Υπολογιστική Γεωμετρία & Εφαρμογές 3Δ Μοντελοποίησης

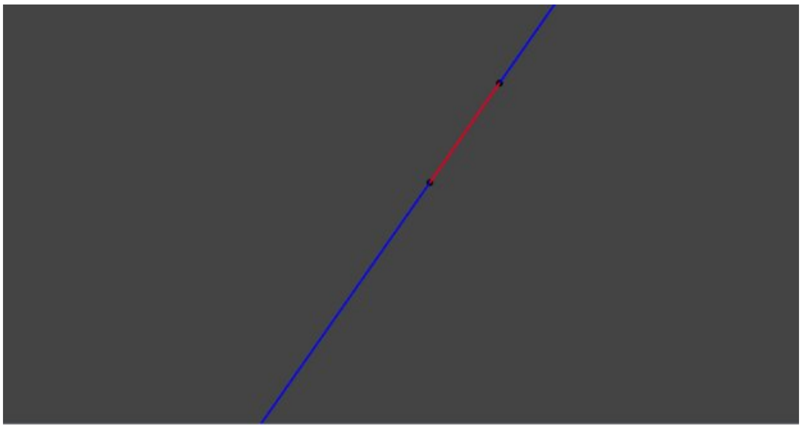
Εργαστήριο 2

Γιώργος Μπολάτογλου

Αρ. Μητρώου: 228424

Task 1

Καλούμαστε να απεικονίσουμε 2 αυθαίρετα σημεία Α και Β, ένα ευθύγραμμο τμήμα και μια ευθεία ΑΒ.



Task 2

Απεικονίστε το σημείο στο οποίο πατιέται/σύρεται το ποντίκι. Χρησιμοποιείστε διαφορετικό χρώμα για το παραπάνω σημείο ανάλογα με το αν βρίσκεται αριστερά/δεξιά από το ΑΒ.

Για να το πετύχουμε αυτό, υλοποιούμε συνάρτηση που ελέγχει αν το σημείο που που κάνουμε click/pressed βρίσκεται πάνω ή κάτω από την ευθεία ΑΒ.

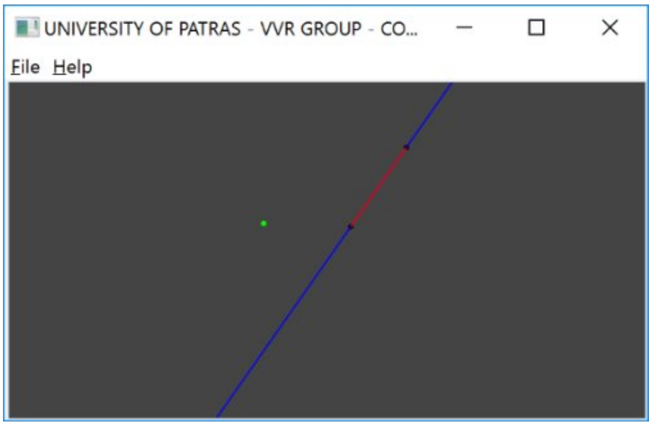
Έχουμε βρει την εξίσωση της ευθείας: (y-y1)=((y1-y2)/(x1-x2))*(x-x1) , όπου A=(x1,y1) και B=(x2,y2).

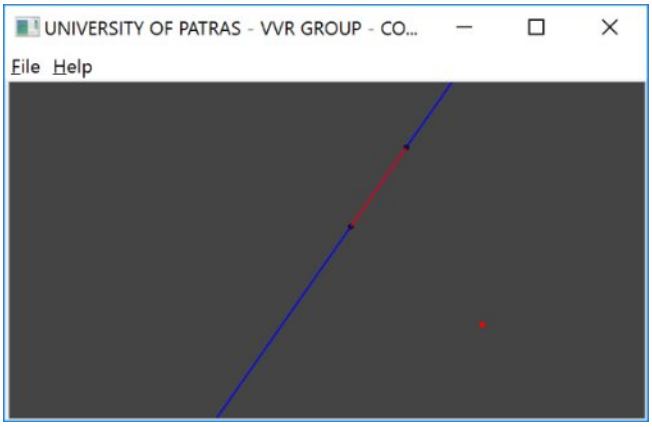
Άρα βάζουμε το σημείο που κλικάρουμε (x,y) στην εξίσωση της ευθείας.

