ακολουθία διαγραμμάτων (σχήμα 1.4) παρουσιάζεται η προσθήκη ενός καταφυγίου στην περίπτωση που ο χάρτης έχει 15 καταφύγια.



Σχήμα 1.5: Αλληλουχία βημάτων για την προσθήκη ενός νέου σημείου σε υπάρχον διάγραμμα.



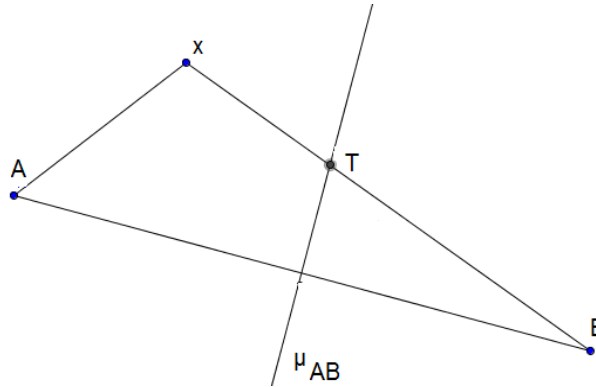
Σε μια προσπάθεια αυτοματοποίησης της διαδικασίας προσθήκης ενός νέου καταφυγίου, η κατάληξη είναι ο παρακάτω αλγόριθμος: Έστω ένας χάρτης με  $n$  καταφύγια  $V = \{κ_1, κ_2, \dots, κ_n\}$  και προστίθεται το καταφύγιο  $κ_{n+1}$

- Βήμα 1: Εντοπίζεται το πλησιέστερο καταφύγιο του  $κ_{n+1}$ , έστω  $κ_c$
- Βήμα 2: Σχεδιάζεται η μεσοκάθετος του ευθύγραμμου τμήματος  $κ_c κ_{n+1}$ , έστω  $εξ$ . Δεν επιτρέπεται στην μεσοκάθετο  $εξ$  να βγει έξω από τα όρια της περιοχής που ορίζεται στον  $V$  από το  $κ_c$ .
- Βήμα 3: Εάν η  $εξ$  δεν τέμνει άλλη μεσοκάθετη του χάρτη  $V$ , τότε η προσθήκη τελειώσε και ακολουθεί το Βήμα 5. Εάν βρεθεί σημείο τομής που έχει ήδη εξεταστεί, η προσθήκη τελειώσε και ακολουθεί το Βήμα 5. Εάν όχι Ακολουθεί το Βήμα 4.
- Βήμα 4: Η  $εξ$  τέμνει μία ή δύο μεσοκαθέτους. Εάν τέμνει δύο, ακολουθώντας πορεία αντίθετη των δεικτών του ρολογιού, επιλέγεται η τομή που συναντάται πρώτα.  
Η μεσοκάθετος που τέμνεται με την  $εξ$  είναι μεταξύ του  $κ_c$  και ενός άλλου καταφυγίου, έστω  $κ_p$ . Επιστροφή στο Βήμα 2, αντικαθιστώντας το καταφύγιο  $κ_c$  με το καταφύγιο  $κ_p$ .
- Βήμα 5: Οι νέες μεσοκάθετες που δημιουργήθηκαν από την παραπάνω διαδικασία ορίζουν την περιοχή του  $κ_{n+1}$ . Διαγράφονται τυχόν τμήματα μεσοκαθέτων που βρίσκονται μέσα στην περιοχή. Ο χάρτης  $V = \{κ_1, κ_2, \dots, κ_n, κ_{n+1}\}$  έχει ανανεωθεί.

Ο παραπάνω αλγόριθμος ανήκει στην κατηγορία των "αυξητικών" ή "προοδευτικών" αλγορίθμων (incremental algorithms). Η ορθότητα του αλγορίθμου αποδεικνύεται εύκολα χρησιμοποιώντας τη μέθοδο της Μαθηματικής επαγωγής, εκμεταλλευόμενοι τον τρόπο ανανέωσης του χάρτη που περιγράφηκε παραπάνω.

Απόδειξη

**I.** Γίνεται έλεγχος εάν ισχύει για  $n=2$ . Θεωρούνται δύο σημεία  $A$  και  $B$ .



Σχήμα 1.6: Απόδειξη πως τα σημεία αριστερά της μεσοκαθέτου είναι πλησιέστερα στο σημείο A

Σχεδιάζεται η μεσοκάθετος του AB, έστω  $\mu_{AB}$ .

Χωρίς βλάβη της γενικότητας, θεωρείται σημείο X αριστερά της μεσοκαθέτου. Θα πρέπει να δειχθεί πως  $AX < BX$ .

Έστω T το σημείο τομής του BX με την  $\mu_{AB}$ .

Για το AT, από το βασικό θεώρημα της μεσοκαθέτου, ισχύει πως  $AT = BT$ .

Στο τρίγωνο AXT, ισχύει πως

$$(XT) = (BX) - (BT) \quad (1.1)$$

Αλλά, από την τριγωνική ανισότητα στο τρίγωνο AXT ισχύει

$$(XT) + (AT) > (XA) \quad (1.2)$$

Η (1.2) λόγω της (1.1) γίνεται

$$(BX) - (BT) + (AT) > (XA) \Rightarrow (BX) - (BT) + (BT) > (XA) \Rightarrow (AX) < (BX) \quad (1.3)$$

**II.** Έστω ένας χάρτης V όπου τα όρια του ορίζουν περιοχές τα σημεία των οποίων είναι πλησιέστερα στο καταφύγιο κάθε περιοχής.

**III.** Εξετάζεται εάν ο χάρτης V έχει την ίδια ιδιότητα με παραπάνω, εάν προσθέσω ένα νέο καταφύγιο, με τον αλγόριθμο που περιγράφηκε παραπάνω. Έστω X ένα σημείο της νέας περιοχής.

Η απόδειξη είναι απλή, εάν ληφθεί υπόψη η απόδειξη στο βήμα (A) και εάν εξετασθεί και συγκριθεί η απόσταση  $X_{kv+1}$  με την απόσταση  $X_{ki}$ , για όλα τα υπόλοιπα καταφύγια. Πάντα υπάρχει μία μεσοκάθετος  $\mu_i$  που η χρήση της βοηθάει στην απόδειξη πως:

$$X_{kv+1} < X_{ki} \quad (1.4)$$

## 1.5.2 Διαγράμματα Voronoi

Το παραπάνω πρόβλημα πρώτα μελετήθηκε συστηματικά από τον Ρώσο μαθηματικό Georgy Feodosevich Voronoi, γι' αυτό και τα διαγράμματα που το επιλύουν ονομάζονται διαγράμματα Voronoi.

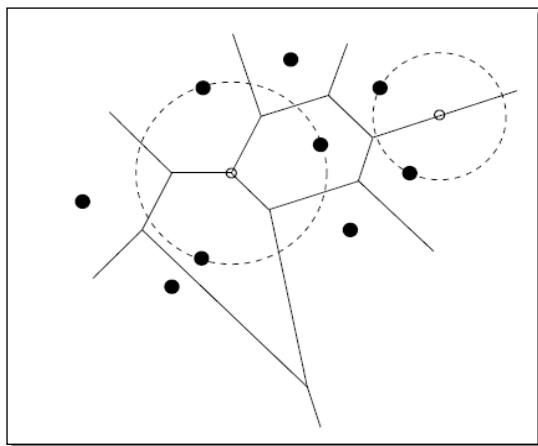
Το πρόβλημα στο επίπεδο ορίζεται ως εξής: Δοθέντων  $n$  σημείων στο επίπεδο  $K = \{κ_1, κ_2, \dots, κ_n\}$  (τα οποία ονομάζονται κέντρα) ορίζεται μία μη κενή περιοχή σημείων  $R_i$  γύρω από κάθε  $κ_i$ . Ορίζεται επίσης η απόσταση  $d_i$  των σημείων της περιοχής  $R_i$  από το κέντρο  $κ_i$ . Η περιοχή  $R_i$  ονομάζεται κελί Voronoi όταν τα σημεία της βρίσκονται πλησιέστερα στο  $κ_i$  από κάθε άλλο κέντρο. Ως κελί Voronoi δηλαδή ορίζεται η περιοχή σημείων για την οποία ισχύει:

$$R_i = \{x \in R^2 \mid d(x, κ_i) \leq d(x, κ_j) \forall i \neq j\} \quad (1.5)$$

Ζητείται να βρεθεί για κάθε  $κ_i$  ένα κελί Voronoi.

Τα διαγράμματα Voronoi έχουν κάποιες πολύ όμορφες και χρήσιμες ιδιότητες, τρεις εκ των οποίων είναι οι παρακάτω:

1. Ένα διάγραμμα Voronoi με  $n$  κέντρα ( $n \geq 3$ ) έχει το πολύ  $3n-6$  ακμές και  $2n-5$  κορυφές (κορυφή ονομάζεται το σημείο τομής ακμών).
2. Κάθε κορυφή ενός διαγράμματος Voronoi αποτελεί κέντρο κύκλου ο οποίος διέρχεται από 3 κέντρα και στο εσωτερικό του δεν υπάρχει κανένα κέντρο.
3. Κάθε σημείο μιας ακμής ενός διαγράμματος Voronoi είναι κέντρο κύκλου που διέρχεται από 2 κέντρα και στο εσωτερικό του δεν υπάρχει κανένα κέντρο.



Σχήμα 1.7: Απεικόνιση των βασικών ιδιοτήτων ενός διαγράμματος Voronoi

Οι αποδείξεις των παραπάνω στηρίζονται στην Θεωρία Γράφων και κάποιος μπορεί να διαβάσει τις αποδείξεις τους στα πονήματα των (Aurenhammer & Klein).

### 1.5.3 Αλγόριθμοι κατασκευής Διαγραμμάτων Voronoi

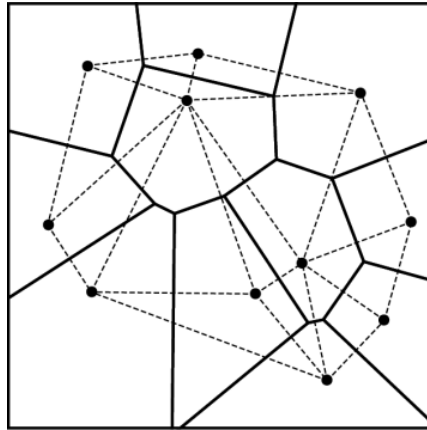
Στο τελευταίο μέρος της εργασίας, οι εφαρμογές των διαγραμμάτων Voronoi είναι πολλές και χρήσιμες. Γι' αυτό το σκοπό είναι επιτακτική ανάγκη να ανευρεθούν αποδοτικοί αλγόριθμοι κατασκευής τους, υπό το πρίσμα του απαραίτητου χρόνου που χρειάζεται ένας υπολογιστής να υπολογίσει το διάγραμμα, αλλά και της απαιτούμενης μνήμης που θα χρησιμοποιήσει.

Παραπάνω περιγράφηκε ήδη ένας αλγόριθμος κατασκευής διαγραμμάτων Voronoi, ο οποίος εντάσσεται στην κατηγορία των αυξητικών ή διαδοχικών αλγόριθμων κατασκευής. Οι αλγόριθμοι αυτής της κατηγορίας είναι ιδιαίτερα αργοί. Ευτυχώς τα τελευταία 30 χρόνια έχουν ανακαλυφθεί αλγόριθμοι πολύ πιο αποδοτικοί.

### 1.5.4 Τριγωνοποίηση Delaunay και Περιορισμένα διαγράμματα Voronoi

Κοιτώντας το πρόβλημα της κατασκευής ενός διαγράμματος Voronoi από μία άλλη οπτική γωνία, αποδεικνύεται εύκολα πως ισοδύναμο (δυτικό) πρόβλημα είναι και η Τριγωνοποίηση Delaunay (σχήμα 1.7), με τις εξής αντιστοιχίες:

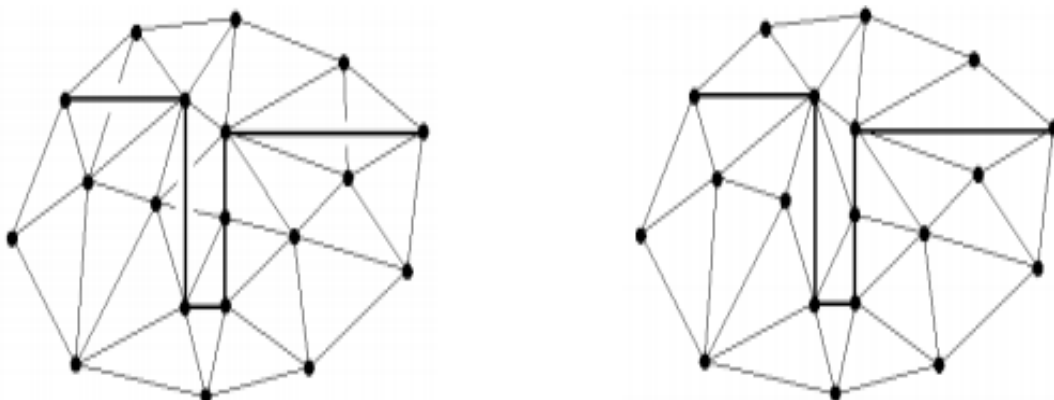
- Τα κελιά Voronoi αντιστοιχούν σε εστίες, δηλαδή κορυφές τριγώνων Delaunay.
- Κάθε ζεύγος γειτονικών κελιών (δηλαδή ακμή Voronoi) αντιστοιχεί σε μία ακμή Delaunay, που ορίζεται από τις δύο εστίες: οι ευθείες των δύο ακμών είναι κάθετες μεταξύ τους.
- Κάθε κορυφή Voronoi αντιστοιχεί σε ένα τρίγωνο Delaunay.



Σχήμα 1.8: Διάγραμμα Voronoi (συνεχής γραμμές) και η αντίστοιχη Τριγωνοποίηση Delaunay (διακεκομμένες γραμμές).

Η Τριγωνοποίηση Delaunay είναι ένας πολύ βολικός τρόπος να μελετηθεί μία επέκταση των διαγραμμάτων Voronoi, τα Περιορισμένα Διαγράμματα Voronoi.

Πολλές φορές κάποια σημεία παρόλο που είναι γεωγραφικά γειτονικά, δεν μπορούν να θεωρηθούν γείτονες. Για παράδειγμα στο χάρτη του δάσους στο προηγούμενο παράδειγμα αν υπάρχει ένα ποτάμι ανάμεσα σε δύο διπλανά καταφύγια, δε θα σχεδιαστεί η διαδρομή ενός εκδρομέα να πηγαίνει απ' ευθείας από το ένα στο άλλο, διότι προφανώς αυτό δεν είναι εφικτό. Όταν εμφανίζονται τέτοιου είδους περιορισμοί, τότε υπάρχει η ανάγκη για το περιορισμένο διάγραμμα Voronoi (Chew, 1989). Για να κατασκευαστεί δηλαδή αυτό το διάγραμμα, στην είσοδο, εκτός από τα σημεία, δίνονται και οι περιοριστικές ακμές (σχήμα 1.8). Χρησιμοποιώντας χωρίς ιδιαίτερες αλλαγές τη μέθοδο της γραμμής σάρωσης που περιγράφεται εδώ (Fortune, 1987). [Βασίλειος Τσακανίκας & Γεώργιος Τσαπακίδης Τα διαγράμματα Voronoi]



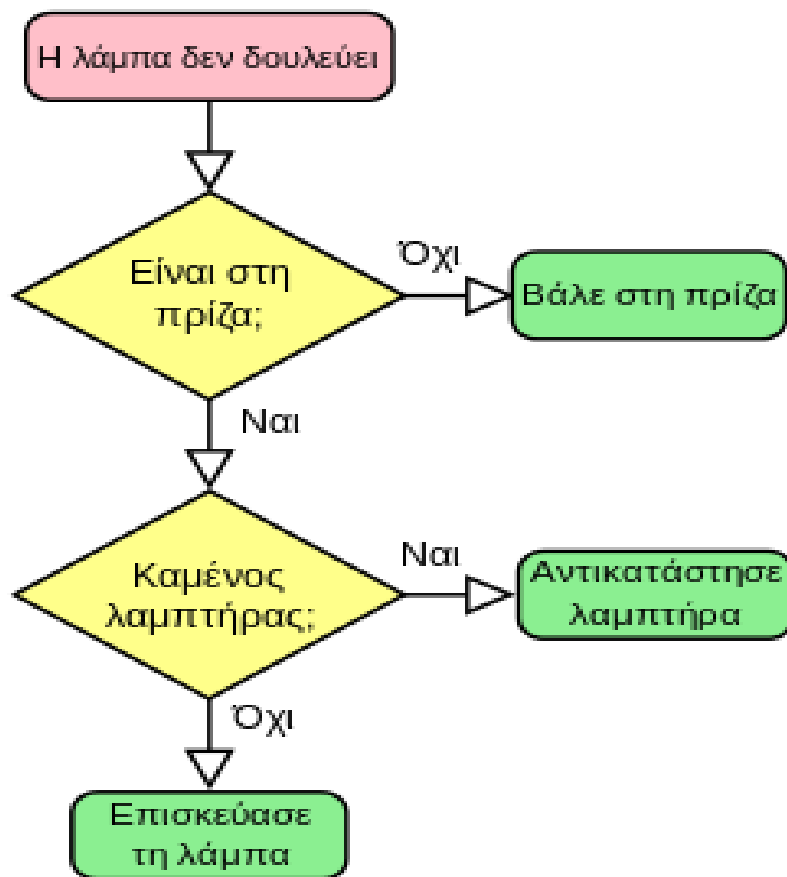
Σχήμα 1.9: Τριγωνοποίηση Delaunay με περιορισμό

## ΚΕΦΑΛΑΙΟ 2

### ΑΛΓΟΡΙΘΜΟΣ – ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

#### 2.1 Αλγόριθμος

Ως **αλγόριθμος** ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος. Πιο απλά **αλγόριθμο** ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σαφείς και έχουν ως σκοπό την επίλυση κάποιου προβλήματος.



Σχήμα 2.1: διάγραμμα ροής, το οποίο απεικονίζει τον αλγόριθμο ελέγχου και επισκευής μιας λάμπας η οποία δεν λειτουργεί.

Η λέξη **αλγόριθμος** προέρχεται από μία μελέτη μαθηματικού , η οποία περιείχε συστηματικές τυποποιημένες λύσεις αλγεβρικών προβλημάτων και αποτελεί ίσως την πρώτη πλήρη πραγματεία άλγεβρας.

Έτσι η λέξη αλγόριθμος καθιερώθηκε αργά τα επόμενα χίλια χρόνια με την έννοια "συστηματική διαδικασία αριθμητικών χειρισμών". Τη σημερινή της σημασία την οφείλει στη γρήγορη ανάπτυξη των ηλεκτρονικών υπολογιστών στα μέσα του 20ου αιώνα. Ωστόσο το νόημα είναι πως η κατάτμηση μιας σύνθετης εργασίας σε διακριτά βήματα που εκτελούνται διαδοχικά, είναι ο πιο πρακτικός τρόπος επίλυσης πολλών προβλημάτων.

Τα βήματα δημιουργίας αλγόριθμου είναι:

- Διατύπωση του προβλήματος
- Κατανόηση του προβλήματος
- Λύση του προβλήματος
- Διατύπωση του αλγόριθμου
- Έλεγχος της λύσης

Οι αλγόριθμοι θα πρέπει να πληρούν κάποια πρότυπα και να διατυπώνονται με συγκεκριμένο τρόπο. Έτσι ένας αλγόριθμος πρέπει να ικανοποιεί τα επόμενα κριτήρια:

- Καθοριστικότητα (Definiteness)  
Κάθε κανόνας του ορίζεται επακριβώς και η αντίστοιχη διεργασία είναι συγκεκριμένη. Κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της. Π.χ. Σε μία διαίρεση να λαμβάνεται υπόψη και η περίπτωση όπου ο διαιρετέος λαμβάνει μηδενική τιμή. Τυπικές περιπτώσεις η διαίρεση με το μηδέν, υπόριζος ποσότητα αρνητική, κλπ. Προβλήματα καθοριστικότητας αντιμετωπίζονται συχνά με τη λογική της επιλογής, δηλ. Αν  $a > 0$  τότε ..... αλλιώς .....
- Περατότητα (Finiteness)  
Κάθε εκτέλεση είναι πεπερασμένη, δηλαδή τελειώνει ύστερα από έναν πεπερασμένο αριθμό διεργασιών ή βημάτων. Μία διαδικασία που δεν τελειώνει μετά από συγκεκριμένο/πεπερασμένο αριθμό βημάτων λέγεται απλά υπολογιστική διαδικασία.
- Αποτελεσματικότητα (Effectiveness)  
Είναι μηχανιστικά αποτελεσματικός, δηλαδή όλες οι διαδικασίες που περιλαμβάνει μπορούν να πραγματοποιηθούν με ακρίβεια και σε πεπερασμένο χρόνο "με μολύβι και χαρτί". Κάθε μεμονωμένη εντολή

του αλγορίθμου να είναι απλή (και όχι σύνθετη). Δηλαδή μία εντολή δεν αρκεί να έχει ορισθεί αλλά πρέπει να είναι και εκτελέσιμη.

- Να έχει είσοδο δεδομένων, επεξεργασία και έξοδο αποτελεσμάτων. (Input - Output)

Κατά την εκκίνηση εκτέλεσης του αλγορίθμου καμία, μία ή περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον αλγόριθμο. Η περίπτωση που δε δίνονται τιμές δεδομένων εμφανίζεται όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων απλών εντολών.

Δίδει τουλάχιστον ένα μέγεθος ως αποτέλεσμα που εξαρτάται κατά κάποιο τρόπο από τις αρχικές εισόδους. Ο αλγόριθμος πρέπει να δημιουργεί τουλάχιστον μία τιμή (δεδομένων) ως αποτέλεσμα προς το χρήστη ή προς ένα άλλο αλγόριθμο.

Οι βασικοί τρόποι περιγραφής και αναπαράστασης ενός αλγορίθμου είναι τέσσερις:

1. Ελεύθερο κείμενο, που αποτελεί τον πιο αδόμητο τρόπο παρουσίασης αλγορίθμου. Ελλοχεύει η δημιουργία μιας μη εκτελέσιμης κατάστασης παραβιάζοντας έτσι το κριτήριο της αποτελεσματικότητας.
2. Διάγραμμα ροής, που συνιστά έναν πιο γραφικό τρόπο παρουσίασης του αλγορίθμου. Η χρήση διαγραμμάτων ροής δεν είναι η πλέον ενδεδειγμένη λύση για ένα πρόβλημα και για αυτό χρησιμοποιούνται σπάνια στη βιβλιογραφία.
3. Φυσική γλώσσα που εκτελείται κατά βήματα. Σε αυτή την περίπτωση μπορεί να παραβιαστεί το κριτήριο του καθορισμού μεταξύ των βημάτων.
4. Κωδικοποίηση του αλγορίθμου σε ψευδογλώσσα ή γλώσσα προγραμματισμού. Έτσι ο αλγόριθμος παρουσιάζεται πιο συνοπτικός, συμπαγής ενώ πληροί και τις προϋποθέσεις του δομημένου προγραμματισμού.

Οι αλγόριθμοι μπορούν να υλοποιηθούν από προγράμματα ηλεκτρονικών υπολογιστών, μολονότι συχνά σε περιορισμένες μορφές. Ένα λάθος στον σχεδιασμό ενός αλγόριθμου για την λύση ενός προβλήματος μπορεί να οδηγήσει σε αποτυχίες/βλάβες στο εφαρμοσμένο πρόγραμμα.

Οι αλγόριθμοι δεν υλοποιούνται μόνο ως προγράμματα υπολογιστών, αλλά συχνά επίσης και με άλλα μέσα, όπως π.χ. σε ένα βιολογικό νευρικό δίκτυο, ή σε ένα ηλεκτρονικό κύκλωμα, ή σε μια μηχανική συσκευή.



## 2.2 Γλώσσα προγραμματισμού

Γλώσσα προγραμματισμού λέγεται μια τεχνητή γλώσσα που μπορεί να χρησιμοποιηθεί για τον έλεγχο μιας μηχανής, συνήθως ενός υπολογιστή. Οι γλώσσες προγραμματισμού (όπως άλλωστε και οι ανθρώπινες γλώσσες) ορίζονται από ένα σύνολο συντακτικών και εννοιολογικών κανόνων, που ορίζουν τη δομή και το νόημα, αντίστοιχα, των προτάσεων της γλώσσας.

Οι γλώσσες προγραμματισμού χρησιμοποιούνται για να διευκολύνουν την οργάνωση και διαχείριση πληροφοριών, αλλά και για την ακριβή διατύπωση αλγορίθμων. Ορισμένοι ειδικοί χρησιμοποιούν τον όρο γλώσσα προγραμματισμού μόνο για τυπικές γλώσσες που μπορούν να εκφράσουν όλους τους πιθανούς αλγορίθμους.

Υπάρχουν χιλιάδες διαφορετικές γλώσσες προγραμματισμού, και κάθε χρόνο δημιουργούνται περισσότερες για να καλύψουν πληθώρα εφαρμογών σε όλους τους επαγγελματικούς και μη τομείς.

Κάθε γλώσσα προγραμματισμού έχει το δικό της σύνολο τυπικών προδιαγραφών (ή κανόνων) που αφορούν το συντακτικό, το λεξιλόγιο και το νόημα της. Για πολλές γλώσσες που χρησιμοποιούνται ευρέως και έχουν χρησιμοποιηθεί για αρκετό χρονικό διάστημα (π.χ. C, C++, Java, Scheme), υπάρχουν ειδικοί φορείς τυποποίησης, οι οποίοι μέσα από τακτές συναντήσεις δημιουργούν, τροποποιούν ή επεκτείνουν τις τυπικές προδιαγραφές που διέπουν τη χρήση μιας γλώσσας προγραμματισμού. Άλλες γλώσσες δεν περιγράφονται σε κάποιο επίσημο πρότυπο αλλά ορίζονται μόνο με βάση κάποια υλοποίησή τους (που αποτελεί το *de facto* πρότυπο), όπως η Python που περιγράφεται από την υλοποίηση CPython.

### 2.2.1 Κατηγοριοποίηση γλωσσών προγραμματισμού

Δεν υπάρχει απλός τρόπος να κατηγοριοποιηθούν οι γλώσσες προγραμματισμού. Αυτό συμβαίνει γιατί συνήθως κάθε γλώσσα προγραμματισμού περιέχει επιρροές από πολλές προηγούμενες γλώσσες, συνδυάζοντας θετικά στοιχεία και προσθέτοντας νέα. Χαρακτηριστικά που εμφανίζονται σε μια γλώσσα και έχουν θετική αποδοχή, συνήθως υιοθετούνται από μεταγενέστερες γλώσσες ακόμα και αν πρόκειται για γλώσσες που ανήκουν σε διαφορετική κατηγορία.

Η κατηγοριοποίηση είναι ακόμα πιο περίπλοκη για το λόγο ότι πολλές γλώσσες συνήθως ανήκουν σε παραπάνω από μία κατηγορίες. Για

παράδειγμα, η Java είναι τόσο αντικειμενοστραφής όσο και παράλληλη γλώσσα, δεδομένου ότι υποστηρίζει την οργάνωση των δεδομένων και υπολογισμών σε αντικείμενα, αλλά επιτρέπει επίσης και τη δημιουργία προγραμμάτων με ταυτόχρονα νήματα (threads) που εκτελούνται παράλληλα.

Εξαιτίας της δυσκολίας στην κατηγοριοποίηση, οι γλώσσες προγραμματισμού κατηγοριοποιούνται με διάφορους τρόπους. Οι συνηθέστεροι τρόποι είναι:

➤ *με βάση τον τρόπο οργάνωσης του προγράμματος*

- **Διαδικαστικές γλώσσες** (procedural) όπου το πρόγραμμα είναι οργανωμένο σε διαδικασίες, που αποτελούνται από σειρές εντολών που περιγράφουν αλγορίθμους. Παραδείγματα γλωσσών που ανήκουν σε αυτή την κατηγορία είναι η Pascal ή η C.
- **Αντικειμενοστραφείς γλώσσες** (object-oriented) όπου το πρόγραμμα είναι οργανωμένο σε αντικείμενα. Ένα αντικείμενο είναι μια μονάδα που αποτελείται από την περιγραφή κάποιων δεδομένων και την περιγραφή των αλγορίθμων που τα επεξεργάζονται. Ένα αντικειμενοστραφές πρόγραμμα αποτελείται από διάφορα αντικείμενα που αλληλεπιδρούν μεταξύ τους. Παραδείγματα αντικειμενοστραφών γλωσσών είναι η Java ή η C++.
- **Συναρτησιακές γλώσσες** (functional) όπου οι υπολογισμοί εκφράζονται ως εφαρμογές μαθηματικών συναρτήσεων, σε αντίθεση με τα άλλα είδη προγραμματισμού όπου οι υπολογισμοί εκφράζονται ως σειρές εντολών, όπου η κάθε μία αλλάζει με κάποιο τρόπο την κατάσταση του συστήματος. Θεωρητικό τους υπόβαθρο είναι ο λ-λογισμός. Χαρακτηριστικές συναρτησιακές γλώσσες είναι η Lisp, η Haskell και η OCaml.

➤ *με βάση τον στόχο που έχει η γλώσσα*

- **Γλώσσες γενικής χρήσης.** Σε αυτήν την κατηγορία ανήκουν γλώσσες που δημιουργήθηκαν για τον προγραμματισμό γενικών εφαρμογών, καθώς και πολλές εκπαιδευτικές γλώσσες που αποδείχτηκαν χρήσιμες για την ανάπτυξη γενικών εφαρμογών, όπως η Pascal.
- **Γλώσσες προγραμματισμού συστημάτων,** που χρησιμοποιούνται συνήθως για τον προγραμματισμό λειτουργικών συστημάτων ή οδηγών (drivers) υλικού, όπου χρειάζεται πολλές φορές ο

προγραμματιστής να έχει έλεγχο και γνώση του πώς λειτουργεί το υλικό. Η πιο συχνά χρησιμοποιούμενη γλώσσα προγραμματισμού συστημάτων είναι η C.

- **Γλώσσες σεναρίων (scripting).** Αυτές οι γλώσσες χρησιμοποιούνται συνήθως για τη γρήγορη ανάπτυξη μικρών προγραμμάτων, σε περιπτώσεις που ο χρόνος του προγραμματιστή είναι πιο πολύτιμος από την ταχύτητα εκτέλεσης του προγράμματος, όπως για παράδειγμα συμβαίνει όταν το πρόγραμμα απλά αυτοματοποιεί απλές λειτουργίες. Παραδείγματα γλωσσών σεναρίων (scripting) είναι η Perl, η Python, η Ruby ή τα κελύφη του λειτουργικού συστήματος Unix (shells).
- **Γλώσσες ειδικών εφαρμογών.** Σε αυτή την κατηγορία ανήκουν γλώσσες που αναπτύχθηκαν ειδικά για μια συγκεκριμένη εφαρμογή. Για παράδειγμα, η γλώσσα PostScript είναι σχεδιασμένη ειδικά για να περιγράφονται με λεπτομέρεια κείμενα προς εκτύπωση, ενώ η γλώσσα Matlab είναι σχεδιασμένη για την επεξεργασία πινάκων από αριθμητικά δεδομένα.
- **Παράλληλες ή κατανεμημένες γλώσσες.** Στη συγκεκριμένη κατηγορία ταξινομούνται γλώσσες που επιτρέπουν τη ανάπτυξη παράλληλων προγραμμάτων, όπου πολλές εντολές εκτελούνται ταυτόχρονα σε πολλούς υπολογιστές, έτσι ώστε το τελικό αποτέλεσμα να προκύψει γρηγορότερα. Οι παράλληλες γλώσσες προσφέρουν συνήθως εύκολους τρόπους επικοινωνίας μεταξύ των νημάτων που εκτελούνται παράλληλα, καθώς και τρόπους ώστε να δημιουργούνται καινούριες παράλληλες εκτελέσεις. Παραδείγματα γλωσσών που ανήκουν (και) σε αυτή την κατηγορία είναι η Go, η Java, η Erlang, η MultiLisp ή η Cilk.
- **Εκπαιδευτικά προγραμματιστικά περιβάλλοντα.** Σε αυτή την κατηγορία ανήκουν εκπαιδευτικές; γλώσσες προγραμματισμού οι οποίες απευθύνονται σε αρχάριους προγραμματιστές για την κατασκευή μικροεφαρμογών. Είναι κατάλληλες για την εκμάθηση προγραμματισμού σε μικρές ηλικίες. Παραδείγματα τέτοιων γλωσσών είναι η LOGO, το Game Maker και το App Inventor.

➤ *με βάση τον τρόπο που περιγράφουν το ζητούμενο αποτέλεσμα.*

- **Προστακτικές γλώσσες προγραμματισμού (imperative)** είναι οι γλώσσες που περιγράφουν το ζητούμενο αποτέλεσμα, δίνοντας μια σειρά εντολών που όταν εκτελεστούν παράγουν το ζητούμενο αποτέλεσμα. Τέτοιες γλώσσες είναι η C, η Java αλλά και η OCaml.

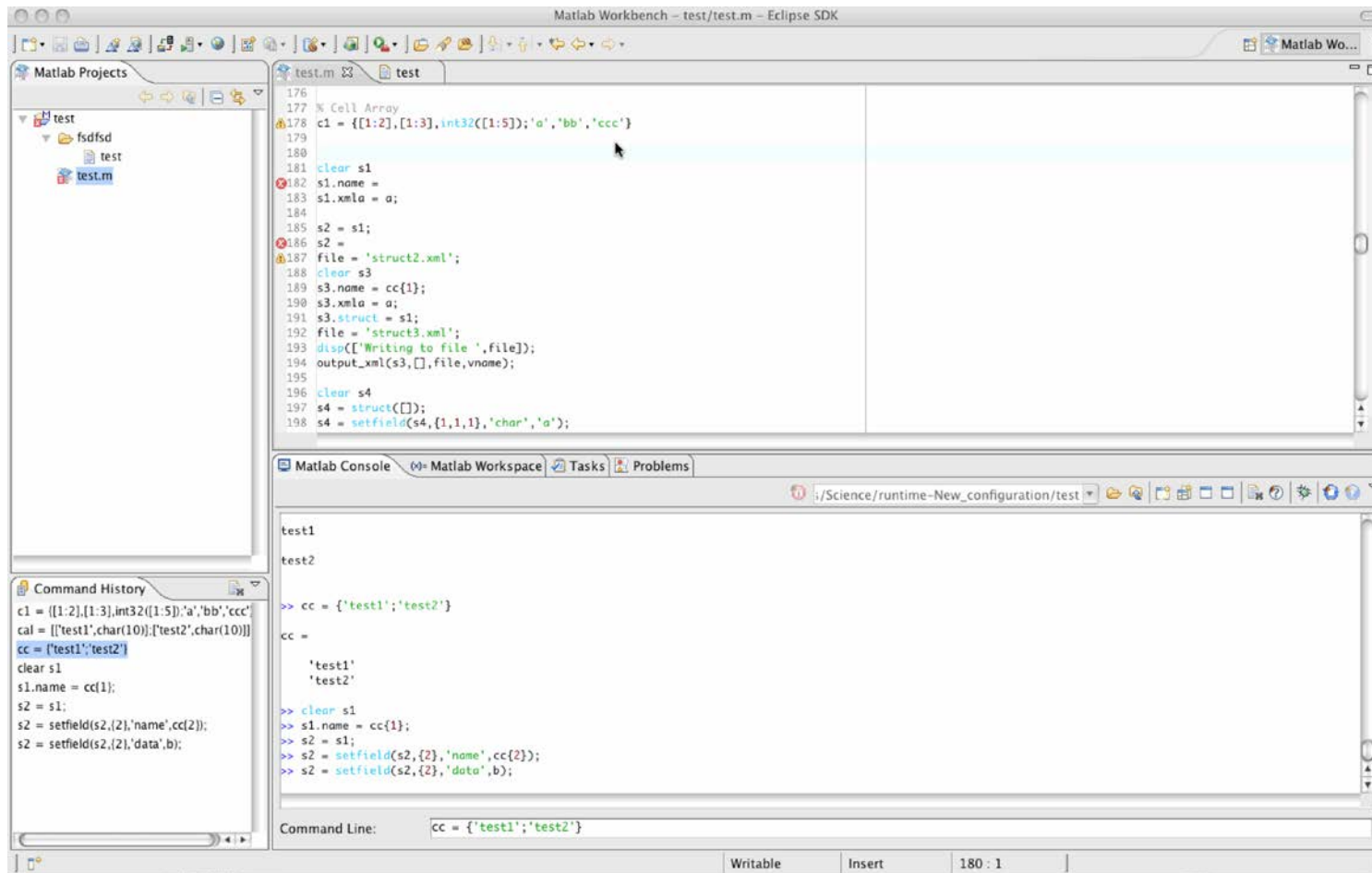
- **Δηλωτικές γλώσσες** προγραμματισμού (declarative) είναι οι γλώσσες που περιγράφουν το ζητούμενο αποτέλεσμα χρησιμοποιώντας τις ιδιότητες που έχει, και όχι τον τρόπο με τον οποίο υπολογίζεται. Παραδείγματα δηλωτικών γλωσσών είναι η Haskell, η SQL και η Prolog.

