

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ &
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
-Εξάμηνο 7ο

ΤΕΧΝΟΛΟΓΙΑ ΠΟΛΥΜΕΣΩΝ -
ΑΝΑΦΟΡΑ ΓΙΑ ΤΗΝ ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ
ΤΟΥ ΜΑΘΗΜΑΤΟΣ

Βερέμης Αλέξανδρος

ΑΜ: 03115063

Αρχικά, θα ήθελα να αναφέρω ότι συμπεριλήφθηκαν όλες οι ζητούμενες λειτουργικότητες. Πιο συγκεκριμένα, οι χώροι στάθμευσης και οι πτήσεις περιέχουν και χρησιμοποιούνται με βάση τα αναγνωριστικά στοιχεία της εκφώνησης.

Στο φάκελο medialab μπορούμε να βάλουμε διάφορα text-σενάρια για airport και setup, τα οποία μπορούμε να φορτώσουμε μέσω του load και να τα τρέξουμε μέσω του start.

5 sec πραγματικού χρόνου αντιστοιχούν σε 1 min χρόνου της εφαρμογής.

Στο κεντρικά δεξιό μέρος έχουμε συμπεριλάβει τα αιτήματα προσγείωσης και εξυπηρέτησης με την απάντηση να δίνεται μέσω ενός pop-up παραθύρου.

Για την επιλογή χώρου στάθμευσης, στην περίπτωση που υπάρχουν πολλαπλές κατηγορίες ικανές να εξυπηρετήσουν, ως κριτήριο έχουμε θέσει να μπαίνει στην κατηγορία με τον μικρότερο αριθμό Κατηγορίας, δηλαδή στην πρώτη που είναι εύκαιρη.

Ο ακριβής χρόνος αναχώρησης μιας πτήσης επιλέγεται τυχαία από 0 έως το διπλάσιο του αρχικού χρόνου αναχώρησης και έχουμε συμπεριλάβει τις κατάλληλες εκπτώσεις σε περίπτωση που φύγει νωρίτερα του αρχικού χρόνου. Ενώ χρεώνουμε διπλά, αν φύγει αργότερα.

Μετά από κάθε αναχώρηση, πραγματοποιείται διαγραφή της πτήσης, ενώ ελέγχουμε αν μπορούν οι πτήσεις που είναι on hold να μπουν σε κάποια θέση.

Το GUI θα έλεγα ότι είναι σχετικά απλό και εύκολο για το χρήστη.

Για τη γραφική διεπαφή χρήστη χρησιμοποίησα JavaFX και παρακάτω στο screenshot μπορείτε να δείτε ότι υπάρχουν όλα όσα ζητήθηκαν για το GUI.

Όσον αφορά τις λοιπές απαιτήσεις, ακολουθήθηκαν πιστά οι αρχές σχεδίασης του αντικειμενοστραφούς προγραμματισμού. Ενώ, για το javadoc χρησιμοποιήθηκε η κλάση AirportState.java !

Κάποιες επιπρόσθετες σημαντικές πληροφορίες όσον αφορά την δική μου υλοποίηση:

1) Χρησιμοποίησα SceneBuilder για το GUI του JavaFX, το οποίο με διευκόλυνε πάρα πολύ γιατί έγραφε κώδικα σε fxml ενώ εγώ απλά επέλεγα τα κατάλληλα components.

2) Υπάρχει μια διαφοροποίηση σε σχέση με την εκφώνηση και τη χρήση που κάνω εγώ στα MenuItems Start και Load. Δηλαδή, στη δική μου υλοποίηση το Start κάνει ένα κομμάτι της δουλειάς του Load καθώς η αναζήτηση και το μήνυμα σφάλματος σε περίπτωση μη ύπαρξης κάποιου σεναρίου εμφανίζεται όταν πατήσουμε το Start. Με το Load απλά προωθούμε ποιο σενάριο θέλουμε να τρέξουμε.

3) Στο message component του κάτω μέρους του GUI, κάθε 5 λεπτά έχουμε βάλει να σβήνονται όλα τα μηνύματα για να μην υπάρξει overflow.

4) Η απεικόνιση των διαθέσιμων χώρων στάθμευσης έγινε με τη χρήση Pie Charts και επίσης όταν κάποιος χώρος στάθμευσης δεν υπάρχει στο σενάριο τον έχουμε κάνει να είναι invisible.

5) Στο δεξιό κεντρικό μέρος για να γίνει εισαγωγή κάποιας πτήσης που θέλει να προσγειωθεί, πρέπει όλα τα στοιχεία που εισάγονται να είναι ορθά (να συμβαδίζουν με τον τύπο τους), αλλιώς εμφανίζεται κατάλληλο μήνυμα σφάλματος στο message component που βρίσκεται στο κάτω μέρος.

6) Τέλος, χρησιμοποιήθηκαν αγγλικοί χαρακτήρες για να μην υπάρξουν προβλήματα συμβατότητας, περισσότερο για μια επιπρόσθετη ασφάλεια, παρόλου που η Java υποστηρίζει ελληνικούς χαρακτήρες.

Screenshot της εφαρμογής εν ώρα λειτουργίας:

