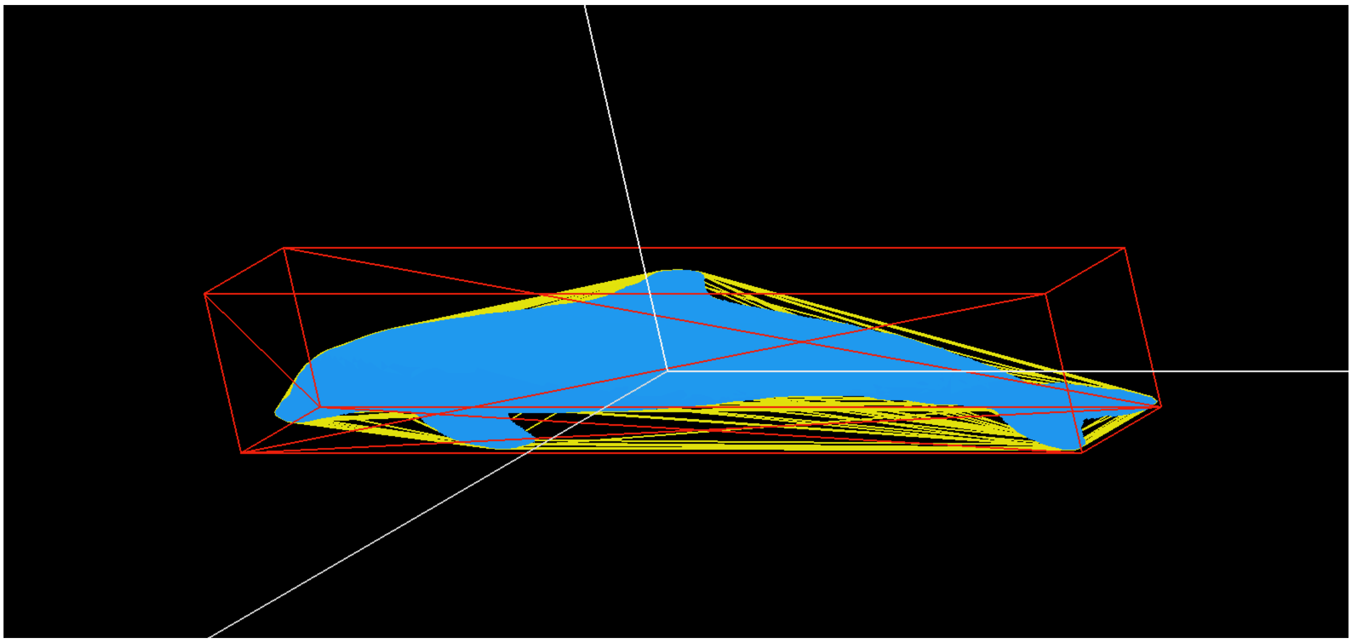


Υπολογιστική Γεωμετρία & 3Δ Μοντελοποίηση

Μάριος-Φώτιος Μπίκος

ΑΜ: 7323

Πανεπιστήμιο Πατρών

2012-2013

8ο Εξάμηνο

ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ 1

Υπολογιστική Γεωμετρία & 3Δ Μοντελοποίηση

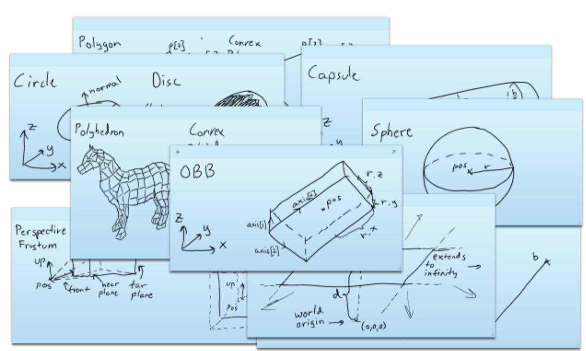
2012-2013

MODEL3D_2

Εισαγωγή

Η εργασία αυτή αναπτύχθηκε στα πλαίσια του μαθήματος Υπολογιστική Γεωμετρία & 3Δ Μοντελοποίηση του τμήματος Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών την περίοδο 2012-2013.
Στόχος ήταν η ανάπτυξη αλγορίθμων για την κατανόηση εννοιών όπως το κυρτό περίβλημα και ο χάρτης αποστάσεων σε 2 και 3 διαστάσεις και η ανίχνευση σύγκρουσης σε 3 διαστάσεις.
Τα μοντέλα που μου ανατέθηκαν ήταν τα Model_2d_test.poly και 3D_2.obj αρχεία.

Περιβάλλον ανάπτυξης της εφαρμογής: Γλώσσα προγραμματισμού και εργαλεία ανάπτυξης



Για την απαλλακτική αυτή εργασία χρησιμοποιήθηκαν οι παρακάτω βιβλιοθήκες σχετικά με την υπολογιστική γεωμετρία:

- Geolib (http://www.geolib.co.uk/)
- MathGeoLib (http://clb.demon.fi/MathGeoLib/nightly/)

Η 1η βιβλιοθήκη χρησιμοποιήθηκε στο 1ο ερώτημα της εργασίας όπου το μοντέλο μας ήταν στις 2 διαστάσεις, ενώ στο 2ο και 3ο ερώτημα χρησιμοποιήσαμε την βιβλιοθήκη MathGeoLib έτσι ώστε να εκμεταλλευτούμε τις μεθόδους που δίνονται για τις 3 διαστάσεις καθώς και για να ορίσουμε στοιχεία στις 3 διαστάσεις όπως σφαίρες,πολύεδρα,σημεία 3 διαστάσεων κλπ.

Αξίζει να αναφερθεί ότι παρά το γεγονός ότι η MathGeoLib προσφέρει έτοιμες μεθόδους για την εύρεση κυρτού περιβλήματος 3 διαστάσεων, ΔΕΝ πήραμε τίποτα έτοιμο και υλοποιήσαμε τους αλγορίθμους από την αρχή.

Η εργασία υλοποιήθηκε με χρήση OpenGL/C++ στο Visual Studio 2012.Κύριος στόχος μου ήταν η λειτουργικότητα του project και η αλληλεπίδραση με τον χρήστη ώστε να αντιληφθεί τις έννοιες που ζητούνται από τα ερωτήματα.Αφού συνέβη αυτό προσπάθησα να βελτιώσω ορισμένα σημεία και να απεικονίσω συγκεκριμένες πτυχές των προβλημάτων που επιλύθηκαν.

Το αρχικό πλάνο δεν περιελάμβανε την δημιουργία animation και κίνησης των αντικειμένων, ωστόσο για να κατανοήσει καλύτερα ο χρήστης ένα αντικείμενο στο χώρο των 3 διαστάσεων, κάτι τέτοιο κρίθηκε τελικά απαραίτητο.

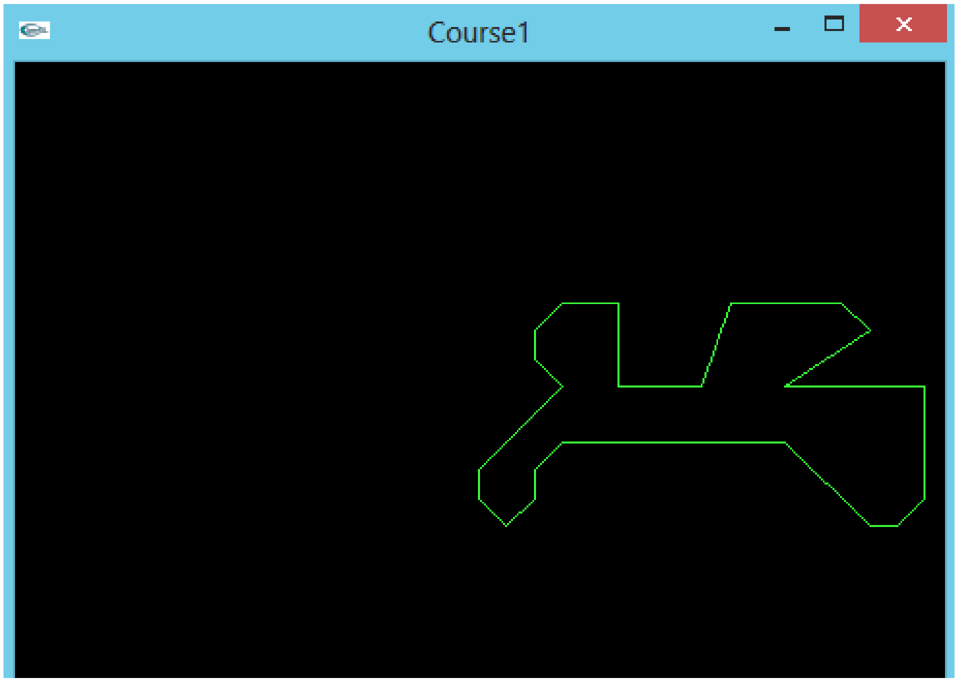
Χρησιμοποίησα πολλά πράγματα τα οποία είχα υλοποιήσει στις προαιρετικές εργασίες του μαθήματος κατά τη διάρκεια του εξαμήνου και φάνηκαν ιδιαίτερα χρήσιμες.

ΕΡΩΤΗΜΑ 1Ο

Υπολογίστε το κυρτό περίβλημα του Model_2D. Στη συνέχεια υπολογίστε το χάρτη αποστάσεων του κυρτού περιβλήματος από το υποκείμενο μοντέλο και αποθηκεύστε το σε κατάλληλη δομή. Βρείτε ένα τρόπο να απεικονίσετε το χάρτη αποστάσεων (π.χ. χρωματικά ή γεωμετρικά).

