



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών
Εξάμηνο 7ο

Τελική εργασία στο μάθημα Τεχνολογία Πολυμέσων

Περιγραφή Εφαρμογής

Στα πλαίσια της εργασίας θα υλοποιηθεί μια έκδοση του κλασσικού παιχνιδιού «Κρεμάλα», όπου ο χρήστης θα παίζει με αντίπαλο τον υπολογιστή.

1. Σχεδιασμός και υλοποίηση λογικής (50%)

Ο υπολογιστής θα επιλέγει από λίστα λέξεων (το ενεργό λεξικό) μια λέξη και ο παίκτης θα πρέπει να βρεί ποια λέξη είναι αυτή. Ο σκοπός του παιχνιδιού είναι να βρεθεί ο σωστός χαρακτήρας για κάθε κενό ώστε να σχηματιστεί η επιλεγμένη λέξη με 5 λανθασμένες επιλογές το πολύ.

A. Δημιουργία λεξικού

Η εφαρμογή θα επιλέγει τυχαία την λέξη για κάθε γύρο παιχνιδιού από ένα αρχείο που ονομάζουμε λεξικό. Πρόκειται για ένα αρχείο κειμένου που περιλαμβάνει μια λέξη σε κάθε γραμμή του. Η εφαρμογή θα πρέπει να σχεδιαστεί με γενικό τρόπο ώστε να μπορεί να υποστηρίξει πολλαπλά λεξικά. Θεωρούμε πως υπάρχει ένας προκαθορισμένος φάκελος με όνομα **“medialab”** που περιλαμβάνει μια σειρά από τέτοια αρχεία με υποψήφιας λέξεις, τα οποία θα πρέπει να ονομάζονται ως **“hangman_DICTIONARY - ID.txt”**.

Για την δημιουργία τους ο χρήστης θα πρέπει να προσδιορίσει το DICTIONARY-ID (πρόκειται για ένα οποιοδήποτε αλφαριθμητικό). Η εφαρμογή θα λαμβάνει περιεχόμενο κειμένου από τη web υπηρεσία Open Library [1]. Από τα διαθέσιμα REST APIs θα χρησιμοποιήσουμε το Books API [2] και πιο συγκεκριμένα το “Works API”. Για το σκοπό αυτό, ο χρήστης θα προσδιορίζει το “Open Library ID” για κάποιο από τα διαθέσιμα βιβλία και στη συνέχεια η εφαρμογή μέσω της κλήσης **<https://openlibrary.org/works/OPEN-LIBRARY-ID.json>** θα λαμβάνει τα στοιχεία με βάση το πρότυπο ανταλλαγής δεδομένων JSON. Από τα διαθέσιμα αντικείμενα που περιλαμβάνει η περιγραφή μας ενδιαφέρει το **“description”**, και πιο συγκεκριμένα το πεδίο **“value”** τα περιεχόμενα του οποίου θα χρησιμοποιεί η εφαρμογή για την δημιουργία του λεξικού.

Για το σκοπό αυτό, η εφαρμογή θα δημιουργεί μια λίστα με όλες τις λέξεις της διαθέσιμης περιγραφής που έχουν τουλάχιστον 6 γράμματα. Οι επιλεγμένες λέξεις θα

πρέπει να καταχωρούνται με όλα τα γράμματα κεφαλαία. Θα πρέπει να αγνοείτε όλα τα σημεία στίξης.

Τέλος, το σύνολο των λέξεων που προκύπτουν για να θεωρηθεί ένα έγκυρο λεξικό και να αποθηκευτεί στο αντίστοιχο αρχείο (*hangman_DICTIONARY -ID.txt*, μια λέξη σε κάθε γραμμή) θα πρέπει να πληρούνται οι ακόλουθοι περιορισμοί:

- Κάθε λέξη πρέπει να περιλαμβάνεται μόνο μια φορά.
- Ένα λεξικό θα πρέπει να περιλαμβάνει τουλάχιστον 20 υποψήφιες λέξεις.
- Δεν πρέπει να υπάρχουν λέξεις με λιγότερα από 6 γράμματα.
- Τουλάχιστον 20% των λέξεων θα πρέπει να αποτελούνται από 9 ή περισσότερα γράμματα.

Για κάθε μία από τις παραπάνω περιπτώσεις μπορείτε να ορίσετε τον αντίστοιχο τύπο εξαίρεσης: ***InvalidCountException***, ***UndersizeException***, ***InvalidRangeException***, ***UnbalancedException***.

