



Σχολή Θετικών Επιστημών και Τεχνολογίας
Προπτυχιακό Πρόγραμμα Σπουδών
Πληροφορική

Προγραμματιστικό Project - ΠΛΗΠΡΟ
«Trivia – Ανάπτυξη ενός παιχνιδιού γνώσεων»

Γεώργιος Αποστολίδης
Αριστείδης Καρβούνης
Γεωργία Κουκουτού

Επιβλέπουσα καθηγήτρια:
Αντωνία Στεφανή

Αθήνα, Ιούνιος 2022

Περιεχόμενα

Εισαγωγή	4
1. Ανάλυση του προβλήματος	4
1.1. Απαιτήσεις και αναμενόμενα αποτελέσματα.....	4
1.2. Περιγραφή της εφαρμογής	4
1.3. Διαμοιρασμός εργασιών	8
2. Ορισμός προγραμματιστικών προδιαγραφών	8
2.1. Διεπαφή χρήστη (View).....	8
2.2. Game State (Controller)	9
2.3. Επικοινωνία με API και διαχείριση αρχείου υψηλότερων βαθμολογιών (Model) ...	10
3. Ποιότητα λογισμικού και testing	11
4. Προτάσεις για περαιτέρω ανάπτυξη	12
5. Επιπλέον πληροφορίες.....	12

Πίνακας Εικόνων

Εικόνα 1: Αρχικό παράθυρο εφαρμογής.....	5
Εικόνα 2: Βασικό παράθυρο παιχνιδιού	5
Εικόνα 3: Ενημέρωση για τερματισμό παιχνιδιού	6
Εικόνα 4: Επιβεβαίωση συνέχισης παιχνιδιού	6
Εικόνα 5: Ενημέρωση για τερματισμό υπολειπόμενου χρόνου	6
Εικόνα 6: Μήνυμα συγχαρητηρίων για την επίτευξη υψηλότερης βαθμολογίας.....	7
Εικόνα 7: Προσθήκη νέας υψηλότερης βαθμολογίας στην αρχική λίστα	7
Εικόνα 8: Παράληψη συμπλήρωσης υποχρεωτικών πεδίων από το χρήστη	11

Εισαγωγή

Σκοπός της παρούσας έκθεσης είναι να καλύψει τις τεχνικές λεπτομέρειες που σχετίζονται με την ανάπτυξη του τελικού προγραμματιστικού project της θεματικής ενότητας ΠΛΗΠΡΟ. Στόχος της είναι να παρουσιάσει τη μεθοδολογία που ακολουθήθηκε για την ανάπτυξη της εφαρμογής, τις απαιτήσεις που τέθηκαν, τον σχεδιασμό του λογισμικού και τις τεχνικές λεπτομέρειες υλοποίησης.

1. Ανάλυση του προβλήματος

1.1. Απαιτήσεις και αναμενόμενα αποτελέσματα

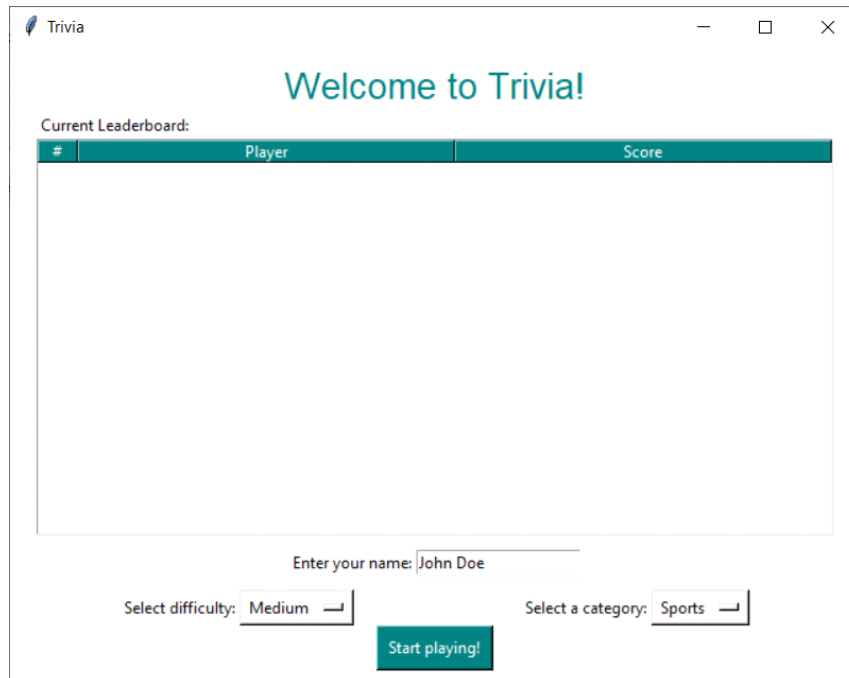
Σκοπός του project είναι η υλοποίηση ενός παιχνιδιού γνώσεων (trivia). Το παιχνίδι περιλαμβάνει ερωτήσεις πολλαπλής επιλογής και σωστού λάθους και ο χρήστης καλείται να απαντήσει σε όσες περισσότερες ερωτήσεις μπορεί μέσα σε 180 δευτερόλεπτα. Οι ερωτήσεις προέρχονται από διάφορες κατηγορίες (π.χ. Γενικών Γνώσεων, Γεωγραφίας, Ιστορία, κλπ) και διαφορετικά επίπεδα δυσκολίας (Εύκολο, Μέτριο, Δύσκολο).

