



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Εξάμηνο 7^ο: Τεχνολογία Πολυμέσων

Αναφορά Εξαμηνιαίας Εργασίας

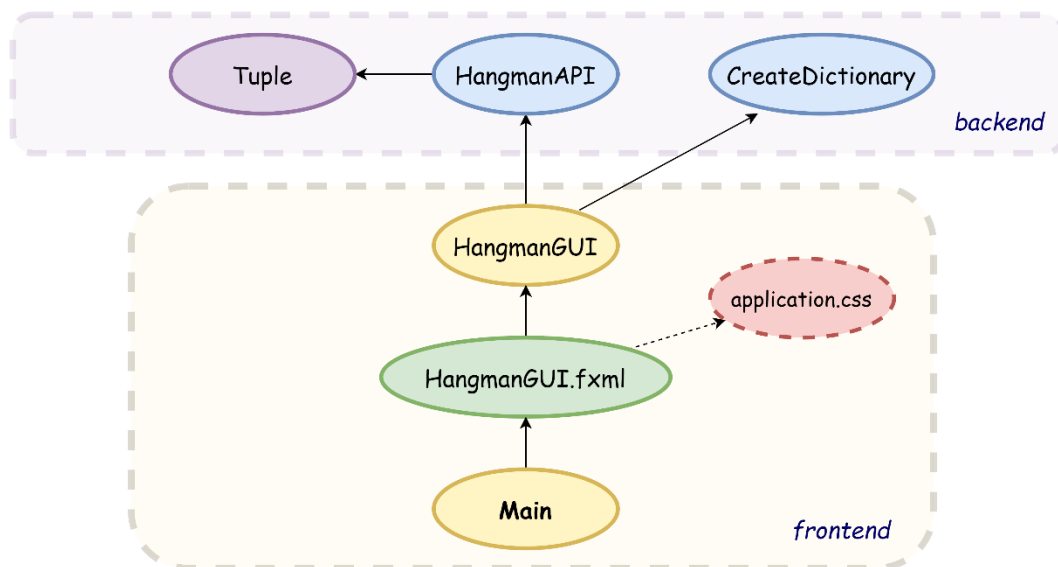
Χρήστος (Κίτσος) Ορφανόπουλος – el17025

Εισαγωγή

Το ζητούμενο της εξαμηνιαίας εργασίας ήταν η υλοποίηση μιας τροποποιημένης έκδοσης του παιχνιδιού «Κρεμάλα» σε **Java**, και πιο συγκεκριμένα η δημιουργία ενός GUI με χρήση του εργαλείου **JavaFX**.

Σχεδιασμός υλοποίησης

Κατά την υλοποίηση της εφαρμογής, τηρώντας της αρχές του αντικειμενοστραφούς προγραμματισμού, χώρισα τις απαιτούμενες λειτουργικότητες σε διαφορετικά αντικείμενα. Πιο συγκεκριμένα έγινε διάκριση των λειτουργικοτήτων της διεπαφής GUI (**frontend**) από τη λογική του προγράμματος (**backend**). Ο γράφος εξαρτήσεων (**dependency graph**) του παιχνιδιού παρουσιάζεται παρακάτω:



Επεξήγηση λειτουργικότητας κάθε κλάσης:

- **Main:** η κλάση που εκτελεί ο χρήστης. Εξαρτάται από την κλάση **HangmanGUI** (μέσω του βοηθητικού αρχείου **HangmanGUI.fxml**).
- **HangmanGUI:** η βασική κλάση για τη διεπαφή του χρήστη. Υλοποιεί μεθόδους που διαχειρίζονται τα διάφορα κουμπιά (dropdowns, textboxes etc.) που βλέπει ο χρήστης. Επίσης διαχειρίζεται τα διάφορα γραφικά στοιχεία της διεπαφής
- **HangmanAPI:** η κλάση αυτή δημιουργεί αντικείμενα **hangman** τα οποία υλοποιούν όλη τη λογική του παιχνιδιού.