Παράδειγμα αλληλεπίδρασης με Open Library ID:

- Μέσω του website (<https://openlibrary.org>) διαλέγουμε ένα από τα διαθέσιμα βιβλία.
- Ελέγχουμε ότι η διαθέσιμη περιγραφή δεν είναι υπερβολικά σύντομη.
- Κρατάμε το Open Library ID π.χ για το *"A Game of Thrones, Book One of A Song of Ice and Fire"* είναι το *"OL31390631M"*.
- Στο κώδικα της εφαρμογής χρησιμοποιούμε την κλήση: ***https://openlibrary.org/works/OL31390631M.json*** για να πάρουμε την συνολική περιγραφή.
- Τέλος, χρησιμοποιώντας το JAVA API για την διαχείριση JSON δεδομένων κρατάμε τα περιεχόμενα του πεδίου ***"value"*** του αντικειμένου με όνομα ***"description"***.

B. Εκτέλεση παιχνιδιού

Βήματα:

1. Επιλέγεται, με τυχαίο τρόπο, μία λέξη από το σύνολο των λέξεων που ανήκουν στο ενεργό λεξικό.
2. Σε κάθε γύρο ο παίκτης καλείται να επιλέξει ένα χαρακτήρα. Για κάθε κενό θα υπάρχει μία λίστα με υποψήφιους χαρακτήρες από τους οποίους μπορεί να επιλέξει ο παίκτης.
3. Οι χαρακτήρες αυτοί θα είναι ταξινομημένοι με βάση την πιθανότητά να είναι κάποιος από αυτούς η σωστή επιλογή για την δεδομένη κενή θέση.

4. Η πιθανότητα αυτή προκύπτει με βάση το ποσοστό των λέξεων που ανήκουν στο υποσύνολο των υποψήφίων κρυμμένων λέξεων και που έχουν στην υποψήφια κενή θέση τον δεδομένο χαρακτήρα. Μία λέξη ανήκει σε αυτό το υποσύνολο εάν έχει ίσο πλήθος χαρακτήρων και εμφανίζει κοινούς χαρακτήρες με την κρυμμένη λέξη για θέσεις για τις οποίες έχει γίνει ήδη η επιλογή.

5. Η επιλογή ενός χαρακτήρα έχει δύο ενδεχόμενα αποτελέσματα:

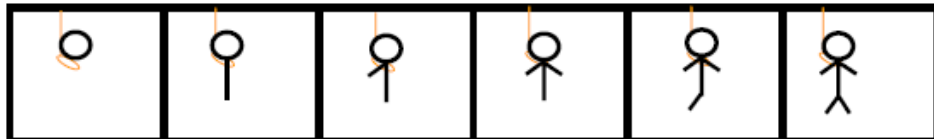
a. **Να βρούμε τον κρυμμένο χαρακτήρα.** Τότε ο χαρακτήρας αυτός εμφανίζεται στην αντίστοιχη θέση και απομακρύνονται από το σύνολο των υποψήφίων λέξεων οι λέξεις εκείνες που στην αντίστοιχη θέση δεν υπάρχει ο χαρακτήρας που επιλέξαμε. Για τις υπόλοιπες κενές θέσεις, υπολογίζουμε ξανά τις προαναφερθείσες πιθανότητες με βάση το νέο σύνολο υποψήφίων λέξεων που προέκυψε.

Ο παίκτης κερδίζει πόντους σύμφωνα με τους παρακάτω κανόνες:

- 5 πόντους αν η πιθανότητα του γράμματος ήταν μεγαλύτερη ή ίση από 0.6
- 10 πόντους αν η πιθανότητα του γράμματος ήταν μικρότερη από 0.6 και μεγαλύτερη ή ίση από 0.4
- 15 πόντους αν η πιθανότητα του γράμματος ήταν μικρότερη από 0.4 και μεγαλύτερη ή ίση από 0.25
- 30 πόντους αν η πιθανότητα του γράμματος ήταν μικρότερη από 0.25

b. **Να μην βρούμε τον κρυμμένο χαρακτήρα.** Σε αυτή την περίπτωση, απομακρύνουμε από το σύνολο των υποψήφίων λέξεων τις λέξεις που έχουν τον χαρακτήρα που επιλέξαμε στην συγκεκριμένη θέση και υπολογίζουμε εκ νέου τις πιθανότητες ύπαρξης κάθε γράμματος για κάθε κενή θέση με βάση το νέο υποσύνολο υποψήφίων λέξεων που προκύπτει. Τέλος μειώνουμε τον μετρητή των «ευκαιριών» κατά ένα και προβάλλουμε την κατάλληλη εικόνα.

