Είσοδος – Έξοδος Διαχείριση γεγονότων – Απόδοση 2Δ σκηνών

Περιεχόμενα ενότητας:

Δημιουργία παραθύρων γραφικών

Διαχείριση γεγονότων - Αλληλεπιδραστικές εφαρμογές

Αποκοπή - Μετασχηματισμός παρατήρησης

Εκκίνηση βιβλιοθήκης GLUT

•Διαδικασίες που εξαρτώνται από την αρχιτεκτονική του συστήματος (platform-dependent). Οι αντίστοιχες εντολές περιλαμβάνονται στη βιβλιοθήκη GLUT.

Αρχικοποίηση της GLUT με την εντολή glutInit:

```
void glutInit(int argc, char **argv);
```

argv: μητρώο με τα ορίσματα που περνάμε στο πρόγραμμα μέσω της γραμμής εντολών DOS (command line arguments)

argc: πλήθος των ορισμάτων που περνάμε μέσω της; γραμμής εντολών DOS.

•Το πρώτο όρισμα (argv[0]) επιστρέφει το όνομα του εκτελέσιμου αρχείου.

Παράδειγμα εκκίνησης GLUT

Εκτελέσιμο αρχείο:
MyOpenGLApp.exe

Σύνταξη εντολής στη γραμμή εντολών:
C:\> MyOpenGLApp someArgument1 someArgument2

Παράμετροι γραμμής εντολών
argv[0] = “MyOpenGLApp”
argv[1] = “someArgument1”
argv[2] = “someArgument2”.

argc = 3

Θέση και έκταση παραθύρου σχεδίασης

void glutInitWindowPosition(int x, int y);

x,y :οι συντεταγμένες του άνω αριστερού άκρου του παραθύρου

glutInitWindowSize: Ορίζει τις διαστάσεις της επιφάνειας σχεδίασης

void glutInitWindowSize(int width, int height);

width,height: το αρχικό πλάτος και ύψος της επιφάνειας σχεδίασης της εφαρμογής σε pixels

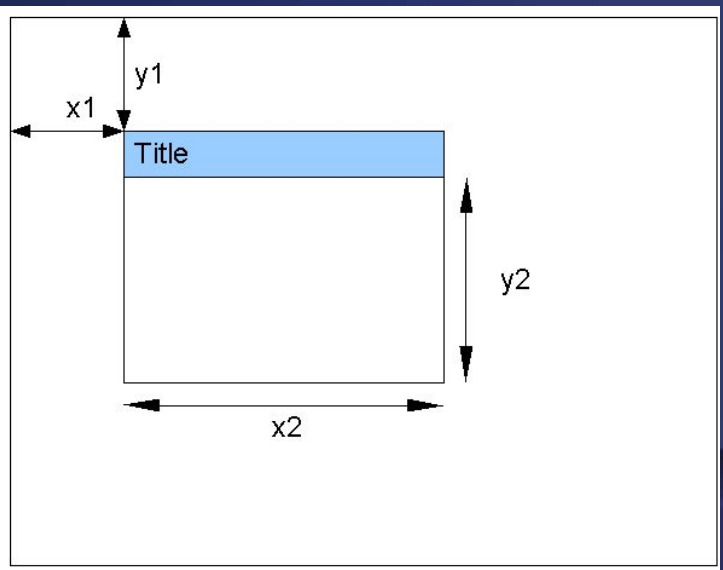
Οι διαστάσεις της επιφάνειας σχεδίασης μπορούν να μεταβληθούν από το χρήστη με τη χρήση του ποντικιού (reshape).

Εμφάνιση του παραθύρου σχεδίασης

•Το παράθυρο σχεδίασης εμφανίζεται μετά την εκτέλεση της glutCreateWindow.

int glutCreateWindow(const char *title);

title: ο τίτλος του παραθύρου




```
glutInitWindowPosition ( x1, y1 );
glutInitWindowSize ( x2, y2 );
glutCreateWindow ( “Title” );
```

Παράμετροι λειτουργίας παραθύρων (1)

Κατά την αρχικοποίηση του παραθύρου σχεδίασης ορίζουμε τις παραμέτρους απεικόνισης της σκηνής:

- Μοντέλο ενταμίευσης (στατικές σκηνές ή για κινούμενα γραφικά)
- Χρησιμοποιούμενο χρωματικό μοντέλο

Οι ρυθμίσεις αυτές δίνονται με την εντολή glutInitDisplayMode

```
void glutInitDisplayMode(unsigned int mode);
```

Παράμετροι λειτουργίας παραθύρων (2)

void glutInitDisplayMode(unsigned int mode);

mode: καθορίζει το χρωματικό μοντέλο ή/και το μοντέλο ενταμίευσης

GLUT_SINGLE: απλή ενταμίευση (σχεδίαση στατικών σκηνών)
GLUT_DOUBLE: τεχνική διπλής ενταμίευσης (για τη σχεδίαση κινουμένων γραφικών).

GLUT_RGB: χρωματικό μοντέλο RGB
GLUT_RGBA: χρωματικό μοντέλο RGBA.
GLUT_INDEX: Χρήση του μοντέλου χρωματικών πινάκων (colour tables).

Παράμετροι λειτουργίας παραθύρων (3)

Οι σταθερές μπορούν να συνδυαστούν στο όρισμα της glutInitDisplayMode με τη χρήση τελεστών OR (|)

Πχ.
-Χρήση του χρωματικού μοντέλου RGB
-Εφαρμογή διπλής ενταμίευσης
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);

Προκαθορισμένη ρύθμιση:
-Χρωματικό μοντέλο RGB,
-Απλή ενταμίευση

Γεγονότα

•Γεγονός: η καταγραφή κάποιας δραστηριότητας του συστήματος (συνήθως από κάποια συσκευή εισόδου)

•Μία καταγραφή γεγονότος περιέχει πληροφορίες που το προσδιορίζουν επακριβώς (πχ συντεταγμένες θέσης του δείκτη του ποντικιού όταν πιέστηκε πλήκτρο του)

•Τα γεγονότα καταγράφονται σε μια ουρά (queue) , την οποία διαχειρίζεται το λειτουργικό σύστημα

•Το προγραμματιστικό μοντέλο της ανταπόκρισης σε γεγονότα παρέχει τη δυνατότητα υλοποίησης αλληλεπιδραστικών εφαρμογών (event-driven programming).