### 2.2.2 MatLab

Το MATLAB (matrix laboratory) είναι ένα περιβάλλον αριθμητικής υπολογιστικής και μια προγραμματιστική γλώσσα τέταρτης γενιάς. Αποθηκεύει και κάνει τις πράξεις με βάση την άλγεβρα μητρών. Η τρέχουσα έκδοσή του είναι η R2015a η οποία κυκλοφόρησε τον Μάρτιο του 2015.[1]

Χρησιμοποιείται κατά κύριο λόγο για την επίλυση μαθηματικών προβλημάτων, ωστόσο είναι πολύ "ισχυρό" και μπορεί να χρησιμοποιηθεί και για προγραμματισμό καθώς περιέχει εντολές από την C++. Στον τομέα των γραφικών όσον αφορά τον μαθηματικό κλάδο μπορεί να υλοποιήσει συναρτήσεις πραγματικές, μιγαδικές, πεπλεγμένες συναρτήσεις δύο μεταβλητών και άλλες. Όσον αφορά στον στατιστικό κλάδο μπορεί να υλοποιήσει ιστογράμματα, τομεογράμματα, ραβδοδιαγράμματα, εμβοδογράμματα και άλλα.

Ανάπτυξη λογισμικού παραγωγής και χωροθέτησης τυχαίων σημείων με τη χρήση μαθηματικών μοντέλων



Εικόνα 2.1: Τυπική μορφή του γραφικού περιβάλλοντος της MATLAB

### 2.2.3 WPF (Visual Basic. NET + XAML)

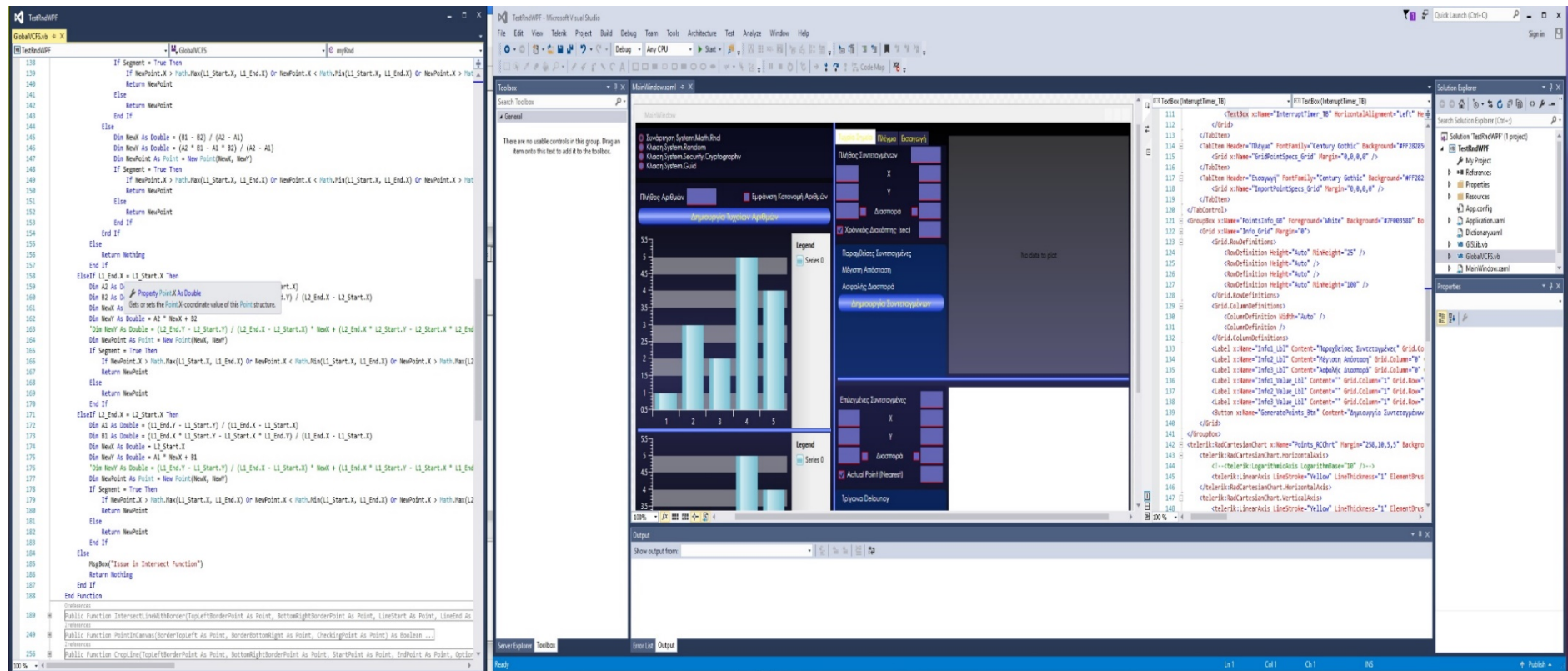
Η WPF (Windows Presentation Foundation) είναι ένα γραφικό υποσύστημα για οπτική απόδοση γραφικών περιβαλλόντων για εφαρμογές Windows που βασίζεται στο DirectX. Η WPF παρέχει ένα συνεπές μοντέλο προγραμματισμού για την ανάπτυξη εφαρμογών και διαχωρίζει το περιβάλλον χρήστη (Frontend) από την αλγοριθμική λογική (Backend).

Στη WPF η σχεδίαση του Frontend βασίζεται στη γλώσσα προγραμματισμού XAML (eXtensible Application Markup Language) η οποία στη σύνταξή της βασίζεται στις αρχές της XML.

Για τη συγγραφή της αλγοριθμικής λογικής η WPF έχει τη δυνατότητα να χρησιμοποιήσει γλώσσες προγραμματισμού που βασίζονται στο .NET framework όπως Visual Basic NET και C#.

Η Visual Basic. NET είναι ένα εργαλείο ανάπτυξης που μπορεί να χρησιμοποιηθεί για να δημιουργηθούν εφαρμογές οι οποίες θα εκτελούν χρήσιμες εργασίες και θα δείχνουν εντυπωσιακές με την εφαρμογή διάφορων ρυθμίσεων. Χρησιμοποιώντας τη Visual Basic. NET, δημιουργούνται εφαρμογές για το λειτουργικό Windows, τον ιστό, φορητές συσκευές και ένα πλήθος άλλων περιβαλλόντων και ρυθμίσεων. Το πιο σημαντικό πλεονέκτημα της Visual Basic. NET, είναι ότι έχει σχεδιασθεί έτσι ώστε να κάνει ακόμα πιο παραγωγική την καθημερινή εργασία, ειδικά αν χρειάζεται να χρησιμοποιηθούν πληροφορίες σε βάσεις δεδομένων ή να δημιουργηθούν λύσεις για το διαδίκτυο, αλλά ένα επιπλέον όφελος είναι ότι από τη στιγμή που κατανοηθεί πλήρως το περιβάλλον ανάπτυξης του Visual Studio, θα είναι εύκολο να χρησιμοποιηθούν τα ίδια εργαλεία για να γραφούν προγράμματα με τη Microsoft C, τη Microsoft C++ και άλλα εργαλεία και μεταγλωττιστές.

Ανάπτυξη λογισμικού παραγωγής και χωροθέτησης τυχαίων σημείων με τη χρήση μαθηματικών μοντέλων



Εικόνα 2.2: Τυπική μορφή του γραφικού περιβάλλοντος της WPF

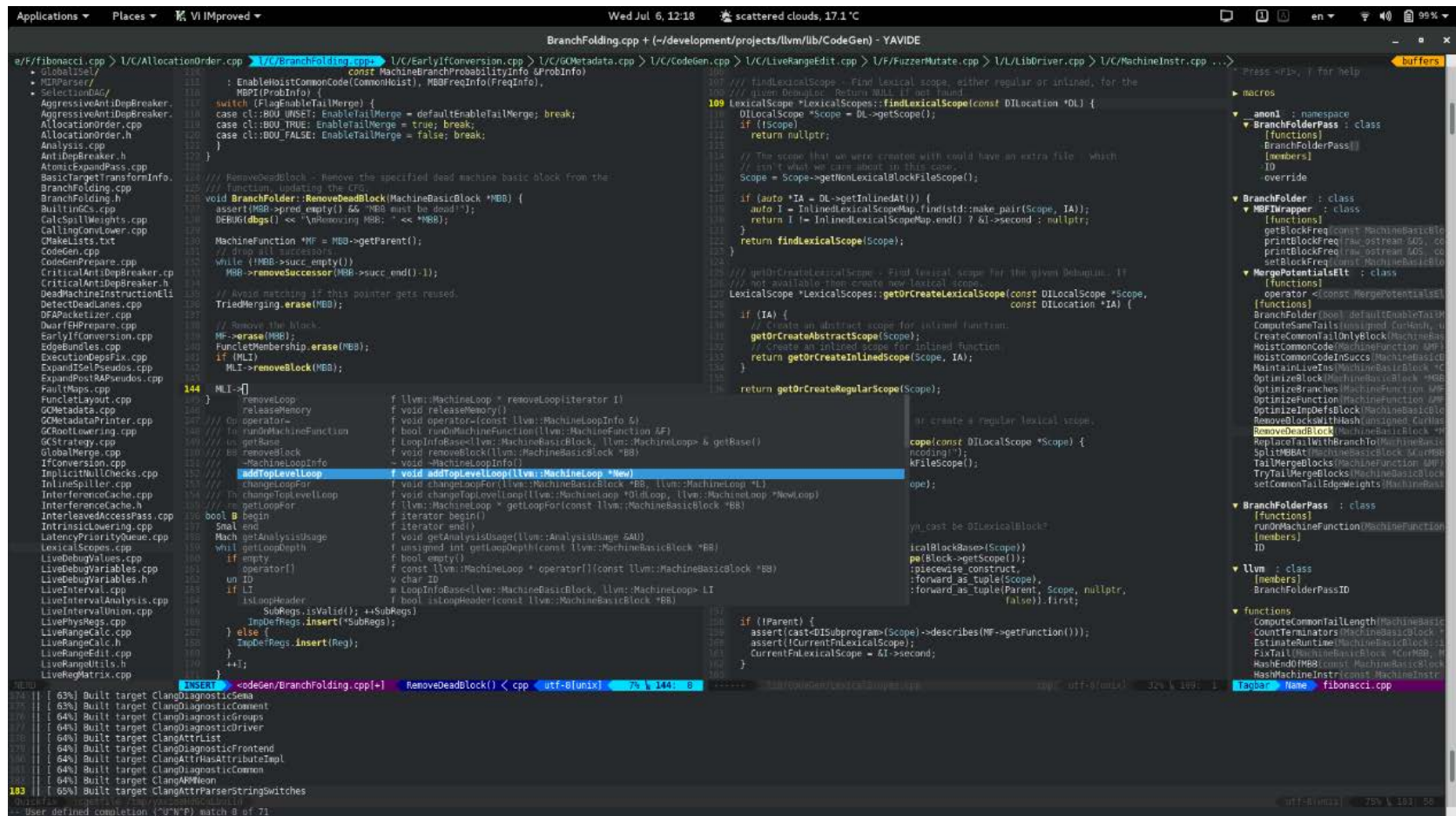
## 2.2.4 C++

Η C++ είναι μια γενικού σκοπού γλώσσα προγραμματισμού Η/Υ. Θεωρείται μέσου επιπέδου γλώσσα, καθώς περιλαμβάνει έναν συνδυασμό χαρακτηριστικών από γλώσσες υψηλού και χαμηλού επιπέδου. Είναι μια μεταγλωττιζόμενη γλώσσα πολλαπλών παραδειγμάτων, με τύπους. Υποστηρίζει δομημένο, αντικειμενοστραφή και γενικό προγραμματισμό.

Η γλώσσα αναπτύχθηκε το 1979 στα εργαστήρια Bell της AT&T, ως βελτίωση της ήδη υπάρχουσας γλώσσας προγραμματισμού C, και αρχικά ονομάστηκε "C with Classes", δηλαδή C με Κλάσεις. Μετονομάστηκε σε C++ το 1983. Οι βελτιώσεις ξεκίνησαν με την προσθήκη κλάσεων, και ακολούθησαν, μεταξύ άλλων, εικονικές συναρτήσεις, υπερφόρτωση τελεστών, πολλαπλή κληρονομικότητα, πρότυπα κ.α.



Ανάπτυξη λογισμικού παραγωγής και χωροθέτησης τυχαίων σημείων με τη χρήση μαθηματικών μοντέλων



Εικόνα 2.3: Τυπική μορφή του γραφικού περιβάλλοντος της C++

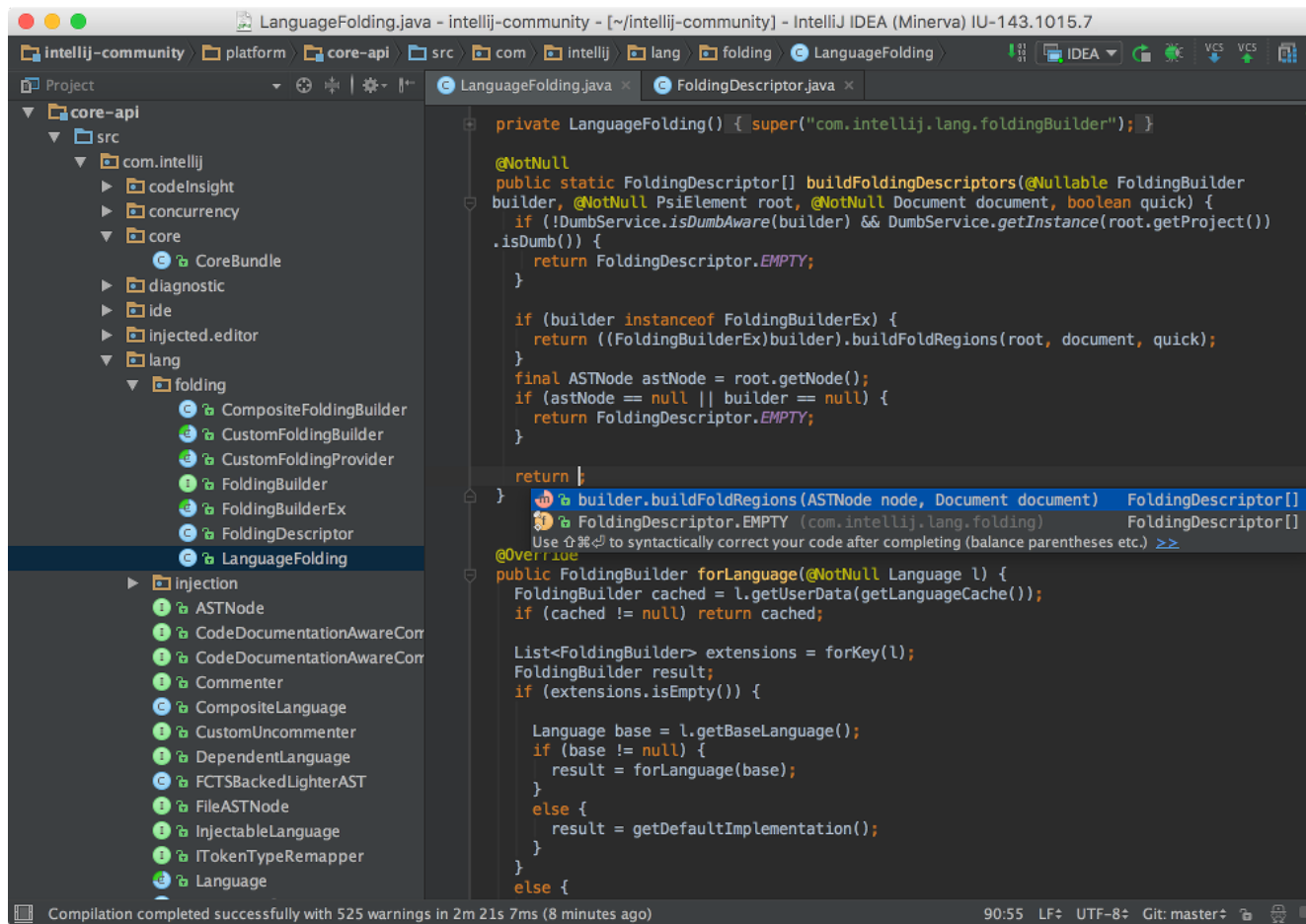
### 2.2.5 Java

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems.

Στις αρχές του 1991, η Sun αναζητούσε το κατάλληλο εργαλείο για να αποτελέσει την πλατφόρμα ανάπτυξης λογισμικού σε μικρο-συσκευές (έξυπνες οικιακές συσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών). Τα εργαλεία της εποχής ήταν γλώσσες όπως η C++ και η C. Μετά από διάφορους πειραματισμούς προέκυψε το συμπέρασμα ότι οι υπάρχουσες γλώσσες δεν μπορούσαν να καλύψουν τις ανάγκες τους.

Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι "κατανοητά" από κάθε υπολογιστή ανεξάρτητα του είδους. Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός κώδικας (assembly) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine ή VM ή EM στα ελληνικά).

Ανάπτυξη λογισμικού παραγωγής και χωροθέτησης τυχαίων σημείων με τη χρήση μαθηματικών μοντέλων



Εικόνα 2.4: Τυπική μορφή του γραφικού περιβάλλοντος της Java

## 2.3 Επιλογή γλώσσας προγραμματισμού

Για την επιλογή της γλώσσας προγραμματισμού εφαρμόστηκε η μέθοδος της πολυκριτηριακής ανάλυσης χρησιμοποιώντας επιλεγμένα κριτήρια και βάρη, με τιμές από 0 (δεν έχει το συγκεκριμένο χαρακτηριστικό) έως 5 (έχει το συγκεκριμένο χαρακτηριστικό στη μέγιστη τιμή) και συντάχθηκε ο πίνακας 2.1.

Αναλυτικότερα ο πίνακας μας γνωστοποιεί αρχικά ότι στην **εκμάθηση της γλώσσας** (δηλαδή πόσο εύκολα ένας προγραμματιστής μπορεί να μάθει τον τρόπο σύνταξης των εντολών και των συναρτήσεων), η VB .NET φαίνεται να υπερτερεί έναντι του MATLAB και της Java, ενώ η C++ θεωρείται πιο δύσκολη σε αυτό τον τομέα.

Κάπως αντίστοιχα είναι και στην **ευκολία της σύνταξης** των αλγορίθμων αφού η VB .NET ξεχωρίζει για την ευκολία της, ενώ η σχέση στην ευκολία σύνταξης μεταξύ των υπολοίπων είναι περίπου ίδια.

Για το **φιλικό περιβάλλον του IDE** και των εργαλείων που διαθέτουν υπάρχει μια κλιμακωτή διαφορά μεταξύ τους, με πρώτη τη VB .NET δεύτερη τη Java τρίτη τη MATLAB και τέταρτη εκείνη με τον πιο δύσχρηστο τρόπο λειτουργίας τη C++.

Περνώντας στα εργαλεία (εντολές και έτοιμες συναρτήσεις) που διαθέτει η κάθε μια και στις **προεγκατεστημένες βιβλιοθήκες**, η C++ δεν διαθέτει καμία, η Java διαθέτει αρκετές, ενώ εκείνες που έχουν τις περισσότερες προεγκατεστημένες βιβλιοθήκες είναι η MATLAB και η VB .NET.

Εδώ πρέπει να αναφερθεί ότι υπάρχουν **διαθέσιμες βιβλιοθήκες** και επεκτάσεις αυτών που μπορούν να προστεθούν στη C++, περισσότερες στη MATLAB και τη Java, ενώ η VB .NET διαθέτει τη μεγαλύτερη ποικιλία σε πρόσθετες βιβλιοθήκες. Όσον αφορά στην "ταχύτητα εκτέλεσης αλγορίθμου" η C++ είναι η γρηγορότερη των άλλων γλωσσών προγραμματισμού, μιας και επικοινωνεί πιο άμεσα με τον επεξεργαστή. Ακολουθεί η MATLAB και στη συνέχεια η VB .NET και η Java με περίπου ίσους χρόνους εκτέλεσης εντολών και συναρτήσεων.

Στη **δημιουργία γραφικού περιβάλλοντος** η VB είναι εκείνη που κατέχει την πρώτη θέση αφού δημιουργεί τις πιο ελκυστικές απεικονίσεις με δεύτερη τη Java και τρίτη τη MATLAB. Τελευταία έρχεται η C++ όπου δε διαθέτει έτοιμα στοιχεία ελέγχου και τα οποία θα πρέπει να σχεδιαστούν.

Η C++ δεν διαθέτει αυτή τη δυνατότητα να δημιουργήσει **απ' ευθείας εκτελέσιμο αρχείο** ενώ οι υπόλοιπες γλώσσες προγραμματισμού έχουν

ενσωματωμένη αυτή τη δυνατότητα στα IDE τους.

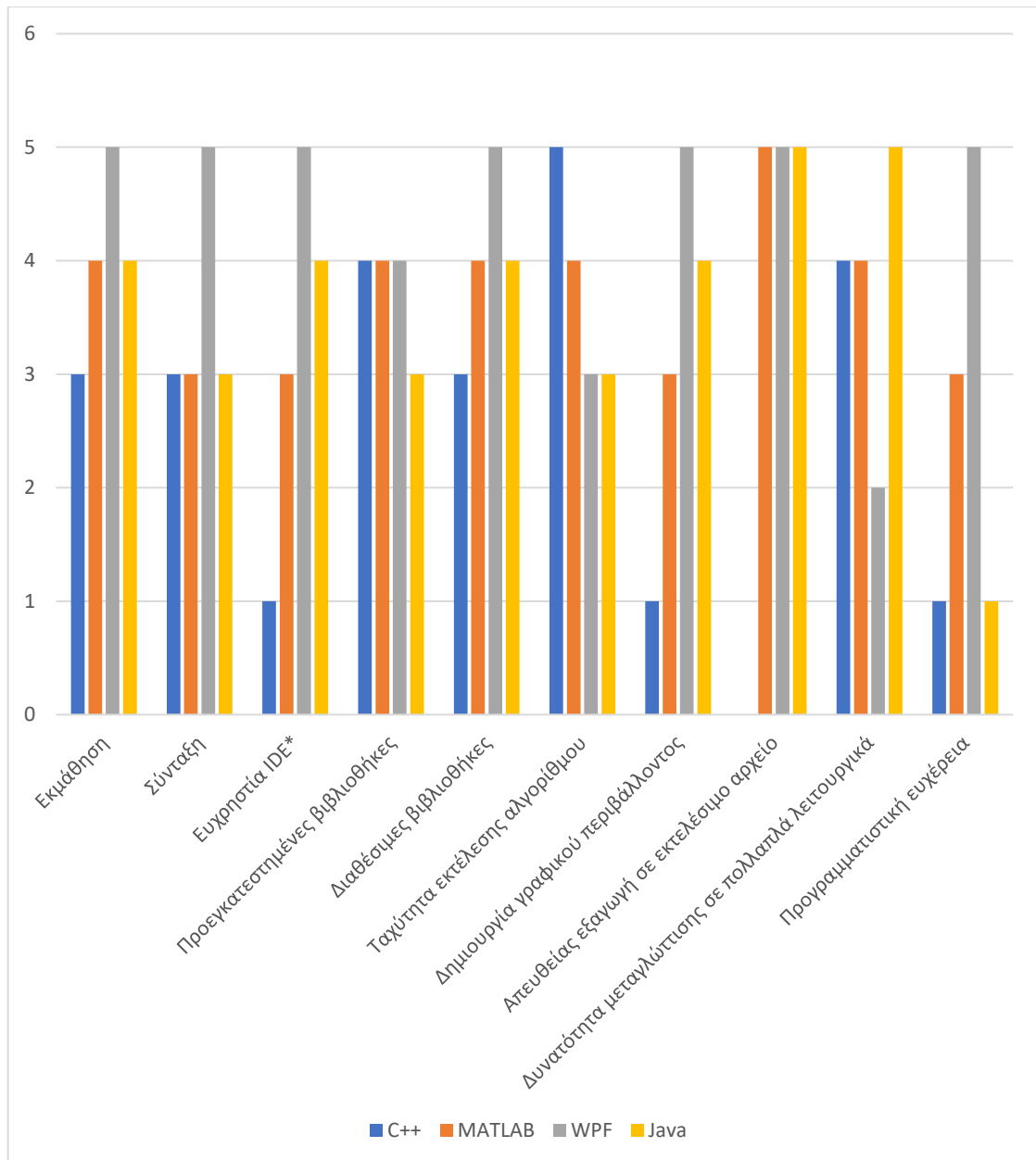
Συνεχίζοντας, η C++ και η Java μπορούν με **μεταγλωττιστούν σε πολλαπλά λειτουργικά**, στη MATLAB αυτή η διαδικασία είναι πιο περιορισμένη ενώ η VB .NET μπορεί να μεταγλωττιστεί μόνον σε λειτουργικά Windows.

Τέλος η κάθε γλώσσα προγραμματισμού έχει κάποιο **βασικό χαρακτηριστικό** για το οποίο προτείνεται. Η C++ λοιπόν προτείνεται για χαμηλού επιπέδου προγραμματισμό, η MATLAB για προσομοιώσεις ενώ η VB .NET και η Java για δημιουργία εφαρμογών.

	C++	MATLAB	WPF	Java
Εκμάθηση	3	4	5	4
Σύνταξη	3	3	5	3
Ευχρηστία IDE*	1	3	5	4
Προεγκατεστημένες βιβλιοθήκες	4	4	4	3
Διαθέσιμες βιβλιοθήκες	3	4	5	4
Ταχύτητα εκτέλεσης αλγορίθμου	5	4	3	3
Δημιουργία γραφικού περιβάλλοντος	1	3	5	4
Απευθείας εξαγωγή σε εκτελέσιμο αρχείο	0	5	5	5
Δυνατότητα μεταγλώττισης σε πολλαπλά λειτουργικά	4	4	2	5
Προτείνεται για:	προγραμματισμό χαμηλού επιπέδου	προσομοίωση	δημιουργία εφαρμογών	δημιουργία εφαρμογών
Προγραμματιστική ευχέρεια	1	3	5	1
Σύνολο	25	37	<b>44</b>	36

Πίνακας 2.1: Συνοπτική αξιολόγηση γλωσσών προγραμματισμού

\*Ολοκληρωμένο περιβάλλον ανάπτυξης (IDE - Integrated Development Environment)



Πίνακας 2.2: Αξιολογήση γλωσσών προγραμματισμού σε ραβδόγραμμα

Ένα ολοκληρωμένο περιβάλλον ανάπτυξης είναι μία σουίτα λογισμικού που βοηθάει στην ανάπτυξη προγραμμάτων υπολογιστή. Συνήθως ένα IDE περιλαμβάνει κάποιον επεξεργαστή πηγαίου κώδικα, έναν μεταγλωττιστή, εργαλεία αυτόματης παραγωγής κώδικα, αποσφαλματωτή, συνδέτη, σύστημα ελέγχου εκδόσεων και εργαλεία κατασκευής γραφικών διασυνδέσεων χρήστη για τις υπό ανάπτυξη εφαρμογές.



## ΚΕΦΑΛΑΙΟ 3 ΑΝΑΛΥΣΗ ΛΟΓΙΣΜΙΚΟΥ

### 3.1 Γενικά

Η πολυκριτηριακή ανάλυση του προηγούμενου κεφαλαίου έδειξε ότι η ιδανική γλώσσα για τη σύνταξη του αλγορίθμου για την παραγωγή και χωροθέτηση τυχαίων σημείων με τη χρήση των μαθηματικών μοντέλων των Delaunay και Voronoi, είναι η WPF (Visual Basic. NET + XAML).

Η πληθώρα των βιβλιοθηκών που βασίζονται στο .NET Framework ήταν ένας από τους κυριότερους λόγους επιλογής της συγκεκριμένης γλώσσας στη πολυκριτηριακή ανάλυση. Μετά από σύγκριση πολλών εξωτερικών βιβλιοθηκών για την ολοκλήρωση του αλγορίθμου επιλέχθηκαν να εφαρμοστούν οι ακόλουθες:

Βιβλιοθήκη	Περιγραφή
GeoAPI	Βιβλιοθήκη προχωρημένων Γεωμετρικών κλάσεων και Συναρτήσεων
MIConvexHull	Βιβλιοθήκη συναρτήσεων και κλάσεων μαθηματικών μοντέλων γραφικών κυρτού κελύφους
NetTopologySuite	Βιβλιοθήκη συναρτήσεων και κλάσεων Γεωγραφικών Πληροφοριακών Συστημάτων
NetTopologySuite.I O	Βιβλιοθήκη συναρτήσεων και κλάσεων εισαγωγής και εξαγωγής δεδομένων Γεωγραφικών Πληροφοριακών Συστημάτων

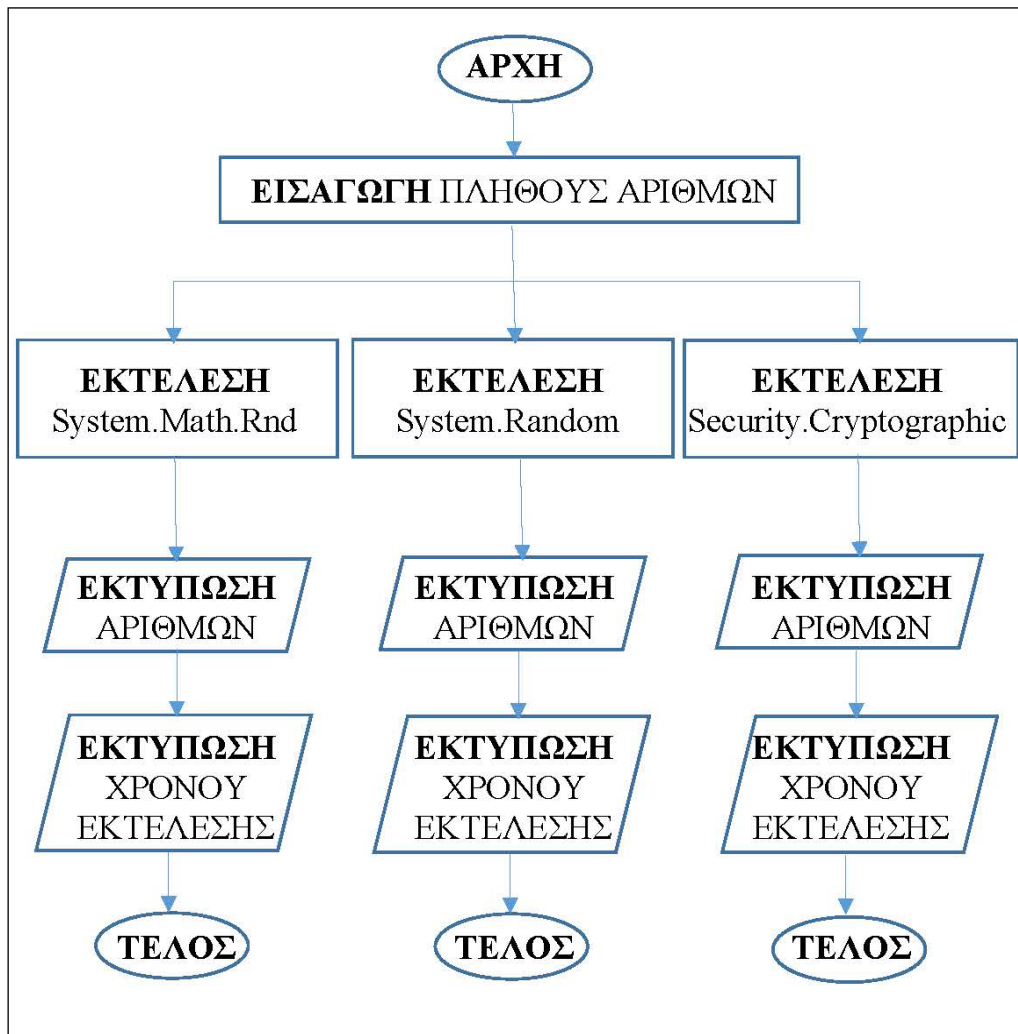
*Πίνακας 3.1: Βοηθητικές εξωτερικές βιβλιοθήκες που εφαρμόστηκαν*

Επιπλέον για την καλύτερη απεικόνιση των αποτελεσμάτων προστέθηκε η πλατφόρμα της Telerik “UI for WPF”. Η Telerik έχει εξελίξει στοιχεία ελέγχου (controls) που αποδίδουν στην εφαρμογή μια πιο προηγμένη, σύγχρονη και δυναμική παρουσίαση.