Για κάθε λανθασμένη επιλογή γράμματος θα πρέπει να αφαιρούνται 15 πόντοι. Οι συνολικοί πόντοι του παίκτη δεν μπορεί ποτέ να είναι αρνητικοί.



6. Το παιχνίδι ολοκληρώνεται:

a. Όταν συμπληρωθούν τα κενά με τους σωστούς χαρακτήρες και ο παίκτης κερδίζει

- b. Όταν γίνουν 6 λανθασμένες επιλογές και ο παίκτης χάνει. Σε αυτή την περίπτωση θα πρέπει να αυτόματα να αποκαλύπτεται η ζητούμενη λέξη.

B.2. Δημιουργία γραφικής διεπαφής (30%)

Θα πρέπει να σχεδιάσετε και να υλοποιήσετε την κατάλληλη γραφική διεπαφή χρήστη (Graphical User Interface - GUI) χρησιμοποιώντας το πλαίσιο JavaFX [3][4].

Σημείωση: Στη συνέχεια παρουσιάζονται οι βασικές προδιαγραφές για την γραφική διεπαφή, για όλες τις λεπτομέρειες της τελικής υλοποίησης μπορείτε να κάνετε όποιες επιλογές θέλετε σχετικά με την εμφάνιση και τη γενικότερη αλληλεπίδραση του χρήστη με την εφαρμογή, χωρίς καμία επίπτωση στην τελική βαθμολογία. Για παράδειγμα, μπορείτε να επιλέξετε μια απλή απεικόνιση για τα διάφορα στοιχεία χρησιμοποιώντας διαφορετικά χρώματα και κείμενο ή να συνδυάσετε εικόνες με διάφορα χαρακτηριστικά από το JavaFX ώστε να δημιουργήσετε ένα αποτέλεσμα που να αντιστοιχεί σε μια σύγχρονη εφαρμογή. Σε κάθε περίπτωση, δεν υπάρχει λόγος να κάνετε υπερβολικά πολύπλοκο το συγκεκριμένο μέρος της εργασίας.

Για την δημιουργία του γραφικού περιβάλλοντος της εφαρμογής θα πρέπει να ακολουθήσετε τις παρακάτω γενικές οδηγίες:

1. Δημιουργήστε το κεντρικό «παράθυρο» της εφαρμογής με τίτλο “MediaLab Hangman” και ορίστε τις κατάλληλες διαστάσεις.
2. Χωρίστε το παράθυρο σε τρία βασικά μέρη.
3. Στο πάνω μέρος της οθόνης θα εμφανίζονται συγκεντρωτικές πληροφορίες:
 - a. Πλήθος διαθέσιμων λέξεων στο ενεργό λεξικό.
 - b. Συνολικοί πόντοι παίκτη στο τρέχον παιχνίδι.
 - c. Ποσοστό επιτυχών επιλογών γραμμάτων στο τρέχον παιχνίδι.
4. Στο μεσαίο τμήμα που θα αποτελείται από δύο επιμέρους τμήματα:
 - a. Το πρώτο, που θα βρίσκεται αριστερά, θα αντιστοιχεί στη γραφική παρουσίαση της λέξης και της εικόνας που αντιστοιχεί στο πλήθος των λανθασμένων γραμμάτων από τον παίκτη. Θα πρέπει να παρουσιάζει με ευδιάκριτο τρόπο τα κενά και τα σωστά επιλεγμένα γράμματα. Υπάρχει πλήρης ελευθερία για τον τρόπο παρουσίασης των συγκεκριμένων πληροφοριών καθώς και πιθανών επιπρόσθετων πληροφοριών της επιλογής σας. Τα περιεχόμενα θα πρέπει να ενημερώνονται αυτόματα μετά από κάθε επιλογή του παίκτη και ανάλογα με το αποτέλεσμα της.
 - b. Το δεύτερο, που θα βρίσκεται δεξιά, θα παρουσιάζει τη λίστα με τους υποψήφιους χαρακτήρες για τις κενές θέσεις. Υπάρχει πλήρης ελευθερία για τον τρόπο παρουσίασης των συγκεκριμένων πληροφοριών. Τα

περιεχόμενα θα πρέπει να ενημερώνονται αυτόματα μετά από κάθε επιλογή του παίκτη και ανάλογα με το αποτέλεσμα της.

5. Στο κάτω μέρος της οθόνης:

- a. Θα υπάρχει η κατάλληλη φόρμα που θα επιτρέπει στον παίκτη να επιλέγει κάποιο γράμμα για κάποια συγκεκριμένη θέση. Μετά την επιλογή θα πρέπει να ανανεώνονται τα κατάλληλα πεδία της γραφικής διεπαφής.

