ΟΜΑΔΑ Α

1. void drawAPlane(){ glBegin(GL_QUADS);

glVertex3f(-1, -1, 0);

glVertex3f(-1, 1, 0);

glVertex3f(1, 1, 0);

glVertex3f(1, -1, 0); glEnd();

}

2. Η OpenGL υποστηρίζει το χρωματικό μοντέλο RGB, σε μια ελαφρώς εκτεταμένη μορφή. Προσθέτει μια τέταρτη συνιστώσα στο χρώμα, η οποία ονομάζεται alpha και παίζει το ρόλο “συντελεστής διαφάνειας”. Ο “συντελεστής διαφάνειας” χρησιμοποιείται όταν συνδυάζονται τα χρώματα μεταξύ τους. Το αναθεωρημένο μοντέλο ονομάζεται μοντέλο RGBA. Τα χρώματα στην OpenGL προσδιορίζονται καθορίζοντας τιμές για το χρωματικό μοντέλο RGBA μεταξύ 0.0 και 1.0. Για παράδειγμα η εντολή glColor3f (0.0f , 0.0f , 1.0f , 1.0f); // (R, G, B, A) καθορίζει το μπλέ χρώμα με πλήρη αδιαφάνεια.

3. Οι εντολές OpenGL χρησιμοποιούν το πρόθεμα gl και αρχικά κεφαλαία γράμματα για κάθε λέξη που συνθέτει το όνομα της εντολής (π.χ. gClearColor()). Πολλές από τις συναρτήσεις της δέχονται προκαθορισμένα ορίσματα (συμβολικές σταθερές) που αρχίζουν με GL_, γράφονται με κεφαλαία γράμματα και οι λέξεις διαχωρίζονται με κάτω παύλα (π.χ. GL_COLOR_BUFFER_BIT). Πολλές εντολές χρησιμοποιούν αριθμούς και γράμματα ως επίθυμα (suffix) για να μπορούμε να χρησιμοποιούμε διαφορετικό αριθμό και τύπο ορισμάτων. Π.χ. η εντολή glVertex3f(); δηλώνει ότι ως όρισματα δέχεται 3 float αριθμούς.

4. Το πρότυπο της OpenGL είναι ανεξάρτητο πλατφόρμας, περιορίζεται όμως στη δυνατότητα εμφάνισης του προγράμματος. Η OpenGL Utility Toolkit (GLUT) είναι μια βιβλιοθήκη η οποία προσφέρει εντολές εντολές εισόδου- εξόδου προσφέροντας έτσι δυνατότητα αλληλεπίδρασης του χρήστη του προγράμματος με αυτό. Η βιβλιοθήκη GLUT περιλαμβάνει εντολές απεικόνισης παραθύρων στην οθόνη, δημιουργίας menus, διαχείρισης γεγονότων κλπ. Όλες οι εντολές της ξεκινούν με το πρόθεμα glut.

5. `glOrtho (-3.0f, 3.0f, -3.0f, 3.0f, 1.0f, 100.0f);`

6. α)Τα επικρατέστερα Λειτουργικά Συστήματα (Windows, Unix, Linux, Mac OS) υποστηρίζουν την OpenGL.

Σωστό. Η OpenGL είναι ανεξάρτητη λειτουργικού συστήματος και συνήθως αλληλεπιδρά με την κάρτα γραφικών ώστε να πετύχει ταχύτερη απόδοση γραφικών.

β)Η OpenGL μπορεί αποκλειστικά να κληθεί (is callable) από τις γλώσσες προγραμματισμού C / C++ (δηλαδή υπάρχει μοναδικό language binding).

Σωστό. Μια βιβλιοθήκη που υλοποιεί το πρότυπο της OpenGL μπορεί να συνταχθεί σε οποιαδήποτε γλώσσα προγραμματισμού (η OpenGL είναι πρότυπο ανεξάρτητο πλατφόρμας).

γ) Το μοναδικό περιβάλλον ανάπτυξης προγραμμάτων OpenGL είναι το DEV C++.

Λάθος. Η OpenGL είναι ένα σύνολο εντολών (Application Programming Interface – API) που μας επιτρέπει την δημιουργία τριδιάστατων γραφικών. Δεν είναι γλώσσα προγραμματισμού αλλά μπορεί να χρησιμοποιηθεί με μια πληθώρα γλωσσών προγραμματισμού (C, C++, Java και άλλες). Συνεπώς DEV C++ δεν είναι το μοναδικό περιβάλλον ανάπτυξης προγραμμάτων.

δ) Η OpenGL περιέχει εντολές επιλογής (τύπου If ... else).

Σωστό. Η OpenGL χρησιμοποιεί μια μηχανή καταστάσεων (state machine) για να επικοινωνεί με την εφαρμογή. Σε αυτή την μηχανή καταστάσεων η OpenGL παραμένει διαρκώς σε μια κατάσταση μέχρι να αλλάξει η εφαρμογή την κατάσταση.

ε) Οι εντολές της OpenGL ξεκινούν με το πρόθεμα gl.

Σωστό. Οι εντολές OpenGL χρησιμοποιούν το πρόθεμα gl και αρχικά κεφαλαία γράμματα για κάθε λέξη που συνθέτει το όνομα της εντολής (π.χ. gClearColor()). Πολλές από τις συναρτήσεις της δέχονται προκαθορισμένα ορίσματα (συμβολικές σταθερές) που αρχίζουν με GL_, γράφονται με κεφαλαία γράμματα και οι λέξεις διαχωρίζονται με κάτω παύλα (π.χ. GL_COLOR_BUFFER_BIT).