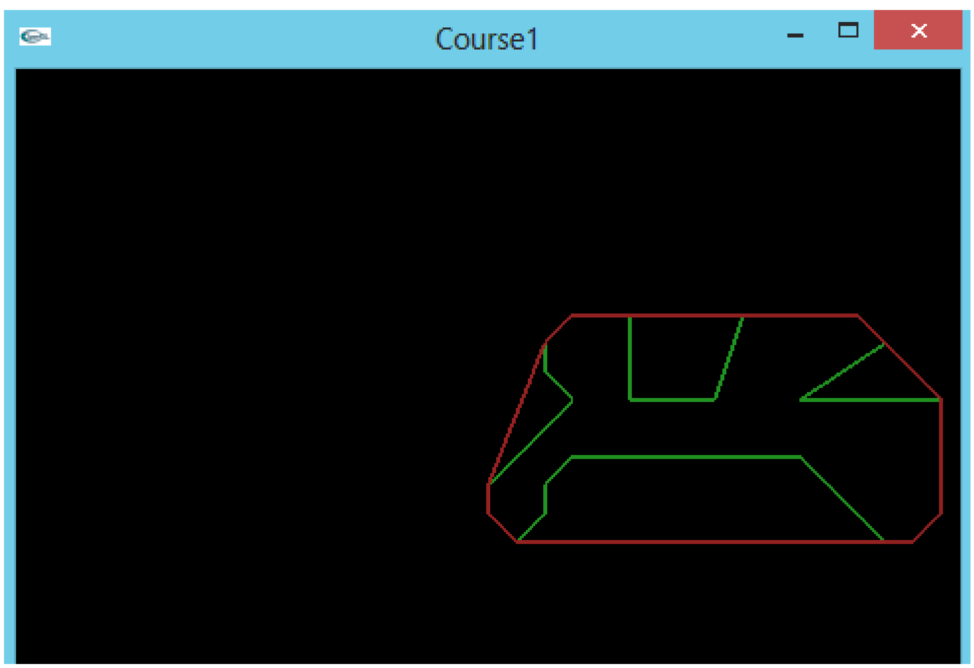
Για το παραπάνω ερώτημα εργαστήκα ως εξής:

Αρχικά, ήταν απαραίτητο να διαβάσουμε τα σημεία που συνθέτουν το αρχικό μας μοντέλο.Μέσα από την κατάλληλη διαδικασία που χρησιμοποιεί η C++ διαβάσαμε τα σημεία από το αρχείο model_2d_test.poly και τα βάλαμε σε ένα C2DPointSet. Έπειτα, εύκολα δημιουργήσαμε το αντίστοιχο πολύγωνο από τα σημεία αυτά.



Με πράσινο χρώμα απεικονίζεται το πολύγωνο μας

Έπειτα υπολόγισα το κυρτό περίβλημα εύκολα μέσα από την αντίστοιχη μέθοδο της βιβλιοθήκης μας, μιας και όπως μας είχε επισημάνει ο κ.Μουστάκας στις 2 διαστάσεις μπορούμε να πάρουμε αμέσως την μέθοδο αυτή, και την είχαμε υλοποιήσει και στις προαιρετικές εργασίες.

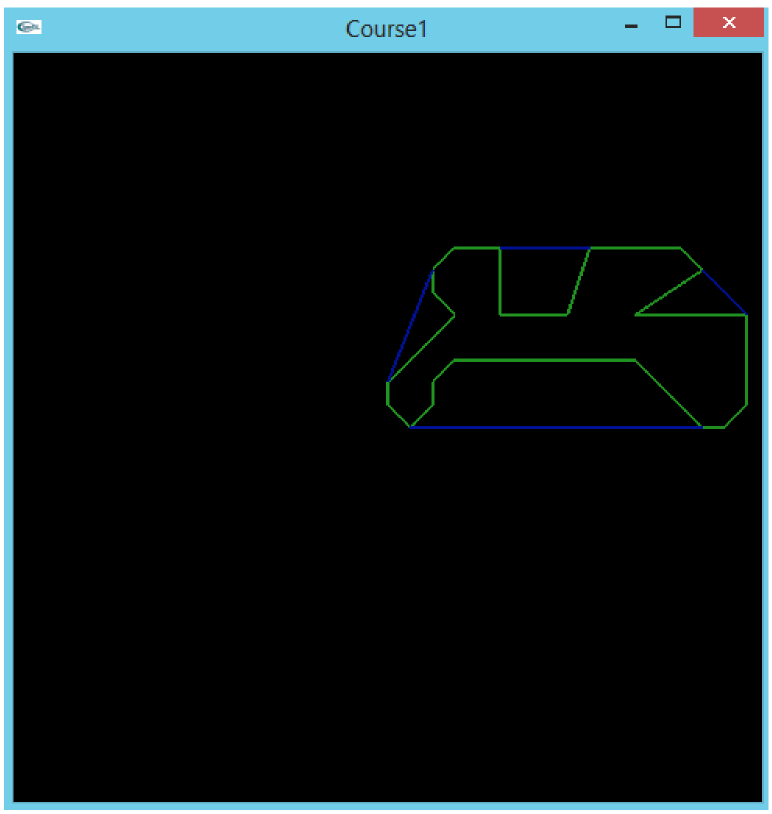


Με κόκκινο χρώμα απεικονίζεται το κυρτό περίβλημα του πολυγώνου μας

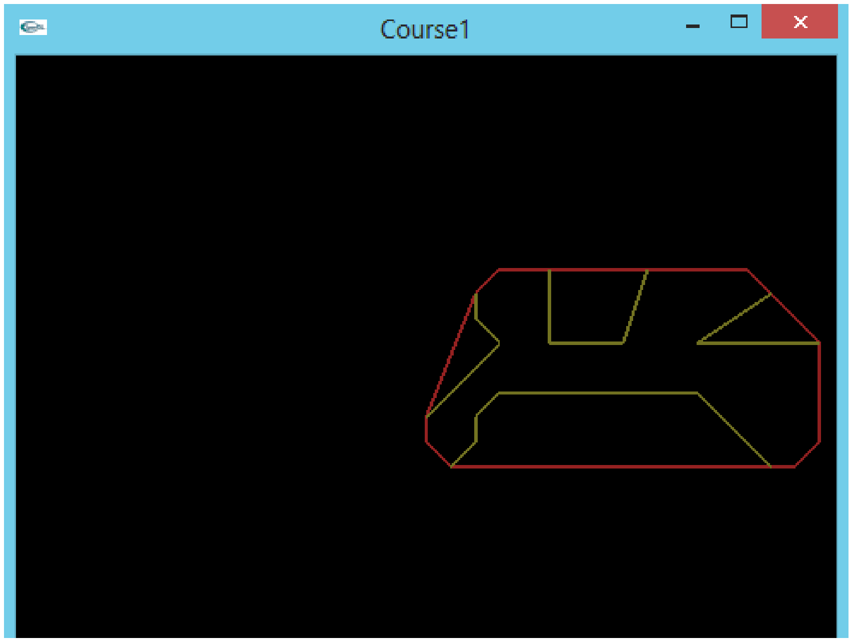
Το επόμενο βήμα προκειμένου να φτάσουμε στην δημιουργία του χάρτη αποστάσεων ήταν να βρούμε 2 σειρές από ευθύγραμμα τμήματα.Τα ευθύγραμμα τμήματα του πολυγώνου που είναι μόνο μέρη του πολυγώνου και όχι του convex hull,δηλαδή ουσιαστικά όλα τα ευθύγραμμα τμήματα στο εσωτερικό του κυρτού περιβλήματος και όλα τα ευθύγραμμα τμήματα τα οποία είναι μέρη του κυρτού περιβλήματος αλλά δεν είναι ευθ. Τμήματα του πολυγώνου.

Για να γίνει όλο αυτό χρειάστηκε να σπάσουμε το πολύγωνο σε ευθ. τμήματα, αλλά και να σπάσουμε το κυρτό περίβλημα σε ευθ. Τμήματα.Επειδή μάλιστα η βιβλιοθήκη δεν μας επιτρέπει να δούμε αν 2 ευθ. τμήματα είναι ίδια, χρειάστηκε να συγκρίνουμε τα ευθ. τμήματα με βάση το αρχικό και τελικό τους σημείο.Δηλαδή ενα ευθ. τμήμα είναι ίδιο με ένα άλλο αν το αρχικό σημείο του πρώτου είναι ίδιο με το αρχικό του δεύτερου και ομοίως τα τελικά σημεία, είτε αν το τελικό σημείο του πρώτου τμήματος είναι ίδιο με το αρχικό σημείο του δεύτερου τμήματος και το αρχικό σημείο του πρώτου τμήματος είναι ίδιο με το τελικό του δεύτερου.