Ανάλογα τώρα με το που κλικάρουμε ή με το που θα σύρουμε το ποντίκι ενώ το έχουμε πατημένο μπορούμε να ξεχωρίσουμε 3 περιπτώσεις.

- Άν (y-y1)-((y1-y2)/(x1-x2))*(x-x1)>0 τότε το σημείο βρίσκεται πάνω από την ευθεία
- Άν (y-y1)-((y1-y2)/(x1-x2))*(x-x1)<0 τότε το σημείο βρίσκεται κάνω από την ευθεία
- Άν (y-y1)-((y1-y2)/(x1-x2))*(x-x1)=0 τότε το σημείο βρίσκεται πάνω από την ευθεία

Επομένως, με αυτόν τον έλεγχο η συνάρτηση μας επιστρέφει ποια από τις πρώτες δύο συνθήκες ισχύει(το αν είναι πάνω στην ευθεία στην προκειμένη περίπτωση δεν μας ενδιαφέρει). Αν ισχύει η πρώτη δίνουμε στο σημείο μας πράσινο χρώμα, αλλιώς αν ισχύει η δεύτερη, κόκκινο.





Task 3:

Βρείτε και απεικονίστε το κυρτό περίβλημα των σημείων που βρίσκονται στην μεταβλητή C2DPointSet m_point_cloud, υλοποιώντας τον αλγόριθμο O(3).

Ο αλγόριθμος που υλοποιούμε είναι ο εξής:

Αρχικά έχουμε όλα τα σημεία n από τα οποία θέλουμε να βγάλουμε το κυρτό περίβλημα.

Για κάθε ζεύγος σημείων (περιορισμός: το ένα σημείο δεν είναι ίδιο με το άλλο) βρίσκουμε την γραμμή που τα ενώνει. Στην συνέχεια, για όλα τα υπόλοιπα (n-2) σημεία ελέγχουμε αν η ευθεία που δημιουργήσαμε τα αφήνει όλα αριστερά. Αν ναι, τότε προσθέτουμε αυτήν την γραμμή στο κυρτό περίβλημα. Επαναλαμβάνουμε την διαδικασία μέχρις ότου να ελεγχθούν όλες οι ευθείες με όλα τα σημεία.

Ο έλεγχος γίνεται με την κλήση μιας καινούριας τώρα συνάρτησης η οποία χρησιμοποιεί την ορίζουσα των δύο διανυσμάτων AB και AP, όπου A και B τα δύο σημεία του ζεύγους που δημιουργούν την ευθεία και P το σημείο που θέλουμε να ελέγξουμε. Η ορίζουσα δύο διανυσμάτων υπολογίζεται ως εξής:

$\det(AB,AP) = (B_x - A_x)(P_y - A_y) - (B_y - A_y)(P_x - A_x)$

Ξανά μπορούμε να διακρίνουμε 3 περιπτώσεις:

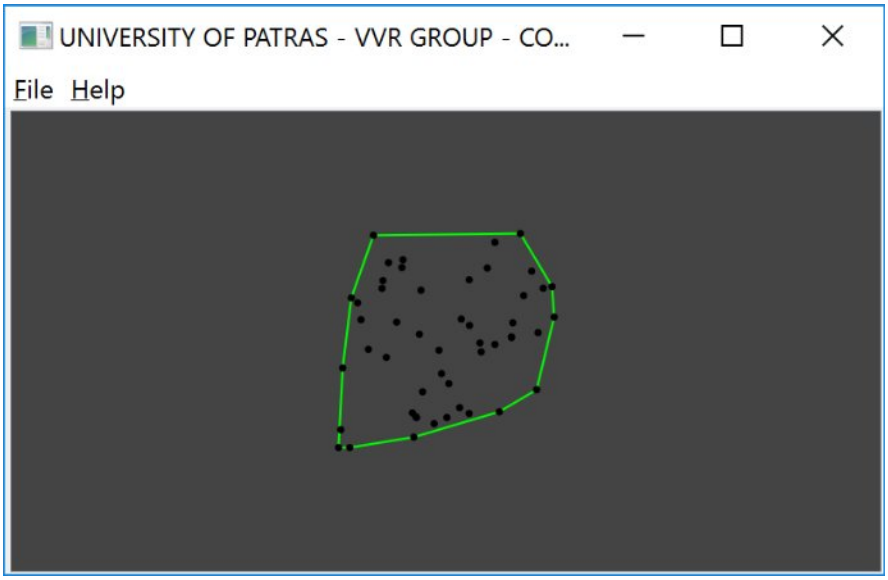
- $\det(AB,AP)>0$ τότε το διάνυσμα AP βρίσκεται αριστερά του AB
- $\det(AB,AP)<0$ τότε το διάνυσμα AP βρίσκεται δεξιά του AB
- $\det(AB,AP)=0$ τότε το διάνυσμα AP βρίσκεται πάνω στο AB

