

Προγραμματισμός II

1^ο project

Διδάσκων:

Χ. Τρυφωνόπουλος

Παράδοση μέχρι: Κυριακή 14/05/2023 ώρα 23.59
Προσωπική εξέταση: θα ανακοινωθεί μέσω eclass

ΣΗΜΑΝΤΙΚΕΣ ΣΗΜΕΙΩΣΕΙΣ:

1. Στο αρχείο που γράφετε τον κώδικα για κάθε εργασία πρέπει ΟΠΩΣΔΗΠΟΤΕ να βάλετε σε σχόλια τα ονόματα, τους A.M., και τα username/email των μελών της ομάδας (ομάδες αυστηρά 2 ατόμων, ατομικές εργασίες δεν γίνονται δεκτές). Όλα τα σχόλια και τα μηνύματα του προγράμματός σας πρέπει να είναι με λατινικούς χαρακτήρες (και όχι στα ελληνικά). Ακολουθεί παράδειγμα:
/* Nikos Tselikas
AM: 2022201500666
dit15666@uop.gr

Christos Tryfonopoulos
AM: 2022201600777
dit16777@uop.gr
*/
2. Αφού έχετε ολοκληρώσει την εργασία που θέλετε να παραδώσετε την υποβάλετε στο eclass στο υποσύστημα «Εργασίες φοιτητών». Προσοχή: μόνο 1 άτομο από την ομάδα χρειάζεται να παραδώσει την εργασία μέσω του e-class! Η υποβολή πρέπει να γίνει ΠΡΙΝ την ημερομηνία παράδοσης. Παραδίδετε ΜΟΝΟ τα αρχεία με τον κώδικα (με κατάληξη .c ή/και .h) σε ένα συμπιεσμένο αρχείο (το οποίο θα φέρει τα ονόματα της ομάδας π.χ., TselikasTryfonopoulos.zip) και ΟΧΙ τα εκτελέσιμα μετά την μεταγλώττιση. Προσοχή: τα προγράμματα που θα παραδώσετε θα πρέπει να κάνουν compile και να τρέχουν στο pelopas.uop.gr. Ασκήσεις οι οποίες δεν κάνουν compile ή δεν τρέχουν στα παραπάνω μηχανήματα του τμήματος θα μηδενίζονται.
3. Περιπτώσεις αντιγραφής θα μηδενίζονται μαζί με όλες τις ασκήσεις που έχουν ήδη παραδώσει και οι εμπλεκόμενοι δεν θα έχουν δικαίωμα παράδοσης άλλων ασκήσεων. Επιπλέον θα παραπέμπονται για περαιτέρω κυρώσεις στα αρμόδια όργανα του Τμήματος.
4. Η ημερομηνία παράδοσης είναι αυστηρή, και η παράδοση γίνεται μόνο μέσω του eclass και όχι με email στον διδάσκοντα ή στους βοηθούς του μαθήματος. Ασκήσεις που παραδίδονται μετά τη λήξη της προθεσμίας δε γίνονται δεκτές.

ΕΚΦΩΝΗΣΗ ΕΡΓΑΣΙΑΣ

Στην εργασία αυτή καλείστε να κατασκευάσετε ένα πρόγραμμα που θα επιτρέπει το παίξιμο του γνωστού παιχνιδιού «Ναρκαλιευτής» των Windows (υπήρχε σε όλες τις εκδόσεις μέχρι και τα Windows 8), πάνω από το οποίο η γενιά μας έχει περάσει άπειρες ώρες ψάχνοντας να βρει νάρκες. Σκοπός του παιχνιδιού που καλείστε να φτιάξετε είναι ο παίκτης να σηματοδοτήσει όλες τις νάρκες που υπάρχουν σε ένα ναρκοπέδιο ή να πεθάνει προσπαθώντας. ☺

Παρακάτω περιγράφονται οι κανόνες του παιχνιδιού και η λειτουργικότητα που καλείστε να υλοποιήσετε. Περισσότερες λεπτομέρειες για τους κανόνες του παιχνιδιού μπορείτε να βρείτε στη [Wikipedia](https://en.wikipedia.org/wiki/Minesweeper), αλλά θα συζητήσουμε και στο μάθημα, στο εργαστήριο, ή/και στην περιοχή συζητήσεων του μαθήματος στο eClass.