6. Προσθέστε ένα menu bar που θα περιλαμβάνει:

- a. Menu “Application” με επιλογές:

- i. Start: Έναρξη νέας εκτέλεσης με βάση το επιλεγμένο λεξικό. Αν υπάρχει κάποια ενεργή εκτέλεση θα πρέπει να διακόπτεται και να αρχίζει μια νέα με κατάλληλη αρχικοποίηση όλων των παραμέτρων και των πληροφοριών που υπάρχουν στην γραφική διεπαφή.
- ii. Load: Μέσω ενός popup παραθύρου ο χρήστης θα προσδιορίζει το “DICTIONARY-ID”. Στη συνέχεια η εφαρμογή θα αναζητά το αντίστοιχο αρχείο (hangman_ DICTIONARY-ID.txt) στο προκαθορισμένο φάκελο “medialab” και αν υπάρχει θα γίνεται η κατάλληλη αρχικοποίηση της εφαρμογής, διαφορετικά θα εμφανίζεται μήνυμα σφάλματος.
- iii. Create: Μέσω ενός popup παραθύρου ο χρήστης θα προσδιορίζει το DICTIONARY-ID που θα αποδοθεί στο νέο λεξικό και το “Open Library ID” που αντιστοιχεί στο βιβλίο του οποίου η περιγραφή θα χρησιμοποιηθεί για την εξαγωγή των υποψηφίων λέξεων. Στη συνέχεια θα εκτελείται η διαδικασία που περιγράφεται στην προηγούμενη ενότητα για την δημιουργία του λεξικού, αν το λεξικό δεν είναι έγκυρο (έχουν προκύψουν εξαιρέσεις) θα εμφανίζεται το κατάλληλο μήνυμα σφάλματος.
- iv. Exit: Τερματισμός εφαρμογής.

- b. Menu “Details” με επιλογές:

- i. Dictionary: Μέσω ενός popup παραθύρου θα παρουσιάζει το ποσοστό των λέξεων του ενεργού λεξικού με 6 γράμματα, 7 έως 9 γράμματα και 10 ή περισσότερα γράμματα.
- ii. Rounds: Μέσω ενός popup παραθύρου θα παρουσιάζει για τα 5 τελευταία ολοκληρωμένα παιχνίδια τις παρακάτω πληροφορίες: επιλεγμένη λέξη, πλήθος προσπαθειών και νικητή (παίκτης ή υπολογιστής).
- iii. Solution: Το παιχνίδι καταχωρείται ως χαμένο για τον παίκτη και γίνεται αποκάλυψη της επιλεγμένης λέξης.

B.3. Λοιπές απαιτήσεις (20%)

- Η υλοποίηση θα πρέπει να ακολουθεί τις αρχές σχεδίασης του αντικειμενοστραφούς προγραμματισμού (OOP design principles).
- Σε μια κλάση της επιλογής σας θα πρέπει κάθε public μέθοδος που περιέχει να είναι τεκμηριωμένη σύμφωνα με τις προδιαγραφές του εργαλείου javadoc [5].

Σημείωση: Για οτιδήποτε δεν είναι ξεκάθαρο από την εκφώνηση μπορείτε να κάνετε τις δικές σας παραδοχές και υποθέσεις. Γενικότερα, στην εκφώνηση περιγράφονται οι βασικές απαιτήσεις που πρέπει να ακολουθεί η εφαρμογή σας, ωστόσο μπορείτε να κάνετε δικές σας σχεδιαστικές παραδοχές προσπαθώντας να κάνετε πιο ρεαλιστική την εφαρμογή, χωρίς παράλληλα να κάνετε πολύπλοκη την υλοποίηση.

Παραδοτέα

- Το project (του IDE της επιλογής σας) με την υλοποίηση της εφαρμογής.
- Μια σύντομη (το πολύ 2 σελίδες) αναφορά που θα περιέχει το σχεδιασμό της υλοποίησης, θα αναφέρετε όποια λειτουργικότητα δεν έχετε υλοποιήσει καθώς και όποιες επιπρόσθετες παραδοχές έχετε κάνει. Σε καμία περίπτωση δεν θα πρέπει να συμπεριλάβετε τμήματα κώδικα.

Αναφορές

[1] <https://openlibrary.org/developers/api>

[2] <https://openlibrary.org/dev/docs/api/books>

[3] <https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm>

[4] https://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm

[5] <https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html>