Δημήτρης Κελέσης A.M.: 03115037

Γενικός σχεδιασμός Συστήματος

Η υλοποίηση της εργασίας στηρίζεται σε κάποιες βασικές κλάσεις που υλοποιούν και ενέχουν τις αντίστοιχες λειτουργικότητες, πέραν των βασικών για set και get. Συγκεκριμένα αρχικά δημιουργήθηκε η κλάση Position που περιέχει την θέση κάθε παίκτη και μία μέθοδο για σύγκριση πιθανής ισότητας δύο θέσεων εξισώνοντας τις αντίστοιχες μεταβλητές θέσεις. Στη συνεχεία δημιουργήθηκε η βασική κλάση Player στην οποία αποθηκεύονται όλα τα δεδομένα που είναι σχετικά με τον παίκτη που μετέχει στο παιχνίδι. Σε αυτήν την κλάση υλοποιείται και η συνάρτηση της κίνησης για κάθε παίκτη. Συγκεκριμένα διαβάζεται σε μορφή πίνακα το ταμπλό και από αυτό τοποθετούνται σε πίνακα οι έγκυρες θέσεις στις οποίες μπορεί να βρεθεί κάποιος παίκτης, οι οποίες είναι όλες οι θέσεις που περιέχουν γράμματα εκτός από αυτές που περιέχουν το F. Έτσι κάθε φορά που ρίχνουμε το ζάρι τροφοδοτούμε την συνάρτηση με τις κατάλληλες μεταβλητές και διασχίζοντας κυκλικά τον πίνακα βρίσκουμε την μελλοντική μας θέση. Τέλος πραγματοποιούμε δύο έξτρα ελέγχους για πιθανή ταύτιση της μελλοντικής θέσης κάποιου παίκτη με τις θέσεις των άλλων δύο παικτών στο ταμπλό. Σε περίπτωση που ισχύει κάτι τέτοιο ο παίκτης μετακινείται είτε στην αμέσως επόμενη είτε στην αμέσως προηγούμενη θέση διασφαλίζοντας έτσι την σωστή μετακίνηση και την αποφυγή ύπαρξης δύο παικτών στο ίδιο τετράγωνο. Υλοποιούνται δύο παρόμοιες συναρτήσεις κίνησης, μία που δίνει την ακριβή επόμενη θέση και μία άλλη που δίνει σε μορφή λίστας όλες τις διαδοχικές θέσεις που θα διασχίσει μέχρι την τελική θέση. Από την δεύτερη συνάρτηση το αποτέλεσμα της τροφοδοτεί τους ελέγχους σχετικά με το αν επιτρέπεται να πάρει χρήματα από την τράπεζα και να χτίσει εισόδους αναλόγως αν στην διαδρομή του πέρασε από το δημαρχείο ή την τράπεζα. Επίσης υλοποιείται η συνάρτηση πληρωμής από έναν παίκτη σε κάποιον άλλο. Για να γίνει αυτό δίδονται τα κατάλληλα ορίσματα και τροποποιούνται κατάλληλα τα χρήματα στους δύο μετέχοντες ελέγχοντας παράλληλα για πιθανή χρεοκοπία κάποιου εξ αυτών. Αν υπάρξει χρεοκοπία επιστρέφεται κατάλληλη αληθοτιμή ώστε να αφαιρεθεί από το παιχνίδι και πραγματοποιούνται οι ζητούμενες ενέργειες σχετικά με τα ξενοδοχεία και τις πύλες που έχει στην κατοχή του. Κατόπιν δημιουργήθηκε η κλάση Hotel με σκοπό την αποθήκευση όλων των πληροφοριών σχετικά με τα ξενοδοχεία που υπάρχουν στο ταμπλό και μία συνάρτηση κατάλληλης εκτύπωσης αυτών των στοιχείων, πέραν των βασικών χαρακτηριστικών τους προστέθηκε πεδίο για την αποθήκευση του επιπέδου αναβάθμισης του κάθε ενός από αυτά. Για την υλοποίηση του ζαριού δημιουργήθηκε η κλάση Dice με την μέθοδο σχετικά με την δίκαιη ρίψη να υλοποιείται όπως αυτή βρέθηκε σε βιβλιογραφία στο διαδίκτυο και την μέθοδο για την μη δίκαιη ρίψη που αποφασίζει σχετικά με το κόστος κάποιας επέκτασης να υλοποιείται από τον γράφοντα. Για την ύπαρξη ρολογιού στην διεπαφή που θα ξεκινά να μετρά από την έναρξη της εφαρμογής μέχρι το κλείσιμο αυτής ή μέχρι το σταμάτημα της δημιουργήθηκε η κλάση LiveDateSwing που υλοποιεί το χρονόμετρο μέσω της διαφοράς μεταξύ της τρέχουσας ώρας και της αρχικής ώρας εκκίνησης του παιχνιδιού. Για να μπορέσει να γίνει αυτό και να μπορεί να τρέχει ανεξάρτητα από το υπόλοιπο πρόγραμμα χρησιμοποιείται νέο thread το οποίο εκκινείται από το κυρίως πρόγραμμα κατά την έναρξη αυτού και σταματάει μετά την λήξη του είτε από το ει-

