

Βασίλειος Παπασακελλάριου AM:1762

Ιωάννης-Αριστείδης Παπασακελλάριου AM:2520

Η Άσκηση μεταγλωττίζεται με την εντολή g++ askhsh1.cpp -lGL -lGLU -lglut -o ask1. Μετα τρέχετε το εκτελέσιμο που προκύπτει ως ./ask1

Τρόπος Σχεδίασης και Ιδιαιτερότητες:

Στην αρχή εμφανίζεται ένα παραθυρό μεγεθους 600 * 500 όπου ο χρήστης μπορεί να σχεδιάσει διάφορα πολύγωνα της αρεσκείας του. Όταν αποφασίσει ότι θέλει να σχεδιάσει μπορεί να πατήσει το μεσαίο κλικ του ποντικιού με το οποίο εμφανίζεται ένα μενού με διάφορες επιλογές. Με τις επιλογές LINECOLOR και FILLCOLOR ο χρήστης μπορεί να επιλέξει χρώμα για το γέμισμα των πολύγωνων και γραμμής αντιστοίχα. Επειτα στο μενού ACTION ο χρήστης μπορεί να επιλέξει την επιλογή POLYGON. Τότε ο χρήστης μπορεί να κάνει αριστερό κλικ στην οθόνη να επιλέγει κορυφές και ταυτόχρονα να εμφανίζονται οι γραμμές του πολύγωνου. Αν πατήσει δεξί κλικ ενώνεται η τελευταία κορυφή με την πρώτη και ουσιαστικά δημιουργείται ένα πολύγωνο. Όταν ο χρήστης αποφασίσει ότι θέλει να κάνει αποκοπή των πολύγωνων που σχεδίασε τότε από το μενού στην επιλογή ACTION μπορεί να επιλέξει την επιλογή CLIPPING. Τότε ο χρήστης μπορεί να επιλέξει την πρώτη κορυφή και με πατημένο το αριστερό κλικ και μετακινώντας το ποντίκι προς τα πάνω και δεξιά μπορεί να βλέπει το ορθογώνιο αποκοπής που σχεδιάζεται. Όταν αποφασίσει ποιο θέλει να είναι το ορθογώνιο αποκοπής πρέπει να πατήσει και δεύτερο κλικ έτσι ώστε να σχεδιαστεί το παραθυρό αποκοπής. **Πρέπει η πρώτη κορυφή του ορθογωνιου αποκοπής να επιλεγεί από κάτω και αριστερά και η δεύτερη πάνω και δεξιά!!** Επειτα ο χρήστης μπορεί να δει τα αποκομμένα πολύγωνα με χρώμα πράσινο στο περιγράμμα τους έτσι ώστε να ξεχωρίζουν από τα αρχικά πολύγωνα και να καταλάβει ο χρήστης που έγινε η αποκοπή.

Υπάρχει μια επιπλέον επιλογή στο υπομενού ACTION η οποία είναι η CLEAR SCHEME όπου αμα την επιλέξει ο χρήστης όταν είναι σε mode που σχεδιάζει πολύγωνα, καθαρίζει ότι έχει σχεδιάσει ο χρήστης μέχρι εκείνη την στιγμή και μπορεί να ξαναζωγραφίσει από την αρχή. Αυτή η επιλογή δουλεύει μόνο όταν ο χρήστης έχει πατήσει POLYGON πριν.

Τέλος στο υπομενού ACTION με την επιλογή EXTRUDE εμφανίζεται στην οθόνη ένα μήνυμα που ζητάει το μήκος εξώθησης και μέσα στο ορθογώνιο που εμφανίζεται πληκτρολογεί τον αριθμό που θέλει. Μετά πρέπει να πατήσει ENTER για να γίνει το παραθυρό τρισδιάστατης αποδοσης.....

Global μεταβλητές:

GLsizei windowsWidth , windowsHeight : δηλώνουν το μήκος και το ύψος του παραθύρου εμφάνισης.

int numberOfVertices: δηλώνει τον αριθμό των κορυφών του τρεχόντος πολύγωνου που σχεδιάζει ο χρήστης.

Int numberOfPolygons: δηλώνει τον αριθμό των πολύγωνων που έχει σχεδιάσει ο χρήστης. Αυξάνεται κάθε φορά που ο χρήστης αποφασίζει ότι τελείωσε με την σχεδίαση του πολύγωνου του (όταν πατάει δεξί κλικ) και με την προϋπόθεση ότι το πολύγωνο που εφτιαξε δεν είναι αυτοτετμονομένο.

int currentP = 0: Δηλώνει το τρέχον πολύγωνο που σχεδιάζεται και παίρνει χρώμα γραμμής και γέμισματος.

bool colorForLinesSelected[50]: Πίνακας που δηλώνει για κάθε πολυγώνο αν έχει επιλεγεί χρώμα γραμμής. Χρησιμοποιείται έτσι ώστε αν έχει σχεδιαστεί και επιλεγεί χρώμα γραμμής από τον χρήστη ή όχι να μην γίνει άλλη αλλαγή στο χρώμα γραμμής.

bool colorForFillSelected[50]: Πίνακας που δηλώνει για κάθε πολυγώνο αν έχει επιλεγεί χρώμα γεμίσματος. Χρησιμοποιείται έτσι ώστε αν έχει σχεδιαστεί και επιλεγεί χρώμα γεμίσματος από τον χρήστη ή όχι να μην γίνει άλλη αλλαγή στο χρώμα γεμίσματος.

bool design: Μεταβλητή που ενημερώνεται αναλόγα με το αν ο χρήστης μπορεί να σχεδιάσει πολυγώνο στην οθόνη ή όχι.

bool PolygonMode: Δηλώνει ότι ο χρήστης είναι στο Mode όπου σχεδιάζει τα πολυγώνα. Χρησιμοποιείται στην συνάρτηση drawScheme και σε άλλες συναρτήσεις.

bool clippingMode: Δηλώνει ότι ο χρήστης θέλει να κάνει αποκοπή πολυγώνων. Χρησιμοποιείται στην drawScheme και σε άλλες περιπτώσεις.

bool polygonInterrupted: Δηλώνει αν το πολυγώνο που σχεδίασε ο χρήστης είναι αυτοτεταμένο ή όχι. Χρησιμοποιείται στη συνάρτηση polygonCheckForIntersection.

bool polygonNotNormal: Δηλώνει αν ένα πολυγώνο είναι κανονικό. Χρησιμοποιείται στην συνάρτηση process.

bool polygon_triangles_draw: Χρησιμοποιείται για να δηλώσει αν πρέπει να φανούν τα τρίγωνα που δημιουργήθηκαν μετά την τριγωνοποίηση των πολυγώνων πατώντας T ή t.

clipped_polygon_triangles_draw: Χρησιμοποιείται για να δηλώσει αν πρέπει να φανούν τα τρίγωνα των πολυγώνων που προέκυψαν μετά την αποκοπή των κανονικών πολυγώνων που σχεδίασε ο χρήστης χρησιμοποιώντας το ορθογώνιο αποκοπής, πατώντας T ή t.

bool userInputMode: Δηλώνει ότι ο χρήστης επιθυμεί να κάνει εξώθηση των πολυγώνων και να δώσει ένα μήκος εξώθησης. Χρησιμοποιείται στην συνάρτηση drawScheme.

bool draw3dPolygons: Χρησιμοποιείται για να δηλώσει ότι θα ζωγραφιστούν στην οθόνη τα 3d πολυγώνα μετά το μήκος εξώθησης που έχει δώσει ο χρήστης και πατώντας Enter. Χρησιμοποιείται από την drawScheme.

bool motionRectangle: Χρησιμοποιείται στην drawMotionClippingRectangle για να σχεδιάζει ή όχι στην οθόνη το κινούμενο ορθογώνιο αποκοπής.

int keyTimesPressed : Χρησιμοποιείται για τις φορές που ο χρήστης πατάει το πλήκτρο T. Κάθε φορά που πατιέται το πλήκτρο T αυξάνεται. Αρχικοποιείται σε -1 έτσι ώστε την πρώτη φορά που πατιέται το πλήκτρο T να γίνει 0 έτσι ώστε όταν γίνει πράξη με το mod 2 το αποτέλεσμα να είναι άρτιο και να εμφανιστούν τα τρίγωνα τριγωνοποίησης. Γιατί τα τρίγωνα τριγωνοποίησης εμφανίζονται μόνο κάθε φορά που η μεταβλητή είναι άρτιος αριθμός (Επεξήγηση αργότερα όταν θα πριγραψουμε την drawScheme).

int keyTimesPressed1: Χρησιμοποιείται με τον ίδιο σκοπό όπως και η keyTimesPressed μόνο που απευθύνεται για τα τρίγωνα τριγωνοποίησης των πολυγώνων που αποκοπήκαν.

float angleOfView, float lookX , lookY , lookZ : Χρησιμοποιούνται για την μετακίνηση και την σκοπία της κάμερας

float cameraPosX cameraPosZ : Χρησιμοποιούνται για να δηλώσουν την θέση της κάμερας.

vector <char> z_deph

int extrusion_length: Εδώ αποθηκεύεται το μήκος εξώθησης που θέλει ο χρήστης για να εξωθησει τα πολυγωνα. Πρώτα κάθε φορά που ο χρήστης πληκτρολογεί τον αριθμό αυτός αποθηκεύεται σαν ένας vector από χαρακτήρες και στο τέλος όταν ο χρήστης πατήσει Enter γίνεται ακέραιος αριθμός έτοιμος να χρησιμοποιηθεί για εξώθηση.

```
struct vertex {  
    int x;  
    int y;  
};
```

Δηλώνει τις συντεταγμένες x,y των κορυφών(σημείων που δίνει ο χρήστης πατώντας αριστερό κλικ με το ποντίκι στην οθόνη) των πολυγώνων.

vector <vertex> vertices, clip, clipPoints, clippedPoly: Είναι μεταβλητοί πίνακες όπου κάθε στοιχείο τους είναι μια κορυφή. **Ο πίνακας vertices** χρησιμοποιείται για να αποθηκεύει τις τρέχουσες κορυφές πολυγώνων που δίνει ο χρήστης κατά την διάρκεια της σχεδίασης με το ποντίκι. Κάθε κλικ που είναι ουσιαστικά μια κανονική κορυφή του πολυγώνου του αποθηκεύεται σε αυτόν τον πίνακα. **Ο πίνακας clip** δηλώνει τις 2 κορυφές που δίνει ο χρήστης όταν φτιάχνει ο ορθογώνιο αποκοπής. **Ο πίνακας clipPoints** αποτελείται από 4 κορυφές οι οποίες είναι οι κορυφές του ορθογωνίου αποκοπής.

vector <vertex> triangle_vertices: Εδώ αποθηκεύονται οι κορυφές των τριγώνων των πολυγώνων αλλά και των πολυγώνων που προέκυψαν μετά από αποκοπή, μετά την τριγωνοποίηση.

vector <vector<vertex> > triangle_database, clipped_polygons_triangles: Είναι πίνακες όπου κάθε στοιχείο τους είναι ένας πίνακας που περιέχει τις κορυφές των τριγώνων που προέκυψαν μετά από τριγωνοποίηση του κάθε πολυγώνου. Δηλαδή κάθε θέση τους αναφέρεται σε κάθε ξεχωριστό πολυγώνο και περιέχει τις κορυφές των τριγώνων τους. **Ο πίνακας triangle_database,** αναφέρεται στα αρχικά πολυγώνα που δημιούργησε ο χρήστης και **ο πίνακας clipped_polygons_triangles** στα αποκομμένα πολυγώνα.

vector <vector<vertex> > database, clippedPolygons: Εδώ αποθηκεύονται οι κορυφές των πολυγώνων κάθε ξεχωριστού πολυγώνου που φτιάχνει ο χρήστης και αντιστοίχα οι κορυφές των αποκομμένων πολυγώνων. Κάθε στοιχείο δηλαδή αυτών των πινάκων είναι ένας πίνακας που αναφέρεται σε κάθε ξεχωριστό πολυγώνο και τις κορυφές τους.

vertex vertexCoordinate: Εδώ αποθηκεύονται οι τρέχουσες συντεταγμένες της κορυφής του τρέχοντος πολυγώνου που φτιάχνει ο χρήστης πατώντας αριστερό κλικ.