Κατηγορίες γεγονότων (1)

α) Γεγονός σχεδίασης:

Εμφανίζεται κάθε φορά που το σύστημα θα διαπιστώσει ότι απαιτείται σχεδίαση ή επανασχεδιασμός της σκηνής.

Εγείρεται:

-κατά τη μετακίνηση του παραθύρου στην οθόνη

-κατά την επαναφορά του παραθύρου της εφαρμογής στο προσκήνιο (restore)

-με ρητή εντολή επανασχεδιασμού της σκηνής από τον προγραματιστή

glutPostRedisplay: εντολή επανασχεδιασμού της σκηνής

```
void glutPostRedisplay( );
```

Κατηγορίες γεγονότων (2)

β) Γεγονός αλλαγής διαστάσεων παραθύρου (reshape event):

Εγείρεται:

- Κατά την πρώτη φορά που σχεδιάζεται το παράθυρο (κατά την εκκίνηση της εφαρμογής)
- κάθε φορά που ο χρήστης μεταβάλλει χειροκίνητα τις διαστάσεις του παραθύρου σχεδίασης.

γ) Γεγονότα πληκτρολογίου (keyboard events):

Εμφανίζονται κάθε φορά που πιέζουμε ένα πλήκτρο.

Δύο υποκατηγορίες:

- Γεγονός πληκτρολόγησης χαρακτήρα
- Γεγονός πληκτρολόγησης ειδικού πλήκτρου

Κατηγορίες γεγονότων (3)

δ) Γεγονότα ποντικιού:

Εμφανίζονται κατά τη χρήση του ποντικιού.

- Πίεση/απελευθέρωση των πλήκτρων του ποντικιού,
- ενεργή κίνηση του δείκτη (κίνηση του δείκτη του ποντικιού με ένα από τα πλήκτρα του πιεσμένο)
- παθητική κίνηση (κίνηση του δείκτη του ποντικιού χωρίς πιεσμένο πλήκτρο).

ε) Γεγονότα χρονισμού:

Εγείρονται σε συγκεκριμένη χρονική στιγμή. Λειτουργούν ως μετρητές (timers) για τον προγραμματισμό εργασιών (scheduling).

στ) "Γεγονός αδρανείας":

Έγείρονται επ' άπειρον.

Χρήσιμα για την επ' άπειρον εκτέλεση κώδικα (π.χ. για τη διαρκή ανανέωση των καρέ σε εφαρμογές κινουμένων γραφικών).

Διαχείριση γεγονότων – Κύκλος διαχείρισης γεγονότων

Συναρτήσεις κλήσης ή διαχειριστές γεγονότων:

Συναρτήσεις που αντιστοιχίζονται σε γεγονότα και εκτελούνται όποτε αυτά εγείρονται.

Κύκλος διαχείρισης γεγονότων:

```
while (true)  //Εκτέλεση βρόχου επ' άπειρον
{
• if (γεγονός σχεδιασμού/επανασχεδιασμού)
      { εκτέλεση κώδικα σχεδιασμού/επανασχεδιασμού; }
• if (αλλαγή διαστάσεων παραθύρου)
      {εκτέλεση διαχειριστή αλλαγής διάστασεων παραθύρου; }
• if (εμφάνιση γεγονότος πληκτρολογίου)
      { εκτέλεση διαχειριστή πληκτρολογίου/ποντικιού }
• if (εμφάνιση γεγονότος ποντικιού)
      {εκτέλεση συνάρτησης διαχείρισης ποντικιού}
      ……..
• εκτέλεση συνάτησης διαχείρισης "γεγονότος αδρανείας" ;
}
```

Ενεργοποίηση κύκλου διαχείρισης γεγονότων

• Για την ανταπόκρίνεται σε γεγονότα, πρέπει να ενεργοποιηθεί ο κύκλος διαχείρισης γεγονότων (event processing loop).

• Η ενεργοποίηση του κύκλου διαχείρισης γεγονότων γίνεται με την εντολή glutMainLoop:

void glutMainLoop( );

• *Μετά την εκτέλεση της* glutMainLoop, *η εφαρμογή θα αναμένει επ' άπειρον και θα ανταποκρίνεται σε γεγονότα*.

•Η glutMainLoop χρησιμοποιείται πάντα στο τέλος της συνάρτησης main.

•Γεγονότα για οποία δεν έχει καταχωρηθεί διαχειριστής δεν έχουν καμία επίδραση στην εκτέλεση του προγράμματος.

Οι διαχειριστές γεγονότων πρέπει να καταχωρηθούν πριν την εκτέλεση της glutMainLoop.

Διαχείριση σχεδίασης/επανασχεδιασμού

void glutDisplayFunc(display):

- Καταχωρεί τη display για τη διαχείριση σχεδιασμού/επανασχεδιασμού της σκηνής . Όλες οι ρουτίνες σχεδίασης συνήθως τοποθετούνται στο εσωτερικό της συνάρτησης display.

- Ο διαχειριστής display δεν επιστρέφει τιμή και δε δέχεται ορίσματα.

void display( );

Η display εκτελείται:

- κατά τη μετακίνηση του παραθύρου σχεδίασης
- κατά την επαναφορά κρυμμένου παραθύρου στο προσκήνιο
- όταν ο προγραμματιστής απαιτεί τον επενασχεδιασμό της σκηνής, με την εντολή glutPostRedisplay.

