

README

Στο συγκεκριμένο project επιχείρησα να υλοποιήσω τα ζητούμενα με τρόπο τέτοιο ώστε να πληρούνται οι προϋποθέσεις για μια σωστή εργασία και παράλληλα το παιχνίδι να είναι όσο το δυνατόν ρεαλιστικό. Επιχείρησα δηλαδή να ισορροπήσω τις αναλογίες στα χαρακτηριστικά ηρώων, τεράτων και αντικειμένων, ώστε το παιχνίδι να μην γίνεται τελικά υπερβολικά δύσκολο ή υπερβολικά εύκολο. Έκανα λοιπόν τις εξής παραδοχές:

A) Κάθε ήρωας μπορεί να κουβαλά μαζί του ως και 5 αντικείμενα του κάθε είδους.

B) Η εμπειρία που συγκεντρώνει μετά από κάθε νίκη αντιστοιχεί με το 20πλάσιο του αριθμού των τεράτων που πολέμησε και του επιπέδου του ήρωα (πρέπει η εμπειρία να ξεπερνάει το 100πλάσιο του επιπέδου ώστε ο ήρωας να ανεβεί επίπεδο), ενώ τα χρήματα που λαμβάνει αντιστοιχούν στο 5πλάσιο.

Γ) Ο χάρτης αποτελείται από έναν πίνακα 5x10, τα άκρα του οποίου αποτελούν τα σύνορα του χάρτη, ενώ μέσα στον χάρτη εκτός από τα κοινά, υπάρχουν και τετράγωνα αγοράς για να βοηθήσουν τον παίκτη στο ταξίδι του.

Με την είσοδο του παίκτη σε κοινό τετράγωνο υπάρχει 1/5 περιπτώσεις να ξεκινήσει μια μάχη με 1 ως 5 τέρατα, τα οποία έχουν επιλεγθεί τυχαία από τα 30 που βρίσκονται στον χάρτη, επιπέδου ίσου με τον μέσο όρο των επιπέδων των ηρώων. Επίσης, κάθε τετράγωνο αγοράς έχει 5 αντικείμενα κάθε είδους, τα οποία έχουν επιλεγθεί τυχαία από 10 που βρίσκονται συνολικά στον χάρτη.

Όσον αφορά την μεθοδολογία:

Η αναπαράσταση του χάρτη έγινε ως εξής:

Στην κλάση grid που αναπαριστά τον χάρτη υπάρχει ένας πίνακας char 5x10 με x όπου υπάρχει μη προσβάσιμο τετράγωνο, c για κάθε κοινό

τετράγωνο `m` για τα τετράγωνα αγορών και `s` για το σημείο εκκίνησης. Ο constructor του `grid` διαβάζει τον πίνακα και δημιουργεί έναν νέο τύπου `tile` και ίδιων διαστάσεων, όπου αντιστοιχίζει κάθε χαρακτήρα με τον αντίστοιχο τύπο τετραγώνου.

Η αναπαράσταση των αντικειμένων έγινε ως εξής:

Στην κλάση `grid` υπάρχουν 3 πίνακες `monsternames`, `itemnames`, `spellnames` με 30 θέσεις ο καθένας, στους οποίους έχουν αποθηκευτεί τα ονόματα τεράτων αντικειμένων και ξορκιών που πήρα από το `lists`. Ο constructor του `grid` διαβάζει τους πίνακες `itemnames`, `spellnames` και δημιουργεί 6 πίνακες 10 θέσεων. Ο πίνακας `weapons` τύπου `weapon` παίρνει τα 10 πρώτα στοιχεία του πίνακα `itemnames` και δημιουργεί 10 `weapons` με τα ονόματα αυτά. Με αντίστοιχο τρόπο λειτουργούν και οι υπόλοιποι 5 πίνακες.

Κάθε φορά που ο constructor του `grid` δημιουργεί ένα `common` τετράγωνο περνά με τυχαία επιλογή 15 ονόματα τεράτων μέσα σε πίνακα `string`, τα οποία παίρνει από τον πίνακα `monsternames` καθώς και τις 2 μεταβλητές που συμβολίζουν τις συντεταγμένες του εκάστοτε τετραγώνου. Κάθε φορά που δημιουργεί ένα τετράγωνο αγοράς επιλέγει τυχαία 5 αντικείμενα κάθε είδους και περνά πίνακες με αυτά ως ορίσματα στις αγορές. Για τα ξόρκια δημιουργείται ένας πίνακας 15 θέσεων, όπου στις 5 πρώτες θέσεις μπαίνουν `icespells` στις επόμενες 5 `firespells` και τέλος `lightningspells`. Οι 4 αυτοί πίνακες περνάνε ως ορίσματα στον constructor της αγοράς μαζί με τις συντεταγμένες του συγκεκριμένου τετραγώνου.

