

ΓΡΑΦΙΚΑ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΑ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ

ΠΡΩΤΗ ΑΣΚΗΣΗ

ΟΝΟΜΑΤΕΠΩΝΥΜΑ ΜΕΛΩΝ ΟΜΑΔΑΣ:

ΒΙΚΤΩΡ ΜΕΓΗΡ

ΑΜ:3026

ΒΑΪΑ ΝΤΑΦΟΠΟΥΛΟΥ

ΑΜ:3050

ΕΙΣΑΓΩΓΗ:

Η παρακατω περιγραφη αφορα την υλοποιηση ενος σχεδιαστικου προγραμματος σε OpenGL το οποιο δινει την δυνατοτητα στον χρηστη να σχεδιαζει μη αυτοτεμνουμενα πολυγωνα στα οποια μπορεί να εκτελει τις παρακατω λειτουργιες:

- 1)τριγωνοποιηση
- 2)γεμισμα με χρωμα της επιλογης του
- 3)αποκοπη με ενα ορθογωνιο αποκοπης
- 4)προβολη αντικειμενων στις 3 διαστασεις

Συναρτησεις και λειτουργιες:

main():

Αρχικα, κατασκευασαμε ενα παραθυρο με χρηση ετοιμων συναρτησεων της OpenGL,για τον ορισμο του χρωματος(background),διαστασεων και ονοματος του παραθυρου.Πατώντας middle bottom εμφανιζεται ενα μενου (χρηση της createGLUTMenus) με τις επιλογες ACTION,LINE_COLOR,FILL_COLOR. Καθεμια απο αυτες τις επιλογες οδηγει σε ενα υπο-μενου.

ACTION

Κανοντας αυτη την επιλογη θα εμφανιστει το μενου με τις επιλογες POLYGON,EXIT,CLIPPING και EXTRUDE οι οποιες εξηγουνται παρακατω.

LINE_COLOR

Το υπο-μενου αυτης της επιλογης οδηγει σε μια λιστα με 16 χρωματα απο τα οποια ο χρηστης μπορεί να επιλεξει το χρωμα των γραμμων του καθε πολυγωνου που προκειται να σχεδιασει.

FILL_COLOR

Αντιστοιχα με το παραπανω εδω ο χρηστης επιλεγει το χρωμα με το οποιο θα γεμισει το καθε πολυγωνο που θα σχεδιασει.

Επιπροθετα εκει γινεται η κληση της display,keyboard,mouse και καποιων επιπλεον βασικων συναρτησεων της υπολοιησης μας,οι οποιες περιγραφονται στη συνεχεια αναλυτικα.

-ΔΟΜΕΣ-

Εχουμε δημιουργησει ενα struct με ονομα Points με πεδια δυο float αριθμους που περιεχουν τις συντεταγμενες των σημειων. Επισης φτιαξαμε ενα struct με ονομα Triangles με πεδια 3 σημεια που οριζουν ενα τριγωνο. Τελος φτιαξαμε ενα struct που ονομαζεται Polygon με πεδια:

τον πίνακα `azot`, που περιέχει τα σημεία ενός πολυγώνου, τον πίνακα `tex` ο οποίος έχει τα τρίγωνα ενός πολυγώνου καθώς και τους ακέραιους αριθμούς `counter` (μετρητής σημείων πολυγώνου) και `tricounter` (μετρητής τριγώνων πολυγώνου).

`Init()`:

Αυτή είναι η συνάρτηση αρχικοποίησης μας, στην οποία ορίζεται το `display mode`. Αν ο χρήστης επιλέξει την λειτουργία `extrude` τότε η `init()` θα αρχικοποιήσει το `display mode` για τρισδιάστατη αναπαράσταση αντικειμένων. Διαφορετικά θα αρχικοποιήσει το `display mode` για δισδιάστατη αναπαράσταση.