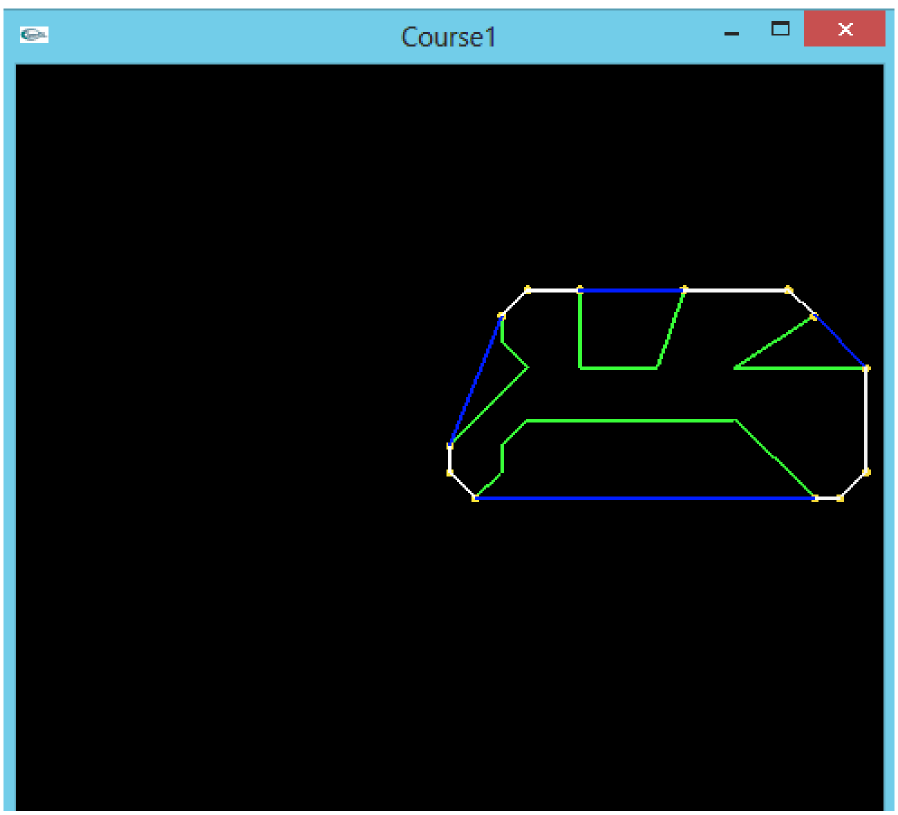
Με αυτή τη λογική βρήκαμε τις 2 σειρές ευθύγραμμων τμημάτων, που θα μας βοηθήσουν στην επίλυση του προβλήματος, συγκρίνοντας κάθε ευθ. τμήμα του πολυγώνου με όλα τα ευθ. τμήματα του κυρτού περιβλήματος και ομοίως για το 2^{o} σετ.



Με μπλε απεικονίζονται τα ευθ. τμήματα που βρίσκονται στο κυρτό περίβλημα αλλά δεν είναι μέρη του πολυγώνου.

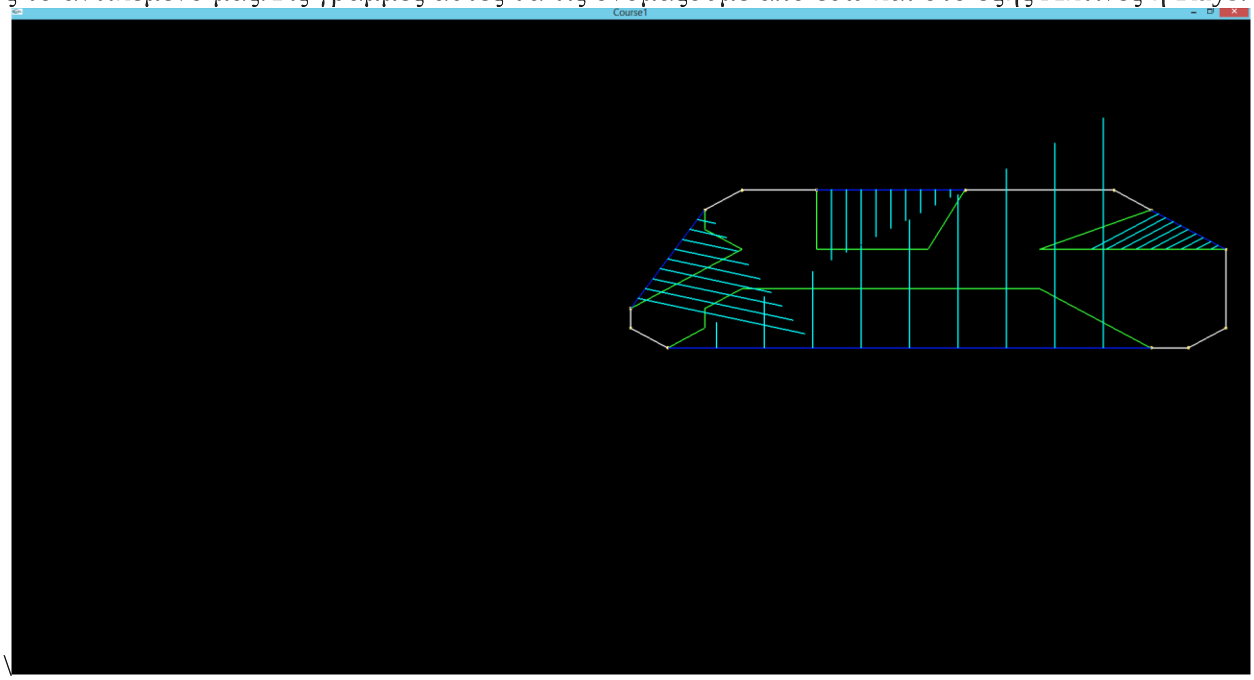


Με χρυσό χρώμα απεικονίζονται τα ευθύγραμμα τμήματα που είναι μέρη του πολυγώνου μας, αλλά δεν είναι μέρη του κυρτού περιβλήματος του πολυγώνου μας



Εδώ απεικονίζεται καλύτερα το μοντέλο μας.Βλέπετε με μπλέ χρώμα τα ευθ. τμήματα του κυρτού περιβλήματος που δεν είναι μέρος του πολυγώνου,με πράσινο χρώμα τα εσωτερικά ευθ. τμήματα και με λευκό χρώμα τα ευθ. τμήματα που ανήκουν και στο κυρτό περίβλημα και στο πολύγωνο μας.

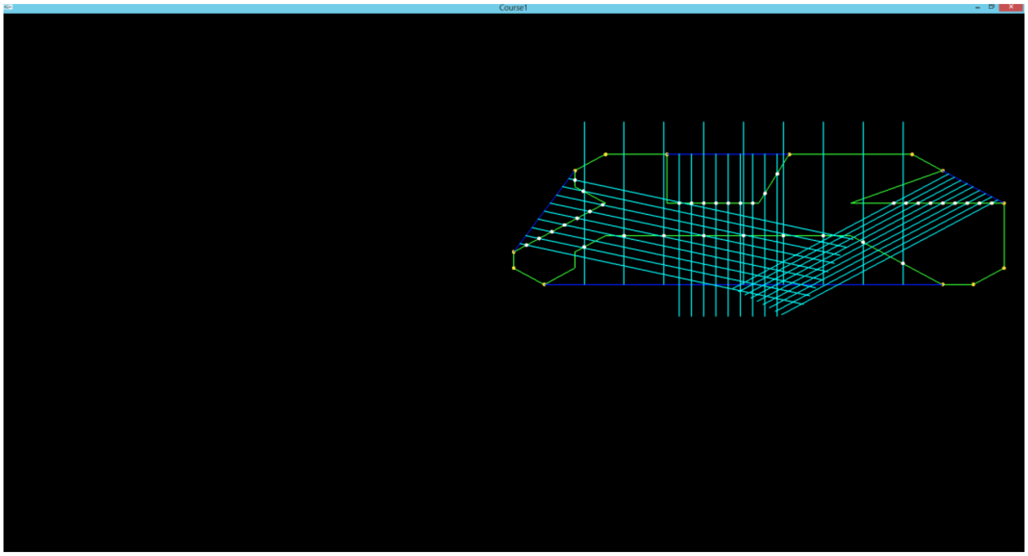
Από εκεί και έπειτα προκειμένου να βρούμε το χάρτη αποστάσεων κάνουμε τα εξής:
Παίρνουμε κάθε ευθ. τμήμα που είναι μέρος του κυρτού περιβλήματος, αλλά όχι του πολυγώνου μας και το σπάμε σε διαδοχικά ευθύγραμμα τμήματα με πολύ μικρό βήμα,δηλαδή σε κάθε βήμα προχωράμε πχ στο 0.02% του μήκους του ευθ. τμήματος και από εκεί ορίζουμε ευθ. τμήμα με αρχή το σημείο εκείνο και τέλος το τελικό σημείο του ευθ. τμήματος.
Έπειτα στρέφουμε το κάθε τμήμα που σπάσαμε προηγουμένως κατά 90 μοίρες έτσι ώστε να κοιτάνε προς το αντικέμενο μας.Τις γραμμές αυτές θα τις ονομάζουμε από εδώ και στο εξής Ακτίνες ή Rays.



Εδώ με γαλάζιο χρώμα απεικονίζονται οι «ακτίνες» που ρίχνουν τα ευθ. τμήματα προς το αντικείμενο μας.

