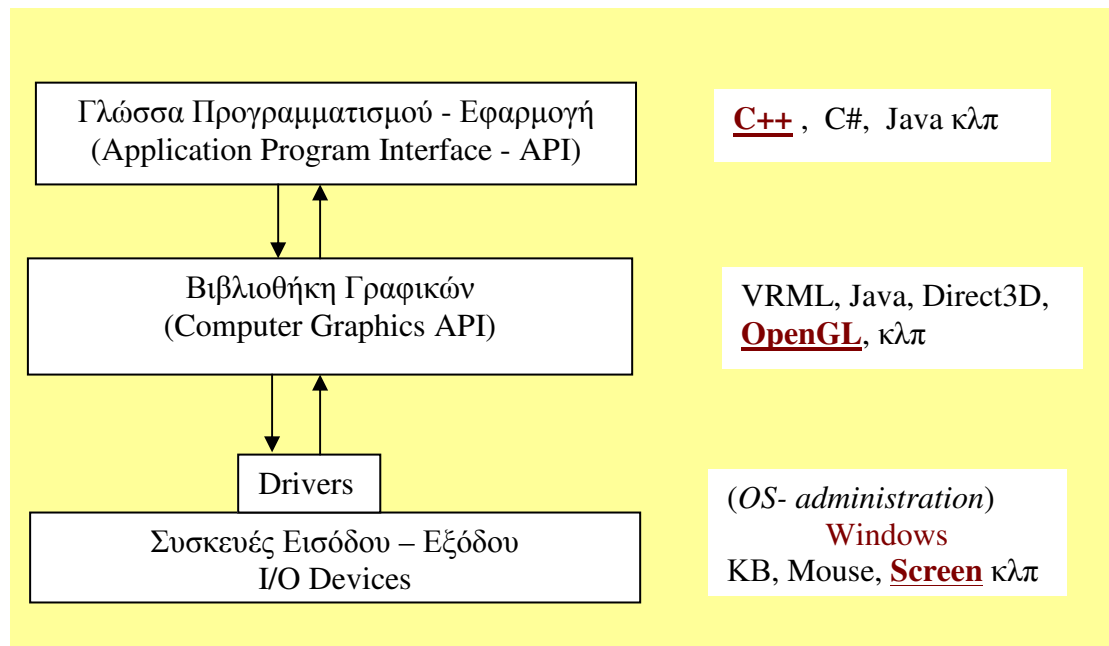




ΓΡΑΦΙΚΑ ΥΠΟΛΟΓΙΣΤΩΝ

- Υλικό
- **Λογισμικό**
 - Ειδικά πακέτα σχεδιασμού
 - **Γενικά προγραμματιστικά πακέτα**
 - Ενσωματωμένα σε μια γλώσσα προγραμματισμού
 - Ξεχωριστές Βιβλιοθήκες



Θέματα Γραφικών Η/Υ

- Θεωρία Αλγορίθμων σχεδίασης
- Μοντέλα Σχεδίασης (Imaging Models)
- Υπολογιστική Αρχιτεκτονική (Pipeline Architecture)
- **Προγραμματισμός**

Απλά παραδείγματα OpenGL : [demo.exe](#), [earth.exe](#), [mandelbrot.exe](#)



ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ

http://www.opengl.org/wiki/Getting_started

1. **Visual C++** : Windows-to-OpenGL interface (WGL)
 - a. Library : **GLU** Ενσωματωμένη
 - b. Headers: #include <window.h>, <GL/gl.h>, <GL/glu.h>
2. **Visual C++ MFC** : Windows-to-OpenGL interface (WGL)
 - a. Library : GLU Ενσωματωμένη
 - b. Headers: #include <afxwin.h>, <GL/gl.h>, <GL/glu.h>, <GL/gluaux.h>
3. **Console C++ API** : OpenGL Utility Toolkit (**GLUT**)
 - a. Library : GLUT (glut32.dll, glut32.lib)
 - b. Headers: #include <GL/glut32.h>

Εγκατάσταση Βιβλιοθήκης GLUT

1. Αντιγραφή του glut32.dll στο φάκελο “..\System” (ή System32) των Windows.
2. Αντιγραφή του glut32.lib στο φάκελο “..\Lib” της Visual C++.
3. Αντιγραφή του glut32.h στο φάκελο “..\include\GL” της Visual C++.

