



Μέρος (α): Δημιουργία Λεξικού

Για αυτό το μέρος της εφαρμογής σχεδιάσαμε τη λειτουργία "Create". Τρέχουμε την Main και ανοίγει η εφαρμογή, το main page της οποίας φαίνεται στην πρώτη εικόνα.

Επιλέγουμε Application → Create και ανοίγει ένα νέο fxm1 page στο οποίο εισάγει ο χρήστης τα Dictionary ID και OPEN LIBRARY ID. Ο κώδικας στο backend "χτυπάει" το αντίστοιχο URL και επιστρέφονται τα δεδομένα της περιγραφής του επιλεγμένου βιβλίου. Φροντίζουμε να κρατήσουμε κάθε λέξη της περιγραφής μόνο μία φορά (για αυτό χρησιμοποιήσαμε ένα Set) και αφαιρέσαμε όλες τις λέξεις με λιγότερα από 6 γράμματα. Εφόσον οι παραπάνω έλεγχοι γίνονται πάντα, έχουμε εξασφαλίσει ότι το λεξικό μας θα περιέχει σίγουρα μοναδικές λέξεις άνω των 6 γραμμάτων, και άρα δεν χρειάζεται να υλοποιήσουμε τις ζητούμενες εξαιρέσεις InvalidCountException και InvalidRangeException. Μετά από τους ελέγχους αυτούς, εξετάζουμε αν οι εναπομείνουσες λέξεις είναι άνω των 20 και τουλάχιστον το 20% εξ αυτών αποτελείται από 9 ή περισσότερα γράμματα. Αν η πρώτη συνθήκη δεν πληρούται τότε προκύπτει UndersizeException, κι αν η δεύτερη συνθήκη δεν πληρούται τότε προκύπτει UnbalancedException. Το λεξικό, αν πληροί τις άνω προδιαγραφές, αποθηκεύεται στον φάκελο me-dialab, διαφορετικά προκύπτει η αντίστοιχη εξαίρεση και κάποιο μήνυμα σφάλματος στον χρήστη.

Επιπλέον έχουμε υλοποιήσει τις παρακάτω εξαιρέσεις: NoDescriptionException για όταν το επιλεγμένο βιβλίο του OPEN LIBRARY δεν περιέχει περιγραφή και, UnfilledTextException για όταν ο χρήστης κλικάρει σε κάποιο κουμπί χωρίς να συμπληρώσει τα απαραίτητα πεδία πρώτα. Η δεύτερη αυτή εξαίρεση προκύπτει σε κάθε αντίστοιχη περίπτωση.

Έχοντας φορτώσει, ύστερα, ένα από τα λεξικά (λειτουργία Load) μπορούμε να επιλέξουμε Details → Dictionary για να μας εμφανιστούν τα ζητούμενα στατιστικά. Για το testing χρησιμοποιούμε την περιγραφή του βιβλίου "Nemesis" του Jo Nesbo. Φορτώνουμε το λεξικό και ο χρήστης με την προαναφερθείσα επιλογή μπορεί να δει τα στατιστικά σε ένα pop-up παράθυρο.

Επιλέγουμε Application → Start και το παιχνίδι ξεκινάει, δείχνοντας στο main page τα ζητούμενα στοιχεία: Available Words, Total Points (0), Correct guesses ("-" στην αρχή που το ποσοστό δεν ορίζεται) και Tries Remaining (6).

