

Εργασία για το μάθημα
“Εισαγωγή στο Δομημένο Προγραμματισμό”

Θέμα:

Animation ευθύγραμμης ομαλής κίνησης με ελαστικές κρούσεις.

Ομάδα:

Ονοματεπώνυμο1	Εξάμηνο1 AM1
Ονοματεπώνυμο2	Εξάμηνο2 AM2
Ονοματεπώνυμο3	Εξάμηνο3 AM3

Ημερομηνία Παράδοσης:

1. Εισαγωγή

Στην παρούσα εργασία ζητείται να αναπτύξουμε ένα πρόγραμμα το οποίο θα προσομοιώνει την ευθύγραμμη ομαλή κίνηση ενός σώματος το οποίο όμως θα υφίσταται ελαστική κρούση στα τοιχώματα ενός παραλληλόγραμμου “κουτιού” όπου περικλείεται. Συγκεκριμένα, το πρόγραμμα θα παράγει διαδοχικές εικόνες δύο διαστάσεων, σε κάθε μία από τις οποίες θα απεικονίζονται τα σταθερά τοιχώματα και το σωματίδιο σε διαφορετική θέση κάθε φορά. Οι εικόνες αυτές που θα είναι τα στιγμιότυπα της κίνησης, πρέπει να εναλλάσσονται με ταχύτητα αρκετή ώστε στο θεατή να δίνεται η ψευδαίσθηση της κίνησης.

Οι εικόνες θα παράγονται με χρήση απλών χαρακτήρων `ascii`, δηλαδή δε ζητείται χρήση κάποιας βιβλιοθήκης γραφικών με την οποία να ελέγχονται τα `pixels` της οθόνης. Το κινούμενο σώμα μπορεί να παριστάνεται με ένα μεμονωμένο χαρακτήρα όπως `'o'` ή `'*'` που θα αλλάζει θέση στην οθόνη καθώς τα στιγμιότυπα θα διαδέχονται το ένα το άλλο.

Ζητείται επίσης να δίνεται στο χρήστη η δυνατότητα να καθορίζει λεπτομέρειες όπως το μέγεθος και οι αναλογίες των πλευρών του πλαισίου, η ταχύτητα εκτέλεσης της προσομοίωσης, η διεύθυνση της αρχικής ταχύτητας, η αρχική θέση του σώματος και η συνολική διάρκεια της κίνησης. Αυτές μπορεί να παρέχονται ως ορίσματα από τη γραμμή εντολών, με διάλογο με το χρήστη που θα εισάγει πληροφορίες από το πληκτρολόγιο ή να διαβάζονται από κάποιο αρχείο. Στην παρούσα υλοποίηση έχουν επιλεγεί οι δύο πρώτοι τρόποι.

Όπως γνωρίζουμε από τη στοιχειώδη φυσική, ελαστική ανάκλαση σημαίνει ότι το συγκρουόμενο σώμα που πέφτει σε ακλόνητο τοίχωμα δεν χάνει συνολικά ταχύτητα αλλά αντιστρέφεται η συνιστώσα της ταχύτητας που είναι κάθετη στο τοίχωμα, ενώ η άλλη παραμένει ανεπηρέαστη. Επίσης, γνωρίζουμε ότι η συνολική κίνηση του σώματος μπορεί να θεωρηθεί ως ταυτόχρονη μετατόπιση σε δύο άξονες κάθετους μεταξύ τους. Αυτά σημαίνουν ότι αν οι πλευρές του πλαισίου είναι παράλληλες σε δύο άξονες αναφοράς, ας πούμε `Ox` και `Oy` (με `O` συμβολίζουμε την αρχή των αξόνων), τότε σε κάθε έναν από αυτούς θα εκτελείται περιοδική κίνηση. Η περίοδος της κάθε κίνησης είναι ανάλογη με τη διάσταση (μήκος) του κουτιού στον αντίστοιχο άξονα και αντιστρόφως ανάλογη της ταχύτητάς του παράλληλα με αυτόν (δηλαδή της αντίστοιχης συνιστώσας της συνολικής ταχύτητας). Το συμπέρασμα είναι ότι αρκεί να βρούμε πώς θα προσομοιώσουμε την ευθύγραμμη περιοδική κίνηση και να εφαρμόσουμε την ίδια τεχνική και για τους δύο άξονες ώστε να πετύχουμε την κίνηση στο επίπεδο.