7. Η glutInitWindowPosition(int x, int y); καθορίζει τη θέση στην οθόνη, στην οποία θα εμφανιστεί το παράθυρο της εφαρμογής (συντεταγμένη της άνω αριστερής κορυφής) όπου x παράμετρος η απόσταση του παραθύρου σε pixels από την πάνω αριστερή γωνία της οθόνης στον άξονα X και y παράμετρος η απόσταση του παραθύρου σε pixels από την πάνω αριστερή γωνία της οθόνης στον άξονα Y . π.χ. glutInitWindowPosition(150,150); Το παράθυρο θα τοποθετηθεί 150,150 pixels από την πάνω αριστερή γωνία της οθόνης.

Η glutInitWindowSize(int width, int height); καθορίζει το πλάτος και ύψος του παραθύρου της εφαρμογής σε pixels όπου width το πλάτος του παραθύρου σε pixels και height το μήκος του παραθύρου σε pixels. Π.χ. glutInitWindowSize(400, 400); // Το παράθυρο θα έχει μέγεθος 400×400 pixels.

Η glutCreateWindow(char *name); εμφανίζει το παράθυρο της εφαρμογής στην οθόνη και του αποδίδει έναν τίτλο.

Π.χ. glutCreateWindow("ΙΕΚ ΔΗΜΗΤΡΑ"); // Το παράθυρο θα έχει τον τίτλο ΙΕΚ ΔΗΜΗΤΡΑ

8. OpenGL core library

Βασική βιβλιοθήκη (OpenGL core library ή GL):

Η βασική βιβλιοθήκη της OpenGL έχει σχεδιαστεί ως μια βελτιωμένη διεπαφή ανεξάρτητη από το hardware, περιέχει τις κύριες εντολές σχεδίασης όπως η σχεδίαση βασικών γεωμετρικών σχημάτων, ο ορισμός χρωμάτων κλπ. Όλες οι εντολές της βιβλιοθήκης αυτής διακρίνονται από το πρόθεμα gl.

Πολλές από τις συναρτήσεις της δέχονται προκαθορισμένα ορίσματα (συμβολικές σταθερές) τα οποία έχουν οριστεί στη βιβλιοθήκη και αντιστοιχούν σε διάφορες παραμέτρους ή καταστάσεις λειτουργίας. Οι σταθερές αυτές ξεκινούν με το πρόθεμα GL_.

OpenGL Utility Library (GLU):

Η βιβλιοθήκη GLU είναι χτισμένη πάνω στην κορυφή της OpenGL και περιλαμβάνει συναρτήσεις που εκτελούν σύνθετους αλγορίθμους όπως π.χ. τον καθορισμό μητρώων προβολής και το σχηματισμό σύνθετων καμπυλών και επιφανειών. Κάθε υλοποίηση της OpenGL εμπεριέχει τη βιβλιοθήκη GLU. Όλες οι εντολές της βιβλιοθήκης GLU ξεκινούν με το πρόθεμα glu_ (π.χ., gluLookAt, gluPerspective).

OpenGL Utility Toolkit (GLUT):

Η OpenGL είναι ένας γρήγορος και ευέλικτος τρόπος για την επικοινωνία με το hardware των γραφικών του υπολογιστή χωρίς να ενδιαφέρουν το χρήστη οι λεπτομέρειες υλοποίησης του. Από την άλλη, δεν προσφέρει καθόλου λειτουργίες GUI (Graphical User Interface), δηλαδή δεν έχει τη δυνατότητα να ανοίξει και να κλείσει παράθυρα στο λειτουργικό σύστημα, να ζωγραφίσει σε αυτά, ούτε να καταλάβει το πάτημα ενός πλήκτρου ή την κίνηση του ποντικιού, ούτε μπορεί να διαβάσει ένα αρχείο από το δίσκο. Μια από τις πιο διαδεδομένες βιβλιοθήκες για αυτό το σκοπό είναι το GLUT (OpenGL Utility Toolkit) που είναι και αυτή σχεδιασμένη (η εργαλειοθήκη) να τρέχει σε πολλά λειτουργικά συστήματα. Η GLUT προσφέρει ένα σύνολο εντολών που αναλαμβάνουν να ανοίξουν και να κλείσουν εύκολα παράθυρα, να καταγράψουν το πάτημα ενός πλήκτρου ή την κίνηση

του ποντικιού). Η βιβλιοθήκη αυτή περιλαμβάνει ακόμη εντολές απεικόνισης παραθύρων στην οθόνη, δημιουργία μενού, κλπ. Όλες οι εντολές της ξεκινούν με το πρόθεμα glut_.

Οι βιβλιοθήκες που χρησιμοποιούνται δηλώνονται στο πάνω μέρος του προγράμματος από τις εντολές

```
#include <gl\gl.h>
#include <gl\glu.h>
 #include <gl\glut.h>
```

9. Η glutMainLoop() ενεργοποιεί τον κύκλο διαχείρισης γεγονότων (event processing loop). Στον κύκλο αυτό, η εφαρμογή αναμένει επ' άπειρον και ανταποκρίνεται σε γεγονότα, όπως λ.χ. στο πάτημα ενός κουμπιού, στην αλλαγή του σκηνικού ή στην κίνηση του ποντικιού. Από το πρόθεμά της, παρατηρούμε ότι η συγκεκριμένη εντολή εμπεριέχεται στη βιβλιοθήκη GLUT, αφού το πρότυπο της OpenGL, ως πρότυπο ανεξάρτητο πλατφόρμας, δεν ορίζει διαδικασίες εισόδου - εξόδου.

