

Κεφάλαιο 3ο

Μετασχηματισμοί συντεταγμένων

Εισαγωγή

Στο 1^ο Κεφάλαιο αναλύσαμε βασικές εντολές σχεδίασης, μέσω των οποίων ο προγραμματιστής μπορεί να καθορίσει τις συντεταγμένες της σκηνής στις οποίες επιθυμεί να σχεδιάσει γεωμετρικά σχήματα. Ωστόσο υπάρχουν περιπτώσεις στις οποίες είναι επιθυμητή η επιβολή ενός μετασχηματισμού συντεταγμένων ή μιας αλυσίδας μετασχηματισμών συντεταγμένων, πριν τον προσδιορισμό της θέσης στην οποία θα σχεδιαστεί ένα σημείο. Ουσιαστικά επιθυμούμε, με τον ορισμό ενός σημείου με συντεταγμένες (x, y, z) και βάσει ενός κανόνα αντιστοίχισης, το σημείο να αποδοθεί σε μια θέση της σκηνής με συντεταγμένες (x', y', z') .

Στο κεφάλαιο αυτό αναλύεται η λογική που ακολουθεί η μηχανή της OpenGL, σε ό,τι αφορά την εφαρμογή μετασχηματισμών συντεταγμένων. Παρουσιάζονται τα μητρώα στοιχειωδών μετασχηματισμών και οι αντίστοιχες εντολές ορισμού τους. Επιπλέον αναλύεται η διαδικασία καθορισμού σύνθετων μετασχηματισμών, οι οποίοι παράγονται από τη συνδυασμένη εφαρμογή στοιχειωδών γραμμικών μετασχηματισμών.

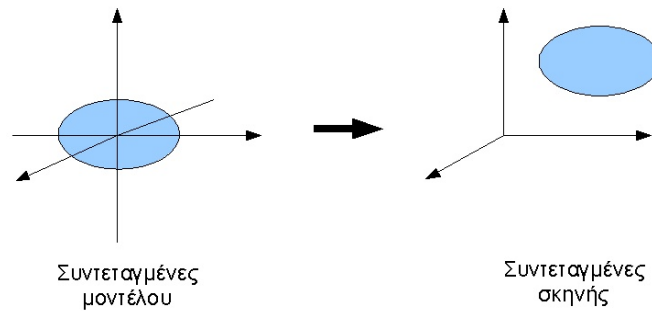
3.1 Συντεταγμένες μοντέλου – Μετασχηματισμός μοντέλου

Εκ πρώτης όψεως, η διαδικασία αντιστοίχισης ενός σημείου σε μια θέση διαφορετική από την αρχικά δηλωμένη, πιθανόν να φαίνεται μια άσκοπη διαδικασία στον προγραμματιστή. Ωστόσο, υπάρχουν περιπτώσεις στις οποίες ένας τέτοιος κανόνας αντιστοίχισης αποδεικνύεται αναγκαίος.

α) Η χρησιμότητα των μετασχηματισμών συντεταγμένων αναδεικνύεται στην περίπτωση που χρησιμοποιούμε ένα γεωμετρικό σχήμα που έχει οριστεί από τρίτο προγραμματιστή σε μια display list. Στην περίπτωση που ο τρίτος προγραμματιστής κατασκευάζει μια λίστα απεικόνισης, ορίζει ένα σύνθετο σχήμα σε ένα βολικό για αυτόν σύστημα συντεταγμένων. Συνήθως ορίζει το σχήμα κοντά στην αρχή των αξόνων. Το εύρος των συντεταγμένων στο οποίο εκτείνεται το σχήμα επιλέγεται αυθαίρετα ή βάσει της κοινής λογικής. Οι συντεταγμένες με τις οποίες δηλώνεται ένα προτύπο σχήμα σε μια λίστα απεικόνισης συχνά αναφέρονται ως **συντεταγμένες μοντέλου**.

Ωστόσο, είναι προφανές ότι, στην πλειοψηφία των περιπτώσεων, οι συντεταγμένες μοντέλου δεν εκφράζουν την επιθυμητή θέση ή/και τις επιθυμητές διαστάσεις με τις οποίες επιθυμεί να αποδώσει ένας χρήστης το σχήμα στη σκηνή του. Επιπλέον, εάν ο χρήστης της λίστας επιθυμεί να επαναχρησιμοποιήσει τον κώδικά της για τη σχεδίαση πολλαπλών σχημάτων στην ίδια σκηνή, θα ήταν αδύνατο να το επιτύχει με

απλή εκτέλεση της λίστας απεικόνισης. Κάθε εκτέλεση της λίστας απεικόνισης θα σχεδιάζε το σχήμα στην ίδια (αρχικά καθορισμένη) θέση και με τις ίδιες διαστάσεις που έχουν οριστεί στη λίστα απεικόνισης. Είναι λοιπόν εμφανές ότι σε ορισμένες περιπτώσεις επιθυμούμε να επιβάλλουμε μετασχηματισμούς που μετασχηματίζουν τις συντεταγμένες του πρότυπου σχήματος σε κατάλληλες θέσεις και με κατάλληλες διαστάσεις στη σκηνή όπως φαίνεται στο σχ. 3.1.



Σχ. 3.1: Παράδειγμα μετασχηματισμού μοντέλου

Ο μετασχηματισμός αυτός ονομάζεται **μετασχηματισμός μοντέλου (modelview transformation)** και εκτελείται ορίζοντας ένα μητρώο μετασχηματισμού το οποίο καθορίζει τις συντεταγμένες σκηνής (x', y', z') που θα αντιστοιχιστούν σε ένα σημείο που έχει δηλωθεί με συντεταγμένες μοντέλου (x, y, z) .

3.2 Μητρώο μετασχηματισμού

Η μηχανή καταστάσεων της OpenGL προβλέπει τη χρήση δύο διαφορετικών μητρώων, τα οποία, συνδυαζόμενα, παράγουν την τελική απεικόνιση της σκηνής στην οθόνη του χρήστη. Τα μητρώα αυτά είναι το **μητρώο μετασχηματισμού μοντέλου (modelview matrix)** και το **μητρώο προβολής (projection matrix)**.

α) Το μητρώο μετασχηματισμού μοντέλου σχηματίζεται βάσει των μαθηματικών σχέσεων που καθορίζουν το μετασχηματισμό μοντέλου. Δηλαδή τα στοιχεία του μητρώου έχουν τιμές τέτοιες, ούτως ώστε η δήλωση ενός σημείου με συντεταγμένες (x, y, z) να οδηγεί στην απόδοση του σημείου (x', y', z') στη σχεδιαζόμενη σκηνή βάσει της σχέσης.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = M_{\text{modelview}} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

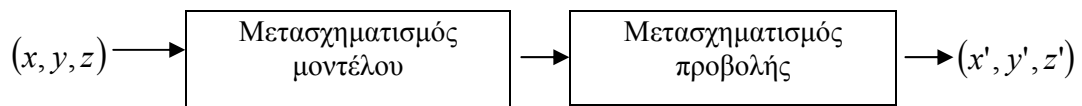
Επιπλέον, στο μητρώο μετασχηματισμού μοντέλου εμπεριέχονται οι μετασχηματισμοί που εφαρμόζονται προκειμένου η σκηνή να αποδοθεί από διαφορετικές οπτικές γωνίες. Στην περίπτωση αυτή, η

περιγραφή της σκηνής ανάγεται σε σύστημα συντεταγμένων που η θέση του καθορίζεται από την οπτική γωνία του θεατή και οι μετασχηματισμοί που εμπλέκονται στη διαδικασία αυτή ενσωματώνονται στο μητρώο μετασχηματισμού μοντέλου. Ο μετασχηματισμός οπτικής γωνίας θα αναλυθεί στην ενότητα “Μετασχηματισμός οπτικής γωνίας” της ενότητας “Προβολές”.