Το τελικό σκορ του χρήστη υπολογίζεται με βάση τον αριθμό των σωστών και λανθασμένων απαντήσεων, το βαθμό δυσκολίας και το χρόνο απάντησης κάθε ερώτησης. Το παιχνίδι περιλαμβάνει δέκα γύρους. Εάν ο χρήστης απαντήσει σωστά σε τουλάχιστον έξι ερωτήσεις, έχει τη δυνατότητα να προχωρήσει σε επόμενο γκρουπ ερωτήσεων μέχρι είτε να συγκεντρώσει λιγότερες από έξι σωστές απαντήσεις είτε να τελειώσει ο χρόνος είτε να τελειώσουν οι γύροι.

Σε οποιαδήποτε στιγμή κατά τη διάρκεια του παιχνιδιού, ο χρήστης μπορεί να περιηγηθεί στις επόμενες ή προηγούμενες ερωτήσεις ενός γκρουπ και να αλλάξει την επιλογή του στις ερωτήσεις που έχουν ήδη απαντηθεί. Σε αυτή την περίπτωση, στον υπολογισμό του σκορ υπολογίζεται ο νέος χρόνος απάντησης. Εάν ο χρήστης έχει επιτύχει υψηλότερη βαθμολογία τότε του εμφανίζεται σχετικό μήνυμα και η βαθμολογία του καταχωρείται στο αρχείο υψηλότερων βαθμολογιών.