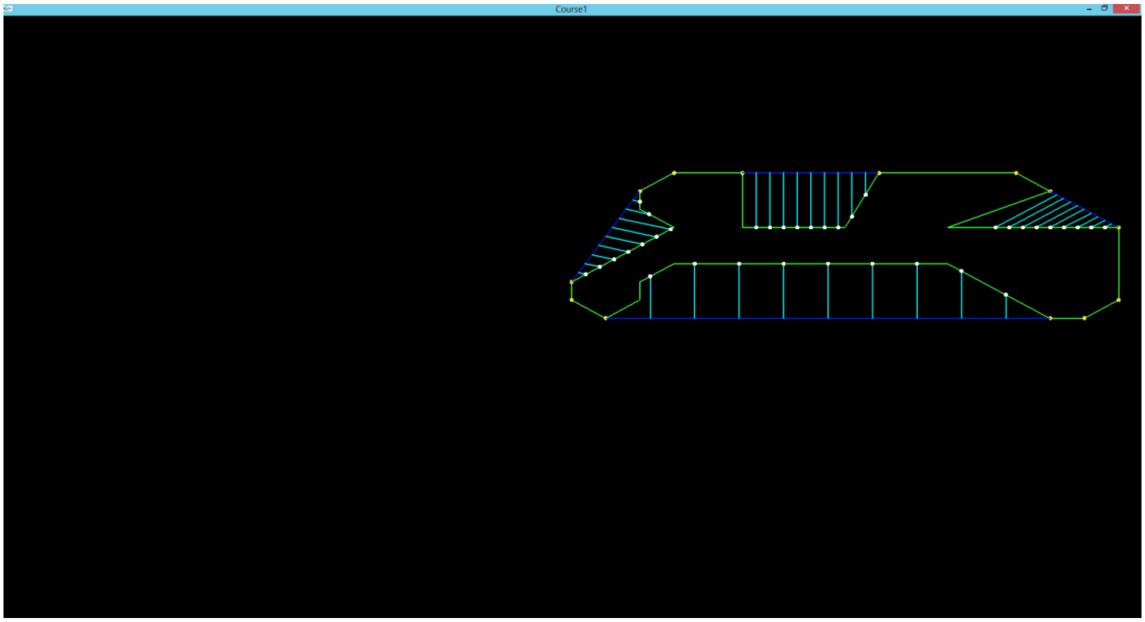
Όπως φαίνεται και στην εικόνα μας, η κάθε ακτίνα δεν είναι το ίδιο μήκος με τη διπλανή της.Μάλιστα κάποιες είναι τόσο μικρές που δεν τέμνουν καν το αντικείμενο μας.Οπότε θα ορίσουμε το μήκος κάθε ακτίνας να είναι αρκετά μεγάλο ώστε να τέμνει το αντικείμενο μας.Γενικά γνωρίζουμε ότι τα αντικείμενα μας θα βρίσκονται στα όρια του αρχικού ορθογωνίου μας δηλαδή από (-50,50) σε x,y συντεταγμένες.
Αρα πχ εδώ θα έπρεπε να βάλουμε μήκος 100 ώστε να είμαστε σίγουροι.

Στη συνέχεια ελέγχουμε για κάθε τέτοια ακτίνα, όλα τα εσωτερικά ευθ. τμήματα δηλαδή όλα ευθ. τμήματα που ανήκουν στο πολύγωνο μας και όχι στο κυρτό περίβλημα και βρίσκουμε σημεία τομής.Όπως γίνεται κατανοητό κάτι τέτοιο θα επιφέρει πολλά σημεία τομής κάθε ακτίνας.Ωστόσο εμείς μέσα από κατάλληλα διαμορφωμένο κώδικα πήραμε ως σημείο τομής αυτό που βρίσκεται πιο κοντά στο αρχικό σημείο της ακτίνας μας και το οποίο είναι το σωστό.



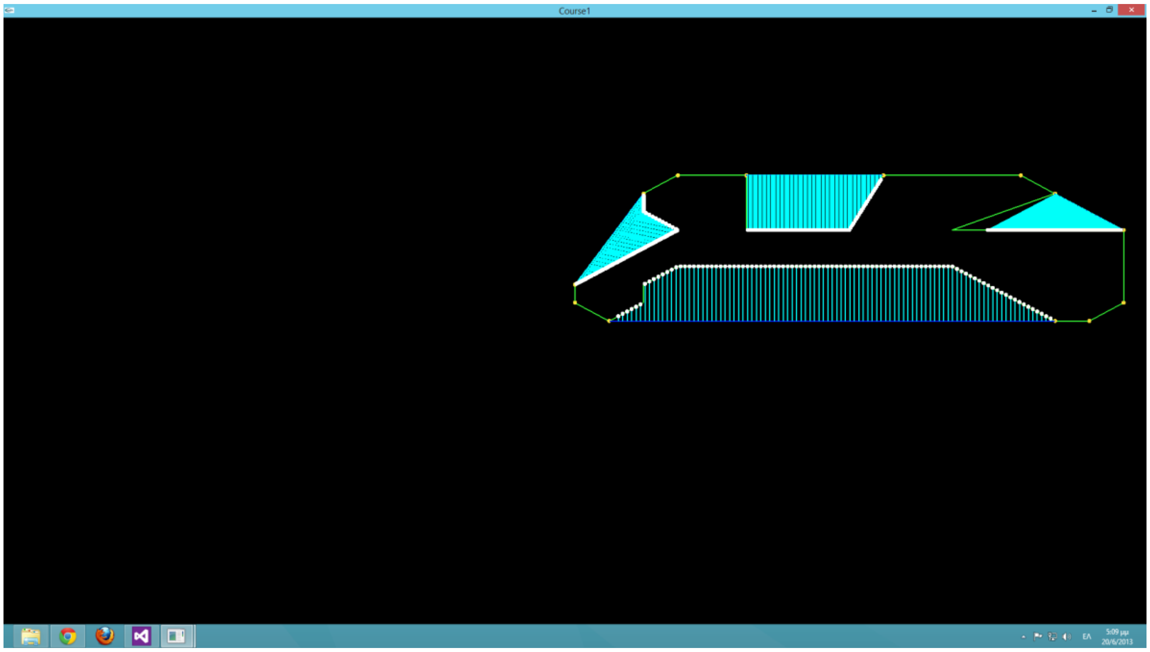
Στο σχήμα φαίνονται τα ομοιόμορφα rays με γαλάζιο χρώμα(με μήκος όχι 100 αλλά λίγο μικρότερο για λόγους εμφάνισης) και με λευκό χρώμα τα σωστά σημεία τομής με το αντικείμενο μας.

Στο παραπάνω σχήμα βλέπουμε ωστόσο ότι παρά το γεγονός ότι βρήκαμε τα σημεία τομής με το αντικείμενο οι ακτίνες συνεχίζουν και μετά την πρόσκρουση.Οπότε εύκολα μετασχηματίζουμε τις ακτίνες ώστε σαν τελικό σημείο να έχουν τα σημεία τομής.



Το σχήμα μας με το σωστό μήκος ακτίνων με τελικό σημείο το σημείο της αντίστοιχης τομής

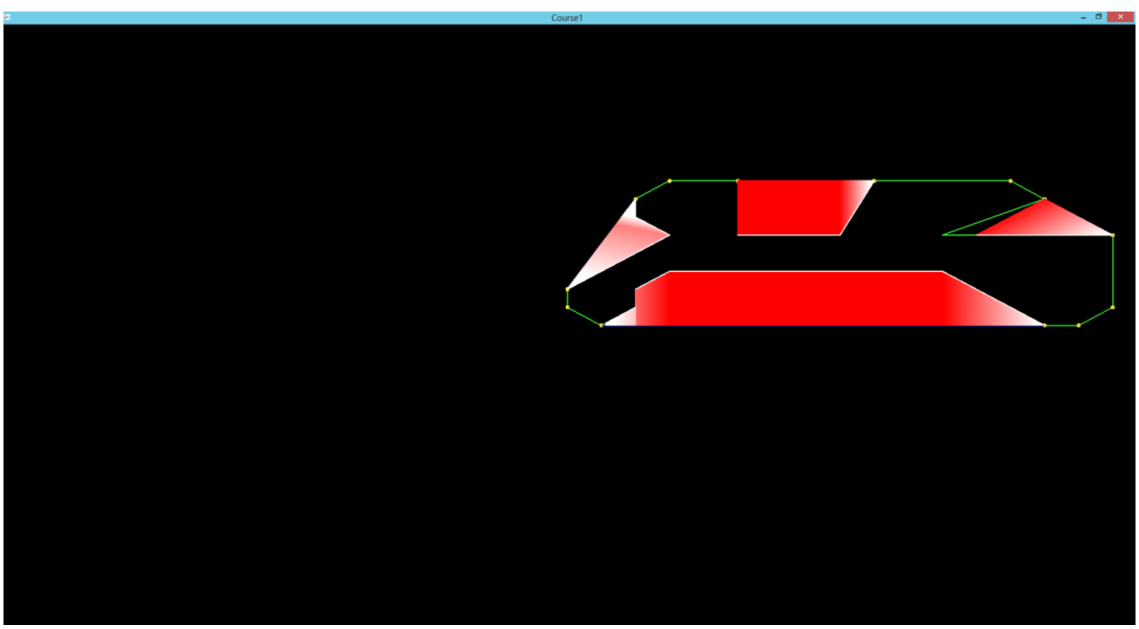
Όπως φαίνεται και από τα σχήματα μας ως τώρα, η ακρίβεια που πήραμε για να σχεδιάσουμε τις ακτίνες δεν επαρκεί για τη δημιουργία του χάρτη αποστάσεων.Οπότε θα πρέπει να πάρουμε πολύ μεγαλύτερη ακρίβεια δηλαδή πολύ περισσότερες ακτίνες να φεύγουν από κάθε μπλε πλευρά. Αυτό το καταφέρνουμε πολύ εύκολα γυρίζοντας πίσω στο κομμάτι κώδικα που πήραμε για τα διαδοχικά ευθ. τμήματα και παίρνοντας μικρότερο βήμα.



Το σχήμα με περισσότερες ακτίνες από πριν

Στο τελικό στάδιο και προκειμένου να πετύχουμε τον σωστό χάρτη αποστάσεων χρωματικά, θα πάρουμε πολύ μεγάλη ακρίβεια.Έπειτα θα κάνουμε το εξής:
Θα ελέγξουμε το σύνολο όλων των ακτίνων και θα βρούμε αυτή την ακτίνα με το μεγαλύτερο μήκος, καθώς και αυτή με το μικρότερο μήκος(που προφανώς θα είναι πολύ κοντά στο 0)
Επομένως θα ζωγραφίσουμε τις ακτίνες έτσι ώστε όσες ακτίνες είναι πιο κοντά στο μέγιστο μήκος ακτίνας να είναι πιο κόκκινες και όσο το μήκος ακτίνας τείνει προς το 0 τόσο να γίνεται και πιο λευκή η ακτίνα.Για παράδειγμα αν μία ακτίνα έχει μήκος 5 και το μέγιστο μήκος ακτίνας είναι 10, τότε προφανώς η ακτίνα μας θα είναι μία μίξη με 50% κόκκινο και 50% λευκό, δηλαδή ροζ.