2. Ανάλυση

Για να πετύχουμε την ψευδαίσθηση της κίνησης με εναλλαγή εικόνων είναι προφανές ότι θα πρέπει η οθόνη να καθαρίζει περιοδικά ώστε να εμφανίζεται κάθε φορά η ανανεωμένη εικόνα με τη νέα θέση του σώματος. Αυτό μπορούμε να το πετύχουμε έχοντας μέσα στο πρόγραμμα στην κατάλληλη θέση, μια εντολή όπως

```
system("clear");
```

για Linux/UNIX ή ανάλογη εντολή για άλλα λειτουργικά συστήματα.

Για να ρυθμίζουμε πόσος χρόνος θα περνά από στιγμιότυπο σε στιγμιότυπο και επομένως την ταχύτητα της προσομοίωσης μπορούμε να εισάγουμε ένα βρόχο επανάληψης όπως `for` ή `while` που θα έχει αποκλειστικό σκοπό να εισάγει μια χρονική καθυστέρηση.

Τα τοιχώματα του πλαισίου είναι σταθερά, οπότε αρκεί να καθοριστεί από την αρχή πώς θα σχεδιάζονται. Το μόνο που θα αλλάζει από εικόνα σε εικόνα θα είναι η θέση του κινούμενου σώματος. Για να σχεδιάσουμε τις οριζόντιες γραμμές του πλαισίου θα εμφανίσουμε το χαρακτήρα του αρνητικού προσήμου `'-'` τον κατάλληλο αριθμό φορές. Παρόμοια, για να σχεδιάσουμε τις κατακόρυφες πλευρές, θα χρησιμοποιήσουμε το χαρακτήρα `'|'` δύο φορές και

ενδιάμεσα, τον κατάλληλο αριθμό κενών διαστημάτων. Μόνο ένα κενό διάστημα, διαφορετικό σε κάθε στιγμιότυπο, θα αντικαθίσταται από τον χαρακτήρα που θα παριστάνει το κινητό σώμα, έστω '*'.

Τα παραπάνω μπορούμε να τα πετύχουμε αν ορίσουμε έναν πίνακα κατάλληλων διαστάσεων όπου θα αποθηκεύουμε τους παραπάνω χαρακτήρες, '-', '|', κενά διαστήματα και '*', ώστε με την εκτύπωση των γραμμών του να εμφανίζεται η επιθυμητή εικόνα. Τότε, το πρόβλημα προσομοίωσης της κίνησης ανάγεται στο να βρίσκουμε κάθε φορά τους κατάλληλους δείκτες i και j του στοιχείου του πίνακα που θα περιέχει τον χαρακτήρα αναπαράστασης του κινητού. Τέλος, για αισθητικούς λόγους, το πλαίσιο θα πρέπει να εμφανίζεται ελαφρά μετατοπισμένο οριζοντίως και καθέτως σε σχέση με την πάνω αριστερή γωνία της οθόνης, πράγμα που γίνεται εύκολα με μερικούς χαρακτήρες αλλαγής γραμμής και μερικά επιπλέον κενά διαστήματα στην αρχή κάθε γραμμής του πίνακα των στιγμιότυπων.

Εφόσον δε θα δουλέψουμε με γραφικά αλλά με απλούς `ascii` χαρακτήρες, η μονάδα μέτρησης για το οριζόντιο πλάτος θα είναι το διάστημα (όσο ένας χαρακτήρας) και για το κατακόρυφο ύψος η κάθετη μετατόπιση που έχουμε όταν εισάγουμε μια αλλαγή γραμμής. Επειδή αυτά τα δύο δεν έχουν ακριβώς το ίδιο μήκος, αν θέλουμε π.χ. να φτιάξουμε ένα τετράγωνο πλαίσιο θα πρέπει να πειραματιστούμε λίγο με διάφορες τιμές για το πλάτος και το ύψος μέχρι να το πετύχουμε.

Ας υποθέσουμε ότι θέλουμε να κάνουμε ένα πλαίσιο 40 X 30 (πλάτος X ύψος, στις μονάδες που ορίσαμε). Έστω ότι το κινητό ξεκινά από την πάνω αριστερή γωνία και σε κάθε αλλαγή πλαισίου μετατοπίζεται κατά μία θέση οριζόντια δεξιά και μία θέση προς τα κάτω. Τότε, τότε μετά από 30 αλλαγές εικόνας θα πρέπει να αρχίσει να ανεβαίνει αφού θα έχει φτάσει στο κάτω μέρος του πλαισίου ενώ μετά από 40 εικόνες θα πρέπει να αρχίσει να κινείται προς τα αριστερά αντί για δεξιά.