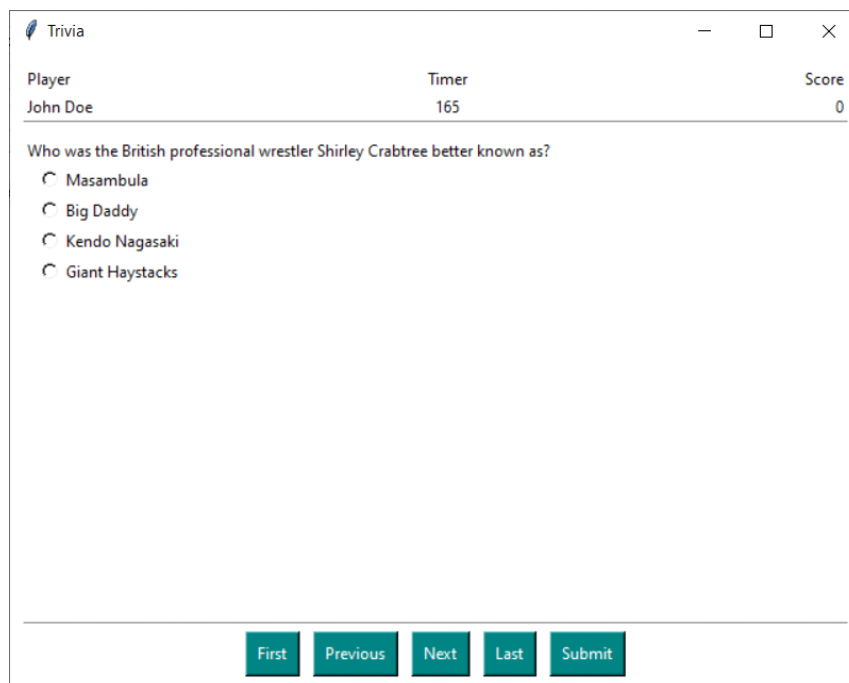
1.2. Περιγραφή της εφαρμογής

Κατά την έναρξη του παιχνιδιού, εμφανίζονται οι υψηλότερες δέκα βαθμολογίες. Ο παίκτης πρέπει να εισάγει το όνομα του και να επιλέξει κατηγορία ερωτήσεων και επίπεδο δυσκολίας (Εικόνα 1).



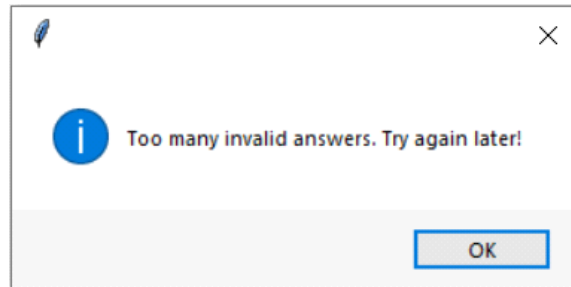
Εικόνα 1: Αρχικό παράθυρο εφαρμογής

Στη συνέχεια, μεταφέρεται στην επόμενη οθόνη και το παιχνίδι ξεκινάει. Όπως φαίνεται στην Εικόνα 2, ο χρήστης έχει τη δυνατότητα να απαντήσει την τρέχουσα ερώτηση, να περιηγηθεί σε επόμενες ή προηγούμενες ερωτήσεις για να αλλάξει την απάντηση του εάν επιθυμεί, καθώς και να υποβάλει τις απαντήσεις του.

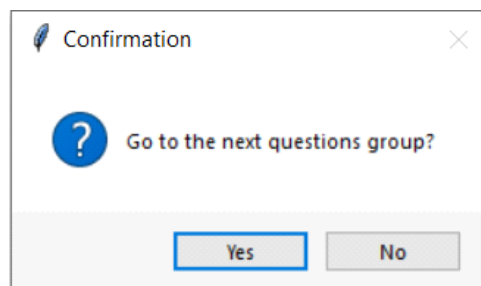


Εικόνα 2: Βασικό παράθυρο παιχνιδιού

Έπειτα από την υποβολή των ερωτήσεων, εάν ο χρήστης έχει συγκεντρώσει πάνω από τρεις λανθασμένες ερωτήσεις στο τρέχον γκρουπ εμφανίζεται σχετικό μήνυμα (Εικόνα 3) και το παιχνίδι τελειώνει. Διαφορετικά, ο παίκτης μπορεί να συνεχίσει στο επόμενο γκρουπ ερωτήσεων (Εικόνα 4).

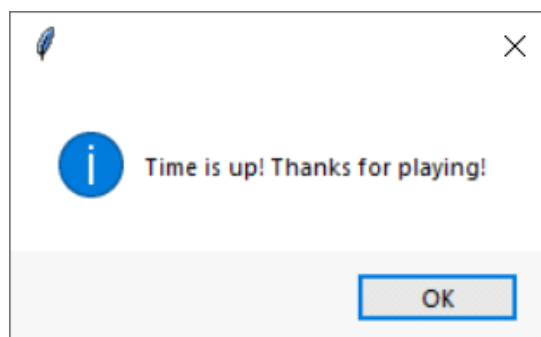


Εικόνα 3: Ενημέρωση για τερματισμό παιχνιδιού



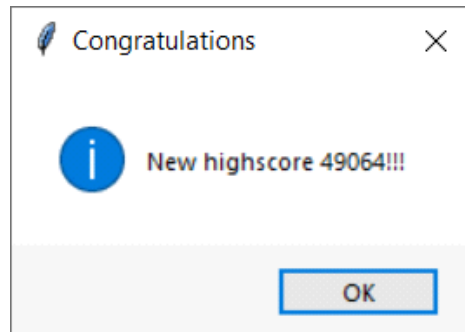
Εικόνα 4: Επιβεβαίωση συνέχισης παιχνιδιού

Στην περίπτωση που ο χρόνος τελειώσει χωρίς ο παίκτης να έχει υποβάλλει τις απαντήσεις του, το παιχνίδι τερματίζει (Εικόνα 5) και ο παίκτης μεταφέρεται στην αρχική οθόνη, από όπου και μπορεί να ξεκινήσει νέο παιχνίδι εάν επιθυμεί.