1. Κανόνες του παιχνιδιού

Το παιχνίδι περιλαμβάνει ένα ταμπλό αρχικών διαστάσεων NxM (όχι αναγκαστικά τετράγωνο δηλαδή) στο οποίο έχουν τοποθετηθεί νάρκες με τυχαίο τρόπο. Το παιχνίδι εκκινεί με όλα τα τετράγωνα κρυμμένα και ο χρήστης καλείται να αποκαλύψει ένα τετράγωνο δίνοντας τις συντεταγμένες του. Το παιχνίδι τελειώνει

- αν ο παίκτης ανοίξει τετράγωνο που περιέχει νάρκη, οπότε έχει χάσει,
- αν ο παίκτης επιλέξει να τερματίσει το παιχνίδι, ή
- αν ο παίκτης ολοκληρώσει και την τελευταία πίστα του παιχνιδιού.

Σημειώνεται ότι ο παίκτης αλλάζει αυτόματα πίστα όταν αποκαλύψει στην τρέχουσα πίστα όλα τα τετράγωνα που δεν περιέχουν νάρκη. Η διαδικασία αλλαγής πίστας περιγράφεται αναλυτικά στην Ενότητα 6.

Στο διπλανό σχήμα φαίνεται ένα παιχνίδι (από την έκδοση του παιχνιδιού για Windows 3.1) που έχει τελειώσει με επιτυχία ο παίκτης (φυσικά δεν θα υλοποιήσετε το γραφικό περιβάλλον με τις σημαίες κλπ.).



Οι σημαίες έχουν μπει για να σηματοδοτήσουν ποια τετράγωνα έχουν νάρκες, ενώ οι αριθμοί σε κάθε τετράγωνο δείχνουν πόσα γειτονικά τους τετράγωνα έχουν νάρκη. Ως γειτονικά ενός τετραγώνου, ορίζονται τα τετράγωνα που βρίσκονται πάνω, κάτω, δεξιά, αριστερά, και διαγώνια προς όλες τις κατευθύνσεις (πάνω-αριστερά, πάνω-δεξιά, κάτω αριστερά, κάτω-δεξιά).

Το παιχνίδι εκκινεί με όλα τα τετράγωνα του ταμπλό κλειστά και με τον παίκτη να πρέπει να διαλέξει να ανοίξει κάποιο τετράγωνο. Όταν ο παίκτης ανοίξει όλα τα τετράγωνα της τρέχουσας πίστας που δεν έχουν νάρκη περνάει την πίστα και το παιχνίδι αυτόματα του δείχνει την επόμενη πίστα.

Μετά από κάθε άνοιγμα τετραγώνου του παίκτη, και εφόσον δεν άνοιξε τετράγωνο με νάρκη, θα πρέπει να ανοίγουν αυτόματα ένα ή περισσότερα τετράγωνα με τη διαδικασία που περιγράφεται αναλυτικά στις Ενότητες 4 και 5. Το παιχνίδι τελειώνει αν ο παίκτης ανοίξει τετράγωνο με νάρκη, αν επιλέξει να τερματίσει το παιχνίδι, ή αν ολοκληρώσει και την τελευταία πίστα του παιχνιδιού.

2. Αρχή του παιχνιδιού [20 μονάδες]

Με την εκτέλεση του προγράμματος, ο χρήστης θα καθορίζει για μία μόνο φορά και μέχρι να τερματίσει το παιχνίδι:

1. Τις αρχικές διαστάσεις N και M του ταμπλό (το ταμπλό θα πρέπει να έχει ελάχιστο και μέγιστο μέγεθος, το οποίο θα καθορίσετε εσείς και θα αιτιολογήσετε το σκεπτικό σας στην εξέταση, αλλά γενικά δεν είναι ανάγκη να είναι τετράγωνο).
2. Το επίπεδο δυσκολίας του παιχνιδιού, το οποίο καθορίζεται από το ποσοστό των τετραγώνων που θα είναι νάρκες:
 - στο εύκολο παιχνίδι οι νάρκες θα είναι το 10% των συνολικών τετραγώνων της πίστας,
 - στο μέτριο παιχνίδι οι νάρκες θα είναι το 15% των συνολικών τετραγώνων της πίστας,
 - στο δύσκολο παιχνίδι οι νάρκες θα είναι το 20% των συνολικών τετραγώνων της πίστας,
 - και στο ακατόρθωτο παιχνίδι οι νάρκες θα είναι το 25% των συνολικών τετραγώνων της πίστας.

Σημειώνεται ότι αν το πλήθος των ναρκών βγαίνει δεκαδικός αριθμός, μπορείτε να το στρογγυλοποιήσετε όπως προτιμάτε, δηλαδή στον κοντινότερο, στον αμέσως μικρότερο ή στον αμέσως μεγαλύτερο ακέραιο.

Με βάση τα παραπάνω σε ένα παιχνίδι με 200 τετράγωνα (π.χ., με ταμπλό 10x20) στο μέτριο επίπεδο θα υπάρχουν 30 νάρκες και όταν ανοίξει όλα τα τετράγωνα που δεν έχουν νάρκη, η επόμενη πίστα θα μεγαλώσει όπως περιγράφεται στην Ενότητα 6. Σημειώνεται ότι σε κάθε νέα πίστα οι νάρκες τοποθετούνται από την αρχή στο ταμπλό σε νέες θέσεις, οι οποίες παραμένουν (προφανώς) οι ίδιες μέχρι ο παίκτης να αλλάξει ξανά πίστα.

Μετά τον καθορισμό των παραπάνω παραμέτρων θα:

- δεσμεύεται δυναμικά η απαραίτητη μνήμη για το αρχικό ταμπλό (λύσεις με στατικούς πίνακες δεν θα γίνονται δεκτές και δεν θα βαθμολογούνται!),
- γεμίζεται το ταμπλό σε τυχαίες θέσεις του με τον κατάλληλο αριθμό ναρκών (προσοχή, αν μία νάρκη πέφτει πάνω σε άλλη θα πρέπει να της αλλάζετε θέση ώστε τελικά να είναι σωστός ο συνολικός αριθμός ναρκών στο ταμπλό).

Η τυχαιότητα εξασφαλίζεται με χρήση της συνάρτησης rand(). Μην ξεχάσετε να την αρχικοποιήσετε!

3. Αναπαράσταση του ταμπλό [10 μονάδες]

Αρχικά, θα εμφανίζεται το ταμπλό με τα στοιχεία που περιγράψαμε παραπάνω και θα ξεκινά το παιχνίδι, το οποίο θα συνεχίζεται σε πίστες. Η πρώτη πίστα ξεκινά με ένα ταμπλό στις αρχικές διαστάσεις που έδωσε ο χρήστης, και συνεχίζεται αυξάνοντας και τις δυο διαστάσεις κατά 5 τετράγωνα σε κάθε επόμενη πίστα (μέχρι το μέγιστο επιτρεπτό μέγεθος, οπότε και ολοκληρώνεται το παιχνίδι). Σε κάθε πίστα, γεμίζει το ταμπλό εκ νέου με τον ανάλογο αριθμό από

νάρκες (ανάλογα με τη δυσκολία) και το παιχνίδι εξελίσσεται σε γύρους προτρέποντας τον χρήστη να δώσει μία κατάλληλη εντολή.