β) Το μητρώο προβολής σχηματίζεται από κανόνες αντιστοίχισης που καθορίζουν τον τρόπο με τον οποίο η παρατηρούμενη σκηνή θα προβληθεί στο επίπεδο παρατήρησης του θεατή (επίπεδο προβολής) και η επεξεργασία του θα αναλυθεί στο επόμενο κεφάλαιο.

Τα μητρώα μετασχηματισμού μοντέλου και προβολής έχουν διαστάσεις 4×4 .

Η συνδυασμένη επίδραση των μητρώων μετασχηματισμού μοντέλου και προβολής στις συντεταγμένες των οριζόμενων σημείων, σχηματίζει την εικόνα που τελικά αποδίδεται στην οθόνη του υπολογιστή. Η διαδικασία φαίνεται στο Σχ. 3.2.



Σχ. 3.2 Διάγραμμα μετασχηματισμού συντεταγμένων στη μηχανή της OpenGL

Επομένως, στις συντεταγμένες (x, y, z) πρώτα επιδρά το μητρώο μετασχηματισμού μοντέλου $M_{modelview}$ και κατόπιν το μητρώο προβολής $M_{projection}$.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = M_{projection} \cdot M_{modelview} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Στην OpenGL, σε κάθε χρονική στιγμή, είναι δυνατή η πρόσβαση μόνο σε ένα από τα δύο μητρώα. Ο χρήστης καθορίζει ποιο από τα μητρώα επιθυμεί να τροποποιήσει την εκάστοτε χρονική στιγμή με την εντολή **glMatrixMode**:

glMatrixMode(matrixMode);

όπου *matrixMode* σταθερά που παίρνει μία από τις τιμές:

GL_MODELVIEW: μετάβαση στην κατάσταση επεξεργασίας του μητρώου μετασχηματισμού μοντέλου

GL_PROJECTION: μετάβαση στην κατάσταση επεξεργασίας του μητρώου προβολής.

Τα μητρώα μετασχηματισμού μοντέλου και προβολής έχουν ως αρχική τιμή τον πίνακα I_4 :

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

το οποίο ουσιαστικά υποδηλώνει ότι οι συντεταγμένες που δηλώνονται δεν υφίστανται κανένα μετασχηματισμό:

$$\begin{aligned} x' &= x \\ y' &= y \\ z' &= z \end{aligned}$$

Πρέπει να επισημάνουμε ότι τα μητρώα μετασχηματισμών, ως μεταβλητές κατάστασης, διατηρούν τις τιμές που τους έχουν ανατεθεί την τελευταία φορά. Συνεπώς, εάν τροποποιηθούν και στο ενδιάμεσο της εκτέλεσης του προγράμματος απαιτηθεί αρχικοποίησή τους, αυτή θα πρέπει να δηλωθεί ρητά από τον προγραμματιστή.

Η αρχικοποίηση των μητρώων προβολής και μετασχηματισμού μοντέλου στο μητρώο I_4 γίνεται με την εντολή *glLoadIdentity* :

void glLoadIdentity ();

η οποία αρχικοποιεί το μητρώο μετασχηματισμού που επεξεργαζόμαστε στην εκάστοτε χρονική στιγμή. Δεδομένου ότι η μηχανή καταστάσεων της OpenGL διατηρεί τις τελευταίες ρυθμίσεις που δόθηκαν, ο προγραμματιστής, πριν δώσει εντολή αρχικοποίησης μητρώου, θα πρέπει να ελέγξει τη ροή εκτέλεσης του προγράμματός του και να βεβαιωθεί ότι η εντολή θα αρχικοποιήσει το επιθυμητό μητρώο μετασχηματισμού.

```
glMatrixMode (GL_MODELVIEW) ;  
.....  
.....  
.....  
glLoadIdentity() ; // Αρχικοποίηση του μητρώου μετασχηματισμού μοντέλου
```

Αρχικοποιεί το μητρώο μετασχηματισμού μοντέλου

```
glMatrixMode (GL_PROJECTION) ;  
.....  
.....  
.....  
glLoadIdentity() ; //Αρχικοποίηση του μητρώου προβολής
```

3.3 Στοιχειώδεις μετασχηματισμοί

Στην ενότητα αυτή αναλύουμε τα μητρώα στοιχειωδών μετασχηματισμού και τις εντολές δήλωσης των στοιχειωδών μετασχηματισμών μετατόπισης, κλιμάκωσης, κλίσης και περιστροφής

3.3.1 Μετατόπιση

Μετατόπιση σημείου στο χώρο κατά (x_{tr}, y_{tr}, z_{tr}) περιγράφεται από τις εξισώσεις

$$\begin{aligned}x' &= x + x_{tr} \\y' &= y + y_{tr} \\z' &= z + z_{tr}\end{aligned}$$

Το αντίστοιχο μητρώο μετατόπισης 4×4 έχει τη μορφή:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_{tr} \\ 0 & 1 & 0 & y_{tr} \\ 0 & 0 & 1 & z_{tr} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Στην OpenGL η μετατόπιση των συντεταγμένων της σκηνής στο χώρο κατά σταθερές (x_{tr}, y_{tr}, z_{tr}) εκτελείται με την εντολή **glTranslate***:

glTranslatef (GLfloat xtr, GLfloat ytr, GLfloat ztr);

glTranslated (GLdouble xtr, GLdouble ytr, GLdouble ztr);

3.3.2 Κλιμάκωση

Στην κλιμάκωση, οι συντεταγμένες πολλαπλασιάζονται με ένα σταθερό ανά διεύθυνση συντελεστή βάσει των σχέσεων:

$$\begin{aligned}x' &= s_x \cdot x \\y' &= s_y \cdot y \\z' &= s_z \cdot z\end{aligned}$$

όπου s_x , s_y και s_z οι επιβαλλόμενοι συντελεστές κλιμάκωσης κατά τις διευθύνσεις x , y και z αντίστοιχα.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Στην OpenGL, η κλιμάκωση εκτελείται με την εντολή **glScale***:

glScalef(GLfloat sx, GLfloat sy, GLfloat sz);

glScaled(GLdouble sx, GLdouble sy, GLdouble sz);

όπου s_x , s_y , s_z οι συντελεστές κλιμάκωσης κατά τις διευθύνσεις x , y , z αντίστοιχα.