Διαχείριση αλλαγής διαστάσεων παραθύρου

void glutReshapeFunc(reshape);

Καταχωρεί τη συνάρτηση reshape για τη διαχείριση γεγονότων αλλαγής διαστάσεων παραθύρου.

void reshape(int width, int height);

width height. οι νέες διαστάσεις της επιφάνειας σχεδίασης

Διαχείριση πληκτρολόγησης χαρακτήρων

3) void glutKeyboardFunc(keyboard):

Καταχωρεί τη συνάρτηση keyboard για τη διαχείριση γεγονότων πληκτρολόγησης χαρακτήρων (character keys).

void keyboard(unsigned char key, int x, int y);

Ορίσματα της keyboard:

key: ο χαρακτήρας του πλήκτρου που πιέστηκε.

x, y: προσδιορίζουν τη θέση του δείκτη του ποντικιού στην επιφάνεια σχεδίασης, όταν πιέστηκε το πλήκτρο. Ορίζονται σε σύστημα συντεταγμένων με ακέραιες τιμές, με αρχή των αξόνων την πάνω αριστερή γωνία.

•Ανάλογα με το πλήκτρο που πιέστηκε, υπάρχει η δυνατότητα εκτέλεσης διαφορετικού κώδικα, χρησιμοποιώντας στο εσωτερικό της συνάρτησης keyboard δομές τύπου switch, if/else κλπ.

Διαχείριση ειδικών πλήκτρων

void glutSpecialFunc(specialKey):

Καταχωρεί τη συνάρτηση specialKey για τη διαχείριση γεγονότων πληκτρολόγησης ειδικών πλήκτρων (special keys).

void specialKey(int key, int x, int y);

x,y: η θέση στην επιφάνεια σχεδίασης όπου βρισκόταν ο δείκτης του ποντικιού όταν πιέστηκε το πλήκτρο (ακέραιες τιμές συντεταγμένων, με αρχή στην πάνω αριστερή γωνία της επιφάνειας)

key: αριθμητική τιμή που καθορίζει το ειδικό πλήκτρο που πιέστηκε.

Στην OpenGL, κάθε ειδικό πλήκτρο εκπροσωπείται από μία συγκεκριμένη συμβολική σταθερά.

Κατηγορίες ειδικών πλήκτρων

Κατηγορία	Σύμβολο στο πληκτρολόγιο	Αντίστοιχη αριθμητική σταθερά στην OpenGL
Function keys	F1- F2 - … - F12	GLUT_KEY_F1 …. GLUT_KEY_F12
Πλήκτρα κατεύθυνσης (arrow keys)	↑, ↓, ←, →	GLUT_KEY_UP GLUT_KEY_DOWN GLUT_KEY_LEFT GLUT_KEY_RIGHT
Διάφορα πλήκτρα	Page Up Page Down Home End Insert	GLUT_KEY_PAGE_UP GLUT_KEY_PAGE_DOWN GLUT_KEY_HOME GLUT_KEY_END GLUT_KEY_INSERT

Διαχειριστής χρήσης πλήκτρων ποντικιού

void glutMouseFunc(mouseClicked):

Καταχωρεί τη συνάρτηση που θα διαχειρίζεται την πίεση πλήκτρων ποντικιού (έστω mouseClicked).

void mouseClicked(int button, int state, int x, int y);

button: το κουμπί του ποντικιού που πιέστηκε:

GLUT_LEFT_BUTTON: Πιέστηκε το αριστερό κουμπί.

GLUT_MIDDLE_BUTTON: Πιέστηκε το μεσαίο κουμπί.

GLUT_RIGHT_BUTTON: Πιέστηκε το το δεξί κουμπί.

state: καθορίζει αν το κουμπί του ποντικιού πιέστηκε ή απελευθερώθηκε (διαφορετικά γεγονότα)

GLUT_DOWN: Πίεση του πλήκτρου του ποντικιού.

GLUT_UP: Απελευθέρωση του πλήκτρου του ποντικιού.

x, y: η θέση του δείκτη του ποντικιού κατά την εμφάνιση του γεγονότος

Διαχείριση ενεργής κίνησης ποντικιού

6)void glutMotionFunc(activeMouseMotion):

Καταχωρεί τη συνάρτηση που θα διαχειρίζεται την ενεργή κίνηση του ποντικιού (έστω activeMouseMotion)

void activeMouseMotion(int x, int y);

x, y: η θέση του δείκτη του ποντικιού κατά την εμφάνιση του γεγονότος

Διαχείριση παθητικής κίνησης ποντικιού

6)void glutPassiveMotionFunc(passiveMouseMotion):

Καταχωρεί τη συνάρτηση που θα διαχειρίζεται την παθητική κίνηση του ποντικιού (έστω passiveMouseMotion)

void passiveMouseMotion(int x, int y);

x, y: η θέση του δείκτη του ποντικιού κατά την εμφάνιση του γεγονότος

Διαχείριση «γεγονότος αδρανείας»

8) void glutIdleFunc(animate):

Καταχωρεί τη συνάρτηση animate για τη διαχείριση του "γεγονότος αδρανείας".

void animate( );

• Η animate θα εκτελείται επαναληπτικά σε κάθε κύκλο διαχείρισης γεγονότων.