Στο ταμπλό, όταν εμφανίζεται στην οθόνη, θα πρέπει οι γραμμές και οι στήλες να είναι αριθμημένες (ξεκινώντας από το 1, όχι από το 0), ώστε να διευκολύνεται ο χρήστης στον εντοπισμό των θέσεων. Για την αναπαράσταση του ταμπλό χρησιμοποιούνται το «παπάκι» (@) για να σηματοδοτήσει μία νάρκη, η δίεση (#) για κλειστό τετράγωνο, οι αριθμοί 1-8 για τον αριθμό των γειτονικών τετραγώνων που έχουν νάρκες, και η τελεία (.) για ανοιγμένο κενό τετράγωνο. Προσέξτε ότι τα παραπάνω σύμβολα καθώς και η αρίθμηση του ταμπλό χρησιμοποιούνται για την εμφάνιση του στον παίκτη και δεν έχουν σχέση με το τι αποθηκεύετε και πώς το αποθηκεύετε στον πίνακα της υλοποίησής σας. Επίσης, για να είναι πιο ευανάγνωστο το ταμπλό μεταξύ των μεμονωμένων στοιχείων θα πρέπει στη εκτύπωση (και όχι κατ' ανάγκη στον πίνακα αποθήκευσης) να αφήνετε ένα κενό (space) μεταξύ των στηλών.

Το παιχνίδι παίζεται σε γύρους· σε κάθε γύρο θα εμφανίζεται το ταμπλό στην τρέχουσα κατάστασή του, πόσα τετράγωνα έχει ανοίξει μέχρι εκείνη τη στιγμή ο παίκτης, και μια ερώτηση για το ποιο τετράγωνο θέλει να ανοίξει στην επόμενη κίνηση.

Στο Σχήμα 1 φαίνεται το παράδειγμα ενός αρχικού ταμπλό μεγέθους 7 γραμμών και 9 στηλών, ενώ στο Σχήμα 2 φαίνεται το ταμπλό ενός παιχνιδιού σε εξέλιξη (προσέξτε ότι ο παίκτης δεν έχει ακόμα σημειώσει όλες τις νάρκες και ότι οι νάρκες που έχουν σημειωθεί θεωρούνται κλειστά τετράγωνα – άλλωστε αν τα είχε ανοίξει ο χρήστης θα είχε χάσει).

```
      1 2 3 4 5 6 7 8 9
-----
1| # # # # # # # # #
2| # # # # # # # # #
3| # # # # # # # # #
4| # # # # # # # # #
5| # # # # # # # # #
6| # # # # # # # # #
7| # # # # # # # # #

0 blocks are open.
Make your move(s):
```

Σχήμα 1. Αρχή του παιχνιδιού

```
      1 2 3 4 5 6 7 8 9
-----
1| @ 1 . . . . 1 # #
2| 1 1 . . . 1 2 # #
3| 1 1 . . 1 2 # # #
4| # 2 . . 1 @ # # #
5| # 3 . . 1 # # # #
6| # 4 1 . 2 # # # #
7| # @ 1 . 2 # # # #
```

```
34 blocks are open.
Make your move(s):
```

Σχήμα 2. Παιχνίδι σε εξέλιξη

4. Παίζοντας το παιχνίδι [30 μονάδες]

Για να παίξει κάποιος το παιχνίδι δίνει κατάλληλες εντολές με συγκεκριμένη μορφή και σύνταξη τις οποίες καταλαβαίνει και εκτελεί το παιχνίδι. Παρακάτω περιγράφονται οι εντολές και τα αποτελέσματα κάθε μίας από αυτές.