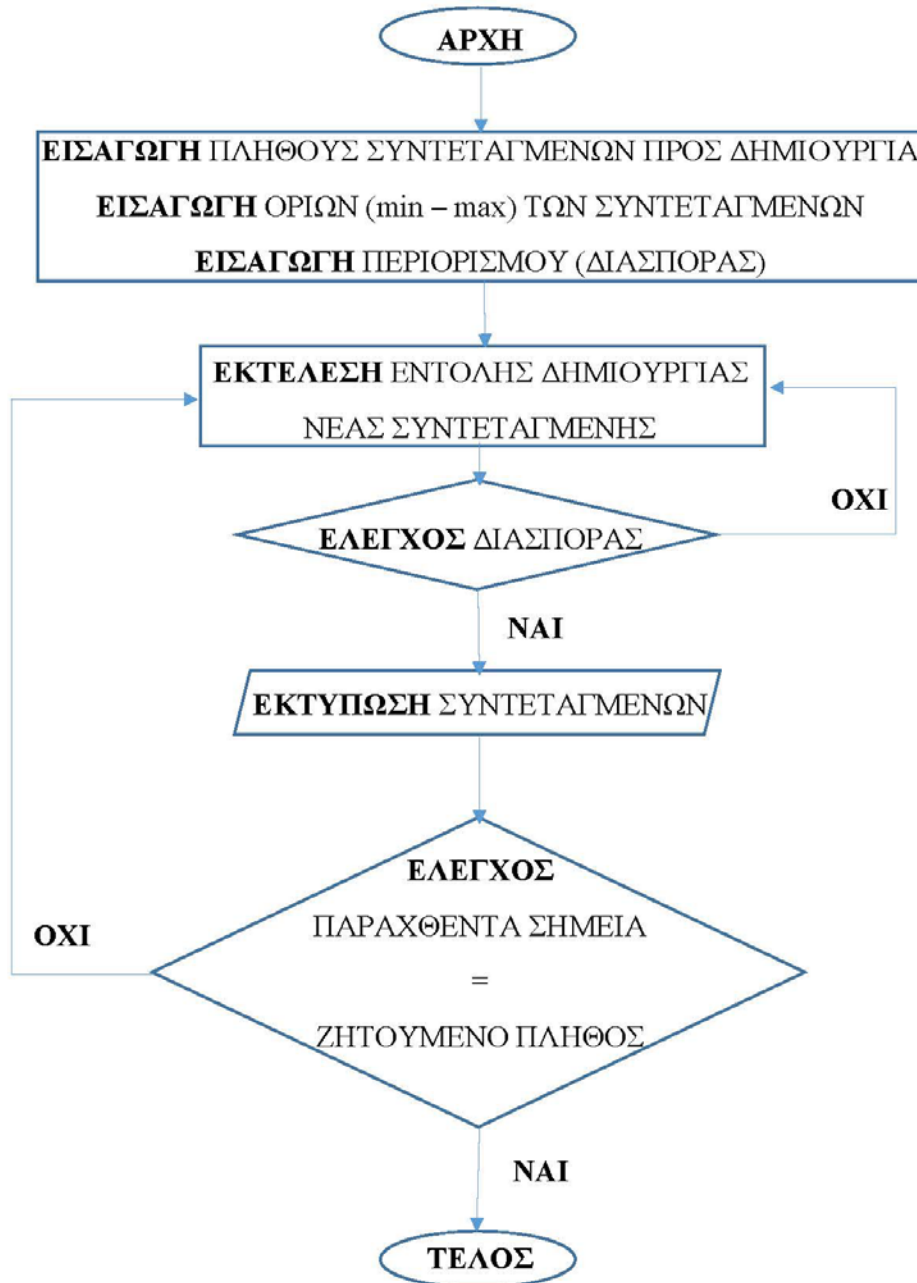
Κατά την ανάπτυξη του αλγορίθμου για την επίλυση του προβλήματος, δημιουργήθηκαν τρία διαγράμματα ροής τα οποία αποτελούν τη βασική σκέψη για τη λειτουργία του προγράμματος.

Στο πρώτο γίνεται ο έλεγχος και η σύγκριση μεταξύ των συναρτήσεων της τυχειότητας της παραγωγής αριθμών έτσι ώστε να επιλεγεί η καταλληλότερη. Στο σχήμα 3.1 παρουσιάζεται με τη μορφή διαγράμματος ροής, η συγκεκριμένη διαδικασία.



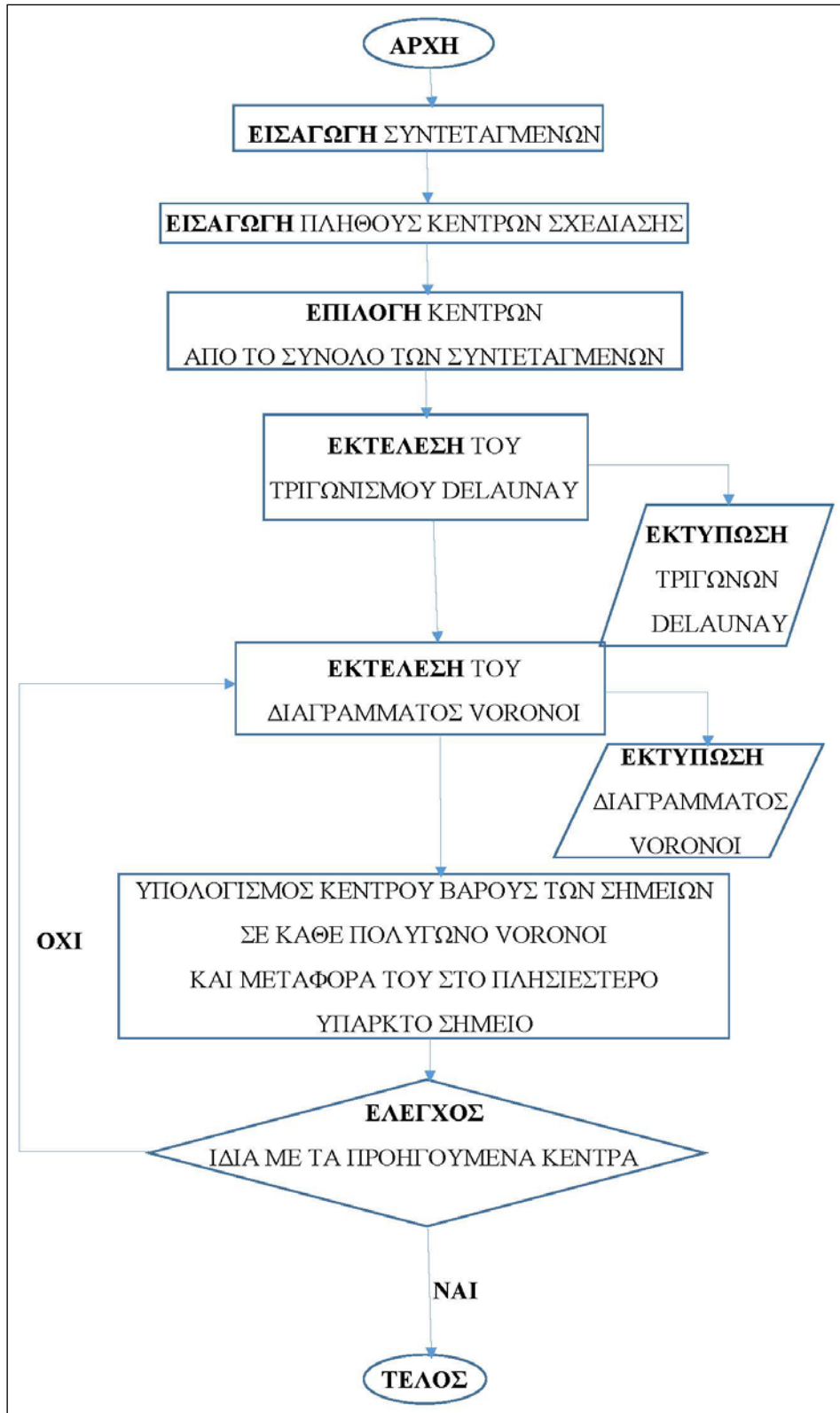
Σχήμα 3.1: Διάγραμμα ροής γενικής ακολουθίας 1<sup>ης</sup> φάσης του προγράμματος

Στο δεύτερο στάδιο γίνεται η δημιουργία συντεταγμένων και η απεικόνιση τους σε καμβά.



Σχήμα 3.2: Διάγραμμα ροής γενικής ακολουθίας 2ης φάσης του προγράμματος

Στο τελευταίο στάδιο εφαρμόζονται τα μαθηματικά μοντέλα του Delaunay και του Voronoi για την βελτιστοποίηση της χωροθέτησης των σημείων.



Σχήμα 3.3: Διάγραμμα ροής γενικής ακολουθίας 3ης φάσης του προγράμματος

## 3.2 Έλεγχος – Επιλογή συνάρτησης τυχαίων σημείων

### 3.2.1 Έλεγχος συναρτήσεων-κλάσεων

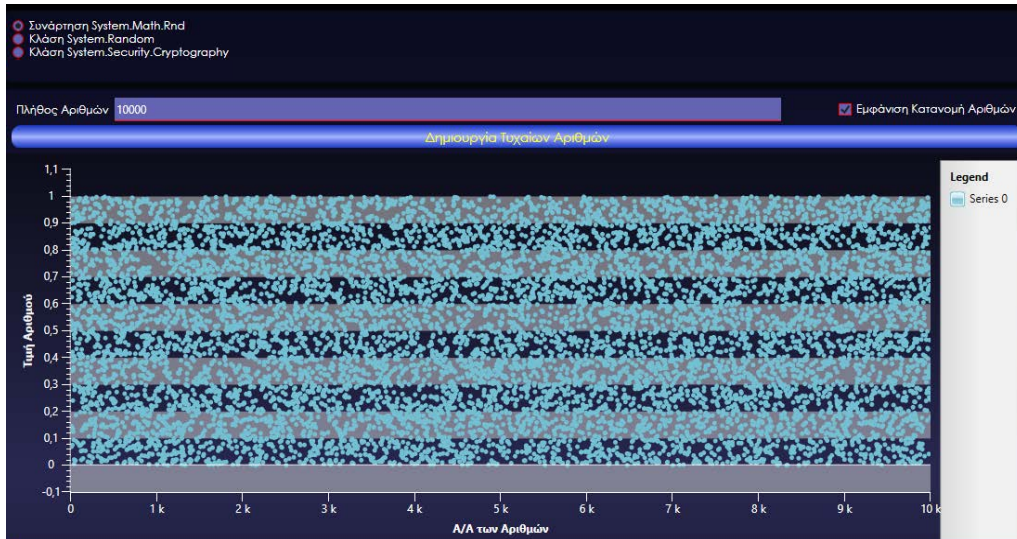
Στη θεωρία, αλλά και στην πράξη, αναφέρονται συχνά οι έννοιες "τυχαίο" και "ορισμένο". Για τα καλύτερα αποτελέσματα της εφαρμογής του αλγορίθμου, για ορισμένες συντεταγμένες στο χώρο που δίνονται από τον χρήστη, ο κώδικας θα εκτελεστεί με τυχαίες συντεταγμένες στο χώρο, που θα παραχθούν από μια γεννήτρια αριθμών. Για τον σκοπό αυτό, συγκρίθηκαν τρεις συναρτήσεις παραγωγής σημείων (από τις τέσσερις συναρτήσεις που αναφέρονται), ως προς την τυχειότητα και ως προς την ταχύτητα εκτέλεσής τους.

Στόχος της πρώτης φάσης της διπλωματικής, είναι να βρεθεί η **βέλτιστη συνάρτηση ή κλάση**, ώστε οι τιμές που θα επιστρέφονται από αυτές να είναι όσο το δυνατόν περισσότερο τυχαίες, δηλαδή η πιθανότητα να επιστρέψει ίδια τιμή με κάποια από τις προηγούμενες (ακόμα και σε προηγούμενες εκτελέσεις της εφαρμογής), να είναι όσο το δυνατόν μικρότερη.

Για το σκοπό αυτό συγκρίθηκαν οι ακόλουθες κλάσεις ή συναρτήσεις:

#### ➤ ***Συνάρτηση System.Math.Rnd***

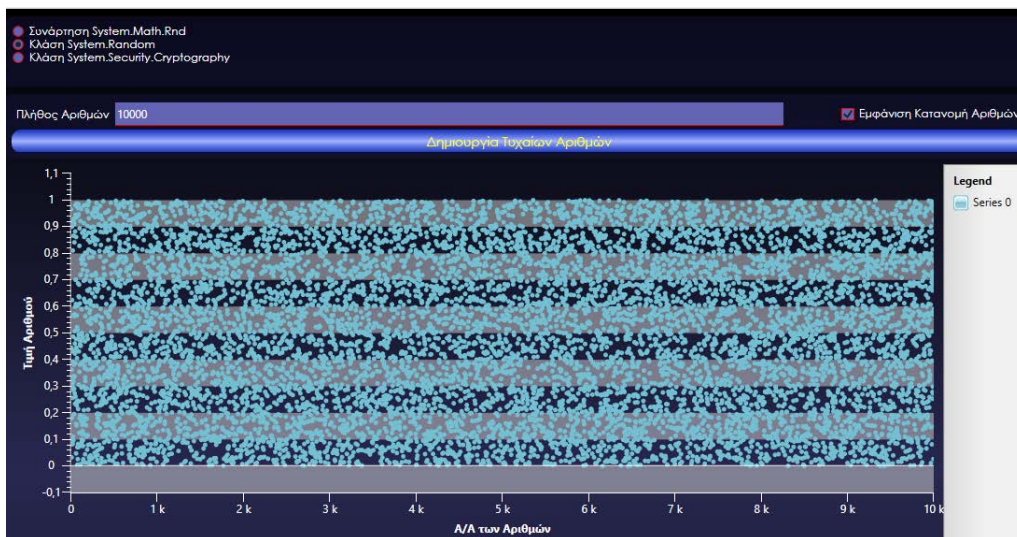
Η Math.Rnd είναι μια συνάρτηση της κλάσης Math που χρησιμοποιείται κατά κύριο λόγο εντός μαθηματικών μοντέλων, χωρίς ο προγραμματιστής να έχει μεγάλες απαιτήσεις για την απόλυτη τυχειότητα της τιμής που επιστέφει, αλλά θέλει να έχει πολύ μεγάλη ταχύτητα εκτέλεσης της συνάρτησης αυτής.



Εικόνα 3.1: Εκτέλεση της συνάρτησης *System.Math.Rnd* (για 10000 νούμερα)

➤ **Κλάση *System.Random***

Η συνάρτηση *System.Random* έχει στόχο την παραγωγή πραγματικά τυχαίων αριθμών, πριν εκτελεστεί η συνάρτηση *Random.Next*. Για την παραγωγή τυχαίου αριθμού πρέπει να αρχικοποιηθεί η κλάση. Ο αλγόριθμος της κλάσης *Random* βασίζεται στον αλγόριθμο γεννήτριας αφαιρετικού τυχαίου αριθμού του Donald E. Knuth.

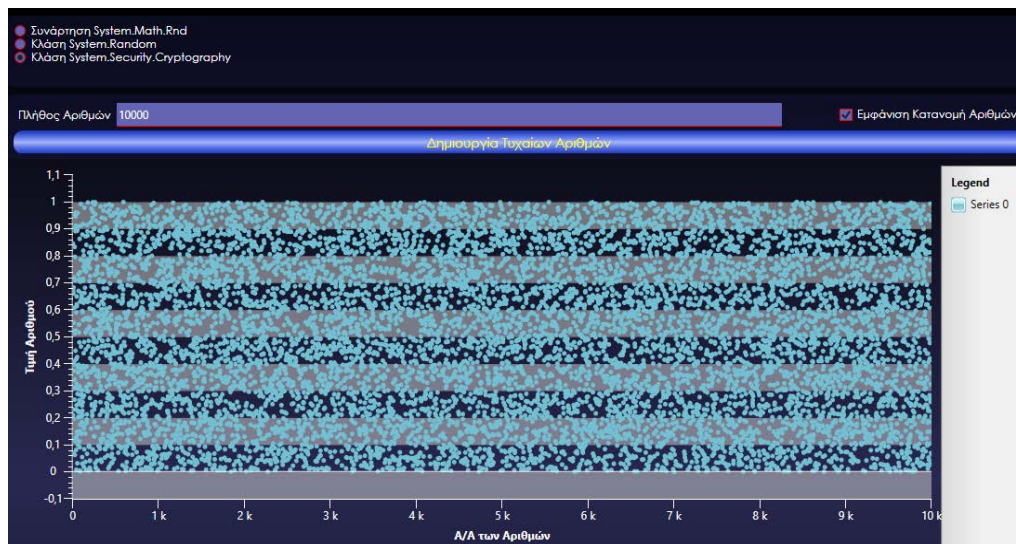


Εικόνα 3.2: Εκτέλεση της κλάσης *System.Random* (για 10000 νούμερα)



### ➤ Κλάση *System.Security.Cryptography*

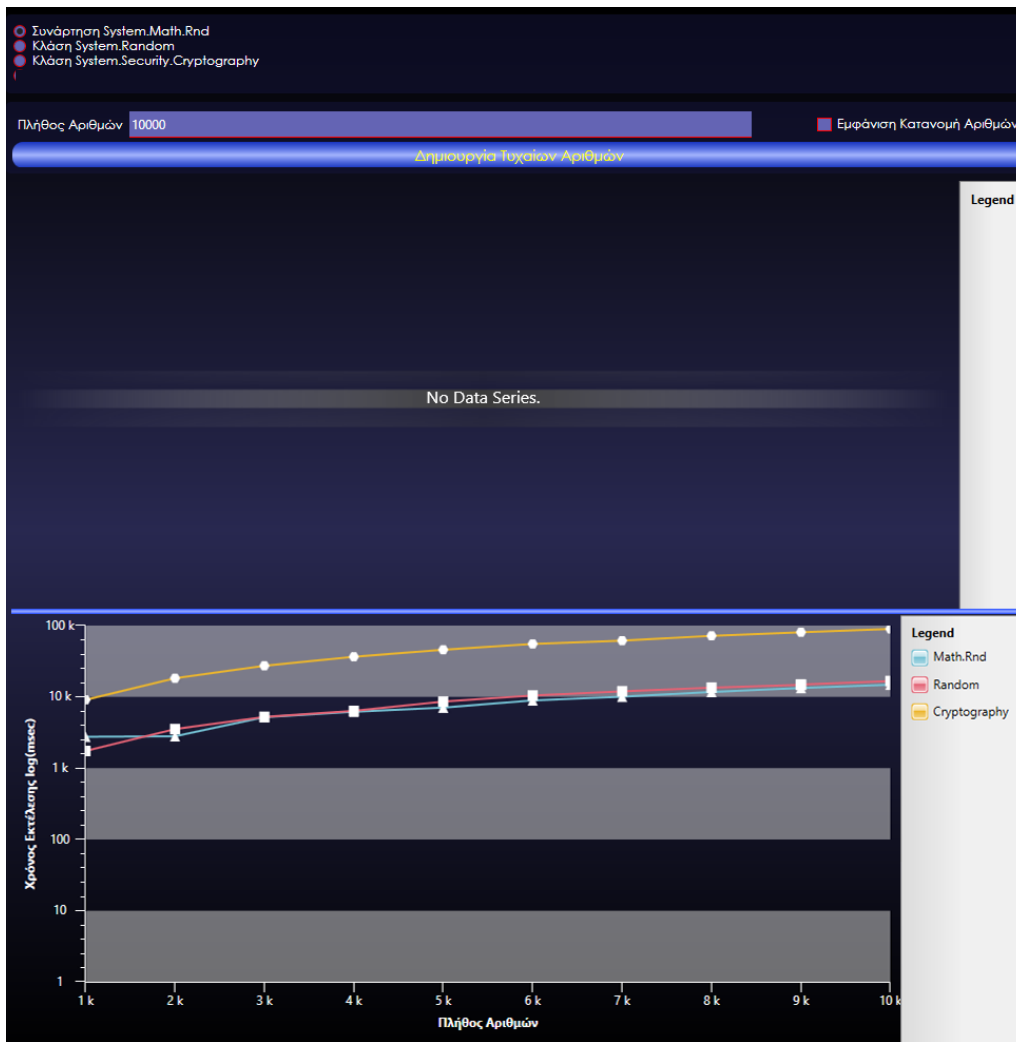
Η κλάση *System.Security.Cryptography* έχει ως στόχο να δημιουργεί, μοναδικά κρυπτογραφικά ισχυρές ακολουθίες τυχαίων αριθμών, πετυχαίνοντας ακόμα μεγαλύτερη τυχειότητα, αλλά ο χρόνος παραγωγής είναι μεγαλύτερος. Η συνάρτηση αυτή ενδείκνυται για εφαρμογές ασφαλείας.



Εικόνα 3.3: Εκτέλεση της κλάσης *Security.Cryptography* (για 10000 νούμερα)

Οι παραπάνω συναρτήσεις εκτελέστηκαν και συγκρίθηκαν μεταξύ τους, ως προς τη διασπορά (τυχειότητα) των αριθμών που παρήγαγαν (10.000 αριθμοί). Δεν εντοπίστηκε κάποιο μοτίβο επανάληψης, με αποτέλεσμα να μην καθιστά η τυχειότητα της κάθε συνάρτησης, κριτήριο επιλογής για την συνέχεια του αλγορίθμου στην παραγωγή συντεταγμένων.

Επομένως το βασικό κριτήριο επιλογής, της συνάρτησης παραγωγής τυχαίων αριθμών είναι ο χρόνος, γιατί ενώ οι αρχικοί χρόνοι παραγωγής τυχαίων αριθμών είναι μικροί, στη συνέχεια αυξάνονται εκθετικά, καθώς ζητείται από τον αλγόριθμο να παράξει συντεταγμένες (X,Y), να ελέγξει την πιθανή επανάληψη των συντεταγμένων και να τις απεικονίσει σε καρτεσιανό σύστημα αναφοράς. Επομένως μια μικρή διαφορά στην αρχική παραγωγή των αριθμών, μπορεί να είναι μεγάλη διαφορά στο χρόνο εκτέλεσης του αλγορίθμου.



Εικόνα 3.4: Διάγραμμα χρόνου-πλήθους αριθμών

Κατά την εκτέλεση ζητήθηκε από τον αλγόριθμο να συγκρίνει τους χρόνους παραγωγής αριθμών (από 1.000 έως 10.000 αριθμούς με βήμα 1.000) και προέκυψε το διάγραμμα χρόνου – πλήθους (εικόνα 3.4), όπου με γαλάζια γραμμή απεικονίζεται η `System.Math.Rnd`, με κόκκινη γραμμή η `System.Random` και με κίτρινη η `System.Security.Cryptography`.

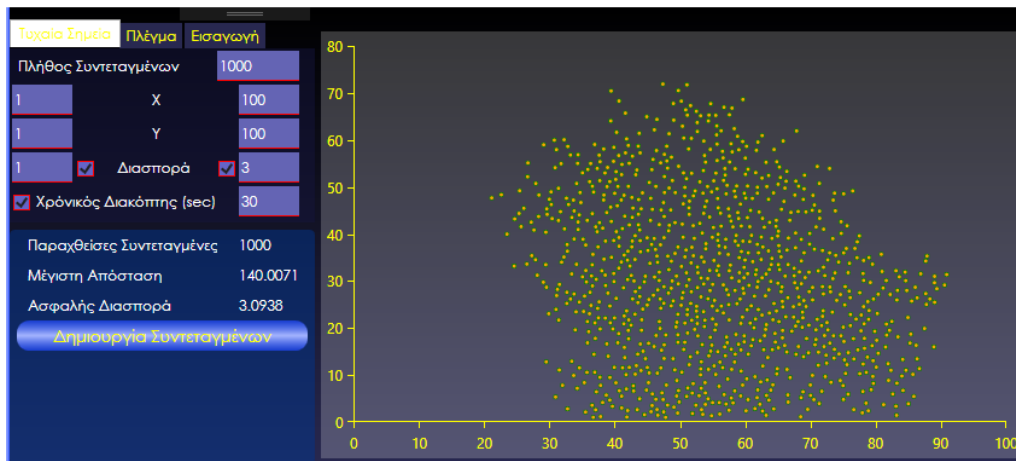
Παρατηρούμε ότι στη συνάρτηση `System.Security.Cryptography` σε κάθε βήμα των 1.000 αριθμών, ο χρόνος παραγωγής αριθμών είναι αισθητά πιο μεγάλος, σε σχέση με τις άλλες δυο.

Οι συναρτήσεις `System.Math.Rnd` και `System.Random` έχουν μικρή διαφορά μεταξύ τους, με την `System.Math.Rnd` να υπερισχύει από άποψη χρόνου, σχεδόν σε όλα τα βήματα των 1.000 αριθμών.

### 3.2.2 Δημιουργία – Εισαγωγή Σημείων

Το δεύτερο στάδιο της εφαρμογής, δημιουργεί τα σημεία που θα χρησιμοποιηθούν στο τρίτο στάδιο της εφαρμογής. Εδώ δίνονται τρεις επιλογές

#### 3.2.2.1 Δημιουργία τυχαίων συντεταγμένων



Εικόνα 3.5: Παράδειγμα παραγωγής τυχαίων συντεταγμένων

Στο πεδίο «Πλήθος Συντεταγμένων» εισάγονται τα σημεία που θα δημιουργηθούν βασιζόμενα στην επιλεγμένη κλάση ή στη συνάρτηση του προηγούμενου σταδίου. Το πεδίο ορισμού του πεδίου είναι οποιοσδήποτε θετικός ακέραιος από 1 έως 2147483647.

Λόγω του ότι η εμφάνιση των παραγόμενων σημείων είναι δυναμική (κάθε σημείο που δημιουργείται, εμφανίζεται απευθείας στον καμβά) ο πολύ μεγάλος αριθμός συντεταγμένων προς δημιουργία μπορεί να προκαλέσει επιβράδυνση ή διακοπή του προγράμματος.

Στα πεδία αριστερά και δεξιά του «X» ορίζεται η ελάχιστη και μέγιστη τιμή του άξονα X.

Στα πεδία αριστερά και δεξιά του «Y» ορίζεται η ελάχιστη και μέγιστη τιμή του άξονα Y.

Το πεδίο ορισμού των τεσσάρων αυτών πεδίων είναι οποιοσδήποτε πραγματικός αριθμός από  $-1.7976931348623157E+308$  έως  $1.7976931348623157E+308$ . Η χρήση πολύ μεγάλων αριθμών ίσως προκαλέσει διακοπή της εφαρμογής.

Τα επόμενα πεδία σχετίζονται με τη διασπορά των σημείων. Ως διασπορά ορίζεται η κατανομή των σημείων στον καμβά. Το αριστερό πεδίο (που ενεργοποιείται όταν είναι επιλεγμένο το διπλανό του, πεδίο



ενεργοποίησης) ορίζει την ελάχιστη απόσταση που θα έχουν όλα τα σημεία μεταξύ τους. Κάθε νέο σημείο ελέγχεται για την απόσταση που έχει με όλα τα προηγούμενα δημιουργημένα σημεία. Εάν κάποια από τις αποστάσεις είναι μικρότερη από την τιμή του πεδίου τότε αυτή απορρίπτεται.

Το δεξί πεδίο (που ενεργοποιείται όταν είναι επιλεγμένο το διπλανό του, πεδίο ενεργοποίησης) ορίζει τη μέγιστη απόσταση που μπορεί να έχει ένα σημείο από τα υπόλοιπα σημεία. Κάθε νέο σημείο ελέγχεται για την απόσταση που έχει με όλα τα προηγούμενα δημιουργημένα σημεία. Εάν δεν υπάρξει ένα σημείο με μικρότερη απόσταση από την τιμή του πεδίου τότε αυτή απορρίπτεται.

Το πεδίο «Χρονικός διακόπτης (sec)» (εάν είναι τσεκαρισμένο το πεδίο ενεργοποίησης) μας βοηθά στην απρόσκοπτη λειτουργία του προγράμματος στις περιπτώσεις που οι τιμές διασποράς που έχουν δώσει δυσκολεύει τη δημιουργία νέων σημείων. Κατά τη λειτουργία του συγκεκριμένου σταδίου της εφαρμογής μετράται ο χρόνος μεταξύ των δημιουργημένων σημείων. Εάν ο χρόνος των δευτερολέπτων που έχει οριστεί στο πεδίο ξεπεραστεί τότε εμφανίζει μήνυμα εάν ο χρήστης θέλει να συνεχιστεί τη προσπάθεια ή να σταματήσει τη διαδικασία χωρίς να έχει ολοκληρωθεί η δημιουργία των ζητηθέντων σημείων.

Στο γραφικό περιβάλλον του συγκεκριμένου σταδίου ακολουθούν πεδία με πληροφορίες. Το πρώτο πεδίο αρχίζει να ενημερώνεται μετά την εκκίνηση της δημιουργίας των συντεταγμένων και ενημερώνει για τον αριθμό των σημείων που έχουν παραχθεί.

Το επόμενο πεδίο δηλώνει την «μέγιστη απόσταση» που μπορεί να οριστεί. Ουσιαστικά είναι η μέγιστη απόσταση εντός του καμβά, δηλαδή η διαγώνιος του. Η πληροφορία αυτή μας βοηθά να δηλώσουμε το πεδίο της μέγιστης απόστασης της διασποράς.

Όπως και το προηγούμενο, έτσι και το τελευταίο πεδίο πληροφορίας έχει σχέση με την διασπορά. Η «ασφαλής διασπορά» δηλώνει τη μέγιστη θεωρητική τιμή που μπορεί να δεχτεί η ελάχιστη απόσταση διασποράς ώστε να μπορέσουν να παραχθούν όλα τα ζητηθέντα σημεία. Η λογική υπολογισμού της ασφαλούς διασποράς στηρίζεται στην μικρότερη απόσταση που θα υπήρχε σε έναν κάνναβο με σημεία και διαστάσεις καμβά ίδια με αυτά που έχει ζητήσει ο χρήστης προηγουμένως.

### 3.2.2.2 Δημιουργία συντεταγμένων σταθερής απόστασης (Grid)

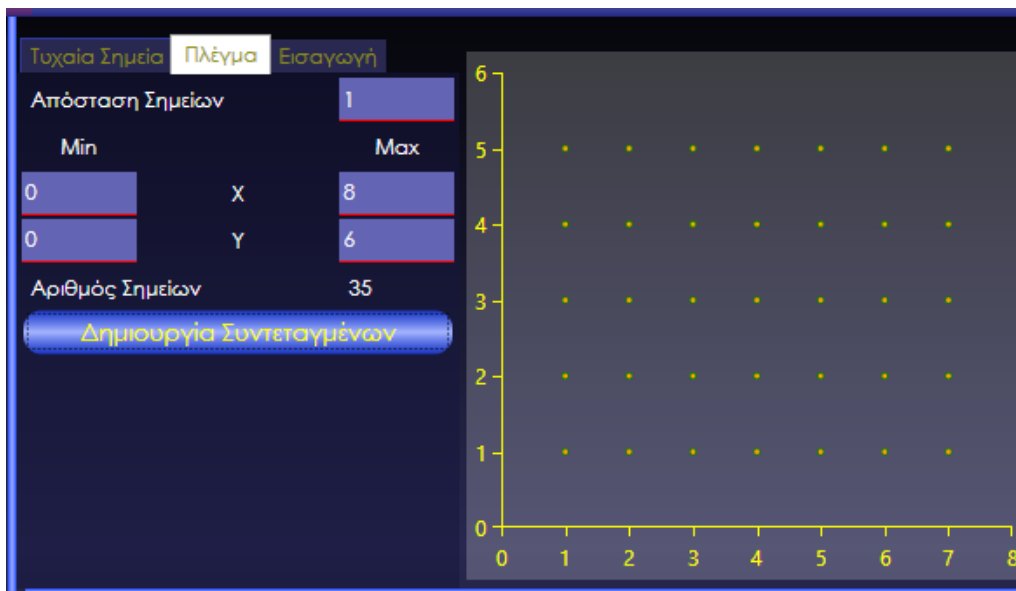
Σε αυτή την επιλογή εισάγεται η απόσταση ανάμεσα στα σημεία στο πεδίο εισαγωγής «Απόστασή Σημείων». Το πεδίο ορισμού του πεδίου είναι οποιοσδήποτε θετικός ακέραιος από 1 έως 2147483647.

Στα πεδία αριστερά και δεξιά του «X» ορίζουμε την ελάχιστη και μέγιστη τιμή του άξονα X.

Στα πεδία αριστερά και δεξιά του «Y» ορίζουμε την ελάχιστη και μέγιστη τιμή του άξονα Y.

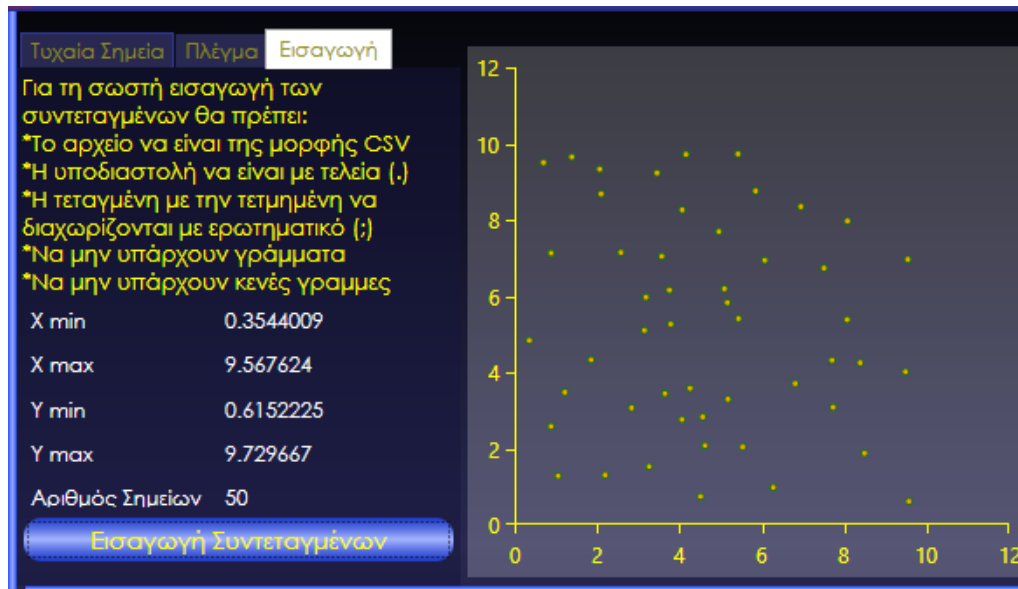
Το πεδίο ορισμού του πεδίου είναι οποιοσδήποτε θετικός ακέραιος από 1 έως 2147483647. Η χρήση πολύ μεγάλων αριθμών ίσως προκαλέσει διακοπή της εφαρμογής.

Παρακάτω, εμφανίζεται το πεδίο που παρέχει ως πληροφορία το σύνολο των σημείων που δημιουργήθηκαν.



Εικόνα 3.6: Παράδειγμα παραγωγής συντεταγμένων σταθερής απόστασης

### 3.2.2.3 Εισαγωγή Σημείων



Εικόνα 3.7: Παράδειγμα παραγωγής τυχαίων συντεταγμένων

Η τελευταία επιλογή μας δίνει την δυνατότητα να εισαχθούν σημεία που έχουν εξαχθεί από μια τρίτη εφαρμογή. Επιλέγοντας το κουμπί «Εισαγωγή συντεταγμένων» εμφανίζεται ένα παράθυρο διαλόγου που δίνει την δυνατότητα να επιλεγεί το προς εισαγωγή αρχείο.

Για τη σωστή εισαγωγή θα πρέπει το αρχείο είτε να είναι της μορφής csv, είτε να πληροί τους ακόλουθους κανόνες.