•Η animate χρήσιμοποιείται σε περιπτώσεις που εκτελούμε συνεχείς μεταβολές της σκηνής και επιθυμούμε τακτικό επανασχεδιασμό της (κινούμενα γραφικά)

Διαχείριση γεγονότος χρονισμού

void glutTimerFunc(unsigned int execTime, void timed(int val), value):

Καταχωρεί τη συνάρτηση timed για τη διαχείριση του γεγονότος χρονισμού.

void timed(int val);

Η timed θα εκτελεστεί σε προγραμματισμένο χρονικό διάστημα τουλάχιστον execTime msec από τη στιγμή που ενεργοποιείται ο κύκλος διαχείρισης γεγονότων.

value: βοηθητικό όρισμα που χρησιμοποιείται εάν θέλουμε να περάσουμε μια τιμή στο εσωτερικό του διαχειριστή timed

Αλλαγή και επανασχεδιασμός σκηνής

•Εάν χρησιμοποιούμε ένα διαχειριστή για να τροποποιήσουμε το περιεχόμενο της σκηνής, οι αλλαγές δεν εμφανίζοται αυτόματα στην οθόνη

•Πρέπει επιπρόσθετα να παραγάγουμε ένα γεγονός επανασχεδιασμού δίνοντας την εντολή glutPostRedisplay στο τέλος του κώδικα του διαχειριστή.

Π.χ.

```
someCallbackFunction
{
//Τροποποίηση σκηνής
glutPostRedisplay();
}
//Στη συνέχεια θα εκτελεστεί ο διαχειριστής επανασχεδιασμού
```

Απεικόνιση διδιάστατων σκηνών

- Συντεταγμένες σκηνής
- Συντεταγμένες συσκευής
- Αποκοπή
- Μετασχηματισμός παρατήρησης

Συντεταγμένες σκηνής

Σημαντική παρατήρηση:

Όταν δηλώνουμε τη θέση μιας κορυφής με τις εντολές σχεδίασης glVertex *οι συντεταγμένες που δίνουμε δεν ορίζουν κάποιο συγκεκριμένο pixel της οθόνης*.

Σύστημα συντεταγμένων σκηνής: σύστημα συντεταγμένων με απεριόριστο εύρος τιμών και αναλυτικότητα (δεκαδικές υποδιαιρέσεις).

Τα σημεία ορίζονται στο *σύστημα συντεταγμένων σκηνής*

Π.χ.

glVertex2*(0.1, 100000);

Έγκυρη δήλωση διότι τα ορίσματα δεν αντιστοιχούν στη θέση κάποιου pixel

οι δεκαδικές υποδιαιρέσεις είναι επιτρεπτές

δεν υπάρχουν περιορισμοί για το εύρος τιμών

Συντεταγμένες συσκευής

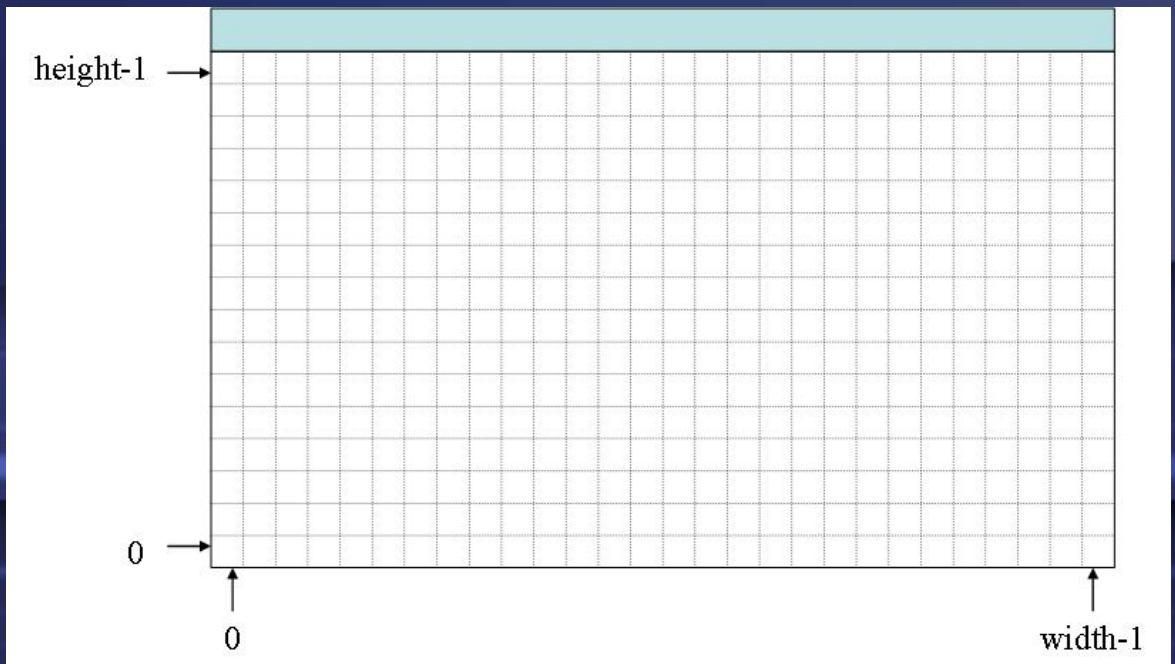
Συντεταγμένες συσκευής: Κάθε pixel της επιφάνειας σχεδίασης προσδιορίζεται από ένα ζεύγος ακεραίων συντεταγμένων.

Πχ σε επιφάνεια σχεδίασης διαστάσεων width, height

Εύρος κατακόρυφων συντεταγμένων [0, width-1] με ακέραιο βήμα

Εύρος οριζόντιων συντεταγμένων[0, height-1]

Η αρχή του συστήματος συντεταγμένων συσκευής βρίσκεται στην κάτω αριστερή θέση της επιφάνειας σχεδίασης.



