

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
«Εφαρμογή παρακολούθησης θέσεων παικτών σε
γήπεδο»



Του φοιτητή
Σακελλαρίου Γεώργιου
Αρ. Μητρώου: 174935

Επιβλέπων
Δεληγιάννης Ιγνάτιος
Βαθμίδα Καθηγητής

Ημερομηνία 16-06-2022

Τίτλος Δ.Ε. Λογισμικό Ανάλυσης Ποδοσφαιρικού Αγώνα

Κωδικός Δ.Ε. 21295

Όνοματεπώνυμο φοιτητή/τών: Σακελλαρίου Γεώργιος

Όνοματεπώνυμο εισηγητή: Δεληγιάννης Ιγνάτιος

Ημερομηνία ανάληψης Δ.Ε. 10-10-2021

Ημερομηνία περάτωσης Δ.Ε. 16-06-2021

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Σακελλαρίου Γεωργίου που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιερώνω τη πτυχιακή εργασία στην οικογένεια μου, και στον υπεύθυνο καθηγητή μου που με στήριξαν και βοήθησαν με τον δικό τους τρόπο στην ολοκλήρωση αυτής της εργασίας».

Πρόλογος

Σε μία εποχή σαν την σημερινή είναι εύκολο να παρατηρήσει κάποιος ότι ο αθλητισμός εντάσσεται όλο και περισσότερο στις ζωές των ανθρώπων, είτε αυτό γίνεται με άμεσο τρόπο είτε με έμμεσο. Αυτό έχει ως αποτέλεσμα την κορύφωση του ανταγωνισμού των διάφορων ομάδων ειδικά στο άθλημα του ποδοσφαίρου. Πλέον κάθε λεπτομέρεια μετράει, είναι πολύ σημαντικό να ξέρει κάποιος την οποιαδήποτε κίνηση της ομάδας του, και την χρονική στιγμή που πραγματοποιήθηκε αυτή, ώστε να πάρει τις κατάλληλες αποφάσεις. Έτσι πολλοί από τους προπονητές έχουν αρχίσει να συνδυάζουν την τεχνολογική πρόοδο με τους παραδοσιακούς τρόπους προπόνησης των αθλητών ώστε να πετύχουν το καλύτερο δυνατό αποτέλεσμα και να οδηγήσει την ομάδα του στην νίκη.

Περίληψη

Η συγκεκριμένη πτυχιακή εργασία έχει ως βασικό στόχο να φέρει στις μικρότερες αθλητικές διοργανώσεις ένα εργαλείο ανάλυσης ποδοσφαιρικών αγώνων. Υπάρχουν αρκετές εφαρμογές που αποσκοπούν στο ίδιο αποτέλεσμα, όπου πολλές από αυτές χρησιμοποιούν διαφορετικές προσεγγίσεις, ωστόσο είναι κατά βάση συνδρομητικές και συνήθως αξιοποιούν ανθρώπινο δυναμικό κάνοντας το κόστος αυτών αρκετά υψηλό. Έτσι οι μικρότερες ομάδες δεν μπορούν βάση του χαμηλότερου προϋπολογισμού τους να αποκτήσουν τέτοιου είδους λογισμικό, και ως εκ τούτου δεν μπορούν να προετοιμαστούν κατάλληλα σε σύγκριση με τις μεγαλύτερες ομάδες πράγμα που κάνει τον αθλητισμό και το άθλημα λιγότερο δίκαιο. Ο τρόπος που το πετυχαίνει αυτό είναι αρκετά απλός. Δίνει στον εκάστοτε προπονητή την δυνατότητα να προσθέτει μόνος του τις ενέργειες που έχει πραγματοποιήσει ο κάθε παίχτης κατά την διάρκεια του αγώνα χωρίς να χρειάζεται να μεσολαβήσει κάποιος τρίτος σε αυτό. Παράλληλα μέσω υπολογιστικής όρασης και μηχανικής μάθησης το πρόγραμμα πραγματοποιεί κάποιες αυτόματες ενέργειες για να επισπεύσει την διαδικασία αυτή έτσι ώστε ο προπονητής να βγάλει ένα πόρισμα σε όσο το δυνατόν συντομότερο χρονικό διάστημα.