Τιμές του συντελεστή κλιμάκωσης ίσες με -1 εξάγουν την **ανάκλαση (reflection)** του σχήματος ως προς τη διεύθυνση εφαρμογής του συντελεστή. Π.χ. τα συμμετρικά ενός σχήματος ως προς τα επίπεδα $x = 0$, $y = 0$ και $z = 0$ προκύπτουν με τα αντίστοιχα μητρώα ανάκλασης:

$$S_{REFX} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, S_{REFY} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, S_{REFZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.3.3 Κλίση

Στους μετασχηματισμούς κλίσης, η τιμή μίας από τις συντεταγμένες x, y, z των σημείων μεταβάλλεται γραμμικά ως προς μία ή περισσότερες εκ των άλλων δύο συντεταγμένων.

Π.χ. στο ακόλουθο παράδειγμα η συντεταγμένη x κάθε σημείου μεταβάλλεται γραμμικά ως προς τη συντεταγμένη y :

$$x' = x + a \cdot y$$

Σε σε μορφή μητρώου έχουμε

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Ουσιαστικά στο παραπάνω παράδειγμα η συντεταγμένη x κάθε σημείου μεταβάλλεται γραμμικά ως προς την απόσταση του σημείου από την ευθεία $y = 0$ (που είναι ίση με y).

Στο παρακάτω παράδειγμα έχουμε γραμμική μεταβολή της συντεταγμένης z ενός σημείου συναρτήσει της κάθετης απόστασής του από τις ευθείες $x = 0$ και $y = 0$.

$$z' = a \cdot x + b \cdot y + z$$

και αντίστοιχα σε μορφή μητρώου:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Ο παραπάνω μετασχηματισμός κλίσης εκλαμβάνει ως ευθείες αναφοράς τις $x = 0$ και $y = 0$ (δηλαδή σημείο που βρίσκεται στην τομή των ευθειών αυτών δε μεταβάλλει τη συντεταγμένη z κατά το μετασχηματισμό κλίσης). Όταν θέλουμε να επιβάλλουμε μετασχηματισμό κλίσης ως προς τυχαίες ευθείες αναφοράς $x = x_0$, $y = y_0$, τότε οι εξισώσεις του παραπάνω παραδείγματος μετασχηματισμού κλίσης θα έχουν τη μορφή

$$z' = a(x - x_0) + b \cdot (y - y_0) + z$$

και σε μορφή μητρώου:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & 1 & -(a \cdot x_0 + b \cdot y_0) \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

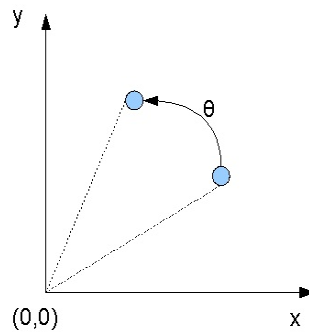
Με παρόμοιο τρόπο μπορούν να οριστούν μετασχηματισμοί κλίσης και για τις συντεταγμένες x' , y' .

Για τον ορισμό μετασχηματισμών κλίσης στην OpenGL δεν υπάρχει συγκεκριμένη εντολή. Ωστόσο, υπάρχει η δυνατότητα ορισμού τους, μέσω της απευθείας ανάθεσης τιμών στα στοιχεία του μητρώου μετασχηματισμού μοντέλου από τον προγραμματιστή, όπως θα δούμε στην ενότητα “Άμεση ανάθεση τιμών σε μητρώα μετασχηματισμού”.

3.3.4 Περιστροφή

Η περιστροφή ενός σημείου κατά γωνία ϕ στις δύο διαστάσεις επί του επιπέδου XY και ως προς άξονα περιστροφής που διέρχεται την αρχή των αξόνων δίνεται από τις σχέσεις:

$$\begin{aligned}x' &= \cos \phi \cdot x - \sin \phi \cdot y \\y' &= \sin \phi \cdot x + \cos \phi \cdot y\end{aligned}$$



Σχ. 3.3 Περιστροφή στις δύο διαστάσεις ως προς την αρχή των αξόνων (άξονας περιστροφής ο Oz)

Για να επεκτείνουμε την περιγραφή του παραπάνω μετασχηματισμού στον τρισδιάστατο χώρο (όπου θεωρούμε ως άξονα περιστροφής τον άξονα Oz) θεωρούμε την πρόσθετη συνθήκη $z' = z$.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \phi & -\sin \phi & 0 & 0 \\ \sin \phi & \cos \phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Από τις παραπάνω σχέσεις μπορούμε να εξαγάγουμε τους τύπους περιστροφής και ως προς τους άξονες Ox και Oy . Ένας μνημονικός κανόνας για το σχηματισμό των αντιστοίχων μαθηματικών τύπων είναι η κυκλική εναλλαγή των μεταβλητών x, y, z στις σχέσεις, σύμφωνα με την αλληλουχία:

$$x \rightarrow y \rightarrow z \rightarrow x$$

Συνεπώς, με την πρώτη αντικατάσταση, προκύπτουν οι σχέσεις.

$$\begin{aligned}x' &= x \\y' &= \cos \phi \cdot y - \sin \phi \cdot z \\z' &= \sin \phi \cdot y + \cos \phi \cdot z\end{aligned}$$

οι οποίες εκφράζουν περιστροφή σημείου στο χώρο κατά γωνία ϕ με άξονα περιστροφής τον άξονα Ox . Σε μορφή τρισδιάστατου μητρώου έχουμε:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi & 0 \\ 0 & \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Με μια επιπλέον κυκλική εναλλαγή, προκύπτουν οι τύποι περιστροφής ως προς τον άξονα Oy .

$$\begin{aligned} x' &= \cos \phi \cdot x + \sin \phi \cdot z \\ y' &= y \\ z' &= -\sin \phi \cdot x + \cos \phi \cdot z \end{aligned}$$

ή εναλλακτικά, σε μορφή μητρώου:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Στην OpenGL μπορούμε να εκτελέσουμε μετασχηματισμούς περιστροφής ως προς οποιαδήποτε άξονα περιστροφής. Αυτό επιτυγχάνεται *ορίζοντας τις συνιστώσες του διανύσματος που ορίζει τη διεύθυνση ενός άξονα περιστροφής. Ο άξονας περιστροφής διέρχεται από την αρχή των αξόνων.* (Η περιστροφή σκηνης ως προς άξονα περιστροφής που διέρχεται από τυχαίο σημείο στο χώρο θα αναλυθεί σε παράδειγμα στο τέλος του κεφαλαίου.)

Μετασχηματισμοί περιστροφής εκτελούνται με την εντολή ***glRotate****:

glRotatef (GLfloat angle, GLfloat vx, GLfloat vy, GLfloat vz);

glRotated (GLdouble angle, GLdouble vx, GLdouble vy, GLdouble vz);

όπου:

angle: η γωνία περιστροφής σε μοίρες

v_x, v_y, v_z : οι συνιστώσες του διανύσματος που εκφράζει τη διεύθυνση του άξονα περιστροφής (Ο άξονας περιστροφής διέρχεται από την αρχή των αξόνων του συστήματος συντεταγμένων.)

