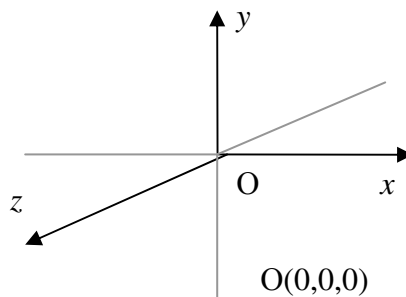
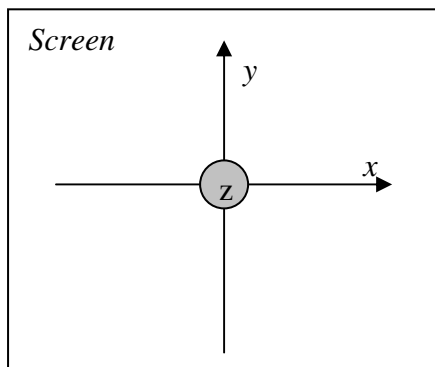


## ΣΧΕΔΙΑΣΗ ΣΕ ΤΡΕΙΣ ΔΙΑΣΤΑΣΕΙΣ (3D)

-ΠΡΟΒΟΛΕΣ και ΠΑΡΑΤΗΡΗΣΗ -

Η OpenGL σχεδιάζει *άμεσα* σε τρεις διαστάσεις  $Oxyz$ . Η σχεδίαση σε δύο διαστάσεις  $Oxy$  γίνεται με την αναγωγή  $z=0$ .

### 1. Το σταθερό 3D σύστημα αναφοράς και Μετασχηματισμοί



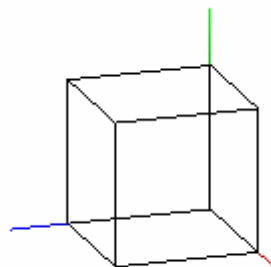
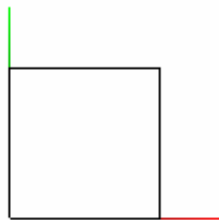
Οι κορυφές των αντικειμένων ορίζονται με τρεις συντεταγμένες

`glVertex3f( x, y, z )` ή `glVertex3fv( p )`,  $p=\{x,y,z\}$

Αν δοθούν μόνο τα  $(x,y)$  τότε θεωρείται πως  $z=0$ .

Τα πρωταρχικά αντικείμενα (points, lines, strips, polygons) ορίζονται αντίστοιχα. Ιδιαίτερη προσοχή πρέπει να δοθεί στη σωστή φορά διαγραφής για τον ορισμό πρόσοψης

(βλ. `code50.cpp`).



## 2. Μετασχηματισμοί :

Ο χώρος του συστήματος συντεταγμένων καθώς και τα αντικείμενα σχεδίασης, δηλαδή οι κορυφές τους (*vertices*), μπορούν να μετασχηματιστούν εφαρμόζοντας μια σειρά μετασχηματισμών που αντιστοιχούν σε *πίνακες μετασχηματισμών* (*matrices*).

Η εφαρμογή των μετασχηματισμών «προσδιορίζει» τον τρέχοντα πίνακα μετασχηματισμού (*Current Transformation Matrix – CTM*) που προσδιορίζεται με την εντολή

```
glMatrixMode(-----)
```

Υπάρχουν τρεις βασικοί μετασχηματισμοί

**ModelView** : μετασχηματισμοί αντικειμένων και παρατήρησης

```
glMatrixMode(GL_MODELVIEW);
```

**Projection** : Μετασχηματισμοί που απεικονίζουν το σκηνικό σε έναν σταθερό 3D χώρο (*3D Projection Space*).

```
glMatrixMode(GL_PROJECTION);
```

**Viewport** : Προβολή του 3D Projection Space στο επίπεδο της οθόνης.

```
glViewport(-----);
```

\* Ο μετασχηματισμός viewport δεν ανήκει στην ομάδα των matrix operations. Εφαρμόζεται ανεξάρτητα και πάντα πριν την απεικόνιση στην οθόνη.