Abstract

The main goal of this thesis is to bring a football analysis tool to smaller football teams. There are several applications with the same goal, many of which use different approaches; however, they are mostly subscription and usually utilize human resources making their costs quite high. So, the smaller teams cannot afford this kind of software based on their lower budget, and therefore cannot be properly prepared compared to the larger teams, which makes the sport less fair. The way to achieve this goal is quite simple. It gives the respective coach himself the ability to add the actions that each player has performed during the match without the need for a third party to mediate in it. At the same time, through computer vision and machine learning, the program performs some automatic actions to speed up this process so that the coach can draw a conclusion in the shortest possible time.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την κοινότητα του ανοιχτού κώδικα που έχει καταφέρει να κάνει δυνατή την πραγματοποίηση οποιασδήποτε ιδέας. Χωρίς την καθημερινή συμβολή ανθρώπων τέτοιων κοινοτήτων ο προγραμματισμός θα ήταν λιγότερο προσβάσιμος.

Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract	vii
Ευχαριστίες	viii
Περιεχόμενα	ix
Κατάλογος Σχημάτων	xii
Συντομογραφίες.....	xiii
Κεφάλαιο 1ο: Η μεθοδολογία προγραμματισμού Scrum	14
1.1 Εισαγωγή.....	14
1.2 Τι είναι τοScrum;.....	14
1.3 Το Scrum Framework.....	14
1.4 Scrum artifacts.....	15
Κεφάλαιο 2ο: Η γλώσσα προγραμματισμού Python	16
2.1 Εισαγωγή.....	16
2.2 Γενικές πληροφορίες για την Python	16
2.3 Η φιλοσοφία της Python.....	16
2.4 Οι βιβλιοθήκες της Python	17
2.4.1 PySide6.....	17
2.4.2 Η βιβλιοθήκη Pandas.....	18
2.4.3 Η βιβλιοθήκη OpenCV	19
2.4.4 Η βιβλιοθήκη PyTube	19
2.4.5 Το API Pyplot.....	20
2.4.6 Η βιβλιοθήκη NumPy.....	20
2.4.7 Η βιβλιοθήκη PyTorch	21
2.5 Επίλογος.....	21
Κεφάλαιο 3ο: Το γραφικό περιβάλλον της εφαρμογής.....	21
3.1 Εισαγωγή.....	21
3.2 Ανάλυση του γραφικού περιβάλλοντος της εφαρμογής.....	22
3.2.1 Μενού μετακίνησης.....	22
3.2.2 Η κεφαλίδα της εφαρμογής.....	23
3.2.3 Το κεντρικό γραφικό στοιχείο της εφαρμογής.....	23
3.2.4 Το υποσέλιδο της εφαρμογής.....	24

3.2.5	Γραφικά στοιχείο επισύναψης βίντεο.....	24
3.2.6	Δημιουργία στρατηγικής – Lineup Builder.....	26
3.2.7	Δημιουργία στατιστικών	27
3.2.8	Εμφάνιση στατιστικών	29
Κεφάλαιο 4ο:	Οδηγίες εγκατάστασης.....	32
4.1	Εισαγωγή.....	32
4.2	Εγκατάσταση Python.....	32
4.3	Εγκατάσταση Git.....	32
4.4	Λήψη του κώδικα της εφαρμογής	32
4.5	Λήψη βιβλιοθηκών και εκκίνηση της εφαρμογής.....	33
Κεφάλαιο 5ο:	Ανάλυση Κώδικα	33
5.1	Εισαγωγή.....	33
5.2	Η δομή του Project	33
5.3	Δημιουργία του User Interface	35
5.3.1	Ο φάκελος Modules.....	35
5.4	Ανάλυση του save_data.py.....	36
5.4.1	Εισαγωγή.....	36
5.4.2	Ανάλυση Μεθόδων.....	36
5.5	Ανάλυση του csv_calculations.py	37
5.5.1	Εισαγωγή.....	37
5.5.2	Ανάλυση μεθόδων	37
5.6	Ανάλυση του youtube_downloader.py	38
5.6.1	Εισαγωγή.....	38
5.7	Ανάλυση του string_manipulation.py	38
5.7.1	Ανάλυση Μεθόδων.....	39
5.8	Ανάλυση του filesystem_changes.py	40
5.8.1	Εισαγωγή.....	40
5.8.2	Ανάλυση Μεθόδων.....	40
5.9	Ανάλυση του φακέλου football_formation_creator	41
5.9.1	Εισαγωγή.....	41
5.9.2	Ανάλυση Αρχείων	42
5.10	Ανάλυση του φακέλου player_detection.....	42
5.10.1	Εισαγωγή.....	42
5.10.2	Λεπτομέρειες για την Αναγνώριση Παιχτών	43
5.11	Ανάλυση του track_players.py	44

