

## Περιγραφή ηλεκτρονικού παιχνιδιού 21

(σύνθετη περίπτωση δηλ. ο υπολογιστής κάνει και το παίχτη και την μάνα.)

Ξεκινώντας το πρόγραμμα ζητάω από τον χρήστη τον αριθμό των ατόμων που θα παίξουν καθώς και το συνολικό πόσο που θα μπορεί να χρησιμοποιήσει κάθε παίχτης στο παιχνίδι! Έπειτα δεσμεύω δυναμικά με την χρήση της `calloc` έναν πίνακα `a` μεγέθους ίσου με το αριθμό των παιχτών ο οποίος θα είναι είδους `struct player` όπου `struct player` είναι μια δομή που έχω φτιάξει για να περιέχει όλα τα στοιχεία ενός παίχτη δηλ. το όνομα του (`name`), το συνολικό ποσό που του απομένει στο παιχνίδι (`Sposo`), το ποντάρισμα του σε κάθε γύρα (`rontar`) όπου αλλάζει σε κάθε νέα γύρα και τέλος του πόντους του (`rontoi`) όπου σε κάθε νέα γύρα μηδενίζουν! Στην συνέχεια ζητάω από τον χρήστη να μου δώσει τα ονόματα των παιχτών που θα παίξουν βάζοντας τον κάθε παίχτη σε μια αριθμημένη θέση (η οποία ουσιαστικά δεν έχει καμία σημασία αφού το τραπέζι είναι κυκλικό)! Μετά καλώ μια συνάρτηση για την επιλογή της μάνας (`myrand`) η οποία επιστρέφει τυχαία ένα αριθμό από το με 0 με τον αριθμό των ατόμων μείων 1. (Γενικά η `myrand` είναι μια συνάρτηση η οποία επιστρέφει αποτέλεσμα τύπου `int` και δέχεται ως όρισμα έναν ακέραιο `a` ο οποίος καθορίζει το εύρος των αποτελεσμάτων που μπορεί να επιστρέψει η `myrand` π.χ αν `a=6` τότε το εύρος είναι από μηδέν μέχρι 5). Έτσι έχω ένα τυχαίο αριθμό όπου ο παίχτης που κάθεται στην θέση του πίνακα `a` με εκείνο τον αριθμό θα είναι η αρχική μάνα! Έπειτα ξεκινάω ένα 'μεγάλο' βρόγχο επανάληψης (κάθε φορά που ξαναρχίζει από την αρχή αυξάνω την θέση της μάνας κατά 1) ο οποίος τελειώνει μόνο όταν ένας παίχτης μαζέψει όλα τα λεφτά των αντιπάλων του δηλ. τελειώσει το παιχνίδι (ο έλεγχος αυτός γίνεται στο τέλος του βρόγχου)! Ακολουθώ καλώ μια συνάρτηση επαναδιάταξης (`erapnad`) η οποία αυτό που κάνει είναι να μετακινεί μέσα στον πίνακα `a` όλους τους παίχτες (κυκλικά βέβαια προκειμένου να μην χαθεί η σειρά τους) με σκοπό να έρθει στην πρώτη θέση του πίνακα (`a[0]`) η μάνα. Έτσι να μπορώ να έχω σαν γενικό κανόνα ότι σε κάθε γύρα η μάνα βρίσκεται στην πρώτη θέση του πίνακα `a` (λόγω του ότι η μάνα δεν παραμένει σταθερή). Αν η μάνα είναι ο υπολογιστής (`pc`) έχω ορίσει ότι θα παίζεται ως κάβα το 1/6 του συνολικού ποσού που θα έχει εκείνη την στιγμή ο υπολογιστής αλλιώς αν η μάνα είναι οποιοσδήποτε άλλος παίχτης ζητάω από αυτόν να 'πεί' ποια είναι η κάβα την οποία θέλει και αυτός απαντάει σύμφωνα με τις προτιμήσεις του! Έπειτα ξεκινάει ένας νέος 'μικρότερος' από τον αρχικό βρόγχος όπου τερματίζει αν η μάνα έχει χάσει όλη την κάβα της (στο τέλος αυτού του 'μικρότερου' βρόγχου ελέγχω αν η κάβα είναι μηδέν) είτε αν τα έχει τριπλασιάσει την κάβα της (έχω ένα μετρητή (`metritis2`) ο οποίος αρχίζει να μετράει αφού η μάνα τριπλασιάσει τα λεφτά της και στο τέλος του μικρότερου βρόγχου ελέγχουμε αν έχει γίνει ο μετρητής αυτός ίσος με το των αριθμό των ατόμων που παίζουν απέναντι στη μάνα) και έχει παιχτεί και μια τελευταία γύρα. Στην συνέχεια γίνεται το μοίρασμα του πρώτου φύλλου και ανακοινώνεται σε κάθε παίχτη ποιο είναι το φύλλο του και οι πόντοι του! Αυτά γίνονται με την χρήση 2 συναρτήσεων, της `moirasm` και της `showcard`. Η `moirasm` (επιστρέφει μια μεταβλητή `struct fullo` όπου `struct`