\* Υπάρχουν και καταστάσεις πινάκων για την εφαρμογή σε “textures” και “color masks”

### 3. Προβολές

Το σκηνικό προβάλλεται στο 3D Projection Space που αποτελεί και χώρο αποκοπής. Δηλαδή το Viewport προβάλλει στην οθόνη μόνο το σκηνικό το οποίο βρίσκεται μέσα στο 3D Projection Space.

\* Ο ορισμός του τύπου προβολής γίνεται με κατάλληλο μετασχηματισμό της κατάστασης προβολής

```
glMatrixMode (GL_PROJECTION)1
```

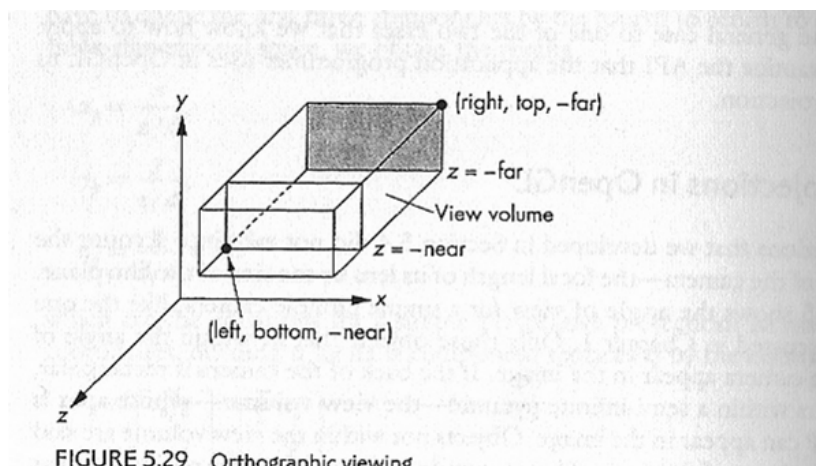
Στην OpenGL ορίζονται δύο τύποι προβολών

#### α) Ορθογραφική ή παράλληλη προβολή :

Το 3D projection space ορίζεται ως ένα ορθογώνιο παραλληλεπίπεδο.

Ο παρατηρητής θεωρείται ότι βρίσκεται στη θέση (0,0,+inf) και βλέπει προς την αντίθετη κατεύθυνση του άξονα z (προς τα αρνητικά).

Δεν δίνεται η αίσθηση του βάθους – απόστασης προς τον άξονα z .



Ο χώρος θέασης (3D projection space) ορίζεται με την συνάρτηση

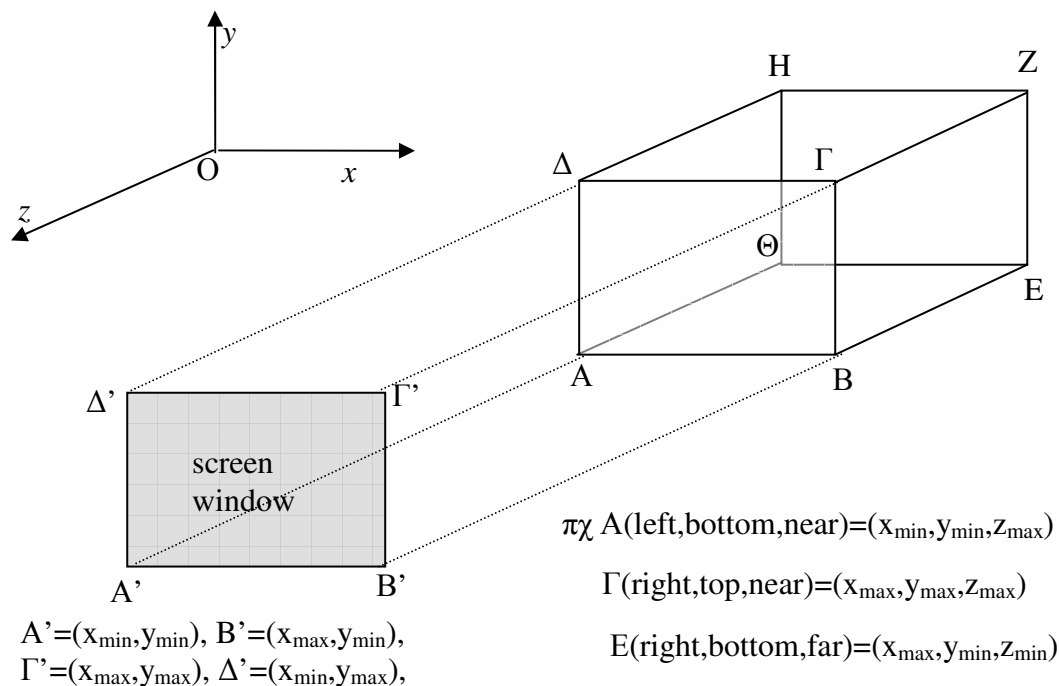
```
glOrtho(left, right, bottom, top, near, far);
```

