User's CallBacks, REDISPLAY and ANIMATION

Callback functions : Συναρτήσεις που καλούνται και εκτελούνται κατ΄ εκτίμηση του λειτουργικού συστήματος (flags : on/off) ανάλογα με γεγονότα που συνδέονται με το παράθυρο της εφαρμογής (window events) ή την αλληλεπίδραση με τις μονάδες εισόδου -ποντίκι ή πληκτρολόγιο – (keyboard or mouse events)

`glutMainLoop();`

```
glutDisplayFunc(...);
glutReshapeFunc( ...)
glutKeyboardFunc(...)
glutMouseFunc(...)
.....
void glutIdleFunc(void (*func)(void));
```

- `void glutIdleFunc(void (*func)(void));`

Δηλώνεται μια συνάρτηση του χρήστη ως "callback". Η συνάρτηση αυτή δεν έχει ορίσματα και καλείται-εκτελείται σε κάθε CPU κύκλο και όταν δεν υπάρχουν "events" προς επεξεργασία.

(βλ code40Idle.cpp)

* Βελτιστοποίηση με τη χρήση Διπλού Buffer

Η επεξεργασία των γραφικών γίνεται σε δύο buffers τον "front" και τον "back" οι οποίοι λειτουργούν ανεξάρτητα. πχ ο front buffer είναι αυτός που παρουσιάζεται στην οθόνη και ο back buffer αυτός που σχεδιάζεται στο παρασκήνιο. Σε κάθε κάλεσμα της Display μπορούμε να εναλλάσσουμε τους buffers έτσι ώστε να επιτυγχάνεται μια γρήγορη ομαλή μετάβαση μεταξύ των frames.

`glutInitDisplayMode(GLUT_DOUBLE| ....)` : Δήλωση χρήσης διπλού buffer
`glutSwapBuffers()` : εναλλαγή των buffers

(βλ code40Idle2.cpp)

Σημείωση. Αναδρομικές (recursive) κλήσεις των "callback" συναρτήσεων είναι δυνατές αλλά το κόστος σε μνήμη του "function-call overhead" είναι σημαντικό. Η "αναδρομικότητα" (recurrence) πρέπει να συνοδεύεται <u>πάντα</u> από κατάλληλη έξοδο.
(βλ code40recur.cpp)

Μια συνάρτηση "Idle callback" , πχ `myIdle()`, μπορεί να χρησιμοποιηθεί για τη δημιουργία «κινούμενων γραφικών» αν περικλείει τα παρακάτω βήματα

1. Αλλαγή των παραμέτρων σχεδίασης (συντεταγμένες αντικειμένων-σκηνικού ή των πινάκων μετασχηματισμού CMT)
2. Επανάκληση της callback συνάρτησης DisplayFunc με την εντολή `glutPostRedisplay();`
3. Κατάλληλος έλεγχος καθυστέρησης (delay procedure)

Χρήση Χρονομέτρου (TIMER)

void `glutTimerFunc` (unsigned int `n` , void `(*func)`(int val), `val`);

`n` : Χρόνος καθυστέρησης (milliseconds) πριν το κάλεσμα της συνάρτησης χρονομέτρου `func( )`.
`func` : Η συνάρτηση που εκτελείτε από την timer callback function.
`val` : Ακέραια τιμή που μπορεί να περαστεί ως όρισμα στη `func`.

```
glutTimerFunc( n , myFunction, val)
```

- Η εκτέλεση της `glutTimerFunc(...)` καθυστερεί τον «βρόγχο γεγονότων» (event loop) για `n` msec και στη συνέχεια, με την πρώτη ευκαιρία του επεξεργαστή, καλείται (ως callback) η συνάρτηση του χρήστη `myFunction`.

- Η `myFunction` μπορεί να δεχτεί έναν ακέραιο ως όρισμα (`val`) το οποίο το δηλώνουμε ως τρίτο όρισμα στην `glutTimerFunc` . Ο ακέραιος αυτός συμπεριφέρεται ως «static» και χρησιμοποιείται για τον έλεγχο της διαδικασίας.

- Η εκτέλεση της `glutTimerFunc` καθαρίζει τη μνήμη (stack memory) της συνάρτησης `myFunction` . Έτσι η `myFunction` μπορεί να μπεί σε αναδρομική διαδικασία εκτέλεσης με κάλεσμα από την `glutTimerFunc` χωρίς πρόβλημα στη διαδικασία "function-call overhead".

Η `glutTimerFunc` μπορεί να χρησιμοποιηθεί για animation αν η σχετιζόμενη συνάρτηση `myFunction` περικλείει τα παρακάτω βήματα

1. Αλλαγή των παραμέτρων σχεδίασης (συντεταγμένες αντικειμένων-σκηνικού ή των πινάκων μετασχηματισμού CMT)
2. Επανάκληση της callback συνάρτησης DisplayFunc με την εντολή `glutPostRedisplay();`
3. Αναδρομικό κάλεσμα της ίδιας (της myFunction) με την χρήση της `glutTimerFunc.`

(βλ code42.cpp)