fullo είναι μια δομή που έχω φτιάξει και κάθε στοιχείο της ουσιαστικά είναι ένα φύλλο της τράπουλας από το 7 μέχρι τον άσσο και αποτελείται από τρία πεδία ,το νούμερο, το σύμβολο και την αξία του δηλ. για πόσους πόντους μετράει το φύλλο). Αυτή καλεί την mygrand 2 φορές ώστε να πάρει 2 τυχαίους αριθμούς και με την χρήση αυτών διαλέγει μέσα από δύο πίνακες τον αριθμό του και το σύμβολο του φύλλου. Η showcadd η οποία δέχεται ως ορίσματα τον αριθμό του φύλλου και το σύμβολο και εκτυπώνει το φύλλο στην οθόνη! Έπειτα ξεκινάει ένας νέος βρόγχος (for) ο οποίος ουσιαστικά τερματίζει είτε όταν η μάνα παίξει με όλους τους παίκτες δηλ. τελειώσει μια μοιρασιά είτε όταν μάνα έχει χάσει όλα τα λεφτά της προτού τελειώσει η μοιρασιά είτε αν είχαν 3πλασιαστεί τα λεφτά της μάνας και έχει παιχτεί μια ολόκληρη γύρα ! Μετά γίνεται ένας έλεγχος για τον αν ο παίχτης με τον οποίο θα παίξει μάνα είναι ο υπολογιστής ή όχι. Αν δεν είναι γίνεται ερώτηση στον παίχτη για το ποιο θέλει να είναι το ποντάρισμα του. Αντίθετα αν ο παίχτης είναι ο υπολογιστής καλείται μια συνάρτηση (pcountar) η οποία δέχεται ως όρισμα την αξία του πρώτου φύλλου που έχει πάρει ο υπολογιστής και δίνει το ανάλογο ποντάρισμα(αν η αξία είναι μεγαλύτερη του 9 το ποντάρισμα θα είναι το  $\frac{1}{4}$  της κάβας που έχει υπάρχει εκείνη την στιγμή ,αν η αξία είναι μικρότερη του 5 τότε θα είναι το  $\frac{1}{8}$  της κάβας και στις υπόλοιπες περιπτώσεις θα είναι το  $\frac{1}{6}$  της κάβας)! Ακολούθως αν ο παίχτης δεν είναι ο υπολογιστής ρωτάτε αν θέλει και άλλο φύλλο. Αν απαντήσει ναι (y) τραβάει και άλλο φύλλο και η αξία αυτού του φύλλου προστίθεται στους πόντους που έχει ήδη αλλιώς αν απαντήσει όχι σταματάει στους πόντους που έχει μέχρι εκείνη την στιγμή! Αυτή η διαδικασία της ερώτησης και της απάντησης συνεχίζεται όσο ο παίχτης απαντάει ναι (y) και οι πόντοι του λιγότεροι ή ίσοι με 21 .Αντίθετα αν ο παίχτης είναι ο υπολογιστής τότε αυτός ζητάει φύλλο όσο οι πόντοι είναι κάτω από 17 και σταματάει στην περίπτωση που οι πόντοι του γίνουν 18 και πάνω! Όσον αφορά την περίπτωση που είναι 17 όπου είναι ένα 'διχασμένο' σημείο για έναν παίχτη καθώς η μάνα πάντα κερδίζει τις ισοπαλίες καλώ την mygrand και με πιθανότητα 50%-50% ο υπολογιστής ζητάει φύλλο ή όχι! Επίσης είναι χρήσιμο να αναφέρω ότι έχω και έναν μετρητή (metritis) ο οποίος κάθε φορά ο παίχτης ζητάει ένα φύλλο(είτε αυτός είναι ο υπολογιστής είτε όχι) αυξάνεται κατά ένα και όταν έρχεται η σειρά του επόμενου παίχτη να παίξει με την μάνα αυτός ο μετρητής μηδενίζει! Υπάρχει και μια εξαίρεση : Σε περίπτωση που ο παίχτης έχει τραβήξει 2 άσσους δεν καίγεται, αντίθετα κερδίζει κατευθείαν χωρίς να παίξει η μάνα και το ποσό που έχει ποντάρει διπλασιάζεται και προσθέτετε στο συνολικό ποσό του(αυτόν τον έλεγχο τον κάνω τσεκάροντας αν ο μετρητής (metritis) είναι 2 και οι συνολικοί πόντοι του παίχτη είναι ίσοι με 22).Επιπρόσθετα στο παιχνίδι 21 που υλοποίησα υπάρχει ένας κανόνας που λέει ότι αν ο παίχτης έχει τραβήξει 5 φύλλα και οι πόντοι του δεν έχουν υπερβεί τους 20 τότε η μάνα κερδίζει μόνο αν μαζέψει 21 πόντους(τι συμβαίνει σ αυτήν την περίπτωση θα πούμε πιο κάτω)! Τώρα νομίζω πως είναι χρήσιμο να περιγράψω δύο πολύ βασικές συναρτήσεις την niki και την itta οι οποίες και οι δύο είναι τύπου void.Η συνάρτηση niki καλείται όταν ο παίχτης κερδίσει ( με οποιοδήποτε τρόπο και αν γίνει αυτό) και δέχεται ως ορίσματα έναν αριθμό (px i ) που δείχνει σε ποια θέση του πίνακα a βρίσκονται τα στοιχεία του παιχτεί που κέρδισε, τον πίνακα a που περιέχει τα στοιχεία όλων των παικτών, και την διεύθυνση μνήμης της μεταβλητής όπου είναι