<sup>1</sup> Προεπιλεγόμενη κατάσταση μετασχηματισμού.

ή σε ένα  $Oxyz$  σύστημα συντεταγμένων

$glOrtho(x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$

- Τα ορίσματα της  $glOrtho()$  είναι τύπου  $GLdouble$
- Προεπιλογή  $glOrtho(-1, 1, -1, 1, -1, 1)$
- Η  $glOrtho2D(x_{min}, x_{max}, y_{min}, y_{max})$ , που χρησιμοποιείται για τα διδιάστατα γραφικά, είναι ισοδύναμη με την  $glOrtho(x_{min}, x_{max}, y_{min}, y_{max}, -1, 1)$ .



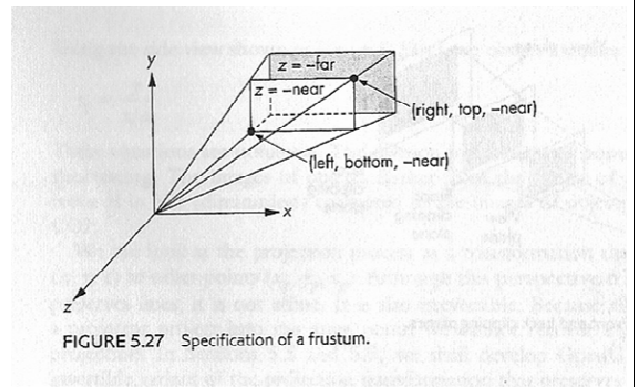
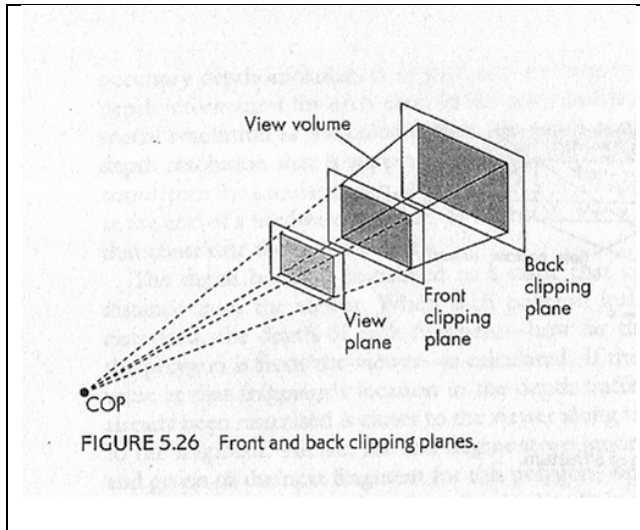
Το επίπεδο προβολής θεωρείται το  $z=z_{max} = -near$

### β) Προβολή με προοπτική (Perspective Projection) :

Το 3D projection space ορίζεται ως ένα frustum (κόλουρη πυραμίδα)

Ο παρατηρητής βρίσκεται στο  $(0,0,0)$  και παρατηρεί προς την αντίθετη κατεύθυνση του άξονα  $z$  (προς τα αρνητικά).

Τα αντικείμενα που είναι κοντά στον παρατηρητή προβάλλονται μεγαλύτερα από αυτά που βρίσκονται μακριά.



Επειδή η διεύθυνση παρατήρησης είναι προς την αρνητική φορά του άξονα  $z$ , τα παραλληλόγραμμα που ορίζουν το χώρο θέασης πρέπει να ορίζονται σε επίπεδα για αρνητικά  $z$ .

α) Ο χώρος θέασης ορίζεται με την συνάρτηση

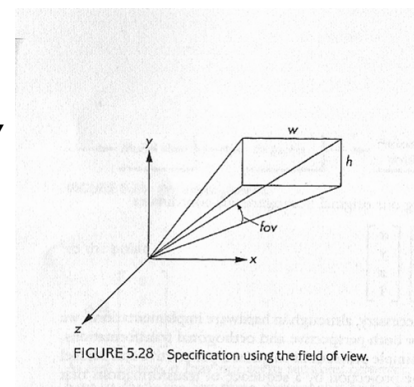
```
glFrustum(left, right, bottom, top, near, far);
```

Τα ορίσματα δηλώνονται όπως και στην `glOrtho` αλλά οι τιμές `near` και `far` είναι **πάντα θετικές** (δηλαδή ο χώρος θέασης βρίσκεται σε  $z < 0$  αφού  $z_{\max} = -near$  και  $z_{\min} = -far$ )

β) Ο χώρος θέασης ορίζεται με την συνάρτηση

```
gluPerspective(fovy, aspect, near, far);
```

όπου η γωνία `fovy` αναφέρεται στη γωνία θέασης του πάνω και κάτω επιπέδου του χώρου θέασης και το `aspect` το λόγο κλίμακας μεταξύ της οριζόντιας και της κάθετης διεύθυνσης.



\*βλ. [Κώδικας code51.cpp](#)

## 4. Διαμόρφωση Παρατήρησης (τοποθέτηση κάμερας)

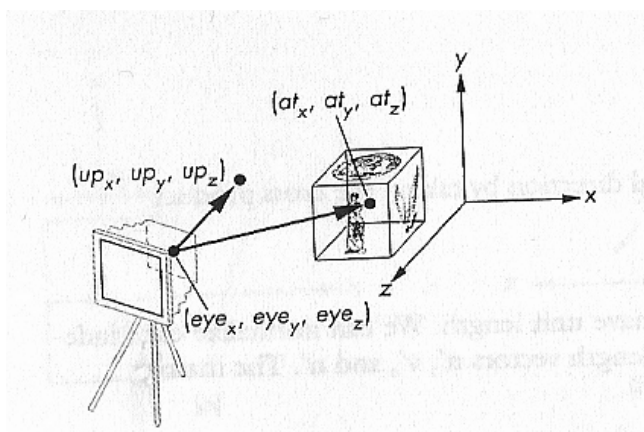
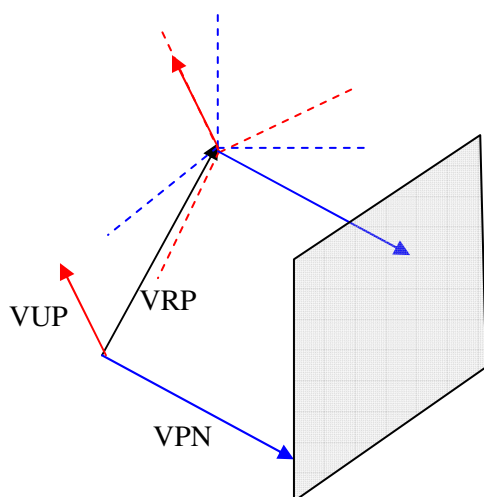
Ο προσδιορισμός της παρατήρησης περιλαμβάνει α) το σημείο τοποθέτησης της κάμερας και β) τον προσανατολισμό της