Έτσι σχηματίζουμε έναν χάρτη αποστάσεων πολυγώνου 2 διαστάσεων!



Ο χάρτης αποστάσεων του μοντέλου μας.Όσο πιο κόκκινο το χρώμα τόσο πιο πολύ απέχει το κυρτό περίβλημα από το αντικείμενο μας και όσο πιο λευκό τόσο κοντά είναι το αντικείμενο με το πολύγωνο μας.

Τέλος, αξίζει να αναφερθεί ότι με το πάτημα των πλήκτρων I & O ο χρήστης μπορεί να κάνει zoom in και zoom out αντίστοιχα για να δει καλύτερα το αντικείμενο και να πατήσει SPACEBAR για να πάρει το αντικείμενο την αρχική του θέση

ΕΡΩΤΗΜΑ 2^{O}

Υπολογίστε, υλοποιώντας τον αυξητικό αλγόριθμο, το κυρτό περίβλημα του Model_3D. Στη συνέχεια υπολογίστε το χάρτη αποστάσεων του κυρτού περιβλήματος από το υποκείμενο μοντέλο και αποθηκεύστε το σε κατλάλληλη δομή. Βρείτε ένα τρόπο να απεικονίστε το χάρτη αποστάσεων (π.χ. χρωματικά ή γεωμετρικά).

Το ερώτημα αυτό κρίθηκε πολύ πιο απαιτητικό από το προηγούμενο.Συγκεκριμένα συναντήσαμε αρκετά μεγάλη δυσκολία στην υλοποίηση του αυξητικού αλγορίθμου του κυρτού περιβλήματος στις 3 διαστάσεις, όχι μόνο διότι ήταν το 1^{ο} πρόβλημα που λύναμε στις 3 διαστάσεις, αλλα και γιατί είναι αρκετά δυσκολότερο κάποιος να κατανοήσει ορισμένες έννοιες σε επίπεδο 3 διαστάσεων.

Για το πρόβλημα αυτό χρειάστηκε να εισάγουμε και να χρησιμοποιήσουμε την βιβλιοθήκη MathGeoLib που βρήκαμε στο διαδίκτυο ώστε να εκμεταλλευτούμε ορισμένες δυνατότητες που δεν μας εξασφάλιζε η χρήση της GeoLib,η οποία στον τρισδιάστατο χώρο παρέχει μόνο σημεία 3 διαστάσεων και τίποτα άλλο, σε αντίθεση με την MathGeoLib που παρέχει τα εργαλεία όπως σφαίρες,πολύεδρα κλπ καθώς και κατάλληλες μεθόδους για να υλοποιήσουμε τον αυξητικό αλγόριθμο κυρτού περιβλήματος.

<u>Ωστόσο πρέπει να τονίσουμε ότι δεν πήραμε τίποτα έτοιμο από τη βιβλιοθήκη για την υλοποίηση του αυξητικού αλγορίθμου.</u>

<u>Επίσης για την είσοδο του αρχείου μοντέλου δεν πρόλαβα να υλοποίησω τις κατάλληλες εντολές ώστε ορισμένοι χαρακτήρες στην αρχή και στη μέση του αρχείου να αποφεύγονται(αυτοί που απορρέουν από το Blender).ΕΠΟΜΕΝΩΣ ΑΝ Ο ΧΡΗΣΤΗΣ ΘΕΛΕΙ ΝΑ ΦΟΡΤΩΣΕΙ ΤΟ ΜΟΝΤΕΛΟ ΤΟΥ,ΤΟΤΕ ΑΥΤΟ ΘΑ ΠΡΕΠΕΙ ΝΑ ΑΠΑΡΤΙΖΕΤΑΙ ΑΠΟ συντεταγμένες με v μπροστά και facets με f μπροστά!!!</u>

Επιπλέον, επειδή έχουμε να κάνουμε με 3 διαστάσεις υλοποιήθηκε ένα menu με λειτουργίες πληκτρολογίου ώστε να μπορέσουμε να χειριστούμε το μοντέλο μας και να δούμε κάθε νέα υλοποίηση(π.χ κυρτό περίβλημα με facets,χάρτη αποστάσεων κλπ) πατώντας το αντίστοιχο πλήκτρο.

<u>Το MENU μας έχει ως εξής:</u>

> *F1: Δείχνει το αντικείμενο με τα facets ζωγραφισμένα στο εσωτερικό ή μόνο ως γραμμές με μπλε χρώμα*
>
> *F2: Δείχνει το κυρτό περίβλημα του αντικειμένου μας με κίτρινο χρώμα ως ζωγραφισμένα τρίγωνα*
>
> *F3: Δείχνει το κυρτό περίβλημα του αντικειμένου μας με κίτρινο χρώμα ως γραμμές*
>
> *F4: Δείχνει τον χάρτη αποστάσεων(λευκό-κόκκινο χρώμα)*

F5: Δείχνει το AABB με κόκκινο χρώμα(σε ειδική μορφή για να είναι εμφανές-χωρισμένο σε τρίγωνα)

F6: Εμφανίζει τους xyz άξονες με λευκό χρώμα

Button I : Zoom In

Button O: Zoom Out

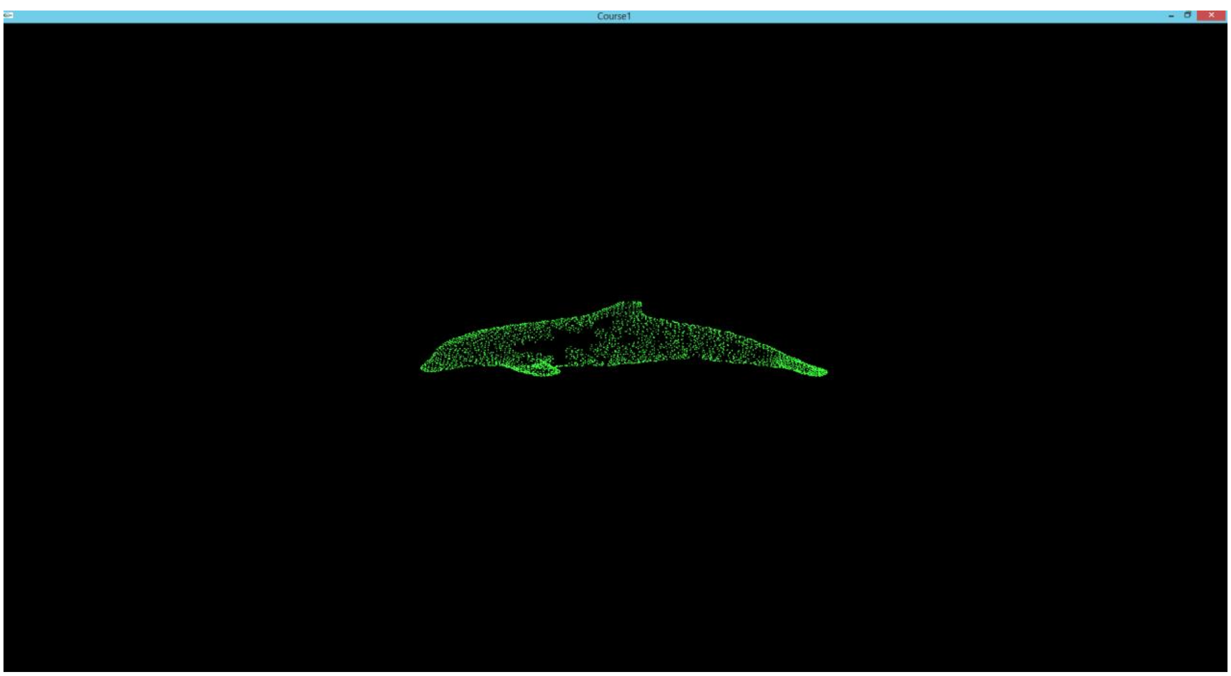
Buttons E,D,S,F,R,T: Περιστρέφει τη θέα του μοντέλου