Βιβλιογραφία

(**EA**) E. Angel, Interactive Computer Graphics, 4th edition 2006, Addison-Wesley
(**HB**) D.Hearn and H. Baker, Computer Graphics with OpenGL, Prentice Hall, 2004

Βασικές Βοηθητικές Ιστοσελίδες

http://poseidon.csd.auth.gr/LAB_COURSES/graphics/index.htm
<http://users.auth.gr/voyatzis/cg>

Ηλεκτρονικές πηγές (Internet)

<http://www.opengl.org/>
<http://www.sgi.com/products/software/opengl/>
<http://nehe.gamedev.net/>
<http://www.codeproject.com/opengl/>
<http://www.deitel.com/ResourceCenters/Programming/OpenGL/OpenGLTrainingCourses/tabid/922/Default.aspx>

κλπ



ΣΥΣΤΗΜΑΤΑ ΣΥΝΤΕΤΑΓΜΕΝΩΝ ΣΧΕΔΙΑΣΗΣ

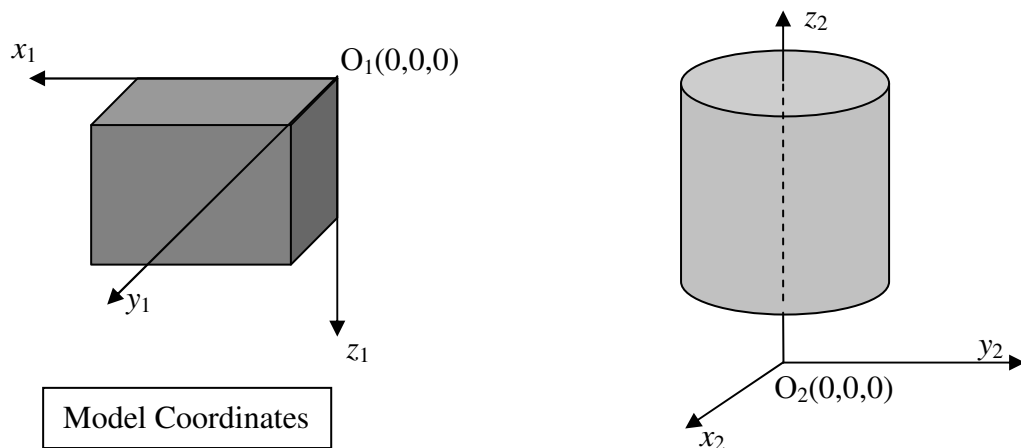
Γραφικό Αντικείμενο

- Γεωμετρική περιγραφή σχήματος (κορυφές)
- Γεωμετρική περιγραφή θέσης (αποστάσεις – διευθύνσεις)

Στην OpenGL οι συντεταγμένες των vertices των αντικειμένων δίνονται πάντα σε ένα **καρτεσιανό σύστημα συντεταγμένων Oxyz**.

