

Γνωριμία με την OpenGL

Στην πρώτη εργαστηριακή άσκηση θα γνωρίσουμε το περιβάλλον ανάπτυξης που θα χρησιμοποιήσουμε για την εισαγωγή στην OpenGL και θα μελετήσουμε τον τρόπο λειτουργίας ενός απλού προγράμματος που χρησιμοποιεί την OpenGL και το GLUT για να ζωγραφίσει ένα αντικείμενο στην οθόνη.

Περιγραφή εφαρμογής

Όλο το πρόγραμμα βρίσκεται μέσα στο αρχείο **main.cpp**. Ξεκινώντας με την συνάρτηση `main` που είναι και η είσοδος της εφαρμογής πρέπει να αρχικοποιήσουμε το `Glut` καλώντας τις συναρτήσεις

```
glutInit(&argc, argv);  
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA);
```

Η `glutInitDisplayMode()` δέχεται μια πληθώρα επιλογών για την αρχικοποίηση της OpenGL που είναι πέρα από τους σκοπούς του πρώτου εργαστηρίου να εξετάσουμε. Στην συγκεκριμένη περίπτωση απλά επιλέγουμε το μοντέλο χρώματος που θα χρησιμοποιήσουμε (Red – Green – Blue και διαφάνεια) και ότι θα χρησιμοποιήσουμε διπλό `buffer` για την απεικόνιση.

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 ανταλλάσσουμε τους ρόλους των `buffer` και συνεχίζουμε. Έτσι η οθόνη δείχνει πάντα μια ολοκληρωμένη εικόνα της σκηνής.

Στην συνέχεια δημιουργούμε το παράθυρο που θα απεικονίσουμε την τελική εικόνα με το σετ εντολών:

```
glutInitWindowSize(800, 600);  
glutInitWindowPosition(150,150);  
glutCreateWindow("Computer Graphics - Lab 1");
```

Το παράθυρο θα έχει μέγεθος 800×600 pixels, θα τοποθετηθεί 150,150 pixels από την πάνω αριστερή γωνία της οθόνης και το τίτλο που δηλώσαμε.

Το παρακάτω σετ εντολών μας δίνουν μια ιδέα για το πώς δουλεύει ουσιαστικά το GLUT:

```
glutDisplayFunc(renderScene);  
glutKeyboardFunc(keyPressFunc);  
glutIdleFunc(renderScene);
```

Οι εντολές αυτές καθορίζουν τι θα συμβεί σε περίπτωση όταν λάβει χώρα ένα γεγονός που μας ενδιαφέρει και που θέλουμε να επεξεργαστούμε. Τέτοια συμβάντα είναι για

παράδειγμα το πάτημα ενός πλήκτρου, η κίνηση του ποντικιού, η εντολή να επανασχεδιάσουμε το παράθυρο λόγω μετακίνησης του.

Το GLUT μας δίνει την δυνατότητα να ορίσουμε τι θέλουμε να γίνεται όταν λαμβάνει χώρα ένα τέτοιο γεγονός μέσω μια σειράς εντολών της μορφής:

```
Glut_OnomaEntolis_Func(συνάρτηση που θα καλείται);
```

Στην εφαρμογή ενδιαφερόμαστε για 3 γεγονότα, όταν πατηθεί κάποιο πλήκτρο, όταν δοθεί εντολή να επανασχεδιαστεί το παράθυρο και όταν το GLUT δεν κάνει τίποτα (είναι δηλαδή idle). Αυτή η στιγμή είναι κατάλληλη για να σχεδιάσουμε το αντικείμενο στην οθόνη.

Το GLUT μας επιτρέπει να ορίσουμε συναρτήσεις που θα εξυπηρετούν διάφορα συμβάντα. Αυτός είναι και ο μόνος τρόπος να προσθέσουμε λειτουργικότητα σε μια GLUT εφαρμογή.