vector <int> polygonSizes: Κάθε στοιχείο του πίνακα αυτού περιγράφει το πλήθος των κορυφών κάθε πολυγώνου που έχει σχεδιαστεί. Χρησιμοποιείται για να γίνει η αποκοπή των πολυγώνων.

vertex motionClippingRectPoints[2]: Πίνακας 2 θέσεων όπου κάθε στοιχείο του είναι μια κορυφή. Ουσιαστικά η πρώτη κορυφή είναι το xmin, ymin που δίνει ο χρήστης με το πρώτο πατήμα του ποντικιού και η 2η η xmax,ymax που ανανεώνεται κάθε φορά καθώς ο χρήστης μετακινεί το ποντίκι για να αποφασίσει που θέλει να τελειώσει το πολυγώνο αποκοπής.

```
struct Color {  
    double red;  
    double green;  
    double blue;  
}
```

Δομή που περιγράφει ένα χρώμα.

Color colorForFill[50], colorForLines[50]: Κάθε στοιχείο περιέχει την δομή που περιγράφει το χρώμα γεμίσματος ή γραμμής κάθε πολυγώνου.

```
struct Edge{  
    vertex firstVertex;  
    vertex secVertex;  
}
```

Δομή που περιγράφει μια ακμή του πολυγώνου. Κάθε ακμή αποτελείται από 2 κορυφές.

vector <Edge> edges: Πίνακας όπου χρησιμοποιείται για να βρεθούν οι ακμές του τρεχόντος πολυγώνου που σχεδιάζει ο χρήστης. Κάθε στοιχείο του είναι μια ακμή που αποτελεί το μεχρί τρεχόντος πολυγώνου.

vector <vector<Edge> > polygonEdges: Είναι ένας πίνακας όπου περιέχει τις ακμές για κάθε σχεδιασμένο πολυγώνο. Κάθε στοιχείο του δηλαδή είναι ένας πίνακας που αναφέρεται σε κάθε ξεχωριστό πολυγώνο και περιέχει τις ακμές που το αντιπροσωπεύουν.

ΣΥΝΑΡΤΗΣΕΙΣ

void findPolygonEdges: Για το πολυγώνο που μόλις δημιουργήθηκε πατώντας δεξί κλικ και δεν ήταν αυτοτεμένο βρίσκει τις ακμές του και τις αποθηκεύει στον πίνακα **polygonEdges**.

float Area(vector <vertex> contour), bool InsideTriangle(int Ax, int Ay, int Bx, int By, int Cx, int Cy, int Px, int Py), bool Snip(vector <vertex> contour, int u, int v, int w, int n, int *V)

χρησιμοποιούνται στην συνάρτηση **process** και είναι βοηθητικές για την τριγωνοποίηση των κανονικών πολυγώνων.

vector<vertex> Process(vector <vertex> vertices) : Συνάρτηση υπεύθυνη για την τριγωνοποίηση και την εύρεση των κορυφών των τριγώνων του τρεχόντος πολυγώνου. Παιρεί σαν όρισμα τις κορυφές του πολυγώνου και επιστρέφει ένα πίνακα τύπου **vector<vertex>** όπου περιέχει τις κορυφές των τριγώνων που υπολογίστηκαν.

float Area1(vector <vertex> contour, int polysize) : Βοηθητική συνάρτηση που χρησιμοποιείται στη συνάρτηση **ProcessClippedPolygons**.

vector<vertex> ProcessClippedPolygons(vector <vertex> vertices, int polysize): Συνάρτηση υπεύθυνη για την τριγωνοποίηση και την εύρεση των κορυφών των τριγώνων των αποκομμένων πολυγώνων. Παιρνει σαν όρισμα τις κορυφές των αποκομμένων πολυγώνων και το μέγεθος τους (δηλαδή το πλήθος των κορυφών) και επιστρέφει ένα πίνακα τύπου **vector<vertex>** όπου περιέχει τις κορυφές των τριγώνων που υπολογίστηκαν.

void init(void): Συνάρτηση αρχικοποίησης του παραθύρου σχεδίασης πολυγώνων. Θέτει το χρώμα φόντου σε ασπρο έπειτα μπαίνει σε mode προβολής, ανανεώνει τους matrices δημιουργεί το πεδίο προβολής σε όλο το παράθυρο, ενεργοποιεί κάποια χαρακτηριστικά έτσι ώστε να φαίνονται πιο

