

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΛΟΓΙΣΜΙΚΟΥ
Η γλώσσα προγραμματισμού C

ΕΡΓΑΣΤΗΡΙΟ 2.2: Εκφράσεις, πίνακες και βρόχοι +

3 Μαΐου 2018

Το σημερινό εργαστήριο αποτελεί συνέχεια του Εργαστηρίου 2, εμβαθύνοντας σε εκφράσεις, πίνακες και βρόχους της γλώσσας C.

Η άσκηση αποτελεί μέρος των πιο προχωρημένων εργασιών του μαθήματος και μπορεί να παραδοθεί μέχρι το τέλος του εξαμήνου, υποχρεωτικά από όσους δεν έχουν επιλέξει απαλλακτική εργασία (project), προαιρετικά από τους υπόλοιπους.

Άσκηση: «Ένα παιχνίδι με κίνηση πάνω σε επίπεδο»

Ξεκινήστε μια νέα εργασία και δημιουργήστε ένα αρχείο προγράμματος με όνομα *ONOMA_lab2.2.c*, όπου αντί για *ONOMA* θα βάλετε το ονοματεπώνυμό σας. Συνδέστε το αρχείο αυτό με την εργασία.

Στην άσκηση αυτή, θα χρησιμοποιήσετε την τεχνική λήψης τυχαίων αριθμών που είδατε στο εργαστήριο 1.1 για να αρχικοποιήσετε κάποιο ταμπλό με τυχαίο τρόπο, και στη συνέχεια να κινείστε πάνω σε αυτό με κατάλληλη είσοδο στο πληκτρολόγιο. Σκοπός του παιχνιδιού είναι να περάσετε από 4 προκαθορισμένα σημεία του ταμπλό, χωρίς όμως να πέσετε σε κάποια από τις παγίδες, οι οποίες τοποθετούνται τυχαία στην αρχή του παιχνιδιού, και αυτό να το επιτύχετε με μικρό αριθμό κινήσεων.

Το ταμπλό θα είναι ένας πίνακας χαρακτήρων, έστω *table*, μεγέθους 31×41. Οι χαρακτήρες αρχικά είναι όλοι κενοί (SPACE), με ένα χαρακτήρα που συμβολίζει τον παίκτη, πχ. 'O', στο κέντρο του ταμπλό, και τέσσερις χαρακτήρες που σημειώνουν τα σημεία από τα οποία ο παίκτης πρέπει να περάσει, πχ. 'x'. Δηλώστε το ταμπλό καθολικά.

Στην αρχή της *main()* αρχικοποιήστε το ταμπλό, δίνοντας σε όλους τους χαρακτήρες την τιμή ' ', εκτός από το χαρακτήρα του παίκτη, ο οποίος θα μπει στη θέση *table[15][20]*. Επίσης, χρησιμοποιήστε τη συνάρτηση *rand()* για να πάρετε τυχαίους αριθμούς μεταξύ 0 και 30 για μια τυχαία γραμμή και μεταξύ 0 και 40 για μια τυχαία στήλη, επαναλαμβάνοντας για τέσσερα συνολικά ζεύγη συντεταγμένων γραμμής και στήλης, όπου θα μουν οι χαρακτήρες 'x'. Αποθηκεύστε τα ζεύγη αυτά σε κατάλληλο πίνακα, και τοποθετήστε τους χαρακτήρες στο ταμπλό. Επαναλάβετε για τέσσερις ακόμα φορές, ώστε να πάρετε τέσσερα ακόμα ζεύγη συντεταγμένων γραμμής και στήλης. Αποθηκεύστε και αυτά τα ζεύγη σε κατάλληλο πίνακα, χωρίς όμως να σημειώσετε στο ταμπλό κάποιον χαρακτήρα. Τα ζεύγη αυτά είναι οι θέσεις των παγίδων, όπου αν βρεθεί ο παίκτης χάνει.

Το παιχνίδι θα παιχτεί με επαναλαμβανόμενη εκτύπωση του ταμπλό. Η εκτύπωση θα πρέπει να φροντίσει να παρουσιάζει εξωτερικά όρια στο ταμπλό. Έτσι, θα πρέπει

πρώτα να εκτυπώνει μια γραμμή με '-', στη συνέχεια 31 γραμμές οι οποίες να αρχίζουν με '|', να συνεχίζουν με τη συμβολοσειρά της γραμμής του πίνακα και να τελειώνουν με '|', και τέλος να εκτυπώνει άλλη μια γραμμή με '-'. Υλοποιήστε την εκτύπωση με τη βοήθεια διαδοχικών κλήσεων της γνωστής συνάρτησης *printf()*.