5.11.1	Εισαγωγή.....	44
5.11.2	Ανάλυση Μεθόδων.....	44
5.12	Το αρχείο distance.py.....	45
5.12.1	Εισαγωγή.....	45
5.12.2	Ανάλυση Μεθόδων.....	45
5.13	Το αρχείο zoom_into_video.py.....	47
5.13.1	Εισαγωγή.....	47
5.13.2	Ανάλυση Μεθόδων.....	47
5.14	Το αρχείο graph_generator.py.....	48
5.14.1	Εισαγωγή.....	48
5.14.2	Ανάλυση Μεθόδων.....	48
5.15	Ανάλυση της main.py.....	49
5.15.1	Εισαγωγή.....	49
5.15.2	Ανάλυση των import.....	50
5.15.3	Οι global μεταβλητές.....	50
5.15.4	Ανάλυση της μεθόδου __init__().....	50
5.15.5	Ανάλυση της μεθόδου buttonClick().....	51
5.15.6	Ανάλυση των μεθόδων mousePressEvent() και keyPressEvent().....	53
5.15.7	Οι Μέθοδοι change_actions(), change_timestamps(), delete_player_names().....	53
5.15.8	Η μέθοδος on_load_video_request().....	54
5.15.9	Η μέθοδος set_minute().....	54
5.15.10	Η μέθοδος fix_audio().....	54
5.15.11	Η μέθοδος slider_moved().....	54
5.15.12	Η μέθοδος fill_tables().....	54
5.15.13	Η μέθοδος show_diagram().....	55
5.15.14	Η μέθοδος on_download_requested() και resizeEvent().....	55
Κεφάλαιο 6ο:	Συμπεράσματα ή/και προτάσεις βελτίωσης.....	55
BIBΛΙΟΓΡΑΦΙΑ.....		58

Κατάλογος Σχημάτων

Σχήμα 1.1: Scrum Methodology Diagram.....	15
Σχήμα 3.1: Application's Navigation Menu	22
Σχήμα 3.2: Application's Header.....	23
Σχήμα 3.3: Application's Main Widget	23
Σχήμα 3.4: Application's footer.....	24
Σχήμα 3.5: Import Video Widget	25
Σχήμα 3.6: Import Video Locally.....	25
Σχήμα 3.7: Import Cloud Video	26
Σχήμα 3.8: Lineup Builder	27
Σχήμα 3.9: Statistics Generator	27
Σχήμα 3.10: Player Detection Analyzed Video.....	28
Σχήμα 3.11: Zoom Into Player	29
Σχήμα 3.12: Statistics Showcase	30
Σχήμα 3.13: Player Statistics Chart.....	30
Σχήμα 3.14: Action Family Chart	31
Σχήμα 3.15: Action Per Player	31
Σχήμα 3.16: All Actions Chart	32
Σχήμα 5.1: Project Content Root.....	34
Σχήμα 5.2: Modules Directory:.....	35
Σχήμα 5.3: Lineup Builder Directory	42
Σχήμα 5.4: Correlogram Image of Detection	44
Σχήμα 5.5: Deepsort & YoloV5 text file:	45
Σχήμα 5.6: Cleaned Deepsort & YoloV5 Data	46
Σχήμα 5.7: Distance Between Players.....	46
Σχήμα 5.8: MainWindow() Structure	49