ΟΜΑΔΑ Β

1. Η βασική δομή μιας εφαρμογής σε OpenGL αποτελείται απο δύο κυρίως κομμάτια. Το πρώτο κομμάτι είναι η αρχικοποίηση όπου καθορίζονται οι παράμετροι και οι βασικές μεταβλητές απαραίτητες για την δημιουργία της σκηνής. Οι παράμετροι αυτοί σχετίζονται με το μέγεθος της οθόνης, τα χρώμματα, το είδος της σκηνής(τρείς ή δύο διαστάσεις) κτλ. Σε δεύτερη φάση μια OpenGL εφαρμογή δημιουργεί την σκηνή συνεχώς και αλληλεπιδρά με τον χρήστη. Εδώ δηλαδή, έχουμε συναρτήσεις όπου δημιουργούν κάθε ένα Pixel στην οθόνη, αλληλεπιδρούν με το πληκτρολόγιο και το ποντίκι και είναι υπεύθυνες για το resize του παραθύρου.

2. Αναφορικά έχουμε τα:

1 glutDisplayFunc
2 glutOverlayDisplayFunc
3 glutReshapeFunc
4 glutKeyboardFunc
5 glutMouseFunc
6 glutMotionFunc, glutPassiveMotionFunc
7 glutVisibilityFunc
8 glutEntryFunc
9 glutSpecialFunc
10 glutSpaceballMotionFunc
11 glutSpaceballRotateFunc
12 glutSpaceballButtonFunc
13 glutButtonBoxFunc
14 glutDialsFunc
15 glutTabletMotionFunc
16 glutTabletButtonFunc
17 glutMenuStatusFunc
18 glutIdleFunc
19 glutTimerFunc

3. Η συνάρτηση glTranslate πολλαπλασιάζει τον τρέχων πίνακα με έναν translation πίνακα. Η συνάρτηση αυτή παράγει ένα translation πίνακα βάσει των x y z. Ο τρέχων πίνακας πολλαπλασιάζεται με αυτόν το translation πίνακα και το γινόμενο αντικαθιστά τον τρέχοντα πίνακα. Έχει τις συναρτήσεις C, void glTranslated(GLdouble x,GLdouble y,GLdouble z) και void glTranslatef(GLfloat x,GLfloat y,GLfloat z),όπου οι παράμετροι x, y, z καθορίζουν τις x, y,και z συντεταγμένες ενός translation vector.

Η συνάρτηση glRotate πολλαπλασιάζει τον τρέχων πίνακα με έναν rotation πίνακα. Η συνάρτηση αυτή παράγει μια επριστροφή της γωνίας των αξόνων περί το διάνυσμα x,y,z. Ο τρέχων πίνακας πολλαπλασιάζεται με αυτόν το rotation πίνακα και το γινόμενο αντικαθιστά

τον τρέχοντα πίνακα. Έχει τις συναρτήσεις C,void glRotated(GLdouble angle,GLdouble x,GLdouble y,GLdouble z) και void glRotatef(GLfloat angle,GLfloat x,GLfloat y,GLfloat z); όπου οι παράμετροι angle, x, y, z καθορίζουν τη γωνία περιστροφής και τις x, y, z συντεταγμένες αντίστοιχα.

Η συνάρτηση glScale πολλαπλασιάζει τον τρέχων πίνακα με έναν γενικό scaling πίνακα. Η συνάρτηση αυτή παράγει μια ανομοιόμορφη κλιμάκωση (scaling) στους x,y,z άξονες. Οι τρεις παράμετροι x,y,z καθορίζουν την επιθυμητή κλιμάκωση κατά μήκος των x,y,z αξόνων. Έχει τις συναρτήσεις C, void glScaled(GLdouble x, GLdouble y, GLdouble z) και void glScalef(GLfloat x,GLfloat y,GLfloat z) ,όπου οι παράμετροι x, y, z των x,y,z αξόνων αντίστοιχα. Επιπρόσθετα, η οι συναρτήσεις glViewport glortho αξιοποιούνται στο να πολλαπλασιαστού με έναν πίνακα που ορίζει πόσο stretch πρέπει να γίνει. Το ορθό ορίζει το νέο ορθοκανονικό σύστημα συντεταγμένων και το viewport αλλάζει το view που θα έχει το object πανω στο νέο σύστημα συντεταγμένων.

4. Με τους τύπους δεδομένων της OpenGL επιτυγχάνεται κυρίς καλύτερο compatibility γιατί η OpenGL είναι cross-platform δηλαδή ανεξάρτητη λειτουργικού, και ως εκ τούτου, δεν απαιτείται περεταίρω κώδικας ή compile.