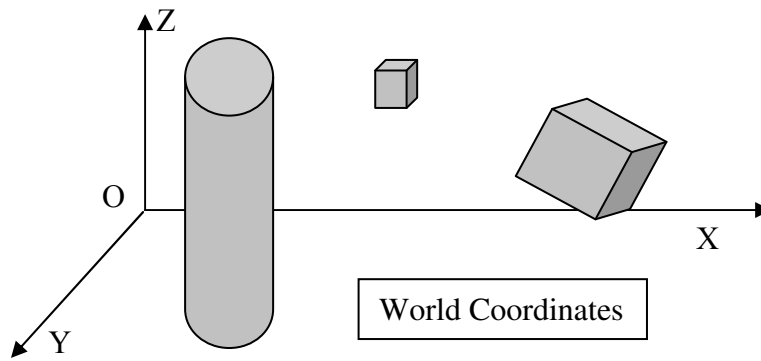
Object – Model coordinates

Η πλήρης εφαρμογή αποτελείται εν' γένει από ένα σύνολο αντικειμένων A_1, A_2, \dots, A_i . Η σχεδίαση ενός αντικειμένου γίνεται σε ένα ανεξάρτητο σύστημα συντεταγμένων $(Oxyz)_i$.



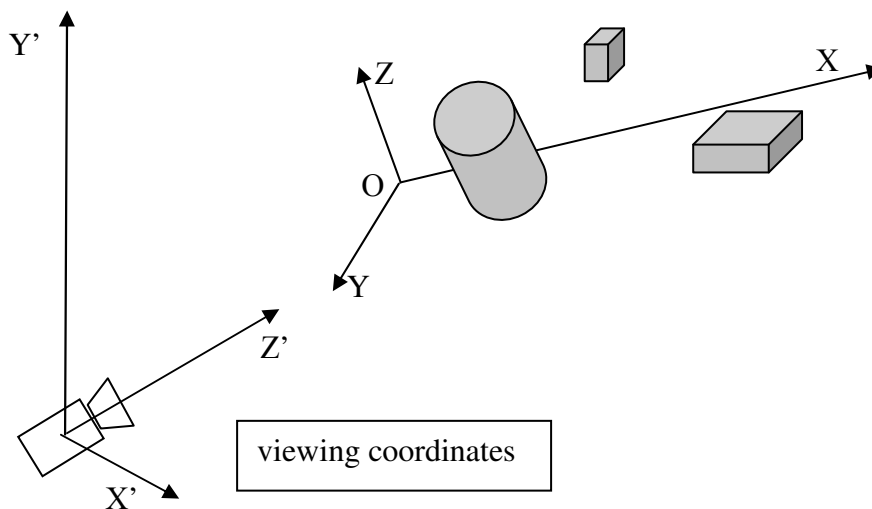
World Coordinates

Ένα συγκεκριμένο καρτεσιανό σύστημα συντεταγμένων $OXYZ$ στο οποίο προσαρμόζονται όλα τα αντικείμενα σχεδίασης της εφαρμογής μετά από κατάλληλους μετασχηματισμούς **μετατόπισης, κλίμακας και περιστροφής**.



Viewing (or eye-camera) Coordinates

Το σύστημα $OXYZ$ αποτελεί ένα σταθερό (fixed) σύστημα για την εφαρμογή και το σχεδιασμό του σκηνικού. Η γωνία και η θέση παρατήρησης ορίζουν ένα νέο σύστημα συντεταγμένων $O'X'Y'Z'$ που ονομάζεται σύστημα παρατήρησης (viewing system)



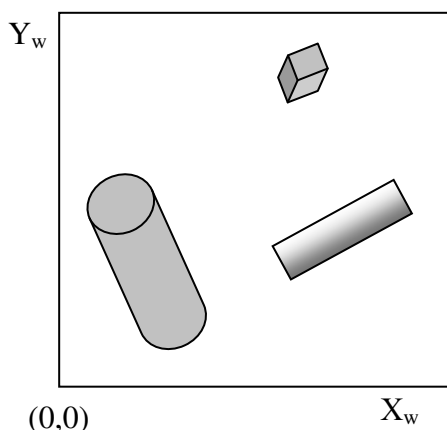
Normalized Coordinates

Οι συντεταγμένες παρατήρησης X' , Y' και Z' προσαρμόζονται σε κατάλληλες κανονικοποιημένες μονάδες ($\langle X' \rangle$, $\langle Y' \rangle$, $\langle Z' \rangle$) έτσι ώστε το σκηνικό να είναι έτοιμο να προβληθεί στην οθόνη.



