Θέμα εργασίας: Ανάπτυξη παιχνιδιού λέξεων με Python

Ονοματεπώνυμο: Γιαπλές Νικόλαος (Α.Μ. 1092823)

Τμήμα του έργου που ανέλαβα:

Στον χωρισμό των βασικών εργασιών για την υλοποίηση του κώδικα ανέλαβα το "τεχνικό" κομμάτι του μενού, δηλαδή ό,τι δεν έχει να κάνει με γραφικά στοιχεία. Πιο συγκεκριμένα, ασχολήθηκα με τα εξής προβλήματα:

1. Δόμηση κώδικα μέσα στον οποίο θα δημιουργηθούν τα γραφικά στοιχεία, με τρόπο που να διευκολύνει την σύνδεση του μενού με τα υπόλοιπα κομμάτια του τελικού προγράμματος.
2. Εύρεση τρόπου αλλαγής και αποθήκευσης ρυθμίσεων (το χρώμα του υποβάθρου, την ένταση της μουσικής και των εφέ ήχου) ώστε ο χρήστης να έχει την δυνατότητα να αλλάζει χαρακτηριστικά του προγράμματος.

Αφότου λύθηκαν αυτά τα προβλήματα, βοήθησα στην σύνδεση των διαφορετικών κομματιών του προγράμματος διορθώνοντας bugs που προέκυψαν.

1) <u>Δόμηση κώδικα:</u>

Το αρχείο "final_menu.py" του πυγαίου κώδικα περιέχει δύο κλάσεις, καθώς και μία συνάρτηση. Αναλυτικότερα:

- Συνάρτηση main(): δημιουργεί το παράθυρο του μενού κατά την εκτέλεση του προγράμματος και κατά την αλλαγή μεταξύ παραθύρων, αρχικοποιώντας τα γραφικά στοιχεία και τους ήχους που χρησιμοποιούνται και φορτώνοντας τις ρυθμίσεις.
- Κλάση Menu(): περιέχει τα γραφικά στοιχεία του αρχικού παραθύρου του μενού, καθώς και διάφορες συναρτήσεις απαραίτητες για την λειτουργία του
- Κλάση Settings(): περιέχει τα γραφικά στοιχεία του παραθύρου "settings" που δημιουργείται όταν ο χρήστης πατήσει το κουμπί "SETTINGS" στο κύριο μενού. Επιπλέον, μέσω αυτής της κλάσης ο χρήστης μπορεί να τροποποιήσει τις ρυθμίσεις.

2) Ρυθμίσεις:

Υπάρχουν αρκετά “configuration formats” (τύποι αρχείου που περιέχουν δεδομένα οργανωμένα κατά μία συγκεκριμένη “γλώσσα” με στόχο την διευκόλυνση την ανάγνωση των στοιχείων από μία γλώσσα προγραμματισμού). Πλέον χρησιμοποιούνται τέσσερεις τέτοιες γλώσσες κατά κύριο λόγο, οι XML, JSON, YAML και HCL. Δεδομένου της περιπλοκότητας των XML και JSON ως προς το πρόβλημα, καθώς και την εξειδικευμένη φύση της HCL, κατέληξα ότι η YAML είναι η καταλληλότερη.

Ένα αρχείο .yaml είναι δομημένο με τρόπο παρόμοιο των λεξικών της Python. Δηλαδή, σε κάθε γραμμή γράφονται δύο πράγματα, ένα “κλειδί” και η τιμή που του αντιστοιχεί, διαχωρισμένα με : (π.χ. αν θέλουμε να αντιστοιχήσουμε στο κλειδί “score” τον αριθμό 20, θα γράφαμε ‘score: 20’). Η YAML υποστηρίζει και πιο περίπλοκες δομές δεδομένων πέρα των απλών integers, strings, floats κ.λπ., όμως δεν ήταν αναγκαίο να χρησιμοποιήσουμε αυτή την δυνατότητα σε αυτή την περίπτωση.

Η διεπαφή με το αρχείο config.yaml που περιέχει τις ρυθμίσεις και το leaderboard.yaml που αποθηκεύει τους χρήστες και το σκορ που πέτυχαν γίνεται με την βιβλιοθήκη PyYAML. Η βιβλιοθήκη αυτή διευκολύνει την ανάγνωση του κάθε αρχείο με την συνάρτηση **yaml.load()** που παίρνει ως ορίσματα την διεύθυνση του αρχείου και ένα αντικείμενου τύπου Loader(εμπεριέχεται στην βιβλιοθήκη), και μας επιστρέφει ένα λεξικό που μπορεί να χρησιμοποιηθεί απευθείας. Έπειτα, για την αποθήκευση αλλαγών στις ρυθμίσεις χρησιμοποιείται η **yaml.dump()** με ορίσματα ένα λεξικό και την διεύθυνση του αρχείου το οποίο θέλουμε να τροποποιήσουμε ή να δημιουργήσουμε αν δεν υπάρχει. Η δυνατότητα αυτή μας επιτρέπει να αντιμετωπίσουμε την περίπτωση όπου ο χρήστης δεν έχει τα απαραίτητα αρχεία με έναν βρόγχο try-except, δημιουργώντας τα κατά την πρώτη εκτέλεση του προγράμματος. Έτσι, δεν είναι αναγκαίο να περιέχονται αυτά τα αρχεία μαζί με το εκτελέσιμο.