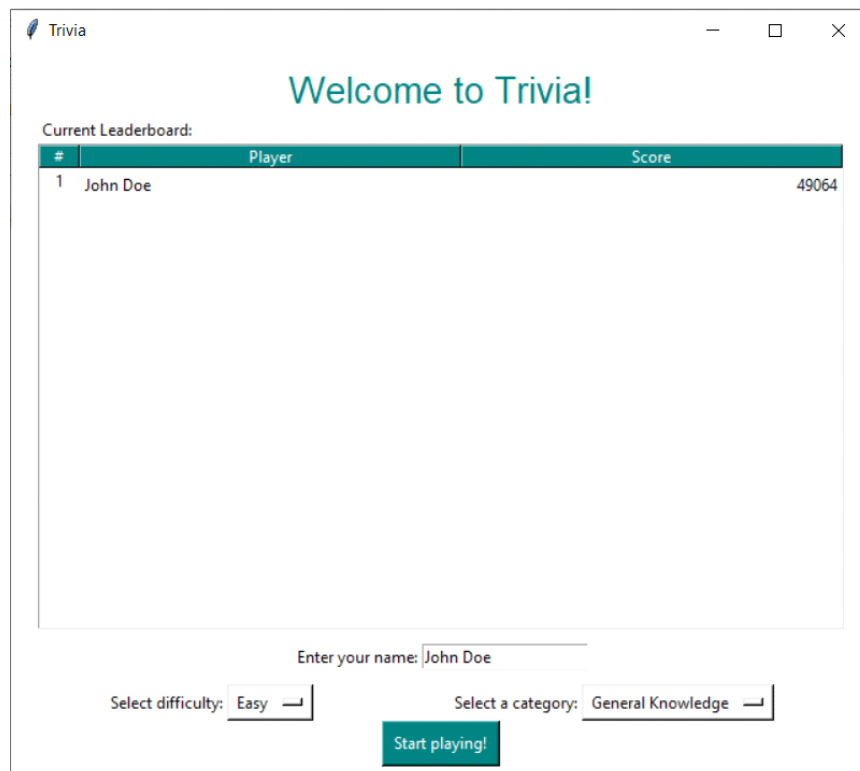


Εικόνα 5: Ενημέρωση για τερματισμό υπολειπόμενου χρόνου

Μετά το τέλος του παιχνιδιού, εάν ο παίκτης έχει επιτύχει υψηλότερη βαθμολογία, τότε του εμφανίζεται σχετικό μήνυμα (Εικόνα 6) και το σκορ του προστίθεται στον αρχικό πίνακα των δέκα υψηλότερων βαθμολογιών (Εικόνα 7).



Εικόνα 6: Μήνυμα συγχαρητηρίων για την επίτευξη υψηλότερης βαθμολογίας



Εικόνα 7: Προσθήκη νέας υψηλότερης βαθμολογίας στην αρχική λίστα

Συνοπτική παρουσίαση τεχνολογιών και βιβλιοθηκών που χρησιμοποιήθηκαν:

- Python 3.
- Βιβλιοθήκη **tkinter** για τη σχεδίαση και υλοποίηση του γραφικού περιβάλλοντος.

- Βιβλιοθήκη **requests** για την επικοινωνία με την Open Trivia DB για την συλλογή των ερωτήσεων και των κατηγοριών τους.

1.3. Διαμοιρασμός εργασιών

Η εργασία χωρίστηκε σε τρία κομμάτια, τα οποία αφορούν το σχεδιασμό του γραφικού περιβάλλοντος, την επικοινωνία με την Open Trivia DB και τη διαχείριση του αρχείου με τις υψηλότερες βαθμολογίες και την διαχείριση της κατάστασης της εφαρμογής (game state), δηλαδή την διαχείριση των παρακάτω μεταβλητών: όνομα χρήστη, κατηγορίες ερωτήσεων, επίπεδο δυσκολίας, υπολογισμός σκορ, ερωτήσεις, υπολειπόμενος χρόνος, απαντήσεις χρήστη.