δικό κουμπί στο menu Stop. Η μία από τις δύο βασικές κλάσεις της εφαρμογής είναι η κλάση Board που ουσιαστικά υλοποιεί το ταμπλό και τις ενέργειες για κάθε ειδικά διαμορφωμένο κουτάκι που βρίσκεται σε αυτό. Αρχικά το ταμπλό διαβάζεται από το κατάλληλο αρχείο εισόδου και δημιουργείται ο αντίστοιχος πίνακας πληροφορίας. Επίσης δημιουργούνται πίνακες σχετικά με την τοποθεσία των εισόδων, τους ιδιοκτήτες των ξενοδοχείων και το που δείχνουν οι είσοδοι σε κάποιοι τετράγωνο. Οι παραπάνω μεταβλητές στους πίνακες ανανεώνονται μέσω των κατάλληλων ενεργειών. Υλοποιείται η μέθοδος για εύρεση της θέσης της αφετηρίας από την οποία θα ξεκινήσουν όλοι οι παίκτες καθώς και μέθοδοι που ελέγχουν αν κάποια διαδρομή που δίδεται ως είσοδος περνά είτε από το δημαρχείο είτε από την τράπεζα. Ταυτόχρονα υλοποιούνται κάποιες βασικές μέθοδοι που ενεργοποιούνται όταν βρίσκεται κάποιος παίκτης είτε σε κουτί H είτε σε κουτί C είτε σε κουτί E. Στο κουτί E ενεργοποιούνται σύμφωνα με το παιχνίδι τόσο η ενέργεια για το κουτί C δηλαδή αγορά εισόδου όσο και η ενέργεια για το κουτί E δηλαδή η επέκταση υπάρχοντος ξενοδοχείου. Συγκεκριμένα όταν βρίσκεται κάποιος σε κουτί E ενεργοποιείται η επέκταση και η αγορά εισόδου. Η εφαρμογή ενημερώνει για την επέκταση περιμένει η εφαρμογή από τον χρήση να πληκτρολογήσει το id του ξενοδοχείου που επιθυμεί να αναβαθμίσει από τα διαθέσιμα που έχει στην κατοχή του. Σε περίπτωσή που δεν δοθεί έγκυρο ξενοδοχείο, δηλαδή ξενοδοχείο που δεν είναι στην κατοχή του συγκεκριμένου παίκτη τότε αυτός ενημερώνεται με κατάλληλο μήνυμά λάθους για μη έγκυρο ξενοδοχείο. Αν πληκτρολογηθεί έγκυρο ξενοδοχείο προς αναβάθμιση τότε το πρόγραμμα ελέγχει το επίπεδο αναβάθμισης του και συγκεκριμένα αν το τρέχον ενοίκιο του ξενοδοχείου είναι ίδιο με το μέγιστο δυνατό κόστος που προκύπτει αν το έχουμε κτίσει πλήρως, δηλαδή και εξωτερικό χώρο. Αν ισχύει κάτι τέτοιο τότε δεν γίνεται περαιτέρω αναβάθμιση καθώς αυτό είναι αδύνατο. Σε αντίθετη περίπτωση ρίχνεται εσωτερικά από την εφαρμογή ένα μη δίκαιο ζάρι που υλοποιεί την πιθανότητα σχετικά με τις κτίσεις όπως αυτή δίδεται από τις προδιαγραφές. Αν απορριφθεί τότε δεν γίνεται τίποτα σε αντίθετη περίπτωση αναπροσαρμόζεται ή μη το κόστος για την αναβάθμιση του ξενοδοχείου και στη συνέχεια αφαιρούνται τα αντίστοιχα χρήματα από τον παίκτη και ταυτόχρονα ενημερώνεται το νέο επίπεδο και ενοίκιο για το ξενοδοχείο. Όταν βρίσκεται σε κουτί H τότε ο αλγόριθμος ελέγχει αν το/α ξενοδοχείο/α είναι αγορασμένα από κάποιον άλλο παίκτη και αν ναι τότε αν έχουν επεκταθεί έστω και σε κάποιο βαθμό. Στην δεύτερη περίπτωση δεν το θεωρεί διαθέσιμο προς αγορά ξενοδοχείο και δεν το εμφανίζει αλλιώς εμφανίζει σαν δυνατή επιλογή. Ο χρήστης συμπληρώνει στο αντίστοιχο πεδίο το id του ξενοδοχείου και γίνεται η πληρωμή, είτε μεταξύ παίκτη και τράπεζα είτε μεταξύ παίκτη και παίκτη ανανεώνοντας τα χρήματα του καθενός. Σε περίπτωση λάθους εισαγωγής εμφανίζεται ανάλογο μήνυμα. Τέλος αν υπάρχει στην διαδρομή η τράπεζα ή το δημαρχείο ενεργοποιούνται οι αντίστοιχες λειτουργικότητες για απαίτηση χρημάτων και για κτίση εισόδου αντίστοιχα.