Σχεδίαση σκηνής

Συντετεταγμένες σκηνής/συσκευής:

Απαρίτητητες έννοιες για να κατανοήσουμε το μηχανισμό απόδοσης μιας σκηνής σε συσκευή εξόδου

Σχεδίαση σκηνής: διαδικασία δύο βημάτων

-Αποκοπή: Επιλογή τμήματος της σκηνής για απεικόνιση στη συσκευή εξόδου

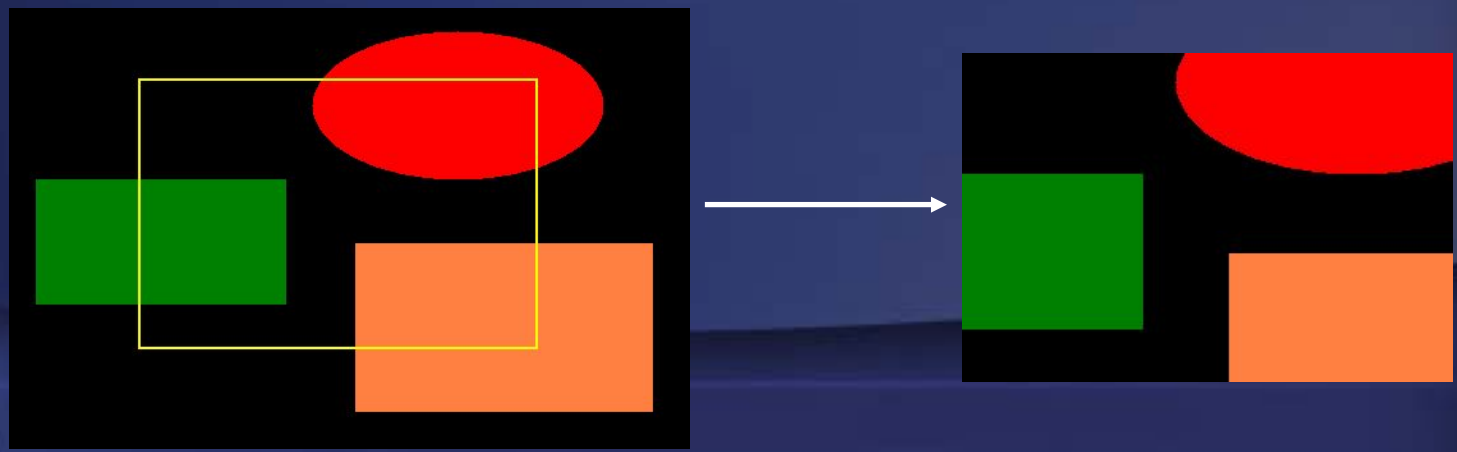
-Μετασχηματισμός παρατήρησης: Απόδοση του αποκομμένου τμήματος της σκηνής στην επιφάνεια σχεδίασης

Αποκοπή – Παράθυρο αποκοπής

Αποκοπή: η διαδικασία επιλογής ενός τμήματος της σκηνής για την αναπαράστασή του στη συσκευή εξόδου

Στις δύο διαστάσεις η αποκοπή εκτελείται ορίζοντας τα όρια ενός παραθύρου αποκοπής (clipping window)

Παράθυρο αποκοπής: ένα ορθογώνιο, μέσα στο οποίο περικλείεται το τμήμα της σκηνής που θέλουμε να απομονώσουμε



Το παράθυρο αποκοπής καθορίζεται ως προς το σύστημα συντεταγμένων σκηνής

Αποκοπή στις δύο διαστάσεις

void gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top);

left, right, bottom, top: αριστερό, δεξιό, κάτω και άνω όριο του παραθύρου αποκοπής

Τα όρια του παραθύρου αποκοπής ορίζονται στο σύστημα συντεταγμένων σκηνής.

Το παράθυρο αποκοπής ορίζουμε τις τιμές ενός μητρώου προβολής (οι διαδικασίες αποκοπής και προβολής συσχετίζονται – βλ. Ενότητα 4)

Πριν την αποκοπή πρέπει "να μεταβούμε στην κατάσταση επεξεργασίας μητρώου προβολής" με την εντολή glMatrixMode.

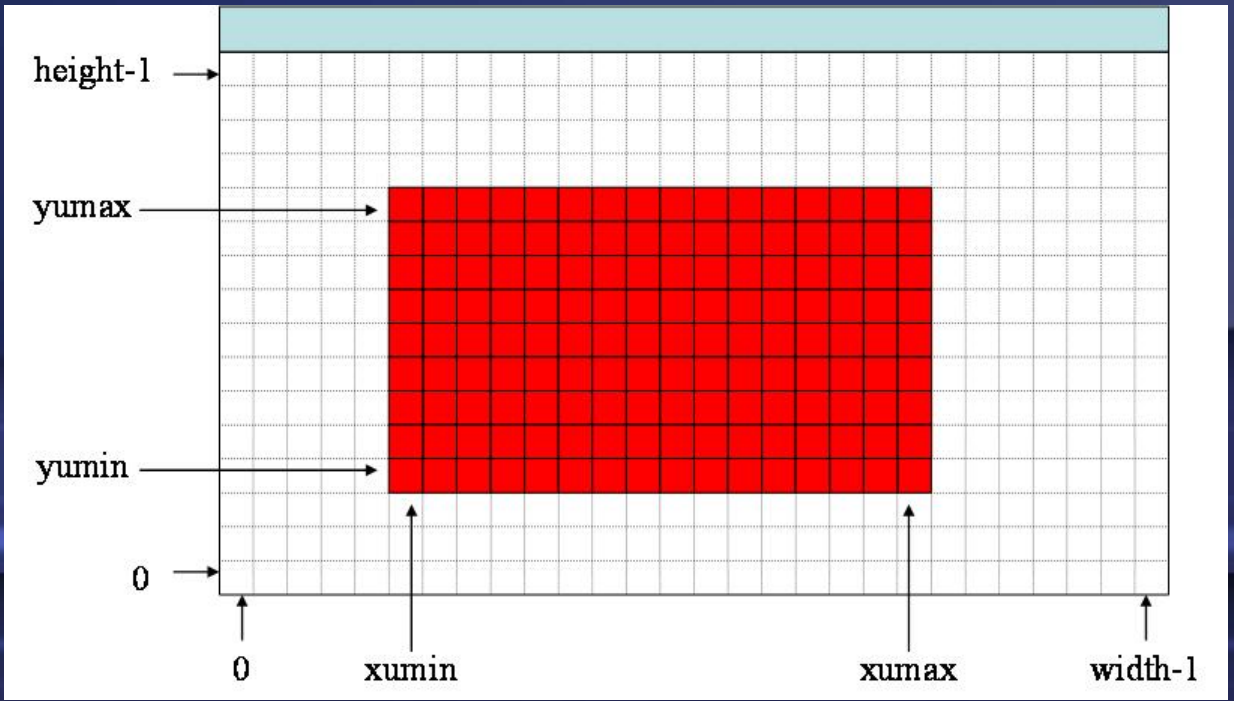
glMatrixMode(GL_PROJECTION);

