

Αναφορά τελικής εργασίας στο μάθημα “Τεχνολογία Πολυμέσων”

Ονοματεπώνυμο: Δημήτρης Δεσποτίδης
ΑΜ: el17066

Στην εργασία αυτή υλοποιήσαμε μία παραλλαγή του κλασικού παιχνιδιού “Κρεμάλα” σε γλώσσα προγραμματισμού Java.

Η δομή της εφαρμογής όπως υλοποιήθηκε είναι η παρακάτω:

- **Hangman:** η αρχική κλάση της εφαρμογής, η οποία αρχικοποιεί τον HangmanController.
- **HangmanController:** ο βασικός controller της εφαρμογής. Φτιάχνει την αρχική γραφική διεπαφή και αρχικοποιεί ένα αντικείμενο Game. Επίσης είναι υπεύθυνο για την επικοινωνία της γραφικής διεπαφής με τις κλάσεις που υλοποιούν τις λειτουργικότητες του παιχνιδιού, καθώς και για την ενημέρωση της γραφικής διεπαφής.
- **Game:** είναι υπεύθυνο για όλη την διαχείριση του παιχνιδιού, από την έναρξη μέχρι το κλείσιμο της εφαρμογής. Με το πάτημα του menu *Start*, αρχικοποιεί ένα αντικείμενο της κλάσης Round, με μία τυχαία λέξη από το επιλεγμένο λεξικό. Επίσης υλοποιεί τις συναρτήσεις της δημιουργίας νέου λεξικού και της ενημέρωσης του log file.
- **Round:** αναλαμβάνει την διαχείριση ενός συγκεκριμένου γύρου, από την έναρξή του με το πάτημα του *Start*, έως το τέλος του, είτε από νίκη του παίκτη, είτε από ήττα, είναι από παράδοση (menu *Solution*).

Αυτές είναι οι βασικές κλάσεις της εφαρμογής. Υπάρχουν έπειτα μερικές δευτερεύουσες κλάσεις που καλούνται από τις παραπάνω και υλοποιούν κάποιες εξειδικευμένες λειτουργίες:

- **IDtoDictionary:** αναλαμβάνει όλη την διαδικασία δημιουργίας ενός νέου λεξικού, χρησιμοποιώντας τις βοηθητικές κλάσεις **FetchURL** και **DictionaryCreator**.
- **[menu name]Controller:** αναλαμβάνει την εμφάνιση και τον έλεγχο του αντίστοιχου pop-up παραθύρου, μετά από το πάτημα του κατάλληλου menu item. Δημιουργούνται από τον HangmanController.
- **[exception name]Exception:** οι ορισμοί των διάφορων custom-made εξαιρέσεων.
- **LetterCandidates, LetterPair, MyPair, logData:** χρήσιμες δομές δεδομένων

Παραδοχές

Κατά την αρχικοποίηση του παιχνιδιού, δίνω μία default τιμή στο επιλεγμένο λεξικό, οπότε το παιχνίδι μπορεί να αρχίσει με απευθείας πάτημα του *Start*, χωρίς να έχει προηγηθεί *Load*.

Ως προς τις εξαιρέσεις, δεν υλοποιήθηκαν καθόλου οι *InvalidCountException* και *InvalidRangeException*, καθώς κατά την δημιουργία του λεξικού δεν βάζω ποτέ την ίδια λέξη δύο φορές, και απορρίπτω τις λέξεις με λιγότερα από 6 γράμματα. Αυτό το έκανα γιατί αλλιώς αυτές οι δύο εξαιρέσεις θα εγείρονταν συνεχώς.

Οι άλλες δύο εξαιρέσεις υλοποιούνται, και εμφανίζεται pop-up alert όταν γίνεται απόπειρα δημιουργίας λεξικού που τις εγείρει.

Παρόμοιο pop-up εμφανίζεται και όταν δίνεται άκυρη είσοδος στα πεδία *Position* και *Letter* της γραφικής διεπαφής.

Επίσης θεωρώ ότι ο χρήστης δεν θα εισάγει κάποιο γράμμα που δεν ανήκει στη λίστα των πιθανοτήτων. Αν το κάνει, απλά αγνοώ την εν λόγω υποβολή.

Επίσης για λόγους ευκολίας του χειρισμού των λεξικών και υπολογισμού των στατιστικών (για το menu *Dictionary*), στις πρώτες δύο γραμμές του κάθε αρχείου-λεξικού αποθηκεύω (α) το συνολικό πλήθος των λέξεων και (β) τα πλήθος των λέξεων με 6, 7-9 και 10+ γράμματα αντίστοιχα.

Τέλος, το ιστορικό των 5 τελευταίων παιχνιδιών αποφάσισα να το υλοποιήσω με ένα log file (*medialab/log.txt*), το οποίο περιέχει ουσιαστικά το πλήρες ιστορικό παιχνιδιών και το αποθηκεύω στο ίδιο directory με τα λεξικά.