ωραία οι γραμμές ποθ σχηματίζουν τα πολυγωνα που σχεδιάζει ο χρήστης και τέλος αρχικοποιεί τα χρώματα για τις γραμμές και το γεμίσμα των πολυγωνων σε μαυρο και ασπρο αντιστοιχα σε περίπτωση που ο χρήστης δεν επιλεξει καποιο χρώμα για αυτα ο ιδιος.

void reshape(GLint width, GLint height): Συνάρτηση που αρχικοποιεί το παραθυρο πίσω στις αρχικές του διαστάσεις όταν ο χρήστης προσπαθίσει να αλαξει τις διαστάσεις του με το ποντικι.

void changeInto3dEnviroment(void): αλλαζει σε projection προβολη ετσι ωστε ο χρήστης να μπορεί να βλέπει τρισδιαστατα τα πολυγωνα του...

bool isIntersecting(vertex p1, vertex p2, vertex q1, vertex q2): Συνάρτηση η οποία παίρνει σαν ορισμα 4 κορυφες οι οποίες αποτελουν 2 ακμες και ελεγχει αν τεμνονται. Επιστρεφει true αν τεμνονται και false αν δεν τεμνονται.

void findEdges(void): Βρίσκει τις μεχρι τοτε ακμες που αποτελουν το πολυγωνο που σχεδιάζει ο χρήστης και τις βάζει στον global πινακα edges.

void polygonCheckForIntersection(void): Συνάρτηση που ελεγχει αν ενα πολυγωνο εχει αυτοτεμνομενες ακμες. Βρίσκει τις ακμες που αποτελουν το πολυγωνο καλωντας την **findEdges()** και επειτα για καθε 2 ακμες του πολυγωνου ελεγχει αν τεμνονται καλοντας την **isIntersecting()** στην οποία περναι σαν παραμετρους τις κορυφες που αποτελουν τις ακμες.

void lastPolygonCheckForIntersection(void): Συνάρτηση που καλειται όταν ο χρήστης παταει δεξι κλικ για να σχηματισει το τελικο πολυγωνο. Ελεγχει όλες τις ακμες του τρεχοντος πολυγωνου αν τεμνονται με την τελευται ακμη που δημιουργηθηκε μετα το δεξι κλικ.

void drawPreviusPolygons(void): Ζωγραφιζει στην οθονη τα προηγουμενα πολυγωνα που εχει σχεδιασει ο χρήστης. Στην αρχη ελεγχει αν εχει σχεδιασει καποιο πολυγωνο, αν οχι δεν σχεδαζει τιποτα αλλιως σχεδιαζει τα πολυγωνα με την GL_LINE_LOOP τρεχοντας στον global πινακα **database** που ειναι αποθηκευμενα .
Αν καποιο πολυγωνο που σχεδιασε ο χρήστης εχει 3 κορυφες αυτοματως υποθετεται οτι ειναι τριγωνο οποτε σχεδιαζει με GL_TRIANGLES,

void drawCurrentPolygon(void): Συνάρτηση που σχεδιαζει τις κορυφες του τρεχοντος πολυγωνου που σχεδιαζει ο χρήστης που αποθηκευονται στον πινακα vertices με GL_POINTS και επειτα σχεδιαζει τις γραμμες που αποτελουν το πολυγωνο με GL_LINE_STRIP.

void drawPolygonsAsTriangles(void): Συνάρτηση που διατρεχει τον πινακα triangle_database με σκοπο να σχεδιασει τα πολυγωνα σαν τριγωνα ετσι ωστε να μπορουν να χρωματιστει το εσωτερικο τους.

void drawPolygonsTriangles(void): Συνάρτηση που διατρεχει τον πινακα triangle_database και σχεδιαζει τα τριγωνα των πολυγωνων ετσι ωστε να φανουν όταν ο χρήστης πατησει το T η t.