Εντοπίζονται κυρίως έξι βασικοί τύποι δεδομένων, μερικοί από αυτούς είναι οι ίδιοι με τους C, άλλοι προστίθενται ειδικά για προγραμματισμό GPU, αυτοί οι τύποι είναι:

float - ένας αριθμός κινητής υποδιαστολής 32 bit hald - έναν αριθμό κινητής υποδιαστολής 16 bit int - ένας ακέραιος 32-bit

fixed - αριθμός σταθερού σημείου 12-bit bool - μια μεταβλητή boolean

sampler * - αντιπροσωπεύει ένα αντικείμενο υφής

Το Cg διαθέτει επίσης τύπους δεδομένων vector και matrix που βασίζονται στους βασικούς τύπους δεδομένων, όπως float3, float4*4, αυτοί οι τύποι δεδομένων είναι αρκετά συνηθισμένοι όταν αξιοποιούνται στον προγραμματισμό τρισδιάστατων γραφικών, το Cg έχει επίσης τύπους δεδομένων struct και array, οι οποίοι λειτουργούν σε παρόμοια τρόπο ισοδύναμα C.

5. Η gluLookAt δίνει την δυνατότητα να τοποθετήσουμε μια κάμερα στην σκηνή και να θέσουμε την κατεύθυνση της. Συντάσσεται ως εξής:
gluLookAt (GLdouble eyex, GLdouble eyey, GLdouble eyez, GLdouble centerx, GLdouble centery, GLdouble centerz, GLdouble upx, GLdouble upy, GLdouble upz);
Η εντολή παίρνει ως παραμέτρους 3 διανύσματα: το [eyex, eyey, eyez] ορίζει τις συντεταγμένες του σημείου θέασης (μάτι), το [centerx, centery, centerz] καθορίζει την κατεύθυνση (τι δείχνει δηλαδή) και το [upx, upy, upz] είναι το μοναδιαίο διάνυσμα που καθορίζει ποια είναι η προς τα πάνω κατεύθυνση της κάμερα (ποιος ο άξονας y).

6. Η glClear() είναι μια συνάρτηση της OpenGl που καθαρίζει τα buffers(ενταμιευτές) και επαναφέρει τις τιμές που υπήρχαν πριν οποιαδήποτε αλλαγή. Ανάλογα με τα buffers που θέλουμε να καθαρίσουμε βάζουμε και τα ανάλογα ορίσματα. Η glClear μπορεί να καθαρίσει ένα και περισσότερα buffersa ταυτόχρονα, με την χρήση με το σύμβολο του bitwise OR.
π.χ. glClear(GL_COLOR_BUFFER_BIT); // καθαρίζεται ο buffer του χρώματος.
Η εντολή glClearColor3f() δεν υπάρχει από όσο γνωρίζω, θα απαντήσω για την εντολή glClearColor(). Η glClearColor() είναι εντολή καθαρισμού της οθόνης. Καθορίζει το χρώμα που χρησιμοποιείται κάθε φορά που εκτελείται. Στην OpenGL το χρώμα του φόντου είναι μία μεταβλητή κατάστασης, η οποία διατηρεί την τιμή που της ανατέθηκε την τελευταία φορά. Το χρώμα καθορίζεται από τα βάρη του στο χρωματικό μοντέλο RGBA. Ως παραμέτρους καταχωρούμε τις τιμές σε εύρος [0,1] του κόκκινου, του πράσινου, του μπλε και της διαφάνειας.
π.χ. glClearColor(1,1,1,1); // Για να χρησιμοποιήσουμε το λευκό χρώμα

7. Στην OpenGL όπως και στα περισσότερα API γραφικών χρησιμοποιούμε κανονικοποιημένες τιμές (normalized values) για τις διαστάσεις και τη θέση των αντικειμένων, που σημαίνει ότι μια τιμή 1 υποδηλώνει ολόκληρο το πλάτος ή το ύψος της οθόνης (ανάλογα με την παράμετρο) ανεξάρτητα από την ανάλυση οθόνης. Αυτό επιτρέπει την εύκολη διατήρηση της αναλογίας διαστάσεων γραφικών αντικειμένων, ανεξάρτητα από τη συσκευή προβολής.

8. glViewport(xmin, ymin, width, height); //perspective

glOrtho (Left, Right, Bottom, Top, Near, Far); //orthographic

9. Μετακίνηση (translation) ενός αντικειμένου με την glTranslatef(GLfloat X, GLfloat Y, GLfloat Z). Η εντολή αυτή παίρνει ως παραμέτρους την μετακίνηση που επιθυμούμε κατά άξονα, κατασκευάζει ένα μετασχηματισμό μετακίνησης T και τον εφαρμόζει (πολλαπλασιάζει) με τον μετασχηματισμό Μοντέλου-Κάμερας. Ένας άλλος τρόπος είναι να μετατοπίσουμε την κάμερα.

10. GL_POINTS Σχεδιάζει σημεία. Κάθε κορυφή είναι ένα σημείο.

```
glBegin(GL_POINTS);   //starts   drawing   of   points
     glVertex3f(1.0f,1.0f,0.0f);//upper-right   corner
     glVertex3f(-1.0f,-1.0f,0.0f);//lower-left corner
glEnd();//end drawing of points
```

GL_LINES Σχεδιάζει γραμμές. Κάθε ζεύγος κορυφών σχεδιάζει μια γραμμή.

```
glBegin(GL_LINES);
glVertex2i(20,20);
```

```
glVertex2i(40,40);
glEnd();
```

GL_LINE_STRIP Σχεδιάζει ενωμένες γραμμές. Κάθε κορυφή μετά τις δύο πρώτες ενώνεται.

```
glBegin(GL_LINE_STRIP);
 glVertex3f(0.2, 0.2,0.0);
 glVertex3f(0.8, 0.2,0.0);
  glVertex3f(0.2, 0.5,0.0);
  glVertex3f(0.8, 0.5,0.0);
 glVertex3f(0.2, 0.8,0.0);
 glVertex3f(0.8,      0.8,0.0);
glEnd();
```

