

Αντικειμενοστραφής Προγραμματισμός

Επεισόδιο II

Κελέση Ελπίδα 9410 elpidakelesi@ece.auth.gr

Μυλωνάς Εμμανουήλ 9508 evmylonas@ece.auth.gr



Εικόνα 1: Μετά από 500 γραμμές κώδικα σε 2 μέρες

Main:

Η βιτρίνα του παιχνιδιού. Ξεκινάμε με το να ορίσουμε δύο global μεταβλητές τύπου Team, έτσι ώστε να είναι ορατές σε όλες τις συναρτήσεις που ακολουθούν. Στη συνάρτηση αρχικών συνθηκών του πρώτου αντικειμένου βάζουμε το red (aka κόκκινο) και στη δεύτερη το blue (μπλε), ώστε να διαχωρίζουμε τις ομάδες. Ζητείται από το χρήστη να δώσει έναν αριθμό, ώστε να επιλέξει την ενέργεια που θέλει να πραγματοποιήσει. Με τη βοήθεια της switch, κατευθύνουμε τον χρήστη στην ανάλογη συνάρτηση. 1 για να οδηγηθεί στην addPlayer, 2 για να δει τα στοιχεία μιας ομάδας μέσω της seeTeamStatus, 3 για να δει τα στοιχεία παίκτη με την seePlayer και 0 για να βγει από το παιχνίδι. Με τη χρήση της while, όσο ο χρήστης δεν πατά το 0, μπορεί να επαναλάβει τη διαδικασία όσες φορές θέλει. Τέλος αφού πατήσει 0, βγαίνει από την while και τελικά και από το πρόγραμμα.

Επεξήγηση επιμέρους συναρτήσεων:

-addPlayer:

Με τον καθορισμό της bool μεταβλητής από το χρήστη, διαλέγεται η ομάδα. Έπειτα, με τη χρήση της if και έλεγχο της μεταβλητής, οδηγούμαστε στη σωστή ομάδα. Εκεί αναλαμβάνει η συνάρτηση setListOfPlayers, (θα αναλυθεί διεξοδικά στην Κλάση Team), που δίνει στο χρήστη την δυνατότητα να βάλει τα επιμέρους στοιχεία που θέλει στο αντικείμενο Παίκτη που θέλει.

-seeTeamStatus

Παρόμοια δομή με παραπάνω, ο χρήστης επιλέγει την ομάδα που θέλει, και μετά με τη συνάρτηση teamStatus εμφανίζονται όλες οι τιμές που έχει βάλει στην ομάδα που διάλεξε.

-seePlayer

Η μεγαλύτερη από τις 3. Δίνει στον χρήστη να επιλέξει από ποια ομάδα θέλει να δει κάποιον παίκτη και το πως θα τον ψάξει. Εάν επιλέξει να τον ψάξει με τη θέση που καταλαμβάνει ο παίκτης στον πίνακα listOfPlayers της ομάδας, τα πράγματα είναι απλά, καθώς ξέρουμε πού να κατευθυνθούμε. Ωστόσο, αφού το listOfPlayers είναι ιδιωτικό, δυστυχώς δεν έχουμε πρόσβαση απλά, ώστε να εκτυπώσουμε κατευθείαν τις τιμές. Για αυτό πρέπει να δημιουργήσουμε στην Κλάση ομάδα getters για κάθε μία από τις μεταβλητές που θα έχει ένα αντικείμενο που ανήκει στο listOfPlayers (στην ουσία, τις μεταβλητές του Player), και να τα εκτυπώσουμε ένα ένα. Η επιλογή με όνομα είναι ελαφρώς πιο σύνθετη. Ο χρήστης καταχωρεί το όνομα το οποίο αποθηκεύεται σε μια μεταβλητή τύπου string. Έπειτα, με μια for, ψάχνουμε σε όλα τα αντικείμενα της listOfPlayers της ομάδας που έχει επιλεγεί, πάλι με την βοήθεια της setListOfPlayersName, μέχρι να βρεθεί το όνομα που ταυτίζεται με το στοιχείο που έβαλε ο χρήστης. Αν επιτευχθεί αυτό, εκτυπώνουμε με τον ίδιο τρόπο με πριν τα στοιχεία αυτού του Παίκτη και με μια break βγαίνουμε από τη for. Το πρόγραμμά μας

κάνει άλλη μια στάση και συγκρίνει την τοπική μεταβλητή string που ορίσαμε εμείς με το input του χρήστη. Σε περίπτωση που δεν είναι ίδιο, δηλαδή εάν η for έτρεξε μέχρι τέλους και δεν βρήκε ποτέ το όνομα που έβαλε ο χρήστης, εμφανίζεται μήνυμα λάθους.

Κλάση Player:

Βασικό κομμάτι για την υλοποίηση του project μας, καθώς η κεφαλίδα της βρίσκεται όχι μόνο στην main αλλά και στο αρχείο υλοποίησης της Κλάσης Team. Περιέχει 10 ιδιωτικές μεταβλητές, οι μόνες εκ των οποίων ίσως δεν εξηγούνται ακριβώς από το όνομά τους είναι η `nvic==number of victories` και η `winnerPossibility` που είναι η πιθανότητα να νικήσει ο παίκτης. Στο δημόσιο μέρος συναντάμε τις 2 συναρτήσεις αρχικών συνθηκών και την 1 τελικών. Η δεύτερη συνάρτηση αρχικών συνθηκών δεν έχει όλα τα ορίσματα των μεταβλητών, καθώς ο χρήστης δεν επεμβαίνει σε κάποιες, πχ το ποσοστό κούρασης (fatigue) . Έπειτα ακολουθούν όλοι οι getters και setters, 10+10 στο σύνολο!