Window Coordinates

Οι συντεταγμένες ($\langle X' \rangle, \langle Y' \rangle, \langle Z' \rangle$) προβάλλονται στο παράθυρο γραφικών της οθόνης στα σημεία (X_w, Y_w) τα οποία μετρούνται σε *pixels*. Αν η μέθοδος «z-buffer» είναι ενεργή τότε το σύστημα κρατάει και πληροφορία βάθους.



ΤΥΠΟΙ ΣΥΝΑΡΤΗΣΕΩΝ ΓΡΑΦΙΚΩΝ

Attribute functions :

Συναρτήσεις που προσαρτούν/ανακτούν ιδιότητες στο/από αντικείμενο (χρώμα, μοτίβο κλπ)

Geometric Transformations:

Τοποθέτηση-προσανατολισμός αντικειμένων στο χώρο.

Evaluators

Συναρτήσεις που παράγουν αυτόματα σημεία-διανύσματα για συγκεκριμένη χρήση

Viewing Transformations :

Διαχείριση παρουσίασης του σκηνικού (απόκρυψη αντικειμένων, φωτισμός, θέση παρατήρησης κλπ)

I/O Interaction :

Συναρτήσεις αλληλεπίδρασης του προγράμματος με τις συσκευές

Control Functions :

Συναρτήσεις ελέγχου (άνοιγμα - κλείσιμο) διαδικασιών και αλγορίθμων

Callback Functions

Συναρτήσεις που εκτελούνται συνεχώς μέσα στο κύκλο λειτουργίας του κώδικα.

Display-Window Managment :

Διαχείριση παραθύρων εφαρμογής – Εξάρτηση από το υπολογιστικό σύστημα/περιβάλλον



Βασικός Κώδικας*

```
//Project >> BasicCode/code11.cpp

#include <stdio.h>
#include <GL/glut.h>

void initmywindow(GLfloat b)
{
    glClearColor(b,b,b,0);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(0.0,100,0.0,100);
}

void mytest()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0,0.0,0.0);
    glBegin(GL_LINES);
        glVertex2i(10.0,10.0);
        glVertex2i(30.0,80.0);
    glEnd();
    glFlush();
}

void main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutCreateWindow("My graphic window");

    initmywindow(0.5);
    glutDisplayFunc(mytest);
    glutMainLoop();
}
```

* HB, σελ. 76-80



Βασική Σύνταξη-Ονοματολογία

1. Η OpenGL χρησιμοποιεί δικούς της τύπους ονοματολογίας δεδομένων (built-in data type names) που είναι συμβατοί σε όλα τα συστήματα που τη χρησιμοποιούν και το μέγεθος (word size) των μεταβλητών αυτών είναι ανεξάρτητο του συστήματος. Η ονομασία ξεκινάει με το πρόθεμα **GL** και ακολουθεί η ονομασία του τύπου μεταβλητής κατά τα πρότυπα της C. Πχ

GLint, GLfloat, GLdouble, κλπ

2. Όλες οι εντολές-συναρτήσεις γραφικών της βιβλιοθήκης της OpenGL ξεκινούν με το πρόθεμα **gl** και στη συνέχεια έχουν ένα όνομα που αρχίζει με κεφαλαίο γράμμα

glEnd(), glFlush(), glBegin(...), glClear(...)