Παράθυρο παρατήρησης

Παράθυρο Παρατήρησης:

Το τμήμα της επιφάνειας σχεδίασης που χρησιμοποιείται για την απόδοση της αποκομμένης σκηνής.

Τα όρια του παραθύρου παρατήρησης δίνονται σε συντεταγμένες συσκευής



Ορισμός παραθύρου παρατήρησης

Η θέση και έκταση του παραθύρου παρατήρησης στην επιφάνεια σχεδίασης καθορίζεται με την εντολή glViewport

void glViewport(GLint xmin, GLint ymin, GLsizei width, GLsizei height);

xmin, ymin: δηλώνουν το κάτω αριστερό άκρο του παραθύρου παρατήρησης

width, height: πλάτος και ύψος του παραθύρου σε pixels

Ορισμός παραθύρου παρατήρησης

•Η glViewport δηλώνεται πάντοτε μέσα στον κώδικα της συνάρτησης που διαχειρίζεται το γεγονός reshape (στη συνάρτηση-όρισμα της εντολής glutReshapeFunc)

•Εάν δε χρησιμοποιηθεί η glViewport, το προκαθορισμένο παράθυρο παρατήρησης εκτείνεται σε ολόκληρη τη διαθέσιμη επιφάνεια σχεδίασης.

•Χειροκίνητη αλλαγή των διαστάσεων του παραθύρου εφαρμογής χωρίς χρήση της glViewPort αναπροσαρμόζει το παράθυρο παρατήρησης, ούτως ώστε να επεκταθεί ολόκληρη την επιφάνεια σχεδίασης.

Διάκριση παραθύρου αποκοπής και παραθύρου παρατήρησης

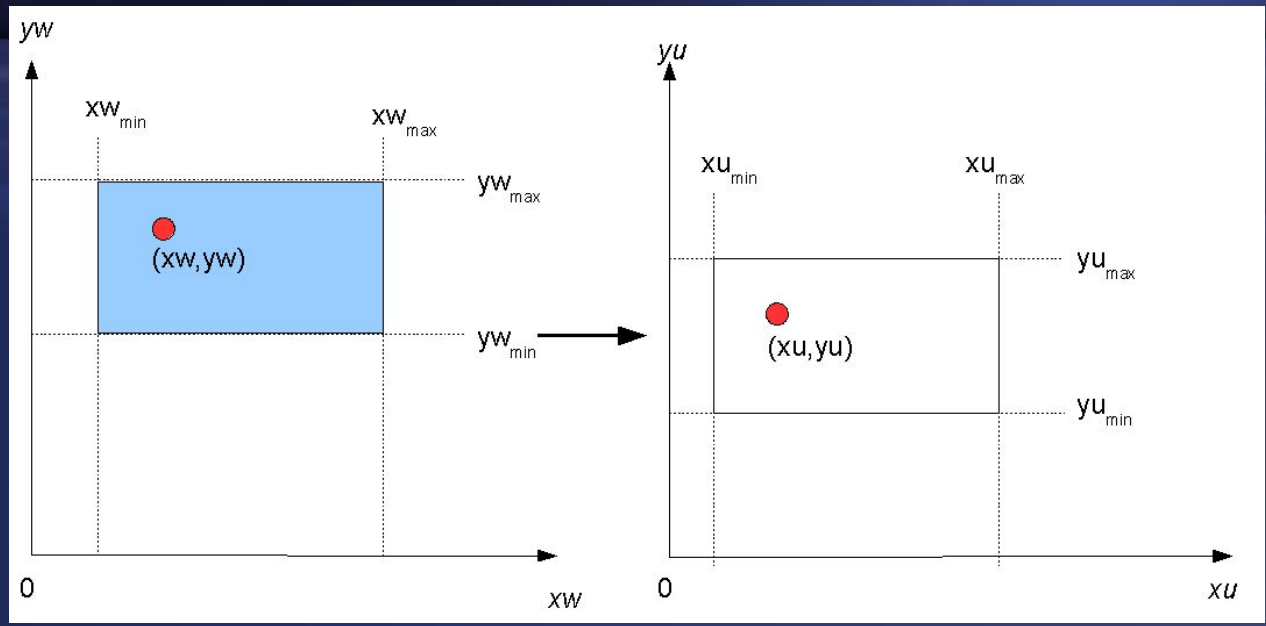
•Το παράθυρο αποκοπής καθορίζει ποιο τμήμα της σκηνής θα σχεδιαστεί,

•Το παράθυρο παρατήρησης καθορίζει πού θα σχεδιαστεί το αποκομμένο τμήμα της σκηνής.

•Μεταβάλλοντας το έυρος του παραθύρου αποκοπής μπορούμε να δημιουργήσουμε εφφέ μεγέθυνσης ή σμίκρυνσης (zoom in – zoom out).

•Αλλαγή αναλογίας πλάτους ύψους -> συμπίεση ή το τέντωμα της σκηνής ως προς μία από τις δύο διαστάσεις (stretching/skewing).