Αν από την αρχή της κίνησης έχουν αλλάξει N πλαίσια ή στιγμιότυπα, ένας τρόπος για να ξέρουμε αν πρέπει να πάει προς τα πάνω ή προς τα κάτω, είναι ανάλογα με το αν η τιμή της ακέραιας διαίρεσης του N διά 30 είναι άρτιος ή περιττός. Σύμφωνα με το παράδειγμά μας, στην αρχή είναι 0 (άρτιος) άρα πάει προς τα κάτω. Μόλις το N περάσει το 30, το $N/30$ γίνεται 1 (περιττός) και η κίνηση αντιστρέφεται. Εντελώς ανάλογο κριτήριο μπορούμε να εφαρμόσουμε για την κίνηση δεξιά-αριστερά. Αν το κινητό ξεκινά από άλλο σημείο αντί για την πάνω αριστερή γωνία, τότε το κριτήριο τροποποιείται ως εξής: $(N + \text{κατακόρυφη αρχική συντεταγμένη})/30$ για την κατακόρυφη κίνηση, $(N + \text{οριζόντια αρχική συντεταγμένη})/40$ για την οριζόντια κίνηση.

Γενικεύοντας, το κριτήριο για την αντιστροφή της κίνησης θα είναι κατά πόσον είναι άρτιες ή περιττές οι παρακάτω ποσότητες:

$$(N + \text{ΚΑΣ}) / \text{Ύψος} \quad (\text{για κατακόρυφη κίνηση}),$$

$$(N + \text{ΟΑΣ}) / \text{Πλάτος} \quad (\text{για οριζόντια κίνηση}),$$

όπου ΟΑΣ = οριζόντια αρχική συντεταγμένη και ΚΑΣ = κατακόρυφη αρχική συντεταγμένη του κινητού σώματος.

Αλλά πού ακριβώς πρέπει να τοποθετηθεί κάθε φορά το κινητό; Ας πάρουμε την οριζόντια κίνηση. Η ποσότητα $N + \text{ΚΑΣ}$ δε μπορεί παρά να είναι άθροισμα ακέραιου πολλαπλάσιου του Πλάτους συν κάποια επιπλέον μετατόπιση X . Αυτό το X θέλουμε να βρούμε και προφανώς θα είναι το υπόλοιπο της διαίρεσης του $N + \text{ΚΑΣ}$ με το Πλάτος. Αν το κινητό πηγαίνει προς τα δεξιά, τότε η θέση του θα είναι

$$X = (N + \text{ΚΑΣ}) \% \text{Πλάτος}$$

(με $\%$ συμβολίζουμε το υπόλοιπο της διαίρεσης με το Πλάτος). Αν πηγαίνει προς τα αριστερά,

τότε η παραπάνω μετατόπιση θα είναι σε σχέση με τη δεξιά πλευρά. Εφόσον μετράμε τις αποστάσεις από την αριστερή πλευρά, θα πρέπει να πάρουμε

$$X = \text{Πλάτος} - (N + \text{ΚΑΣ}) \% \text{Πλάτος}.$$

Εντελώς ανάλογο θα είναι το σκεπτικό μας και για την κατακόρυφη μετακίνηση.

Τέλος, οι περισσότερες από τις παραμέτρους που ζητείται να μπορεί να εισάγει ο χρήστης είναι εύκολο να δοθούν είτε από τη γραμμή εντολών είτε με κάποιο συνδυασμό ερωτήσεων από το χρήστη και ανάγνωσης δεδομένων. Πιο συγκεκριμένα, για την ταχύτητα της προσομοίωσης έχουμε να κάνουμε την εξής παρατήρηση: το πόσο γρήγορα εκτελείται η προσομοίωση εξαρτάται κυρίως από το βρόχο της χρονοκαθυστέρησης που αναφέραμε αρχικά. Επομένως, μπορεί να οριστεί μία μεταβλητή που αυξομειώνει τα όρια στα οποία μεταβάλλεται κάποιος μετρητής του βρόχου.

Η αρχική θέση δίνεται ως ένα ζεύγος συντεταγμένων για την αρχική οριζόντια και κατακόρυφη μετατόπιση. Ο χρήστης πρέπει να ενημερώνεται για τις διαστάσεις του πλαισίου ώστε να δώσει συντεταγμένες εντός των ορίων του, αλλιώς η εφαρμογή των παραπάνω τύπων θα έχει ως αποτέλεσμα το κινητό να διαπερνά τα όρια του κουτιού.