Μέρος (β): Το Παιχνίδι

Καταρχάς, επισημαίνουμε ότι έχουν προστεθεί σε όλα τα πιθανά σημεία που μπορεί να προκληθεί πρόβλημα, προειδοποιητικά μηνύματα για τον χρήστη (πχ δεν μπορεί να εισάγει γράμμα στην κρυμμένη λέξη αν δεν έχει πατήσει πρώτα START). Οι θέσεις γραμμάτων είναι από 1 έως το μήκος της εκάστοτε κρυμμένης λέξης. Έχουμε φροντίσει ώστε η είσοδος να πρέπει να είναι ένα γράμμα κι ένας φυσικός αριθμός. Το γράμμα μπορεί να είναι κεφαλαίο

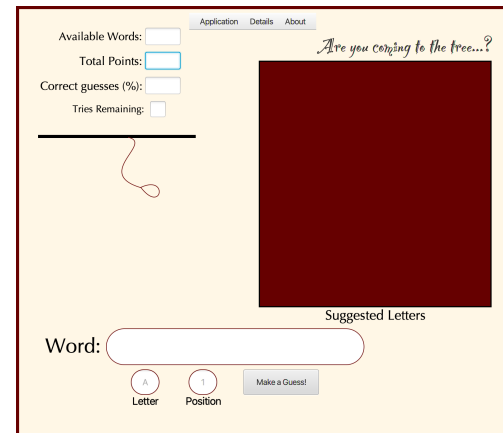


Figure 1: Main Page

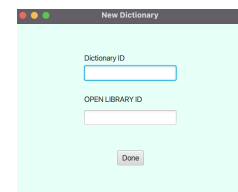


Figure 2: Create

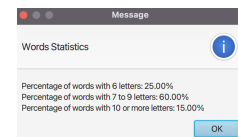


Figure 3: Stats

ή μη (γίνεται κατάλληλη επεξεργασία κι έλεγχος ώστε να γίνονται δεκτά και τα δύο) αρκεί να είναι στην Αγγλική γλώσσα. Αν είναι σε άλλη γλώσσα, θεωρείται λάθος γράμμα (όχι άκυρη είσοδος). Ο αριθμός εισόδου (που αντιστοιχεί στην θέση που θέλουμε να βάλουμε το γράμμα) πρέπει να είναι στα επιτρεπτά όρια για να είναι έγκυρη η είσοδος.

Σε κάθε άλλη περίπτωση βγαίνει κατάλληλο μήνυμα (και `InvalidInputException`) που πληροφορεί τον χρήστη για τη μορφή μιας έγκυρης εισόδου. Επιπλέον, μαντεψιά σε θέση γράμματος που έχει βρεθεί δεν επιτρέπεται από το παιχνίδι (εγείρεται μήνυμα στον χρήστη).

Με την εκκίνηση του παιχνιδιού, φαίνεται στον δεξιά πίνακα η λίστα προτάσεων γραμμάτων. Για κάθε θέση προτείνουμε γράμματα με βάση μια πιθανότητα (όσο αριστερότερα τόσο πιθανότερο ένα γράμμα). Για τον υπολογισμό αυτής της πιθανότητας ακολουθούμε την εξής διαδικασία: (1) κρατάμε σε ένα σύνολο τις λέξεις που είναι ισομήκεις με την κρυμμένη λέξη, (2) για κάθε θέση υπολογίζουμε την πιθανότητα εμφάνισης κάθε γράμματος. Από το σύνολο αυτό έχουμε εκ των προτέρων αφαιρέσει την κρυμμένη λέξη, καθώς παρατηρήσαμε ότι σε μια-δυο σωστές μαντεψιές σε κάθε θέση έχει απομείνει μόνο το σωστό γράμμα. Αυτό συμβαίνει επειδή σε κάθε σωστή επιλογή αφαιρούμε τις λέξεις που δεν έχουν το επιλεχθέν γράμμα στην θέση επιλογής και σε κάθε λάθος αφαιρούμε τις λέξεις που το έχουν.

Γενικά, η παραπάνω διαδικασία οδηγεί το σύνολο προτάσεων να μειώνεται απότομα σε κάθε σωστή μαντεψιά.

Ολες οι ζητούμενες λειτουργίες έχουν υλοποιηθεί.