Η φορά περιστροφής καθορίζεται από τη φορά του διανύσματος (v_x, v_y, v_z) σύμφωνα με τον κανόνα του δεξιού χεριού.

Π.χ., για να περιστρέψουμε ένα σημείο κατά 30 μοίρες ως προς άξονα περιστροφής που είναι παράλληλος με το διάνυσμα $v = (1,2,1)$ και διέρχεται από την αρχή των αξόνων, η εντολή περιστροφής συντάσσεται ως εξής:

```
glRotate{fd} (30, 1, 2, 1);
```

Παράδειγμα: Μετασχηματισμός μετατόπισης

```
#include <glut.h>

void display()
{
    glLoadIdentity();
    glClearColor(1,1,1,0);
    glClear(GL_COLOR_BUFFER_BIT);

    glLineWidth(3);

    glColor3f(0,0,1);

    glBegin(GL_LINES);

    //Drawing the Oy axis
    glVertex2f(0,-24);
    glVertex2f(0,24);

    //Drawing the Ox axis
    glVertex2f(-32,0);
    glVertex2f(32,0);
    glEnd();

    //Drawing a rectangle
    glRecti(0,0,20,10);

    glColor3f(0,1,0);

    //All declared shapes will be translated with respect to the coordinates
    given.
    glTranslatef(10,5,0);

    //This rectangle will be drawn translated.
    glRecti(0,0,20,10);

    glFlush();
}

int main(int argc, char** argv)
{
    glutInit(&argc,argv);
    glutInitWindowPosition(50,50);
    glutInitWindowSize(640,480);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutCreateWindow("Translation example");

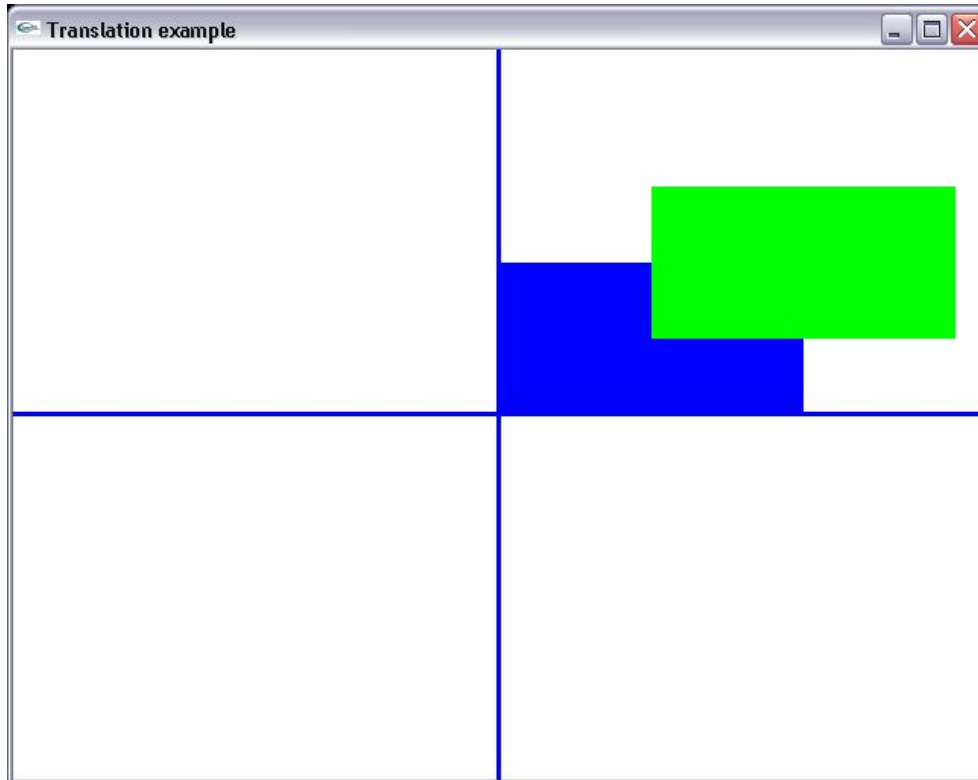
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-32,32,-24,24);
```

```

glMatrixMode(GL_MODELVIEW);
glutDisplayFunc(display);
glutMainLoop();

return 0;
}

```



Παράδειγμα: Μετασχηματισμός περιστροφής

```

#include <glut.h>

void display()
{
    glLoadIdentity();
    glClearColor(1,1,1,0);
    glClear(GL_COLOR_BUFFER_BIT);

    glLineWidth(3);

    glColor3f(0,0,1);

    glBegin(GL_LINES);

    //Drawing the Oy axis
    glVertex2f(0,-24);
    glVertex2f(0,24);

    //Drawing the Ox axis
    glVertex2f(-32,0);
    glVertex2f(32,0);

    glEnd();
}

```

```

//Drawing a blue rectangle
glRecti(0,0,20,10);

glColor3f(0,1,0);

//All shapes declared after this command will be rotated clockwise by 45
degrees, with respect to the coordinates given.
glRotatef(45,0,0,1);

//A green rectangle will be drawn rotated by 45 degrees.
glRecti(0,0,20,10);

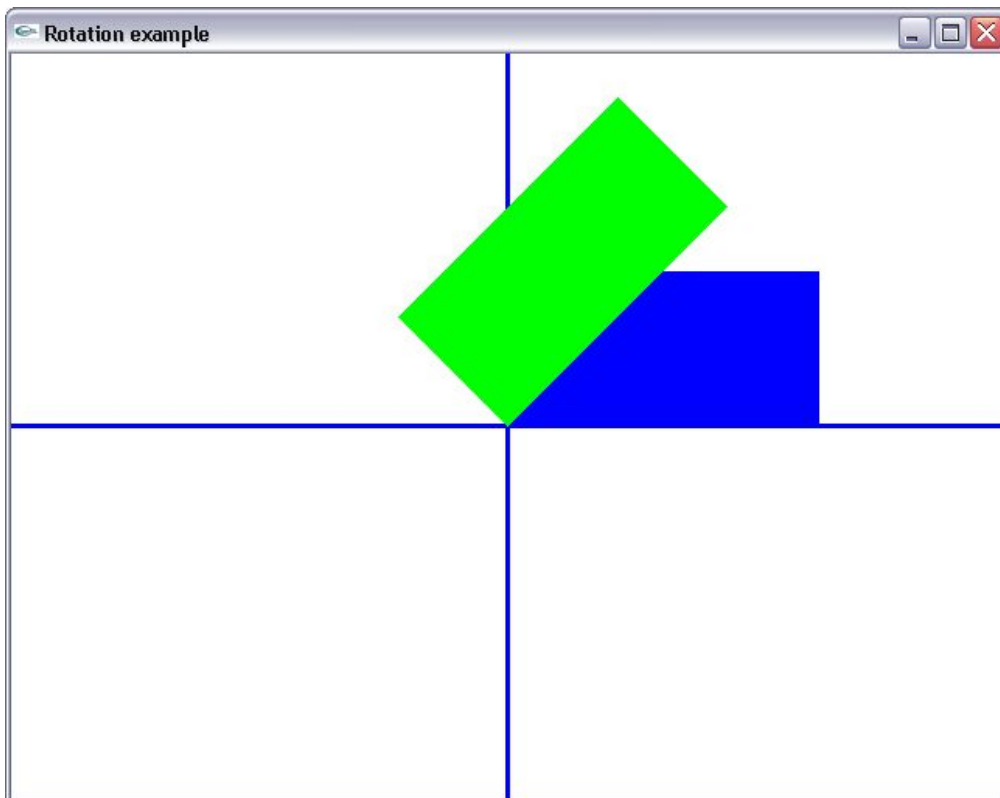
glFlush();
}

int main(int argc, char** argv)
{
    glutInit(&argc,argv);
    glutInitWindowPosition(50,50);
    glutInitWindowSize(640,480);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutCreateWindow("Rotation example");

    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-32,32,-24,24);
    glMatrixMode(GL_MODELVIEW);
    glutDisplayFunc(display);
    glutMainLoop();

    return 0;
}