Τέλος, για την κατεύθυνση της ταχύτητας ζητούνται δύο ακέραιοι που εκφράζουν το λόγο της οριζόντιας προς την κατακόρυφη κίνηση. Αν αυτούς τους συμβολίσουμε με dx και dy , τότε οι προηγούμενοι τύποι μπορούν να ξαναγραφούν ως εξής:

$$((dx/dy) * N + \text{ΚΑΣ}) / \Upsilon\psi\omicron\varsigma \text{ (για κατακόρυφη κίνηση)},$$

$$(N + \text{ΟΑΣ}) / \text{Πλάτος} \text{ (για οριζόντια κίνηση)},$$

Αυτό θα έχει ως αποτέλεσμα η κατακόρυφη κίνηση να επιταχύνεται ή επιβραδύνεται σε σχέση με την οριζόντια, ανάλογα με τη σχέση των δύο παραμέτρων, dx και dy .

3. Δομή του προγράμματος

Το πρόγραμμα είναι αρκετά μικρό ώστε να χρησιμοποιήσουμε ένα μόνο αρχείο. Η δομή του, που αντανακλά και τη ροή του ελέγχου, είναι αρκετά απλή: η `main` παίρνει πληροφορίες για τις διαστάσεις του πλαισίου και την ταχύτητα προσομοίωσης από τη γραμμή εντολών, αλλιώς χρησιμοποιεί καθορισμένες τιμές που έχουν οριστεί ως συμβολικές σταθερές. Κατόπιν, καλείται η συνάρτηση `ask_user`, η οποία ζητά από το χρήστη επιπλέον προαιρετικές πληροφορίες (αρχική θέση κινούμενου σώματος, διεύθυνση αρχικής ταχύτητας). Μετά, εκτελείται ένας βρόχος `while` όπου καλούνται διαδοχικά οι συναρτήσεις `draw_line` (για το σχεδιασμό της πάνω οριζόντιας γραμμής του πλαισίου), `draw_point` (για το σχεδιασμό του εσωτερικού του πλαισίου και του κινητού στην κατάλληλη θέση) και ξανά `draw_line` (για το σχεδιασμό της κάτω οριζόντιας γραμμής). Μετά, καλείται η συνάρτηση `time_delay` που καθυστερεί την εκτέλεση ώστε η προσομοίωση να εξελίσσεται με την επιθυμητή ταχύτητα. Αυτή η συνάρτηση προκαλεί και την έξοδο από το βρόχο με κλήση της συνάρτησης βιβλιοθήκης `exit`, μόλις ξεπεραστεί η διάρκεια της προσομοίωσης.

Αναλυτικότερα, διακρίνουμε τα εξής στοιχεία:

3.1 Συμβολικές σταθερές.

Ορισμένες χρήσιμες τιμές παριστάνονται με συμβολικές σταθερές που ορίζονται με την οδηγία `#define` στην αρχή του αρχείου και επομένως είναι καθολικά ορατές από τις συναρτήσεις του προγράμματος. Αυτές οι σταθερές αντιπροσωπεύουν τις προκαθορισμένες (default) τιμές για ορισμένες βασικές παραμέτρους της προσομοίωσης, οι οποίες και ισχύουν αν δε δοθούν από τη γραμμή εντολών. Αναλυτικότερα, οι σταθερές αυτές είναι οι εξής:

DELAY. Είναι η παράμετρος ελέγχου της χρονοκαυστήρησης. Ορίζεται ίση με 25.

OFFSET. Καθορίζει πόσο θα απέχει οριζοντίως το πλαίσιο από το αριστερό άκρο της οθόνης. Ορίζεται ίση με 10.

WIDTH. Η οριζόντια διάσταση του πλαισίου. Έχει τεθεί ίση με 80.

HEIGHT. Η κάθετη διάσταση του πλαισίου, ορισμένη ίση με 30.

3.2 Συνάρτηση main – περιγραφή.