GL_LINE_LOOP Σχεδιάζει ενωμένες γραμμές. Η τελευταία κορυφή ενώνεται με την πρώτη.

```
glBegin(GL_LINE_LOOP);//start drawing a line loop
    glVertex3f(-1.0f,0.0f,0.0f);//left  of  window
    glVertex3f(0.0f,-1.0f,0.0f);//bottom of window
    glVertex3f(1.0f,0.0f,0.0f);//right  of  window
    glVertex3f(0.0f,1.0f,0.0f);//top of window
  glEnd();//end drawing of line loop
```

GL_TRIANGLES Κάθε τρεις κορυφές σχεδιάζουν ένα τρίγωνο.

```
 glBegin(GL_TRIANGLES);//start drawing triangles
    glVertex3f(-1.0f,-0.25f,0.0f);//triangle  one  first  vertex
    glVertex3f(-0.5f,-0.25f,0.0f);//triangle one second vertex
    glVertex3f(-0.75f,0.25f,0.0f);//triangle one third vertex
  glEnd();//end drawing of triangles
```

GL_TRIANGLE_STRIP Σχεδιάζει συνδεδεμένα τρίγωνα. Κάθε κορυφή μετά τις τρεις πρώτες σχεδιάζει τρίγωνο.

```
glBegin(GL_TRIANGLE_STRIP);
 glVertex3f(0.2, 0.2,0.0);
 glVertex3f(0.8, 0.2,0.0);
  glVertex3f(0.2, 0.5,0.0);
  glVertex3f(0.8, 0.5,0.0);
```

```
   glVertex3f(0.2, 0.8,0.0);
   glVertex3f(0.8,      0.8,0.0);
  glEnd();
```

GL_TRIANGLE_FAN Όπως και η GL_TRIANGLE_STRIP, σχεδιάζει σε σχήμα βεντάλιας.

```
glBegin(GL_TRIANGLE_FAN);
  glVertex3f(0.1, 0.1,0.0);
  glVertex3f(0.6, 0.1,0.0);
   glVertex3f(0.8,0.3,0.0);
  glVertex3f(0.6,0.6,0.0);
  glVertex3f(0.1,0.6,0.0);
  glVertex3f(0.0,0.3,0.0);
 glEnd();
```

GL_QUADS Σχεδιάζει τετράπλευρα. Κάθε τέσσερις κορυφές σχεδιάζουν ένα τετράπλευρο.

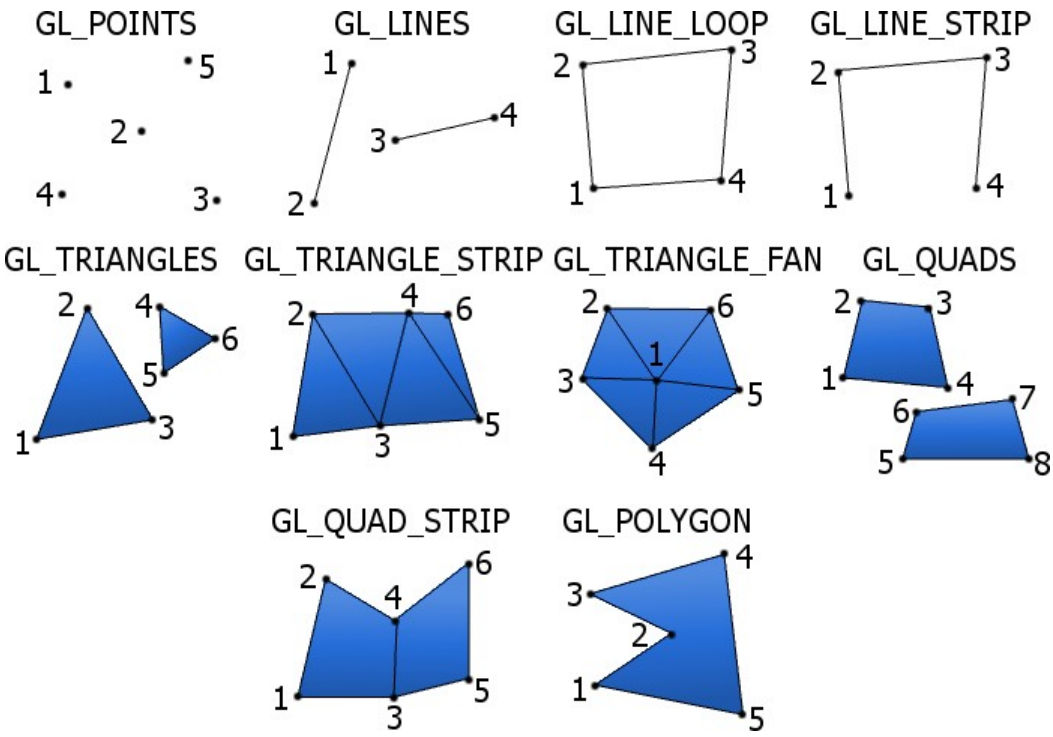
```
 glBegin(GL_QUADS);
  glVertex3f(0.2, 0.2,0.0);
  glVertex3f(0.8, 0.2,0.0);
   glVertex3f(0.6,0.4,0.0);
  glVertex3f(0.4,0.4,0.0);
  glVertex3f(0.4,0.6,0.0);
  glVertex3f(0.6,0.6,0.0);
  glVertex3f(0.8,0.8,0.0);
  glVertex3f(0.2,0.8,0.0);
 glEnd();
```