- **CreateDictionary:** η κλάση αυτή είναι υπεύθυνη για τη δημιουργία των λεξικών που θα χρησιμοποιηθούν στην κρεμάλα
- **Tuple:** η βοηθητική αυτή κλάση χρησιμοποιείται από τη HangmanAPI προκειμένου να ταξινομηθούν σωστά οι επιλογές των γραμμάτων με βάση την πιθανότητα εμφάνισής τους στην κρυμμένη λέξη.

Κατά την υλοποίηση της εφαρμογής θεώρησα ότι ο χρήστης επιλέγει κάθε φορά να καλύψει μόνο ένα συγκεκριμένο κενό στη λέξη, εν αντιθέσει της «κλασσικής» εκδοχής του παιχνιδιού. Επιπλέον διάφορα εξωτερικά αρχεία που χρησιμοποιεί η εφαρμογή βρίσκονται σε απόλυτες θέσεις στο filesystem του υπολογιστή μου. Τέλος θεώρησα ότι αν δεν γίνει load κάποιου συγκεκριμένου λεξικού θα ανοίγει ένα προκαθορισμένο (περιλαμβάνεται στο repository).

Εντός του φακέλου **screenshots** μπορείτε δείτε στιγμιότυπα από τη λειτουργία του παιχνιδιού. Η εφαρμογή ανοίγει by-default σε full-screen.

Αστοχίες & δυνητικές βελτιώσεις

Πέραν κάποιων περιορισμένων bugs που αφορούν την καλύτερη εμφάνιση των αποτελεσμάτων σε διαφορετικούς γύρους του παιχνιδιού και τη δημιουργία των λεξικών (*παρατηρήθηκε ότι πολλά JSON αρχεία δεν περιλάμβαναν στο description το πεδίο "value"*), γενικά μπορεί να ειπωθεί ότι υλοποιήθηκαν όλες οι απαιτούμενες λειτουργικότητες.

Για την περαιτέρω βελτίωση της εφαρμογής στο τεχνικό κομμάτι, θα προσέθετα σε ορισμένα σημεία καλύτερο χειρισμό εξαιρέσεων, και θα διόρθωνα αδύναμα σημεία του GUI, όπως για παράδειγμα το ευκολότερο κλείσιμο των pop-up παραθύρων μέσα από ειδικό κουμπί (αντί για το 'X'). Επιπλέον θα συμπλήρωνα την τεκμηρίωση Javadoc σε όλες τις κλάσεις (βλ. παρακάτω).

Ως προς τη λογική της εφαρμογής παρατηρήθηκε ότι το παιχνίδι είναι υπερβολικά δύσκολο, έτσι θα έδινα κάποια αρχικά hints ανάλογα με το μέγεθος της λέξης, και θα βελτίωνα των ορισμό της πιθανότητας εμφάνισης κάποιου γράμματος.

Λοιπές τεχνικές λεπτομέρειες

Η συγγραφή του κώδικα έγινε κατά κύριο λόγο στο **Visual Studio Code** σε κωδικοποίηση **UTF-8**. Το VSC υποστηρίζει την προβολή ορισμένων χρωματικών emojis (πχ. 🟥 🟦 🟩), τα οποία προστέθηκαν σε σχόλια για λόγους ευαναγνωσιμότητας. Η εκτέλεση του κώδικα έγινε μέσω του **Eclipse IDE**. Ο σχεδιασμός της γραφικής διεπαφής έγινε με χρήση του εργαλείου **SceneBuilder**.

Οι κλάσεις **CreateDictionary**, **HangmanAPI** και **Tuple** ακολουθούν το πρότυπο τεκμηρίωσης **Javadoc**.

Για τη γραφιστική μορφοποίηση της εμφάνισης ορισμένων αντικειμένων της διεπαφής χρησιμοποιήθηκε η γλώσσα σήμανσης **CSS** (βλ. **application.css**).

Τέλος κατά τη συγγραφή του κώδικα χρησιμοποιήθηκε το **GIT** (**private repository** στο **GitHub**) για version control. Το αρχείο **README.md** περιλαμβάνει πιο συγκεκριμένες πληροφορίες για την οργάνωση των αρχείων του repository.