αποθηκευμένο το ποσό της κάβας (&kaba). Αφού δεχτεί αυτά τα στοιχεία κάνει τις εργασίες που συνεπάγονται με την νίκη ενός παίχτη δηλ. αφαιρεί το ποσό που είχε ποντάρει ο παίχτης που κέρδισε (αυτός που βρίσκεται στην θέση i) από το συν. ποσό (Sproso) της μάνας (ο παίχτης που βρίσκεται στην θέση 0) και την κάβα(επειδή θέλουμε να παραμείνει αλλαγμένη η τιμή της κάβας και αφού κλείσει η συνάρτηση γι αυτό την στέλνουμε στην συνάρτηση με αναφορά και όχι με τιμή) και προσθέτει αυτό το ποσό στο συν. ποσό του παίχτη που κέρδισε (αυτός που βρίσκεται στην θέση i).Επίσης μηδενίζει το ποντάρισμα του παίχτη ( $a[i].pontar=0$ ) και τους πόντους του( $a[i].pontoi=0$ ). Τέλος εκτυπώνει αυτά τα στοιχεία και στην οθόνη ενημερώνοντας τους ενδιαφερόμενους παίχτες. Αντίστοιχα η itta κάνει σχεδόν τις ίδιες λειτουργίες απλά αφαιρεί το ποντάρισμα από το συνολικό ποσό (Sproso) του παίχτη και τα προσθέτει στην κάβα και στο συν. ποσό της μάνας. Επίσης και αυτή μηδενίζει το ποντάρισμα του παίχτη ( $a[i].pontar=0$ ) και τους πόντους του. Τέλος εκτυπώνει τα ανανεωμένα στοιχεία. Έτσι κάθε φορά που κερδίζει ή χάνει κάποιος παίχτης απλά καλώ την αντίστοιχη συνάρτηση! Εφόσον τελειώσει το μοίρασμα φύλλων σε κάποιο παίχτη τότε γίνεται ένας έλεγχος για να δούμε αν ο παίχτης έχει κερδίσει ή χάσει ήδη(αν έχει τραβήξει δυο άσσους δηλ. έχει κερδίσει ή έχει τραβήξει παραπάνω από 21 πόντους δηλ. έχει καεί) δηλαδή αν χρειάζεται να τραβήξει μάνα ή όχι. Αυτό το κάνω συγκρίνοντας το ποντάρισμα του παίχτη με το 0 μιας και αν έχει καλεσθεί είτε η συνάρτηση νίκης είτε ήττας το ποντάρισμα θα έχει μηδενιστεί. Σε περίπτωση που είναι διάφορη του μηδενός τότε παίρνει φύλλο και η μάνα. Αν η μάνα είναι ο υπολογιστής τότε υπάρχει ο γενικός κανόνας ότι συνεχίζει και τραβάει όσο το άθροισμα των πόντων των φύλλων του είναι μικρότερο του 17 αφού η μάνα κερδίζει και τις ισοπαλίες (βέβαια στην περίπτωση όπου ο μετρητής (metritis) είναι μεγαλύτερος του 5 τότε ο υπολογιστής τραβάει φύλλα είτε μέχρι να πάει 21 είτε μέχρι να καεί και έτσι καλείται μια από τις συναρτήσεις niki και itta).