GL_QUAD_STRIP Κάθε δύο κορυφές μετά τις 4 πρώτες σχεδιάζουν ένα συνδεδεμένο τετράπλευρο.

```
glBegin(GL_QUAD_STRIP);
  glVertex3f(0.2, 0.2,0.0);
  glVertex3f(0.8, 0.2,0.0);
   glVertex3f(0.2,0.4,0.0);
  glVertex3f(0.8,0.4,0.0);
  glVertex3f(0.2,0.8,0.0);
  glVertex3f(0.8,0.8,0.0);
 glEnd();
```

GL_POLYGON Πολύγωνο. Το πολύγωνο μπορεί να έχει όσες πλευρές θέλετε.

```
    glBegin(GL_POLYGON);//begin drawing of polygon
      glVertex3f(-0.5f,0.5f,0.0f);//first   vertex
      glVertex3f(0.5f,0.5f,0.0f);//second   vertex
      glVertex3f(1.0f,0.0f,0.0f);//third    vertex
      glVertex3f(0.5f,-0.5f,0.0f);//fourth  vertex
      glVertex3f(-0.5f,-0.5f,0.0f);//fifth  vertex
      glVertex3f(-1.0f,0.0f,0.0f);//sixth vertex
    glEnd();//end drawing of polygon
```



11.
```
void keyboard (unsigned char key, int x, int
        y) { switch (key) {
                case 'q':
                case 'Q':
                        exit(0);
                break;
                case 'c':
                case 'C':
                        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
                break;
        }
}
int main(int argc, char** argv) {
        ...
        glutKeyboardFunc(keyboard);
        ....
        glutMainLoop();
        return 0;
}
```

12. Η OpenGL χρησιμοποιεί μια μηχανή καταστάσεων (state machine) για να επικοινωνεί με την εφαρμογή. Σε αυτή την μηχανή καταστάσεων η OpenGL παραμένει διαρκώς σε μια κατάσταση μέχρι να αλλάξει η εφαρμογή την κατάσταση. Παράδειγμα αν θέσουμε το χρώμα που θα χρησιμοποιεί η OpenGL για να ζωγραφίσει ένα μοντέλο, το χρώμα θα παραμείνει στη μνήμη και θα χρησιμοποιείται μέχρι να το αλλάξουμε ή να κλείσουμε την εφαρμογή.

Εφόσον λοιπόν η εφαρμογή καθορίσει το περιβάλλον που θα χρησιμοποιήσει η OpenGL για να απεικονίσει το μοντέλο μας (χρώματα, υφές, πηγές φωτός, κάμερα κλπ), περνάει στο πρώτο στάδιο κατά το οποίο θα μετασχηματίσει και θα φωτίσει (transform and lighting) τα σημεία(vertices) του μοντέλου.

Στην συνέχεια η OpenGL περνά στο στάδιο της ψηφιοποίησης το οποίο λαμβάνει όλες τις πληροφορίες και την γεωμετρία από το προηγούμενο στάδιο (του μετασχηματισμού) και παράγει την τελική, ψηφιακή, εικόνα. Η εικόνα αντιγράφεται στην μνήμη του frame buffer και φτάνει έτσι στην οθόνη του υπολογιστή.

Η OpenGL είναι ένας γρήγορος και ευέλικτος τρόπος να επικοινωνούμε με το hardware γραφικών του υπολογιστή χωρίς να ενδιαφερόμαστε για τις λεπτομέρειες υλοποίησης του. Όμως δεν προσφέρει καθόλου λειτουργίες GUI (Graphical User Interface), δηλαδή δεν έχει την δυνατότητα να ανοίξει και να κλείσει παράθυρα στο λειτουργικό σύστημα, να ζωγραφίσει σε αυτά, ούτε να καταλάβει το πάτημα ενός πλήκτρου ή την κίνηση του ποντικιού, ούτε μπορεί να διαβάσει ένα αρχείο από το δίσκο.

Αυτό έγινε επί σκοπού, μιας και η OpenGL σχεδιάστηκε να τρέχει σε πολλά λειτουργικά συστήματα τα οποία έχουν το δικό τους τρόπους επικοινωνίας με την οθόνη, το ποντίκι, το δίσκο, το πληκτρολόγιο.

Για να γίνει αυτό θα πρέπει να χρησιμοποιήσουμε απευθείας τις εντολές του λειτουργικού μας συστήματος (Win32 εντολές στην περίπτωση των Windows). Εναλλακτικά μπορούμε να χρησιμοποιήσουμε μια από τις έτοιμες βιβλιοθήκες εντολών που υπάρχουν και που θα κάνουν αυτές τις λειτουργίες για εμάς εύκολα.

Μια από τις πιο διαδεδομένες βιβλιοθήκες για αυτό το σκοπό είναι το GLUT (OpenGL Utility Toolkit) το οποίο είναι και αυτό σχεδιασμένο να τρέχει σε πολλά λειτουργικά συστήματα.