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

Κεφάλαιο 1ο: Η μεθοδολογία προγραμματισμού Scrum

1.1 Εισαγωγή

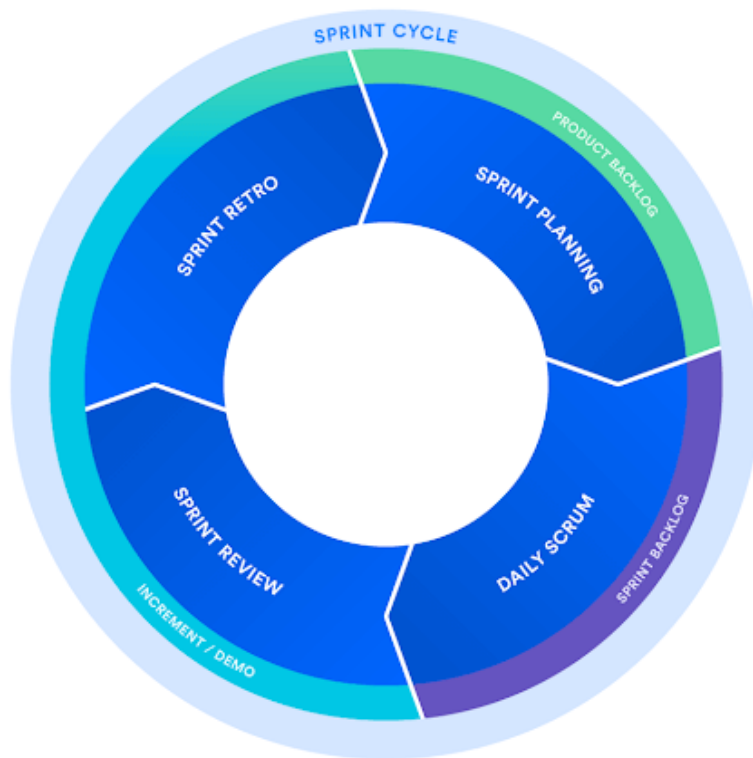
Για την ολοκλήρωση της συγκεκριμένης πτυχιακής χρησιμοποιήθηκε ένα πλαίσιο (framework) για την ανάπτυξη, παράδοση και την διατήρηση των λειτουργιών σε ένα σχετικά σύνθετο περιβάλλον με όνομα Scrum.

1.2 Τι είναι τοScrum;

Το Scrum είναι ένα πλαίσιο που έχει σχεδιαστεί για ομάδες μικρότερες των δέκα ατόμων ενώ πολλές φορές χρησιμοποιείται και από ακόμη λιγότερα άτομα σε ομαδικές εργασίες. Αυτός είναι ένας από τους λόγους που το scrum είναι τόσο δημοφιλές. Συχνά θεωρείται ως ένα ευέλικτο πλαίσιο διαχείρισης έργου. Το scrum περιγράφει ένα σύνολο συναντήσεων, εργαλείων και ρόλων που λειτουργούν από κοινού για να βοηθήσουν τις ομάδες να δομήσουν και να διαχειριστούν τις εργασίες τους.

1.3 Το Scrum Framework

Οι άνθρωποι συχνά πιστεύουν ότι το scrum και το agile είναι το ίδιο πράγμα, επειδή το scrum επικεντρώνεται στη συνεχή βελτίωση, η οποία είναι η βασική αρχή του agile. Ωστόσο το scrum είναι ένα πλαίσιο για την εκτέλεση μίας εργασίας ενώ το agile είναι μία νοοτροπία. Το πλαίσιο scrum βασίζεται στη συνεχή μάθηση και την προσαρμογή σε εναλλασσόμενους παράγοντες. Αναγνωρίζει ότι η ομάδα δεν μπορεί να προβλέψει το πως θα εξελιχθεί ένα έργο και είναι δομημένο για να διευκολύνει τις ομάδες να προσαρμοστούν φυσικά στις μεταβαλλόμενες αυτές συνθήκες και στις απαιτήσεις των χρηστών με την εκ νέου ιεράρχηση προτεραιοτήτων ενσωματωμένη μέσα στους σύντομους κύκλους κυκλοφορίας (re-prioritization built into the process and short release cycles) ώστε α μπορεί η ομάδα να βελτιώνεται συνεχώς.