Μεταξύ διαδοχικών εκτυπώσεων, θα διαβάζεται η κίνηση του παίκτη. Έστω ότι οι χαρακτήρες κίνησης είναι οι 'h', 'j', 'k' και 'l', που παριστάνουν μετακίνηση κατά μία θέση, με τον πρώτο να παριστάνει κίνηση προς τα αριστερά, το δεύτερο προς τα κάτω, τον τρίτο προς τα επάνω και τον τέταρτο προς τα δεξιά. Εδώ μπορείτε να βάλετε και χαρακτήρα τερματισμού, έστω 'q', για πρόωρο τερματισμό του παιχνιδιού. Υλοποιήστε την ανάγνωση χαρακτήρα με τη συνάρτηση *getchar()*, η οποία δεν έχει όρισμα, και επιστρέφει την τιμή τύπου *char* που διαβάστηκε από το πληκτρολόγιο. Κρατήστε τη θέση του παίκτη σε κατάλληλες μεταβλητές γραμμής και στήλης. Με κάθε αποδεκτή κίνηση, αλλάζετε γραμμή ή στήλη κατά 1, εκτός αν ο παίκτης βρίσκεται στην άκρη του ταμπλό και κινείται προς τα έξω, οπότε η κίνηση δεν θα είναι αποδεκτή. Σε περίπτωση αποδεκτής κίνησης, βάλτε κενό στην προηγούμενη θέση του παίκτη, και το χαρακτήρα 'o' στη νέα θέση.

Αν η νέα θέση είναι θέση παγίδας, το παιχνίδι τελειώνει με κατάλληλο μήνυμα. Αν η νέα θέση είναι μία από τις τέσσερις θέσεις χαρακτήρα 'x', συνεχίστε κανονικά, μειώνοντας κατά 1 έναν κατάλληλο μετρητή που έχετε αρχικοποιήσει σε τιμή 4, και εκτυπώνοντας κατάλληλο μήνυμα επιβράβευσης. Αν ο μετρητής αυτός γίνει 0, το παιχνίδι έχει τελειώσει. Χρησιμοποιήστε κι ένα δεύτερο μετρητή για να μετράτε τις κινήσεις του παίκτη, ώστε στο τέλος του παιχνιδιού να εκτυπωθεί και το πλήθος κινήσεων σαν σκορ του παιχνιδιού. Φυσικά το μικρότερο σκορ είναι και το καλύτερο.

Μεταφράστε και δοκιμάστε το πρόγραμμά σας.

Όπως είναι, το παιχνίδι δεν έχει ιδιαίτερο ενδιαφέρον για τον παίκτη, μια που δεν έχει καμιά πληροφόρηση για τις παγίδες του ταμπλό. Για να το κάνετε πιο ενδιαφέρον, να εκτυπώνετε κατάλληλο μήνυμα προσέγγισης παγίδας, χωρίς όμως να αποκαλύπτετε τη θέση της παγίδας. Έτσι, ο παίκτης θα προσπαθεί να πάει γύρω από τις παγίδες, με αποτέλεσμα όμως την αύξηση του πλήθους κινήσεων.

Υλοποιήστε την παραπάνω προσθήκη, ξαναμεταφράστε και δοκιμάστε το πρόγραμμα.

Στη συνέχεια, προσθέστε έναν εξωτερικό βρόχο, με τον οποίο να υλοποιείτε διαδοχικά ταμπλό, αυξανόμενου επιπέδου δυσκολίας. Έτσι, μετά το πρώτο ταμπλό, το δεύτερο ταμπλό μπορεί να έχει έξι παγίδες, το τρίτο δέκα, κοκ. Με την αύξηση του αριθμού παγίδων όμως, εμφανίζεται κίνδυνος να δημιουργούνται περιοχές του ταμπλό που να μην είναι προσβάσιμες από τον παίκτη, οπότε θα πρέπει να προστεθεί κώδικας, ο οποίος θα φροντίζει για κάθε νέα θέση παγίδας να ελέγχει αν δημιουργείται κάποιου είδους αποκλεισμός στο ταμπλό, ώστε η θέση να μη γίνεται αποδεκτή. Υλοποιήστε συνολικά τέσσερα επίπεδα στο παιχνίδι, προσθέτοντας και τον κώδικα ελέγχου θέσης παγίδων.

Τέλος, υλοποιήστε (α) πολλαπλές ζωές του παίκτη, ώστε να μην τερματίζετε το παιχνίδι με την πρώτη παγίδευση, και (β) έναν εναλλακτικό τρόπο αξιολόγησης του παίκτη, βάζοντας αυξανόμενο πλήθος πόντων για κάθε πέρασμα από χαρακτήρα 'x', και δίνοντας ένα τελικό πλήθος πόντων που να εκφράζει το αντίστροφο του πλήθους κινήσεων. Καλύτερο σκορ θα είναι τώρα το μεγαλύτερο.

Μεταφράστε, επαληθεύστε, αποθηκεύστε και υποβάλετε το πρόγραμμά σας.