Το GLUT προσφέρει ένα σύνολο εντολών που αναλαμβάνουν να ανοίξουν και να κλείσουν εύκολα παράθυρα, να καταγράψουν το πάτημα ενός πλήκτρου ή την κίνηση του ποντικιού. Δεν είναι κατάλληλο να γράψουμε μια κανονική εφαρμογή με αυτό (πχ ένα παιχνίδι) όμως είναι πολύ κατάλληλο για εκπαιδευτικές και πρότυπες εφαρμογές

13. Στον πραγματικό κόσμο, ένας παρατηρητής αντιλαμβάνεται την παρουσία επιφανειών από τις ανακλάσεις που αυτές προκαλούν στις ακτίνες των πηγών φωτισμού. Κάθε επιφάνεια γίνεται αντιληπτή με διαφορετικά χαρακτηριστικά ανάλογα με τα χαρακτηριστικά της ανακλαστικότητάς της (χρώμα). Επιπλέον όμως, παίζει ρόλο και η σχετική θέση της κάθε επιφάνειας ως προς τις υπάρχουσες

πηγές φωτισμού. Συνοψίζοντας μπορούμε να πούμε ότι οι βασικές ιδιότητες, οι οποίες καθορίζουν τα χαρακτηριστικά των επιφανειών είναι:

- Το χρώμα
- Προσανατολισμός επιφανείας (NORMALIZATION). Προκειμένου να περιγράψουμε τον προσανατολισμό μιας επιφάνειας σε κάθε σημείο της, χρησιμοποιούμε την έννοια του κανονικού διανύσματος.
- Σκίαση Gouraud, κάθε στοιχειώδης πολυγωνική επιφάνεια χαρακτηρίζεται από το κανονικό της διάνυσμα, το οποίο χρησιμοποιείται για υπολογισμούς από το μοντέλο φωτισμού.
- Υφή. Στις περιπτώσεις απόδοσης φυσικών σκηνών με περίπλοκη μορφολογία, καταφεύγουμε στη σχεδιάση ρεαλιστικών επιφανειών βασιζόμενοι σε προκαθορισμένα πρότυπα επιφανειών. Τα πρότυπα αυτά αποδίδουν με πιστότητα τη μο-ρφολογία των επιφανειών και μπορούν να ληφθούν με τη μορφή ψηφιοποιημένων εικόνων. Στον κόσμο των γραφικών σε Η/Υ ένα τέτοιο πρότυπο απόδοσης αποκαλείται υφή (texture) και η διαδικασία της απόδοσης υφής σε μία επι-φάνεια ονομάζεται απόδοση υφής (texture mapping).
-

14. Στην επιστήμη των υπολογιστών, το double buffering είναι μια τεχνική για την σχεδίαση γραφικών, με την οποία αποκλείουμε εξολοκλήρου (ή σχεδόν εξολοκλήρου) το τραύλισμα, το τρεμοπαίξιμο, το σχίσιμο και άλλα τεχνουργήματα που δημιουργούνται κατά τη σχεδίαση και αναπαράσταση γραφικών.

Η χρήση διπλού buffer (double buffering) κατά την απεικόνιση μιας σκηνής είναι μια βασική μέθοδος αποφυγής «τρεμοπαίγματος»(flickering). Ο τρόπος με τον οποίο δουλεύει είναι ορίζοντας 2 ενδιάμεσες περιοχές στην μνήμη (buffers) στις οποίες αποθηκεύουμε τις εικόνες που δημιουργούμε. Όσο η οθόνη παίρνει (απεικονίζει) την εικόνα από τον έναν buffer εμείς

ζωγραφίζουμε την σκηνή στον άλλον. Έπειτα ανταλλάσουμε τους ρόλους των buffer και συνεχίζουμε. Έτσι η οθόνη δείχνει πάντα μια ολοκληρωμένη εικόνα της σκηνής.

Η εντολή glutSwapBuffers(); θα αντικαταστήσει τον buffer που χρησιμοποιεί η οθόνη με αυτόν που περιέχει το αντικείμενο που μόλις ζωγραφίσαμε. Ανάλογα με αυτόν που χρησιμοποιούμε καθορίζουμε στη συνάρτηση αν θα είναι ο front ή ο back buffer.

Η εντολή glutInitDisplayMode(GLUT_DOUBLE); θα ενεργοποιήσει τη λειτουργία του double buffer. Το RGB mode είναι από το πρόγραμμα αρχικοποιημένο.

15. Μια συνάρτηση κλήσης είναι μια λειτουργία την οποία καλεί η βιβλιοθήκη (GLUT) όταν χρειάζεται να ξέρει πώς να επεξεργαστεί κάτι. Για παράδειγμα όταν η GLUT λαμβάνει μια εισαγωγή κλειδιού από το πληκτρολόγιο ή το ποντίκι χρησιμοποιεί τη συνάρτηση κλήσης glutKeybourdFunc για να μάθει τι να κάνει με το εν λόγω πάτημα πλήκτρων.

Αυτή η τεχνική χρησιμοποιείται επίσης για ζωντανή ανατροφοδότηση/ μετάδοση, όπου καλεί το TrackControl (ή κάτι παρόμοιο) και καθορίζει μια ρουτίνα που θα καλέσει επανειλημμένα για να μετακινηθεί ο χρήστης στη σελίδα.