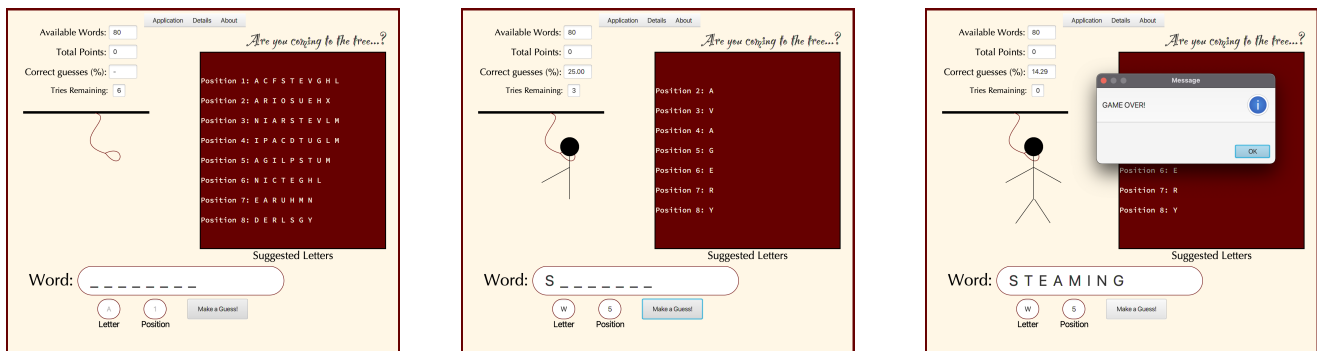


Figure 4: Example of gameplay

Σκιαγράφηση Αρχιτεκτονικής

Η αρχιτεκτονική μας (το src τμήμα της) περιγράφεται συνοπτικά ως εξής: (1) αρχεία με prefix "Aux": αρχεία με βοηθητικές συναρτήσεις που χρησιμοποιούμε στον κύριο κώδικα σαν μαύρα κουτιά, (2) αρχεία με suffix "Exception": περιέχουν την υλοποίηση των custom εξαίρέσεων (πέραν των προαναφερθέντων, έχουμε υλοποιήσει την εξαίρεση `NoStartException` για όταν ο χρήστης πατάει κουμπί που προαπαιτεί να έχει πατηθεί το "Start", την `DictNotExistException` για όταν το επιλεχθέν Dictionary δεν υπάρχει στον φάκελο medialab και την `NoLoadedDictionaryException` για όταν ο χρήστης πατάει κουμπί που προϋποθέτουν την επιλογή ενός dictionary), (3) η Main: κλάση εκκίνησης, (4) αρχεία με suffix "Controller": ελέγχουν τις λειτουργίες κάθε fxml page, ένας controller για κάθε page που επιτελεί λειτουργίες. Έχουμε 5 fxml σελίδες, αλλά 3 controllers, επειδή οι 3 pages είναι εντός των προκαθορισμένων ζητούμενων της εργασίας και οι δύο είναι επιπλέον προσθήκες μας, και δεν επιτελούν λειτουργίες. Πρόκειται για μια σελίδα με τα στοιχεία του Creator και μια ακόμα με τους κανόνες του παιχνιδιού.

Όποιο βοηθητικό αρχείο κώδικα (πχ βοηθητικά jars για την επεξεργασία JSON Objects) έχει χρησιμοποιηθεί συμπεριλαμβάνεται στο τελικό παραδοτέο.

Το επιλεχθέν αρχείο στο οποίο κάθε public μέθοδος είναι τεκμηριωμένη σύμφωνα με τις προδιαγραφές του εργαλείου javadoc είναι το `AuxCheckInput.java`.

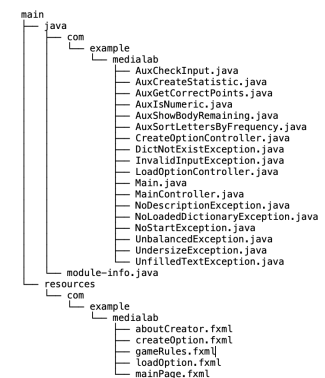


Figure 5: Source Code Tree