Όσον αφορά την κίνηση του παίκτη μέσα στον χάρτη:

Ο παίκτης με τα πλήκτρα WASD μπορεί να περιηγείται στον δισδιάστατο χάρτη και πατώντας τους υπόλοιπους χαρακτήρες που εμφανίζονται στο μενού επιτελούνται οι υπόλοιπες λειτουργίες που περιγράφονται στην εκφώνηση.

Ροή του παιχνιδιού:

Αρχικά δημιουργείται ο χάρτης, τα τετράγωνα και τα αντικείμενά του και καλείται η συνάρτηση `startgame`, η οποία καλωσορίζει τον παίκτη

και δημιουργεί την ομάδα του ανάλογα με τον προτεινόμενο αριθμό ηρώων, τα ονόματά και τις ιδιότητές τους. Η ομάδα αποθηκεύεται στο grid και καλείται η traverse με σκοπό ο παίκτης να περιηγηθεί στον χάρτη. Ξεκινώντας από το σημείο 2,1 του χάρτη (0,1 του πίνακα) ο παίκτης μέσω των κατάλληλων πλήκτρων διασχίζει τον χάρτη. Κάθε φορά που οι συντεταγμένες της ομάδας αλλάζουν καλείται η pure virtual συνάρτηση Enter για το εκάστοτε είδος τετραγώνου, η οποία επιστρέφει true αν η ομάδα μπήκε εν τέλη στο τετράγωνο. Αν η ομάδα εισέλθει σε common τετράγωνο, τότε η enter θα κάνει μια rand το αποτέλεσμα της οποίας θα κρίνει αν θα ξεκινήσει μάχη σε αυτό το τετράγωνο(20% πιθανότητα). Στην περίπτωση που αυτό συμβεί, δημιουργείται μέσω της createmonsterteam μια ομάδα τεράτων στον επίπεδο των ηρώων με 1 έως 5 τέρατα και ξεκινά η μάχη. Αν η ομάδα εισέλθει σε τετράγωνο αγοράς τότε η enter εμφανίζει μενού με επιλογές αγοράς και πώλησης αντικειμένων για κάθε ήρωα με ακριβείς περιγραφές των αντικειμένων προς πώληση.

Όταν ξεκινά μια μάχη:

Όσο οι 2 αντίπαλες ομάδες έχουν έστω και ένα μέλος με υγεία μεγαλύτερη του 0 η μάχη συνεχίζεται σε γύρους. Ο παίκτης μέσω του μενού μάχης επιλέγει τα κατάλληλα πλήκτρα για να επιτεθεί/κάνει ξόρκι σε συγκεκριμένο τέρας με συγκεκριμένο ήρωα ή να χρησιμοποιήσει κάποιο αντικείμενο. Αν αυτή η ενέργεια δεν μπορεί να επιτελεστεί (γιατί ο παίκτης δεν έχει το απαιτούμενο αντικείμενο ή το τέρας που ο παίκτης επέλεξε έχει ήδη ηττηθεί) ή αν επιλεχθεί η εμφάνιση των στοιχείων των χαρακτήρων, τότε ο παίκτης έχει την ευκαιρία να επιλέξει ξανά την ενέργεια του ήρωα. Αφού όλοι οι μη ηττημένοι ήρωες έχουν κάνει μια από τις επιλεγόμενες πράξεις έρχεται η σειρά των τεράτων που είναι ακόμα στην μάχη να επιτεθούν τυχαία στους ήρωες. Στο τέλος κάθε γύρου αυξάνεται η υγεία όλων των όντων και η μαγική δύναμη των ηρώων. Όταν τελικά η μάχη τελειώσει ανεβαίνει στο μισό του κανονικού η υγεία όσων ηρώων η υγεία τελείωσε, αν οι ήρωες έχουν ηττηθεί χάνουν τα μισά χρήματά τους, ενώ αν νίκησαν κερδίζουν τον μισθό και την εμπειρία που τους αναλογεί.

Εν κατακλείδι, επιχείρησα χρησιμοποιώντας τα προγραμματιστικά εργαλεία που διδάχτηκαν να υλοποιήσω όσο απλούστερα μπορώ ένα rpg παιχνίδι όπως περιγράφεται στην εκφώνηση. Έτσι, ο κωδικός μου περιέχει μεγάλο αριθμό getters και setters, το οποίο ελπίζω ότι δεν είναι πρόβλημα. Ακόμη, δεν βρήκα τρόπο να διαφοροποιήσω αρκετά τις rand έτσι ώστε να μην υπάρξουν τα ίδια αποτελέσματα κάθε φορά που τρέχει το πρόγραμμα και παράλληλα όταν πολλές rand τρέχουν στο ίδιο δευτερόλεπτο να μην επιστρέφουν το ίδιο αποτέλεσμα, οπότε επέλεξα να συμβαίνει το δεύτερο. Σε γενικές όμως γραμμές προσπάθησα να υλοποιήσω τις κλάσεις με τον αποδοτικότερο δυνατό τρόπο και την αναπαράσταση των δεδομένων με τον απλούστερο τρόπο.