- Εντολή open(x,y). Με την εντολή αυτή ο παίκτης καθορίζει ποιο τετράγωνο θέλει να ανοίξει δίνοντας την εντολή εισάγοντας τις συντεταγμένες του τετραγώνου που θέλει να ανοίξει. Έτσι για παράδειγμα η εντολή `open(4,9)` σημαίνει «άνοιξε το τετράγωνο που βρίσκεται στη 4^η γραμμή και 9^η στήλη». Αφού ο χρήστης έχει δώσει έγκυρες τιμές, γίνεται έλεγχος αν το τετράγωνο αυτό είναι ήδη ανοιχτό ή όχι. Αν είναι ήδη ανοιχτό, εμφανίζεται κατάλληλο μήνυμα λάθους και το πρόγραμμα ζητάει από το χρήστη να δώσει νέες τιμές. Αν οι τιμές είναι έγκυρες και το τετράγωνο που έδωσε ο χρήστης είναι κλειστό, καλείται κατάλληλη συνάρτηση η οποία ελέγχει αν το συγκεκριμένο τετράγωνο περιέχει νάρκη. Σε αυτήν την περίπτωση το παιχνίδι τερματίζει και ενημερώνεται ο χρήστης ότι έχασε. Αν το τετράγωνο δεν περιέχει νάρκη, τότε θα πρέπει να ανοίξει όπως περιγράφεται στην Ενότητα 5. Προσέξτε ότι θα πρέπει με κάποιον τρόπο να εξασφαλίσετε ότι το πρώτο τετράγωνο που ανοίγει ο παίκτης στο παιχνίδι αποκλείεται να είναι βόμβα. Αυτό μπορεί να συμβεί αν π.χ., γεμίζετε το ταμπλό μετά το πρώτο άνοιγμα τετραγώνου και όποιον άλλο τρόπο σκεφτείτε να εξασφαλίσετε αυτήν την συνθήκη.
- Εντολή mark(x,y). Με την εντολή αυτή ο παίκτης καθορίζει ποιο τετράγωνο θέλει να μαρκάρει ότι περιέχει νάρκη εισάγοντας τις συντεταγμένες του συγκεκριμένου τετραγώνου. Έτσι για παράδειγμα η εντολή `mark(6,2)` σημαίνει «σημείωσε ότι το τετράγωνο που βρίσκεται στην 6^η γραμμή και 2^η στήλη έχει νάρκη». Αφού ο χρήστης έχει δώσει έγκυρες τιμές, γίνεται έλεγχος αν το τετράγωνο αυτό είναι ήδη ανοιχτό ή όχι. Αν είναι ήδη ανοιχτό, εμφανίζεται κατάλληλο μήνυμα λάθους και το πρόγραμμα ζητάει από το χρήστη να δώσει νέες τιμές. Αν οι τιμές είναι έγκυρες και το τετράγωνο που έδωσε ο χρήστης είναι κλειστό, ξαναζωγραφίζεται το ταμπλό με το τετράγωνο που επέλεξε ο χρήστης σημειωμένο με @. Προσέξτε ότι η εντολή `mark(x,y)` είναι αποκλειστικά για διευκόλυνση του χρήστη (για να θυμάται ποια τετράγωνα δεν πρέπει να πατήσει) και δεν έχει επίδραση στο παιχνίδι. Οπότε δεν χρειάζεται να ελέγχετε αν ένα τετράγωνο που σημειώνεται έχει όντως νάρκη.
- Εντολή cheat(x,y). Με την εντολή αυτή ο παίκτης «κλέβει» ζητώντας βοήθεια για το τι έχει πίσω το τετράγωνο που αντιστοιχεί στις συντεταγμένες και το παιχνίδι του απαντά αν το τετράγωνο έχει νάρκη, αν συνορεύει με νάρκες (δίνοντας του και τον αριθμό των ναρκών με τις οποίες συνορεύει), ή αν είναι κενό (δεν συνορεύει με νάρκες ούτε έχει νάρκη). Προσέξτε ότι η εντολή cheat(x,y) δεν ανοίγει τετράγωνα, απλά πληροφορεί τον παίκτη τι βρίσκεται πίσω από το κλειστό τετράγωνο.
- Εντολή exit. Με την εντολή αυτή ο παίκτης ζητάει να τερματίσει το παιχνίδι.

Τέλος, σε όλες τις περιπτώσεις που περιγράφονται παραπάνω, αν ο παίκτης δώσει τις τιμές σε άλλη μορφή από αυτή που περιγράφεται, π.χ. δώσει μη έγκυρη εντολή ή μη έγκυρες τιμές σε κάποια εντολή, το πρόγραμμα θα πρέπει να του εμφανίζει ένα μήνυμα λάθους και να του ζητάει να ξαναδώσει κάποια έγκυρη εντολή.

5. Αναδρομικό άνοιγμα τετραγώνων [20 μονάδες]

Όπως αναφέρθηκε στην προηγούμενη ενότητα, αν ο παίκτης δώσει την εντολή open(x,y) για να ανοίξει κάποιο κλειστό τετράγωνο θα πρέπει να καλείται κατάλληλη συνάρτηση, η οποία ελέγχει αν το συγκεκριμένο τετράγωνο περιέχει νάρκη. Σε αυτήν την περίπτωση το παιχνίδι τερματίζει και ενημερώνεται ο χρήστης ότι έχασε.