α) Σημείο τοποθέτησης της κάμερας (view reference point, **VRP**) : Ένα σημείο  $(X, Y, Z)$  στο χώρο

β) Προσανατολισμός κάμερας

β1) Διεύθυνση επιπέδου φακού (view-plane normal, **VPN**) : Καθορίζεται από το κάθετο διάνυσμά του  $(N_x, N_y, N_z)$ .

β2) Στροφή της κάμερας (view-up vector, **VUP**): Καθορίζεται από διάνυσμα  $(U_x, U_y, U_z)$  που δηλώνει την διεύθυνση του πάνω μέρους της κάμερας σε σχέση με τον χώρο θέασης.



Η τοποθέτηση της κάμερας στην OpenGL γίνεται με την συνάρτηση

```
gluLookAt (eyeX, eyeY, eyeZ, atX, atY, atZ, upX, upY, upZ)
```

όπου

$(eyeX, eyeY, eyeZ) = VRP$

$(atX, atY, atZ)$  : σημείο εστίασης (για καθορισμό διεύθυνσης)

άρα  $(eyeX - atX, eyeY - atY, eyeZ - atZ) = VPN$

$(upX, upY, upZ) = VUP$

- Η `gluLookAt` είναι μετασχηματισμός τύπου **MODELVIEW**

\*βλ. Κώδικας `code52.cpp`

## 5. Η Μέθοδος «αφαίρεσης κρυμμένων επιφανειών» (Hidden-Surface Removal)

Η OpenGL χρησιμοποιεί άμεσα τον αλγόριθμο *z-buffer* για την απόκρυψη αντικειμένων ή τμημάτων αντικειμένων, που βρίσκονται πίσω από άλλα αντικείμενα, με βάση το σύστημα συντεταγμένων ή την θέση-γωνία παρατήρησης.

Η κατάσταση λειτουργίας της δυνατότητας απόκρυψης καθώς και η ενεργοποίησή της γίνεται με τη χρήση των παρακάτω

- `glutInitDisplayMode( ..... | GLUT_DEPTH)`  
Ανοίγει *z-buffer* (μνήμη) για την αποθήκευση πληροφορίας του βάθους του αντικειμένου.
- `glEnable(GL_DEPTH_TEST)`  
Ενεργοποιεί τον αλγόριθμο αφαίρεσης κρυμμένων επιφανειών (*z-buffer*).
- `glClear( ..... | GL_DEPTH_BUFFER_BIT)`  
Καθαρίζει το *z-buffer*

Κώδικας `code53pro.cpp` : χωρίς τη χρήση του αλγορίθμου

Κώδικας `code53.cpp` : με τη χρήση του αλγορίθμου

## Ασκήσεις

4-1. Να κατασκευαστεί μια συνάρτηση `Pyramid(int N)` που να σχεδιαστεί μια πυραμίδα με βάση ένα κανονικό πολύγωνο  $N$  πλευρών, ακτίνας 1 στο επίπεδο  $Oxy$  και με κορυφή στον άξονα  $Oz$  στο σημείο  $(0,0,1)$ .

4.2. Να σχεδιαστεί μια λωρίδα Mobius.

\*4-3. Να πραγματοποιηθεί ένας μηχανισμός κίνησης (με πλήκτρα) μιας κάμερας πάνω σε σφαίρα που να εστιάζει προς το κέντρο έχοντας σταθερό διάνυσμα  $VUP=(0,1,0)$ . Συγκεκριμένα, θεωρώντας τις σφαιρικές συντεταγμένες  $(\rho, \varphi, \theta)$  για τη θέση της κάμερας, να οριστούν πλήκτρα που να αυξομειώνουν τις παραπάνω τρεις συντεταγμένες.