Buttons Y,G,H,J,K,L: Κίνηση της σφαίρας του 3^{ου} ερωτήματος

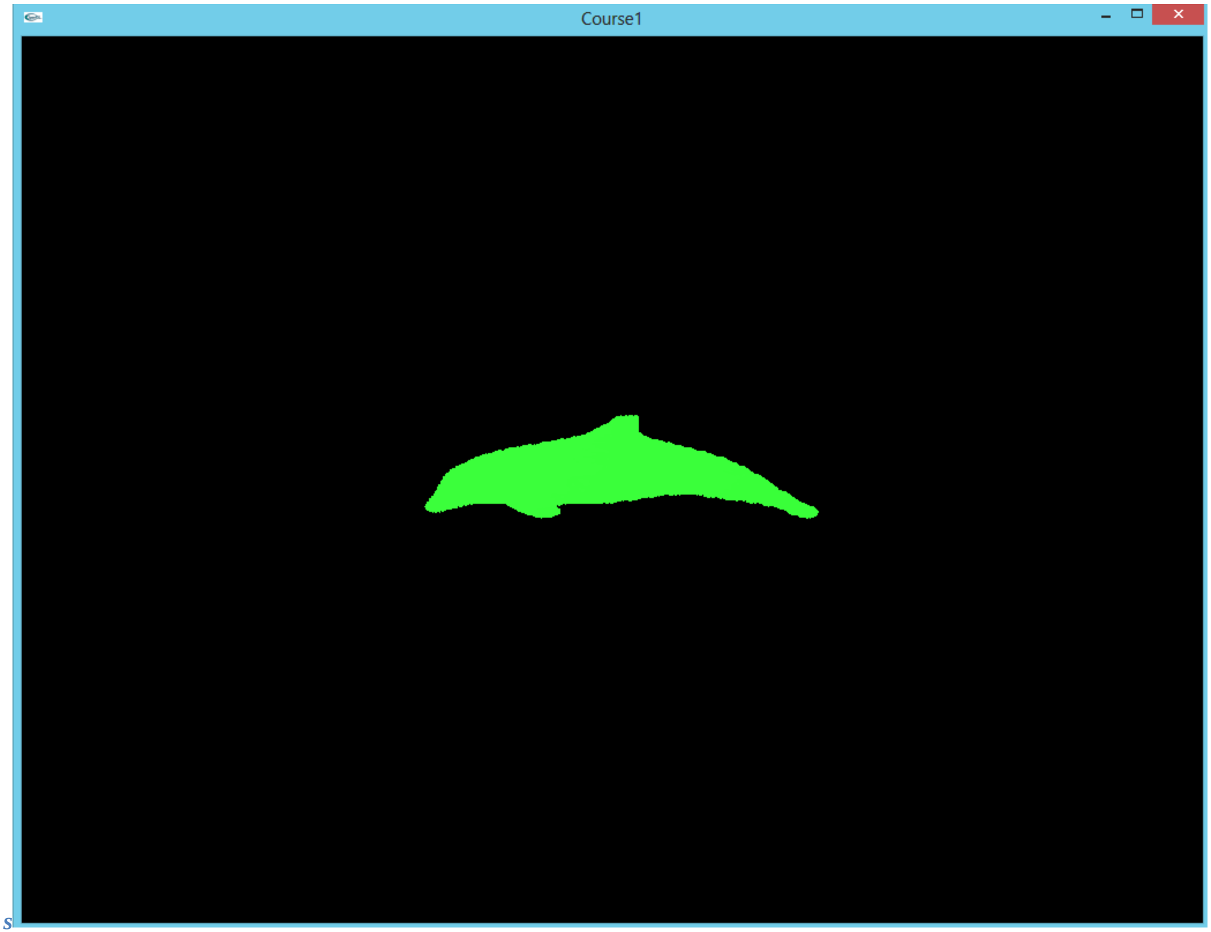
SPACEBAR: Όλα στην αρχική κατάσταση

Για την επίλυση του προβλήματος εργαστήκαμε ως εξής:

Αρχικά μέσα αναλύοντας το αρχείο του μοντέλου μας βρήκαμε τα σημεία του αντικειμένου μας(3D_2) καθώς και τα facets του δηλαδή τα τρίγωνα από τα οποία αποτελείται.



Τα σημεία του μοντέλου μας με βάση το αρχείο 3D_2.obj\z



Το αντικείμενο μας με ζωγραφισμένα τα facets του(ή meshes)

Στη συνέχεια προχωρήσαμε στην υλοποίηση του αυξητικού αλγορίθμου γνωστού και από το μάθημα μας.

Για να γίνει αυτό αρχικά σχεδιάσαμε ένα τετράεδρο που τηρεί τις αρχικές προϋποθέσεις που θέτει ο αλγόριθμος μας και στη συνέχεια για κάθε νέο σημείο ελέγξαμε αν το σημείο βρίσκεται στο εσωτερικό του τετραέδρου ή στο εξωτερικό.

Αν το σημείο βρισκόταν στο εσωτερικό, τότε πηγαίναμε στο επόμενο σημείο, διαφορετικά εκτελούσαμε μια διαφορετική μεθοδολογία. Συγκεκριμένα βρίσκαμε για το σημείο αυτό, ποια είναι τα facets του πολυέδρου μας τα οποία βλέπει το σημείο.Για να το πετύχουμε αυτό μετατρέψαμε κάθε facet του μοντέλου σε plane και ελέγξαμε αν το σημείο βρίσκεται στο επάνω μέρος του plane.

Να αναφέρουμε ότι το πολύεδρο μας κάθε στιγμή φροντίζαμε έτσι ώστε όλα τα facets να έχουν τη νόρμα τους προς τα έξω από το πολύεδρο, έτσι είχαμε ουσιαστικά μια φορά σε κάθε facet αντίστροφη της φοράς του ρολογιού.

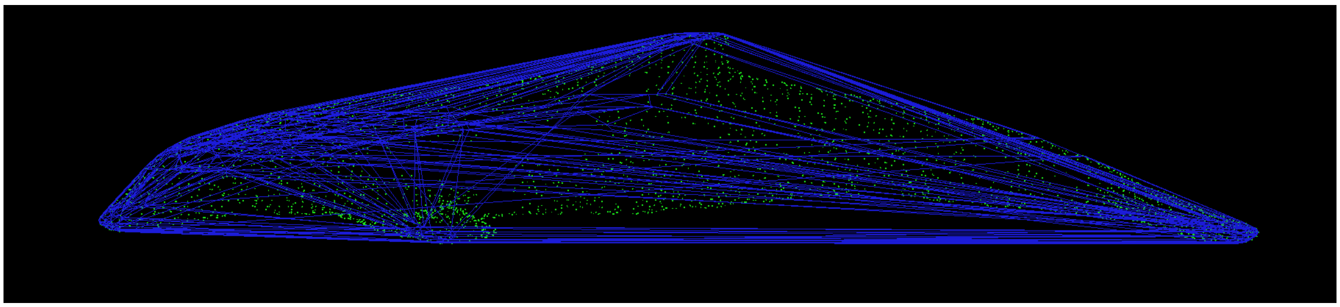
Αφού ελέγχαμε όλα τα facets και βρίσκαμε τα ορατά από το σημείο μας, έπειτα χρειάστηκε να βρούμε τον ορίζοντα που δεν είναι τίποτα άλλο από ευθύγραμμα τμήματα. Για να τα βρούμε, διασχίσαμε όλες τις πλευρές των ορατών facets και όταν κάποια πλευρά εμφανιζόταν μόνο 1 φορά και δεν εμφανιζόταν η αντίθετη της καθόλου, τότε η πλευρά αυτή προφανώς άνηκε στον ορίζοντα.

Παίρνοντας τα σημεία των πλευρών αυτών, δηλαδή 2 σημεία ανά πλευρά και το σημείο το οποίο διερευνούμε, δημιουργούμε νέο facet με τα 3 αυτά σημεία.

Το ίδιο κάνουμε για όλο τον ορίζοντα και στο τέλος διαγράφουμε τα ορατά facets.

Επίσης εκτελούμε την κατάλληλη μέθοδο έτσι ώστε το πολύεδρο μας να ορίζεται στην δομή έτσι ώστε οι νόρμες των facets να βρίσκονται προς τα έξω από το πολύεδρο.

Συνεχίζουμε επαναληπτικά και έτσι κατασκευάζουμε το τελικό πολύεδρο το οποίο είναι και το Convex Hull 3D.



Το 3Δ κυρτό περίβλημα με μπλέ χρώμα και το μοντέλο μας που αποτελείται από πράσινα σημεία

Το επόμενο βήμα ήταν και αυτό που μας δυσκόλεψε περισσότερο από όλα τα ερωτήματα.Η αρχική ιδέα για την υλοποίηση του χάρτη αποστάσεων ήταν η εξής:

Να πάρουμε για κάθε σημείο κάθε έδρας του κυρτού πολυέδρου μας(ή τουλάχιστον για πολλά σημεία κάθε έδρας) ακτίνες προς το εσωτερικό του πολυέδρου δηλαδή προς το μοντέλο μας. Όπως ακριβώς και στη 2D υλοποίηση έπειτα θα βρίσκαμε τα σημεία τομής και θα χρωματίζαμε με βάση τα μήκη των ακτίνων.

<u>Ωστόσο παρά την προσπάθεια αυτή δεν υπήρξε αποτέλεσμα και επομένως βασιστήκαμε σε μία προσέγγιση της παραπάνω λύσης.</u>

Πιο συγκεκριμένα εργαστήκαμε ως εξής:

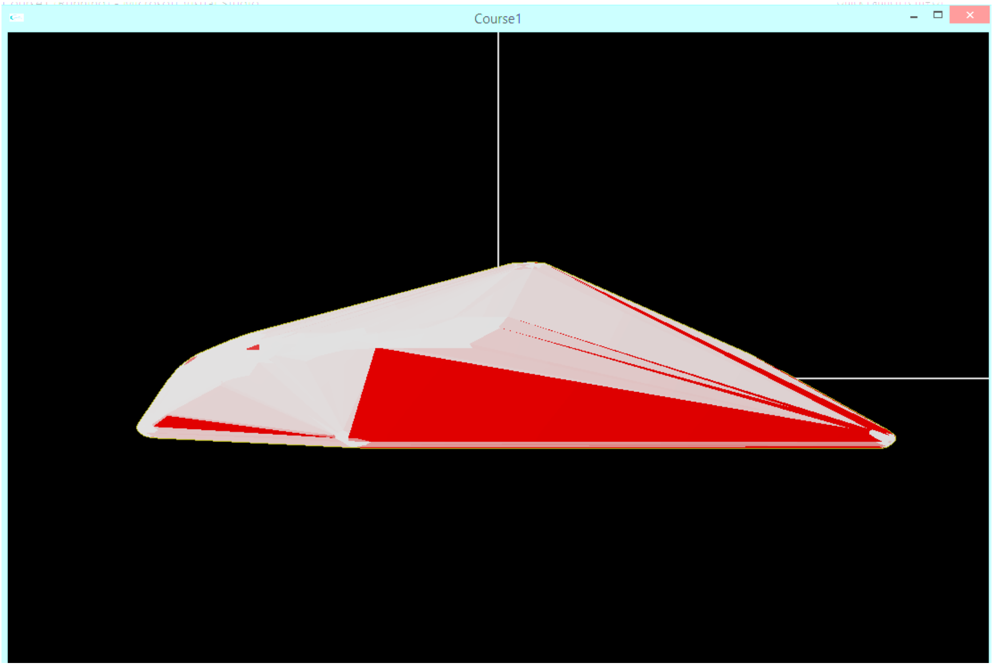
Αντί να πάρουμε πολλές ακτίνες για κάθε έδρα του κυρτού πολύεδρού μας, θα πάρουμε μόνο 1 ακτίνα και αυτή από το βαρύκεντρο της έδρας μας.Αυτή είναι μια προσέγγιση καθώς δεν κατάφερα να υλοποιήσω την ίδια μέθοδο με πολλές ακτίνες να βγαίνουν από την έδρα.