2. Ορισμός προγραμματιστικών προδιαγραφών

Το project, στο σύνολο του ακολουθεί την προσέγγιση του αντικειμενοστραφούς προγραμματισμού. Παρά το γεγονός ότι η Python σαν γλώσσα προγραμματισμού υποστηρίζει τη χρήση τόσο συναρτησιακού όσο και αντικειμενοστραφούς προγραμματισμού, προτιμήθηκε κυρίως η χρήση τεχνικών αντικειμενοστραφούς προγραμματισμού, λόγω ορισμένων πλεονεκτημάτων που προσφέρει, ιδιαίτερα όσον αφορά τον ευκολότερο διαχωρισμό των διαφορετικών τμημάτων της εφαρμογής και την επαναχρησιμοποίηση του κώδικα.

Αναφορικά με την αρχιτεκτονική με βάση την οποία σχεδιάστηκε η εφαρμογή και πραγματοποιήθηκε ο διαχωρισμός των εργασιών, μπορούμε να πούμε ότι η προσέγγιση που επιλέχθηκε ακολουθεί το μοτίβο Model-view-controller (MVC). Το κομμάτι του project που αφορά τον σχεδιασμό του γραφικού περιβάλλοντος αναλαμβάνει το ρόλο της προβολής (view), δηλαδή τις διεπαφές χρήστη. Αντίστοιχα, το κομμάτι της επικοινωνίας με το Open Trivia Api και της διαχείρισης του αρχείου με τα υψηλότερα σκορ έχει το ρόλο του μοντέλου (model) και τέλος, η διαχείριση της κατάστασης της εφαρμογής έχει το ρόλο του ελεγκτή (controller), όπου γίνεται η επεξεργασία των δεδομένων που εισάγει ο χρήστης. Στη συνέχεια περιγράφονται συνοπτικά οι λειτουργίες κάθε μέρους του υποσυστήματος.

2.1. Διεπαφή χρήστη (View)

Το συγκεκριμένο υποσύστημα είναι υπεύθυνο για τον σχεδιασμό και διαχείριση της διεπαφής χρήστη (user interface). Ο σχεδιασμός πραγματοποιήθηκε με τη βοήθεια της βιβλιοθήκης tkinter.

Η συγκεκριμένη βιβλιοθήκη αποτελεί μέρος των βασικών βιβλιοθηκών που είναι ενσωματωμένες στη γλώσσα Python και μπορεί να χρησιμοποιηθεί σε διάφορα λειτουργικά συστήματα (Windows, macOS, Linux κλπ). Βασίζεται στη χρήση παραθύρων (windows), μέσα στα οποία εμπεριέχονται τα διάφορα άλλα αντικείμενα που χρησιμοποιούνται στο σχεδιασμό ενός γραφικού περιβάλλοντος, όπως είναι για παράδειγμα τα κουμπιά, οι ετικέτες ή τα πλαίσια κειμένου, τα οποία ονομάζονται widgets. Η εφαρμογή αποτελείται από δυο βασικά παράθυρα:

- Το αρχικό παράθυρο, που βλέπει ο παίκτης στην αρχή και στο τέλος της εφαρμογής και στο οποίο μπορεί να συμπληρώσει το όνομα του, να επιλέξει επίπεδο δυσκολίας και κατηγορία ερωτήσεων και να δει τις δέκα υψηλότερες βαθμολογίες.
- Το βασικό παράθυρο στο οποίο εμφανίζονται οι ερωτήσεις που καλείται να απαντήσει ο χρήστης και διαδραματίζεται όλο το παιχνίδι. Στο συγκεκριμένο παράθυρο, εκτός από τις ερωτήσεις και τις απαντήσεις τους, εμφανίζεται το όνομα του παίκτη και το σκορ του, ο υπολειπόμενος χρόνος και όλες οι ενέργειες που είναι διαθέσιμες και μπορεί να πραγματοποιήσει ο παίκτης. Αυτές είναι η μετάβαση στην πρώτη/τελευταία ερώτηση ή στην προηγούμενη/επόμενη και η τελική υποβολή των απαντήσεων. Επιπλέον, από αυτό το παράθυρο εμφανίζονται τα σχετικά μηνύματα στον παίκτη ανάλογα με τις περιπτώσεις που περιγράφηκαν παραπάνω (π.χ. ενημέρωση για επίτευξη υψηλότερης βαθμολογίας ή για λήξη του χρόνου).

2.2. Game State (Controller)

Το συγκεκριμένο υποσύστημα εμπεριέχει τη λογική της εφαρμογής, καθώς διαβάζει τις επιλογές που πραγματοποιεί ο χρήστης στο γραφικό περιβάλλον και τις επεξεργάζεται καταλλήλως. Πιο συγκεκριμένα, οι λειτουργίες του υποσυστήματος είναι οι εξής:

- Υπολογισμός του σκορ του παίκτη με βάση τη φόρμουλα που έχει δοθεί και επανυπολογισμός του μετά την ολοκλήρωση κάθε γκρουπ ερωτήσεων.
- Κατασκευή το γκρουπ των ερωτήσεων με βάση τις επιλογές του χρήστη (επίπεδο δυσκολίας και κατηγορία).
- Αποθήκευση της τρέχουσας ερώτησης που εμφανίζεται στο χρήστη, δίνοντας του τη δυνατότητα να μεταφερθεί στις επόμενες ή προηγούμενες ερωτήσεις του γκρουπ ανάλογα με την επιλογή του.

- Αποθήκευση της επιλεγμένης απάντησης του χρήστη για κάθε ερώτηση του γκρουπ, έτσι ώστε να είναι διαθέσιμη για τον υπολογισμό του σκορ και για την εμφάνιση της σε περίπτωση που ο χρήστης επιστρέψει στη συγκεκριμένη ερώτηση για να αλλάξει την απάντησή του.
- Έλεγχος των υποχρεωτικών παραμέτρων που απαιτεί το πρόγραμμα από το χρήστη (όνομα χρήστη, κατηγορία ερωτήσεων, επίπεδο δυσκολίας), για να βεβαιωθεί ότι δεν είναι κενά.
- Υπολογισμός του υπολειπόμενου χρόνου για κάθε γκρουπ ερωτήσεων.
- Διαχείριση της λογικής της τελικής υποβολής των ερωτήσεων, η οποία περιλαμβάνει την παύση υπολογισμού του υπολειπόμενου χρόνου και τον τελικό υπολογισμό του σκορ.

2.3. Επικοινωνία με API και διαχείριση αρχείου υψηλότερων βαθμολογιών (Model)

Μια από τις λειτουργίες του συγκεκριμένου υποσυστήματος είναι η επικοινωνία με το Open Trivia Api για την αποκόμιση των ερωτήσεων και των κατηγοριών τους. Πραγματοποιεί τα εξής requests:

- Ένα get request, το οποίο επιστρέφει ένα json αρχείο με τις πληροφορίες για τις κατηγορίες των ερωτήσεων.
- Ένα get request, το οποίο επιστρέφει ένα json αρχείο με τις πληροφορίες για τις ερωτήσεις (την ερώτηση, τις επιλογές, τη σωστή και τις λάθος απαντήσεις και τη δυσκολία). Με βάση αυτές τις πληροφορίες φτιάχνει αντικείμενα της κλάσης QuestionModel, η οποία αποτελεί το σχέδιο (blueprint) για την κατασκευή των ερωτήσεων του παιχνιδιού.

Για την πραγματοποίηση των αντίστοιχων requests για τα δεδομένα που αφορούν τις κατηγορίες και τις ερωτήσεις χρησιμοποιήθηκε η βιβλιοθήκη requests, η οποία επιτρέπει την πραγματοποίηση HTTP requests.

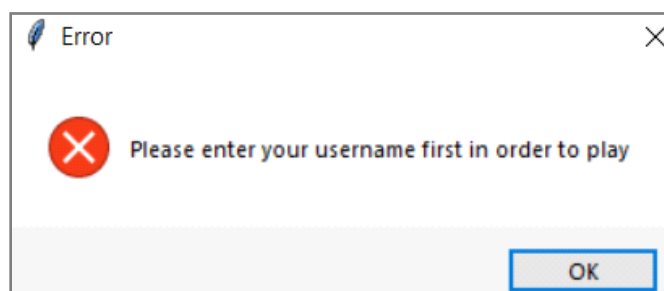
Εκτός από τα παραπάνω, το υποσύστημα διαχειρίζεται και το αρχείο των υψηλότερων βαθμολογιών. Πιο συγκεκριμένα, διαβάζει τα δεδομένα σχετικά με το ποιες είναι οι υψηλότερες βαθμολογίες και τις επιστρέφει, έτσι ώστε να μπορούν να εμφανιστούν στο αρχικό παράθυρο του παιχνιδιού, ενώ παράλληλα, μετά τον τερματισμό του παιχνιδιού ελέγχει εάν ο παίκτης έχει επιτύχει αρκετά υψηλό σκορ και ενημερώνει σχετικά το αρχείο με τις υψηλότερες βαθμολογίες.

3. Ποιότητα λογισμικού και testing

Για την εξασφάλιση της εύρυθμης λειτουργίας της εφαρμογής πραγματοποιήθηκε αμυντικός προγραμματισμός στα δεδομένα που εισάγει ο παίκτης. Αυτά αφορούν:

- Το όνομα του χρήστη.
- Το επίπεδο δυσκολίας των ερωτήσεων.
- Την κατηγορία των ερωτήσεων.

Τα πεδία αυτά είναι υποχρεωτικά, επομένως εάν ο παίκτης δεν εισάγει τα απαραίτητα στοιχεία, εμφανίζεται σχετικό μήνυμα (πχ Εικόνα 8) και το παιχνίδι παραμένει στην αρχική οθόνη, μέχρι να συμπληρωθούν τα συγκεκριμένα πεδία.



Εικόνα 8: Παράληψη συμπλήρωσης υποχρεωτικών πεδίων από το χρήστη

Σε επόμενο επίπεδο, πραγματοποιήθηκε έλεγχος των λειτουργιών της εφαρμογής, σύμφωνα με τις απαιτήσεις που τέθηκαν. Πιο συγκεκριμένα, έγινε έλεγχος των παρακάτω περιπτώσεων, έτσι ώστε να ικανοποιούν τις προϋποθέσεις που τέθηκαν:

- Τερματισμός του παιχνιδιού έπειτα από δέκα γύρους.
- Τερματισμός του παιχνιδιού με σχετικό μήνυμα εάν ο παίκτης δεν έχει συγκεντρώσει τουλάχιστον έξι σωστές απαντήσεις σε μια κατηγορία.
- Τερματισμός του παιχνιδιού με σχετικό μήνυμα εάν ο παίκτης έχει ξεπεράσει το χρόνο.
- Σε περίπτωση επίτευξης υψηλότερου σκορ γίνεται εγγραφή του τελικού σκορ του χρήστη στο αρχείο με τις δέκα υψηλότερες βαθμολογίες, έπειτα από εμφάνιση σχετικού μηνύματος.

4. Προτάσεις για περαιτέρω ανάπτυξη

Ένα παιχνίδι γνώσεων μπορεί να υλοποιηθεί με πολλούς διαφορετικούς τρόπους και να προσφέρει πολυάριθμες λειτουργίες.

Ένας τρόπος με τον οποίο θα μπορούσε να επεκταθεί η εφαρμογή είναι η υποστήριξη περισσότερων από ενός παίκτη. Για παράδειγμα, θα μπορούσαν να υπάρχουν δύο ή παραπάνω παίκτες και εκείνος με το μεγαλύτερο σκορ στο τέλος του παιχνιδιού θα είναι ο νικητής.

Επιπλέον λειτουργίες που θα μπορούσαν να προστεθούν είναι οι εξής:

- Δυνατότητα δημιουργίας προφίλ χρηστών (με τη χρήση μιας βάσης δεδομένων).
- Εμφάνιση παραθύρου αναλυτικών αποτελεσμάτων.
- Ενδιαφέρουσες προκλήσεις και επιβραβεύσεις για τους χρήστες.
- Σύνδεση με κοινωνικά δίκτυα.

5. Επιπλέον πληροφορίες

- Github link: [georgia-koukoutou/trivia \(github.com\)](https://github.com/georgia-koukoutou/trivia)