Βέβαια γνωρίζοντας σε ποια μεριά της ευθείας βρίσκεται το κάθε διάνυσμα, στην ίδια μεριά θα βρίσκεται και το σημείο P, το οποίο θέλουμε να ελέγξουμε.

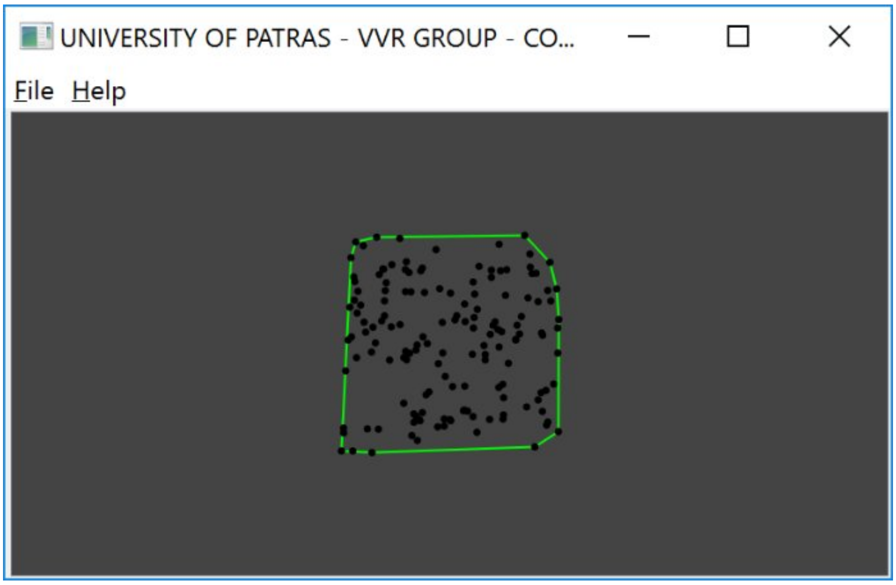
Με αυτό τον τρόπο, σε αντίθεση με την συνάρτηση που υλοποιήσαμε στο task 2, τώρα χρησιμοποιούμε και την φορά της κάθε ευθείας, κάτι που είναι απαραίτητο εφόσον κινούμαστε ωρολογιακά(ή ανθωρολογιακά αν αφήναμε όλα τα σημεία δεξιά, όπου και πάλι θα είχαμε το ίδιο αποτέλεσμα).

(Αν δεν χρησιμοποιούσαμε την κατεύθυνση της ευθείας τότε ελέγχοντας αν η ευθεία του κάθε ζεύγους σημείων αφήνει αριστερά όλα τα υπόλοιπα θα υπολογιζόταν μόνο το άνω κυρτό περίβλημα.)

Κυρτό περίβλημα 50 σημείων:



Κυρτό περίβλημα 150 σημείων:



Άσκηση 1.