void drawClippingRectangle(void) : Συνάρτηση που σχεδιαζει στην οθονη το περιγραμμα του ορθογωνιου αποκοπης απο τις 2 κορυφες που ειναι αποθηκευμενες στον πινακα clip με χρωμα μαυρο. Σχεδιαζει το ορθογωνιο μονο αν ο χρήστης εχει δωσει τις 2 κορυφες που πρεπει.

void drawMotionClippingRectangle(void): Σχεδιαζει το διαδραστικο παραθυρο αποκοπης καθως ο χρήστης μετακινει το ποντικι .

void drawRectanglePoints(void): Απλα ζωγραφιζει τις 2 κορυφες του ορθογωνιου αποκοπης (xmin,ymin)(xmax,ymax). Τις κορυφες που αποφασιε ο χρήστης με κλικ ποντικιου.

void drawClippedPolygons(void): Συνάρτηση που σχεδιάζει τα πολυγωνα που εχουν αποκοπει και εχουν αποθηκευτει (οι κορυφες τους) στον πινακα clippedPolygons.

void drawClippedPolygonsTriangles(void): Συνάρτηση που σχεδιάζει τα τριγωνα των αποκομενων πολυγωνων μετα την τριγωνοποιηση τους.

void drawClippedPolygonsAsTriangles(void): Σχεδιάζει τα αποκομμενα πολυγωνα σαν τριγωνα ετσι ζστε να μπορεί ν γεμιστει το εσωτερικο τους.

void triangulateClippedPolygons(void): Συνάρτηση που τριγωνοποιει τα αποθηκευμενα αποκομμενα πολυγωνα καλοντας την **ProcessClippedPolygons** η οποια για τις κορυφες καθε αποκομμενου πολυγωνου που το αποτελουν και τον αριθμο των κορυφων του τριγωνοποιει και επιστρεφει στον καθολικο πινακα τις κορυφες των τριγωνων. Επειτα ο πινακας με τις κορυφες αποθηκευεται στον καθολικο πινακα **clipped_polygons_triangles** ο οποιος οπως ειπαμε περιεχει τα τριγωνα μετα την τριγωνοποιηση ολων τον αποκομμενων πολυγωνων.

void drawTextArea(void): Σχεδιάζει στην οθονη ενα ορθογωνιο στο οποιο θα φαινεται το μηκος εξωθησης που πληκτρολογει ο χρηστης.

void showScreenText(float x, float y, const char *screenText): Σχεδιάζει στην οθονη στις συντεταγμενες που δινουν οι x και y , ενα μηνυμα το οποιο λει στον χρηστη να πληκτρολογισει το μηκος εξωθησης που θελει. Για να γινει αυτο διατρεχει το string screenText χαρακτηρα χαρακτηρα και τον εκτυπωνει χρησιμοποιοντας την μεθοδο της glut **glutStrokeCharacter()**.

void showUserInput(float x, float y, vector <char> z_value): Σχεδιάζει στην οθονη το μηκος εξωθησης που πληκτρολογει εκεινη την στιγμη ο χρηστης. Για να γινει αυτο διατρεχει τον πινακα z_value και καθε χαρακτηρα τον τυπωνι στην οθονη χρησιμοποιοντας την μεθοδο **glutStrokeCharacter()**.

void draw3dimPolygons(void): Σχεδιάζει τα τρισδιαστα πολυγωνα που εχει σχεδιασει ο χρηστης. Για να γινει αυτο εχουν υπολογιστει οι ακμες του καθε πολυγωνου και για καθε ακμη και τις κορυφες που την αποτελουν σχεδιαζεται ενα quad με την GL_QUADS.