Εδώ ορίζονται μεταβλητές αντίστοιχες με κάποιες από τις προηγούμενες συμβολικές σταθερές, δηλαδή οι delay, width, height, καθώς επίσης και ο μετρητής kount με αρχική τιμή 0, για τη μέτρηση των στιγμιοτύπων της προσομοίωσης, δηλαδή της διάρκειάς της. Ακόμη, ορίζονται οι αρχικές συντεταγμένες x0 και y0 του κινητού και τους δίνεται η τιμή 0 που αντιστοιχεί στην πάνω αριστερή γωνία του πλαισίου (μπορεί να αλλάξει, όπως θα πούμε παρακάτω). Επίσης, ορίζονται οι μεταβλητές dir, diry, των οποίων ο λόγος μας βοηθά να καθορίσουμε την αρχική διεύθυνση της κίνησης, όπως εξηγήσαμε στην ενότητα 2. Τέλος, ορίζεται η μέγιστη διάρκεια max_time, με αρχική τιμή 1000 στιγμιότυπα (μπορεί να αλλάξει, όπως θα πούμε πιο κάτω).

Το πρόγραμμα ξεκινά με τον καθορισμό των delay, width και height με τη βοήθεια του τριαδικού τελεστή για τον έλεγχο ύπαρξης αντίστοιχων ορισμάτων στη γραμμή εντολών. Οι delay, width και height προσδιορίζονται ως το πρώτο, δεύτερο και τρίτο όρισμα της γραμμής εντολών, αντίστοιχα. Αν δεν υπάρχει το αντίστοιχο όρισμα τότε παίρνουν την αντίστοιχη προκαθορισμένη τιμή των συμβολικών σταθερών. Λεπτομέρειες για τον τρόπο παροχής των πληροφοριών από τη γραμμή εντολών δίνονται στις Οδηγίες Χρήσης. Το τμήμα λήψης πληροφοριών από το χρήστη ολοκληρώνεται με την κλήση της συνάρτησης ask_user μέσω της οποίας δίνονται δεδομένα για την αρχική θέση και την αρχική κατεύθυνση της ταχύτητας του κινητού, καθώς και το χρόνο προσομοίωσης (βλ. και Οδηγίες Χρήσης).

Επιλέχτηκε αυτός ο τρόπος, κατά τον οποίο οι πληροφορίες δίνονται από δύο διαφορετικά "κανάλια", με το εξής σκεπτικό: Τα τρία τελευταία δεδομένα (θέση, κατεύθυνση, διάρκεια) είναι αυτά με τα οποία ο χρήστης αναμένεται να "πειραματιστεί" περισσότερο, άρα είναι καλό να υπάρχει ένας συγκεκριμένος διάλογος με το πρόγραμμα. Αντίθετα, η χρονοκαυστήρηση και οι διαστάσεις του κουτιού αφορούν προτιμήσεις του τρόπου εκτέλεσης (θα μπορούσαν επίσης να περιλαμβάνουν χρώματα, σύμβολο κινητού κλπ) που δεν αλλάζουν πολύ συχνά και συνηθίζεται, σε ανάλογες περιπτώσεις, να δίνονται ως επιλογές κατά την κλήση του προγράμματος. Από αυτές, η ταχύτητα εκτέλεσης είναι αυτή που είναι πιθανό να αλλάζει πιο συχνά και έτσι δίνεται ως η πρώτη επιλογή από τη γραμμή εντολών.

Μετά από τη λήψη πληροφοριών αρχίζει ο βρόχος while με συνθήκη 1 (πάντα αληθής) και πρώτη εντολή στο σώμα του, τον καθαρισμό της οθόνης (system("clear")). Τυπώνεται ένα μήνυμα που δείχνει την παρέλευση του χρόνου και καλούνται οι συναρτήσεις draw_line, draw_point και πάλι draw_line για το σχεδιασμό του πλαισίου με την ανανεωμένη θέση του κινητού. Καθαρίζεται η περιοχή προσωρινής αποθήκευσης (buffer) της πρότυπης εξόδου (συνάρτηση fflush) και καλείται η time_delay ώστε να περάσει κάποιος χρόνος μέχρι το επόμενο στιγμιότυπο. Εκεί κλείνει το σώμα του while και τελειώνει η main.

3.3 Συνάρτηση ask_user

Πρωτότυπο: void ask_user(int, int, int*, int*, int*, int*, int*);

Αυτή δέχεται ως ορίσματα το πλάτος και ύψος του πλαισίου ώστε να πληροφορήσει το χρήστη για τις διαστάσεις του κουτιού (με πέρασμα κατά τιμή), καθώς και τις αρχικές συντεταγμένες του κινητού, τις δύο παραμέτρους που καθορίζουν τη σχέση της κατακόρυφης με την οριζόντια μετακίνηση και άρα τη διεύθυνση της ταχύτητας και τέλος, τη συνολική διάρκεια της

προσομοίωσης.