- Η υποδιαστολή των αριθμών να συμβολίζεται με τελεία (.)
- Η τετμημένη και η τεταγμένη να χωρίζονται με ερωτηματικό (;)
- Να μην υπάρχουν γράμματα ή άλλες πρόσθετες πληροφορίες.
- Να μην υπάρχουν κενά διαστήματα
- Να μην υπάρχουν κενές γραμμές

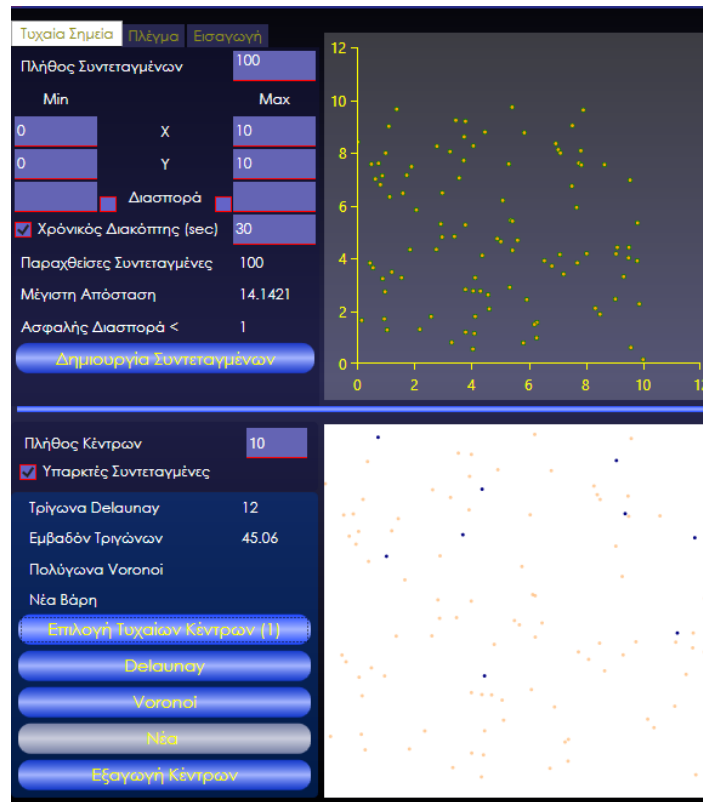
Μετά την εισαγωγή των συντεταγμένων, ενημερώνονται τα πεδία «Xmin» και «Xmax» που δίνουν την πληροφορία της ελάχιστης και μέγιστης τιμής του άξονα X, και τα πεδία «Ymin» και «Ymax» που δίνουν την πληροφορία της ελάχιστης και μέγιστης τιμής του άξονα Y. Στο τέλος εμφανίζεται το πλήθος των εισαχθέντων σημείων.

### 3.2.3 Παραγωγή Διαγράμματος Voronoi

Αφού έχουν δημιουργηθεί τα σημεία όπως έχει αναλυθεί στο κεφάλαιο 3.2.2 ξεκινά η δημιουργία του διαγράμματος Voronoi

Στο πεδίο «επιλεγμένες συντεταγμένες» συμπληρώνεται το πλήθος των σημείων που θέλουμε να έχουμε έως κεφαλικά. Το πεδίο ορισμού του συγκεκριμένου πεδίου είναι όλοι οι θετικοί ακέραιοι από 3 έως το πλήθος των συντεταγμένων που όρισε ο χρήστης στο προηγούμενο στάδιο.

Το επόμενο πεδίο ενεργοποίησης «Υπαρκτές Συντεταγμένες» ορίζει εάν το κέντρο βάρους των σημείων θα δημιουργεί ένα νέο ιδεατό σημείο για τη βελτιστοποίηση ή εάν θα βρίσκει το πλησιέστερο υπαρκτό σημείο.



Εικόνα 3.8: Παράδειγμα δημιουργίας τυχαίων κέντρων

Τα παρακάτω πεδία πληροφορίας δείχνουν τα τρίγωνα Delaunay που δημιουργούνται, το συνολικό εμβαδόν των Τριγώνων αυτών, και το πλήθος των πολυγώνων Voronoi που δημιουργούνται.

Το επόμενο βήμα είναι η εκτέλεση της δημιουργίας συντεταγμένων.

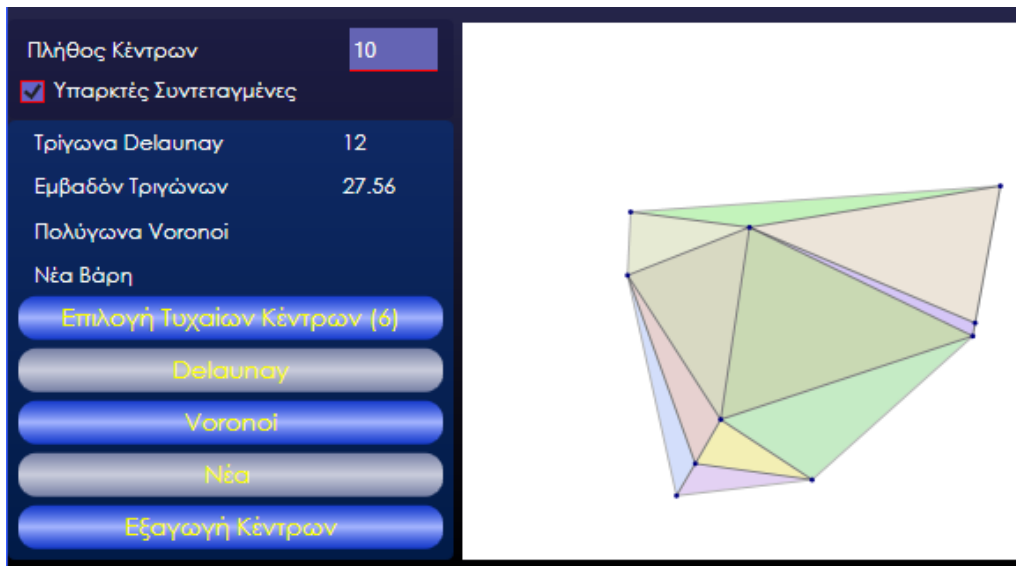
Σε περίπτωση που επαναλυφθεί το πάτημα του κουμπιού για τη «δημιουργία των συντεταγμένων τότε στα ίδια αρχικά σημεία επιλέγονται άλλα κεφαλικά για την εφαρμογή του αλγορίθμου.



Εικόνα 3.9: Παράδειγμα δημιουργίας νέων τυχαίων κέντρων

Στη συνέχεια μπορεί να επιλεγεί να εκτελεστεί είτε ο αλγόριθμος του Delaunay είτε του Voronoi.

Επιλέγοντας το πλήκτρο «Delaunay» εμφανίζονται τα τρίγωνα βασισμένα στα κεφαλικά σημεία.

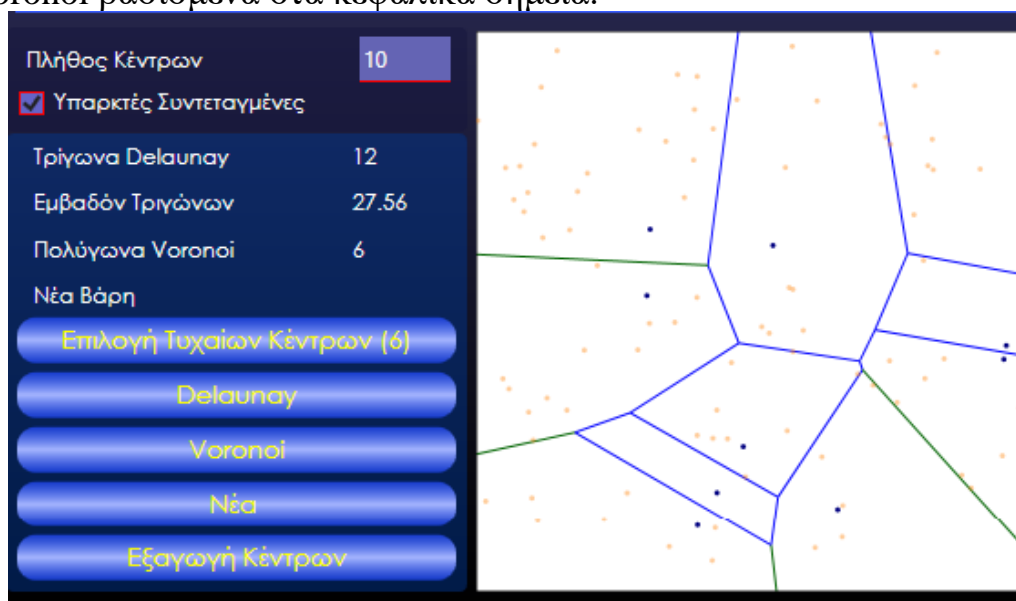


Εικόνα 3.10: Παράδειγμα τριγωνισμού Delaunay

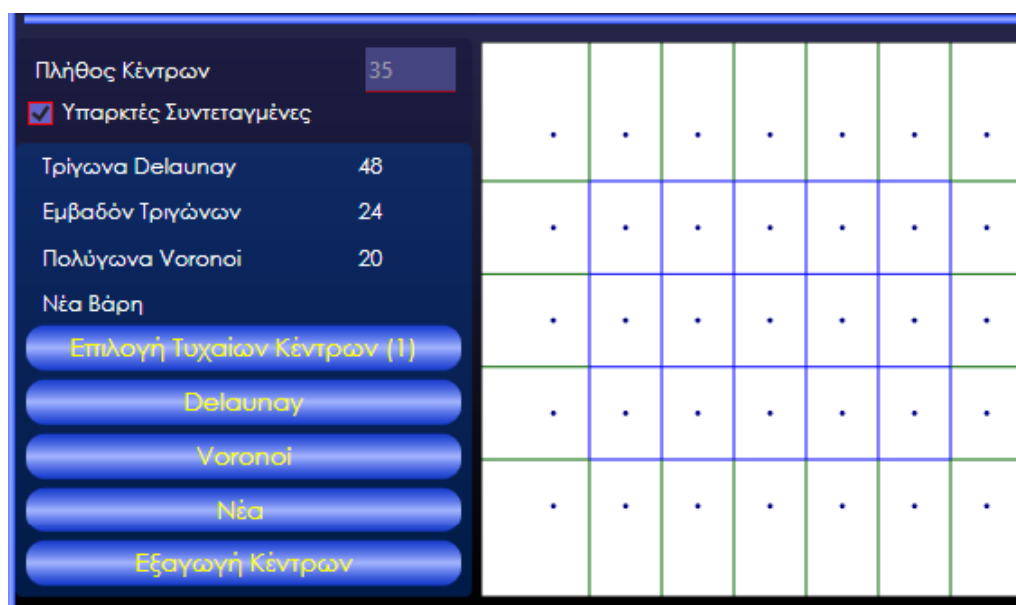


Εικόνα 3.11: Παράδειγμα τριγωνισμού Delaunay στον κάρναβο

Επιλέγοντας το πλήκτρο «Voronoi» εμφανίζονται τα πολύγωνα του Voronoi βασισμένα στα κεφαλικά σημεία.



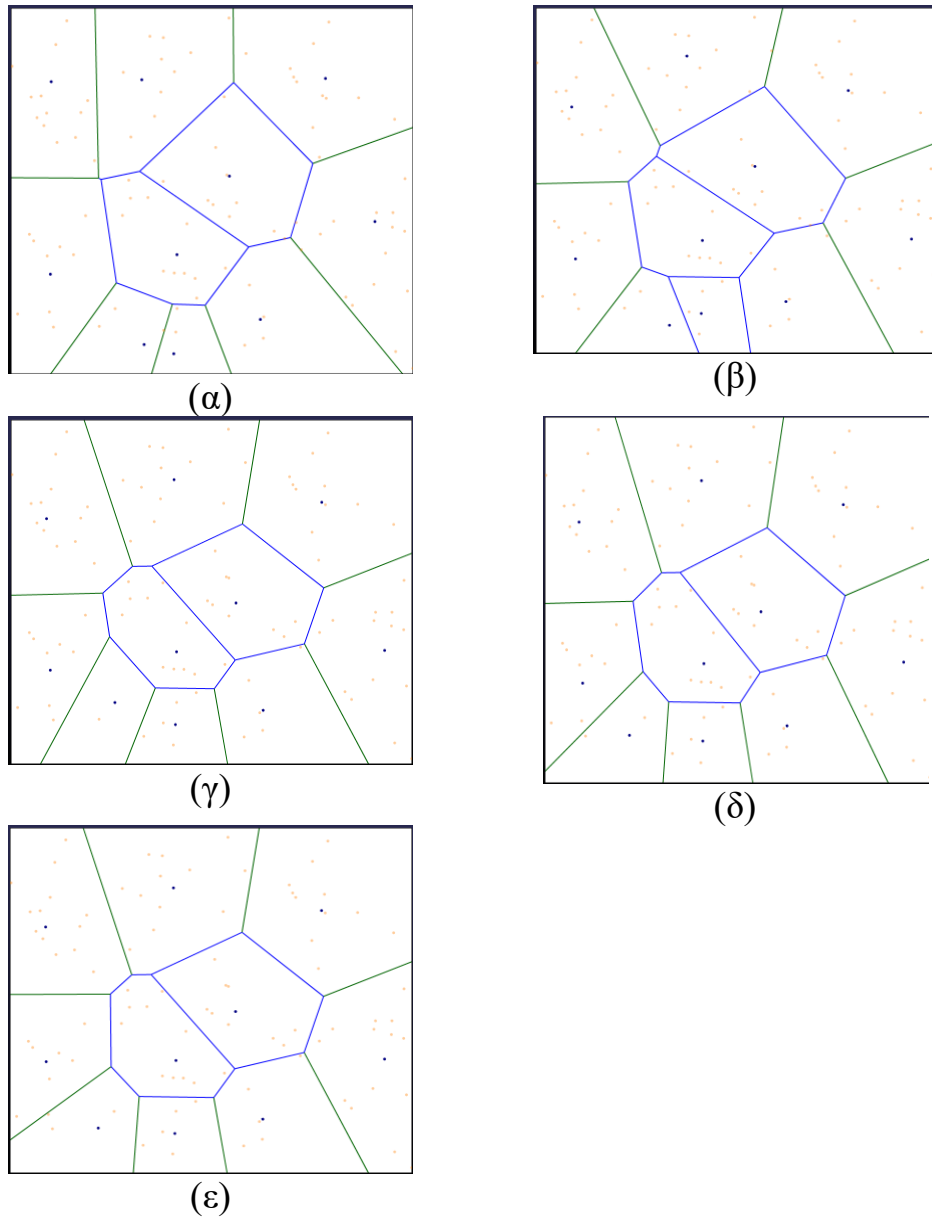
Εικόνα 3.12: Παράδειγμα δημιουργίας πολυγώνων Voronoi



Εικόνα 3.13: Παράδειγμα δημιουργίας πολυγώνων Voronoi στον κάνναβο

Επιλέγοντας το πλήκτρο «Νέα Βάρη» γίνεται ο υπολογισμός του μέσου όρου των σημείων που περικλείονται σε κάθε πολύγωνα ώστε να εξαχθεί το κέντρο βάρους αυτών. Εφόσον έχει επιλεγεί το πεδίο ενεργοποίησης «Actual Point (Nearest)» βρίσκεται το πλησιέστερο σημείο στο κέντρο βάρους και ορίζεται ως κεφαλικό.

Επαναλαμβάνοντας το πάτημα του κουμπιού «Νέα Βάρη» βελτιώνονται συνεχώς τα πολύγωνα Voronoi μέχρι να μην υπάρχει αλλαγή (σχήμα 3.1).



Σχήμα 3.4: Στάδια βελτιστοποίησης πολυγώνων Voronoi





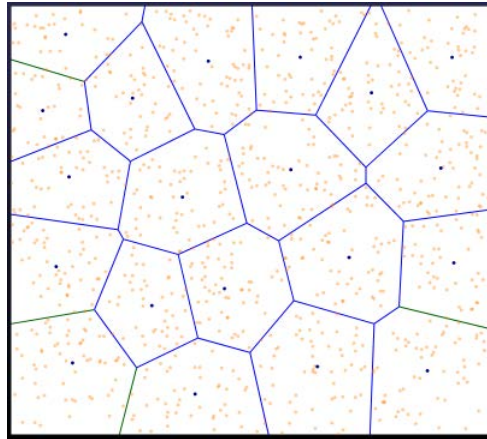
## ΚΕΦΑΛΑΙΟ 4 ΣΥΜΠΕΡΑΣΜΑΤΑ – ΠΡΟΤΑΣΕΙΣ

### 4.1 Συμπεράσματα

Κατά τη χρήση του προγράμματος και έπειτα από διαδοχικές επαναλήψεις με διαφορετικές παραμέτρους εισόδου παρατηρήθηκε ότι ο βέλτιστος αλγόριθμος για την παραγωγή τυχαίων αριθμών συνυπολογιζόμενου του χρόνου παραγωγής αριθμών και αυξημένης τυχειότητας είναι η κλάση **Random**.

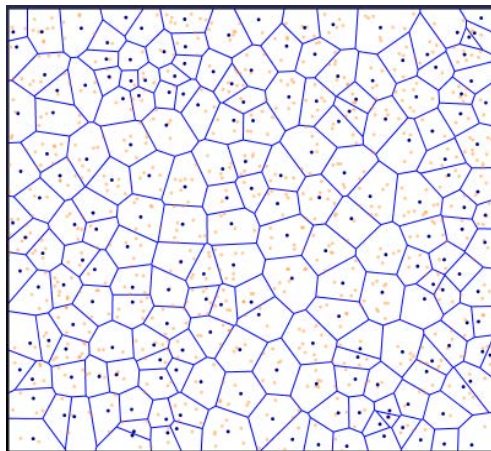
Στο δεύτερο στάδιο ο χρόνος παραγωγής τυχαίων σημείων (συντεταγμένων) είναι πολύ μεγαλύτερος από τον χρόνο που έδειξε ο έλεγχος ταχύτητας παραγωγής τυχαίων αριθμών στο πρώτο στάδιο. Αυτό οφείλεται στο ότι παράγει τυχαία συντεταγμένη και κατά συνέπεια δυο αριθμούς εντός του συγκεκριμένου καννάβου. ( $X_{min}$ - $X_{max}$ ,  $Y_{min}$ - $Y_{max}$ ). Εάν προστεθούν περιορισμοί απόστασης σημείου από σημείο τότε αυτός ο χρόνος αυξάνεται ακόμα περισσότερο. Εάν δηλαδή οριστεί η μικρότερη απόσταση διασποράς τότε η κάθε νέα συντεταγμένη που θα δημιουργηθεί θα ελεγχθεί και δε θα είναι πιο κοντά από την τιμή που ορίστηκε. Το ίδιο συμβαίνει αν οριστεί και μεγαλύτερη απόσταση διασποράς. Το αποτέλεσμα είναι ότι ο χρόνος αυξάνεται ακόμα πιο πολύ όσο ζητούνται περισσότεροι περιορισμοί.

Στο τρίτο στάδιο της εφαρμογής παρατηρείται ότι έχοντας μεγάλο πλήθος συντεταγμένων και μικρό πλήθος κέντρων, οι επαναλήψεις για τη βέλτιστη θέση αυξάνονται, αφού το εμβαδόν του πολυγώνου είναι μεγάλο και περιέχει πολλά σημεία οπότε και υπάρχουν πολλές εναλλαγές στη θέση κέντρου (σχήμα 4.1). Αυτό βελτιστοποιεί το αποτέλεσμα.



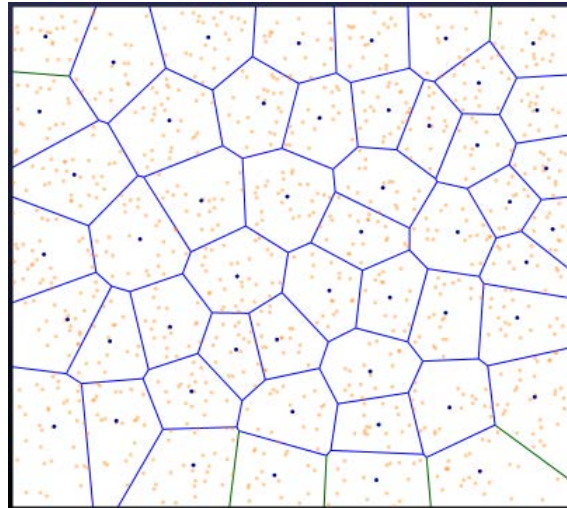
Σχήμα 4.1: Εφαρμογή αλγορίθμου για 1000 σημεία 20 κέντρα

Ένα διαφορετικό αποτέλεσμα προκύπτει όταν υπάρχει μεγάλο πλήθος συντεταγμένων αλλά και μεγάλο πλήθος κέντρων, δηλαδή οι επαναλήψεις για τη βελτιστοποίηση μειώνονται. Το εμβαδόν του πολυγώνου μικραίνει, με αποτέλεσμα να μην περιέχει πολλά σημεία οπότε και να μην υπάρχουν πολλές εναλλαγές στη θέση κέντρου. Το αποτέλεσμα είναι να υπάρχει ανομοιομορφία μεταξύ των εμβαδών των πολυγώνων (σχήμα 4.2).

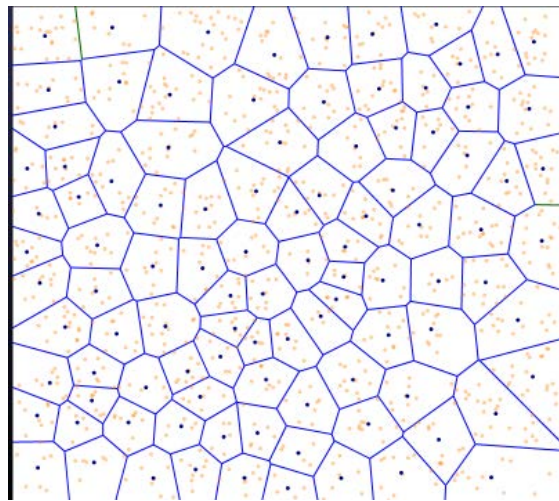


Σχήμα 4.2: Εφαρμογή αλγορίθμου για 1000 σημεία 200 κέντρα

Ένα ενδιαφέρον συμπέρασμα προκύπτει όταν ζητείται το κέντρο του πολυγώνου να μην είναι ένα από τα δοθέντα σημεία. Το πρόγραμμα υπολογίζει το κέντρο βάρους του πολύγωνα (από τα σημεία που περιέχει) και δεν το μεταφέρει στο κοντινότερο υπαρκτό σημείο. Το αποτέλεσμα γίνεται το καλύτερο δυνατόν βάσει των δοθέντων στοιχείων εσόδου (σχήματα 4.3, 4.4).



Σχήμα 4.3: Εφαρμογή αλγορίθμου για 1000 σημεία 50 κέντρα “no actual point”



Σχήμα 4.4: Εφαρμογή αλγορίθμου για 1000 σημεία 100 κέντρα “no actual point”

Μια τέτοια εφαρμογή μπορεί να χρησιμοποιηθεί από σεισμολόγους καθώς υπάρχει η δυνατότητα εισαγωγής των συντεταγμένων όλων των σεισμών (που ενδιαφέρουν τον χρήστη), στη συνέχεια να επιλέξει τον επιθυμητό αριθμό σειсмоγράφων (αισθητήρες σεισμού) και κατά την εκτέλεση, θα δημιουργηθούν ζώνες παρακολούθησης των σεισμών όσες και ο αριθμός των σειсмоγράφων.

Παρόμοια εφαρμογή μπορεί να υπάρξει στην περίπτωση που εισάγονται οι θέσεις αισθητήρων (καπνού, θρύβου, φωτάς, κίνησης κ.α.) και στη συνέχεια επιλέγεται ο επιθυμητός αριθμός κέντρων λήψης σήματος που θα εγκατασταθεί πρόσθετα στους αισθητήρες.

Άλλη εφαρμογή μπορεί να είναι η αναβάθμιση σχολείων σε πρότυπα.

Εισάγονται οι ονομαστικές συντεταγμένες των σχολείων μιας περιοχής ή ολόκληρης της χώρας και στη συνέχεια ο αριθμός των πρότυπων σχολείων προς αναβάθμιση. Η εκτέλεση του αλγορίθμου θα δώσει τις θέσεις των σχολείων που προτείνονται προς αναβάθμιση σε πρότυπα.

## 4.2 Προτάσεις

Στον χώρο του προγραμματισμού οι τροποποιήσεις που μπορούν να γίνουν σε μια εφαρμογή από τον προγραμματιστή της, είναι αναρίθμητες και εξαρτώνται από την έμπνευση τη δημιουργικότητα και την κρίση του. Κατά την ολοκλήρωση του αλγορίθμου προέκυψαν ιδέες για τη βελτίωση και τον εμπλουτισμό της εφαρμογής. Μερικές από τις προτάσεις που προτείνεται να προστεθούν είναι:

Στο πρώτο στάδιο θα μπορούσε να τροποποιηθεί το διάγραμμα απεικόνισης χρόνου-πλήθους, ώστε και ο άξονας του πλήθους να είναι λογαριθμικός και κατά συνέπεια η σύγκριση των τιμών να είναι ευκολότερη.

Επίσης μπορεί να προστεθεί στο δεύτερο στάδιο η δυνατότητα εισαγωγής σημείων σε κάρναβο (grid) όπως επίσης και εισαγωγή σημείων / συντεταγμένων που έχει εξάγει ο χρήστης από άλλη εφαρμογή (σε .csv, .txt, .dxf)

Η διασύνδεση μεταξύ συντεταγμένων του συστήματος ΕΓΣΑ '87 που έχει εισάγει ο χρήστης στην εφαρμογή με απεικόνιση τους σε χάρτες του κτηματολογίου ή άλλων χαρτών (Google Earth, Google Maps, Bing Maps κ.α.), όπως επίσης και οι επισημάνσεις, των παραγόμενων κέντρων από την εκτέλεση της εφαρμογής.

Η εισαγωγή περισσότερων παραμέτρων όπως μέγιστη ή ελάχιστη ακτίνα κέντρων πολυγώνων από τον χρήστη θα δώσει περισσότερα εναλλακτικά αποτελέσματα.

Η προσθήκη αυτόματης εκτέλεσης των επαναλήψεων των νέων βαρών μέχρι τα βέλτιστα πολύγωνα ώστε να αποφευχθεί το συνεχόμενο πάτημα του κουμπιού «Νέα Βάρη».

Η βιβλιοθήκη MIPConvexHull που εμπεριέχει τα μαθηματικά μοντέλα Voronoi & Delanauy σε 2D, εμπεριέχει και σε 3D. Μελλοντικά λοιπόν εφαρμόζοντας αρκετές τροποποιήσεις στο δεύτερο και τρίτο στάδιο του προγράμματος θα μπορούσε να εφαρμοσθεί χωροθέτηση τυχαίων ή μη σημείων στο χώρο (σε 3D απεικόνιση).

## ΒΙΒΛΙΟΓΡΑΦΙΑ

### Ξενόγλωσση

Candeloro, M., Lekkas, A. M., & Sørensen, A. J. (2017). A Voronoi-diagram-based dynamic path-planning system for underactuated marine vessels. *Control Engineering Practice*, 61, 41-54.

Chang, C. Y., Shih, K. P., Lee, S. C., & Chang, S. W. (2006, October). RGP: Active route guiding protocol for wireless sensor networks with obstacles. In *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on* (pp. 367-376). IEEE.

Ide, S., Sumitsuji, S., Yamaguchi, O., & Sakata, Y. (2017). Cardiac computed tomography-derived myocardial mass at risk using the voronoi-based segmentation algorithm: A histological validation study. *Journal of Cardiovascular Computed Tomography*.

Kisore, N. R., & Koteswaraiyah, C. B. (2017). Improving ATM coverage area using density based clustering algorithm and voronoi diagrams. *Information Sciences*, 376, 1-20.

Zhao, Q. H., Li, X. L., Li, Y., & Zhao, X. M. (2017). A fuzzy clustering image segmentation algorithm based on Hidden Markov Random Field models and Voronoi Tessellation. *Pattern Recognition Letters*, 85, 49-55.

Wang, M. T. (2016). Nearest neighbor query processing using the network voronoi diagram. *Data & Knowledge Engineering*, 103, 19-43.

De Kok, T., Van Kreveld, M., & Löffler, M. (2007). Generating realistic terrains with higher-order Delaunay triangulations. *Computational Geometry*, 36(1), 52-65.

Feng, D., Tsolakis, C., Chernikov, A. N., & Chrisochoides, N. P. (2015). Scalable 3D hybrid parallel Delaunay image-to-mesh conversion algorithm for distributed shared memory architectures. *Procedia Engineering*, 124, 18-30.

Khodadoust, J., & Khodadoust, A. M. (2017). Fingerprint indexing based on expanded Delaunay triangulation. *Expert Systems with Applications*, 81, 251-267.

Rahimian, P., & Kearney, J. K. (2017). Optimal Camera Placement for Motion Capture Systems. *IEEE transactions on visualization and computer graphics*, 23(3), 1209-1221.

Vu, C. T., & Li, Y. (2009, March). Delaunay-triangulation based complete coverage in wireless sensor networks. In *Pervasive Computing*

*and Communications, 2009. PerCom 2009. IEEE International Conference on* (pp. 1-5). IEEE.

Yuan, Z., Rong, G., Guo, X., & Wang, W. (2011, June). Generalized Voronoi diagram computation on GPU. In *Voronoi Diagrams in Science and Engineering (ISVD), 2011 Eighth International Symposium on* (pp. 75-82). IEEE.

P.-N. Tan, M.Steinbach, V. Kuma «Introduction to Data Mining», Addison Wesley, 2006

Ioannis Emiris & Vissarion Fisikopoulos, Voronoi diagram and Delaunay triangulation

## **Ελληνική**

Καραγάνης Αναστάσιος & Λέκτορας Μιμής Άγγελος, Λογισμικό για τον υπολογισμό της βέλτιστης θέσης εγκατάστασης με την χρήση διαγραμμάτων Voronoi

Κωνσταντίνος Π. Τσιρογιάννης, Διάγραμμα Voronoi κύκλων και υλοποιήσεις στην CGAL

Βασιλείου Πέτρος, Τριγωνοποίηση Delaunay

Βασίλειος Τσακανίκας & Γεώργιος Τσαπακίδης, Τα διαγράμματα Voronoi. Μία εκπαιδευτική προσέγγιση για την δευτεροβάθμια εκπαίδευση

Αθανάσιος Ε. Ηλιοδρομίτης, Ανάπτυξη μεθοδολογιών χωροθέτησης & αξιολόγησης ασυρμάτων δικτύων γεωαισθητήρων με αλγορίθμους υπολογιστικής γεωμετρίας

## ΙΣΤΟΓΡΑΦΙΑ

[https://en.wikipedia.org/wiki/Windows\\_Presentation\\_Foundation](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation)

<https://en.wikipedia.org/wiki/MATLAB>

[https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

[https://en.wikipedia.org/wiki/Visual\\_Basic\\_.NET](https://en.wikipedia.org/wiki/Visual_Basic_.NET)

<https://en.wikipedia.org/wiki/C%2B%2B>

[https://msdn.microsoft.com/en-us/library/f7s023d2\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/f7s023d2(v=vs.90).aspx)

[https://msdn.microsoft.com/en-us/library/system.random\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.random(v=vs.110).aspx)

[https://msdn.microsoft.com/en-us/library/system.security.cryptography\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.security.cryptography(v=vs.110).aspx)

[https://msdn.microsoft.com/en-us/library/system.guid\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.guid(v=vs.110).aspx)

<https://docs.telerik.com/devtools/wpf/controls/radchartview/overview>

<https://docs.telerik.com/devtools/wpf/controls/radchartview/overview>

<https://github.com/DesignEngrLab/MICConvexHull>

<https://github.com/NetTopologySuite/NetTopologySuite>





## **ΠΑΡΑΡΤΗΜΑ Ι (ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ XAML)**

```

..dio 2015\Projects\TestRndWPF\TestRndWPF\Dictionary.xaml 1
1 <ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/
  presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="clr-namespace:TestRndWPF" xmlns:telerik="http://
  schemas.telerik.com/2008/xaml/presentation">
2   <LinearGradientBrush x:Key="BtnBrush1" EndPoint="0.5,1" StartPoint="0.5,0">
3     <LinearGradientBrush.GradientStops>
4       <GradientStopCollection>
5         <GradientStop Color="#B21D46FF" Offset="0" />
6         <GradientStop Color="#FFA6B6FF" Offset="0.5" />
7         <GradientStop Color="#B21D46FF" Offset="1" />
8       </GradientStopCollection>
9     </LinearGradientBrush.GradientStops>
10  </LinearGradientBrush>
11  <LinearGradientBrush x:Key="BtnBrush1_Hover" EndPoint="0.5,1"
  StartPoint="0.5,0">
12    <LinearGradientBrush.GradientStops>
13      <GradientStopCollection>
14        <GradientStop Color="#FF1D46FF" Offset="0" />
15        <GradientStop Color="#FFA6B6FF" Offset="0.5" />
16        <GradientStop Color="#FF1D46FF" Offset="1" />
17      </GradientStopCollection>
18    </LinearGradientBrush.GradientStops>
19  </LinearGradientBrush>
20  <LinearGradientBrush x:Key="BtnBrush1_Disabled" EndPoint="0.5,1"
  StartPoint="0.5,0">
21    <LinearGradientBrush.GradientStops>
22      <GradientStopCollection>
23        <GradientStop Color="#FF7780a5" Offset="0" />
24        <GradientStop Color="#FFcaccdc" Offset="0.5" />
25        <GradientStop Color="#FF7780a5" Offset="1" />
26      </GradientStopCollection>
27    </LinearGradientBrush.GradientStops>
28  </LinearGradientBrush>
29  <Style TargetType="{x:Type Button}">
30    <Setter Property="Template">
31      <Setter.Value>
32        <ControlTemplate TargetType="{x:Type Button}">
33          <Border Name="border" BorderThickness="0" Padding="4,2"
  BorderBrush="DarkGray" CornerRadius="10"
  Background="{TemplateBinding Background}">
34            <ContentPresenter HorizontalAlignment="Center"
  VerticalAlignment="Center" Name="content" />
35          </Border>
36        </ControlTemplate>
37      </Setter.Value>
38    </Setter>
39    <Setter Property="Background" Value="{StaticResource BtnBrush1}" />
40    <Setter Property="Height" Value="25" />
41    <Setter Property="BorderThickness" Value="0" />
42    <Setter Property="Foreground" Value="Yellow" />
43    <Setter Property="BorderBrush" Value="#007070" />
44    <Setter Property="FontFamily" Value="Century Gothic" />
45    <Setter Property="FontSize" Value="14" />
46    <Style.Triggers>
47      <Trigger Property="IsMouseOver" Value="True">
48        <Setter Property="Background" Value="{StaticResource
  
