



# HUNGER GAMES

Νέα έκδοση

Συγγραφή κώδικα παιχνιδιού:

- Παπαποστόλου Βασιλική
  - AEM:9935
- Γκούμα Βασιλική
  - AEM : 9755

Στην παρούσα εργασία καλούμαστε να γράψουμε τον κώδικα για μια απλουστευμένη παραλλαγή του παιχνιδιού “Hunger Games”.

Η πλειοψηφία των επεξηγήσεων των κλάσεων, μεταβλητών και συναρτήσεων που χρησιμοποιήθηκαν για την συγγραφή του κώδικα ,βρίσκονται μέσα σε αυτόν για καλύτερη κατανόησή του.

Κύριο μέρος στη δημιουργία του κώδικα αποτελεί η δημιουργία του ταμπλό NxM .Το ταμπλό αποτελείται από κουτάκια(πλακίδια) στα οποία ορίζουμε συντεταγμένες σύμφωνα με την εκφώνηση της άσκησης(πχ δεξ εικόνα 3 της εκφώνησης της άσκησης) Άρα το κέντρο του ταμπλό βρίσκεται σε συντεταγμένες (0,0) και τα x ,y μεταβάλλονται ανάλογα με το πλακίδιο ,σύμφωνα με την εικόνα που αναφέρθηκε παραπάνω.

Η αρίθμηση των x ,αφού ο αριθμός των στηλών του ταμπλό είναι M ,είναι από  $-M/2$  έως και  $M/2$ . Η αρίθμηση των y ,αφού ο αριθμός των στηλών του ταμπλό είναι N, είναι από  $-N/2$  έως και  $N/2$ .Κρίνεται ιδιαίτερος σημαντικό να σημειωθεί ότι τα x , y άσχετα με τις διαστάσεις του ταμπλό δεν θα πάρουν ποτέ την τιμή 0.Η παραπάνω παρατήρηση θα αλλάξει ελαφρώς τον τρόπο λειτουργίας ορισμένων συναρτήσεων. Το ταμπλό του παιχνιδιού δημιουργείται με βάση τις τιμές x και y που θα πάρουν οι συντεταγμένες των παγίδων ,εφόδιων, όπλων και το εύρος των τιμών που μπορεί να κινηθεί ο παίκτης(ορίζει τα μέγιστα και ελάχιστα x,y του ταμπλό).

Άρα εφόσον ο πίνακας έχει κέντρο με συντεταγμένες (0,0) ,με βάση τα παραπάνω για να αφαιρέσω τα εξωτερικά πλακίδια του ταμπλό το μόνο που χρειάζεται να κάνω είναι να μειώσω κατά μία μονάδα τις ακραίες τιμές που μπορεί να πάρει το x και το y.Με άλλα λόγια, αρκεί να μειώσω το μέγεθος των γραμμών και των στηλών κατά 2 μονάδες ( συνάρτηση : `public void resizeBoard(Player p1,Player p2)`)

Άλλες συναρτήσεις :

- ο Συναρτήσεις : `public void createRandomWeapon()`,`public void createRandomTrap()`,  
`public void createRandomFood()`

Οι παραπάνω συναρτήσεις δημιουργούν τα όπλα ,τα εφόδια και τις παγίδες του παιχνιδιού ελέγχοντας να βρίσκονται στα όρια τα οποία θα οριστούν στη συνάρτηση `main` .Επίσης να μην είναι πάνω από ένα σε κάθε πλακίδιο ,να έχουν χαρακτηριστικό(μοναδικό ως προς τα άλλα αντικείμενα του ίδιου είδους) `id`,να υπάρχουν όλα τα είδη αντικειμένων για τις συνθήκες που ορίζονται στο συγκεκριμένο πρόγραμμα(αναφέρεται στα όπλα και τις παγίδες) και τέλος για τις συνθήκες που ορίζει το πρόβλημα να αντιστοιχούν 3 όπλα (ένα από κάθε είδος )σε κάθε παίκτη .Για να επιτευχθεί αυτό θα πρέπει για κάθε αντικείμενο που δημιουργείται να γίνεται μία σύγκριση ως προς τα προηγούμενα που έχουν δημιουργηθεί. Όπως έχει δομηθεί το πρόγραμμα δεν μπορούμε να αρχίσουμε την σύγκριση από το πρώτο αντικείμενο που δημιουργείται. Για αυτό πρώτα ορίζουμε ένα και διαμορφώνουμε τα υπόλοιπα με βάση αυτό και αυτά που θα ακολουθήσουν.