Έτσι, για κάθε μοναδική ακτίνα για κάθε έδρα που πήραμε προς το εσωτερικό του κυρτού πολύεδρου ελέγχουμε διαδοχικά όλα τα facets του αντικειμένου μας και βρίσκουμε τα σημεία τομής.
Όπως εργαστήκαμε και στο 1^{ο} ερώτημα, θα κρατήσουμε το σημείο τομής που βρίσκεται πιο κοντά στο αρχικό σημείο της ακτίνας δηλαδή το βαρύκεντρο κάθε έδρας.

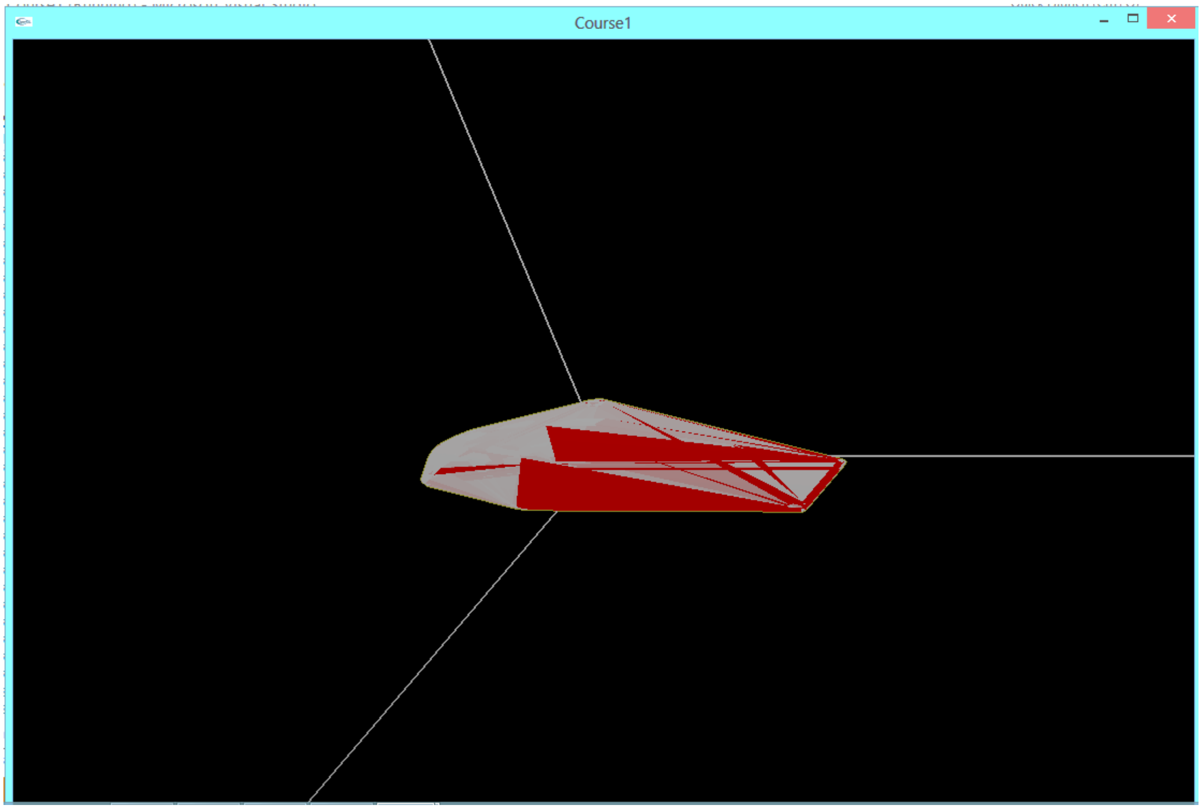
Με βάση αυτή την ακτίνα και το τελικό μήκος της σχεδιάζουμε και την αντίστοιχη έδρα με το χρώμα αυτό. Η χρωματολογία είναι ίδια με το 1^{o} ερώτημα.

Το κόκκινο αντιπροσωπεύει την μέγιστη απόσταση και το λευκό την ελάχιστη.

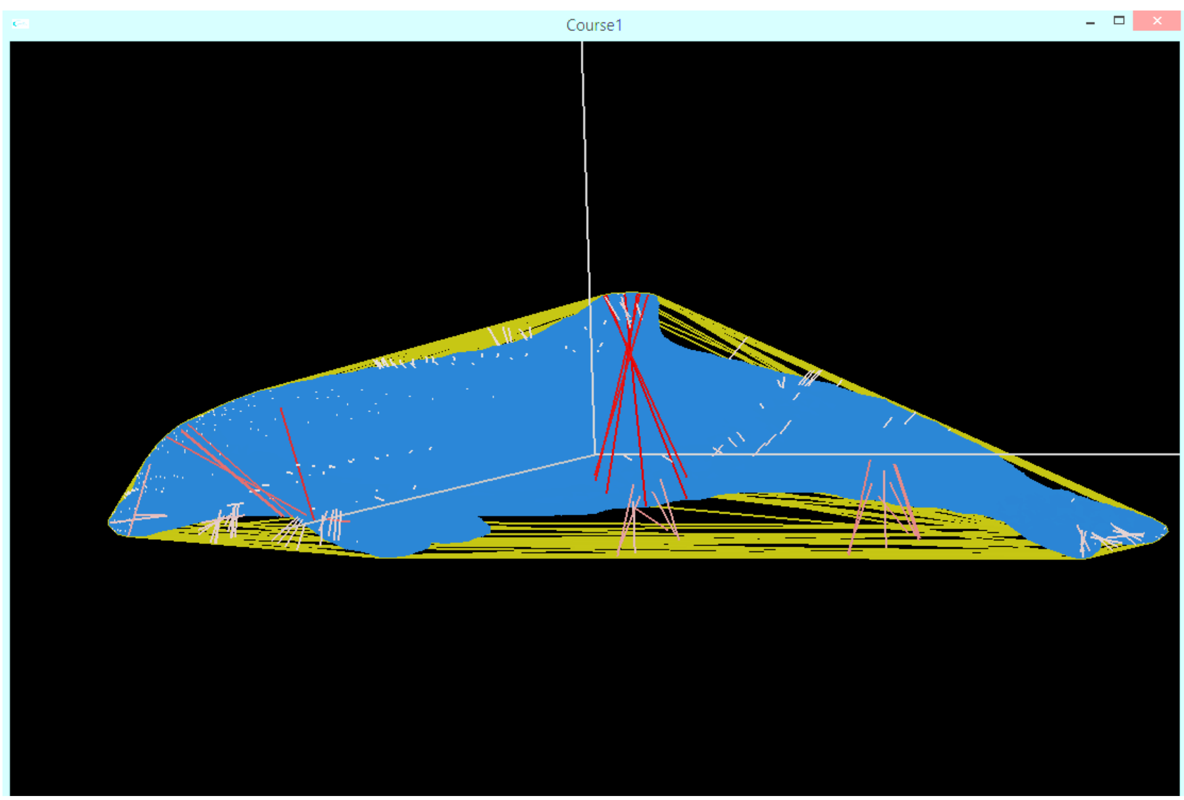
Επίσης υλοποίησαμε το ίδιο ερώτημα αλλά χρωματίσαμε και μόνο τις ακτίνες για να δούμε το αποτέλεσμα, όπως φαίνεται και στις φωτογραφίες.



Ο χάρτης αποστάσεων σε 3D.Όσο πιο κόκκινο τόσο μεγαλύτερη η απόσταση από το Convex Hull 3D προς το αντικείμενο(προσέγγιση)



Ο χάρτης αποστάσεων σε 3D.Όσο πιο κόκκινο τόσο μεγαλύτερη η απόσταση από το Convex Hull 3D προς το αντικείμενο(προσέγγιση) – Από μία άλλη όψη



Το μοντέλο με το χάρτη αποστάσεων σχεδιασμένο με ακτίνες του αντίστοιχου χρώματος