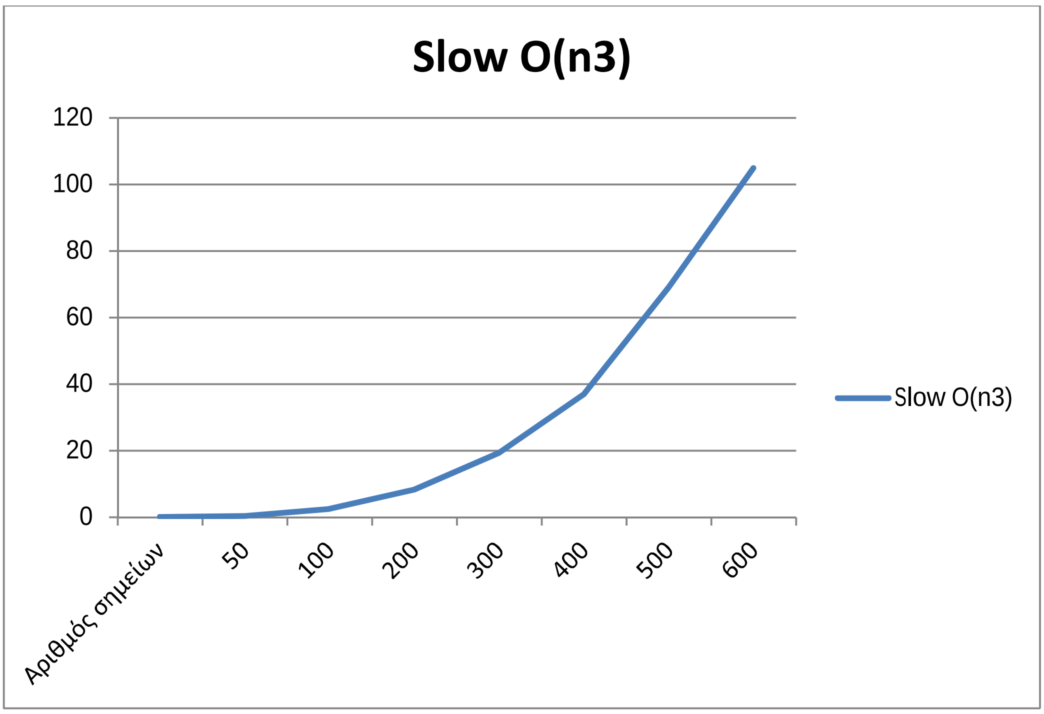
Μετρήστε τον χρόνο εκτέλεσης των δύο αλγορίθμων υπολογισμού κυρτού πολυγώνου, του δικού σας και της GeoLib, για διαφορετικά πλήθη σημείων.

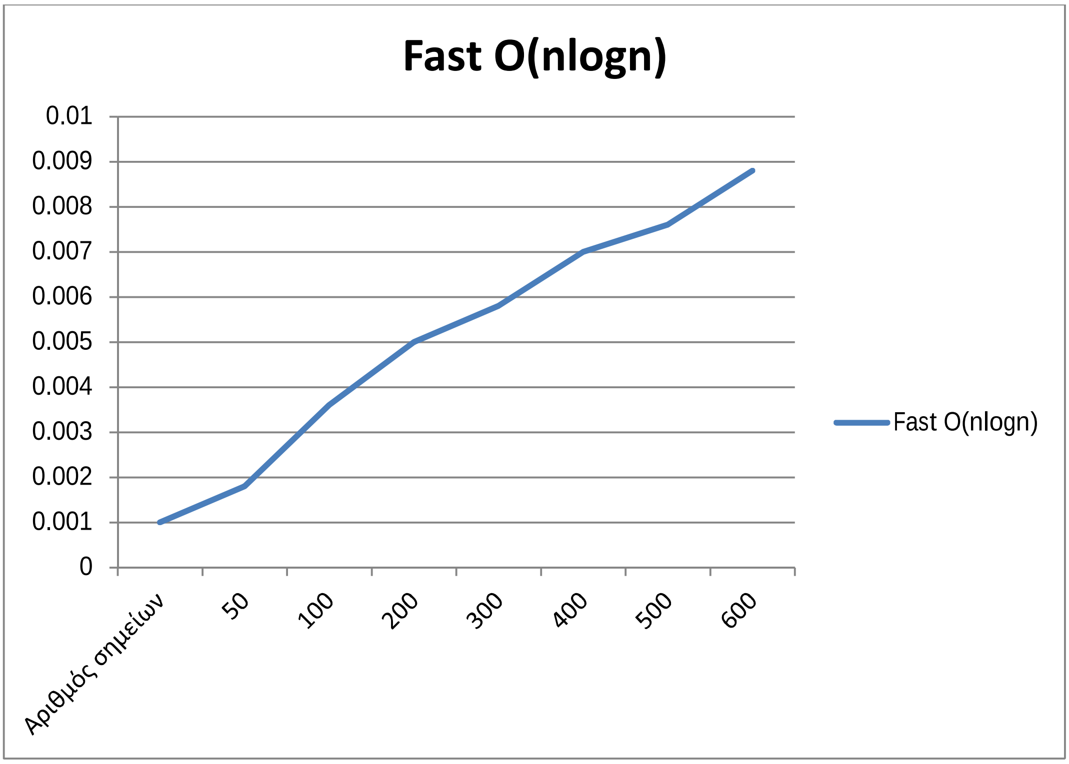
Όλοι οι χρόνοι είναι σε second.