Σχήμα 1.1: Scrum Methodology Diagram

Ενώ το scrum είναι δομημένο δεν είναι τελείως άκαμπτο. Η εκτέλεση του μπορεί να προσαρμοστεί στις ανάγκες οποιουδήποτε οργανισμού. Υπάρχουν πολλές θεωρίες για το πως ακριβώς πρέπει να λειτουργούν οι ομάδες scrum για να είναι επιτυχημένες.

1.4 Scrum artifacts

Στο scrum μπορούμε να ξεχωρίσουμε τρία διαφορετικά artifacts. Τα artifacts είναι κάτι που φτιάχνουμε όπως ένα εργαλείο για την επίλυση ενός προβλήματος. Αυτά τα artifacts είναι τα εξής:

- Το **Product Backlog**, είναι η κύρια λίστα που πρέπει να ολοκληρωθεί και διατηρείται κατά κύριο λόγο από τον product manager. Αυτή είναι μία δυναμική λίστα δυνατοτήτων, απαιτήσεων, βελτιστοποιήσεων και επιδιορθώσεων που λειτουργεί ως είσοδος για το ανεκτέλεστο sprint. Είναι ουσιαστικά η λίστα των «To-Do» της ομάδας. Το ανεκτέλεστο προϊόν επανεξετάζεται συνεχώς, ιεραρχείται εκ νέου και διατηρείται από τον ιδιοκτήτη του προϊόντος επειδή η αγορά αλλάζει συνεχώς, και πρέπει τα σχετικά προβλήματα που προκύπτουν να αλλάζουν συνεχώς.
- Το **Sprint Backlog**, είναι η λίστα με στοιχεία, ιστορίες χρήστη (user stories), ή διορθώσεις σφαλμάτων που επιλέγονται από την ομάδα ανάπτυξης για εφαρμογή στον τρέχοντα κύκλο sprint. Πριν από κάθε sprint στην συνάντηση προγραμματισμού sprint η ομάδα ορίζει σε ποια αντικείμενα από το product backlog του sprint θα εργαστεί. Ένα sprint backlog μπορεί να είναι ευέλικτο και μπορεί να εξελιχθεί κατά τη διάρκεια ενός sprint.

- Το **Increment** (ή αλλιώς sprint goal), είναι το χρήσιμο τελικό προϊόν ενός sprint. Κάθε Increment προστίθεται στις ήδη υπάρχουσες αυξήσεις και επαληθεύεται διεξοδικά. Με αυτό τον τρόπο διασφαλίζεται ότι όλες οι προσαυξήσεις λειτουργούν μαζί. Μπορούν να δημιουργηθούν πολλαπλές προσαυξήσεις σε ένα sprint. Το άθροισμα των προσαυξήσεων παρουσιάζεται στο sprint review υποστηρίζοντας έτσι τον εμπειρισμό. Ωστόσο μία προσαύξηση μπορεί να παραδοθεί στους ενδιαφερόμενους πριν το τέλος του sprint.

Κεφάλαιο 2ο: Η γλώσσα προγραμματισμού Python

2.1 Εισαγωγή

Στην συγκεκριμένη εργασία χρησιμοποιήθηκε κατά κύριο λόγο η γλώσσα προγραμματισμού Python.

2.2 Γενικές πληροφορίες για την Python

Η Python είναι μία διερμηνευόμενη (interpreted) γλώσσα προγραμματισμού, γενικού σκοπού (general-purpose) και υψηλού επιπέδου. Ανήκει στις γλώσσες προστακτικού προγραμματισμού (imperative programming) και υποστηρίζει τόσο το διαδικαστικό όσο και το αντικειμενοστρεφές (object oriented) προγραμματιστικό υπόδειγμα. Ανήκει στις γλώσσες δυναμικού προγραμματισμού και υποστηρίζει την συλλογή απορριμμάτων (garbage collection).