```

```
...dio 2015\Projects\TestRndWPF\TestRndWPF\Dictionary.xaml 2
    BtnBrush1_Hover}" />
49     </Trigger>
50     <Trigger Property="IsEnabled" Value="False">
51         <Setter Property="Background" Value="{StaticResource BtnBrush1_Disabled}" />
52     </Trigger>
53 </Style.Triggers>
54 </Style>
55 </ResourceDictionary>
56
```

```

...dio 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml 1
1 <Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" 2
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:d="http:// 2
  schemas.microsoft.com/expression/blend/2008" xmlns:mc="http:// 2
  schemas.openxmlformats.org/markup-compatibility/2006" xmlns:local="clr- 2
  namespace:TestRndWPF" xmlns:telerik="http://schemas.telerik.com/2008/xaml/ 2
  presentation" x:Class="MainWindow" mc:Ignorable="d" Title="Voronoi Grid 2
  Calculations - Press F1 for Help" Height="561" Width="1000" 2
  Icon="Voronoi.ico" MinHeight="700" MinWidth="1000"> 2
2   <Window.Resources> 2
3     <ResourceDictionary> 2
4       <ResourceDictionary.MergedDictionaries> 2
5         <ResourceDictionary Source="Dictionary.xaml" /> 2
6       </ResourceDictionary.MergedDictionaries> 2
7     </ResourceDictionary> 2
8   </Window.Resources> 2
9
10  <Grid Margin="0,0,0,0"> 2
11    <Grid.RowDefinitions> 2
12      <RowDefinition /> 2
13    </Grid.RowDefinitions> 2
14    <Grid.ColumnDefinitions> 2
15      <ColumnDefinition Width="450" MinWidth="250" /> 2
16      <ColumnDefinition Width="5" /> 2
17      <ColumnDefinition Width="250" MinWidth="250" MaxWidth="250" /> 2
18      <ColumnDefinition Width="*" /> 2
19    </Grid.ColumnDefinitions> 2
20    <Grid.Background> 2
21      <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0"> 2
22        <GradientStop Color="Black" Offset="0" /> 2
23        <GradientStop Color="#FF282850" Offset="0.5" /> 2
24        <GradientStop Color="Black" Offset="1" /> 2
25      </LinearGradientBrush> 2
26    </Grid.Background> 2
27    <GroupBox x:Name="Generator_GB" Header="" Height="83" 2
      VerticalAlignment="Top" Foreground="White" Background="#7F14143C" 2
      BorderBrush="#00D5DFE5" BorderThickness="0"> 2
28      <DockPanel x:Name="Generator_DP" LastChildFill="True" Margin="0"> 2
29        <RadioButton x:Name="MathRnd_RB" Content="Συνάρτηση" 2
          System.Math.Rnd" DockPanel.Dock="Top" 2
          VerticalAlignment="Top" Foreground="White" BorderBrush="Red" 2
          IsChecked="True" Background="#FF6464B4" FontFamily="Century 2
          Gothic" /> 2
30        <RadioButton x:Name="RandomClass_RB" Content="Κλάση" 2
          System.Random" DockPanel.Dock="Top" VerticalAlignment="Top" 2
          Margin="0,0,0,0" Foreground="White" BorderBrush="Red" 2
          Background="#FF6464B4" FontFamily="Century Gothic" /> 2
31        <RadioButton x:Name="RNG_CSP_RB" Content="Κλάση" 2
          System.Security.Cryptography" DockPanel.Dock="Top" 2
          VerticalAlignment="Top" Margin="0,0,0,0" Foreground="White" 2
          BorderBrush="Red" Background="#FF6464B4" FontFamily="Century 2
          Gothic" /> 2
32        <RadioButton x:Name="GUID_RB" Content="Κλάση System.Guid" 2
          DockPanel.Dock="Top" VerticalAlignment="Top" 2
          Margin="0,0,0,0" Foreground="White" BorderBrush="Red" 2
          Background="#FF6464B4" FontFamily="Century Gothic" 2
          IsEnabled="False" /> 2

```



```

...dio 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml 2
33     </DockPanel>
34 </GroupBox>
35 <GroupBox x:Name="SpecsForGenerator_GB" Header="" Height="76"
    VerticalAlignment="Top" Foreground="White" Background="#7F14143C"
    BorderBrush="#00D5DFE5" BorderThickness="0" Margin="0,83,0,0">
36 <Grid x:Name="SpecsForGenerator_Grid" Margin="0,0,0,0">
37 <Label x:Name="NoOfNumbers_Lbl" Content="πλήθος Αριθμών"
    HorizontalAlignment="Left" VerticalAlignment="Top"
    Width="147" Foreground="White" Height="25" Margin="0,0,0,0"
    FontFamily="Century Gothic" />
38 <TextBox x:Name="NoOfNumbers_TB" Height="25"
    TextWrapping="Wrap" VerticalAlignment="Top"
    VerticalContentAlignment="Center" BorderBrush="Red"
    BorderThickness="0,0,0,1" Background="#FF6464B4"
    Foreground="White" Margin="111,0,260,0" />
39 <Button x:Name="RunRandomize_Btn" Content="Δημιουργία Τυχαίων
    Αριθμών" Margin="0,29,0,0" />
40 <CheckBox x:Name="ShowNumberChart_ChkB" Content="Εμφάνιση
    Κατανομή Αριθμών" Height="15" Margin="0,5,0,0"
    VerticalAlignment="Top" Foreground="White"
    Background="#FF6464B4" BorderBrush="Red"
    HorizontalAlignment="Right" Width="199" FontFamily="Century
    Gothic" />
41 </Grid>
42 </GroupBox>
43 <Grid x:Name="Charts_Grid" Margin="0,164,0,10">
44 <Grid.RowDefinitions>
45 <RowDefinition Height="*" />
46 <RowDefinition Height="5" />
47 <RowDefinition Height="*" />
48 </Grid.RowDefinitions>
49 <Grid.ColumnDefinitions>
50 <ColumnDefinition />
51 </Grid.ColumnDefinitions>
52 <telerik:RadChart x:Name="Numbers_RChrt" Grid.Row="0"
    Content="RadChart" Margin="5,1" Grid.RowSpan="2"
    Background="{x:Null}" BorderBrush="#00000000"
    Foreground="{x:Null}" AxisForeground="White"
    AxisElementBrush="White">
53 <telerik:RadChart.SamplingSettings>
54 <telerik:SamplingSettings />
55 </telerik:RadChart.SamplingSettings>
56 </telerik:RadChart>
57 <GridSplitter x:Name="GeneratorCharts_Splitter" Grid.Row="1"
    Height="5" HorizontalAlignment="Stretch" Margin="5,0,5,0">
58 <GridSplitter.Background>
59 <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
60 <GradientStop Color="#FF1D46FF" Offset="0" />
61 <GradientStop Color="#FF1D46FF" Offset="1" />
62 <GradientStop Color="#FFA6B6FF" Offset="0.511" />
63 </LinearGradientBrush>
64 </GridSplitter.Background>
65 </GridSplitter>
66 <telerik:RadChart x:Name="Durations_RChrt" Grid.Row="2"
    Content="RadChart" Margin="5,0" Background="{x:Null}"
    Foreground="{x:Null}" BorderBrush="{x:Null}"

```

```

...dio 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml 3
AxisElementBrush="White" AxisForeground="White" />
67 </Grid>
68 <GridSplitter x:Name="Vertical1_Splitter" HorizontalAlignment="Left"
VerticalAlignment="Stretch" Width="5" Margin="0,0" Grid.Column="1"
Grid.ColumnSpan="1">
69 <GridSplitter.Background>
70 <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
71 <LinearGradientBrush.RelativeTransform>
72 <TransformGroup>
73 <ScaleTransform CenterY="0.5" CenterX="0.5" />
74 <SkewTransform CenterY="0.5" CenterX="0.5" />
75 <RotateTransform Angle="90" CenterY="0.5"
CenterX="0.5" />
76 <TranslateTransform />
77 </TransformGroup>
78 </LinearGradientBrush.RelativeTransform>
79 <GradientStop Color="#FF1D46FF" />
80 <GradientStop Color="#FF1D46FF" Offset="1" />
81 <GradientStop Color="#FFA6B6FF" Offset="0.5" />
82 </LinearGradientBrush>
83 </GridSplitter.Background>
84 </GridSplitter>
85 <Grid x:Name="PointsGrid" Margin="0,10" Grid.Column="2"
Grid.ColumnSpan="2">
86 <Grid.RowDefinitions>
87 <RowDefinition Height="1*" />
88 <RowDefinition Height="5" />
89 <RowDefinition Height="1*" />
90 </Grid.RowDefinitions>
91 <Grid.ColumnDefinitions>
92 <ColumnDefinition />
93 </Grid.ColumnDefinitions>
94 <TabControl x:Name="PointsSystem_TC" Margin="0,0,0,0"
VerticalAlignment="Top" Foreground="White"
Background="#7F14143C" BorderBrush="#00D5DFE5"
BorderThickness="0" HorizontalAlignment="Left">
95 <TabItem x:Name="RandomPoints_TI" Header="Τυχαία Σημεία"
Height="20" VerticalAlignment="Top" FontFamily="Century
Gothic" Background="#FF282850" Foreground="#FF91911A"
BorderBrush="#FF303A9E">
96 <Grid x:Name="RandomPointSpecs_Grid" Margin="0,0,0,0"
VerticalAlignment="Top" Width="250">
97 <Grid.ColumnDefinitions>
98 <ColumnDefinition Width="7*" />
99 <ColumnDefinition Width="3*" />
100 </Grid.ColumnDefinitions>
101 <Label x:Name="NoOfCoordinates_Lbl" Content="Πλήθος
Συντεταγμένων" HorizontalAlignment="Left"
VerticalAlignment="Top" Width="147" Foreground="White"
Margin="0,0,0,0" FontFamily="Century Gothic" />
102 <TextBox x:Name="NoOfCoordinates_TB"
HorizontalAlignment="Left" Height="25" Margin="170,0,0,0"
TextWrapping="Wrap" VerticalAlignment="Top" Width="68"
VerticalContentAlignment="Center" BorderBrush="Red"
BorderThickness="0,0,0,1" Background="#FF6464B4"
Foreground="White" Grid.ColumnSpan="2" />

```

```

...dio 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml 4
103 <TextBox x:Name="Xmin_TB" HorizontalAlignment="Left" 7
      Height="25" Margin="0,50,0,0" TextWrapping="Wrap" 7
      VerticalAlignment="Top" Width="50" 7
      VerticalContentAlignment="Center" BorderBrush="Red" 7
      BorderThickness="0,0,0,1" Background="#FF6464B4" 7
      Foreground="White" ToolTip="min" />
104 <Label x:Name="X_Lbl" Content="X" 7
      HorizontalAlignment="Left" Margin="56,50,0,0" 7
      VerticalAlignment="Top" Width="127" Foreground="White" 7
      Height="25" HorizontalContentAlignment="Center" 7
      FontFamily="Century Gothic" Grid.ColumnSpan="2" />
105 <TextBox x:Name="Xmax_TB" HorizontalAlignment="Left" 7
      Height="25" Margin="13,50,0,0" TextWrapping="Wrap" 7
      VerticalAlignment="Top" Width="50" 7
      VerticalContentAlignment="Center" BorderBrush="Red" 7
      BorderThickness="0,0,0,1" Background="#FF6464B4" 7
      Foreground="White" Grid.Column="1" />
106 <TextBox x:Name="Ymin_TB" HorizontalAlignment="Left" 7
      Height="25" Margin="0,78,0,0" TextWrapping="Wrap" 7
      VerticalAlignment="Top" Width="50" 7
      VerticalContentAlignment="Center" BorderBrush="Red" 7
      BorderThickness="0,0,0,1" Background="#FF6464B4" 7
      Foreground="White" />
107 <Label x:Name="Y_Lbl" Content="Y" 7
      HorizontalAlignment="Left" Margin="56,78,0,0" 7
      VerticalAlignment="Top" Width="127" Foreground="White" 7
      Height="25" HorizontalContentAlignment="Center" 7
      FontFamily="Century Gothic" Grid.ColumnSpan="2" />
108 <TextBox x:Name="Ymax_TB" HorizontalAlignment="Left" 7
      Height="25" Margin="13,78,0,0" TextWrapping="Wrap" 7
      VerticalAlignment="Top" Width="50" 7
      VerticalContentAlignment="Center" BorderBrush="Red" 7
      BorderThickness="0,0,0,1" Background="#FF6464B4" 7
      Foreground="White" Grid.Column="1" />
109 <TextBox x:Name="DispersionMin_TB" 7
      HorizontalAlignment="Left" Height="25" Margin="0,105,0,0" 7
      TextWrapping="Wrap" VerticalAlignment="Top" Width="50" 7
      VerticalContentAlignment="Center" BorderBrush="Red" 7
      BorderThickness="0,0,0,1" Background="#FF6464B4" 7
      Foreground="White" />
110 <CheckBox x:Name="DispersionMin_ChkB" 7
      Content="CheckBox" HorizontalAlignment="Left" 7
      Margin="53,111,0,0" VerticalAlignment="Top" Width="17" 7
      BorderBrush="Red" Background="#FF6464C8" />
111 <Label x:Name="Dispersion_Lbl" Content="Διασπορά" 7
      HorizontalAlignment="Left" Margin="69,105,0,0" 7
      VerticalAlignment="Top" Width="97" Foreground="White" 7
      Height="25" HorizontalContentAlignment="Center" 7
      FontFamily="Century Gothic" />
112 <CheckBox x:Name="DispersionMax_ChkB" 7
      Content="CheckBox" HorizontalAlignment="Left" 7
      Margin="170,111,0,0" VerticalAlignment="Top" Width="17" 7
      BorderBrush="Red" Background="#FF6464C8" 7
      Grid.ColumnSpan="2" />
113 <TextBox x:Name="DispersionMax_TB" 7
      HorizontalAlignment="Left" Height="25" Margin="13,105,0,0" 7

```



```

...dio 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml 5
    TextWrapping="Wrap" VerticalAlignment="Top" Width="50"
    VerticalContentAlignment="Center" BorderBrush="Red"
    BorderThickness="0,0,0,1" Background="#FF646484"
    Foreground="White" Grid.Column="1" />
114 <CheckBox x:Name="InterruptTimer_ChkB"
    Content="Χρόνικος Διακόπτης (sec)"
    HorizontalAlignment="Left" Width="183" Foreground="White"
    Margin="0,152,0,0" FontFamily="Century Gothic"
    Background="#FF6464C8" BorderBrush="Red" IsChecked="True"
    Height="20" VerticalAlignment="Top" Grid.ColumnSpan="2" />
115 <TextBox x:Name="InterruptTimer_TB"
    HorizontalAlignment="Left" Height="25" Margin="13,146,0,0"
    TextWrapping="Wrap" VerticalAlignment="Top" Width="50"
    VerticalContentAlignment="Center" BorderBrush="Red"
    BorderThickness="0,0,0,1" Background="#FF646484"
    Foreground="White" Grid.Column="1" />
116 <Label x:Name="X_Min" Content="Min"
    HorizontalAlignment="Left" Margin="0,28,0,0"
    VerticalAlignment="Top" Width="50" Foreground="White"
    Height="22" HorizontalContentAlignment="Center"
    FontFamily="Century Gothic" />
117 <Label x:Name="X_Max" Content="Max"
    HorizontalAlignment="Left" Margin="13,28,0,0"
    VerticalAlignment="Top" Width="50" Foreground="White"
    Height="22" HorizontalContentAlignment="Center"
    FontFamily="Century Gothic" Grid.Column="1" />
118 <Label x:Name="Info1_Lbl" Content="Παραχθείσες
    Συντεταγμένες" Grid.Column="0" Grid.Row="0"
    Margin="0,177,7,89" Foreground="White"
    HorizontalContentAlignment="Left" FontFamily="Century
    Gothic" />
119 <Label x:Name="Info2_Lbl" Content="Μέγιστη Απόσταση"
    Grid.Column="0" Margin="0,202,7,63" Foreground="White"
    HorizontalContentAlignment="Left" FontFamily="Century
    Gothic" />
120 <Label x:Name="Info3_Lbl" Content="Ασφαλής Διασπορά
    &lt;" Grid.Column="0" Margin="0,228,7,38"
    Foreground="White" HorizontalContentAlignment="Left"
    FontFamily="Century Gothic" Height="25" Width="167" />
121 <Label x:Name="Info1_Value_Lbl" Content=""
    Grid.Row="0" Margin="1,177,10,89" Foreground="White"
    HorizontalContentAlignment="Left" FontFamily="Century
    Gothic" Grid.Column="1" />
122 <Label x:Name="Info2_Value_Lbl" Content=""
    Margin="1,203,10,63" Foreground="White"
    HorizontalContentAlignment="Left" FontFamily="Century
    Gothic" Grid.Column="1" />
123 <Label x:Name="Info3_Value_Lbl" Content=""
    Margin="1,228,10,38" Foreground="White"
    HorizontalContentAlignment="Left" FontFamily="Century
    Gothic" Grid.Column="1" />
124 <Button x:Name="GeneratePoints_Btn"
    Content="Δημιουργία Συντεταγμένων" Margin="0,258,0,8"
    Grid.ColumnSpan="2" />
125 </Grid>
126 </TabItem>

```

```

...dio 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml 6
127 <TabItem Header="Πλέγμα" FontFamily="Century Gothic" 7
    Background="#FF282850" Foreground="#FF9191A" 7
    BorderBrush="#00000000">
128 <Grid x:Name="GridPointSpecs_Grid" Margin="0,0,0,0" 7
    VerticalAlignment="Top" Width="250">
129 <Grid.ColumnDefinitions>
130 <ColumnDefinition Width="0*" />
131 <ColumnDefinition />
132 </Grid.ColumnDefinitions>
133 <Label x:Name="PointsDistance_Lbl" Content="Απόσταση 7
    Σημείων" HorizontalAlignment="Left" 7
    VerticalAlignment="Top" Width="147" Foreground="White" 7
    Margin="0,0,0,0" FontFamily="Century Gothic" 7
    Grid.ColumnSpan="2" />
134 <TextBox x:Name="PointsDistance_TB" 7
    HorizontalAlignment="Left" Height="25" Margin="170,0,0,0" 7
    TextWrapping="Wrap" VerticalAlignment="Top" Width="68" 7
    VerticalContentAlignment="Center" BorderBrush="Red" 7
    BorderThickness="0,0,0,1" Background="#FF646484" 7
    Foreground="White" Grid.ColumnSpan="2" />
135 <Label x:Name="G_Min" Content="Min" 7
    HorizontalAlignment="Left" Margin="0,28,0,0" 7
    VerticalAlignment="Top" Width="50" Foreground="White" 7
    Height="22" HorizontalContentAlignment="Center" 7
    FontFamily="Century Gothic" Grid.ColumnSpan="2" />
136 <Label x:Name="G_Max" Content="Max" 7
    HorizontalAlignment="Left" Margin="188,28,0,0" 7
    VerticalAlignment="Top" Width="50" Foreground="White" 7
    Height="22" HorizontalContentAlignment="Center" 7
    FontFamily="Century Gothic" Grid.ColumnSpan="2" />
137 <Label x:Name="GX_Lbl" Content="X" 7
    HorizontalAlignment="Left" Margin="56,50,0,0" 7
    VerticalAlignment="Top" Width="127" Foreground="White" 7
    Height="25" HorizontalContentAlignment="Center" 7
    FontFamily="Century Gothic" Grid.ColumnSpan="2" />
138 <Label x:Name="GY_Lbl" Content="Y" 7
    HorizontalAlignment="Left" Margin="56,78,0,0" 7
    VerticalAlignment="Top" Width="127" Foreground="White" 7
    Height="25" HorizontalContentAlignment="Center" 7
    FontFamily="Century Gothic" Grid.ColumnSpan="2" />
139 <TextBox x:Name="GXmin_TB" HorizontalAlignment="Left" 7
    Height="25" Margin="0,50,0,0" TextWrapping="Wrap" 7
    VerticalAlignment="Top" Width="50" 7
    VerticalContentAlignment="Center" BorderBrush="Red" 7
    BorderThickness="0,0,0,1" Background="#FF646484" 7
    Foreground="White" ToolTip="min" Grid.ColumnSpan="2" />
140 <TextBox x:Name="GXmax_TB" HorizontalAlignment="Left" 7
    Height="25" Margin="188,50,0,0" TextWrapping="Wrap" 7
    VerticalAlignment="Top" Width="50" 7
    VerticalContentAlignment="Center" BorderBrush="Red" 7
    BorderThickness="0,0,0,1" Background="#FF646484" 7
    Foreground="White" Grid.ColumnSpan="2" />
141 <TextBox x:Name="GYmin_TB" HorizontalAlignment="Left" 7
    Height="25" Margin="0,78,0,0" TextWrapping="Wrap" 7
    VerticalAlignment="Top" Width="50" 7
    VerticalContentAlignment="Center" BorderBrush="Red" 7

```

```

...dio 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml 7
    BorderThickness="0,0,0,1" Background="#FF6464B4"
    Foreground="white" Grid.ColumnSpan="2" />
142     <TextBox x:Name="GYmax_TB" HorizontalAlignment="Left"
    Height="25" Margin="188,78,0,0" TextWrapping="Wrap"
    VerticalAlignment="Top" Width="50"
    VerticalContentAlignment="Center" BorderBrush="Red"
    BorderThickness="0,0,0,1" Background="#FF6464B4"
    Foreground="white" Grid.ColumnSpan="2" />
143     <Button x:Name="GenerateGrid_Btn" Content="Δημιουργία
    Συντεταγμένων" Margin="0,258,0,0" IsEnabled="False"
    Grid.ColumnSpan="2" />
144 </Grid>
145 </TabItem>
146 <TabItem Header="Εισαγωγή" FontFamily="Century Gothic"
    Background="#FF282850" Foreground="#FF91911A"
    BorderBrush="#00000000">
147     <Grid x:Name="ImportPointSpecs_Grid" Margin="0,0,0,0"
    VerticalAlignment="Top" Width="250">
148         <Grid.RowDefinitions>
149             <RowDefinition Height="6*" />
150             <RowDefinition Height="1*" />
151             <RowDefinition Height="1*" />
152             <RowDefinition Height="1*" />
153             <RowDefinition Height="1*" />
154             <RowDefinition Height="1*" />
155         </Grid.RowDefinitions>
156         <Grid.ColumnDefinitions>
157             <ColumnDefinition Width="1*" />
158             <ColumnDefinition Width="2*" />
159         </Grid.ColumnDefinitions>
160         <TextBlock x:Name="Info_TB1k" Grid.Column="0"
    Grid.Row="0" TextWrapping="Wrap" Margin="0,0,0,0"
    VerticalAlignment="Top" Grid.ColumnSpan="2"
    IsEnabled="False" Foreground="Yellow" FontSize="13"><Run
    Text="Για τη σωστή εισαγωγή των συντεταγμένων θα
    πρέπει:" /><LineBreak /><LineBreak /><Run Text="*Το αρχείο
    να είναι της μορφής CSV" /><LineBreak /><Run Text="*Η
    υποδιαστολή να είναι με τελεία (.)" /><LineBreak /><Run
    Text="*Η τεταγμένη με την τετμημένη να διαχωρίζονται με
    ερωτηματικό (;)" /><LineBreak /><Run Text="*Να μην
    υπάρχουν κενές γραμμές " /><LineBreak /><Run /></
    TextBlock>
161         <Label x:Name="Xmin_Lbl" Grid.Column="0" Grid.Row="1"
    Content="X min" Margin="0,0,0,0" VerticalAlignment="Top"
    Foreground="white" HorizontalContentAlignment="Left"
    FontFamily="Century Gothic" />
162         <Label x:Name="Xmax_Lbl" Grid.Column="0" Grid.Row="2"
    Content="X max" Margin="0,0,0,0" VerticalAlignment="Top"
    Foreground="white" HorizontalContentAlignment="Left"
    FontFamily="Century Gothic" />
163         <Label x:Name="Ymin_Lbl" Grid.Column="0" Grid.Row="3"
    Content="Y min" Margin="0,0,0,0" VerticalAlignment="Top"
    Foreground="white" HorizontalContentAlignment="Left"
    FontFamily="Century Gothic" />
164         <Label x:Name="Ymax_Lbl" Grid.Column="0" Grid.Row="4"

```



```

...dio 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml 8
Content="Y max" Margin="0,0,0,0" VerticalAlignment="Top"
Foreground="White" HorizontalContentAlignment="Left"
FontFamily="Century Gothic" />
165 <Label x:Name="Xmin_Value_Lbl" Grid.Column="1"
Grid.Row="1" Content="" Margin="0,0,0,0"
VerticalAlignment="Top" Foreground="White"
HorizontalContentAlignment="Left" FontFamily="Century
Gothic" />
166 <Label x:Name="Xmax_Value_Lbl" Grid.Column="1"
Grid.Row="2" Content="" Margin="0,0,0,0"
VerticalAlignment="Top" Foreground="White"
HorizontalContentAlignment="Left" FontFamily="Century
Gothic" />
167 <Label x:Name="Ymin_Value_Lbl" Grid.Column="1"
Grid.Row="3" Content="" Margin="0,0,0,0"
VerticalAlignment="Top" Foreground="White"
HorizontalContentAlignment="Left" FontFamily="Century
Gothic" />
168 <Label x:Name="Ymax_Value_Lbl" Grid.Column="1"
Grid.Row="4" Content="" Margin="0,0,0,0"
VerticalAlignment="Top" Foreground="White"
HorizontalContentAlignment="Left" FontFamily="Century
Gothic" />
169 <Button x:Name="ImportPoints_Btn" Grid.Column="0"
Grid.Row="5" Content="Εισαγωγή Συντεταγμένων"
Margin="0,0,0,0" VerticalAlignment="Top"
Grid.ColumnSpan="2" />
170 </Grid>
171 </TabItem>
172 </TabControl>
173
174 <telerik:RadCartesianChart x:Name="Points_RCChart"
Margin="258,10,5,5" Background="#36FFFFFF">
175 <telerik:RadCartesianChart.HorizontalAxis>
176 <!--<telerik:LogarithmicAxis LogarithmBase="10" />-->
177 <telerik:LinearAxis LineStroke="Yellow" LineThickness="1"
ElementBrush="Yellow" />
178 </telerik:RadCartesianChart.HorizontalAxis>
179 <telerik:RadCartesianChart.VerticalAxis>
180 <telerik:LinearAxis LineStroke="Yellow" LineThickness="1"
ElementBrush="Yellow" />
181 </telerik:RadCartesianChart.VerticalAxis>
182 <telerik:ScatterPointSeries x:Name="Points_Series2">
183 <telerik:ScatterPointSeries.DefaultVisualStyle>
184 <Style TargetType="{x:Type Path}">
185 <Setter Property="Stroke" Value="Green" />
186 <Setter Property="Fill" Value="Orange" />
187 <Setter Property="StrokeThickness" Value="1" />
188 <Setter Property="Height" Value="4" />
189 <Setter Property="Width" Value="4" />
190 </Style>
191 </telerik:ScatterPointSeries.DefaultVisualStyle>
192 </telerik:ScatterPointSeries>
193 </telerik:RadCartesianChart>
194 <GridSplitter x:Name="PointsGrid_Splitter" Grid.Row="1" Height="5"
HorizontalAlignment="Stretch" Margin="5,0,5,0">

```

```

...dio 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml 9
195         <GridSplitter.Background>
196             <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
197                 <GradientStop Color="#FF1D46FF" Offset="0" />
198                 <GradientStop Color="#FF1D46FF" Offset="1" />
199                 <GradientStop Color="#FFA6B6FF" Offset="0.511" />
200             </LinearGradientBrush>
201         </GridSplitter.Background>
202     </GridSplitter>
203     <GroupBox x:Name="DelVorSpecs_GB" Header="" Height="65"
204         VerticalAlignment="Top" Foreground="White"
205         Background="#7F14143C" BorderBrush="#00D5DFE5"
206         BorderThickness="0" HorizontalAlignment="Left" Width="253"
207         Grid.Row="2">
208         <Grid x:Name="DelVorSpecs_Grid" Margin="0,0,0,0">
209             <Label x:Name="NoOfChosenCoordinates_Lbl" Content="πλήθος
210                 Κέντρων" HorizontalAlignment="Left"
211                 VerticalAlignment="Top" Width="183" Foreground="White"
212                 Height="25" Margin="0,-4,0,0" FontFamily="Century
213                 Gothic" />
214             <Label x:Name="X_Lbl1" Content="X"
215                 HorizontalAlignment="Left" Margin="56,28,0,0"
216                 VerticalAlignment="Top" Width="127" Foreground="White"
217                 Height="25" HorizontalContentAlignment="Center"
218                 FontFamily="Century Gothic" Visibility="Hidden" />
219             <Label x:Name="Y_Lbl1" Content="Y"
220                 HorizontalAlignment="Left" Margin="56,56,0,0"
221                 VerticalAlignment="Top" Width="127" Foreground="White"
222                 Height="25" HorizontalContentAlignment="Center"
223                 FontFamily="Century Gothic" Visibility="Hidden" />
224             <Label x:Name="Dispersion_Lbl1" Content="Διασπορά"
225                 HorizontalAlignment="Left" Margin="69,84,0,0"
226                 VerticalAlignment="Top" Width="97" Foreground="White"
227                 Height="25" HorizontalContentAlignment="Center"
228                 FontFamily="Century Gothic" Visibility="Hidden" />
229             <TextBox x:Name="NoOfBaseCoordinates_TB"
230                 HorizontalAlignment="Left" Height="25" Margin="188,-4,0,0"
231                 TextWrapping="Wrap" VerticalAlignment="Top" Width="50"
232                 VerticalContentAlignment="Center" BorderBrush="Red"
233                 BorderThickness="0,0,0,1" Background="#FF6464B4"
234                 Foreground="White" />
235             <CheckBox x:Name="ActualPoints_ChkB" Content="Υπαρκτές
236                 Συντεταγμένες" HorizontalAlignment="Left" Width="238"
237                 Foreground="White" Margin="0,25,0,-2" FontFamily="Century
238                 Gothic" Background="#FF6464C8" BorderBrush="Red"
239                 IsChecked="True" Height="20" VerticalAlignment="Top" />
240         </Grid>
241     </GroupBox>
242     <GroupBox x:Name="DelVorInfo_GB" Foreground="White"
243         Background="#7F00358D" BorderBrush="#00D5DFE5"
244         BorderThickness="0" Margin="0,65,0,0" HorizontalAlignment="Left"
245         Width="253" Grid.Row="2">
246         <Grid x:Name="DelVorInfo_Grid" Margin="0,0,0,0">
247             <Grid.RowDefinitions>
248                 <RowDefinition Height="Auto" MinHeight="25" />
249                 <RowDefinition Height="Auto" MinHeight="25" />
250                 <RowDefinition Height="Auto" MinHeight="25" />

```

```

...dio 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml 10
219     <RowDefinition Height="Auto" MinHeight="25" />
220     <RowDefinition Height="Auto" MinHeight="25" />
221     <RowDefinition Height="Auto" MinHeight="25" />
222     <RowDefinition Height="Auto" MinHeight="25" />
223     <RowDefinition Height="Auto" MinHeight="25" />
224     <RowDefinition Height="Auto" MinHeight="25" />
225     </Grid.RowDefinitions>
226     <Grid.ColumnDefinitions>
227         <ColumnDefinition Width="175" />
228         <ColumnDefinition />
229     </Grid.ColumnDefinitions>
230     <Label x:Name="Info4_Lbl" Content="Τρίγωνα Delaunay"
231         Grid.Column="0" Grid.Row="0" Margin="4,0"
232         Foreground="White" HorizontalContentAlignment="Left"
233         FontFamily="Century Gothic" />
234     <Label x:Name="Info5_Lbl" Content="Εμβαδόν Τριγώνων"
235         Grid.Column="0" Grid.Row="1" Margin="4,0"
236         Foreground="White" HorizontalContentAlignment="Left"
237         FontFamily="Century Gothic" />
238     <Label x:Name="Info6_Lbl" Content="Πολύγωνα Voronoi"
239         Grid.Column="0" Grid.Row="2" Margin="4,0"
240         Foreground="White" HorizontalContentAlignment="Left"
241         FontFamily="Century Gothic" />
242     <Label x:Name="Info7_Lbl" Content="Νέα Βάση"
243         Grid.Column="0" Grid.Row="3" Margin="4,0"
244         Foreground="White" HorizontalContentAlignment="Left"
245         FontFamily="Century Gothic" />
246     <Label x:Name="Info4_Value_Lbl" Content="" Grid.Column="1"
247         Grid.Row="0" Margin="4,0" Foreground="White"
248         HorizontalContentAlignment="Left" FontFamily="Century
249         Gothic" />
250     <Label x:Name="Info5_Value_Lbl" Content="" Grid.Column="1"
251         Grid.Row="1" Margin="4,0,4,1" Foreground="White"
252         HorizontalContentAlignment="Left" FontFamily="Century
253         Gothic" />
254     <Label x:Name="Info6_Value_Lbl" Content="" Grid.Column="1"
255         Grid.Row="2" Margin="4,0" Foreground="White"
256         HorizontalContentAlignment="Left" FontFamily="Century
257         Gothic" />
258     <Label x:Name="Info7_Value_Lbl" Content="" Grid.Column="1"
259         Grid.Row="3" Margin="4,0" Foreground="White"
260         HorizontalContentAlignment="Left" FontFamily="Century
261         Gothic" />
262     <Button x:Name="DrawPoints_Btn" Content="Επιλογή Τυχαίων
263         Κέντρων" Margin="0,0,0,5" Grid.ColumnSpan="2" Grid.Row="4"
264         IsEnabled="False" />
265     <Button x:Name="Delaunay_Btn" Content="Delaunay"
266         Margin="0,0,0,5" Grid.ColumnSpan="2" Grid.Row="5"
267         IsEnabled="False" />
268     <Button x:Name="Voronoi_Btn" Content="Voronoi"
269         Margin="0,0,0,5" Grid.ColumnSpan="2" Grid.Row="6"
270         IsEnabled="False" />
271     <Button x:Name="NewPoints_Btn" Content="Νέα Κέντρα"
272         Margin="0,0,0,5" Grid.ColumnSpan="2" Grid.Row="7"
273         IsEnabled="False" />
274     <Button x:Name="Output_Btn" Content="Εξαγωγή Κέντρων"

```

```
...dio 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml 11
    Margin="0,0,0,5" Grid.ColumnSpan="2" Grid.Row="8"
    IsEnabled="False" />
243     </Grid>
244     </GroupBox>
245     <Canvas Margin="258,10,5,0" ClipToBounds="True"
    x:Name="drawingCanvas" HorizontalAlignment="Stretch"
    VerticalAlignment="Stretch" Background="White" Grid.Row="2" />
246     </Grid>
247     </Grid>
248 </Window>
249
```

## **ΠΑΡΑΡΤΗΜΑ ΙΙ (ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ VB.NET)**



```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 1
1 Imports MICConvHull
2 Module GlobalVCFS
3     'Global Variables - Classes - Functions - Subs
4     Public Xmin As Double = 0
5     Public Xmax As Double = 0
6     Public Ymin As Double = 0
7     Public Ymax As Double = 0
8     Public NoOfCoordinates As Integer
9     Public NoOfBaseCoordinates As Integer
10    Public PointsCounter As Integer
11    Public InterruptTimer As Integer
12    Public DispersionMinChecked As Boolean
13    Public DispersionMaxChecked As Boolean
14    Public InterruptTimerChecked As Boolean
15    Public DispersionMin As Double
16    Public DispersionMax As Double
17    Public RandomClass As New Random()
18    Public RNG_CSP As New Security.Cryptography.RNGCryptoServiceProvider()
19    Public RNG_CSP_ByteArray = New Byte(3) {}
20    Public CurrentRandomSystem As RandomSystem
21    Public myVertices = New List(Of Vertex)()
22    Public myBaseVertices = New List(Of Vertex)()
23    Public Tx As Double = 0
24    Public Ty As Double = 0
25    Public DelauneyArea As Double = 0
26    Public IntersectBorderPoints As New List(Of Point)
27    Public TopLeftCanvasBorder As Point
28    Public TopRightCanvasBorder As Point
29    Public BottomRightCanvasBorder As Point
30    Public BottomLeftCanvasBorder As Point
31    Public Enum RandomSystem
32        VBMathRnd = 1
33        RandomCls = 2
34        CryptographyClass = 3
35        GUID = 4
36    End Enum
37    Public Structure CropLineInBorders
38        Dim NewPoints() As Point
39        Dim Include As Boolean
40        Dim BorderPointsCount As Integer
41        Dim BorderPoints As List(Of Point)
42        Dim PointInCanvas() As Boolean
43        Dim IntersectWithTopBorder As Boolean
44        Dim IntersectWithRightBorder As Boolean
45        Dim IntersectWithBottomBorder As Boolean
46        Dim IntersectWithLeftBorder As Boolean
47    End Structure
48    Public Structure IntersectWithRectangle
49        Dim IntersectionPoint
50        Dim IntersectionTopBorder As Boolean
51        Dim IntersectionRightBorder As Boolean
52        Dim IntersectionBottomBorder As Boolean
53        Dim IntersectionLeftBorder As Boolean
54    End Structure
55    Public Function myRnd(myRandomSystem As RandomSystem) As Decimal
56        Select Case myRandomSystem

```

```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 2
57     Case RandomSystem.VBMathRnd
58         Return VBMath.Rnd()
59     Case RandomSystem.RandomCls
60         Return RandomClass.NextDouble
61     Case RandomSystem.CryptographyClass
62         RNG_CSP.GetBytes(RNG_CSP_ByteArray)
63         Return BitConverter.ToInt32(RNG_CSP_ByteArray, 0) /  ⤴
           UInt32.MaxValue
64     Case RandomSystem.GUID
65         Return "0," & Guid.NewGuid().ToByteArray().ToString("D39")
66     Case Else
67         Return 0
68     End Select
69 End Function
70 Public Function CalcDistance(X_init As Double, Y_Init As Double, X_new As  ⤴
           Double, Y_new As Double) As Double
71     Return ((X_init - X_new) ^ 2 + (Y_Init - Y_new) ^ 2) ^ 0.5
72 End Function
73 Public Function PointInPolygon(Points() As Point, X As Double, Y As  ⤴
           Double) As Boolean
74     ' Get the angle between the point and the first and last vertices.
75     Dim max_point As Integer = Points.Length - 1
76     'Dim total_angle As Double = GetAngle(Points(max_point).X, Points  ⤴
           (max_point).Y, X, Y, Points(0).X, Points(0).Y)
77     '' Add the angles from the point to each other pair of vertices.
78     'For i As Integer = 0 To max_point - 1
79     '    total_angle += GetAngle(Points(i).X, Points(i).Y, X, Y, Points(i  ⤴
           + 1).X, Points(i + 1).Y)
80     'Next i
81     Dim total_angle As Double = Math.Abs(GetAngle(Points(max_point).X,  ⤴
           Points(max_point).Y, X, Y, Points(0).X, Points(0).Y))
82     ' Add the angles from the point to each other pair of vertices.
83     For i As Integer = 0 To max_point - 1
84         total_angle += Math.Abs(GetAngle(Points(i).X, Points(i).Y, X, Y,  ⤴
           Points(i + 1).X, Points(i + 1).Y))
85     Next i
86     ' The total angle should be 2 * PI or -2 * PI if the point is in the  ⤴
           polygon and close to zero if the point is outside the polygon.
87     Dim TempRes As Double = Math.Abs((2 * Math.PI) - total_angle)
88     Return TempRes < 0.000000000000002
89 End Function
90 Public Function GetAngle(ByVal Ax As Double, ByVal Ay As Double, ByVal Bx  ⤴
           As Double, ByVal By As Double, ByVal Cx As Double, ByVal Cy As Double)  ⤴
           As Double
91     Dim dot_product As Double
92     Dim cross_product As Double
93     ' Get the dot product and cross product.
94     dot_product = DotProduct(Ax, Ay, Bx, By, Cx, Cy)
95     cross_product = CrossProductLength(Ax, Ay, Bx, By, Cx, Cy)
96     ' Calculate the angle.
97     Return Math.Atan2(cross_product, dot_product)
98 End Function
99 Public Function CrossProductLength(ByVal Ax As Double, ByVal Ay As Double,  ⤴
           ByVal Bx As Double, ByVal By As Double, ByVal Cx As Double, ByVal Cy As  ⤴
           Double) As Double
100     ' Return the cross product AB x BC.