Αν το τετράγωνο που ανοίγει ο παίκτης δεν περιέχει νάρκη, τότε θα πρέπει να ανοίξει με βάση τους ακόλουθους κανόνες.

- Αν το τετράγωνο που ανοίγει ο παίκτης γειτονεύει με ένα ή περισσότερα τετράγωνα που έχουν νάρκες, τότε το τετράγωνο ανοίγει, εμφανίζεται κατάλληλος αριθμός που υποδεικνύει τον αριθμό των ναρκών που συνορεύουν με αυτό και η διαδικασία ανοίγματος σταματά.
- Αν το τετράγωνο που ανοίγει ο παίκτης δεν γειτονεύει με τετράγωνα που έχουν νάρκες τότε το τετράγωνο ανοίγει και κατόπιν ελέγχονται και ανοίγουν αναδρομικά όλα τα γειτονικά του τετράγωνα, καθώς και τα γειτονικά αυτών κ.ο.κ. (Θυμηθείτε ότι ως γειτονικά ενός τετραγώνου, ορίζονται τα τετράγωνα που βρίσκονται πάνω, κάτω, δεξιά, αριστερά, και διαγώνια προς όλες τις κατευθύνσεις.) Το αναδρομικό αυτό άνοιγμα σταματά όταν βρούμε ένα «μέτωπο» από τετράγωνα που συνορεύουν με νάρκη (δηλαδή έχουν αριθμό) ή βρούμε τα όρια του ταμπλό.

	1	2	3	4	5	6	7	8	9
1	1	#	#	#	#
2	2	#	#	#	#
3	1	#	#	#	#
4	1	1	.	.	2	#	#	#	#
5	#	2	.	.	1	#	#	#	#
6	#	4	1	.	1	2	#	#	#
7	#	#	1	.	.	1	#	#	#

33 blocks are open.
Make your move(s) :

Σχήμα 3. Παράδειγμα
αναδρομικού ανοίγματος

Στο παράδειγμα του Σχήματος 3, ο παίκτης ξεκίνησε να παίζει με όλο το ταμπλό κλειστό και όταν κάλεσε την open(2,2) βρέθηκε με το ταμπλό στην κατάσταση που φαίνεται μετά από το αναδρομικό άνοιγμα όλων των κενών τετραγώνων μέχρι να βρεθεί το «μέτωπο» από τετράγωνα με αριθμό και τα όρια του ταμπλό.

6. Πίστες και επαναρχικοποίηση ταμπλό [20 μονάδες]

Όταν ο παίκτης ανοίξει όλα τα τετράγωνα που δεν έχουν νάρκη σε μία πίστα, το παιχνίδι αλλάζει αυτόματα πίστα (χωρίς δηλαδή να πατήσει αυτός κάποια εντολή) και πηγαίνει σε επόμενη μεγαλύτερη αλλά ίδιας δυσκολίας. Στην επόμενη πίστα που θα δημιουργηθεί, το ταμπλό θα έχει μεγαλώσει κατά 5 γραμμές και 5 στήλες, καθώς επίσης θα έχει αρχικοποιηθεί με νέες νάρκες σε νέες τυχαίες θέσεις. Σημειώστε ότι οι νάρκες θα πρέπει να είναι όσες υποδεικνύονται από το

επίπεδο παιχνιδιού και τις νέες διαστάσεις του ταμπλό. Αν ο χρήστης περάσει αρκετές πίστες ώστε να φτάσει στις μέγιστες επιτρεπτές διαστάσεις που έχετε ορίσει τότε το παιχνίδι τερματίζεται και σημαίνει ότι ο παίκτης είναι αρκετά γέρος έμπειρος ώστε να έχει παίξει πολλούς Ναρκαλιευτές στα νιάτα του. ☺

7. Παράδειγμα

Παρακάτω φαίνεται ένα παράδειγμα ενός ταμπλό 8 γραμμών και 10 στηλών στο μέτριο επίπεδο.