Στην συνέχεια καλούμε την συνάρτηση

```
init();
```

που στην συνέχεια θα καλέσει εντολές της OpenGL και θα κάνει αρχικοποιήσεις που χρειάζεται να γίνουν μόνο μια φορά. Από αυτές τις εντολές προς το παρόν μας ενδιαφέρουν μόνο οι

```
glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  
glColor3f(1.0f, 1.0f, 0.0f);
```

Η πρώτη καθορίζει το χρώμα στο οποίο θα καθαρίζουμε (clear) το παράθυρο (όταν δοθεί η εντολή). Το χρώμα καθαρισμού δεν είναι ανάγκη να είναι μαύρο, μπορεί να είναι οτιδήποτε. Η δεύτερη καθορίζει το χρώμα με το οποίο θα «βάψουμε» το αντικείμενο όταν το θα το ζωγραφίσουμε την οθόνη.

Το ότι θέτουμε το χρώμα καθαρισμού και χρωματισμού του αντικειμένου μία φορά κατά την αρχικοποίηση της εφαρμογής είναι ένα καλό παράδειγμα λειτουργίας της OpenGL ως μηχανή καταστάσεων (state machine) – μια τιμή που θέτουμε διατηρείται μέχρι να την ξαναλλάξουμε.

Τέλος καλούμε την συνάρτηση που θα αρχίσει το κυρίως loop της εφαρμογής.

```
glutMainLoop();
```

Από αυτό το loop η εφαρμογή θα βγει μόνο όταν κλείσουμε το παράθυρο ή την εφαρμογή. Στο μεταξύ το GLUT θα τρέχει αυτό το loop ακούγοντας για διάφορα γεγονότα (όπως είσοδο από πληκτρολόγιο, κίνηση ποντικιού, κίνηση joystick) και ανάλογα με το αν έχουμε δηλώσει κάποια συνάρτηση για κάποιο συμβάν (με την χρήση της glutXXXXFunc()) την καλεί για να το εξυπηρετήσει.

Μια τέτοια συνάρτηση είναι και η

```
void keyPressFunc(unsigned  
char key, int x, int y)
```

Το GLUT θα καλέσει την συνάρτηση αυτή όταν πατηθεί κάποιο πλήκτρο. Στην συνάρτηση ελέγχουμε αν έχει πατηθεί το πλήκτρο ESC και αν ναι τερματίζουμε την εφαρμογή.

Η συνάρτηση που θα ζωγραφίσει το αντικείμενο στην οθόνη είναι η

```
renderScene(void);
```

Η συνάρτηση αυτή αρχικά καθαρίζει το τρέχον buffer στο χρώμα το οποίο θέσαμε στην `init()` με την εντολή `glClearColor()`.

Στην συνέχεια καλεί μια εντολή GLUT για να ζωγραφίσει το αντικείμενο στον buffer.

```
glutSolidTeapot(15.0);
```

με μέγεθος 15. Η εντολή αυτή καλεί πολλές OpenGL εντολές για να δημιουργήσει και να απεικονίσει το μοντέλο της τσαγιέρας. Το GLUT προσφέρει και άλλες εντολές που δημιουργούν μοντέλα όπως `glutSolidCube()`, `glutSolidCone()`, `glutSolidTetrahedron()`.

Στο τέλος καλεί την εντολή

```
glutSwapBuffers();
```

η οποία θα αντικαθιστά τον buffer που χρησιμοποιεί η οθόνη με αυτόν που περιέχει το αντικείμενο που μόλις ζωγραφίσαμε.

Για να δείτε το double buffering σε λειτουργία δοκιμάστε να σβήσετε αυτή την εντολή και να αντικαταστήσετε την παράμετρο GLUT_DOUBLE με GLUT_SINGLE στην εντολή `glutInitDisplayMode()`.

Η έξοδος του προγράμματος θα πρέπει να είναι η παρακάτω, μια κίτρινη τσαγιέρα:

