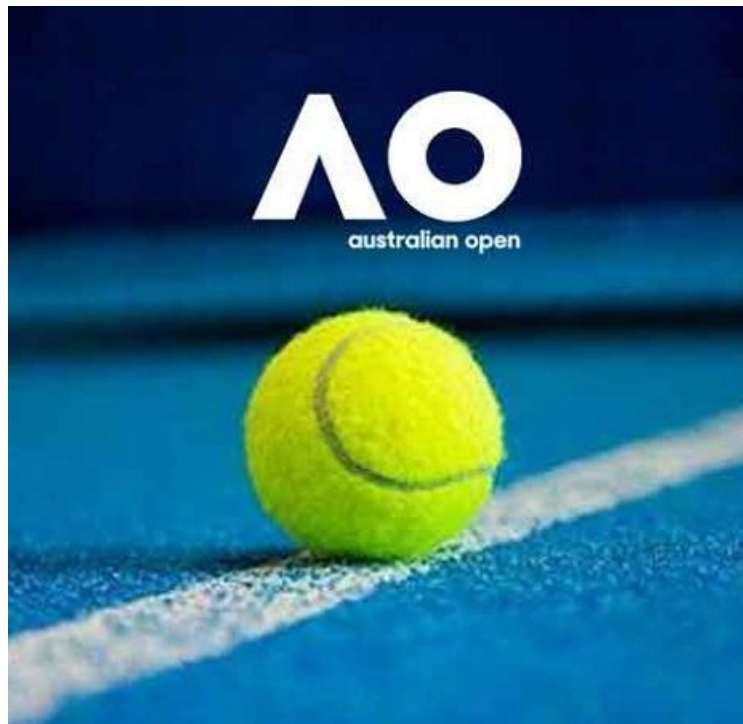





# Australian Open (AUSOPEN)



*Ομαδικό Προγραμματιστικό Project*  
*Θεματική Ενότητα: ΠΛΗΠΡΟ*

Ομάδα 03 – Συμμετέχοντες Φοιτητές:

-  Αλέξανδρος Νάκος (L)
-  Φώτης Γιαλαμάς
-  Καραμπίνης Ιωάννης

2021 - 2022

## Περιεχόμενα

Εισαγωγή .....	3
Στόχος .....	3
Περιγραφή του προβλήματος.....	3
Ενότητες Project .....	4
Αντληση Δεδομένων .....	4
Ομαδοποίηση Παικτών – Κλήρωση .....	5
Αλγόριθμος Κλήρωσης.....	5
Γραφική Διεπαφή – Αλληλεπίδραση με τον χρήστη .....	6
MainPage .....	6
PlayerList .....	7
GamesNoCanvasPage .....	8
ScoresPage .....	9
Περιορισμοί .....	10
Σύνοψη.....	12
Βιβλιογραφία .....	13

## Εισαγωγή

Η παρούσα αναφορά αφορά στο προγραμματιστικό project που εκπονήθηκε στα πλαίσια της εκπαιδευτικής διαδικασίας της εργαστηριακής θεματικής ενότητας «ΠΛΗΠΡΟ» του προγράμματος «ΠΛΗΡΟΦΟΡΙΚΗ» του Ελληνικού Ανοικτού Πανεπιστημίου.

Στη συγκεκριμένη εκπαιδευτική διαδικασία η ομάδα των φοιτητών καλείται να δομήσει ένα ολοκληρωμένο πρόγραμμα που να περιλαμβάνει τόσο Front-end όσο και Back-end υλοποίηση που αφορά στο τουρνουά τένις της Μελβούρνης (*Australian Open*).

Αναφορικά με το τουρνουά τένις της Μελβούρνης, είναι ένα από τα 4 σημαντικότερα τουρνουά τένις στο ετήσιο ημερολόγιο, δίνοντας 2000 βαθμούς και ένα πολύ μεγάλο χρηματικό έπαθλο στον νικητή, καθώς και ανάλογους βαθμούς και χρήματα στους συμμετέχοντες στους προημιτελικούς, ημιτελικούς και τον τελικό αγώνα.

## Στόχος

Στόχος της εργασίας είναι η υλοποίηση ενός ολοκληρωμένου προγράμματος με γραφική διεπαφή σε Python το οποίο θα αντλεί τα δεδομένα παικτών του αυστραλιανού πρωταθλήματος τένις από το σχετικό ιστότοπο και θα τα αποθηκεύει σε μια βάση δεδομένων. Στη συνέχεια ο χρήστης θα δύναται να μπορεί να αντλήσει πληροφορίες για τους παίκτες του τουρνουά καθώς επίσης θα δύναται να υλοποιηθεί η κλήρωση των αγώνων του τουρνουά η οποία θα προβάλλεται στο αντίστοιχο δέντρο αγώνων σε σχετικό παράθυρο.

## Περιγραφή του προβλήματος

Το πρόβλημα υλοποίησης χωρίζεται εννοιολογικά σε τρεις (3) θεματικές ενότητες:

- **Αντληση Δεδομένων** από το διαδίκτυο και καταχώριση σε βάση δεδομένων
- **Ομαδοποίηση Παικτών – Κλήρωση** και καταχώριση στη βάση δεδομένων
- **Ανάπτυξη Γραφικής Διεπαφής** – Αλληλεπίδραση με τον χρήστη

Για κάθε μία από τις ως άνω ενότητες δίνονται πληροφορίες που αφορούν στην πορεία που ακολουθήθηκε από την ομάδα για την υλοποίησή τους.

## Ενότητες Project

### Άντληση Δεδομένων

Πηγή των δεδομένων για την εφαρμογή αποτέλεσε ο επίσημος ιστότοπος του αυστραλιανού πρωταθλήματος τένις που είναι προσβάσιμη στον ακόλουθο σύνδεσμο: <https://ausopen.com>. Αρχικά για την άντληση των δεδομένων υπεύθυνο είναι το πηγαίο αρχείο “DataParser.py”. Στο συγκεκριμένο αρχείο χρησιμοποιείται η βιβλιοθήκη requests της python με την οποία μπορούν να σταλούν αιτήματα τύπου “get” σε συνδέσμους ώστε να αντληθούν τα δεδομένα του παρεχόμενου “link” μέσω “APIs” υπό τη μορφή αρχείων “JSON”.

Κατά τη φόρτωση της σελίδας παρατηρήθηκε ότι η πληροφορία φορτώνεται ασύγχρονα για τους αθλητές οπότε μέσω της κονσόλας του “browser” εντοπίστηκε “object” τύπου “xhr” για τη φόρτωση των παικτών όπου υπήρχε το “API” με τα δεδομένα αυτών. Εκεί βρέθηκαν τα ακόλουθα “links” που χρησιμοποιήθηκαν από την εφαρμογή για να ληφθούν τα δεδομένα των παικτών:

‘https://ausopen.com/event/195321/players?\_format=json’

Από τον παραπάνω σύνδεσμο λαμβάνονται τα ids για κάθε παίκτη (tour\_ids) και στη συνέχεια μέσω συνένωσης αλφαριθμητικού (string concatenation) αναπαράγεται το αντίστοιχο link για κάθε παίκτη όπως φαίνεται στη συνέχεια:

‘https://ausopen.com/sites/default/files/player\_json/' + curr\_tour\_id + '.json’

Στη συνέχεια από τα κατάλληλα πεδία των αρχείων JSON λήφθηκαν τα δεδομένα για κάθε παίκτη και κατόπιν προστίθενται στη βάση δεδομένων τα ακόλουθα πεδία:

firstname, lastname, yearofbirth, birthplace, sglrank

Τέλος, λόγω του γεγονότος ότι η αρίθμηση των παικτών δεν είναι από 1-128, δημιουργείται μία επιπρόσθετη στήλη με όνομα “actual\_rank” στην οποία γίνεται κανονικοποίηση του rank των παικτών με query ταξινόμησης των παικτών και εκχώρησης εκ νέου των αριθμών κατάταξης από 1 έως 128. Αυτό το πεδίο χρησιμοποιείται στη συνέχεια ως το αναγνωριστικό για κάθε παίκτη (player\_id).

## Ομαδοποίηση Παικτών – Κλήρωση

Η εκτέλεση μεταβαίνει πλέον στο αρχείο “Draw.py” το οποίο είναι υπεύθυνο για τη δημιουργία του πίνακα Matches στη βάση δεδομένων, όπου δημιουργούνται τα groups των παικτών, διενεργείται η κλήρωσή τους με τυχαίο τρόπο και κατόπιν η καταχώρηση του πρώτου γύρου των αγώνων στη βάση δεδομένων. Κάθε αγώνας περιλαμβάνει τα ακόλουθα πεδία ή στήλες στη βάση δεδομένων:

Id, Player1Id, Player2Id, Player1Score, Player2Score

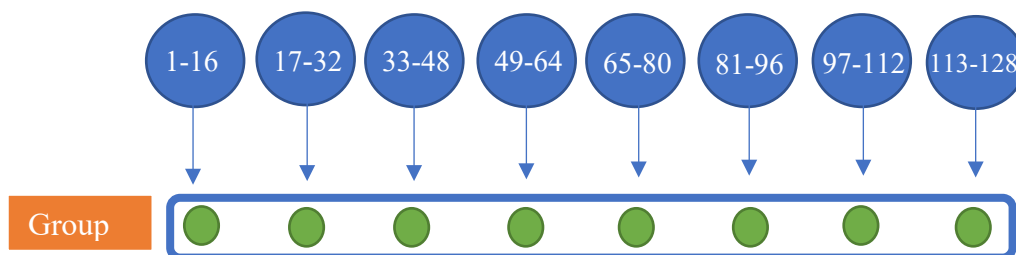
Το Id του κάθε match του πρώτου γύρου δημιουργείται για κάθε αγώνα σύμφωνα με το ακόλουθο πρότυπο:

[A/A Γύρου] . [A/A Γκρουπ] . [A/A Παιχνιδιού]

Σε αυτό το σημείο είναι έτοιμο το πρόγραμμα να δεχτεί την αλληλεπίδραση από το χρήστη για την εκχώρηση των αποτελεσμάτων, να ενημερώσει τη βάση για το αποτέλεσμα και να βρει τον νικητή, διαδικασία που εκτελείται και θα αναλυθεί σε επόμενο στάδιο.

## Αλγόριθμος Κλήρωσης

Η πρώτη ομαδοποίηση των παικτών δημιουργείται με βάση το actual\_rank τους έτσι ώστε να δημιουργηθούν τα ακόλουθα group όπως ακριβώς ζητείται και στις απαιτήσεις του project. Σχηματικά αυτό θα μπορούσε να αναπαρασταθεί από τα ακόλουθα data-pools ως εξής:



Στη συνέχεια καθένα από αυτά τα pools των παικτών ανακατεύεται με τη χρήση της shuffle και χρησιμοποιούνται ως στοίβες από τις οποίες κάθε φορά αντλείται ένας παίκτης. Έτσι δημιουργείται σε κάθε επανάληψη κάθε group των 8 παικτών. Στη συνέχεια επιλέγεται να ξανά-ανακατευτεί το κάθε group των 8 παικτών και κατόπιν τα groups εισέρχονται στη βάση με το πρότυπο που αναφέρθηκε νωρίτερα σε αυτή την ενότητα.