Αντίθετα αν η μάνα δεν είναι ο υπολογιστής τότε ο παίχτης παίρνει φύλλο όσο απαντάει ναι (y) στην ερώτηση αν θέλει άλλο φύλλο μέχρι να απαντήσει όχι (n) ή να καεί (και εδώ αν ο μετρητής (metritis) είναι μεγαλύτερος του τότε επιβάλλεται στον παίχτη να τραβάει φύλλα είτε μέχρι να πάει 21 είτε μέχρι να καεί έτσι καλείται μια από τις συναρτήσεις niki και itta). Αν η μάνα καεί τότε ο παίχτης κερδίζει και καλείται κανονικά η συνάρτηση niki. Αν η μάνα δεν καεί τότε συγκρίνονται οι πόντοι της μάνας και οι πόντοι του παίχτη. Ο παίχτης κερδίζει μόνο αν έχει περισσότερους πόντους απ την μάνα και σε κάθε άλλη περίπτωση κερδίζει η μάνα (και βέβαια όποιος και αν κερδίσει καλείται η αντίστοιχη συνάρτηση niki ή itta). Μετά γίνεται αυτός έλεγχος που ανέφερα και στην αρχή για το αν έχουν μαζευτεί όλα τα λεφτά σε ένα παίχτη δηλ. αν θα εκτελεστεί ο αρχικός μεγάλος κόμβος από την αρχή ή υπάρχει νικητής και τελειώνει το παιχνίδι! Τέλος θα ήθελα να αιτιολογήσω την επιλογή μου σχετικά με το ότι μεταξύ κλήσεων συναρτήσεων που επιστρέφουν τυχαίο αποτέλεσμα αλλά και γενικότερα μέσα στο πρόγραμμα μου χρησιμοποιώ ερωτήσεις του τύπου ("to pc thelei kialo fullo.gia na sunexisoume pata (y) ") και ("na sunexisoume ston epomeno paxiti?? (y/n)") κτλ. Αυτό το κάνω κατά κύριο λόγο για να υπάρχει καλύτερος έλεγχος του παιχνιδιού και να μην εκτυπώνονται συνεχόμενα στοιχεία-αποτελέσματα (π.χ στο μοίρασμα του πρώτου φύλλου να μην εκτυπώνονται τα φύλλα όλων των παιχτών στιγμιαία όλα μαζί),

και δεύτερον για να υπάρχει ανάμεσα στις κλήσεις της rand έστω ένα μικρό κενό χρόνου ώστε να μην επιστρέφονται συνεχόμενα ίδια αποτελέσματα καθώς η συγκεκριμένη συνάρτηση χρησιμοποιεί την ώρα ( το τελευταίο είναι κάτι που παρατήρησα εγώ στις δοκιμές του παιχνιδιού και δεν ξέρω αν ισχύει στην πραγματικότητα)!

**Αριστείδης Μουστάκας**

**AEM: 2380**