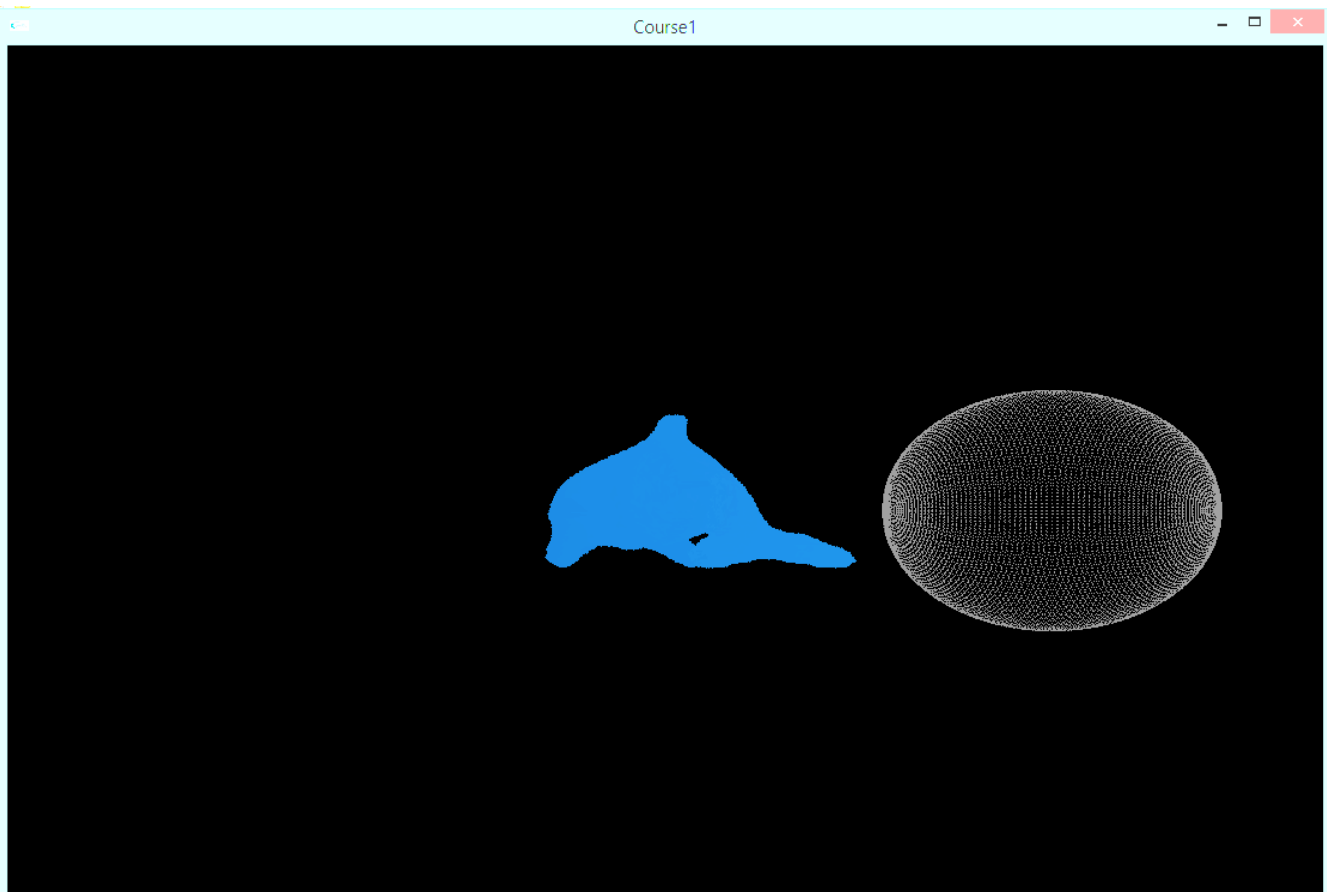
ΕΡΩΤΗΜΑ 3^{O}

Υλοποιήστε αλγόριθμο ανίχνευσης σύγκρουσης μίας κινούμενης σφαίρας ακτίνας R με το Model_3D χρησιμοποιώντας μόνο το κυρτό περίβλημα και το χάρτη αποστάσεων (όχι το ίδιο το Model_3D)

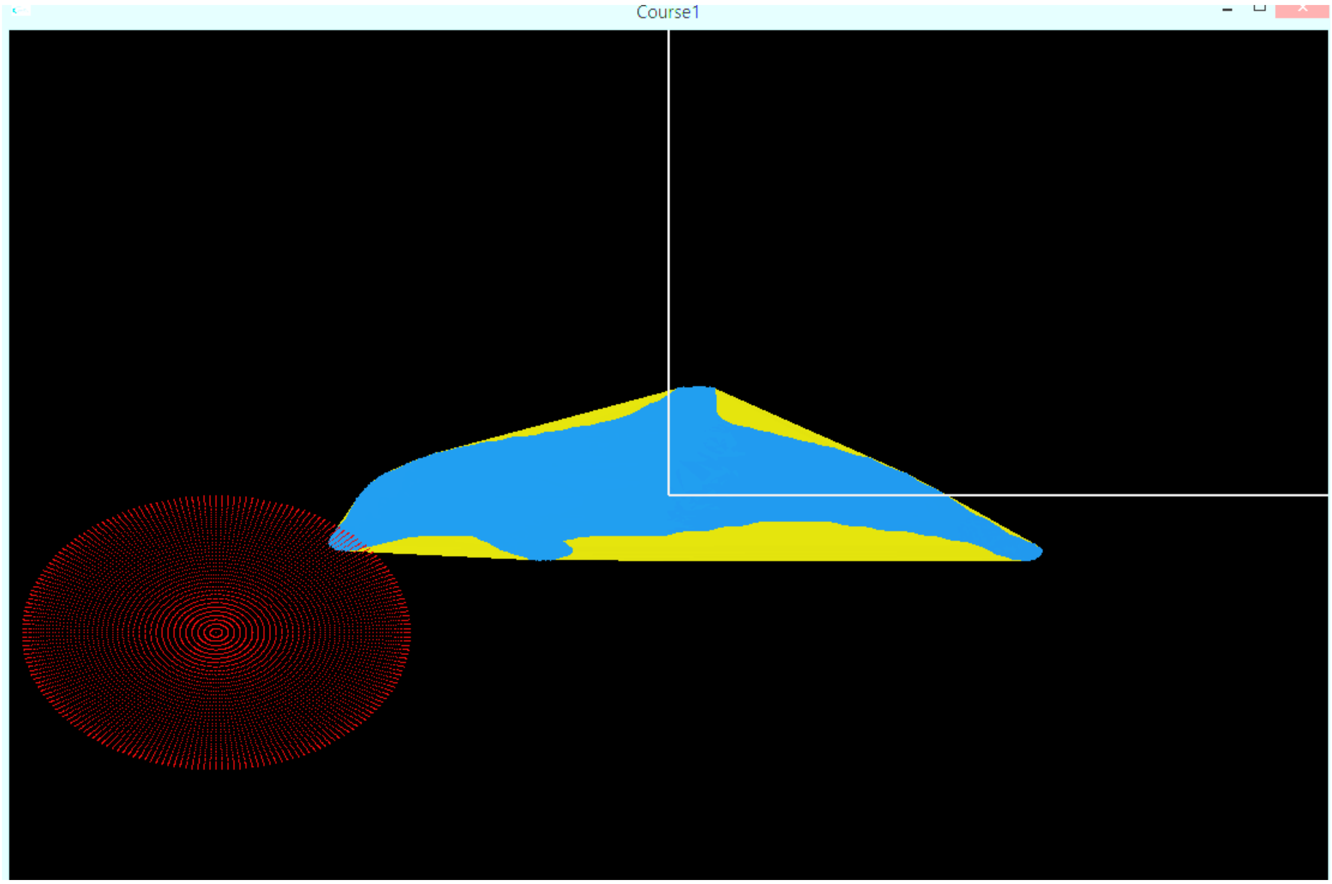
Στο συγκεκριμένο ερώτημα δεν καταφέραμε να υλοποιήσουμε επαρκή λύση όσον αφορά την ανίχνευση σύγκρουσης με βάση το χάρτη αποστάσεων που δημιουργήσαμε στο προηγούμενο ερώτημα. Ωστόσο υλοποιήσαμε ένα παράδειγμα με έναν αλγόριθμο που αξιοποιεί το κυρτό περίβλημα για να ελέγξει αν υπάρχει σύγκρουση της σφαίρας με αυτό.

Πιο συγκεκριμένα αφού δημιουργήσαμε μία σφαίρα από πολλά σημεία στο χώρο, της οποίας η ακτίνα ρυθμίζεται μέσα από τον κώδικα μέσα στην Render ελέγχαμε συνεχώς αν μετά από την κίνηση της σφαίρας αυτή συγκρούεται ή όχι.

Ελέγχουμε την απόσταση κυρτού πολυέδρου με κέντρο της σφαίρας και αν αυτή είναι μικρότερη της ακτίνας τότε έχουμε σύγκρουση οπότεη σφαίρα γίνεται κόκκινη ολόκληρη.



Το αντικείμενο και η σφαίρα στα δεξιά του με γκρι χρώμα



Ανίχνευση σύγκρουσης σφαίρας-CH

Ανασκόπηση της εργασίας - Βελτιώσεις – Επεκτάσεις

Η εργασία δοκιμάστηκε σε σταθερό και φορητό υπολογιστή και λειτούργησε άψογα και στα 2. Χρειάστηκαν περίπου 15 ημέρες για την ολοκλήρωση της συνολικά.
Στο μέλλον αναμένεται να βελτιώσω την εργασία και να ολοκληρώσω κάθε απαίτηση της, ενώ παράλληλα θα επιδιώξω να μειώσω τον χρόνο εκτέλεσης ορισμένων τμημάτων του κώδικα.

Η εργασία αυτή μου έδωσε τη δυνατότητα να υλοποιήσω αλγορίθμους και να γράψω κώδικα που παράγει αποτελέσματα άμεσα ορατά, ενώ παράλληλα χρειάστηκαν αρκετές ώρες για σκέψη του τρόπου επίλυσης των προβλημάτων.Ήταν επίσης ιδιαίτερα ενδιαφέρον το γεγονός ότι μου δόθηκε η δυνατότητα να προσθέσω επιπλέον λειτουργίες για την πλοήγηση του χρήστη.

Για να μπορέσει κάποιος να χρησιμοποιήσει την εφαρμογή πρέπει απαραίτητα να εγκαταστήσει και τις απαιτούμενες βιβλιοθήκες που αναφέραμε.