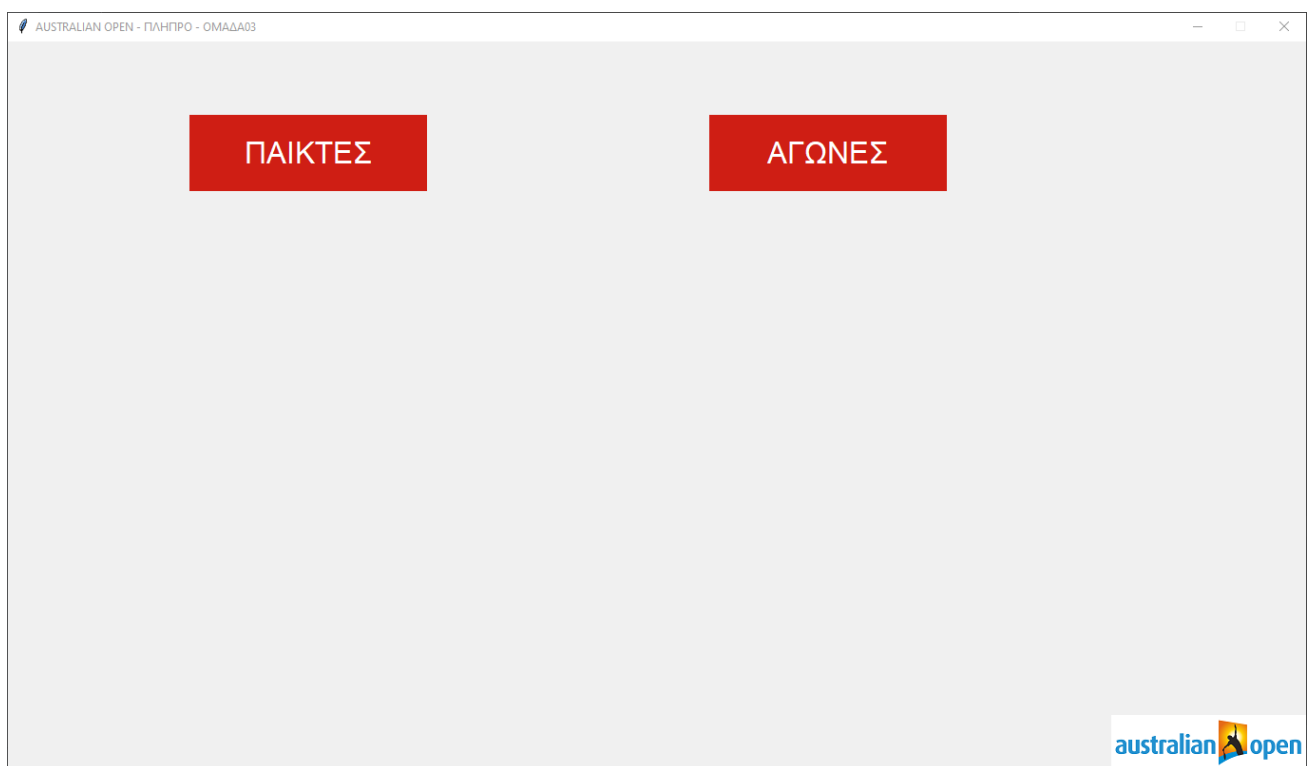
## Γραφική Διεπαφή – Αλληλεπίδραση με τον χρήστη

### MainPage

Για την ανάπτυξη της γραφικής διεπαφής χρησιμοποιήθηκε η βιβλιοθήκη Tkinter της Python. Αρχικά δημιουργείται το object της κλάσης MainPage στο αντικείμενο root του Tkinter. Στη MainPage είναι τοποθετημένα δύο κύρια κουμπιά:

- Παίκτες,
- Αγώνες

τα οποία παρουσιάζονται στο ακόλουθο στιγμιότυπο:



## PlayerList

Σε περίπτωση που επιλεγεί το κουμπί των παικτών το παράθυρο που εμφανίζεται φαίνεται στο ακόλουθο στιγμιότυπο:

Οι λειτουργίες που υπάρχουν στην εν λόγω ενότητα είναι οι ακόλουθες:

### *Αλγόριθμος Σχηματισμού του Δέντρου Τουρνουά*

Για το σχηματισμό του δέντρου των αγώνων σχεδιάστηκε ο παρακάτω αλγόριθμος:

1. Εξωτερική επαναληπτική δομή που διατρέχει τους γύρους. Οι γύροι αντιστοιχούν στις στήλες του δέντρου του τουρνουά που παριστάνονται από τη μεταβλητή `_columns`.
2. Εντός της επαναληπτικής δομής υπολογίζεται το πλήθος των αγώνων του γύρου. Το πλήθος αγώνων αντιστοιχεί στις γραμμές του δέντρου τουρνουά που παριστάνονται από την μεταβλητή `row`.
3. Εσωτερική επαναληπτική δομή που διατρέχει το πλήθος αγώνων-γραμμές στο δέντρο του τουρνουά για τον κάθε γύρο. Το πλήθος αγώνων αλλάζει ως εξής σε κάθε γύρο: 64, 32, 16, 8, 4, 2, 1.
4. Εντός της εσωτερικής επαναληπτικής δομής χρησιμοποιείται αλγόριθμος υπολογισμού του αναγνωριστικού αριθμού του κάθε παιχνιδιού (`match_id`). Οι αριθμοί των αγώνων ακολουθούν τη φιλοσοφία: 1.1.1 – 1.16.4, 2.1.1 – 2.16.4 κτλ. Αφού βρεθεί το `match_id` τότε εκτελείται ένα query που ζητάει από τη βάση να επιστρέψει την αντίστοιχη εγγραφή και να επιστρέψει τις μεταβλητές `name1`, `name2`, `score1`, `score2`. Εάν δεν υπάρχει εγγραφή οι μεταβλητές παίρνουν την τιμή “N/A”.
5. Στη συνέχεια δημιουργείται γραφικά η καταχώρηση του παιχνιδιού στο δέντρο του τουρνουά με τη χρήση της μεθόδου `create_players_rect`, η οποία σε περίπτωση που υπάρχουν οι εγγραφές για τους παίκτες που συμμετέχουν στον αγώνα ενεργοποιεί και το κουμπί του `score` ώστε να δύναται από τον χρήστη να καταχωρήσει το αποτέλεσμα.
6. Αφού ολοκληρωθεί η παραπάνω διαδικασία λήγει ο βρόγχος της εσωτερικής επαναληπτικής δομής και στη συνέχεια τίθενται μεταβλητές που θα βοηθήσουν στο να μετατοπίσουν στη σωστή στήλη και γραμμή του δέντρου του τουρνουά τα παιχνίδια κάθε επόμενου γύρου ώστε να σχηματιστεί σωστά το δέντρο.



ScoresPage

Εκχώρηση του score

## Περιορισμοί

Ο αρχικός σχεδιασμός του project εμπεριείχε ένα αντικείμενο τύπου Canvas που περιλάμβανε τη δημιουργία ορθογωνίων πλαισίων για το σχεδιασμό του δέντρου του τουρνουά. Στη συνέχεια πάνω σε αυτά τα πλαίσια θα έπρεπε να τοποθετηθεί το κουμπί με το οποίο θα ενεργοποιείται το παράθυρο εκχώρησης score. Λόγω του μεγάλου μεγέθους του δέντρου απαιτείται σε κάθε υλοποίηση η χρήση scrollbar κατά την κατακόρυφη διεύθυνση, διαφορετικά θα υπήρχαν προβλήματα στην αναγνωσιμότητα των labels εάν επιλεγόταν να χρησιμοποιηθεί συγκεκριμένη διάσταση που κάλυπτε όλη την οθόνη χωρίς τη χρήση scrollbar.

Ένας από τους βασικότερους περιορισμούς που συναντήθηκαν κατά τη διάρκεια σχεδιασμού αυτού του σταδίου του project ήταν η δημιουργία δεσμού του κάθε κουμπιού με τις θέσεις των ορθογωνίων για να κινούνται τα κουμπιά μαζί με το canvas κατά την κίνηση του scrollbar.

Το πρόβλημα αντιμετωπίστηκε με τη χρήση αποκλειστικά Frame στο Tkinter στο οποίο τοποθετούνται όλα τα widgets. Το Frame διαθέτει εγγενή σύνδεση με το αντικείμενο τύπου Scrollbar και αποτέλεσε την καλύτερη δυνατή επιλογή δεδομένου του διατιθέμενου χρόνου για την αντιμετώπιση του προβλήματος.

Στην πηγή [1] παρατίθεται βίντεο στο οποίο αναφέρεται ο συγκεκριμένος περιορισμός και το γεγονός ότι αποτελούσε γενικότερο πρόβλημα στην κοινότητα. Αναφέρονται και πιθανοί τρόποι παράκαμψης του προβλήματος και όχι αντιμετώπισης αυτού, επομένως επιλέχθηκε να χρησιμοποιηθούν αντικείμενα με πλήρη συμβατότητα μεταξύ τους.





## Σύνοψη

## Βιβλιογραφία

- [1] Codemy.com, «YouTube,» 17 6 2022. [Ηλεκτρονικό]. Available:  
<https://www.youtube.com/watch?v=0WafQCaok6g>.