	ConvexHull	
Αριθμός σημείων	Slow O(n3)	Fast O(nlogn)
50	0,046	0,001
100	0,41	0,0018
200	2,511	0,0036
300	8,265	0,005
400	19,344	0,0058
500	36,923	0,007
600	68,999	0,0076
700	104,826	0,0088

Την ConvexHull_Fast την καλούμε περισσότερες από μία φορές και διαιρούμε το τελικό χρόνο με τον αριθμό των φορών που την καλέσαμε για να καταφέρουμε να έχουμε καλύτερη ακρίβεια μέτρησης χρόνου.

a. Απεικονίστε σε γραφική παράσταση τον χρόνο εκτέλεσης συναρτήσει του πλήθους των σημείων και για τους δύο αλγόριθμους.





b. Είναι κατάλληλη η χρήση γραμμικής κλίμακας αξόνων για την σύγκριση αυτών των δύο αλγορίθμων;

Όπως βλέπουμε και από τα διαγράμματα για το ίδιο πλήθος σημείων, στην περίπτωση του ConvexHull_slow ο χρόνος που απαιτείται είναι πολλές τάξεις μεγέθους παραπάνω από αυτόν του ConvexHull_fast. Άρα, για να συγκρίνουμε τους δύο αλγορίθμους η χρήση γραμμικής κλίμακας αξόνων δεν συνιστάται.

Άσκηση 2.

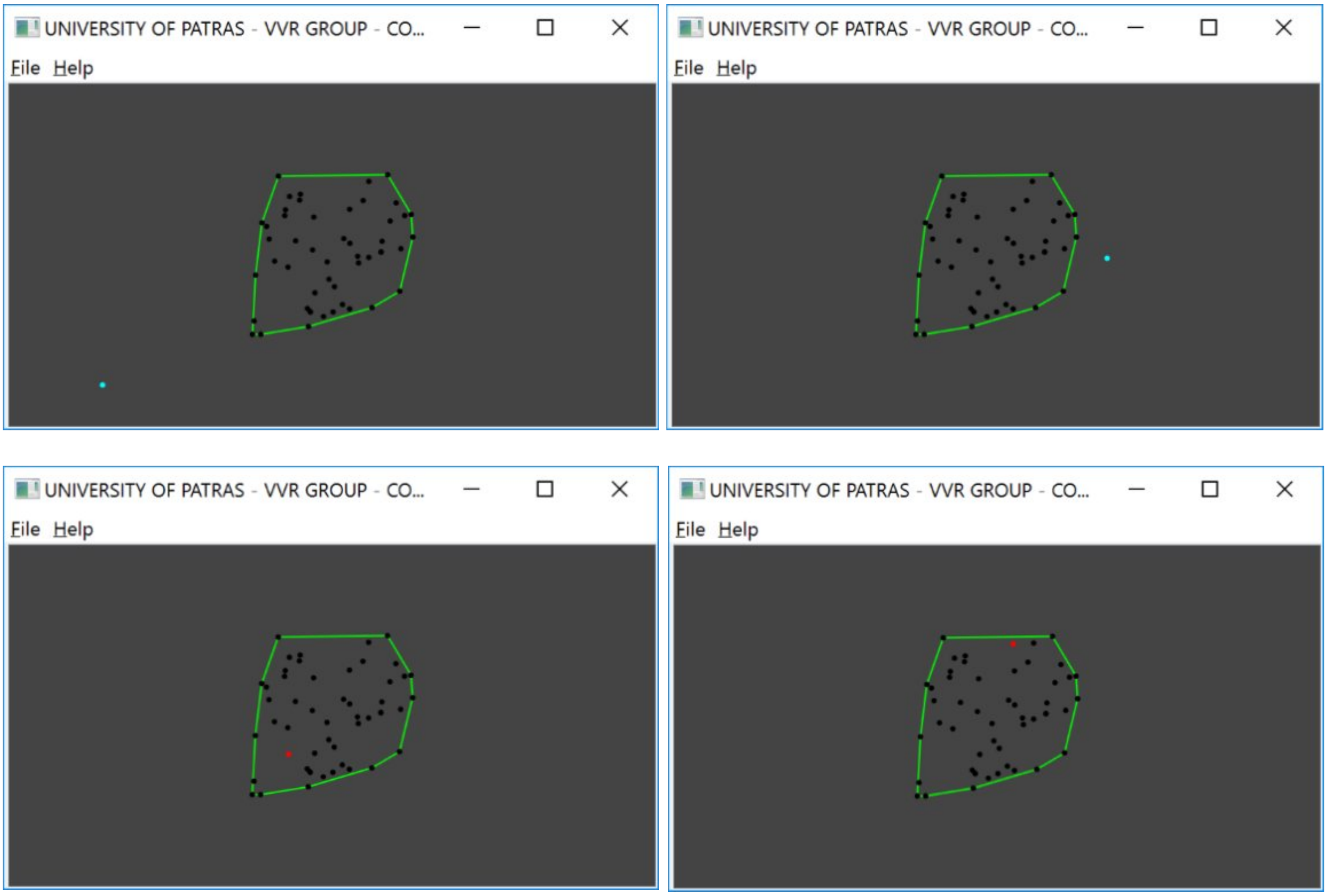
Σχεδιάστε σημείο που ακολουθεί την κίνηση του mouse.