```

```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 3
101 ' The cross product is a vector perpendicular to AB and BC having  ↗
    length |AB| * |BC| * Sin(theta) and with direction given by the  ↗
    right-hand rule.
102 ' For two vectors in the X-Y plane, the result is a vector with X and ↗
    Y components 0 so the Z component gives the vector's length and  ↗
    direction.
103 ' Get the vectors' coordinates.
104 Dim BAX As Double = Ax - Bx
105 Dim BAY As Double = Ay - By
106 Dim BCx As Double = Cx - Bx
107 Dim BCy As Double = Cy - By
108 ' Calculate the Z coordinate of the cross product.
109 Return BAX * BCy - BAY * BCx
110 End Function
111 Private Function DotProduct(ByVal Ax As Double, ByVal Ay As Double, ByVal ↗
    Bx As Double, ByVal By As Double, ByVal Cx As Double, ByVal Cy As ↗
    Double) As Double
112 ' Return the dot product AB · BC.
113 ' Note that AB · BC = |AB| * |BC| * Cos(theta).
114 ' Get the vectors' coordinates.
115 Dim BAX As Double = Ax - Bx
116 Dim BAY As Double = Ay - By
117 Dim BCx As Double = Cx - Bx
118 Dim BCy As Double = Cy - By
119 ' Calculate the dot product.
120 Return BAX * BCx + BAY * BCy
121 End Function
122 Public Function Intersect(ByVal L1_Start As Point, ByVal L1_End As Point, ↗
    ByVal L2_Start As Point, ByVal L2_End As Point, Optional Segment As ↗
    Boolean = True)
123 If L1_End.X <> L1_Start.X And L2_End.X <> L2_Start.X Then
124 Dim A1 As Double = (L1_End.Y - L1_Start.Y) / (L1_End.X - ↗
    L1_Start.X)
125 Dim A2 As Double = (L2_End.Y - L2_Start.Y) / (L2_End.X - ↗
    L2_Start.X)
126 Dim B1 As Double = (L1_End.X * L1_Start.Y - L1_Start.X * ↗
    L1_End.Y) / (L1_End.X - L1_Start.X)
127 Dim B2 As Double = (L2_End.X * L2_Start.Y - L2_Start.X * ↗
    L2_End.Y) / (L2_End.X - L2_Start.X)
128 If Not A1 = A2 Then
129 If A1 = 0 Then
130 Dim NewX As Double = (B1 - B2) / (A2 - A1)
131 Dim NewY As Double = B1
132 Dim NewPoint As Point = New Point(NewX, NewY)
133 If Segment = True Then
134 If NewPoint.X > Math.Max(L1_Start.X, L1_End.X) Or ↗
    NewPoint.X < Math.Min(L1_Start.X, L1_End.X) Or NewPoint.X ↗
    > Math.Max(L2_Start.X, L2_End.X) Or NewPoint.X < Math.Min ↗
    (L2_Start.X, L2_End.X) Or NewPoint.Y > Math.Max ↗
    (L1_Start.Y, L1_End.Y) Or NewPoint.Y < Math.Min ↗
    (L1_Start.Y, L1_End.Y) Or NewPoint.Y > Math.Max ↗
    (L2_Start.Y, L2_End.Y) Or NewPoint.Y < Math.Min ↗
    (L2_Start.Y, L2_End.Y) Then Return Nothing
135 Return NewPoint
136 Else
137 Return NewPoint

```

```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 4
138     End If
139     ElseIf A2 = 0 Then
140         Dim NewX As Double = (B1 - B2) / (A2 - A1)
141         Dim NewY As Double = B2
142         Dim NewPoint As Point = New Point(NewX, NewY)
143         If Segment = True Then
144             If NewPoint.X > Math.Max(L1_Start.X, L1_End.X) Or
NewPoint.X < Math.Min(L1_Start.X, L1_End.X) Or NewPoint.X
> Math.Max(L2_Start.X, L2_End.X) Or NewPoint.X < Math.Min
(L2_Start.X, L2_End.X) Or NewPoint.Y > Math.Max
(L1_Start.Y, L1_End.Y) Or NewPoint.Y < Math.Min
(L1_Start.Y, L1_End.Y) Or NewPoint.Y > Math.Max
(L2_Start.Y, L2_End.Y) Or NewPoint.Y < Math.Min
(L2_Start.Y, L2_End.Y) Then Return Nothing
145             Return NewPoint
146         Else
147             Return NewPoint
148         End If
149     Else
150         Dim NewX As Double = (B1 - B2) / (A2 - A1)
151         Dim NewY As Double = (A2 * B1 - A1 * B2) / (A2 - A1)
152         Dim NewPoint As Point = New Point(NewX, NewY)
153         If Segment = True Then
154             If NewPoint.X > Math.Max(L1_Start.X, L1_End.X) Or
NewPoint.X < Math.Min(L1_Start.X, L1_End.X) Or NewPoint.X
> Math.Max(L2_Start.X, L2_End.X) Or NewPoint.X < Math.Min
(L2_Start.X, L2_End.X) Or NewPoint.Y > Math.Max
(L1_Start.Y, L1_End.Y) Or NewPoint.Y < Math.Min
(L1_Start.Y, L1_End.Y) Or NewPoint.Y > Math.Max
(L2_Start.Y, L2_End.Y) Or NewPoint.Y < Math.Min
(L2_Start.Y, L2_End.Y) Then Return Nothing
155             Return NewPoint
156         Else
157             Return NewPoint
158         End If
159     End If
160 Else
161     Return Nothing
162 End If
163 ElseIf L1_End.X = L1_Start.X Then
164     Dim A2 As Double = (L2_End.Y - L2_Start.Y) / (L2_End.X -
L2_Start.X)
165     Dim B2 As Double = (L2_End.X * L2_Start.Y - L2_Start.X *
L2_End.Y) / (L2_End.X - L2_Start.X)
166     Dim NewX As Double = L1_Start.X
167     Dim NewY As Double = A2 * NewX + B2
168     'Dim NewY As Double = (L2_End.Y - L2_Start.Y) / (L2_End.X -
L2_Start.X) * NewX + (L2_End.X * L2_Start.Y - L2_Start.X *
L2_End.Y) / (L2_End.X - L2_Start.X)
169     Dim NewPoint As Point = New Point(NewX, NewY)
170     If Segment = True Then
171         If NewPoint.X > Math.Max(L1_Start.X, L1_End.X) Or NewPoint.X <
Math.Min(L1_Start.X, L1_End.X) Or NewPoint.X > Math.Max
(L2_Start.X, L2_End.X) Or NewPoint.X < Math.Min(L2_Start.X,
L2_End.X) Or NewPoint.Y > Math.Max(L1_Start.Y, L1_End.Y) Or
NewPoint.Y < Math.Min(L1_Start.Y, L1_End.Y) Or NewPoint.Y >

```

```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCF5.vb 5
    Math.Max(L2_Start.Y, L2_End.Y) Or NewPoint.Y < Math.Min
    (L2_Start.Y, L2_End.Y) Then Return Nothing
172     Return NewPoint
173 Else
174     Return NewPoint
175 End If
176 ElseIf L2_End.X = L2_Start.X Then
177     Dim A1 As Double = (L1_End.Y - L1_Start.Y) / (L1_End.X -
    L1_Start.X)
178     Dim B1 As Double = (L1_End.X * L1_Start.Y - L1_Start.X *
    L1_End.Y) / (L1_End.X - L1_Start.X)
179     Dim NewX As Double = L2_Start.X
180     Dim NewY As Double = A1 * NewX + B1
181     'Dim NewY As Double = (L1_End.Y - L1_Start.Y) / (L1_End.X -
    L1_Start.X) * NewX + (L1_End.X * L1_Start.Y - L1_Start.X *
    L1_End.Y) / (L1_End.X - L1_Start.X)
182     Dim NewPoint As Point = New Point(NewX, NewY)
183     If Segment = True Then
184         If NewPoint.X > Math.Max(L1_Start.X, L1_End.X) Or NewPoint.X <
    Math.Min(L1_Start.X, L1_End.X) Or NewPoint.X > Math.Max
    (L2_Start.X, L2_End.X) Or NewPoint.X < Math.Min(L2_Start.X,
    L2_End.X) Or NewPoint.Y > Math.Max(L1_Start.Y, L1_End.Y) Or
    NewPoint.Y < Math.Min(L1_Start.Y, L1_End.Y) Or NewPoint.Y >
    Math.Max(L2_Start.Y, L2_End.Y) Or NewPoint.Y < Math.Min
    (L2_Start.Y, L2_End.Y) Then Return Nothing
185     Return NewPoint
186 Else
187     Return NewPoint
188 End If
189 Else
190     MsgBox("Issue in Intersect Function")
191     Return Nothing
192 End If
193 End Function
194 Public Function IntersectLineWithBorder(TopLeftBorderPoint As Point,
    BottomRightBorderPoint As Point, LineStart As Point, LineEnd As Point)
    As IntersectWithRectangle
195     Dim CurIntersectWithRectangle As IntersectWithRectangle
196     Dim IntersectionPoint As New Point(0, 0)
197     Dim TopRightBorderPoint = New Point(BottomRightBorderPoint.X,
    TopLeftBorderPoint.Y)
198     Dim BottomLeftBorderPoint = New Point(TopLeftBorderPoint.X,
    BottomRightBorderPoint.Y)
199     If (LineStart.Y < TopLeftBorderPoint.Y Xor LineEnd.Y <
    TopLeftBorderPoint.Y) And Not TopLeftBorderPoint =
    TopRightBorderPoint Then
200         IntersectionPoint = Intersect(TopLeftBorderPoint,
    TopRightBorderPoint, LineStart, LineEnd)
201         IntersectionPoint = New Point(IntersectionPoint.X,
    TopLeftBorderPoint.Y)
202     If Not (IntersectionPoint.X < TopLeftBorderPoint.X Or
    IntersectionPoint.X > BottomRightBorderPoint.X Or
    IntersectionPoint.Y < TopLeftBorderPoint.Y Or
    IntersectionPoint.Y > BottomRightBorderPoint.Y) Then
203         CurIntersectWithRectangle.IntersectionTopBorder = True
204         CurIntersectWithRectangle.IntersectionRightBorder = False

```



```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 6
205 CurIntersectWithRectangle.IntersectionBottomBorder = False
206 CurIntersectWithRectangle.IntersectionLeftBorder = False
207 CurIntersectWithRectangle.IntersectionPoint =
    IntersectionPoint
208 Return CurIntersectWithRectangle
209 End If
210 End If
211 If (LineStart.X > BottomRightBorderPoint.X Xor LineEnd.X >
    BottomRightBorderPoint.X) And Not TopRightBorderPoint =
    BottomRightBorderPoint Then
212 IntersectionPoint = Intersect(TopRightBorderPoint,
    BottomRightBorderPoint, LineStart, LineEnd)
213 IntersectionPoint = New Point(TopRightBorderPoint.X,
    IntersectionPoint.Y)
214 If Not (IntersectionPoint.X < TopLeftBorderPoint.X Or
    IntersectionPoint.X > BottomRightBorderPoint.X Or
    IntersectionPoint.Y < TopLeftBorderPoint.Y Or
    IntersectionPoint.Y > BottomRightBorderPoint.Y) Then
215 CurIntersectWithRectangle.IntersectionTopBorder = False
216 CurIntersectWithRectangle.IntersectionRightBorder = True
217 CurIntersectWithRectangle.IntersectionBottomBorder = False
218 CurIntersectWithRectangle.IntersectionLeftBorder = False
219 CurIntersectWithRectangle.IntersectionPoint =
    IntersectionPoint
220 Return CurIntersectWithRectangle
221 End If
222 End If
223 If (LineStart.Y > BottomRightBorderPoint.Y Xor LineEnd.Y >
    BottomRightBorderPoint.Y) And Not BottomRightBorderPoint =
    BottomLeftBorderPoint Then
224 IntersectionPoint = Intersect(BottomRightBorderPoint,
    BottomLeftBorderPoint, LineStart, LineEnd)
225 IntersectionPoint = New Point(IntersectionPoint.X,
    BottomRightBorderPoint.Y)
226 If Not (IntersectionPoint.X < TopLeftBorderPoint.X Or
    IntersectionPoint.X > BottomRightBorderPoint.X Or
    IntersectionPoint.Y < TopLeftBorderPoint.Y Or
    IntersectionPoint.Y > BottomRightBorderPoint.Y) Then
227 CurIntersectWithRectangle.IntersectionTopBorder = False
228 CurIntersectWithRectangle.IntersectionRightBorder = False
229 CurIntersectWithRectangle.IntersectionBottomBorder = True
230 CurIntersectWithRectangle.IntersectionLeftBorder = False
231 CurIntersectWithRectangle.IntersectionPoint =
    IntersectionPoint
232 Return CurIntersectWithRectangle
233 End If
234 End If
235 If (LineStart.X < TopLeftBorderPoint.X Xor LineEnd.X <
    TopLeftBorderPoint.X) And Not BottomLeftBorderPoint =
    TopLeftBorderPoint Then
236 IntersectionPoint = Intersect(BottomLeftBorderPoint,
    TopLeftBorderPoint, LineStart, LineEnd)
237 IntersectionPoint = New Point(BottomLeftBorderPoint.X,
    IntersectionPoint.Y)
238 If Not (IntersectionPoint.X < TopLeftBorderPoint.X Or
    IntersectionPoint.X > BottomRightBorderPoint.X Or

```

```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 7
    IntersectionPoint.Y < TopLeftBorderPoint.Y Or
    IntersectionPoint.Y > BottomRightBorderPoint.Y) Then
239     CurIntersectWithRectangle.IntersectionTopBorder = False
240     CurIntersectWithRectangle.IntersectionRightBorder = False
241     CurIntersectWithRectangle.IntersectionBottomBorder = False
242     CurIntersectWithRectangle.IntersectionLeftBorder = True
243     CurIntersectWithRectangle.IntersectionPoint =
        IntersectionPoint
244     Return CurIntersectWithRectangle
245 End If
246 End If
247 CurIntersectWithRectangle.IntersectionTopBorder = False
248 CurIntersectWithRectangle.IntersectionRightBorder = False
249 CurIntersectWithRectangle.IntersectionBottomBorder = False
250 CurIntersectWithRectangle.IntersectionLeftBorder = False
251 CurIntersectWithRectangle.IntersectionPoint = Nothing
252 Return CurIntersectWithRectangle
253 End Function
254 Public Function PointInCanvas(BorderTopLeft As Point, BorderBottomRight As
Point, CheckingPoint As Point) As Boolean
255     If BorderTopLeft.X <= CheckingPoint.X And CheckingPoint.X <=
        BorderBottomRight.X And BorderTopLeft.Y <= CheckingPoint.Y And
        CheckingPoint.Y <= BorderBottomRight.Y Then
256         Return True
257     Else
258         Return False
259     End If
260 End Function
261 Public Function CropLine(TopLeftBorderPoint As Point,
        BottomRightBorderPoint As Point, StartPoint As Point, EndPoint As Point,
        Optional ExtendEndPoint As Boolean = False) As CropLineInBorders
262     Dim myCropLineInBorders As New CropLineInBorders
263     ReDim myCropLineInBorders.NewPoints(1)
264     ReDim myCropLineInBorders.PointInCanvas(1)
265     myCropLineInBorders.BorderPoints = New List(Of Point)
266     Dim BottomLeftBorderPoint = New Point(TopLeftBorderPoint.X,
        BottomRightBorderPoint.Y)
267     Dim TopRightBorderPoint = New Point(BottomRightBorderPoint.X,
        TopLeftBorderPoint.Y)
268     myCropLineInBorders.NewPoints(0) = StartPoint
269     myCropLineInBorders.NewPoints(1) = EndPoint
270     myCropLineInBorders.IntersectWithTopBorder = False
271     myCropLineInBorders.IntersectWithRightBorder = False
272     myCropLineInBorders.IntersectWithBottomBorder = False
273     myCropLineInBorders.IntersectWithLeftBorder = False
274     myCropLineInBorders.Include = True
275     myCropLineInBorders.PointInCanvas(0) = PointInCanvas
        (TopLeftBorderPoint, BottomRightBorderPoint, StartPoint)
        'StartPointInCanvas
276     myCropLineInBorders.PointInCanvas(1) = PointInCanvas
        (TopLeftBorderPoint, BottomRightBorderPoint, EndPoint)
        'EndPointInCanvas
277     '--- Both Points In Canvas ---'
278     If myCropLineInBorders.PointInCanvas(0) = True And
        myCropLineInBorders.PointInCanvas(1) = True Then
279         If ExtendEndPoint = True Then MsgBox("Issue in CropLine Function"

```

```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 8
& vbCrLf & "The Line did not extended" & vbCrLf & "(" & 7
StartPoint.X & "|" & StartPoint.Y & ")" (" & EndPoint.X & "|" & 7
EndPoint.Y & ")")
280 GoTo Ret
281 End If
282 '--- First Point In Canvas ---'
283 If myCropLineInBorders.PointInCanvas(0) = True And 7
myCropLineInBorders.PointInCanvas(1) = False Then
284 If Not Intersect(TopLeftBorderPoint, TopRightBorderPoint, 7
StartPoint, EndPoint) = Nothing Then
285 myCropLineInBorders.IntersectWithTopBorder = True
286 myCropLineInBorders.NewPoints(1) = Intersect 7
(TopLeftBorderPoint, TopRightBorderPoint, StartPoint,
EndPoint)
287 myCropLineInBorders.NewPoints(1) = New Point 7
(myCropLineInBorders.NewPoints(1).X, TopRightBorderPoint.Y)
288 End If
289 If Not Intersect(TopRightBorderPoint, BottomRightBorderPoint, 7
StartPoint, EndPoint) = Nothing Then
290 myCropLineInBorders.IntersectWithRightBorder = True
291 myCropLineInBorders.NewPoints(1) = Intersect 7
(TopRightBorderPoint, BottomRightBorderPoint, StartPoint,
EndPoint)
292 myCropLineInBorders.NewPoints(1) = New Point 7
(TopRightBorderPoint.X, myCropLineInBorders.NewPoints(1).Y)
293 End If
294 If Not Intersect(BottomLeftBorderPoint, BottomRightBorderPoint, 7
StartPoint, EndPoint) = Nothing Then
295 myCropLineInBorders.IntersectWithBottomBorder = True
296 myCropLineInBorders.NewPoints(1) = Intersect 7
(BottomLeftBorderPoint, BottomRightBorderPoint, StartPoint,
EndPoint)
297 myCropLineInBorders.NewPoints(1) = New Point 7
(myCropLineInBorders.NewPoints(1).X,
BottomRightBorderPoint.Y)
298 End If
299 If Not Intersect(TopLeftBorderPoint, BottomLeftBorderPoint, 7
StartPoint, EndPoint) = Nothing Then
300 myCropLineInBorders.IntersectWithLeftBorder = True
301 myCropLineInBorders.NewPoints(1) = Intersect 7
(TopLeftBorderPoint, BottomLeftBorderPoint, StartPoint,
EndPoint)
302 myCropLineInBorders.NewPoints(1) = New Point 7
(TopLeftBorderPoint.X, myCropLineInBorders.NewPoints(1).Y)
303 End If
304 If Not (myCropLineInBorders.IntersectWithTopBorder Xor 7
myCropLineInBorders.IntersectWithRightBorder Xor
myCropLineInBorders.IntersectWithBottomBorder Xor
myCropLineInBorders.IntersectWithLeftBorder) Then MsgBox("Issue 7
in CropLine Function" & vbCrLf & "First Point In Canvas" & 7
vbCrLf & "XOR not worked properly")
305 myCropLineInBorders.BorderPoints.Add(myCropLineInBorders.NewPoints 7
(1))
306 GoTo Ret
307 End If
308 '--- Second Point In Canvas ---'

```



```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 9
309 If myCropLineInBorders.PointInCanvas(0) = False And ➤
    myCropLineInBorders.PointInCanvas(1) = True Then
310 If Not Intersect(TopLeftBorderPoint, TopRightBorderPoint, ➤
    StartPoint, EndPoint) = Nothing Then
311 myCropLineInBorders.IntersectWithTopBorder = True
312 myCropLineInBorders.NewPoints(0) = Intersect ➤
    (TopLeftBorderPoint, TopRightBorderPoint, StartPoint, ➤
    EndPoint)
313 myCropLineInBorders.NewPoints(0) = New Point ➤
    (myCropLineInBorders.NewPoints(0).X, TopRightBorderPoint.Y)
314 End If
315 If Not Intersect(TopRightBorderPoint, BottomRightBorderPoint, ➤
    StartPoint, EndPoint) = Nothing Then
316 myCropLineInBorders.IntersectWithRightBorder = True
317 myCropLineInBorders.NewPoints(0) = Intersect ➤
    (TopRightBorderPoint, BottomRightBorderPoint, StartPoint, ➤
    EndPoint)
318 myCropLineInBorders.NewPoints(0) = New Point ➤
    (TopRightBorderPoint.X, myCropLineInBorders.NewPoints(0).Y)
319 End If
320 If Not Intersect(BottomLeftBorderPoint, BottomRightBorderPoint, ➤
    StartPoint, EndPoint) = Nothing Then
321 myCropLineInBorders.IntersectWithBottomBorder = True
322 myCropLineInBorders.NewPoints(0) = Intersect ➤
    (BottomLeftBorderPoint, BottomRightBorderPoint, StartPoint, ➤
    EndPoint)
323 myCropLineInBorders.NewPoints(0) = New Point ➤
    (myCropLineInBorders.NewPoints(0).X, ➤
    BottomRightBorderPoint.Y)
324 End If
325 If Not Intersect(TopLeftBorderPoint, BottomLeftBorderPoint, ➤
    StartPoint, EndPoint) = Nothing Then
326 myCropLineInBorders.IntersectWithLeftBorder = True
327 myCropLineInBorders.NewPoints(0) = Intersect ➤
    (TopLeftBorderPoint, BottomLeftBorderPoint, StartPoint, ➤
    EndPoint)
328 myCropLineInBorders.NewPoints(0) = New Point ➤
    (TopLeftBorderPoint.X, myCropLineInBorders.NewPoints(0).Y)
329 End If
330 If Not (myCropLineInBorders.IntersectWithTopBorder Xor ➤
    myCropLineInBorders.IntersectWithRightBorder Xor ➤
    myCropLineInBorders.IntersectWithBottomBorder Xor ➤
    myCropLineInBorders.IntersectWithLeftBorder) Then MsgBox("Issue ➤
    in CropLine Function" & vbCrLf & "Second Point In Canvas" & ➤
    vbCrLf & "XOR not worked properly")
331 myCropLineInBorders.BorderPoints.Add(myCropLineInBorders.NewPoints ➤
    (0))
332 GoTo Ret
333 End If
334 '--- No Point In Canvas ---'
335 If myCropLineInBorders.PointInCanvas(0) = False And ➤
    myCropLineInBorders.PointInCanvas(1) = False Then
336 Dim PointsAdded As Integer = 0
337 If Not Intersect(TopLeftBorderPoint, TopRightBorderPoint, ➤
    StartPoint, EndPoint) = Nothing Then
338 PointsAdded += 1

```

```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 10
339         myCropLineInBorders.IntersectWithTopBorder = True
340         myCropLineInBorders.NewPoints(PointsAdded - 1) = Intersect
            (TopLeftBorderPoint, TopRightBorderPoint, StartPoint,
            EndPoint)
341         myCropLineInBorders.NewPoints(PointsAdded - 1) = New Point
            (myCropLineInBorders.NewPoints(PointsAdded - 1).X,
            TopRightBorderPoint.Y)
342     End If
343     If Not Intersect(TopRightBorderPoint, BottomRightBorderPoint,
            StartPoint, EndPoint) = Nothing Then
344         PointsAdded += 1
345         myCropLineInBorders.IntersectWithRightBorder = True
346         myCropLineInBorders.NewPoints(PointsAdded - 1) = Intersect
            (TopRightBorderPoint, BottomRightBorderPoint, StartPoint,
            EndPoint)
347         myCropLineInBorders.NewPoints(PointsAdded - 1) = New Point
            (TopRightBorderPoint.X, myCropLineInBorders.NewPoints
            (PointsAdded - 1).Y)
348     End If
349     If Not Intersect(BottomLeftBorderPoint, BottomRightBorderPoint,
            StartPoint, EndPoint) = Nothing Then
350         PointsAdded += 1
351         myCropLineInBorders.IntersectWithBottomBorder = True
352         myCropLineInBorders.NewPoints(PointsAdded - 1) = Intersect
            (BottomLeftBorderPoint, BottomRightBorderPoint, StartPoint,
            EndPoint)
353         myCropLineInBorders.NewPoints(PointsAdded - 1) = New Point
            (myCropLineInBorders.NewPoints(PointsAdded - 1).X,
            BottomRightBorderPoint.Y)
354     End If
355     If Not Intersect(TopLeftBorderPoint, BottomLeftBorderPoint,
            StartPoint, EndPoint) = Nothing Then
356         PointsAdded += 1
357         myCropLineInBorders.IntersectWithLeftBorder = True
358         myCropLineInBorders.NewPoints(PointsAdded - 1) = Intersect
            (TopLeftBorderPoint, BottomLeftBorderPoint, StartPoint,
            EndPoint)
359         myCropLineInBorders.NewPoints(PointsAdded - 1) = New Point
            (TopLeftBorderPoint.X, myCropLineInBorders.NewPoints
            (PointsAdded - 1).Y)
360     End If
361     If PointsAdded = 0 Then
362         myCropLineInBorders.Include = False
363     ElseIf PointsAdded = 2 Then
364         myCropLineInBorders.BorderPoints.Add
            (myCropLineInBorders.NewPoints(0))
365         myCropLineInBorders.BorderPoints.Add
            (myCropLineInBorders.NewPoints(1))
366     Else
367         MsgBox("Issue in CropLine Function" & vbCrLf & "No Point In
            Canvas" & vbCrLf & "Added more than 2 Points")
368     End If
369     End If
370 Ret: Return myCropLineInBorders
371 End Function
372 Public Function NearestPoint(ListOfPoints As List(Of Point), Optional

```

```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 11
    ActualPoint As Boolean = True) As Point
373     Dim DistanceImaginaryPointToPoint(ListOfPoints.Count) As Double
374     Dim X As Double
375     Dim Y As Double
376     Dim Selection As Integer = -1
377     Dim SelectionValue As Double = Double.MaxValue
378     For i = 0 To ListOfPoints.Count - 1
379         X += ListOfPoints(i).X
380         Y += ListOfPoints(i).Y
381     Next
382     X = X / ListOfPoints.Count
383     Y = Y / ListOfPoints.Count
384     If ActualPoint = False Then Return New Point(X, Y)
385     For j = 0 To ListOfPoints.Count - 1
386         DistanceImaginaryPointToPoint(j) = Math.Sqrt((ListOfPoints(j).X -
X) ^ 2 + (ListOfPoints(j).Y - Y) ^ 2)
387     Next
388     For k = 0 To ListOfPoints.Count - 1
389         If DistanceImaginaryPointToPoint(k) <= SelectionValue Then
390             Selection = k
391             SelectionValue = DistanceImaginaryPointToPoint(k)
392         End If
393     Next
394     Selection = Selection
395     Return ListOfPoints(Selection)
396 End Function
397 End Module
398 Public Class Vertex
399     Inherits Shape
400     Implements IVertex
401     Private m_Position As Double()
402     Public ReadOnly Property VertexPosition As Double() Implements
IVertex.Position
403         Get
404             Return Position
405         End Get
406     End Property
407     Protected Overrides ReadOnly Property DefiningGeometry() As Geometry
408     Get
409         Return New EllipseGeometry() With {.Center = New Point(Position
(0), Position(1)), .RadiusX = 1.5, .RadiusY = 1.5}
410     End Get
411 End Property
412 Public Sub New(Optional myfill As Brush = Nothing)
413     Fill = If(myfill, Brushes.Red)
414 End Sub
415 Public Sub New(x As Double, y As Double, Optional myFill As Brush =
Nothing)
416     Me.New(myFill)
417     Position = New Double() {x, y}
418 End Sub
419 Public Function ToPoint() As Point
420     Return New Point(Position(0), Position(1))
421 End Function
422 Public Property Position() As Double()
423     Get

```

```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 12
424     Return m_Position
425     End Get
426     Set
427         m_Position = Value
428     End Set
429 End Property
430 End Class
431 Public Class Cell
432     Inherits TriangulationCell(Of Vertex, Cell)
433     Shared rnd As New Random()
434     Private m_Brush As Brush
435     Private m_Visual As Shape
436     Private m_circumCenter As System.Nullable(Of Point)
437     Private m_centroid As System.Nullable(Of Point)
438     Public Property Brush() As Brush
439     Get
440         Return m_Brush
441     End Get
442     Private Set
443         m_Brush = Value
444     End Set
445 End Property
446 Private Function Det(m As Double(,)) As Double
447     Return m(0, 0) * ((m(1, 1) * m(2, 2)) - (m(2, 1) * m(1, 2))) - m(0, 1) *
        * (m(1, 0) * m(2, 2) - m(2, 0) * m(1, 2)) + m(0, 2) * (m(1, 0) * m
        (2, 1) - m(2, 0) * m(1, 1))
448 End Function
449 Private Function LengthSquared(v As Double()) As Double
450     Dim norm As Double = 0
451     For i As Integer = 0 To v.Length - 1
452         Dim t = v(i)
453         norm += t * t
454     Next
455     Return norm
456 End Function
457 Private Function GetCircumcenter() As Point
458     ' From MathWorld: http://mathworld.wolfram.com/Circumcircle.html
459     Dim points = Vertices
460     Dim m As Double(,) = New Double(2, 2) {}
461     ' x, y, 1
462     For i As Integer = 0 To 2
463         m(i, 0) = points(i).Position(0)
464         m(i, 1) = points(i).Position(1)
465         m(i, 2) = 1
466     Next
467     Dim a = Det(m)
468     ' size, y, 1
469     For i As Integer = 0 To 2
470         m(i, 0) = LengthSquared(points(i).Position)
471     Next
472     Dim dx = -Det(m)
473     ' size, x, 1
474     For i As Integer = 0 To 2
475         m(i, 1) = points(i).Position(0)
476     Next
477     Dim dy = Det(m)

```

```

...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 13
478     ' size, x, y
479     For i As Integer = 0 To 2
480         m(i, 2) = points(i).Position(1)
481     Next
482     Dim c = -Det(m)
483     Dim s = -1.0 / (2.0 * a)
484     Dim r = System.Math.Abs(s) * System.Math.Sqrt(dx * dx + dy * dy - 4 * a * c)
485     Return New Point(s * dx, s * dy)
486 End Function
487 Private Function GetCentroid() As Point
488     Return New Point(Vertices.[Select](Function(v) v.Position(0)).Average(), Vertices.[Select](Function(v) v.Position(1)).Average())
489 End Function
490 Public Property Visual() As Shape
491     Get
492         Return m_Visual
493     End Get
494     Private Set
495         m_Visual = Value
496     End Set
497 End Property
498 Public ReadOnly Property Circumcenter() As Point
499     Get
500         m_circumCenter = If(m_circumCenter, GetCircumcenter())
501         Return m_circumCenter.Value
502     End Get
503 End Property
504 Public ReadOnly Property Centroid() As Point
505     Get
506         m_centroid = If(m_centroid, GetCentroid())
507         Return m_centroid.Value
508     End Get
509 End Property
510 Public Sub New()
511     Visual = New FaceVisual(Me)
512 End Sub
513 Public Class FaceVisual
514     Inherits Shape
515     Private f As Cell
516     Private geometry As Geometry
517     Protected Overrides ReadOnly Property DefiningGeometry() As Geometry
518     Get
519         If geometry IsNot Nothing Then
520             Return geometry
521         End If
522         Dim myPathGeometry = New PathGeometry()
523         Dim pathFigure1 = New PathFigure() With {.StartPoint = New Point(f.Vertices(0).Position(0), f.Vertices(0).Position(1))}
524         For i As Integer = 1 To 2
525             pathFigure1.Segments.Add(New LineSegment(New Point(f.Vertices(i).Position(0), f.Vertices(i).Position(1)), True) With {.IsSmoothJoin = True})
526         Next
527         pathFigure1.IsClosed = True
528         myPathGeometry.Figures.Add(pathFigure1)