Μετασχηματισμός παρατήρησης (2D)



•Παράθυρο αποκοπής με όρια [xwmin,xwmax] [ywmin,ywmax]

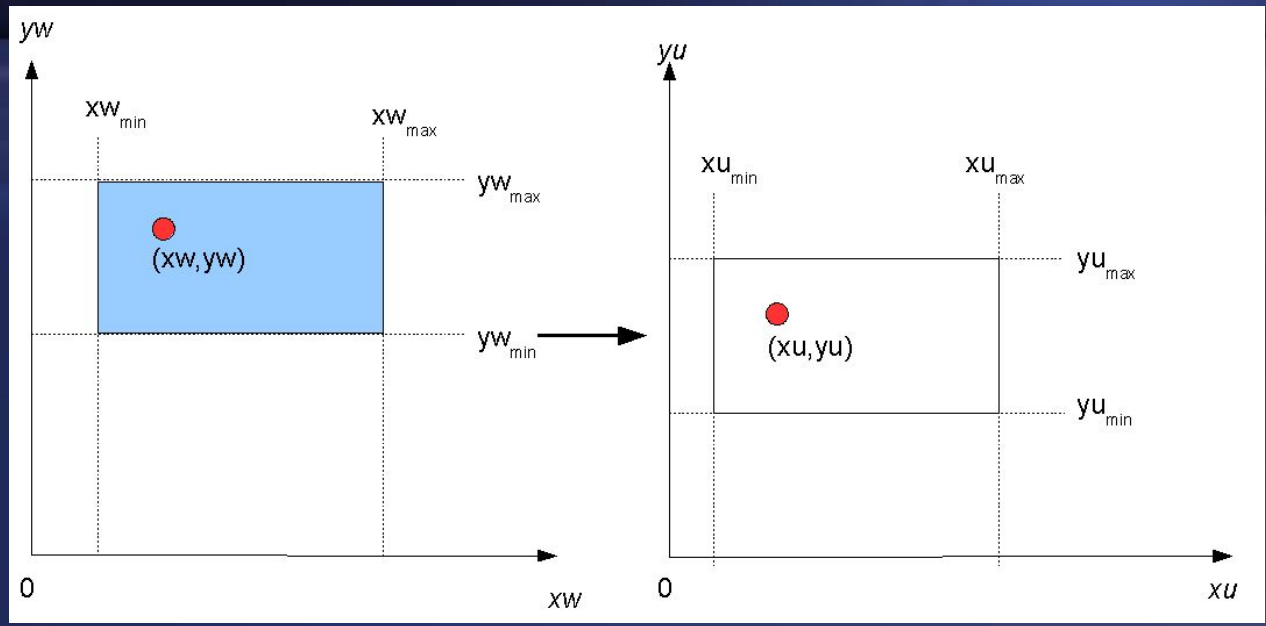
Παράθυρο παρατήρησης με όρια [xumin,xumax], [yumin,yumax]

Κορυφή με συντεταγμένες σκηνής (xw,yw)

Ποιες συντεταγμένες συσκευής (xu,yu) του παραθύρου παρατήρησης αντιστοιχούν στην κορυφή;

Μετασχηματισμός παρατήρησης: μετατροπή συντεταγμένων σκηνής σε συντεταγμένες συσκευής (απαραίτητη διαδικασία για την απεικόνισή της)

Εξισώσεις μετασχηματισμού παρατήρησης (2D)



$$\frac{xw - xw_{\min}}{xw_{\max} - xw_{\min}} = \frac{xu - xu_{\min}}{xu_{\max} - xu_{\min}}$$

$$\frac{yw - yw_{\min}}{yw_{\max} - yw_{\min}} = \frac{yu - yu_{\min}}{yu_{\max} - yu_{\min}}$$

Μητρώο μετασχηματισμού παρατήρησης (2D)

$$xu = s_x \cdot xw + t_x$$
$$yu = s_y \cdot yw + t_y$$

$$s_x = \frac{xu_{\max} - xu_{\min}}{xw_{\max} - xw_{\min}}$$
$$s_y = \frac{yu_{\max} - yu_{\min}}{yw_{\max} - yw_{\min}}$$
$$t_x = \frac{xu_{\min} \cdot xw_{\max} - xu_{\max} \cdot xw_{\min}}{xw_{\max} - xw_{\min}}$$
$$t_y = \frac{yu_{\min} \cdot yw_{\max} - yu_{\max} \cdot yw_{\min}}{yw_{\max} - yw_{\min}}$$

Μετασχηματισμός παρατήρησης σε μορφή μητρώου

$$\begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = M_{w,v} \cdot \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix}$$

$$M_{w,v} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Μετασχηματισμός κανονικοποίησης

πριν το μετασχηματισμό παρατήρησης Η μηχανή της OpenGL κανονικοποιεί τις συντεταγμένες του παραθύρου αποκοπής στο εύρος τιμών στο εύρος τιμών [-1,1] (κανονικοποιημένο τετράγωνο)

$[xw_{min}, xw_{max}] \longrightarrow [-1,1]$

$[yw_{min}, yw_{max}] \longrightarrow [-1,1]$

Μητρώο κανονικοποίησης: προκύπτει αν θεωρήσουμε xumin=-1 και xumax=1

$$M_{w,ns} = \begin{bmatrix} \frac{2}{x_{w\max} - x_{w\min}} & 0 & -\frac{x_{w\min} + x_{w\max}}{x_{w\max} - x_{w\min}} \\ 0 & \frac{2}{y_{w\max} - y_{w\min}} & -\frac{y_{w\min} + y_{w\max}}{y_{w\max} - y_{w\min}} \\ 0 & 0 & 1 \end{bmatrix}$$

Μετασχηματισμός κανονικοποιημένων συντεταγμένων

•Μετατροπή κανονικοποιημένων συντεταγμένων σε συντεταγμένες συσκευής

•Το μητρώο μετασχηματισμού κανονικοποιημένων συντεταγμένων: προκύπτει αν θεωρήσουμε xwmin=-1 και xwmax=1

$$M_{ns,v} = \begin{bmatrix} \frac{x_{u\max} - x_{u\min}}{2} & 0 & \frac{x_{u\min} + x_{u\max}}{2} \\ 0 & \frac{y_{u\max} - y_{u\min}}{2} & \frac{y_{u\min} + y_{u\max}}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

Ρυθμίσεις παραθύρων αποκοπής και παρατήρησης

Σχέση διαστάσεων των παραθύρων αποκοπής και παρατήρησης:
σημαντική όταν ο προγραμματιστής θέλει να αποδώσει τη σκηνή χωρίς ανεπιθύμητες παραμορφώσεις.