void drawScene(void): Εδω γινεται η σχεδιαση της σκηνης οπου καλουνται οι βοηθητικες συναρτησεις σχεδιασης. Οι βοηθητικες συναρτησεις καλουνται αναλογα με το mode που ειναι ο χρηστης. Αν ο χρηστης ειναι σε **PolygonMode** το οποιο σημαινει οτι ο χρηστης ειναι στην αρχικη φαση οπου σχεδιαζει πολυγωνα στην οθονη, τοτε καλεσε τις συναρτησεις που σχεδιαζουν τα πολυγωνα. Αν ειναι σε PolygonMode και οι global μεταβλητη polygon_triangles ειναι true και η μεταβλητη keyTimesPressed εχει αρτια τιμη τοτε ζωγραφισε τα τριγωνα που απαρτιζουν τα πολυγωνα και τα πολυγωνα αλλιως αν η keyTimesPressed εχει περιττη τιμη τοτε απλα ζωγραφιζει τα πολυγωνα μονο. Αν ο χρηστης τωρα ειναι σε **clippingMode** καλεσε τις βασικες συναρτησεις που σχεδιαζουν τα πολυγωνα αλλα επισης καλεσε και τις συναρτησεις που σχεδιαζουν το ορθογωνιο αποκοπης οπως και τις συναρτησεις που σχεδιαζουν τα αποκκομενα πολυγωνα. Τωρα αν ειναι σε clippingMode και ο χρηστης παταει T , αν η μεταβλητη clipped_polygon_triangles_draw ειναι true και η keyTimesPressed1 εχει αρτια τιμη ζωγραφισε τα τριγωνα των αποκκομενων πολυγωνων αλλιως απλα ζωγραφιζει τα αποκκομενα πολυγωνα. Αν ο χρηστης ειναι σε **userInputMode** καλεσε τις συναρτησεις που σχεδιαζουν το ορθογωνιο μεσα στο οποιο θα εμφανιζεται το μηκος εξωθησης που πληκτρολογει ο χρηστης. Αν ο χρηστης ειναι σε **draw3dPolygonsMode** καλει την συναρτηση που ζωγραφιζει τα 3σδιαστατα πολυγωνα Τελος δειχνει στην οθονη τα σχεδια.

void allocateSpace(void): Προσθετει επιπλεον χωρο σε καθε στις αποθηκευμενες κορυφες καθε πολυγωνου αποκοπης. Δηλαδη σε καθε πολυγωνο αποκοπης και τις κορυφες που το αποτελουν και

είναι αποθηκευμένες στον πίνακα `clippedPolygons` προσθέτει επιπλέον κορυφές στο τέλος. Αυτό είναι χρήσιμο για όταν θα γίνει η αποκοπή με την μέθοδο Sutherland-Hodgman.

void findPolygonsSizes(void): Βρίσκει και αποθηκεύει στον πίνακα `polygonsSizes` το πλήθος των κορυφών των πολυγώνων.

void processKey(unsigned char key, int x, int y): Αυτή η συνάρτηση καλείται κάθε φορά που ο χρήστης πατάει συγκεκριμένα κουμπιά στο πληκτρολόγιο. Συγκεκριμένα, αν ο χρήστης είναι σε `userInputMode`, δηλαδή σε `mode` που του ζητείται το μήκος εξώθησης, τότε αν πληκτρολογεί αριθμούς από το 1 έως το 9 αυτοί αποθηκεύονται στον πίνακα `z_deph` σαν χαρακτήρες, αν πατήσει `del` δηλαδή θέλει να σβήσει αυτό που πληκτρολόγησε τότε αφαιρείται από τον πίνακα το τελευταίο νούμερο. Τέλος αν πατήσει `Enter` αυτό σημαίνει ότι τελείωσε με την πληκτρολόγηση του μήκους εξώθησης και ο χρήστης θέλει να δει τα τρισδιάστατα πολύγωνα στην οθόνη οπότε μπαίνει σε `mode 3dPolygon`, μετατρέπεται ο πίνακας χαρακτήρων `z_deph` σε ακέραιο που αποθηκεύεται στην μεταβλητή `extrusion_length` και αλλάζει το περιβάλλον σε 3d καλώντας την συνάρτηση **changeInto3dEnvironment()**.

Αν ο χρήστης πατήσει `t h T` και είναι σε `PolygonMode` αυξάνεται η μεταβλητή `keyTimesPressed`.