```

```
...tudio 2015\Projects\TestRndWPF\TestRndWPF\GlobalVCFS.vb 14
529         Fill = f.Brush
530         geometry = myPathGeometry
531         Return geometry
532     End Get
533 End Property
534 Public Sub New(f As Cell)
535     Stroke = Brushes.Black
536     StrokeThickness = 1.0
537     Opacity = 0.3
538     Me.f = f
539     Dim fill = New SolidColorBrush(Color.FromRgb(CByte(rnd.[Next]
540         (255)), CByte(rnd.[Next](255)), CByte(rnd.[Next](255))))
541     f.Brush = fill
542 End Sub
543 End Class
```



```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 1
1 Imports Telerik.Charting
2 Imports Telerik.Windows.Controls.Charting
3 Imports Telerik.Windows.Controls.ChartView
4 Imports System.Windows.Threading
5 Imports System.ComponentModel
6 Imports System.IO
7 Imports MIConvexHull
8 Partial Public Class MainWindow
9     Inherits Window
10    Private myCurVertices As List(Of Vertex)
11    Private Const NumberOfVertices As Integer = 500
12    Private MyVoronoiMesh As VoronoiMesh(Of Vertex, Cell, VoronoiEdge(Of Vertex, Cell))
13    Public Numbers_Series As New DataSeries()
14    Public DurMathRnd_Series As New DataSeries()
15    Public DurRandomClass_Series As New DataSeries()
16    Public DurCryptographyClass_Series As New DataSeries()
17    Public DurGUID_Series As New DataSeries()
18    Public Points(1, 0) As Double
19    Public BreakOutTimer As DateTime
20    Public WithEvents PointGeneratorBW As New BackgroundWorker
21    #Region "Invokers"
22    Private Delegate Sub SetLbl1(Lbl As Label, Txt As String)
23    Private Delegate Sub SetPoints(PointSystem As ScatterPointSeries, X As Decimal, Y As Decimal)
24    Private Sub UpdateStatusFromPointGeneratorBW(Lbl1 As Label, Txt1 As String)
25        Dim Lbl1Dispatcher As Dispatcher = Lbl1.Dispatcher
26        If Not Lbl1Dispatcher.CheckAccess Then
27            Lbl1Dispatcher.Invoke(New SetLbl1(AddressOf UpdateStatusFromPointGeneratorBW), Lbl1, Txt1)
28        Else
29            Lbl1.Content = Txt1
30        End If
31    End Sub
32    Private Sub UpdatePointsFromPointGeneratorBW(PointSystem As ScatterPointSeries, X As Decimal, Y As Decimal)
33        Dim PointSystemDispatcher As Dispatcher = PointSystem.Dispatcher
34        If Not PointSystemDispatcher.CheckAccess Then
35            PointSystemDispatcher.Invoke(New SetPoints(AddressOf UpdatePointsFromPointGeneratorBW), PointSystem, X, Y)
36        Else
37            PointSystem.DataPoints.Add(New ScatterDataPoint() With {.XValue = X, .YValue = Y})
38        End If
39    End Sub
40    #End Region
41    Private Sub ShowVertices(vertices As List(Of Vertex))
42        For i = 0 To vertices.Count - 1
43            drawingCanvas.Children.Add(vertices(i))
44        Next
45    End Sub
46    Private Sub MakeGrid(n As Integer, vertices As List(Of Vertex))
47        Dim sizeX = drawingCanvas.ActualWidth
48        Dim sizeY = drawingCanvas.ActualHeight
49        For i As Integer = 0 To n - 1

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 2
50     For j As Integer = 0 To n - 1
51         Dim vi = New Vertex(i * sizeX / (n - 1), j * sizeY / (n - 1))
52         vertices.Add(vi)
53     Next
54     Next
55 End Sub
56 Private Sub MakeCircle(n As Integer, vertices As List(Of Vertex))
57     Dim sizeX = drawingCanvas.ActualWidth
58     Dim sizeY = drawingCanvas.ActualHeight
59     For i As Integer = 0 To n - 2
60         Dim x = 0.5 * sizeX + 0.38 * sizeX * Math.Cos(i * 2 * Math.PI / (n
        - 1))
61         Dim y = 0.5 * sizeY + 0.38 * sizeY * Math.Sin(i * 2 * Math.PI / (n
        - 1))
62         vertices.Add(New Vertex(x, y))
63     Next
64 End Sub
65 Private Sub MakeRandom(n As Integer, vertices As List(Of Vertex))
66     Dim r = New Random()
67     Dim sizeX = drawingCanvas.ActualWidth
68     Dim sizeY = drawingCanvas.ActualHeight
69     For i = 0 To n - 1
70         Dim vi = New Vertex(sizeX * r.NextDouble(), sizeY * r.NextDouble
        ())
71         vertices.Add(vi)
72     Next
73 End Sub
74 Private Sub Create(curVertices As IList(Of Vertex))
75     ShowVertices(curVertices)
76     Try
77         MyVoronoiMesh = VoronoiMesh.Create(Of Vertex, Cell)(curVertices)
78     Catch ex As Exception
79         MessageBox.Show(ex.Message, "Error")
80     Return
81 End Try
82 Info4_Value_Lbl.Content = MyVoronoiMesh.Vertices.Count()
83 '---Calculate Area---'
84 Dim CurCell_Ax As Double = 0
85 Dim CurCell_Ay As Double = 0
86 Dim CurCell_Bx As Double = 0
87 Dim CurCell_By As Double = 0
88 Dim CurCell_Cx As Double = 0
89 Dim CurCell_Cy As Double = 0
90 DelauneyArea = 0
91 For Each cell In MyVoronoiMesh.Vertices
92     CurCell_Ax = cell.Vertices(0).VertexPosition(0) / Tx * Xmax + Xmin
93     CurCell_Ay = (Ty - cell.Vertices(0).VertexPosition(1)) / Ty * Ymax
        + Ymin
94     CurCell_Bx = cell.Vertices(1).VertexPosition(0) / Tx * Xmax + Xmin
95     CurCell_By = (Ty - cell.Vertices(1).VertexPosition(1)) / Ty * Ymax
        + Ymin
96     CurCell_Cx = cell.Vertices(2).VertexPosition(0) / Tx * Xmax + Xmin
97     CurCell_Cy = (Ty - cell.Vertices(2).VertexPosition(1)) / Ty * Ymax
        + Ymin
98     DelauneyArea += Math.Abs(CurCell_Ax * (CurCell_By - CurCell_Cy) +
        CurCell_Bx * (CurCell_Cy - CurCell_Ay) + CurCell_Cx *

```



```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 3
(CurCell_Ay - CurCell_By)) / 2
99     Next
100     Info5_Value_Lbl.Content = Math.Round(DelauneyArea, 2)
101     myCurVertices = curVertices
102     Delaunay_Btn.IsEnabled = True
103     Voronoi_Btn.IsEnabled = True
104 End Sub
105 Private Sub DrawPoints_Btn_Click(sender As Object, e As RoutedEventArgs)
    Handles DrawPoints_Btn.Click
106     '---Epilogi simeiw---'
107     Tx = drawingCanvas.ActualWidth
108     Ty = drawingCanvas.ActualHeight
109     Dim myBrush As Brush = Brushes.Red
110     Dim ArrayOfIndexes(0) As Integer
111     myVertices.Clear()
112     myBaseVertices.Clear()
113     Info6_Value_Lbl.Content = ""
114     NoOfBaseCoordinates = Val(NoOfBaseCoordinates_TB.Text)
115     ReDim ArrayOfIndexes(NoOfBaseCoordinates - 1)
116     If NoOfBaseCoordinates > 0 And NoOfBaseCoordinates <= PointsCounter
        Then
117         If NoOfBaseCoordinates = PointsCounter Then
118             For M = 0 To NoOfBaseCoordinates - 1
119                 ArrayOfIndexes(M) = M
120             Next
121         Else
122             For J = 0 To NoOfBaseCoordinates - 1
123 Repeat2:                 Dim TempIndex = RandomClass.Next(0, PointsCounter)
124                         For K = 0 To NoOfBaseCoordinates - 1
125                             If ArrayOfIndexes(K) = TempIndex Then GoTo Repeat2
126                         Next
127                         ArrayOfIndexes(J) = TempIndex
128                     Next
129             End If
130         Else
131             MsgBox("Παρακαλώ επιλέξτε Πλήθος Κέντρων απο 1 έως " &
                PointsCounter & ".")
132             Exit Sub
133         End If
134     End If
135     For i = 0 To PointsCounter - 1
136         If NoOfBaseCoordinates > 0 Then
137             myBrush = New SolidColorBrush(Color.FromArgb(100, 255, 128,
                0))
138             For L = 0 To ArrayOfIndexes.GetLength(0) - 1
139                 If i = ArrayOfIndexes(L) Then
140                     myBrush = Brushes.Navy
141                     Exit For
142                 End If
143             Next
144         Else
145             myBrush = Brushes.Navy
146         End If
147         Dim X = (Points(0, i) - Xmin) / Xmax * Tx
148         Dim Y = Ty - ((Points(1, i) - Ymin) / Ymax * Ty)
149         Dim vi = New Vertex(X, Y, myBrush)
    End Sub

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 4
150     If myBrush Is Brushes.Navy Then
151         myBaseVertices.Add(vi)
152     Else
153         myVertices.Add(vi)
154     End If
155     Next
156     drawingCanvas.Children.Clear()
157     If NoOfBaseCoordinates > 0 Then ShowVertices(myVertices)
158     Create(myBaseVertices)
159     Delaunay_Btn.IsEnabled = True
160     Voronoi_Btn.IsEnabled = True
161     NewPoints_Btn.IsEnabled = False
162     Output_Btn.IsEnabled = True
163     Info7_Value_Lbl.Content = ""
164 End Sub
165 Private Sub btnMakeGrid_Click(sender As Object, e As RoutedEventArgs)
166     Dim vs = New List(Of Vertex)()
167     MakeGrid(10, vs)
168     Create(vs)
169 End Sub
170 Private Sub btnMakeCircle_Click(sender As Object, e As RoutedEventArgs)
171     Dim vs = New List(Of Vertex)()
172     MakeCircle(25, vs)
173     Create(vs)
174 End Sub
175 Private Sub btnMakeFancy_Click(sender As Object, e As RoutedEventArgs)
176     Dim vs = New List(Of Vertex)()
177     MakeGrid(10, vs)
178     MakeCircle(25, vs)
179     Create(vs)
180 End Sub
181 Private Sub Delaunay_Btn_Click(sender As Object, e As RoutedEventArgs)
182     Handles Delaunay_Btn.Click
183     drawingCanvas.Children.Clear()
184     Delaunay_Btn.IsEnabled = False
185     Voronoi_Btn.IsEnabled = True
186     NewPoints_Btn.IsEnabled = False
187     For Each cell In MyVoronoiMesh.Vertices
188         drawingCanvas.Children.Add(cell.Visual)
189     Next
190     ShowVertices(myCurVertices)
191 End Sub
192 Private Shared Function PointInCell(c As Cell, p As Point) As Boolean
193     Dim v1 = c.Vertices(0).ToPoint()
194     Dim v2 = c.Vertices(1).ToPoint()
195     Dim v3 = c.Vertices(2).ToPoint()
196     Dim s0 = IsLeft(v1, v2, p)
197     Dim s1 = IsLeft(v2, v3, p)
198     Dim s2 = IsLeft(v3, v1, p)
199     Return (s0 = s1) AndAlso (s1 = s2)
200 End Function
201 Private Shared Function IsLeft(a As Point, b As Point, c As Point) As
202     Integer
203     Return If(((b.X - a.X) * (c.Y - a.Y) - (b.Y - a.Y) * (c.X - a.X)) > 0,
204         1, -1)
205 End Function

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 5
203 Private Shared Function Center(c As Cell) As Point
204     Dim v1 As Vector = c.Vertices(0).ToPoint()
205     Dim v2 As Vector = c.Vertices(1).ToPoint()
206     Dim v3 As Vector = c.Vertices(2).ToPoint()
207     Return (v1 + v2 + v3) / 3
208 End Function
209 Private Sub ExecuteVoronoi()
210     TopLeftCanvasBorder = New Point(0, 0)
211     TopRightCanvasBorder = New Point(drawingCanvas.ActualWidth, 0)
212     BottomRightCanvasBorder = New Point(drawingCanvas.ActualWidth,
213         drawingCanvas.ActualHeight)
214     BottomLeftCanvasBorder = New Point(0, drawingCanvas.ActualHeight)
215     If Not IntersectBorderPoints Is Nothing Then
216         IntersectBorderPoints.Clear()
217     '--- EDGES ---'
218     For Each edge In MyVoronoiMesh.Edges
219         Dim edge_from As Point = edge.Source.Circumcenter
220         Dim edge_to As Point = edge.Target.Circumcenter
221         Dim CurCropLineInBorders = CropLine(TopLeftCanvasBorder,
222             BottomRightCanvasBorder, edge_from, edge_to)
223         If CurCropLineInBorders.Include = True Then
224             drawingCanvas.Children.Add(New Shapes.Line() With {.X1 =
225                 CurCropLineInBorders.NewPoints(0).X, .Y1 =
226                 CurCropLineInBorders.NewPoints(0).Y, .X2 =
227                 CurCropLineInBorders.NewPoints(1).X, .Y2 =
228                 CurCropLineInBorders.NewPoints(1).Y, .Stroke =
229                 Brushes.Blue})
230             AddEdge(CurCropLineInBorders.NewPoints(0).X,
231                 CurCropLineInBorders.NewPoints(0).Y,
232                 CurCropLineInBorders.NewPoints(1).X,
233                 CurCropLineInBorders.NewPoints(1).Y)
234             If CurCropLineInBorders.BorderPoints.Count > 0 Then
235                 For Each BP As Point In CurCropLineInBorders.BorderPoints
236                     IntersectBorderPoints.Add(BP)
237                 Next
238             End If
239         End If
240     Next
241     '--- CELLS ---'
242     For Each cell In MyVoronoiMesh.Vertices
243         For i As Integer = 0 To 2
244             Dim Iteration = i
245             If cell.Adjacency(i) Is Nothing Then
246                 Dim cell_from = cell.Circumcenter
247                 Dim t = cell.Vertices.Where(Function(myVertex, ParameterJ)
248                     ParameterJ <> Iteration).ToArray()
249                 Dim factor = 100 * IsLeft(t(0).ToPoint(), t(1).ToPoint(),
250                     cell_from) * IsLeft(t(0).ToPoint(), t(1).ToPoint(), Center
251                     (cell))
252                 Dim dir = New Point(0.5 * (t(0).Position(0) + t
253                     (1).Position(0)), 0.5 * (t(0).Position(1) + t(1).Position
254                     (1))) - cell_from
255                 Dim Ffactor As UInt64 = 0
256                 Dim cell_to As Point
257                 'Do
258                 Ffactor += 1

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 6
243 Dim midpoint_dir = (factor * Ffactor) * dir
244 'If factor < 0 And Integer.TryParse(Ffactor / 2,
Nothing) Then
245 ' midpoint_dir = -(factor ^ Ffactor) * dir
246 'Else
247 ' midpoint_dir = (factor ^ Ffactor) * dir
248 'End If
249 cell_to = cell_from + midpoint_dir
250 'Exit Sub
251 'Loop Until (cell_to.X < TopLeftCanvasBorder.X Or
cell_to.X > BottomRightCanvasBorder.X) And (cell_to.Y <
TopLeftCanvasBorder.Y Or cell_to.Y >
BottomRightCanvasBorder.Y)
252 Dim CurCropLineInBorders = CropLine(TopLeftCanvasBorder,
BottomRightCanvasBorder, cell_from, cell_to, True)
253 If CurCropLineInBorders.Include = True Then
254 drawingCanvas.Children.Add(New Shapes.Line() With {.X1 =
CurCropLineInBorders.NewPoints(0).X, .Y1 =
CurCropLineInBorders.NewPoints(0).Y, .X2 =
CurCropLineInBorders.NewPoints(1).X, .Y2 =
CurCropLineInBorders.NewPoints(1).Y, .Stroke =
Brushes.DarkGreen})
255 AddEdge(CurCropLineInBorders.NewPoints(0).X,
CurCropLineInBorders.NewPoints(0).Y,
CurCropLineInBorders.NewPoints(1).X,
CurCropLineInBorders.NewPoints(1).Y)
256 If CurCropLineInBorders.BorderPoints.Count > 0 Then
257 For Each BP As Point In
CurCropLineInBorders.BorderPoints
258 IntersectBorderPoints.Add(BP)
259 Next
260 End If
261 End If
262 End If
263 Next
264 Next
265 '--- Sort Border Intersection Points---'
266 Dim LeftArray As New List(Of Double)
267 Dim TopArray As New List(Of Double)
268 Dim RightArray As New List(Of Double)
269 Dim BottomArray As New List(Of Double)
270 LeftArray.Add(TopLeftCanvasBorder.Y)
271 LeftArray.Add(BottomRightCanvasBorder.Y)
272 TopArray.Add(TopLeftCanvasBorder.X)
273 TopArray.Add(BottomRightCanvasBorder.X)
274 RightArray.Add(TopLeftCanvasBorder.Y)
275 RightArray.Add(BottomRightCanvasBorder.Y)
276 BottomArray.Add(TopLeftCanvasBorder.X)
277 BottomArray.Add(BottomRightCanvasBorder.X)
278 For Each IBPoint In IntersectBorderPoints
279 If IBPoint.X = TopLeftCanvasBorder.X And Not (IBPoint.Y =
TopLeftCanvasBorder.Y Or IBPoint.Y = BottomRightCanvasBorder.Y)
Then LeftArray.Add(IBPoint.Y)
280 If IBPoint.Y = TopLeftCanvasBorder.Y And Not (IBPoint.X =
TopLeftCanvasBorder.X Or IBPoint.Y = BottomRightCanvasBorder.X)
Then TopArray.Add(IBPoint.X)

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 7
281     If IBPoint.X = BottomRightCanvasBorder.X And Not (IBPoint.Y =
        TopLeftCanvasBorder.Y Or IBPoint.Y = BottomRightCanvasBorder.Y)
        Then RightArray.Add(IBPoint.Y)
282     If IBPoint.Y = BottomRightCanvasBorder.Y And Not (IBPoint.X =
        TopLeftCanvasBorder.X Or IBPoint.Y = BottomRightCanvasBorder.X)
        Then BottomArray.Add(IBPoint.X)
283     Next
284     LeftArray.Sort()
285     TopArray.Sort()
286     RightArray.Sort()
287     BottomArray.Sort()
288     '--- Create Edges based in Border Intersection Points---'
289     For Coordinate = 0 To LeftArray.Count - 2
290         AddEdge(TopLeftCanvasBorder.X, LeftArray(Coordinate),
        TopLeftCanvasBorder.X, LeftArray(Coordinate + 1))
291         drawingCanvas.Children.Add(New Shapes.Line() With {.X1 =
        TopLeftCanvasBorder.X, .Y1 = LeftArray(Coordinate), .X2 =
        TopLeftCanvasBorder.X, .Y2 = LeftArray(Coordinate + 1), .Stroke
        = Brushes.Black})
292     Next
293     For Coordinate = 0 To TopArray.Count - 2
294         AddEdge(TopArray(Coordinate), TopLeftCanvasBorder.Y, TopArray
        (Coordinate + 1), TopLeftCanvasBorder.Y)
295         drawingCanvas.Children.Add(New Shapes.Line() With {.X1 = TopArray
        (Coordinate), .Y1 = TopLeftCanvasBorder.Y, .X2 = TopArray
        (Coordinate + 1), .Y2 = TopLeftCanvasBorder.Y, .Stroke =
        Brushes.Black})
296     Next
297     For Coordinate = 0 To RightArray.Count - 2
298         AddEdge(BottomRightCanvasBorder.X, RightArray(Coordinate),
        BottomRightCanvasBorder.X, RightArray(Coordinate + 1))
299         drawingCanvas.Children.Add(New Shapes.Line() With {.X1 =
        BottomRightCanvasBorder.X, .Y1 = RightArray(Coordinate), .X2 =
        BottomRightCanvasBorder.X, .Y2 = RightArray(Coordinate +
        1), .Stroke = Brushes.Black})
300     Next
301     For Coordinate = 0 To BottomArray.Count - 2
302         AddEdge(BottomArray(Coordinate), BottomRightCanvasBorder.Y,
        BottomArray(Coordinate + 1), BottomRightCanvasBorder.Y)
303         drawingCanvas.Children.Add(New Shapes.Line() With {.X1 =
        BottomArray(Coordinate), .Y1 = BottomRightCanvasBorder.Y, .X2 =
        BottomArray(Coordinate + 1), .Y2 =
        BottomRightCanvasBorder.Y, .Stroke = Brushes.Black})
304     Next
305     '--- END - Create Edges based in Border Intersection Points---'
306     If NoOfBaseCoordinates > 0 Then ShowVertices(myVertices)
307     ShowVertices(myCurVertices)
308     Info6_Value_Lbl.Content = GetPolygons()
309 End Sub
310 Private Sub Voronoi_Btn_Click(sender As Object, e As RoutedEventArgs)
        Handles Voronoi_Btn.Click
311     Info7_Value_Lbl.Content = ""
312     VoronoiEdges.Clear()
313     drawingCanvas.Children.Clear()
314     'Voronoi_Btn.IsEnabled = False
315     Delaunay_Btn.IsEnabled = False

```



```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 8
316     NewPoints_Btn.IsEnabled = False
317     ExecuteVoronoi()
318     'Voronoi_Btn.IsEnabled = False
319     Delaunay_Btn.IsEnabled = True
320     NewPoints_Btn.IsEnabled = True
321 End Sub
322 Private Sub NewPoints_Btn_Click(sender As Object, e As RoutedEventArgs)
    Handles NewPoints_Btn.Click
323     Info7_Value_Lbl.Content = Val(Info7_Value_Lbl.Content) + 1
324     Dim TempPoints As New List(Of Point)
325     Dim PolygonPoints() As Point
326     Dim GroupList As New List(Of List(Of Point))
327     Dim PointsList As New List(Of Point)
328     Dim NewBasePoints As New List(Of Point)
329     Dim NewNonBasePoints As New List(Of Point)
330     For i = 0 To Points.GetLength(1) - 1
331         Dim X = (Points(0, i) - Xmin) / Xmax * Tx
332         Dim Y = Ty - ((Points(1, i) - Ymin) / Ymax * Ty)
333         TempPoints.Add(New Point(X, Y))
334         NewNonBasePoints.Add(New Point(X, Y))
335     Next
336     For j = 0 To VoronoiPolygons.Count - 1
337         If Not PointsList.Count = 0 Then PointsList.Clear()
338         ReDim PolygonPoints(VoronoiPolygons(j).NumPoints - 2)
339         For k = 0 To VoronoiPolygons(j).NumPoints - 2
340             PolygonPoints(k) = New Point(VoronoiPolygons(j).Coordinates
                (k).X, VoronoiPolygons(j).Coordinates(k).Y)
341         Next
342         For l = TempPoints.Count - 1 To 0 Step -1
343             If PointInPolygon(PolygonPoints, TempPoints(l).X, TempPoints
                (l).Y) Then
344                 PointsList.Add(TempPoints(l))
345             End If
346         Next
347         NewBasePoints.Add(NearestPoint(PointsList,
                ActualPoints_ChkB.IsChecked))
348     Next
349     For m = NewNonBasePoints.Count - 1 To 0 Step -1
350         For n = NewBasePoints.Count - 1 To 0 Step -1
351             If NewNonBasePoints(m).X = NewBasePoints(n).X And
                NewNonBasePoints(m).Y = NewBasePoints(n).Y Then
352                 NewNonBasePoints.RemoveAt(m)
353             Exit For
354             End If
355         Next
356     Next
357     NoOfBaseCoordinates_TB.Text = NewBasePoints.Count
358
359     '--- Create New Vertices---'
360     myVertices.Clear()
361     myBaseVertices.Clear()
362     Dim myBrush = New SolidColorBrush(Color.FromArgb(100, 255, 128, 0))
363     For p = 0 To NewNonBasePoints.Count - 1
364         myVertices.Add(New Vertex(NewNonBasePoints(p).X, NewNonBasePoints
                (p).Y, myBrush))
365     Next

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 9
366
367     For q = 0 To NewBasePoints.Count - 1
368         myBaseVertices.Add(New Vertex(NewBasePoints(q).X, NewBasePoints
           (q).Y, Brushes.Navy))
369     Next
370     drawingCanvas.Children.Clear()
371     If NoOfBaseCoordinates > 0 Then ShowVertices(myVertices)
372     Create(myBaseVertices)
373     '--- Run Voronoi Diagram ---'
374     VoronoiEdges.Clear()
375     drawingCanvas.Children.Clear()
376     Voronoi_Btn.IsEnabled = False
377     Delaunay_Btn.IsEnabled = False
378     NewPoints_Btn.IsEnabled = False
379     ExecuteVoronoi()
380     Delaunay_Btn.IsEnabled = True
381     NewPoints_Btn.IsEnabled = True
382     'MsgBox("Done!")
383 End Sub
384
385 Private Sub MainWindow_Loaded(sender As Object, e As RoutedEventArgs)
           Handles Me.Loaded
386     Randomize()
387     myRnd(RandomSystem.VBMathRnd)
388     myRnd(RandomSystem.RandomCls)
389     myRnd(RandomSystem.CryptographyClass)
390     'AddHandler RequestTimeout.Tick, AddressOf RequestTimeout_Tick
391
392     InterruptTimer_TB.Text = 30
393     'RequestTimeout.Interval = New TimeSpan(0, 0, CInt
           (InterruptTimer_TB.Text))
394
395     Numbers_Series.Definition = New ScatterSeriesDefinition
396     Numbers_Series.Definition.ShowItemLabels = False
397     Numbers_Series.Definition.Appearance.PointMark.Shape =
           MarkerShape.Circle
398     Numbers_Series.Definition.Appearance.PointMark.StrokeThickness = 0
399     Numbers_RChrt.DefaultView.ChartLegend.UseAutoGeneratedItems = True
400     Numbers_RChrt.DefaultView.ChartArea.ClearSelection()
401     Numbers_RChrt.DefaultView.ChartArea.AxisX.Title = "Α/Α των Αριθμών"
402     Numbers_RChrt.DefaultView.ChartArea.AxisY.Title = "Τιμή Αριθμού"
403     Numbers_RChrt.DefaultView.ChartArea.DataSeries.Add(Numbers_Series)
404
405     DurMathRnd_Series.Definition = New LineSeriesDefinition
406     DurMathRnd_Series.Definition.ShowItemLabels = False
407     DurMathRnd_Series.Definition.Appearance.PointMark.Shape =
           MarkerShape.Triangle
408     DurMathRnd_Series.Definition.Appearance.PointMark.StrokeThickness = 0
409     Durations_RChrt.DefaultView.ChartArea.DataSeries.Add
           (DurMathRnd_Series)
410
411     DurRandomClass_Series.Definition = New LineSeriesDefinition
412     DurRandomClass_Series.Definition.ShowItemLabels = False
413     DurRandomClass_Series.Definition.Appearance.PointMark.Shape =
           MarkerShape.Square
414     DurRandomClass_Series.Definition.Appearance.PointMark.StrokeThickness

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 10
= 0
415 Durations_RChrt.DefaultView.ChartArea.DataSeries.Add ➤
    (DurRandomClass_Series)
416
417 DurCryptographyClass_Series.Definition = New LineSeriesDefinition
418 DurCryptographyClass_Series.Definition.ShowItemLabels = False
419 DurCryptographyClass_Series.Definition.Appearance.PointMark.Shape = ➤
    MarkerShape.Hexagon
420
DurCryptographyClass_Series.Definition.Appearance.PointMark.StrokeThickness = ➤
0
421 Durations_RChrt.DefaultView.ChartArea.DataSeries.Add ➤
    (DurCryptographyClass_Series)
422
423 DurGUID_Series.Definition = New LineSeriesDefinition
424 DurGUID_Series.Definition.ShowItemLabels = False
425 DurGUID_Series.Definition.Appearance.PointMark.Shape = ➤
    MarkerShape.Circle
426 DurGUID_Series.Definition.Appearance.PointMark.StrokeThickness = 0
427 Durations_RChrt.DefaultView.ChartArea.DataSeries.Add(DurGUID_Series)
428
429 Durations_RChrt.DefaultView.ChartLegend.UseAutoGeneratedItems = True
430 Durations_RChrt.DefaultView.ChartArea.AxisX.Title = "Πλήθος Αριθμών"
431 Durations_RChrt.DefaultView.ChartArea.AxisY.Title = "Χρόνος Εκτέλεσης ➤
    log(msec)"
432 Durations_RChrt.DefaultView.ChartArea.AxisY.IsLogarithmic = True
433 Durations_RChrt.DefaultView.ChartArea.DataSeries(0).LegendLabel = ➤
    "Math.Rnd"
434 Durations_RChrt.DefaultView.ChartArea.DataSeries(1).LegendLabel = ➤
    "Random"
435 Durations_RChrt.DefaultView.ChartArea.DataSeries(2).LegendLabel = ➤
    "Cryptography"
436 Durations_RChrt.DefaultView.ChartArea.DataSeries(3).LegendLabel = ➤
    "GUID"
437 End Sub
438 Protected Overrides Sub OnKeyDown(ByVal e As KeyEventArgs)
439     If e.Key = Key.F1 Then
440         Process.Start("Help.pdf")
441     Else
442         MyBase.OnKeyDown(e)
443     End If
444 End Sub
445 Private Sub RunRandomize_Btn_Click(sender As Object, e As RoutedEventArgs) ➤
    Handles RunRandomize_Btn.Click
446     Dim No As Integer
447     Try
448         No = CInt(NoOfNumbers_TB.Text)
449     Catch
450         Exit Sub
451     End Try
452
453     If ShowNumberChart_ChkB.IsChecked = True Then
454         Numbers_Series.Clear()
455         For i = 1 To No
456             Numbers_Series.Add(New ➤
                Telerik.Windows.Controls.Charting.DataPoint(i, myRnd ➤