Ο χρήστης ερωτάται αν θέλει να δώσει επιπλέον πληροφορίες και απαντά με τα γράμματα y ή Y για “ναι” και n ή N για “όχι”. Αν απαντήσει ναι, ακολουθεί μία δομή if που περικλείει τις ερωτήσεις και απαντήσεις. Λεπτομέρειες για τον τρόπο παροχής των πληροφοριών δίνονται στις Οδηγίες Χρήσης.

3.4 Συνάρτηση draw_line.

Πρωτότυπο: void draw_line(int);

Αυτή έχει πολύ απλή λειτουργία. Με τη βοήθεια της putchar και επαναλήψεων for, τυπώνει κενά διαστήματα όσο και η μετατόπιση OFFSET, μετά μια κάθετη γραμμή '|', χαρακτήρες '-' όσους το πλάτος που δίνεται ως όρισμα (width), ξανά μια κάθετη γραμμή και ένα χαρακτήρα '\n'. Χρησιμεύει στο σχεδιασμό της πάνω και κάτω οριζόντιας γραμμής του πλαισίου.

3.5 Συνάρτηση draw_point.

Πρωτότυπο: void draw_point(int, int, int, int, int, int, int);

Αυτή είναι η σημαντικότερη συνάρτηση του προγράμματος. Παίρνει ως ορίσματα κατά σειρά, ένα μετρητή (k, που προφανώς αντιστοιχεί στο kount της main), τις διαστάσεις του πλαισίου (width, height), τις αρχικές συντεταγμένες του κινητού, καθώς και τις παραμέτρους που βοηθούν στον καθορισμό της αρχικής κατεύθυνσης κίνησης. Τοπικά στη συνάρτηση δηλώνεται ο πίνακας χαρακτήρων places με height+1 γραμμές και width+OFFSET+2 στήλες. Αυτές οι διαστάσεις εγγυώνται ότι δε θα έχουμε υπέρβαση των ορίων του πίνακα που θα προκαλούσε ανώμαλο τερματισμό με σφάλμα κατάτμησης.

Στην αρχή, με ένα διπλό for, όλα τα στοιχεία του πίνακα τίθενται ίσα με το κενό διάστημα. Μετά, με ένα for που διατρέχει τις γραμμές του πίνακα, τα στοιχεία στη θέση OFFSET και στην προτελευταία θέση κάθε γραμμής γίνονται ίσα με την κατακόρυφη γραμμή, ώστε να σχεδιαστούν οι κάθετες πλευρές του πλαισίου. Συγχρόνως, το τελευταίο στοιχείο κάθε γραμμής τίθεται ίσο με '\0'. Έτσι, κάθε γραμμή γίνεται ένα αλφαριθμητικό και μπορεί να εκτυπωθεί ως τέτοιο.

Μετά, προσδιορίζεται η οριζόντια μετατόπιση του κινητού, δηλαδή σε ποια στήλη του πίνακα (αντιπροσωπευόμενη από τον αριθμοδείκτη j) θα τοποθετηθεί το σύμβολό του. Υπολογίζεται η ποσότητα xmon ως το άθροισμα του μετρητή στιγμιοτύπων k με την αρχική τετμημένη x0. Βάσει της ανάλυσης που κάναμε, το κριτήριο για κίνηση προς τα δεξιά ή αριστερά είναι κατά πόσον η ποσότητα xmon/width είναι άρτιος ή περιττός, δηλαδή η ποσότητα xmon/width%2 είναι 0 ή 1, αντίστοιχα. Τότε, πάλι βάσει της ανάλυσης, η θέση είναι xmon%width ή width-xmon%width, αλλά την προσαυξάνουμε κατά OFFSET επειδή θέλουμε όλος ο πίνακας να είναι μετατοπισμένος ανάλογα. Αυτά υλοποιούνται με μια δομή επιλογής if, όπου η ποσότητα xmon/width%2 παίζει το ρόλο της συνθήκης. Η υπολογιζόμενη οριζόντια θέση αποθηκεύεται στον αριθμοδείκτη j.