- ο Συνάρτηση: `public void getStringRepresentation()`

Η συνάρτηση αυτή δημιουργεί και επιστρέφει ένα πίνακα NxM τύπου String ο οποίος με την κατάλληλη εκτύπωση μας εμφανίζει τον πίνακα του παιχνιδιού.

Ο τρόπος όμως αρίθμησης των γραμμών και των στηλών του πίνακα διαφέρει με τον τρόπο αρίθμησης των συντεταγμένων x,y .Πρέπει να βρεθεί ένας τρόπος σύγκρισης των στοιχείων του πίνακα με τις συντεταγμένες των στοιχείων του ταμπλό καθώς πρέπει να ελεγχθεί άμα μία θέση του πίνακα αντιστοιχεί σε πλακίδιο που έχουμε δημιουργήσει όπλο ,παγίδα, εφόδιο ή όχι ,ώστε με την κατάλληλη χρήση της να μπορεί να τυπώσει την ύπαρξη ή την απουσία του αντίστοιχου αντικειμένου.

Υπάρχει λοιπόν μια συσχέτιση μεταξύ στηλών(j),γραμμών(i) και συντεταγμένων x,y.

Στο πάνω αριστερά τεταρτημόριο του πίνακα η συσχέτιση γίνεται μέσω των σχέσεων :

$$x = -M/2 + j$$

$$y = N/2 + i$$

Στο πάνω δεξιά τεταρτημόριο του πίνακα η συσχέτιση γίνεται μέσω των σχέσεων :

$$x = -M/2 + j + 1$$

$$y = -N/2 + i$$

Στο κάτω αριστερά τεταρτημόριο του πίνακα η συσχέτιση γίνεται μέσω των σχέσεων :

$$x = -M/2 + j$$

$$y = -N/2 + i + 1$$

Στο κάτω δεξιά τεταρτημόριο του πίνακα η συσχέτιση γίνεται μέσω των σχέσεων :

$$x = -M/2 + j + 1$$

$$y = -N/2 + i + 1$$

Ο λόγος ως προς τη διαφοροποίηση των σχέσεων ανάλογα με το τεταρτημόριο του πίνακα τίθεται στο γεγονός ότι οι μεταβλητές x,y(που συμβολίζουν τις συντεταγμένες) δεν παίρνουν την τιμή 0.Έτσι ενώ οι μεταβλητές i,j που συμβολίζουν την γραμμή και τη στήλη αντίστοιχα παίρνουν όλες τις τιμές εντός των ορίων που ορίστηκαν, οι συντεταγμένες x,y παραλείπουν την τιμή μηδέν .Έτσι οι παραπάνω σχέσεις αλλάζουν μόλις περάσεις τον άξονα των x ή των y(τα σημείο όπου παραλείπονται οι τιμές).

ο Συνάρτηση : `public int[] getRandomMove(Board board)`

Στη συνάρτηση αυτή ,εφόσον είχατε αναφέρει ότι έχουμε τη δυνατότητα να προσθέσουμε παραμέτρους σε συναρτήσεις προτιμήσαμε να βάλουμε την παράμετρο που φαίνεται παραπάνω .Η συνάρτηση αυτή θέτει τυχαία τη κατεύθυνση που θα κινηθεί ο παίκτης (ελέγχοντας άμα έχει τη δυνατότητα να κατευθυνθεί προς τα εκεί ) και ορίζει(και επιστρέφει) τη θέση που θα πάρει ο παίκτης μόλις κινηθεί κατά την κατεύθυνση που τέθηκε .Είναι σημαντικό να σημειωθεί ότι άμα κατά την κίνησή του ο παίκτης περνάει πάνω από τον άξονα των x, το y που συμβολίζει τη συντεταγμένη y της θέσης του παίκτη αυξάνεται ή μειώνεται κατά 2 (αντί για 1) καθώς δεν παίρνει την τιμή 0.Αντίστοιχα άμα κατά την κίνησή του ο παίκτης περνάει πάνω από τον άξονα των y, το x που συμβολίζει τη συντεταγμένη x της θέσης του παίκτη αυξάνεται ή μειώνεται κατά 2 (αντί για 1 )καθώς επίσης δεν παίρνει την τιμή 0.

Η επικεφαλίδα που ζητείται με τα στοιχεία μας στον κώδικα βρίσκεται στην αρχή της κλάσης Game.

Τέλος όπως προαναφέρθηκε, περισσότερες λεπτομέρειες ως προς τις συναρτήσεις που υλοποιήθηκαν υπάρχουν μέσα στον κώδικα.