<pre> 1 2 3 4 5 6 7 8 9 10 ----- 1 # # # # # # # # # # 2 # # # # # # # # # # 3 # # # # # # # # # # 4 # # # # # # # # # # 5 # # # # # # # # # # 6 # # # # # # # # # # 7 # # # # # # # # # # 8 # # # # # # # # # # 0 blocks are open. Make your move(s): </pre> <p>Σχήμα 4. Αρχή του παιχνιδιού.</p>	<pre> 1 2 3 4 5 6 7 8 9 10 ----- 1 1 # # 2 1 1 1 1 # # 3 1 1 . . 1 # # # # # 4 # 2 1 1 1 # # # # # 5 # # # # # # # # # # 6 # # # # # # # # # # 7 # # # # # # # # # # 8 # # # # # # # # # # 25 blocks are open. Make your move(s): </pre> <p>Σχήμα 5. Μετά από την εντολή open(2,2) και το αναδρομικό άνοιγμα των τετραγώνων μέχρι το «μέτωπο» που περιγράφεται στην εκφώνηση.</p>	<pre> 1 2 3 4 5 6 7 8 9 10 ----- 1 1 # # 2 1 1 1 1 # # 3 1 1 . . 1 @ # # # # 4 @ 2 1 1 1 # # # # # 5 # # @ # # # # # # # 6 # # # # # # # # # # 7 # # # # # # # # # # 8 # # # # # # # # # # 25 blocks are open. Make your move(s): </pre> <p>Σχήμα 6. Μετά από τις εντολές mark(4,1), mark(5,3), και mark(3,6) με τις οποίες ο παίκτης σημειώνει τα σημεία που πιστεύει ότι έχει νάρκες.</p>
<pre> 1 2 3 4 5 6 7 8 9 10 ----- 1 1 # # 2 1 1 1 1 # # 3 1 1 . . 1 @ 2 # # # 4 @ 2 1 1 1 # # # # # 5 # # @ # # # # # # # 6 # # # # # # # # # # 7 # # # # # # # # # # 8 # # # # # # # # # # 26 blocks are open. Make your move(s): </pre> <p>Σχήμα 7. Μετά από την εντολή open(3,7). Παρατηρείστε ότι εδώ ανοίγει μόνο το συγκεκριμένο τετράγωνο καθώς συνορεύει με βόμβες (περιέχει αριθμό).</p>	<pre> 1 2 3 4 5 6 7 8 9 10 ----- 1 1 # # 2 1 1 1 1 # # 3 1 1 . . 1 @ 2 3 # # 4 @ 2 1 1 1 1 2 # # # 5 1 2 @ 1 . . 2 # # # 6 . 1 1 1 . . 1 # # # 7 2 # # # # 8 1 # # # # 56 blocks are open. Make your move(s): </pre> <p>Σχήμα 8. Μετά από μερικές ακόμη εντολές open().</p>	<pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 ----- 1 # # # # # # # # # # # # # # 2 # # # # # # # # # # # # # # 3 # # # # # # # # # # # # # # 4 # # # # # # # # # # # # # # 5 # # # # # # # # # # # # # # 6 # # # # # # # # # # # # # # 7 # # # # # # # # # # # # # # 8 # # # # # # # # # # # # # # 9 # # # # # # # # # # # # # # 10 # # # # # # # # # # # # # # 11 # # # # # # # # # # # # # # 12 # # # # # # # # # # # # # # 13 # # # # # # # # # # # # # # Level won! New level challenge! 0 blocks are open. Make your move(s): </pre> <p>Σχήμα 9. Αφού ο παίκτης κέρδισε την πίστα και πέρασε στην επόμενη. Παρατηρείστε ότι και οι δύο διαστάσεις του ταμπλό μεγάλωσαν κατά 5.</p>