Η βασική κλάση που υλοποιεί τα γραφικά είναι η κλάση kati στην οποία ουσιαστικά πραγματοποιείται το παιχνίδι. Αρχικά διαβάζεται από κάποιο αρχείο το ταμπλό και οι πληροφορίες σχετικά με τα ξενοδοχεία. Αφού αρχικοποιηθούν αυτά δημιουργείται το ζάρι του παιχνιδιού και το ταμπλό φορτώνεται σε ένα gridPane. Βρίσκουμε την αφετηρία στην οποία τοποθετούμε τους παίκτες και στη συνέχεια κυκλικά εναλλάσσουμε κυκλικά τους παίκτες. Για τον συμβολισμό των παικτών στο ταμπλό το οποίο αναπαρίσταται όπως μας δίδεται στο αντίστοιχα αρχείο εισόδου χρησιμοποιούμε κατάλληλα χρωματισμένες μεταβλητές P1, P2, P3 που αντιστοιχούν στους παίκτες και αυτές μετακινούνται στο ταμπλό. Κάθε φορά που κάποιος παίκτης βρίσκεται σε ένα κουτί τότε το σύμβολο που αρχικά βρισκόταν εκεί εξαφανίζεται και αντικαθίσταται από το σύμβολο του αντίστοιχου παίκτη και όταν αυτός φύγει από του κουτί τότε αποκαθίσταται. Στα κουτιά κειμένου που βρίσκονται πάνω από τα αντίστοιχα κουμπιά εισόδου γράφει ο χρήστης το id του ξενοδοχείου που είναι στην κατοχή του και επιθυμεί να ενεργήσει πάνω του. Μόλις πατήσει το κουμπί επιβεβαίωσης τότε πραγματοποιείται η αντίσ-

τοιχη ενέργεια και εμφανίζεται ενημερωτικό μήνυμα στους παίκτες. Κάποια από τα κουμπιά είναι μίας χρήσης όπως π.χ. αυτό για την απαίτηση χρημάτων ώστε αν αποφασίσει να απαιτήσει χρήματα ο παίκτης περνώντας από την τράπεζα να το κάνει μόνο μία φορά. Κουμπιά όπως η αναβάθμιση ενός ξενοδοχείου δεν κλειδώνουν μετά το πρώτο πάτημα ώστε σε περίπτωση που πατηθεί κατά λάθος μη έγκυρο ξενοδοχείο να μπορεί ο παίκτης να αξιοποιήσει την σειρά του. Για κάθε επίπεδο αναβάθμισης ένα από τα αντίστοιχα κουτιά που συμβολίζουν το εκάστοτε ξενοδοχείο χρωματίζονται φούξια, ενώ για την αγορά κάποιας εισόδου το κουτί που αντιστοιχεί στην τυχαία επιλεγμένη είσοδο αποκτά background μπλε.
Η υλοποίηση του ρολογιού γίνεται αυτόματα με χρήση νέου thread που ξεκινά με την έναρξη της εφαρμογής σταματά μετά την λήξη της είτε με το πάτημα του κουμπιού stop.
Τέλος το menu υλοποιεί τις ζητούμενες λειτουργικότητες έχοντας σετάρει κατάλληλα τα menuitems. Αξίζει να αναφερθεί πως για να εμφανιστούν οι πληροφορίες για ένα συγκεκριμένο ξενοδοχείο πατάει ο χρήστης το id σε αντίστοιχο textfield και οι πληροφορίες εμφανίζονται με το πάτημα του κουμπιού υποβολής. Πέραν αυτού του κουμπιού τα υπόλοιπα περιέχουν ενημερωτικό και μη διαδραστικό με τον χρήστη περιεχόμενο. Επίσης σαν παραδοχή έχουμε θεωρήσει ότι τα id όλων των ξενοδοχείων δεν ξεπερνούν το id = 17, αν επιθυμούμε να το αλλάξουμε αυτό χρειάζεται μικρή τροποποίηση κάποιων γραμμών κώδικα σε ορισμένες αρχικοποιήσεις μεγέθους ArrayList. Η μοναδική λειτουργικότητα που δεν έχει υλοποιηθεί είναι αυτή του κουμπιού start στο menu γιατί η υλοποίηση ξεκίνησε με την δημιουργία των υπολοίπων και στο τέλος του menu και στην τελική έκδοση και στον τρόπο διαμόρφωσης αυτής ήταν αδύνατη η υλοποίηση. Δυνητικά αν θα μπορούσαμε να επανεκκινήσουμε την εφαρμογή θα έπρεπε απλά να αλλαχθεί η τιμή μίας μεταβλητής που υποδηλώνει τον φάκελο προέλευσης.