3. Πολλές συναρτήσεις γραφικών δέχονται ως όρισμα κάποια σταθερά (συνήθως ακέραια) που δηλώνει τον τρόπο λειτουργίας της συνάρτησης. Οι δεδηλωμένες σταθερές (definitions) της βιβλιοθήκης της OpenGL ξεκινούν με το πρόθεμα **GL_** και στη συνέχεια ακολουθεί λέξη με κεφαλαία. Πχ

GL_PROJECTION, GL_LINES, κλπ

4. Κάποιες εντολές γραφικών τελειώνουν με έναν αριθμό n και ένα σύμβολο από τα "i", ή "f". Ο αριθμός δηλώνει τον αριθμό ορισμάτων της συνάρτησης και το σύμβολο τον τύπο δεδομένων (integer και float αντίστοιχα). Πχ

glColor3f(1.0,0.0,0.0) : τρεις πραγματικοί αριθμοί ως ορίσματα

glVertex2i(30,80) : δύο ακέραιοι ως ορίσματα

(* ή glVertex2i(30.1,80.1) , το δεκαδικό στοιχείο δεν λαμβάνεται υπόψη)

Αν το σύμβολο τα "i", ή "f" το ακολουθεί και το σύμβολο "v" τότε το όρισμα είναι μια **δομημένη μεταβλητή** (struct). Πχ

```
struct Point2D { GLfloat x; GLfloat y; }
```

```
Point2D p={1.2, 3.4};
```

```
glVertex2f(p.x, p.y)
```

or

```
glVertex2fv( p )
```



Βασικές εντολές της GLUT

Οι εντολές της GLUT αρχίζουν με το πρόθεμα **glut** και χρησιμοποιούνται για την δημιουργία και διαχείριση παραθύρων γραφικών μέσα από ένα **προγραμματιστικό περιβάλλον κονσόλας**.

glutInit(&argc, argv) :

Αρχικοποιεί τον τρόπο αλληλεπίδρασης μεταξύ του περιβάλλοντος της εφαρμογής με την βιβλιοθήκη της OpenGL. Η συνάρτηση δέχεται υποχρεωτικά τα βασικά ορίσματα του προγράμματος με τα οποία μπορεί ο χρήστης να ορίσει εσωτερικές διαδικασίες και ελέγχους.

glutInitDisplayMode(mode)

Καθορίζει το τρόπο παρουσίασης των γραφικών (χρώμα, buffering, απόκρυψη αντικειμένων κλπ). Οι διάφορες παράμετροι *mode* χωρίζονται μεταξύ τους με τον τελεστή or (|).

πχ `glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB)` = μονό buffer , χρώμα RGB.

glutInitWindowPosition(pos_x, pos_y):

Αρχική θέση (σε pixels) του παραθύρου γραφικών στην οθόνη. (default: system's control)

glutInitWindowSize(size_x, size_y):

Μέγεθος παραθύρου σε pixels (default=300x300)

glutCreateWindow("title"):

Δημιουργεί και παρουσιάζει το παράθυρο (virtually) λαμβάνοντας υπόψη τα παραπάνω στοιχεία θέσης και μεγέθους. Η παρουσίαση του παραθύρου στην οθόνη γίνεται με την ολοκλήρωση του απαραίτητου κύκλου εντολών.

glutDisplayFunc(user's_function)

Ορίζει (registers) τη συνάρτηση του χρήστη που εκτελεί τη σχεδίαση των γραφικών (*display call back function*). Η συνάρτηση αυτή εκτελείται κάθε φορά που το σύστημα θεωρεί ότι το παράθυρο πρέπει να ανανεωθεί (refresh).

glutMainLoop()

Εκτελεί τις εντολές γραφικών : παρουσιάζει το παράθυρο γραφικών στη οθόνη και εκτελεί τον κώδικα σχεδίασης (*display call back functions*). Ένα πρόγραμμα της OpenGL αποτελείται από ένα σύνολο και μια οργάνωση από *display call back functions*. Επίσης με την `glutMainLoop` το πρόγραμμα εισέρχεται σε κατάσταση *βρόγχου επεξεργασίας* (glut processing loop) με συνεχή έλεγχο των «events» (ενέργειες αλληλεπίδρασης με τις μονάδες εισόδου).