```



```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 11
(CurrentRandomSystem))
457     Next
458 Else
459     Dim Watch As Stopwatch = Stopwatch.StartNew()
460     For i = 1 To No
461         myRnd(CurrentRandomSystem)
462     Next
463     Watch.Stop()
464     Select Case CurrentRandomSystem
465     Case RandomSystem.VBMathRnd
466         DurMathRnd_Series.Add(New Telerik.Windows.Controls.Charting.DataPoint(No, Watch.ElapsedTicks))
467     Case RandomSystem.RandomCls
468         DurRandomClass_Series.Add(New Telerik.Windows.Controls.Charting.DataPoint(No, Watch.ElapsedTicks))
469     Case RandomSystem.CryptographyClass
470         DurCryptographyClass_Series.Add(New Telerik.Windows.Controls.Charting.DataPoint(No, Watch.ElapsedTicks))
471     Case RandomSystem.GUID
472         DurGUID_Series.Add(New Telerik.Windows.Controls.Charting.DataPoint(No, Watch.ElapsedTicks))
473     End Select
474     End If
475 End Sub
476 Private Sub RandomSystem_Checked(sender As Object, e As RoutedEventArgs)
477     Handles MathRnd_RB.Checked,
478     RandomClass_RB.Checked,
479     RNG_CSP_RB.Checked,
480     GUID_RB.Checked
481     If sender Is MathRnd_RB Then CurrentRandomSystem = RandomSystem.VBMathRnd
482     If sender Is RandomClass_RB Then CurrentRandomSystem = RandomSystem.RandomCls
483     If sender Is RNG_CSP_RB Then CurrentRandomSystem = RandomSystem.CryptographyClass
484     If sender Is GUID_RB Then CurrentRandomSystem = RandomSystem.GUID
485 End Sub
486 Private Sub GeneratePoints_Btn_Click(sender As Object, e As EventArgs)
487     Handles GeneratePoints_Btn.Click
488     GeneratePoints_Btn.IsEnabled = False
489     If Trim(Xmin_TB.Text) = String.Empty Or Trim(Xmax_TB.Text) = String.Empty Or Trim(Ymin_TB.Text) = String.Empty Or Trim(Ymax_TB.Text) = String.Empty Or Trim(NoOfCoordinates_TB.Text) = String.Empty Then
490         MsgBox("Please Fill in all the fields", MsgBoxStyle.Exclamation, "Attention...")
491     End If
492     GeneratePoints_Btn.IsEnabled = True
493 End Sub

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 12
492     If Not IsNumeric(Xmin_TB.Text) AndAlso Not IsNumeric(Xmax_TB.Text)  ?
        AndAlso Not IsNumeric(Ymin_TB.Text) AndAlso Not IsNumeric  ?
        (Ymax_TB.Text) AndAlso Not IsNumeric(NoOfCoordinates_TB.Text) Then  ?
493     MsgBox("Please Fill in all the fields with numeric values",  ?
        MsgBoxStyle.Exclamation, "Attention...")  ?
494     GeneratePoints_Btn.IsEnabled = True
495     Exit Sub
496     End If
497     Xmin = Xmin_TB.Text
498     Xmin_TB.Text = Xmin
499     Xmax = Xmax_TB.Text
500     Xmax_TB.Text = Xmax
501     Ymin = Ymin_TB.Text
502     Ymin_TB.Text = Ymin
503     Ymax = Ymax_TB.Text
504     Ymax_TB.Text = Ymax
505     If DispersionMin_ChkB.IsChecked = True Or DispersionMax_ChkB.IsChecked  ?
        = True Then
506     Dim MaxDistance As Double = CalcDistance(Xmin, Ymin, Xmax, Ymax)
507     Dim SafeDistance As Double = ((Math.Abs(Xmax - Xmin) * Math.Abs  ?
        (Ymax - Ymin)) ^ 0.5) / Math.Ceiling(CInt  ?
        (NoOfCoordinates_TB.Text) ^ 0.5)
508     If DispersionMin_ChkB.IsChecked = True Then
509     If Val(DispersionMin_TB.Text) >= MaxDistance Then
510     MsgBox("The Minimum Dispersion Distance should be smaller  ?
        than" & Math.Round(SafeDistance, 4) & ". The maximum  ?
        distance you can use is " & Math.Round(MaxDistance, 4),  ?
        "Warning!")
511     GeneratePoints_Btn.IsEnabled = True
512     Exit Sub
513     End If
514     If Val(DispersionMin_TB.Text) > SafeDistance Then
515     If InterruptTimer_ChkB.IsChecked = False Then
516     Dim Res As MessageBoxResult = MessageBox.Show("The  ?
        Minimum Dispersion Distance should be smaller than " &  ?
        Math.Round(SafeDistance, 4) & ". Its very unlikely to  ?
        generate " & NoOfCoordinates_TB.Text & " coordinates!" &  ?
        vbCrLf & "It is highly recommended to enable Timeout  ?
        option" & vbCrLf & vbCrLf & "Press Yes if you want to  ?
        enable Timeout option" & vbCrLf & "Press No if you want to  ?
        continue without Timeout option" & vbCrLf & "Press Cancel  ?
        if you want to abort", "Warning!",  ?
        MessageBoxButton.YesNoCancel)
517     If Res = MessageBoxResult.Cancel Then
518     GeneratePoints_Btn.IsEnabled = True
519     Exit Sub
520     End If
521     If Res = MessageBoxResult.Yes Then
522     InterruptTimer_TB.Text = 30
523     InterruptTimer_ChkB.IsChecked = True
524     End If
525     Else
526     MsgBox("The Minimum Dispersion Distance should be  ?
        smaller than " & Math.Round(SafeDistance, 4) & ". " &  ?
        vbCrLf & "Its difficult to generate " &  ?
        NoOfCoordinates_TB.Text & " coordinates!")

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 13
527         End If
528     End If
529     If DispersionMax_ChkB.IsChecked = True And Val
530         (DispersionMax_TB.Text) > MaxDistance Then
531         MsgBox("The Maximum Dispersion Option is obsolete because
532             the Maximum Dispersion value you have entered is larger
533             than the Maximum distance (" & Math.Round(MaxDistance, 4)
534             & ")", "Info...")
535     End If
536 End If
537 End If
538 If Xmin > Xmax Then
539     MsgBox("Last time I checked the " & Xmin & " was greater than " &
540         Xmax, MsgBoxStyle.Exclamation, "Προσοχή!")
541     GeneratePoints_Btn.IsEnabled = True
542     Exit Sub
543 End If
544 If Ymin > Ymax Then
545     MsgBox("Last time I checked the " & Ymin & " was greater than " &
546         Ymax, MsgBoxStyle.Exclamation, "Προσοχή!")
547     GeneratePoints_Btn.IsEnabled = True
548     Exit Sub
549 End If
550 If Xmin = Xmax And Ymin = Ymax Then
551     MsgBox("I believe that you are messing with me..." & Ymin,
552         MsgBoxStyle.Exclamation, "Προσοχή!")
553     GeneratePoints_Btn.IsEnabled = True
554     Exit Sub
555 End If
556 If Xmin = Xmax Then
557     Dim ResX As MessageBoxResult
558     ResX = MessageBox.Show("You know that I am going to generate
559         coordinates in a straight line, yes?", "Προσοχή!",
560         MessageBoxButton.YesNo)
561     If ResX = MessageBoxResult.No Then
562         GeneratePoints_Btn.IsEnabled = True
563         Exit Sub
564     End If
565 End If
566 If Ymin = Ymax Then
567     Dim ResY As MessageBoxResult
568     ResY = MessageBox.Show("You know that I am going to generate
569         coordinates in a straight line, yes?", "Προσοχή!",
570         MessageBoxButton.YesNo)
571     If ResY = MessageBoxResult.No Then
572         GeneratePoints_Btn.IsEnabled = True
573         Exit Sub
574     End If
575 End If
576 If DispersionMin_ChkB.IsChecked AndAlso DispersionMax_ChkB.IsChecked
577     AndAlso Val(DispersionMin_TB.Text) = Val(DispersionMax_TB.Text) Then
578     MsgBox("Η Τιμές Ελάχιστης και Μέγιστης Διασποράς δεν μπορούν να
579         είναι ίσες", MsgBoxStyle.Exclamation, "Προσοχή!")
580     GeneratePoints_Btn.IsEnabled = True
581     Exit Sub
582 End If
583 End If
584 End If
585 End If
586 End If
587 End If
588 End If
589 End If

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 14
570
571     If DispersionMin_ChkB.IsChecked AndAlso DispersionMax_ChkB.IsChecked Then
572         MsgBox("Η Τιμές Ελάχιστης και Μέγιστης Διασποράς θα αλλάξουν
573             μεταξύ τους", MsgBoxStyle.Exclamation, "Προσοχή!")
574         Dim VarForReplace = Val(DispersionMin_TB.Text)
575         DispersionMin_TB.Text = Val(DispersionMax_TB.Text)
576         DispersionMax_TB.Text = VarForReplace
577     End If
578     NoOfCoordinates = Val(NoOfCoordinates_TB.Text)
579     InterruptTimer = CInt(InterruptTimer_TB.Text)
580     DispersionMinChecked = DispersionMin_ChkB.IsChecked
581     DispersionMaxChecked = DispersionMax_ChkB.IsChecked
582     InterruptTimerChecked = InterruptTimer_ChkB.IsChecked
583     DispersionMin = Val(DispersionMin_TB.Text)
584     DispersionMax = Val(DispersionMax_TB.Text)
585     Points_Series2.DataPoints.Clear()
586     PointGeneratorBW.RunWorkerAsync()
587 End Sub
588 Private Sub PointGeneratorBW_Dowork(sender As Object, e As
589     DoWorkEventArgs) Handles PointGeneratorBW.Dowork
590     '-----Generate-----'
591     ReDim Points(1, 0)
592     PointsCounter = 0
593     Dim InitialPointFlag As Boolean = True
594     Dim TempX As Double
595     Dim TempY As Double
596     Dim DispersionCounter As Integer = 0
597     BreakOutTimer = Now
598     For i = 1 To NoOfCoordinates
599 Repeat: DispersionCounter += 1
600     If BreakOutTimer.AddSeconds(InterruptTimer) < Now Then
601         If InterruptTimerChecked Then
602             Dim Res As MessageBoxResult = MessageBox.Show("Το χρονικό
603                 περιθώριο των " & InterruptTimer & " δευτερολέπτων για την
604                 εύρεση του επόμενου στοιχείου τελείωσε." & vbCrLf &
605                 "Θέλετε να συνεχίσετε;", "Προσοχή...",
606                 MessageBoxButton.YesNo)
607             If Res = MessageBoxResult.No Then Exit Sub
608             BreakOutTimer = Now
609         End If
610     End If
611     Dim x1 As Integer = 0
612     Dim y1 As Integer = 0
613     If Xmin = Xmax Then
614         TempX = Xmax
615     Else
616         x1 = 0
617         Do
618             x1 += 1
619             TempX = Xmin * (1 - myRnd(CurrentRandomSystem)) + Xmax *
620                 myRnd(CurrentRandomSystem)
621         Loop Until TempX >= Xmin AndAlso TempX <= Xmax
622     End If
623     If Ymin = Ymax Then
624         TempY = Ymax

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 15
618 Else
619     y1 = 0
620     Do
621         y1 += 1
622         TempY = Ymin * (1 - myRnd(CurrentRandomSystem)) + Ymax *
myRnd(CurrentRandomSystem)
623         Loop Until TempY >= Ymin AndAlso TempY <= Ymax
624         Console.WriteLine("Random X: " & x1 & " | Random Y: " & y1 & " |
Dispersion Counter: " & DispersionCounter & vbCrLf)
625     End If
626
627     '-----Check for Dispersion-----'
628     If (DispersionMinChecked = True Or DispersionMaxChecked = True)
And Points.GetLength(1) > 0 And Not Points(0, 0) = 0 Then
629         Dim DispersionMaxFlag As Boolean = False
630         For j = 0 To Points.GetLength(1) - 1
631             Dim TempDistance = CalcDistance(Points(0, j), Points(1,
j), TempX, TempY)
632             If TempDistance < DispersionMin And DispersionMinChecked =
True Then GoTo Repeat 'Na απέχουν όλα
μεταξύ τους τουλάχιστον DispersionMin
633             If TempDistance < DispersionMax And DispersionMaxChecked =
True Then DispersionMaxFlag = True 'Na απέχει
τουλάχιστον με ένα άλλο σημείο το πολύ DispersionMax
634         Next
635         If DispersionMaxFlag = False And DispersionMaxChecked = True
Then GoTo Repeat
636     End If
637     PointsCounter = Points.GetLength(1)
638     If InitialPointFlag = True Then
639         InitialPointFlag = False
640         PointsCounter = 0
641     Else
642         ReDim Preserve Points(1, PointsCounter)
643     End If
644     Points(0, PointsCounter) = CDec(TempX)
645     Points(1, PointsCounter) = CDec(TempY)
646     UpdateStatusFromPointGeneratorBW(Info1_Value_Lbl, PointsCounter +
1)
647     UpdatePointsFromPointGeneratorBW(Points_Series2, TempX, TempY)
648     DispersionCounter = 0
649     If InterruptTimerChecked = True Then BreakOutTimer = Now
650     Next
651     PointsCounter = Points.GetLength(1)
652 End Sub
653 Private Sub PointGeneratorBW_RunWorkerCompleted(sender As Object, e As
RunWorkerCompletedEventArgs) Handles PointGeneratorBW.RunWorkerCompleted
654     GeneratePoints_Btn.IsEnabled = True
655     DrawPoints_Btn.IsEnabled = True
656 End Sub
657 Private Sub NoOfCoordinates_TB_TextChanged(sender As Object, e As
EventArgs) Handles NoOfCoordinates_TB.TextChanged
658     If IsNumeric(NoOfCoordinates_TB.Text) = False Then
NoOfCoordinates_TB.Text = "0"
659     NoOfCoordinates_TB.Text = NoOfCoordinates_TB.Text.ToString.Replace
(", ", ", ")

```



```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 16
660 NoOfCoordinates_TB.Text = NoOfCoordinates_TB.Text.ToString.Replace
    (" ", "")
661 NoOfCoordinates_TB.CaretIndex = NoOfCoordinates_TB.Text.Length
662 Info2_Value_Lbl.Content = Math.Round(CalcDistance(Cdbl(Val
    (Xmin_TB.Text)), Cdbl(Val(Ymin_TB.Text)), Cdbl(Val(Xmax_TB.Text)),
    Cdbl(Val(Ymax_TB.Text))), 4)
663 If Val(NoOfCoordinates_TB.Text) <> 0 Then Info3_Value_Lbl.Content =
    Math.Round(((Math.Abs(Cdbl(Val(Xmax_TB.Text)) - Cdbl(Val
    (Xmin_TB.Text))) * Math.Abs(Cdbl(Val(Ymax_TB.Text)) - Cdbl(Val
    (Ymin_TB.Text)))) ^ 0.5) / Math.Ceiling(CInt
    (NoOfCoordinates_TB.Text) ^ 0.5), 4)
664 End Sub
665 Private Sub Xmin_TB_TextChanged(sender As Object, e As EventArgs) Handles
    Xmin_TB.TextChanged
666 If IsNumeric(Xmin_TB.Text) = False Then Xmin_TB.Text = "0"
667 Xmin_TB.Text = Xmin_TB.Text.ToString.Replace(".", ",")
668 Xmin_TB.CaretIndex = Xmin_TB.Text.Length
669 Info2_Value_Lbl.Content = Math.Round(CalcDistance(Cdbl(Val
    (Xmin_TB.Text)), Cdbl(Val(Ymin_TB.Text)), Cdbl(Val(Xmax_TB.Text)),
    Cdbl(Val(Ymax_TB.Text))), 4)
670 If Val(NoOfCoordinates_TB.Text) <> 0 Then Info3_Value_Lbl.Content =
    Math.Round(((Math.Abs(Cdbl(Val(Xmax_TB.Text)) - Cdbl(Val
    (Xmin_TB.Text))) * Math.Abs(Cdbl(Val(Ymax_TB.Text)) - Cdbl(Val
    (Ymin_TB.Text)))) ^ 0.5) / Math.Ceiling(CInt
    (NoOfCoordinates_TB.Text) ^ 0.5), 4)
671 End Sub
672 Private Sub Xmax_TB_TextChanged(sender As Object, e As EventArgs) Handles
    Xmax_TB.TextChanged
673 If IsNumeric(Xmax_TB.Text) = False Then Xmax_TB.Text = "0"
674 Xmax_TB.Text = Xmax_TB.Text.ToString.Replace(".", ",")
675 Xmax_TB.CaretIndex = Xmax_TB.Text.Length
676 Info2_Value_Lbl.Content = Math.Round(CalcDistance(Cdbl(Val
    (Xmin_TB.Text)), Cdbl(Val(Ymin_TB.Text)), Cdbl(Val(Xmax_TB.Text)),
    Cdbl(Val(Ymax_TB.Text))), 4)
677 If Val(NoOfCoordinates_TB.Text) <> 0 Then Info3_Value_Lbl.Content =
    Math.Round(((Math.Abs(Cdbl(Val(Xmax_TB.Text)) - Cdbl(Val
    (Xmin_TB.Text))) * Math.Abs(Cdbl(Val(Ymax_TB.Text)) - Cdbl(Val
    (Ymin_TB.Text)))) ^ 0.5) / Math.Ceiling(CInt
    (NoOfCoordinates_TB.Text) ^ 0.5), 4)
678 End Sub
679 Private Sub Ymin_TB_TextChanged(sender As Object, e As EventArgs) Handles
    Ymin_TB.TextChanged
680 If IsNumeric(Ymin_TB.Text) = False Then Ymin_TB.Text = "0"
681 Ymin_TB.Text = Ymin_TB.Text.ToString.Replace(".", ",")
682 Ymin_TB.CaretIndex = Ymin_TB.Text.Length
683 Info2_Value_Lbl.Content = Math.Round(CalcDistance(Cdbl(Val
    (Xmin_TB.Text)), Cdbl(Val(Ymin_TB.Text)), Cdbl(Val(Xmax_TB.Text)),
    Cdbl(Val(Ymax_TB.Text))), 4)
684 If Val(NoOfCoordinates_TB.Text) <> 0 Then Info3_Value_Lbl.Content =
    Math.Round(((Math.Abs(Cdbl(Val(Xmax_TB.Text)) - Cdbl(Val
    (Xmin_TB.Text))) * Math.Abs(Cdbl(Val(Ymax_TB.Text)) - Cdbl(Val
    (Ymin_TB.Text)))) ^ 0.5) / Math.Ceiling(CInt
    (NoOfCoordinates_TB.Text) ^ 0.5), 4)
685 End Sub
686 Private Sub Ymax_TB_TextChanged(sender As Object, e As EventArgs) Handles
    Ymax_TB.TextChanged

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 17
687     If IsNumeric(Ymax_TB.Text) = False Then Ymax_TB.Text = "0"
688     Ymax_TB.Text = Ymax_TB.Text.ToString.Replace(".", ",")
689     Ymax_TB.CaretIndex = Ymax_TB.Text.Length
690     Info2_Value_Lbl.Content = Math.Round(CalcDistance(Cdbl(Val
        (Xmin_TB.Text)), Cdbl(Val(Ymin_TB.Text)), Cdbl(Val(Xmax_TB.Text)),
        Cdbl(Val(Ymax_TB.Text))), 4)
691     If Val(NoOfCoordinates_TB.Text) <> 0 Then Info3_Value_Lbl.Content =
        Math.Round(((Math.Abs(Cdbl(Val(Xmax_TB.Text)) - Cdbl(Val
        (Xmin_TB.Text))) * Math.Abs(Cdbl(Val(Ymax_TB.Text)) - Cdbl(Val
        (Ymin_TB.Text)))) ^ 0.5) / Math.Ceiling(CInt
        (NoOfCoordinates_TB.Text) ^ 0.5), 4)
692 End Sub
693 Private Sub DispersionMin_TB_TextChanged(sender As Object, e As EventArgs)
        Handles DispersionMin_TB.TextChanged
694     If IsNumeric(DispersionMin_TB.Text) = False Then DispersionMin_TB.Text
        = "0"
695     DispersionMin_TB.Text = DispersionMin_TB.Text.ToString.Replace(",",
        ".")
696     DispersionMin_TB.CaretIndex = DispersionMin_TB.Text.Length
697     If Cdbl(DispersionMin_TB.Text) < 0 Then DispersionMin_TB.Text =
        Math.Abs(Cdbl(DispersionMin_TB.Text))
698 End Sub
699 Private Sub DispersionMax_TB_TextChanged(sender As Object, e As EventArgs)
        Handles DispersionMax_TB.TextChanged
700     If IsNumeric(DispersionMax_TB.Text) = False Then DispersionMax_TB.Text
        = "0"
701     DispersionMax_TB.Text = DispersionMax_TB.Text.ToString.Replace(",",
        ".")
702     DispersionMax_TB.CaretIndex = DispersionMax_TB.Text.Length
703     If Cdbl(DispersionMax_TB.Text) < 0 Then DispersionMax_TB.Text =
        Math.Abs(Cdbl(DispersionMax_TB.Text))
704 End Sub
705 Private Sub Output_Btn_Click(sender As Object, e As RoutedEventArgs)
        Handles Output_Btn.Click
706     Dim Output_SFD As New Microsoft.Win32.SaveFileDialog
707     Output_SFD.FileName = "PolyCenters"
708     Output_SFD.DefaultExt = ".csv"
709     Output_SFD.Filter = "CSV Documents (*.csv)|*.csv"
710     Dim OutputRes = Output_SFD.ShowDialog
711     If OutputRes = True Then
712         Dim WriteObj As New StreamWriter(Output_SFD.FileName)
713         Dim CentersQty As Integer = myBaseVertices.Count
714         For i = 0 To CentersQty - 1
715             Dim Gx As Double = (Cdbl(myBaseVertices(i).VertexPosition(0))
                * Xmax / Tx) + Xmin
716             Dim Gy As Double = ((Ty - Cdbl(myBaseVertices
                (i).VertexPosition(1))) * Ymax / Ty) + Ymin
717             WriteObj.WriteLine(Gx & ";" & Gy)
718             Console.WriteLine(Gx & ";" & Gy)
719         Next
720         WriteObj.Close()
721     End If
722 End Sub
723 Private Sub ImportPoints_Btn_Click(sender As Object, e As RoutedEventArgs)
        Handles ImportPoints_Btn.Click
724     Dim Load_SFD As New Microsoft.Win32.OpenFileDialog

```

```

... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 18
725 Load_SFD.DefaultExt = ".csv"
726 Load_SFD.Filter = "CSV Documents (.csv)|*.csv|Text Documents (.txt)|
*.txt|All Files (*.*)|*.*"
727 Dim LoadRes = Load_SFD.ShowDialog
728 If LoadRes = True Then
729     ImportPoints_Btn.IsEnabled = False
730     DrawPoints_Btn.IsEnabled = False
731     Dim ReadObj As StreamReader = New StreamReader(Load_SFD.FileName)
732     Points_Series2.DataPoints.Clear()
733     ReDim Points(1, 0)
734     Xmin = Double.MaxValue
735     Xmax = Double.MinValue
736     Ymin = Double.MaxValue
737     Ymax = Double.MinValue
738     Do While ReadObj.Peek() >= 0
739         Dim TempLine As String = ReadObj.ReadLine()
740         Dim CurPos = Points.GetLength(1) - 1
741         Points(0, CurPos) = Trim(TempLine.Substring(0,
TempLine.IndexOf(";"))).Replace(".", ",")
742         Points(1, CurPos) = Trim(TempLine.Substring(TempLine.IndexOf
(";") + 1)).Replace(".", ",")
743         If Points(0, CurPos) < Xmin Then Xmin = Points(0, CurPos)
744         If Points(0, CurPos) > Xmax Then Xmax = Points(0, CurPos)
745         If Points(1, CurPos) < Ymin Then Ymin = Points(1, CurPos)
746         If Points(1, CurPos) > Ymax Then Ymax = Points(1, CurPos)
747         Points_Series2.DataPoints.Add(New ScatterDataPoint() With
{.XValue = Points(0, CurPos), .YValue = Points(1, CurPos)})
748         ReDim Preserve Points(1, CurPos + 1)
749     Loop
750     PointsCounter = Points.GetLength(1) - 1
751     ReDim Preserve Points(1, PointsCounter - 1)
752     Xmin_Value_Lbl.Content = Xmin
753     Xmax_Value_Lbl.Content = Xmax
754     Ymin_Value_Lbl.Content = Ymin
755     Ymax_Value_Lbl.Content = Ymax
756     ImportPoints_Btn.IsEnabled = True
757     NoOfBaseCoordinates_TB_TextChanged(sender, Nothing)
758 End If
759 End Sub
760 Private Sub NoOfBaseCoordinates_TB_TextChanged(sender As Object, e As
TextChangedEventArgs) Handles NoOfBaseCoordinates_TB.TextChanged
761     If Integer.TryParse(NoOfBaseCoordinates_TB.Text, Nothing) And Val
(NoOfBaseCoordinates_TB.Text) > 0 And Val
(NoOfBaseCoordinates_TB.Text) <= PointsCounter Then
762         DrawPoints_Btn.IsEnabled = True
763     Else
764         DrawPoints_Btn.IsEnabled = False
765     End If
766     Delaunay_Btn.IsEnabled = False
767     Voronoi_Btn.IsEnabled = False
768     NewPoints_Btn.IsEnabled = False
769     Output_Btn.IsEnabled = False
770 End Sub
771 Private Sub Grid_TB_TextChanged(sender As Object, e As
TextChangedEventArgs) Handles PointsDistance_TB.TextChanged,
772

```



```

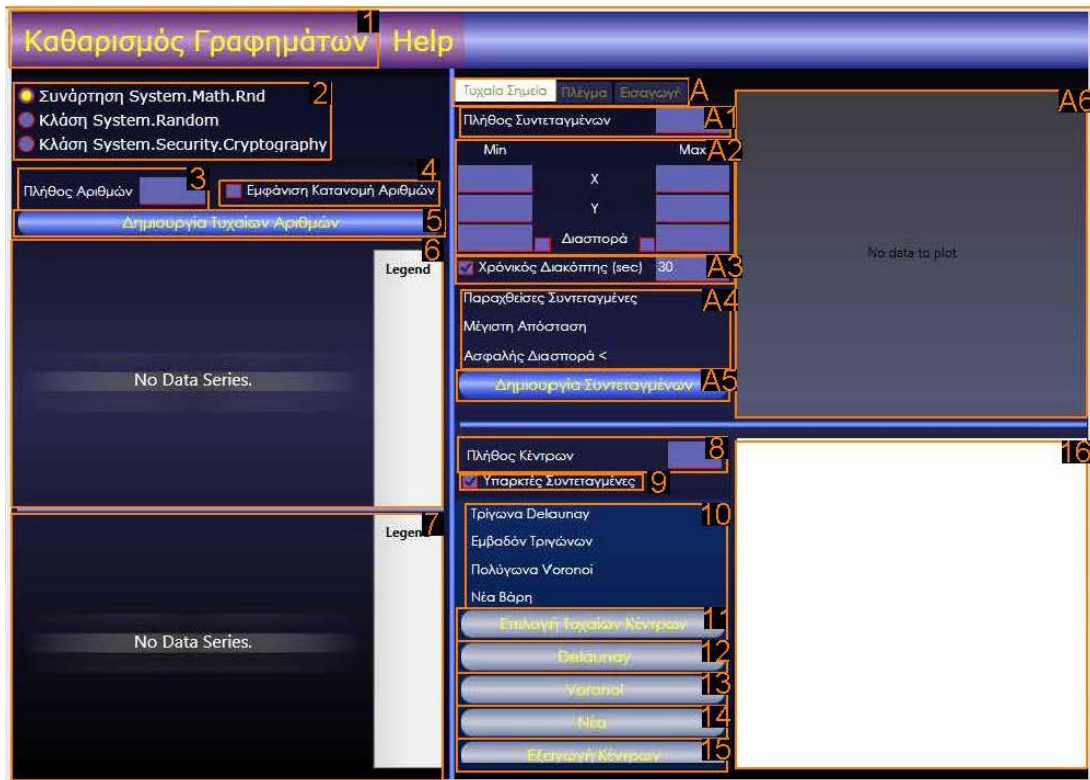
... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 19
    GXmin_TB.TextChanged,
773                                     ?
    GYmin_TB.TextChanged,
774                                     ?
    GXmax_TB.TextChanged,
775                                     ?
    GYmax_TB.TextChanged
776 If Integer.TryParse(PointsDistance_TB.Text, Nothing) And ?
    Integer.TryParse(GXmin_TB.Text, Nothing) And Integer.TryParse ?
    (GYmin_TB.Text, Nothing) And Integer.TryParse(GXmax_TB.Text, ?
    Nothing) And Integer.TryParse(GYmax_TB.Text, Nothing) Then ?
777     GenerateGrid_Btn.IsEnabled = True
778 Else
779     GenerateGrid_Btn.IsEnabled = False
780 End If
781 End Sub
782 Private Sub GenerateGrid_Btn_Click(sender As Object, e As RoutedEventArgs) ?
    Handles GenerateGrid_Btn.Click
783     If GXmin_TB.Text = GXmax_TB.Text Then
784         MsgBox("Xmin και Xmax έχουν την ίδια τιμη")
785         Exit Sub
786     End If
787     If GYmin_TB.Text = GYmax_TB.Text Then
788         MsgBox("Ymin και Ymax έχουν την ίδια τιμη")
789         Exit Sub
790     End If
791     If Val(GXmax_TB.Text) < Val(GXmin_TB.Text) Then
792         Dim AltX = GXmin_TB.Text
793         GXmin_TB.Text = GXmax_TB.Text
794         GXmax_TB.Text = AltX
795     End If
796     If Val(GYmax_TB.Text) < Val(GYmin_TB.Text) Then
797         Dim AltY = GYmin_TB.Text
798         GYmin_TB.Text = GYmax_TB.Text
799         GYmax_TB.Text = AltY
800     End If
801     If (Val(GXmax_TB.Text) - Val(GXmin_TB.Text)) / 2 < ?
    PointsDistance_TB.Text Then ?
802         MsgBox("Επιλέξτε μικρότερη απόσταση σημείων ή επιφάνεια με ?
    μεγαλύτερο απόσταση στον άξονα X") ?
803         Exit Sub
804     End If
805     If (Val(GYmax_TB.Text) - Val(GYmin_TB.Text)) / 2 < ?
    PointsDistance_TB.Text Then ?
806         MsgBox("Επιλέξτε μικρότερη απόσταση σημείων ή επιφάνεια με ?
    μεγαλύτερο απόσταση στον άξονα Y") ?
807         Exit Sub
808     End If
809     Xmin = Val(GXmin_TB.Text)
810     Xmax = Val(GXmax_TB.Text)
811     Ymin = Val(GYmin_TB.Text)
812     Ymax = Val(GYmax_TB.Text)
813     Dim PointsStep As Double = Val(PointsDistance_TB.Text)
814     Points_Series2.DataPoints.Clear()
815     ReDim Points(1, 0)
816     Dim StepX As Double = Xmin + PointsStep

```

```
... 2015\Projects\TestRndWPF\TestRndWPF\MainWindow.xaml.vb 20
817 Dim StepY As Double = Ymin + PointsStep
818 PointsCounter = 0
819 Do While StepY < Ymax
820     Do While StepX < Xmax
821         PointsCounter += 1
822         ReDim Preserve Points(1, PointsCounter - 1)
823         Points(0, PointsCounter - 1) = StepX
824         Points(1, PointsCounter - 1) = StepY
825         Points_Series2.DataPoints.Add(New ScatterDataPoint() With {
            .XValue = StepX, .YValue = StepY})
826         StepX += PointsStep
827     Loop
828     StepY += PointsStep
829     StepX = Xmin + PointsStep
830 Loop
831 End Sub
832 End Class
```



## **ΠΑΡΑΡΤΗΜΑ ΙΙΙ ΕΓΧΕΙΡΙΔΙΟ**



1. Ενεργό κουμπί «Καθαρισμός Γραφημάτων». Καθαρίζει τα αποτελέσματα στα Πεδία γραφημάτων 6 και 7
2. Επιλογή συνάρτησης ή κλάσης για την παραγωγή τυχαίων αριθμών, για σύγκριση διασποράς και χρόνου μεταξύ των υπολοίπων συναρτήσεων (6) ή κλάσεων ή για την παραγωγή τυχαίων συντεταγμένων (7)
3. Πεδίο εισαγωγής κειμένου για την συμπλήρωση του πλήθους αριθμών που θα εμφανιστούν
4. Επιλέγοντας το πεδίο ενεργοποίησης «Εμφάνιση Κατανομής Αριθμών» εμφανίζονται στο σημειόγραμμα (6) το πλήθος των αριθμών που ζητήθηκε στο (3)
5. Ενεργό κουμπί για τη δημιουργία τυχαίων αριθμών
6. Σημειόγραμμα (κατακόρυφος άξονας: εύρος τιμής αριθμού, οριζόντιος άξονας: αύξουσα αριθμηση).
7. Διάγραμμα χρόνου εκτέλεσης συνάρτησης ή κλάσης ανά πλήθος δημιουργημένων αριθμών.

## A. Παραγωγή Τυχαίων Σημείων

1. Πεδίο εισαγωγής κειμένου για την εισαγωγή πλήθους συντεταγμένων προς δημιουργία
2. Φόρμα συμπλήρωσης επίλογων για τη δημιουργία σημείων
  - a. Πεδίο κειμένου για την εισαγωγή του κατώτατου ορίου στον άξονα των  $X$
  - b. Πεδίο κειμένου για την εισαγωγή του ανώτατου ορίου στον άξονα των  $X$
  - c. Πεδίο κειμένου για την εισαγωγή του κατώτατου ορίου στον άξονα των  $Y$
  - d. Πεδίο κειμένου για την εισαγωγή του ανώτατου ορίου στον άξονα των  $Y$
  - e. Πεδίο κειμένου για την εισαγωγή της ελάχιστης απόστασης οποιασδήποτε συντεταγμένης από οποιαδήποτε άλλη.
  - f. Πεδίο ενεργοποίησης για την ενεργοποίηση της χρήσης της ελάχιστης απόστασης οποιασδήποτε συντεταγμένης από οποιαδήποτε άλλη.
  - g. Πεδίο ενεργοποίησης για την ενεργοποίηση της χρήσης της μέγιστης απόστασης οποιασδήποτε συντεταγμένης από οποιαδήποτε άλλη.
  - h. Πεδίο κειμένου για την εισαγωγή της μέγιστης απόστασης οποιασδήποτε συντεταγμένης από οποιαδήποτε άλλη.
3. Εισαγωγή ανώτατου χρόνου λειτουργίας (διάρκεια μέχρι τη διακοπή) χωρίς τη δημιουργία νέας συντεταγμένης.
4. Πεδίο Πληροφοριών
  - a. Παραχθείσες Συντεταγμένες
  - b. Μέγιστη Απόσταση εντός του καμβά
  - c. Ελάχιστη ασφαλής απόσταση διασποράς
5. Κουμπί εκτέλεσης διαδικασίας παραγωγής συντεταγμένων
6. Γραφική αναπαράσταση των παραχθέντων συντεταγμένων σε σημειόγραμμα

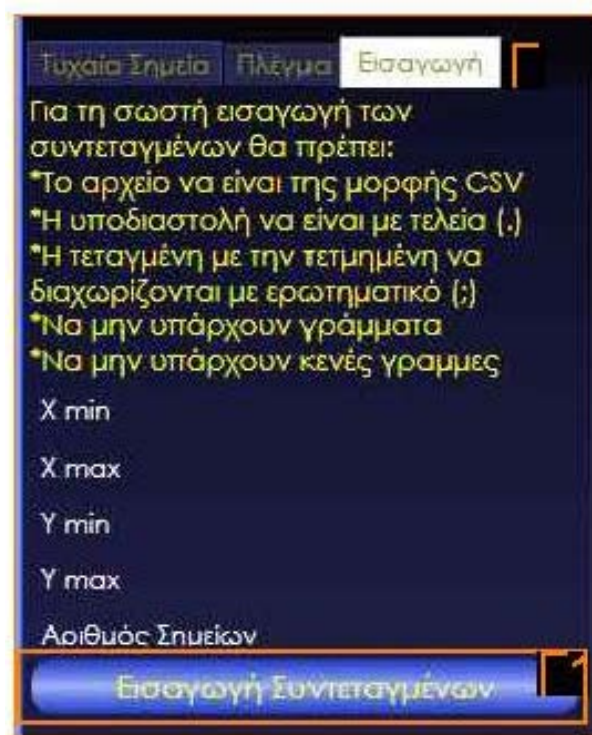
## B. Δημιουργία Πλέγματος



1. Το βήμα δημιουργίας σημείων οριζόντια και κάθετα.
2. Φόρμα συμπλήρωσης επίλογων για τη δημιουργία πλέγματος
  - a. Πεδίο κειμένου για την εισαγωγή του κατώτατου ορίου στον άξονα των X
  - b. Πεδίο κειμένου για την εισαγωγή του ανώτατου ορίου στον άξονα των X
  - c. Πεδίο κειμένου για την εισαγωγή του κατώτατου ορίου στον άξονα των Y
  - d. Πεδίο κειμένου για την εισαγωγή του ανώτατου ορίου στον άξονα των Y
3. Πεδίο Πληροφοριών που μας δηλώνει το πλήθος των δημιουργημένων σημείων.
4. Κουμπί εκτέλεσης διαδικασίας δημιουργίας πλέγματος



## Γ. Εισαγωγή σημείων



1. Πεδίο Πληροφοριών
  - a. Το κατώτατο όριο στον άξονα των X
  - b. Το ανώτατο όριο στον άξονα των X
  - c. Το κατώτατο όριο στον άξονα των Y
  - d. Το ανώτατο όριο στον άξονα των Y
  - e. Το πλήθος των εισαχθέντων συντεταγμένων.
2. Ενεργό κουμπί που εκκινεί την διαδικασία της επιλογής αρχείου προς εισαγωγή και εισαγωγή των συντεταγμένων.

8. Πεδίο κειμένου για την εισαγωγή των επιλεγμένων συντεταγμένων ως κέντρα.  
Η τιμή πρέπει να είναι μεγαλύτερη του 3 και μικρότερη του πλήθους των δημιουργημένων συντεταγμένων.
9. Πεδίο ενεργοποίησης για την ενεργοποίηση της επιλογής του κοντινότερου σημείου ως κέντρο.
10. Πεδίο Πληροφοριών
  - a. Πλήθος τριγώνων Delaunay
  - b. Συνολικό εμβαδό τριγώνων Delaunay
  - c. Πλήθος Πολύγωνων Voronoi
11. Κουμπί εκτέλεσης για τη δημιουργία στον καμβά Απεικόνισης τις παραχθείσες συντεταγμένες και τις συντεταγμένες που έχουν επιλεγθεί ως κέντρα.  
Κάθε επανάληψη δημιουργεί στον καμβά απεικόνισης τις ίδιες παραχθείσες συντεταγμένες αλλά διαφορετικές συντεταγμένες που έχουν επιλεγθεί ως κέντρα.
12. Κουμπί εκτέλεσης για την εμφάνιση στον καμβά Απεικόνισης τα τρίγωνα Delaunay βάσει των παραχθέντων συντεταγμένων και των συντεταγμένων που έχουν επιλεγθεί ως κέντρα.
13. Κουμπί εκτέλεσης για την εμφάνιση στον καμβά Απεικόνισης των πολυγώνων Voronoi βάσει των παραχθέντων συντεταγμένων και των συντεταγμένων που έχουν επιλεγθεί ως κέντρα.
14. Κουμπί εκτέλεσης για την εμφάνιση στον καμβά Απεικόνισης των νέων πολυγώνων Voronoi βάσει των παραχθέντων συντεταγμένων και των νέων συντεταγμένων που έχουν επιλεγθεί ως κέντρα βάσει της προηγούμενης εκτέλεσης Voronoi.
15. Καμβάς γραφιστικής απεικόνισης σημείων κ σχεδιασμών.