Ακολουθεί ο προσδιορισμός της κατακόρυφης μετατόπισης (γραμμή του πίνακα, αριθμοδείκτης i). Όπως και πριν, ορίζουμε μια ποσότητα ymon, αλλά βάσει της ανάλυσής μας, θα εισάγουμε και το λόγο των παραμέτρων καθορισμού διεύθυνσης, dirx/dir_y. Η εντολή

```
ymon = (int)(k * (!dir_y?height:((float)dir_x)/((float)dir_y)));
```

υπολογίζει την παράσταση dir_x/dir_y (πραγματική και όχι ακέραια διαίρεση, εξ ου και η μετατροπή τύπου – (float)). Ο λόγος αυτός πολλαπλασιάζει το μετρητή k, σύμφωνα με το σκεπτικό που εκθέσαμε στην ενότητα 2. Για την απίθανη περίπτωση που έχει δοθεί από το χρήστη τιμή dir_y = 0, ο λόγος dir_x/dir_y αντικαθίσταται από την τιμή height (μια αρκετά μεγάλη τιμή, δηλαδή). Αυτό το πετυχαίνουμε με τη βοήθεια του τριαδικού τελεστή. Η υπολογιζόμενη παράσταση μετατρέπεται σε int και αποδίδεται στην ymon.

Στη συνέχεια, εξασφαλίζεται ότι η ymon θα είναι θετική (στην παρούσα εργασία δεν έχουμε

επεκτείνει την ανάλυση σε αρνητικές τιμές) και της προσθέτουμε την αρχική τεταγμένη y0. Η συνέχεια είναι εντελώς ανάλογη με αυτή για την οριζόντια κίνηση, με μόνη διαφορά ότι δεν προσθέτουμε OFFSET (δε χρειάζεται γιατί έχουμε ήδη αφήσει μερικές κενές γραμμές ήδη από τη main). Η υπολογιζόμενη κατακόρυφη θέση αποθηκεύεται στον αριθμοδείκτη i.

Έχοντας προσδιορίσει τα i και j που αντιστοιχούν στο κινητό, το στοιχείο places[i][j] τίθεται ίσο με τον κατάλληλο χαρακτήρα. Αν τα i και j συμπίπτουν με τιμές που αντιστοιχούν στα άκρα του πλαισίου εκπέμπεται ήχος. Ακολούθως, τυπώνονται οι γραμμές του πίνακα ως αλφαριθμητικά.

3.6 Συνάρτηση time_delay

Πρωτότυπο: void time_delay(int, int*, int);

Σε αυτή τη συνάρτηση εκτελείται ένα διπλό for που δεν κάνει τίποτε άλλο από το να αυξάνει τους μετρητές του. Δεν υπάρχει κάποιος ειδικός λόγος για τον οποίο χρησιμοποιήσαμε διπλό for, πέρα από το ότι μπορούμε να πετύχουμε μεγαλύτερη διάρκεια με τη βοήθεια σταθερών και μεταβλητών int και όχι long, αφού είναι σα να κάναμε ένα for με μέγιστη τιμή μετρητή το γινόμενο αυτών στα δύο επιμέρους for. Στο εσωτερικό for, η μέγιστη τιμή στη συνθήκη ελέγχου είναι πολλαπλασιασμένη επί delay και αυτό επιτρέπει την αυξομείωση της διάρκειας του διπλού βρόχου.

Τέλος, εδώ γίνεται τόσο η αύξηση του χρονικού μετρητή kount όσο και ο έλεγχος (σύγκριση με max_t) που επιτρέπει την έξοδο από το πρόγραμμα με κλήση της συνάρτησης πρότυπης βιβλιοθήκης exit. Η μεταβολή του kount και η συνθήκη ελέγχου συγχωνεύονται σε μία εντολή με τη μεταθεματική (postfix) χρήση του τελεστή αύξησης ++. Η μεταβολή του kount είναι και ο λόγος για τον οποίο περνιέται στην time_delay μέσω αναφοράς και όχι μέσω τιμής.

4. Οδηγίες Εγκατάστασης και Χρήσης

Το πρόγραμμα δεν απαιτεί κάποια ιδιαίτερη εγκατάσταση, παρά μόνο μεταγλώττιση με οποιοδήποτε μεταγλωττιστή της C. Δε χρησιμοποιούνται οποιαδήποτε αρχεία εισόδου ούτε απαιτούνται εξωτερικές βιβλιοθήκες εκτός από την πρότυπη της C. Όσον αφορά τη χρήση του, ισχύουν τα παρακάτω:

4.1 Ορίσματα γραμμής εντολών

Αυτά είναι προαιρετικά αλλά πρέπει να δίνονται με ορισμένη σειρά, και συγκεκριμένα:

- *ταχύτητα προσομοίωσης,*
- *πλάτος (οριζόντια διάσταση) πλαισίου και*
- *ύψος (κατακόρυφη διάσταση) πλαισίου.*

Για να δοθεί το πλάτος πρέπει να υπάρχει και το όρισμα της ταχύτητας και για να δοθεί το ύψος πρέπει να δίνονται τα δύο προηγούμενα κι αυτό γιατί το πρόγραμμα καταλαβαίνει την πληροφορία που αντιπροσωπεύουν τα ορίσματα με βάση τη σειρά που δίνονται. Κάθε όρισμα μπορεί να έχει οποιαδήποτε ακέραια τιμή από 1 και πάνω, αν και δεν θα δώσουν όλες ικανοποιητικά αποτελέσματα (προφανώς δε θέλουμε πολύ μεγάλη καθυστέρηση ούτε πλαίσιο που δε χωράει στην οθόνη).