```



3.4 Σύνθετοι μετασχηματισμοί - Ομογενείς συντεταγμένες

Όπως μπορεί να παρατηρήσει ο αναγνώστης, για την αναπαράσταση των μετασχηματισμών χρησιμοποιούνται μητρώα διαστάσεων 4×4 , δηλαδή μητρώα μεγαλύτερα κατά μία διάσταση, σε σχέση με την διάσταση του διανύσματος που επαρκεί για την περιγραφή ενός σημείου στον τρισδιάστατο χώρο. **Η χρήση μητρώων διάστασης 4×4 οφείλεται στην ανάγκη της αναπαράστασης των σύνθετων μητρώων μετασχηματισμού υπό τη μορφή γινομένου μητρώων στοιχειωδών μετασχηματισμών.**

Εάν ένας σύνθετος μετασχηματισμός περιλαμβάνει στοιχειώδεις μετασχηματισμούς μετατόπισης, τότε **τα μητρώα μετατόπισης είναι αδύνατον να ενοποιηθούν σε ένα σύνθετο μητρώο με τη μορφή γινομένου, εάν χρησιμοποιούμε μητρώα διαστάσεων 3×3** . Λόγω του περιορισμού αυτού, επιλέγουμε την επέκταση της διάστασης των μητρώων και την αναπαράσταση πλέον των σημείων με τη μορφή τετραδιάστατων διανυσμάτων. Συγκεκριμένα ένα σημείο με τρισδιάστατο διάνυσμα (x, y, z) αναπαρίσταται με τη μορφή ομογενών συντεταγμένων (x', y', z', h) όπου $(x', y', z', h) = \left(\frac{x}{h}, \frac{y}{h}, \frac{z}{h}, h \right)$. Στην περίπτωση μας η παράμετρος h λαμβάνει την τιμή 1, οπότε αναγόμαστε στην αναπαράσταση της μορφής $(x, y, z, 1)$. (Υπάρχουν περιπτώσεις που χρησιμοποιούμε την αναπαράσταση των ομογενών συντεταγμένων με τιμές της παραμέτρου h διάφορες του 1, όπως στην περίπτωση της προοπτικής προβολής.)

Έτσι π.χ., για να εφαρμόσουμε έναν σύνθετο μετασχηματισμό περιστροφής και μετατόπισης, αρχικά ορίζουμε το μητρώο περιστροφής R διαστάσεων 4×4 :

$$R = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Κατόπιν, για να εφαρμόσουμε ένα μετασχηματισμό μετατόπισης, ορίζουμε το μητρώο μετατόπισης T :

$$T = \begin{bmatrix} 1 & 0 & 0 & x_{tr} \\ 0 & 1 & 0 & y_{tr} \\ 0 & 0 & 1 & z_{tr} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Πρέπει να επισημάνουμε ότι **οι εντολές δήλωσης στοιχειωδών μετασχηματισμών δεν αντικαθιστούν το τρέχον μητρώο μετασχηματισμού**. Αντίθετα, **το μητρώο του στοιχειώδους μετασχηματισμού**

πολλαπλασιάζει το τρέχον μητρώο μετασχηματισμού μοντέλου από δεξιά. Συνεπώς μετά τη δήλωση ενός στοιχειώδους μετασχηματισμού, το νέο μητρώο μετασχηματισμού μοντέλου θα είναι:

$$C' = C \cdot M$$

Δεδομένου ότι στον πολλαπλασιασμό μητρώων δεν ισχύει η αντιμεταθετική ιδιότητα, έχει μεγάλη σημασία η διαδοχή με την οποία επιβάλλονται οι στοιχειώδεις μετασχηματισμοί. Επιπλέον, το γεγονός ότι οι νέοι μετασχηματισμοί πολλαπλασιάζουν το τρέχον μητρώο από δεξιά, σε συνδυασμό με το γεγονός ότι οι συντεταγμένες των σημείων εκφράζονται στην OpenGL με τη μορφή διανυσμάτων στηλών, σημαίνει ότι **οι πίνακες μετασχηματισμού θα πρέπει να δηλωθούν στον κώδικα με την αντίστροφη διαδοχή από αυτή με την οποία επιδρούν στις συντεταγμένες (x, y, z) .**

Επομένως, αν θεωρήσουμε δύο μητρώα μετασχηματισμού M_1, M_2 και θέλουμε πρώτα να επιβάλλουμε στις συντεταγμένες (x, y, z) το μετασχηματισμό που εκφράζεται με το μητρώο M_1 και κατόπιν το μετασχηματισμό που εκφράζεται με το μητρώο M_2 , ο σύνθετος μετασχηματισμός περιγράφεται ως εξής

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = M_2 \cdot M_1 \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

δηλαδή το σύνθετο μητρώο μετασχηματισμού είναι το εξής:

$$C = M_2 \cdot M_1$$

Επομένως στον κώδικα του προγράμματος, πρώτα θα δηλωθεί ο μετασχηματισμός που εκφράζεται με το μητρώο M_2 και κατόπιν ο μετασχηματισμός που εκφράζεται με το μητρώο M_1 .