Αν ο χρήστης πατήσει `t h T` και είναι σε `clippingMode` αυξάνεται η μεταβλητή `keyTimesPressed1`.

void specialKeys(int key, int x, int y): Η συνάρτηση αυτή καλείται κάθε φορά που ο χρήστης πατάει τα βελάκια. Αναλόγα με το βελάκι ανανεώνονται οι παραμετροί της `gluLookAt`.

void mouse(GLint button, GLint state, GLint x, GLint y): Η συνάρτηση αυτή καλείται κάθε φορά που ο χρήστης πατάει αριστερό ή δεξί κλικ. Συγκεκριμένα αν ο χρήστης είναι σε `mode` που μπορεί να σχεδιάζει πολύγωνο, και πατήσει αριστερό κλικ τότε η τρέχουσα κορυφή αποθηκεύεται στον `global` πίνακα `vertices` ο οποίος περιέχει τις κορυφές του τρέχοντος πολυγώνου που σχεδιάζει ο χρήστης και αυξάνονται οι κορυφές. Αν έχει δώσει παραπάνω από 3 κορυφές ο χρήστης το οποίο σημαίνει ότι έχουν δημιουργηθεί παραπάνω από 3 ακμές τότε βρες τις τρέχουσες ακμές και ελέγξε αν οι τέμνονται. Αν τέμνονται τότε διεγράψε τις κορυφές του πολυγώνου που έχει δώσει ο χρήστης και τις ακμές, οπότε ότι έχει σχεδιάσει ο χρήστης μέχρι τότε εξαφανίζεται από την οθόνη.

Αν ο χρήστης είναι σε `mode` που αποφασίζει τις κορυφές του ορθογωνίου αποκοπής τότε αποθηκεύει την πρώτη κορυφή που δίνει. Αν δώσει άλλη μια κορυφή τότε βρίσκει τις κορυφές του ορθογωνίου αποκοπής. Έπειτα πριν γίνει η αποκοπή των πολυγώνων αντιγράφεται ο πίνακας που περιέχει τις κορυφές κάθε πολυγώνου που έχει σχεδιάσει ο χρήστης σε έναν άλλο πίνακα ο οποίος στο τέλος θα περιέχει τις κορυφές των αποκκομμένων πολυγώνων, βρίσκει και αποθηκεύει το πλήθος των κορυφών των διαφορετικών πολυγώνων και καλείται η συνάρτηση που υλοποιεί τον αλγόριθμο Sutherland_Hodgman στον οποίο δίνουμε σαν ορίσματα τις κορυφές του ορθογωνίου αποκοπής και το πλήθος των κορυφών του. Έπειτα τριγωνοποιούνται τα αποκκομμένα πολύγωνα.

Αν ο χρήστης πατήσει δεξί κλικ και το πολύγωνο δεν είναι αυτοτεταμένο, αν το πολύγωνο δεν είναι τρίγωνο το τριγωνοποιεί. Αν κατά την διάρκεια της τριγωνοποίησης το πολύγωνο αποδεικνύεται ότι είναι κανονικό τότε αποθηκεύονται οι κορυφές του έπειτα αποθηκεύονται και οι κορυφές που προκύπτουν από την τριγωνοποίηση, αυξάνεται ο αριθμός των πολυγώνων που έχει σχεδιάσει ο χρήστης και βρίσκονται και οι ακμές του. Τέλος διαγράφονται οι τρέχουσες κορυφές του τρέχοντος πολυγώνου έτσι ώστε να αποθηκευτούν οι καινούργιες αν ο χρήστης αποφασίσει να σχεδιάσει άλλο πολύγωνο.

void menuOption(GLint selectedOption): Συνάρτηση που καλείται όταν ο χρήστης πατάει σε επιλογές του μενού `ACTION`. Αν πατήσει `POLYGON` μπαίνει σε `mode` όπου σχεδιάζει πολύγωνα. Επίσης αλαζονται οι μεταβλητές που ορίζουν αν ο χρήστης επέλεξε χρώμα γραμμής ή γεμίσματος σε `true` έτσι ώστε να μην γίνεται περαιτέρω αλλαγή σε γεμίσμα και γραμμή.

Αν πατήσει `CLIPPING` μπαίνει σε `mode` αποκοπής των πολυγώνων που έχει σχεδιάσει.

Αν πατήσει EXTRUDE μπαίνει σε mode όπου μπορεί να πληκτρολογήσει ένα μήκος εξώθησης.
Αν πατήσει CLEAR SCENE διαγραφονται οι πληροφορίες κορυφών πολυγώνων και τριγώνων
όπως και ακμών και μηδενίζεται ο μετρητής που καθορίζει ποσα πολυγωνα εχει σχεδιασει ο
χρηστης.