•Ο προγραμματιστής χρειάζεται να επιλέξει προσεκτικά προσεκτικά την αναλογία πλάτους-ύψους (aspect ratio) του παραθύρου αποκοπής και του παραθύρου παρατήρησης.

Πχ Σχεδίαση κύκλου
Η αναλογία πλάτους-ύψους και των δύο παραθύρων πρέπει να είναι η ίδια αλλιώς θα σχεδιαστεί έλλειψη.

Προσοχή στις συμβάσεις των συντεταγμένων!!

Στην περίπτωση διαχείρισης γεγονότων:
συντεταγμένες που λαμβάνουμε από τα ορίσματα των συναρτήσεων κλήσης ΔΕΝ ειναι συντεταγμένες συσκευής (Η αρχή των αξόνων στο άνω αριστερή γωνία ενώ οι συντεταγμένες συσκευής θεωρούμε ως αρχή των αξόνων την κάτω ατιστερή γωνία)

Ένα σημειο που περιγράφεται σε ένα διαχειριστή γεγονότος με τις συντεταγμένες xe,ye θα περιγράφεται ως προς το σύστημα συντεταγμένων συσκευής με τις τιμές xu=xe και yu=(height-1)-ye

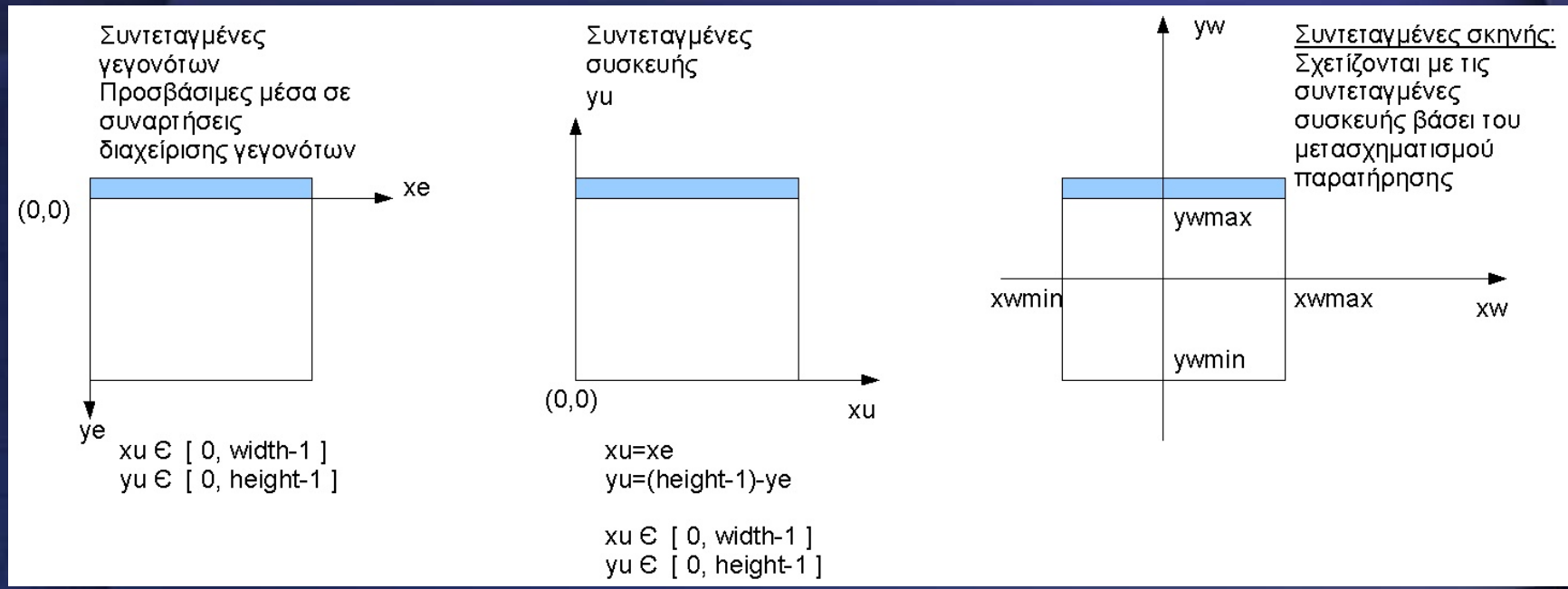
Χειροκίνητη επιλογή σημείων για σχεδίαση

Εάν θέλουμε, κάνοντας κλικ σε ένα pixel, να ορίσουμε ένα σημείο στη θέση του το σημείοπρέπει να εκφράστεί στην εντολή glVertex ως προς το σύστημα συντεταγμένων σκηνής.

Αντιστροφή μετασχηματισμού παρατήρησης:
έυρεση των συντεταγμένων σκηνής xw,yw συναρτήσει των συντεταγμένων συσκευής xu,yu)

Αλληλουχία μετασχηματισμών:
Συντεταγμένες διαχειριτή ⟶ Συντεταγμένες συσκευής
Συντεταγμένες συσκευής ⟶ Συντεταγμένες σκηνής

Απεικόνιση συστημάτων συντεταγμένων!!



Τέλος ενότητας!