Μερικά παραδείγματα θα κάνουν πιο κατανοητά τα παραπάνω. Εστω ότι το όνομα του εκτελέσιμου αρχείου είναι animate. Τότε:

% ./animate (κανένα όρισμα· θα ισχύσουν οι default τιμές).

% ./animate 50 (χρονοκαθυστέρηση 50, δυο φορές πιο αργά από το default)

```
% ./animate 50 60 (καθυστέρηση 50, πλάτος πλαισίου 60)
```

```
% ./animate 50 60 40 (καθυστέρηση 50, πλάτος 60, ύψος 40).
```

4.2 Διάλογος με το χρήστη

Μόλις αρχίσει η εκτέλεση, ο χρήστης θα δει το παρακάτω μήνυμα:

```
% ./animate
```

```
Do you want to provide initial coordinates, direction, and simulation duration?
```

```
(Yy/Nn) [κέρσορας]
```

Αν δώσει η ή N (για "όχι"), θα αρχίσει η προσομοίωση, σύμφωνα και με τα ορίσματα της γραμμής εντολών, αν υπήρχαν. Αν δώσει γ ή Y (για "ναι"), ο διάλογος θα συνεχιστεί με τρόπο που απεικονίζεται στο ακόλουθο παράδειγμα:

```
% ./animate
```

```
Do you want to provide initial coordinates, direction, and simulation duration?
```

```
(Yy/Nn) y
```

```
Width = 80. Give me x0: 0
```

```
Height = 30. Give me y0: 15
```

```
Give me the vertical/horizontal ratio [v/h]: 5/2
```

```
Give me the number of frames: 1000
```

Αρχικά, δόθηκε η πληροφορία ότι το πλάτος είναι ίσο με 80, ώστε ο χρήστης να ξέρει σε ποια όρια μπορεί να βάλει την αρχική θέση του κινητού, και ζητήθηκε η αρχική συντεταγμένη x_0 (οριζόντια θέση). Ο χρήστης έδωσε 0 (τέρμα αριστερά). Μετά, δόθηκε το ύψος που είναι ίσο με 30 και ζητήθηκε η αρχική θέση ως προς την κατακόρυφη διεύθυνση, y_0 . Ο χρήστης έδωσε 15, δηλαδή στο μέσον της κατακόρυφης απόστασης από το πάνω άκρο.

Στη συνέχεια ζητήθηκαν οι παράμετροι που καθορίζουν το λόγο της κατακόρυφης (vertical) προς την οριζόντια (horizontal) κίνηση (dy, dx). Όπως ζητά και το πρόγραμμα με την ένδειξη $[v/h]$, οι δύο αριθμοί πρέπει να δίνονται ως λόγος της μορφής x/x , δηλαδή ο χρήστης πρέπει να δώσει δύο ακέραιους αριθμούς χωρισμένους από την πλάγια, $/$, και χωρίς κενά ανάμεσα. Όσο πιο μεγάλος είναι ο πρώτος σε σχέση με το δεύτερο αριθμό, τόσο πιο γρήγορα θα κινείται το σώμα στην κατακόρυφη διεύθυνση σε σχέση με την οριζόντια – και το αντίστροφο.

5. Λεπτομέρειες ανάπτυξης

Το πρόγραμμα αναπτύχθηκε στο λειτουργικό σύστημα SUSE Linux Enterprise Desktop 10.3 με χρήση του μεταγλωττιστή GNU για τη γλώσσα C (gcc version 4.1.0). Επίσης, έγιναν επιτυχείς δοκιμές σε Windows XP με χρήση του ολοκληρωμένου περιβάλλοντος ανάπτυξης DevC++, έκδοση 5.

6. Βιβλιογραφία

Βασίλη Σεφερίδη "C για αρχάριους", εκδόσεις Κλειδάριθμος, Αθήνα 1995.