Στο Σχήμα 4 βλέπετε το αρχικό ταμπλό που εμφανίζεται στον παίκτη, ενώ στο Σχήμα 5 φαίνεται το ταμπλό αφού ο παίκτης έχει δώσει μία εντολή `open()` και έχουν ανοίξει αναδρομικά τα κενά κελιά μέχρι το «μέτωπο» που περιγράφεται στην Ενότητα 5. Στο Σχήμα 6 ο παίκτης έχει σημειώσει κάποια τετράγωνα στα οποία πιστεύει ότι έχουν νάρκες (παρατηρείστε ότι τα τετράγωνα αυτά δεν θεωρούνται ανοικτά και δεν προσμετρώνται ως τέτοια). Στο Σχήμα 7 ο παίκτης ανοίγει ένα τετράγωνο που τυχαίνει να συνορεύει με νάρκες και άρα να έχει αριθμό· παρατηρείστε ότι το άνοιγμα ενός τέτοιου τετραγώνου δεν προκαλεί το αναδρομικό άνοιγμα άλλων και άρα το τετράγωνο αυτό είναι το μόνο που ανοίγει. Κατόπιν, στο Σχήμα 8 ο παίκτης προχωρά στο παιχνίδι και, αφού κερδίσει το τρέχον επίπεδο, το παιχνίδι αυτόματα τον πηγαίνει στο επόμενο όπου οι γραμμές και οι στήλες έχουν αυξηθεί κατά 5 (και ανάλογα έχουν αυξηθεί και οι κρυμμένες νάρκες αφού αυξήθηκαν τα τετράγωνα του ταμπλό). Κατ' αυτόν τον τρόπο το παιχνίδι θα συνεχιζόταν μέχρι το ταμπλό να φτάσει στις μέγιστες επιτρεπτές διαστάσεις ή ο χρήστης να βαρεθεί να κάνει τον ναρκαλιευτή ή να νιώσει πολύ νέος για να παίζει τέτοια παιχνίδια και να τερματίσει το παιχνίδι δίνοντας την εντολή `exit`.

8. Υλοποίηση και bonus

- Για να πάρετε (κάποιες από) τις μονάδες που αναφέρονται θα πρέπει να μπορείτε να επιδείξετε τη λειτουργικότητα που αντιστοιχεί στις μονάδες αυτές. Η υλοποίηση χωρίς τη δυνατότητα επίδειξης (π.χ., επειδή το πρόγραμμα κολλάει) δεν συνεπάγεται ότι θα βαθμολογηθείτε για την υλοποιημένη λειτουργικότητα.
- Λύσεις με στατικούς πίνακες δε θα γίνονται δεκτές και δεν θα διορθώνονται.
- Θα πρέπει να ελέγχετε κάθε φορά την εγκυρότητα των εντολών που σας δίνει ο παίκτης και να ενημερώνετε τον παίκτη με κατάλληλα μηνύματα αν κάτι πήγε στραβά.
- Η τυχαιότητα θα υλοποιηθεί με τη συνάρτηση `rand()`. Μην ξεχάσετε την αρχικοποίηση της `rand()`, τους ελέγχους στη δέσμευση μνήμης και την απελευθέρωσή της το συντομότερο δυνατό (όταν δεν την χρειάζεστε άλλο).

Ενσωμάτωση έξτρα χαρακτηριστικών στο παιχνίδι (σε συνεννόηση με τον διδάσκοντα), προσθήκες που έχετε να προτείνετε, ή ακόμη και η υλοποίηση καλών/έξυπνων λύσεων σε πρακτικά προβλήματα της εργασίας μπορεί να πάρει μέχρι 10% bonus στη βαθμολογία, ανάλογα με τη δυσκολία της υλοποίησης. Ως ενδεικτική προσθήκη αναφέρεται η υλοποίηση βοήθειας με τη μορφή υποδείξεων προς το χρήστη ως προς το ποιο τετράγωνο να ανοίξει. Η βοήθεια μπορεί να προτείνει στον παίκτη να ανοίξει ένα τετράγωνο που συνεπάγεται το άνοιγμα πολλών άλλων ή να του προτείνει την καλύτερη επιλογή, δηλαδή το τετράγωνο που θα προκαλέσει το άνοιγμα των περισσότερων δυνατών τετραγώνων.

Καλή δουλειά!