`mouse()`:

Η συνάρτηση αυτή καθορίζει τι θα συμβεί με κάθε κλικ που κάνει ο χρήστης με το ποντίκι. Συγκεκριμένα όταν ο χρήστης πατήσει `middle bottom` ανοίγει το κύριο μενού. Έπειτα μετά την επιλογή `POLYGON` θα μπορεί να ζωγραφίζει σημεία με κάθε αριστερό πατήμα του ποντικιού ενώ με το δεξί πατήμα δηλώνει πως θέλει να κλείσει το πολύγωνο του. Στην περίπτωση που ο χρήστης κάνει την επιλογή `CLIPPING` με το αριστερό πατήμα θα ορίσει τα σημεία του ορθογωνίου αποκοπής, ενώ αν επιλέξει `EXTRUDE` κρατώντας πατημένο το αριστερό κλικ ο χρήστης μετακινεί την κάμερα δεξιά ή αριστερά αναλόγα προς τα που θα το συρει.

`mouseMove()`:

Αυτό το τμήμα κώδικα αφορά μόνο την λειτουργία `extrude`. Αυτό που κάνει είναι ότι επιτρέπει στον χρήστη να περιστρέφει την κάμερα σερνώντας το ποντίκι είτε δεξιά είτε αριστερά.

`keyboard()`:

Η συνάρτηση `keyboard` αφορά τα πλήκτρα που έχει την δυνατότητα να πατήσει ο χρήστης και να γίνουν κάποιες λειτουργίες. Συγκεκριμένα, πατώντας `escape` γίνεται εξόδος από το γραφικό περιβάλλον και το παραθύρο κλείνει, ενώ με το πλήκτρο `<T>` ο χρήστης επιλέγει να γίνει τριγωνοποίηση στο/α πολυγώνου/α που έχει σχεδιάσει και με την επαναληψη πατήματος του `<T>` τα τρίγωνα θα εξαγανίζονται. Μετά από κάθε πατήμα του `<T>` πρέπει να πατηθεί αριστερό κλικ για να γίνουν οι λειτουργίες.

`pressKey()` και `releaseKey()`:

Με αυτές τις δύο συναρτήσεις δίνουμε την δυνατότητα στο χρήστη να μετακινείται μέσα στον 3D χώρο με τα `arrow keys` (μπρος-πίσω), αφού έχει επιλεγεί η λειτουργία `extrude`. Όταν ο χρήστης πατήσει ένα πλήκτρο ξεκινά η κίνηση και όταν το αφήσει σταματάει.

Ο συνδυασμός της περιστροφής της κάμερας δεξιά, αριστερά (mouseMove) και της κίνησης μπρος, πίσω επιτρέπει στο χρήστη να δει το extruded πολυγώνο από κάθε οπτική γωνία.

display():

Η display είναι υπεύθυνη για την γραφική αναπαράσταση των σημείων που σχεδιάζει ο χρήστης καθώς και την αποθήκευση αυτών στον πίνακα azor. Επιπλέον εκεί γίνεται η αποθήκευση ολοκληρωμένου του πολυγώνου στον πίνακα array, όταν ο χρήστης ολοκληρώσει την σχεδίαση του.

Βασική εργασία αυτής της συνάρτησης είναι να ενώνει τα σημεία που έδωσε ο χρήστης με γραμμές. Όταν ο χρήστης κλείσει το πολυγώνο γίνεται γεμίσμα αυτό με το χρώμα που έχει επιλέξει. Ακόμη φροντίζει να καλεί κατάλληλες συναρτήσεις για τον έλεγχο των μη αυτοτεμνόμενων πολυγώνων. Αν παραβιαστεί η σύμβαση αυτή για οποιαδήποτε ακμή πολυγώνου τότε το σημείο δεν ζωγραφίζεται και η σχεδίαση συνεχίζεται από το τελευταίο σημείο. Στην περίπτωση που ο χρήστης κάνει τριγωνοποίηση εκείνη ζωγραφίζει τα τρίγωνα με πράσινο (πάντα) χρώμα. Αν πρώτα έχει γίνει αποκοπή και έπειτα τριγωνοποίηση ζωγραφίζει τα νέα τρίγωνα των νέων πολυγώνων που έχουν δημιουργηθεί και σβηνει κάθε παλιά τριγωνοποίηση.

nonselintersecting() και closeintersection():

Οι δύο αυτές συναρτήσεις που καλούνται από την display είναι υπεύθυνες για τον έλεγχο αυτοτεμνόμενου πολυγώνου. Εξετάζουν δηλαδή αν το νέο σημείο που έδωσε ο χρήστης παραβιάζει την σύμβαση που ζητήθηκε να ακολουθήσουμε. Η πρώτη κάνει έλεγχο για όλα τα σημεία ενώ η δεύτερη μόνο για την ένωση του τελευταίου σημείου με το πρώτο ενός πολυγώνου.

