

## ΑΝΤΙΚΕΙΜΕΝΑ ΩΣ ΛΙΣΤΕΣ\*

### A. ΑΠΛΑ ΑΝΤΙΚΕΙΜΕΝΑ

#### Κατασκευή

```
glNewList (ID_number, mode);  
           ...κώδικας αντικειμένου – vertices  
glEndList ();
```

int **ID\_number**: Ακέραιος αριθμός για την ταυτοποίηση του αντικειμένου  
const **mode**: GL\_COMPILE or GL\_COMPILE\_AND\_EXECUTE

#### Κλήση

```
glCallList(ID_number);
```

#### Δημιουργία νέου ID number

```
new_index = glGenList ();
```

(\*έλεγχος: glIsList (number) -- GL\_TRUE or GL\_FALSE)

---

### A. ΣΥΝΘΕΤΑ ΑΝΤΙΚΕΙΜΕΝΑ

Δημιουργούμε σύνθετα αντικείμενα ως πίνακες λιστών απλών αντικειμένων

#### Κατασκευή

Δέσμευση ενός συνόλου από αριθμούς λιστών

```
new_index = glGenLists (N);
```

\* Αν k ήταν ο τελευταίος δεσμευμένος ID\_Number, τότε η παραπάνω εντολή δεσμεύει επιπλέον N αριθμούς

Για τον κάθε νέο αριθμό, αντιστοιχούμε ένα αντικείμενο :

```
glNewList (ID_number, mode);  
           ...κώδικας αντικειμένου – vertices  
glEndList ();
```

---

\* (βλ project GList)

όπου  $new\_index \leq ID\_number < new\_index + N$

### κλήση ενός συνόλου λιστών

Έστω ότι διαθέτουμε ένα σύνολο  $N$  με  $N$  λίστες  $(1, 2, 3, \dots, N)$

Έστω ότι θέλουμε να καλέσουμε προς σχεδίαση τα  $k$  αντικείμενα με αριθμούς

$$0 < n_i < N, \quad i=1, 2, \dots, k$$

τα οποία τα δηλώνουμε σε ένα πίνακα (τύπου Byte ή Integer)

$$L = \{n_1, n_2, \dots, n_k\}$$

>>κλήση με

```
glCallLists(k, type, L);
```

$k$ : αριθμός λιστών για σχεδίαση,  $L$  πίνακας τύπου `type` που περιέχει τα `ID_numbers` των λιστών που θα σχεδιαστούν.

\*Μπορούμε να θέσουμε ένα `offset` στο σύνολο  $N$  με την εντολή

```
glListBase(OffsetValue);
```

Μετά την παραπάνω εντολή η `glCallLists(k, type, L)` θα σχεδιάσει τα αντικείμενα  $n_i + offsetValue$ .

Αν δεν δηλωθεί `OffsetValue` τότε το `offset` θεωρείται ίσο με μηδέν.

\*διαγραφή  $n$  λιστών του συνόλου  $N$  από τη θέση  $m$  (`+offsetValue`)

```
glDeleteLists(m, n);
```