a. Διαφοροποιείστε τα σημεία που βρίσκονται εντός/εκτός του κυρτού πολυγώνου με χρήση διαφορετικού χρώματος.

Λογική αλγορίθμου:

Χρησιμοποιούμε τον έλεγχο (με την ορίζουσα) όπως αυτός περιγράφηκε στα προηγούμενα ερωτήματα. Για να είναι λοιπόν ένα σημείο εντός του κυρτού πολυγώνου πρέπει να βρίσκεται δεξιά από <u>όλες</u> τις ευθείες που δημιουργούνται ωρολογιακά από το κάθε σημείο του κυρτού πολυγώνου με το ακριβώς προηγούμενό του. Δηλαδή για κάθε ένα απο αυτά τα διανύσματα που σχηματίζουν αυτά τα σημεία(εστω A και B δύο από αυτα και P το σημείο που κάνουμε mouse click/pressed) πρέπει $det(AB,AP)= (B_x-A_x)(P_y-A_y)-(B_y-A_y)(P_x-A_x) < \mathbf{0}$. Αν ισχύει έστω και για ένα διάνυσμα $det(AB,AP)>0$, τότε το σημείο P βρίσκεται απ' έξω.

Χρωματίζουμε ,λοιπόν ,το σημείο που βρίσκεται εντός του κυρτού πολυγώνου με κόκκινο χρώμα και τα εκτός με γαλάζιο, όπως φαίνεται παρακάτω.



b. Στην περίπτωση σημείου εκτός κυρτού πολυγώνου:

Σχεδιάστε τα ευθύγραμμα τμήματα από το σημείο έως τις ακραίες κορυφές του πολυγώνου που είναι ορατές και επιπλέον χρωματίστε τις ακμές του πολυγώνου που είναι ορατές από το σημείο.

- Λογική αλγορίθμου χρωματισμού των ακμών που είναι ορατές από το σημείο:

Κάνοντας τον προηγούμενο έλεγχο, αν είναι μέσα στο πολύγωνο το σημείο ή οχι, βρίσκουμε ταυτόχρονα από ποιες ευθείες το σημείο που κάνουμε click/pressed βρίσκεται αριστερά (αφού κινούμαστε ωρολογιακά), δηλάδη έξω απ΄το πολύγωνο. Σε αυτήν την περίπτωση ισχύει $\det(AB,AP)>0$. Αυτές είναι και οι ευθείες που είναι ορατές απο το σημείο,τις οποίες και εμφανίζουμε με χρώμα κόκκινο.

- Λογική αλγορίθμου σχεδιασμού των ευθυγράμμων τμημάτων από το σημείο έως τις ακραίες κορυφές του πολυγώνου που είναι ορατές:

Στα προηγούμενα ερώτημα τα σημεία που απαρτίζουν το κυρτό πολύγωνο είναι αποθηκευμένα σε C2DPointSet. Για την επίλυση αυτού του ερωτήματος , λοιπόν, διακρίνουμε δύο περιπτώσεις. Στην πρώτη τα ευθ. τμήματα που βλέπουν το σημείο σχηματίζονται από οποιαδήποτε σημεία του PointSet εκτός του 1ου <u>ή</u> του τελευταίου. Στην δεύτερη περίπτωση σχηματίζονται από οποιαδήποτε σημεία του PointSet και <u>περιλαμβάνουν</u> <u>και</u> το 1ο <u>και</u> το τελευταίου του στοιχείο(έχουμε θέμα ασυνέχειας δηλαδή).

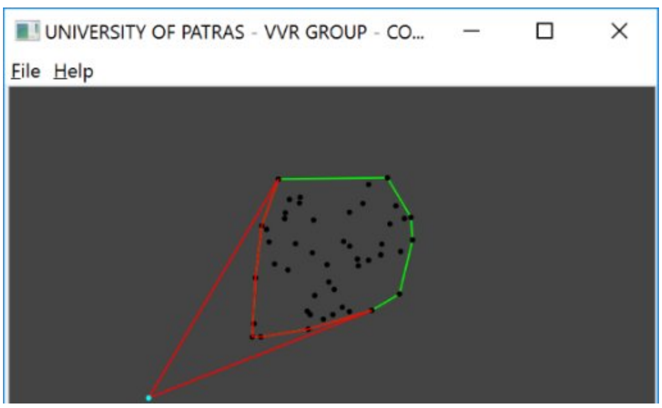
Όλα τα ευθύγραμμα τμήματα που βλέπουν το σημείο αποθηκεύονται σε C2DLineSet, πριν περαστούν στο canvas.

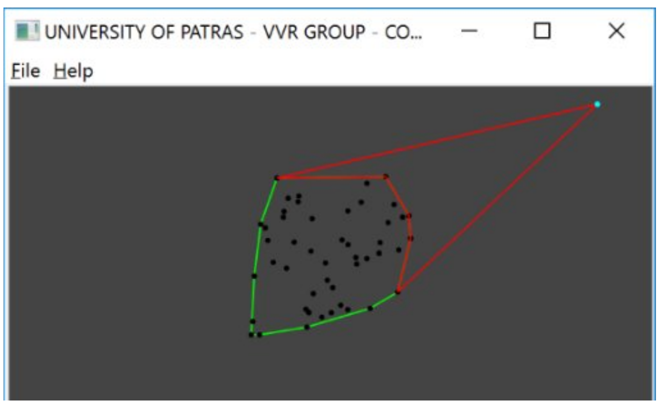
1η περίπτωση:

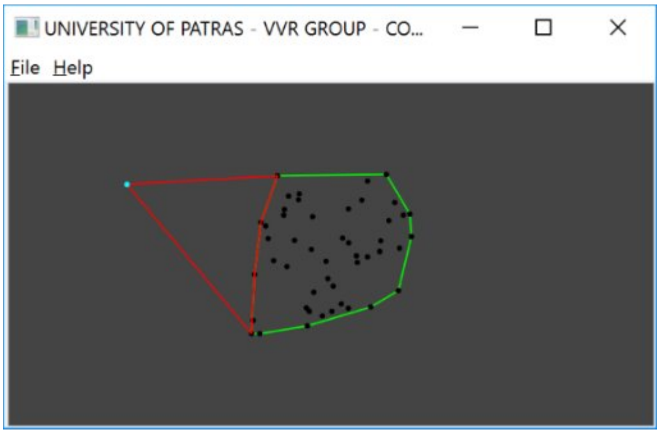
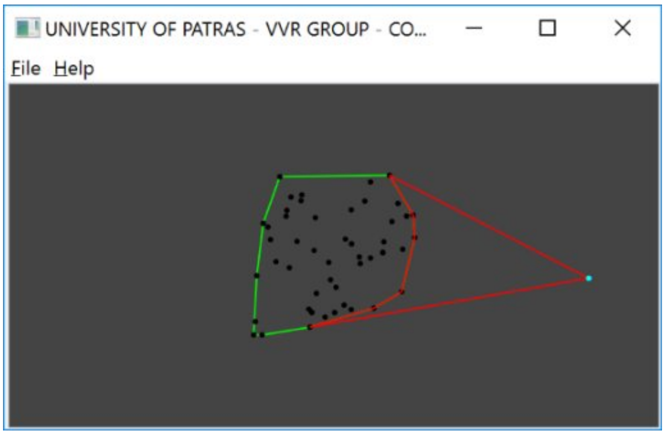
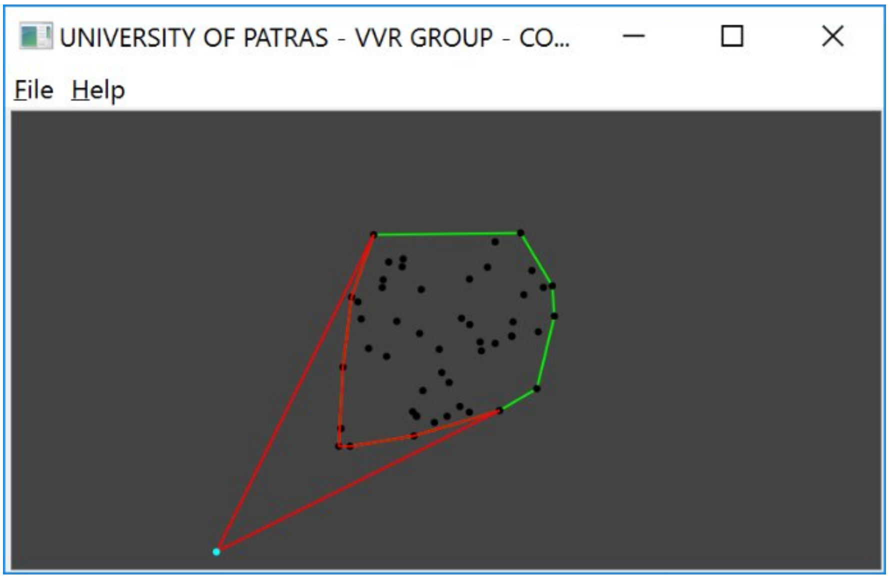
Δεν έχουμε θέματα ασυνέχειας και άρα τα ευθ. τμήματα είναι αποθηκευμένα το ένα μετά το άλλο με την σειρά. Έχοντας τα έτσι αποθηκευμένα μπορούμε εύκολα να πάμε στο πρώτο από αυτά και να δημιουργήσουμε ένα ευθύγραμμο τμήμα από το σημείο που κάναμε mouse click/pressed στο σημείο που αρχίζει αυτό το ευθύγραμμο τμήμα. Αντίστοιχα στο τελευταίο, δημιουργούμε ένα ευθύγραμμο τμήμα από το σημείο που κάναμε mouse click/pressed στο σημείο που τελειώνει αυτό.

2η περίπτωση:

Τώρα που και το πρώτο και το τελευταίο σημείο του PointSet αντιστοιχεί σε ευθύγραμμο τμήμα που είναι ορατό από το σημείο που κάναμε click δεν μπορούμε να χρησιμοποιήσουμε την παραπάνω μεθοδολογία, αφού κάποια από τα ενδιάμεσα ευθ. τμήματα δεν είναι ορατά από το σημείο μας. Όμως, μιας και τα σημεία από τα οποία απαρτίζονται τα ευθ. τμήματα είναι πάντα το ένα μετά το άλλο βρίσκουμε εκείνο το σημείο από το οποίο ξεκινά το πρώτο ωρολογιακά ευθ. τμήμα που είναι ορατό από το σημείο μας. Ανατρέχοντας στο LineSet(με τα αποθηκευμένα ευθ. τμήματα που είναι ορατά) βρίσκουμε το ακριβώς προηγούμενό του ευθ. τμήμα από το οποίο παίρνουμε το σημείο που τελειώνει. Τέλος σχεδιάζουμε ευθύγραμμα τμήματα που ενώνει το σημείο που κάναμε κλικ με τα δύο σημεία που βρήκαμε παραπάνω.






UNIVERSITY OF PATRAS - VVR GROUP - CO...
File Help

UNIVERSITY OF PATRAS - VVR GROUP - CO...
File Help

UNIVERSITY OF PATRAS - VVR GROUP - CO...
File Help