Στο παράδειγμα των διαδοχικών μετασχηματισμών περιστροφής και μετατόπισης, θα πρέπει να δηλώσουμε τους μετασχηματισμούς στον κώδικα με την αντίστροφη διαδοχή από αυτή με την οποία επενεργούν στις συντεταγμένες (x, y, z) . Επομένως, η ορθή διαδοχή των πολλαπλασιασμών μητρώων είναι η εξής:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_{tr} \\ 0 & 1 & 0 & y_{tr} \\ 0 & 0 & 1 & z_{tr} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = T \cdot R \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Ο μετασχηματισμός εκφράζεται υπό τη μορφή ενός ενοποιημένου μητρώου ως εξής:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & x_{tr} \\ \sin \theta & \cos \theta & 0 & y_{tr} \\ 0 & 0 & 1 & z_{tr} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

και ο σχηματισμός αυτού του σύνθετου μητρώου και δηλώνεται σε μορφή κώδικα ως εξής:

```
glTranslate{fd}(xtr, ytr, ztr);  
glRotate{fd}(\theta, 0, 0, 1);
```

Παράδειγμα: Σύνθετος μετασχηματισμός (περιστροφή και μετατόπιση)

```
#include <glut.h>  
  
void display()  
{  
  
    glLoadIdentity();  
    glClearColor(1,1,1,0);  
    glClear(GL_COLOR_BUFFER_BIT);  
  
    glLineWidth(3);  
  
    glColor3f(0,0,1);  
  
    glBegin(GL_LINES);  
  
    //Drawing the Oy axis  
    glVertex2f(0,-24);  
    glVertex2f(0,24);  
  
    //Drawing the Ox axis  
    glVertex2f(-32,0);  
    glVertex2f(32,0);  
  
    glEnd();  
  
    //Drawing a blue rectangle  
    glRecti(0,0,20,10);  
  
    glColor3f(0,1,0);  
  
    //Transforms are given in the reverse order (the first one applied on the  
coordinates is the last one declared)  
    //          T          T  
    //[x' y' z'] = T * R * [x y z]  
  
    glTranslatef(10,5,0);  
    glRotatef(45,0,0,1);
```

```

//The composite transform will be applied on this rectangle.
glRecti(0,0,20,10);

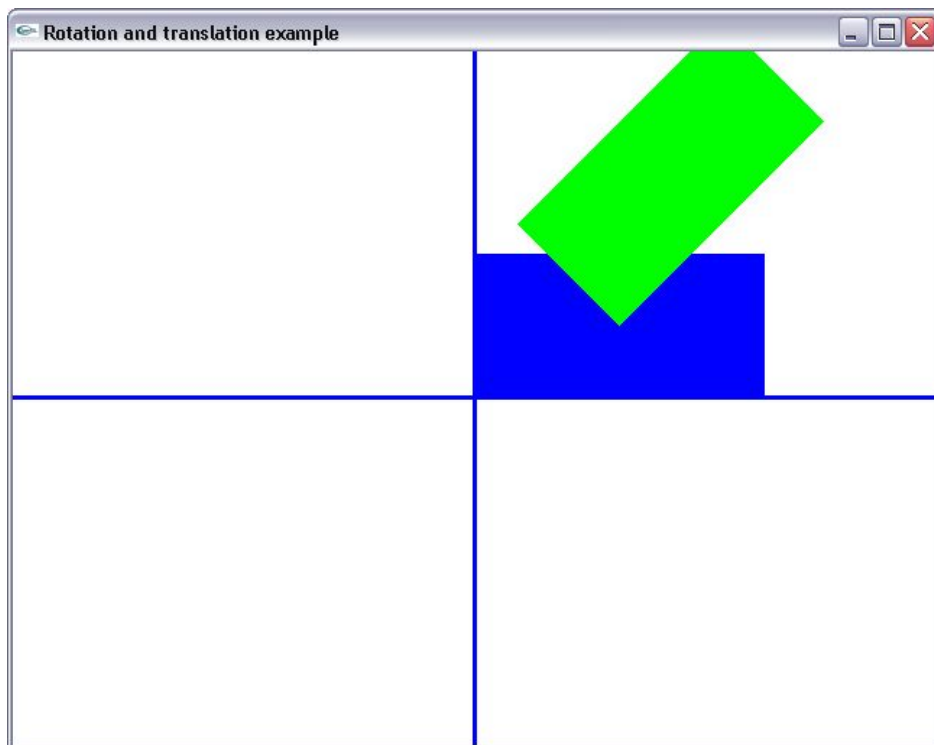
glFlush();
}

int main(int argc, char** argv)
{
    glutInit(&argc,argv);
    glutInitWindowPosition(50,50);
    glutInitWindowSize(640,480);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutCreateWindow("Rotation and translation example");

    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-32,32,-24,24);
    glMatrixMode(GL_MODELVIEW);
    glutDisplayFunc(display);
    glutMainLoop();

    return 0;
}

```



3.5 Άμεσος ορισμός μητρώων μετασχηματισμού

Στην OpenGL ο προγραμματιστής έχει την ευχέρεια να παρακάμψει τις παραπάνω συναρτήσεις δήλωσης μετασχηματισμών και να τροποποιήσει απευθείας τις τιμές του τρέχοντος μητρώου μετασχηματισμού. Έτσι έχει τη δυνατότητα να συνθέσει μετασχηματισμούς που δε μπορούν να εκφραστούν από το συνδυασμό των ανωτέρω εντολών. Υπάρχουν οι εξής δυνατότητες:

α) Αντικατάσταση του τρέχοντος μητρώου μετασχηματισμού

Στην περίπτωση αυτή αναθέτουμε απευθείας τιμές στο μητρώο μετασχηματισμού. Η αντικατάσταση γίνεται με την εντολή:

```
void glLoadMatrixf(GLfloat *elem16);
```

ή

```
void glLoadMatrixd(GLdouble *elem16 );
```

όπου *elem16* μητρώο 4×4 που περιέχει τις προς ανάθεση τιμές. Οι τιμές δηλώνονται στήλη προς στήλη, δηλαδή με την παρακάτω διαδοχή:

$$elem16 = \begin{bmatrix} elem[0] & elem[4] & elem[8] & elem[12] \\ elem[1] & elem[5] & elem[9] & elem[13] \\ elem[2] & elem[6] & elem[10] & elem[14] \\ elem[3] & elem[7] & elem[11] & elem[15] \end{bmatrix}$$

β) Πολλαπλασιασμός τρέχοντος μητρώου μετασχηματισμού με αυθαίρετο μητρώο

Στην περίπτωση που δεν θέλουμε να αντικαταστήσουμε το τρέχον μητρώο μετασχηματισμού από το μητρώο *elem16*, αλλά επιθυμούμε να πολλαπλασιάσουμε το τελευταίο με το μητρώο *elem16*, χρησιμοποιούμε την εντολή *glMultMatrix**:

```
glMultMatrixf(GLfloat *elem16);
```

ή

```
glMultMatrixd (GLdouble *elem16);
```

όπου *elem16* μητρώο μετασχηματισμού δοσμένο υπό τη μορφή ομογενών συντεταγμένων.