Συνεπώς χρησιμοποιούνται για τις παρακάτω διαδικασίες:

- Αναπροσαρμογή μεγέθους ή ανασύνταξη παραθύρου
- Εισαγωγή κλειδιού από χρήστη (ποντίκι, πληκτρολόγιο)
- Animation (καθιστούν πολλά πλαίσια)

Η βασικη τους δομή είναι όνομα_συνάρτησης(όνομα_συνάρτησης_γεγονότος που θα διαχειρηστεί), π.χ. glutKeyboardFunc(my_key_events_func); – glutMouseFunc (my_mouse_events_func);

16. Οι συναρτήσεις με τις οποίες γίνεται μοντελοποίηση επιφανειών που ανακλούν στην OpenGl είναι οι παρακάτω:

genType reflect(genType *I*, genType *N*);

genDType reflect(genDType *I*, genDType *N*);

17. Το φως πηγής κατά την μοντελοποίηση πηγών φωτισμού χωρίζεται σε 3 συνιστώσες, οι οποίες είναι οι εξής:

- Ένταση και χρώμα πηγής
- Θέση πηγής
- Κατεύθυνση εκπομπής - Γωνία αποκοπής

18. Η OpenGL χρησιμοποιεί μια μηχανή καταστάσεων (state machine) για να επικοινωνεί με την εφαρμογή. Σε αυτή την μηχανή καταστάσεων η OpenGL παραμένει διαρκώς σε μια κατάσταση μέχρι να αλλάξει η εφαρμογή την κατάσταση. Παράδειγμα αν θέσουμε το χρώμα που θα χρησιμοποιεί η OpenGL για να ζωγραφίσει ένα μοντέλο, το χρώμα θα παραμείνει στη μνήμη και θα χρησιμοποιείται μέχρι να το αλλάξουμε ή να κλείσουμε την εφαρμογή. Εφόσον λοιπόν η εφαρμογή καθορίσει το περιβάλλον που θα χρησιμοποιήσει η OpenGL για να απεικονίσει το μοντέλο μας (χρώματα, υφές, πηγές φωτός, κάμερα κλπ), περνάει στο πρώτο στάδιο κατά το οποίο θα μετασχηματίσει και θα φωτίσει (transform and lighting) τα σημεία(vertices) του μοντέλου. Στην συνέχεια η OpenGL περνά στο στάδιο της ψηφιοποίησης το οποίο λαμβάνει όλες τις πληροφορίες και την γεωμετρία από το προηγούμενο στάδιο (του μετασχηματισμού) και παράγει την τελική, ψηφιακή, εικόνα. Η εικόνα αντιγράφεται στην μνήμη του frame buffer και φτάνει έτσι στην οθόνη του υπολογιστή. Αυτή η ακολουθία βημάτων που ακολουθεί η OpenGL για να απεικονίσει το μοντέλο λέγεται graphics pipeline (διασωλήνωση).

19. Η ανάθεση συντεταγμένων υφής γίνεται με τη χρήση της εντολής glTexCoord{1,2}{i,s,f,d}.
Για ορίσματα κινητής υποδιαστολής (GLfloat) οι συναρτήσεις είναι:

void glTexCoord1f (GLfloat s); για μμονοδιάστατες υφές

ή

void glTexCoord2f (GLfloat s, GLfloat t); για δισδιάστατες υφές.

20. Συχνά, είναι βολικό η περιγραφή ενός σύνθετου γεωμετρικού σήματος να περικλείεται σε μια αυτόνομη ενότητα κώδικα, η οποία θα εκτελείται κάθε φορά που θέλουμε να σχεδιάσουμε αυτό το σχήμα. Στην OpenGL τη δυνατότητα αυτή δίνουν οι λίστες απεικόνισης (display lists). Οι λίστες απεικόνισης διευκολύνουν την επαναχρησιμοποίηση κώδικα και απαλλάσσουν τον προγραμματιστή από περιττές επαναλαμβανόμενες δηλώσεις του ίδιου σύνθετου σχήματος. Μια λίστα απεικόνισης περικλείεται μεταξυ δύο εντολών: των glNewList και glEndList. Όταν σε ένα πρόγραμμα ορίζονται πολλαπλές λίστες, μια αυτόνομη λίστα απεικόνισης διακρίνεται από τις υπόλοιπες βάσει ενός ακεραίου αριθμού που παίζει το ρόλο του “αναγνωριστικού αριθμού” της (list identifier).

Προκειμένου να αποδώσουμε αναγνωριστικούς αριθμούς σε λίστες απεικόνισης, πρέπει να δεσμεύσουμε το απαιτούμενο εύρος τιμών. Η δέσμευση αυτή γίνεται με την εντολή glGenLists().

GLuint glGenLists(GLint range);

Όπου range το πλήθος των αναγνωριστικών που θέλουμε να χρησιμοποιήσουμε. Η συνάρτηση επιστρέφει μια ακέραιη τιμή που αντιστοιχεί στον πρώτο αναγνωριστικό αριθμό.

Π.χ. με τη σύνταξη: listID=glGenLists(2); παράγουμε 2 identifiers. Ο πρώτος έχει ακέραιη τιμή listID, ο δεύτερος listID+1 και ούτω καθ’ εξής. Η ανάθεση αναγνωριστικής τιμής σε μια λίστα απεικόνισης γίνεται κατά την έναρξη της δήλωσής της στην εντολή glNewList:

void glNewList(GLuint listID, GLenum listMode); Π.χ. σύνταξη λίστας απεικόνισης:

glNewList();// Εντολές δήλωσης σχήματος glEndList();Μεταξύ των εντολών glNewList και glEndList ορίζουμε το σύνθετο γεωμετρικό σχήμα της λίστας απεικόνισης, χρησιμοποιώντας τις εντολές σχεδιάσης σχημάτων που αναφέραμε παραπάνω.