ProcessMenuEvents():

Ανάλογα με την επιλογή που κάνει ο χρήστης από το μενού και υπο-μενού η ProcessMenuEvents κάνει τις αντίστοιχες αναθέσεις τιμών που αφορούν global μεταβλητές και τα χρώματα γραμμών και γεμίσματος.

CopyPoly() και Triangulate::Process():

Η copyPoly αντιγράφει στον πίνακα των τριγώνων rex ενός πολυγώνου τα τρίγωνα που προκύπτουν από τον αλγόριθμο τριγωνοποίησης. Η copyPoly() καλεί την συνάρτηση Triangulate::Process() η οποία παίρνει σαν ορίσμα τα σημεία ενός πολυγώνου και με την χρήση των παρακάτω συναρτήσεων, Area, InsideTriangle, Snip, υλοποιεί την τριγωνοποίηση του πολυγώνου.

Χρησιμοποιούμε το τμήμα κώδικα που δίνεται (e-course).

Clipping(),SuthHodgClip(), clip(), clearClip(), x_intersect() και y_intersect() :
Αυτες οι συναρτησεις αφορουν την αποκοπη των πολυγωνων. Οταν ο χρηστης ορισει το ορθογωνιο αποκοπης καλειται η SuthHodgClip() η οποια με την βοηθεια της clip() υλοποιει τον αλγοριθμο Sutherland-Hodgman. Αυτος ο αλγοριθμος δημιουργει το νεο clipped πολυγωνο και το αποθηκευει στον array. Επειτα, σχεδιαζεται το ορθογωνιο αποκοπης με μπορντω χρωμα και τελος καλειται η clearClip() που κανει λευκη την περιοχη εκτος του ορθογωνιου αποκοπης.
Για τον αλγοριθμο Sutherland-Hodgman χρησιμοποιουμε τον κωδικα απο την παρακατω σελιδα: <https://www.geeksforgeeks.org/polygon-clipping-sutherland-hodgman-algorithm-please-change-bmp-images-jpeg-png/> .

display3D():

Η συναρτηση αυτη υλοποιει την αναπαρασταση των αντικειμεων στις 3 διαστασεις. Για να γινει αυτο αλλαζει τον τροπο προβολης του παραθυρου ωστε να βλεπει ο χρηστης τα αντικειμενα σε μορφη μοντελου. Αρχικα, ζητειται απο τον χρηστη το μηκος εξωθησης .Στη συνεχεια, γινεται προοπτικη προβολη και καθοριζεται η θεση της καμερας στο χωρο με την συναρτηση gluLookAt της OpenGL. Σε καθε κληση της συναρτησης display3D υπολογιζεται η καινουργια θεση της καμερας μεσω της συναρτησης computePos().

Πρωτα, ζωγραφιζονται τα QUADS με το LINE_COLOR που διαλεξε ο χρηστης. Επειτα ζωγραφιζονται με το FILL_COLOR η βαση και η κορυφη του πολυγωνου ,οι οποιες εχουν τριγωνοποιηθει, χρησιμοποιωντας TRIANGLES. Τελος ζωγραφιζονται οι γραμμες των πολυγωνων της κορυφης και της βασης καθως και των QUADS. Η συναρτηση δινει ακομη την δυνατοτητα να γινει τριγωνοποιηση στην κορυφη και στην βαση του 3Δ πολυγωνου πατωντας <T>.Αν ξανα πατηθει το πληκτρο <T> τα τριγωνα εξαφανιζονται.

PrintPoint():

Ειναι μια βοηθητικη κυριως συναρτηση την οποια χρησιμοποιουμε για διαγνωστικους λογους και τυπωνει τις συντεταγμενες ενος σημειου.

Bonus 1:

Βαλαμε το μηκος εξωθησης να ειναι ενα πεδιο του struct polygon ετσι ωστε καθε πολυγωνο να εχει ενα μοναδικο μηκος και στη συνεχεια ζηταμε απο τη χρητη να δινει το μηκος αυτο για καθε πολυγωνο.