Επισημαίνουμε ότι η εντολή *glMultMatrix** πολλαπλασιάζει το μητρώο *elem16* με το μητρώο μετασχηματισμού *C* από δεξιά. Δηλαδή, το μητρώο μετασχηματισμού *C'* που προκύπτει μετά τη διαδικασία, είναι το εξής:

$$C' = C \cdot elem16$$

Παράδειγμα: Ορισμός μητρώου κλίσης με την απευθείας ανάθεση τιμών στο μητρώο μετασχηματισμού μοντέλου

```
#include <glut.h>
```

```

GLfloat *elem16;

void display()
{
    glLoadIdentity();
    glClearColor(1,1,1,0);
    glClear(GL_COLOR_BUFFER_BIT);

    glColor3f(0,0,1);

    glBegin(GL_LINES);

    //Drawing the y axis
    glVertex2f(0,-24);
    glVertex2f(0,24);

    //Drawing the x axis
    glVertex2f(-32,0);
    glVertex2f(32,0);
    glEnd();

    //Drawing a rectangle.
    glRecti(0,0,20,10);

    glColor3f(0,1,0);

    //Loading the shearing matrix in place of the current modelview matrix
    glLoadMatrixf(elem16);

    //This rectangle will be drawn sheared.
    glRecti(0,0,20,10);

    glFlush();
}

int main(int argc, char** argv)
{
    glutInit(&argc,argv);
    glutInitWindowPosition(50,50);
    glutInitWindowSize(640,480);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutCreateWindow("Shearing transform");

    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(-32,32,-24,24);

    glMatrixMode(GL_MODELVIEW);
    elem16=new GLfloat[16];

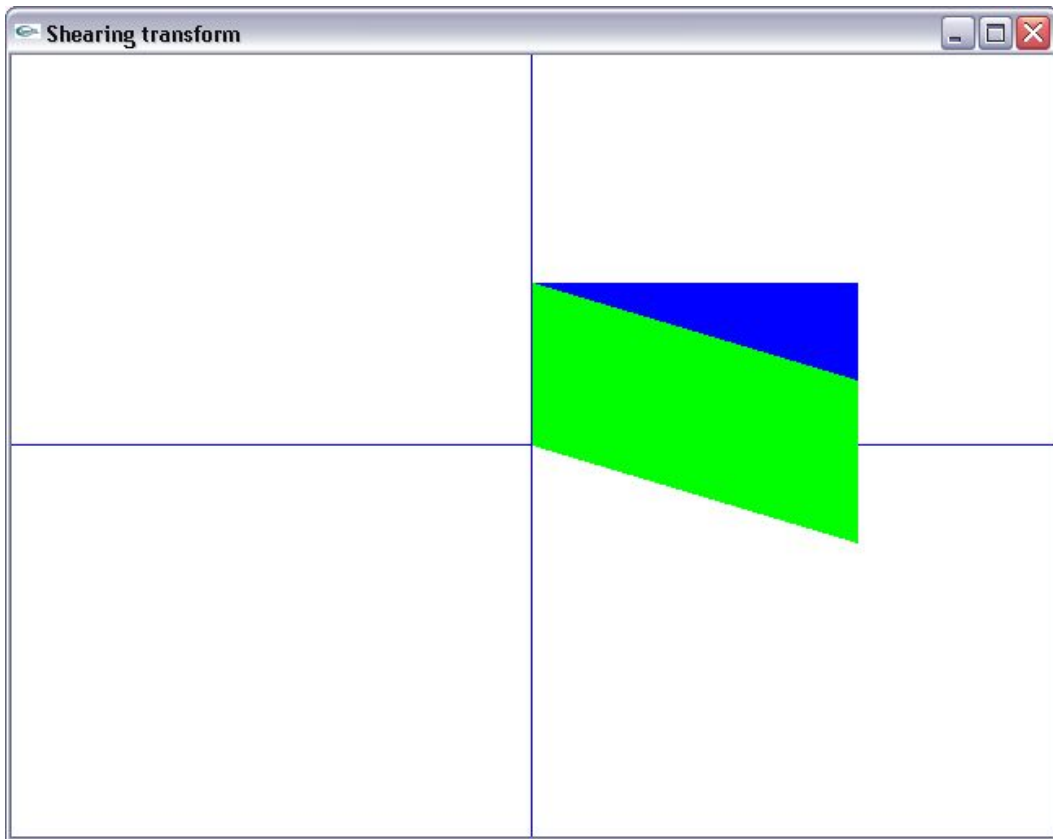
    //Defining the shearing matrix
    //
    //      | 1  0  0  0 |
    //      S=|-0.3 1  0  0|   y'=-0.3x+y
    //      | 0  0  1  0 |
    //      | 0  0  0  1 |

    elem16[0]=1;   elem16[1]=-0.3; elem16[2]=0;   elem16[3]=0;
    elem16[4]=0;   elem16[5]=1;   elem16[6]=0;   elem16[7]=0;
    elem16[8]=0;   elem16[9]=0;   elem16[10]=1;  elem16[11]=0;
    elem16[12]=0;  elem16[13]=0;   elem16[14]=0;  elem16[15]=1;

    glutDisplayFunc(display);
    glutMainLoop();
}

```

```
}  
    return 0;  
}
```



3.6 Στοιβες μητρώων μετασχηματισμού

Για κάθε κατηγορία μητρώου μετασχηματισμού (μοντέλου και προβολής) η μηχανή της OpenGL προβλέπει την ύπαρξη μιας στοιβάς, η οποία προσφέρει τη δυνατότητα αποθήκευσης πολλαπλών προφίλ για κάθε μητρώο μετασχηματισμού, με σκοπό τη μελλοντική τους χρήση.

Κατά την έναρξη εκτέλεσης ενός προγράμματος, στη στοιβα κάθε μητρώου μετασχηματισμού ορίζεται μόνο ένα μητρώο στην κορυφή της. Το μητρώο αυτό αναφέρεται ως το ενεργό μητρώο μετασχηματισμού και είναι αυτό που βρίσκεται σε ισχύ την εκάστοτε χρονική στιγμή (Σχ. 3.4). Οποιαδήποτε τροποποίηση εκτελούμε με τη δήλωση εντολών στοιχειωδών μετασχηματισμών επιδρά στο ενεργό μητρώο μετασχηματισμού. Όσα μητρώα βρίσκονται χαμηλότερα στη στοιβα μένουν ανεπηρέαστα.

Η αποθήκευση πολλαπλών προφίλ για ένα μητρώο μετασχηματισμού εκτελείται τροποποιώντας το ενεργό μητρώο και προωθώντας το προς τις χαμηλότερες θέσεις της στοιβάς μητρώου. Η προώθηση του ενεργού μητρώου προς τα κάτω εκτελείται με την εντολή *glPushMatrix*:

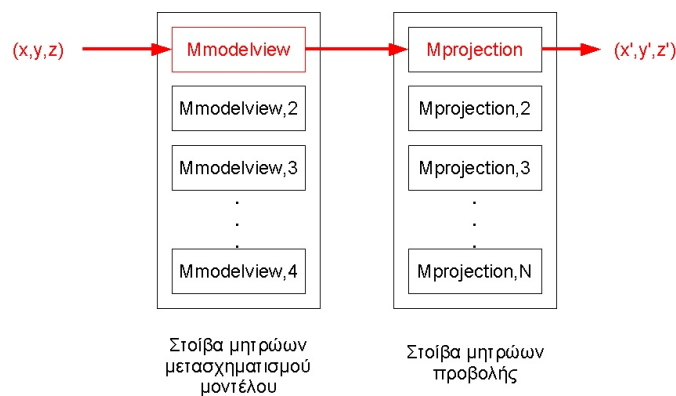
```
void glPushMatrix ( );
```

η οποία μεταφέρει το ενεργό μητρώο ένα επίπεδο προς τα κάτω στη στοίβα. Αμέσως μετά την εκτέλεση της εντολής, το ενεργό μητρώο και το αμέσως επόμενο μητρώο στη στοίβα έχουν τις ίδιες τιμές. Ωστόσο οι επόμενες εντολές τροποποίησης επιδρούν μόνο στο ενεργό μητρώο, αφήνοντας τα υπόλοιπα μητρώα της στοίβας άθικτα. Επανεκτέλεση της εντολής **glPushMatrix** μεταθέτει κατά μία θέση πιο κάτω στη στοίβα το ενεργό μητρώο μετασχηματισμού, καθώς και όλα τα μητρώα που αποθηκεύτηκαν προηγουμένως.

Η ανάκληση των μητρώων από τη στοίβα γίνεται βάσει της λογικής last in-first out, με την εντολή **glPopMatrix**.

void glPopMatrix ();

Η εκτέλεση της **glPopMatrix** μεταθέτει κάθε μητρώο κατά μία θέση παραπάνω στη στοίβα. Επομένως το μητρώο που βρισκόταν στην πρώτη θέση κάτω από το ενεργό μητρώο, αναλαμβάνει πλέον το ρόλο του ενεργού μητρώου. Το προηγούμενο ενεργό μητρώο διαγράφεται.



Σχ. 3.4: Στοίβες μητρώων μετασχηματισμού

Η επιλογή της στοίβας μητρώων στην οποία επενεργούν οι εντολές **glPushMatrix** και **glPopMatrix** καθορίζεται μέσω της προαναφερθείσας εντολής **glMatrixMode**.

Π.χ.

```
glMatrixMode (GL_MODELVIEW);
```

```
glPushMatrix(); //Προώθηση του ενεργού μητρώου μετασχηματισμού μοντέλου κατά ένα επίπεδο  
βαθύτερα στη στοίβα μητρώων μετασχηματισμού μοντέλου
```

```
glMatrixMode (GL_PROJECTION);
```

```
glPopMatrix(); //Ανάκληση του πρώτου μητρώου προβολής από τη στοίβα μητρώων προβολής και  
χρήση του ως «ενεργό μητρώο».
```

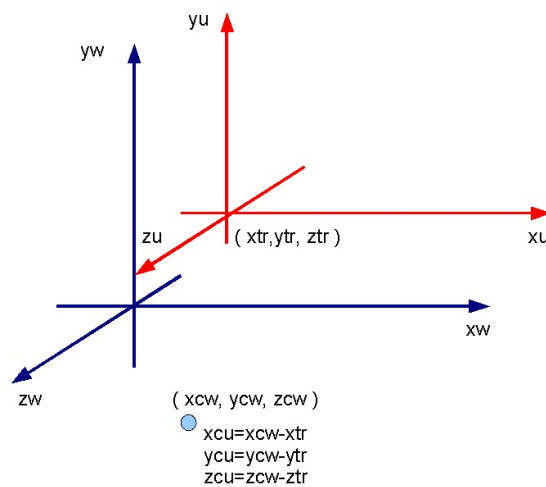
3.7 Αλλαγή συστήματος συντεταγμένων

Οι εντολές δήλωσης στοιχειωδών μετασχηματισμών που αναλύσαμε προηγουμένως, μπορούν να χρησιμοποιηθούν και για την αλλαγή συστήματος συντεταγμένων. Αυτή η διαδικασία είναι χρήσιμη εάν

θέλουμε να αναγάγουμε την περιγραφή της σκηνής ως προς σύστημα συντεταγμένων που προκύπτει από το αρχικά καθορισμένο σύστημα με διαδικασίες στοιχειωδών μετασχηματισμών.

3.7.1 Μετατόπιση συστήματος συντεταγμένων

Έστω ότι θέλουμε να αναπαραστήσουμε μια σκηνή ως προς σύστημα συντεταγμένων που η αρχή των αξόνων του βρίσκεται στο σημείο (x_{tr}, y_{tr}, z_{tr}) ως προς το αρχικά καθορισμένο σύστημα συντεταγμένων σκηνής και οι άξονές του έχουν την ίδια διεύθυνση με το αρχικό (Σχ. 3.5).



Σχ. 3.5: Μετατόπιση συστήματος συντεταγμένων

Είναι προφανές ότι η περιγραφή της σκηνής ως προς το μετατοπισμένο σύστημα συντεταγμένων ισοδυναμεί με μετατόπιση όλων των σημείων της σκηνής κατά $(-x_{tr}, -y_{tr}, -z_{tr})$. Προκειμένου λοιπόν να εξάγουμε αυτή την αναπαράσταση, αρκεί να μετατοπίσουμε όλη τη σκηνή, ούτως ώστε το νέο σύστημα συντεταγμένων να συμπίπτει με το αρχικό. Συνεπώς ο μετασχηματισμός συστήματος συντεταγμένων είναι.

$$T = \begin{bmatrix} 1 & 0 & 0 & -x_{tr} \\ 0 & 1 & 0 & -y_{tr} \\ 0 & 0 & 1 & -z_{tr} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

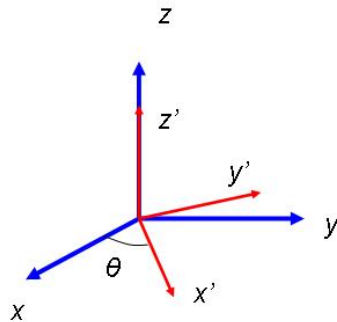
Στην OpenGL η μετατόπιση του συστήματος συντεταγμένων εκτελείται με μια απλή εντολή μετατόπισης:

```
glTranslate{fd} (-xtr, -ytr, -ztr);
```

3.7.2 Περιστροφή συστήματος συντεταγμένων

Θεωρούμε ένα σύστημα συντεταγμένων με άξονες x' , y' και z' που προκύπτουν με περιστροφή του συστήματος κατά γωνία θ . Όπως και στην προηγούμενη περίπτωση η αναπαράσταση της σκηνής ως προς το νέο σύστημα συντεταγμένων προκύπτει εάν θεωρήσουμε την περιστροφή της σκηνής κατά γωνία $-\theta$.

Ας θεωρήσουμε σύστημα συντεταγμένων που προκύπτει με περιστροφή του αρχικού κατά γωνία θ και με άξονα περιστροφής τον άξονα Oz (Σχ. 3.6).



Σχ. 3.6: Περιστροφή συστήματος συντεταγμένων

Στην περίπτωση αυτή, η αναπαράσταση στο νέο σύστημα (x', y', z') ενός σημείου (x, y, z) προκύπτει βάσει του μετασχηματισμού:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$