Οι επόμενες συναρτήσεις που υλοποιήθηκαν είναι οι δράσεις που έχουν οι παίκτες στο παιχνίδι. Πρώτη η `practice`, που «επεμβαίνει», με την βοήθεια της `setSkills`, στο ποσοστό skills, όταν ο παίκτης εξασκείται, με ποσοστιαία αύξηση. Σε περίπτωση όμως που η μεταβλητή skills ξεπεράσει το 100, της δίνουμε την τιμή 100.

Ακολουθεί η `gettingTired` (λίγο αδόκιμο όνομα αλλά θα το αντέξουμε). Με τη σειρά της, καθορίζει την κούραση που αποκτά ένας παίκτης όταν δουλεύει. Με τη χρήση της `rand`, και μερικών ακόμα.. `trick`, δημιουργήσαμε τυχαία αύξηση από 20 έως 40 %, απόλυτα. Τελικά, το νέο ποσοστό, αφού ελεγχθεί σε σύγκριση με το 100, αντικαθιστά το προηγούμενο με τον αντίστοιχο setter.

Έπειτα, έχουμε την `popular` (όσο και να προσπαθήσαμε, δεν μπορέσαμε να το γυρίσουμε σε ρήμα...) . Αφού δημιουργήσουμε έναν τυχαίο αριθμό από -10 έως 10, τον ελέγχουμε με το 0 και το 100 (όσο και να προσπαθήσει κανείς δεν μπορεί να γίνει τόσο αντιπαθητικός ώστε το ποσοστό του να φτάσει σε αρνητικά επίπεδα). Στη συνέχεια, πάλι αντικαθιστούμε το καινούργιο ποσοστό με το παλιό με τον setter.

Οι Παίκτες έχουν μια κενή μέρα, όπου επιλέγεται τυχαία τι θα κάνουν. Έτσι δημιουργήσαμε την `choose`. Παίρνουμε έναν τυχαίο αριθμό, από το 0 έως το 99. Εάν είναι 50 και πάνω, θεωρούμε πως ο παίκτης κοιμάται, συνεπώς η κούραση του μηδενίζεται. Εάν είναι από 0 έως και 49, θεωρούμε πως εξασκείται, άρα αυξάνεται η τεχνική του. Επειδή όμως αυξάνεται απόλυτα, δεν μπορούμε να χρησιμοποιήσουμε την `practice`, καθώς εκείνη ρυθμίζει ποσοστιαία αύξηση!

Το `highlight` του παιχνιδιού (ίσως μετά τις αποχωρήσεις), οι διαγωνισμοί σαφώς και αποτελούν μια δράση των παικτών. Έτσι έχουμε την `compete`, που φροντίζει να «κουράσει» τους παίκτες που συμμετέχουν σε ένα τυχαίο ποσοστό 10 με 20%.

Τέλος φτιάξαμε και μια status, που εκτυπώνει όλες τις μεταβλητές του αντικειμένου!!

Κλάση Team:

Εξίσου αναγκαία για το πρόγραμμά μας, η Κλάση Team. Αποτελείται από 4 ιδιωτικές μεταβλητές, το χρώμα της, τον αριθμό των νικών που έχουν κάνει οι παίκτες της στους διαγωνισμούς, την πιθανότητα που έχει να νικήσει το παιχνίδι, και έναν πίνακα 11 στοιχείων, ένα για κάθε παίκτη. Ο πίνακας αυτός είναι τύπου Player, δηλαδή τα στοιχεία του θα αποτελούν αντικείμενα της Κλάσης Player. Στα δημόσια μέρη, ξεκινάμε με τους 2 constructors. Από το χρήστη ζητείται σε πρώτο στάδιο να δηλώσει μόνο το χρώμα της ομάδας. Οι υπόλοιπες μεταβλητές ορίζονται από το πρόγραμμα, σύμφωνα με τα ζητούμενα, πχ το ποσοστό της τεχνικής κατάρτισης ένας τυχαίος αριθμός μέχρι το 80. Αρχικοποίηση για τη listOfPlayers δε χρειάζεται. Με το που καλείται, δημιουργούνται 11 αντικείμενα Player, τα οποία όμως αρχικοποιούνται από τους constructor της Κλάσης Players. Ακολουθεί ο destructor και οι set και get για τις μεταβλητές. Ενδιαφέρον παρουσιάζουν οι συναρτήσεις που ασχολούνται με την listOfPlayers. Η set δεν δέχεται ορίσματα, αλλά ζητά από το χρήστη κατά τη διάρκεια να εισάγει τα κατάλληλα ορίσματα για να καλύψει τις μεταβλητές που έχουν τα αντικείμενα τύπου Player. Για τον λόγο που αναφέρθηκε στη συνάρτηση main, δημιουργούμε ξεχωριστά για κάθε παράμετρο του αντικειμένου και μία get. Έτσι έχουμε πολλές συναρτήσεις με ονόματα getListOfPlayersΚάτι, ανάλογα με τις μεταβλητές του Player. Τέλος, κάναμε μια status, που χρησιμοποιείται και στην main, ώστε να προβάλλουμε όλα τα στοιχεία μιας Ομάδας.