2.3 Η φιλοσοφία της Python

Ο κύριος στόχος της Python είναι η αναγνωσιμότητα κώδικα και η ευκολία χρήσης της. Το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα απ' ό,τι θα ήταν δυνατόν σε άλλες γλώσσες όπως η C++ ή η Java. Διακρίνεται λόγω του ότι έχει πολλές βιβλιοθήκες που διευκολύνουν αρκετές συνηθισμένες και μη εργασίες και για την ταχύτητα εκμάθησής της. Το κύριο αρνητικό το οποίο φέρει είναι η ταχύτητα της καθώς είναι διερμηνευόμενη γλώσσα προγραμματισμού και όχι μεταγλωττιζόμενη όπως για παράδειγμα η C++. Για αυτό τον λόγο είναι πλήρως ακατάλληλη για γραφή λειτουργικών συστημάτων και εφαρμογών με κύριο στόχο τους την ταχεία εκτέλεση λειτουργιών. Σε περιπτώσεις που η ταχύτητα είναι απαραίτητη μπορεί να γίνει μεταφορά των κρίσιμων σε ταχύτητα μεθόδων σε μονάδες επέκτασης (extension modules) γραμμένες σε γλώσσες όπως C. Εργαλεία όπως το PyPy που είναι ένας *εκείνης της ώρας* (just-in-time) μεταγλωττιστής ή το Cython που πραγματοποιεί άμεσες κλήσεις στο API της C μέσω του διερμηνευτή της Python μειώνουν τον χρόνο εκτέλεσης εφαρμογών.

Οι βασικές αρχές της Python αναφέρονται περιληπτικά στο έγγραφο The Zen of Python και είναι:

- Το όμορφο είναι καλύτερο από το άσχημο
- Το ρητό είναι καλύτερο από το άρρητο
- Το απλό είναι καλύτερο από το σύνθετο
- Το σύνθετο είναι καλύτερο από το περίπλοκο
- Η ευκολία ανάγνωσης κώδικα μετράει

2.4 Οι βιβλιοθήκες της Python

Στην πραγματικότητα μία βιβλιοθήκη είναι μία συλλογή από βιβλία ή ένα δωμάτιο ή χώρος όπου πολλά βιβλία μπορούν να αποθηκευτούν για να χρησιμοποιηθούν αργότερα. Με τον ίδιο τρόπο στον κόσμο του προγραμματισμού μία βιβλιοθήκη είναι μία συλλογή από προ-μεταγλωττισμένους κώδικες (pre-compiled code) οι οποίοι μπορούν να χρησιμοποιηθούν σε ένα πρόγραμμα για συγκεκριμένες καλά καθορισμένες εργασίες. Εκτός από τέτοιου είδους κώδικες, μία βιβλιοθήκη μπορεί να περιέχει πληροφορίες για αυτή (documentation), δεδομένα διαμόρφωσης (documentation data), κλάσεις, τιμές κ.λπ. Μία βιβλιοθήκη της Python είναι ουσιαστικά μία συλλογή μονάδων που σχετίζονται μεταξύ τους. Περιέχει δέσμες κώδικα που μπορούν να χρησιμοποιηθούν επανειλημμένα σε διαφορετικά προγράμματα. Κάνει τον προγραμματισμό απλούστερο και βολικό για τον προγραμματιστή καθώς δεν χρειάζεται να γράφουμε τον ίδιο κώδικα επανειλημμένα για διαφορετικά προγράμματα. Οι βιβλιοθήκες παίζουν ζωτικό ρόλο σε τομείς του προγραμματισμού όπως είναι η Μηχανική μάθηση, στην Επιστήμη Δεδομένων, σε Οπτικοακουστικά προγράμματα κ.α. Παρακάτω παρουσιάζονται οι βιβλιοθήκες της Python που χρησιμοποιήθηκαν καθώς και οι λόγοι που επιλέχθηκαν οι βιβλιοθήκες αυτές.

2.4.1 PySide6

Σε αυτή την εφαρμογή για τη δημιουργία του γραφικού περιβάλλοντος χρήστη χρησιμοποιήθηκε η βιβλιοθήκη PySide6 από την «The Qt Company». Η PySide6 μας παρέχει πρόσβαση στο πλαίσιο (framework) της Qt, καθώς και σε εργαλεία για την γρήγορη δημιουργία δεσμεύσεων για οποιοσδήποτε βιβλιοθήκες της C++. Αναπτύσσεται ως δωρεάν πρόγραμμα ανοιχτού κώδικα (Free Open Source Software - FOSS) και δέχεται διορθώσεις από εκατοντάδες χρήστες σε όλο τον κόσμο.

2.4.1.1 Προβλήματα τρόπου λειτουργίας της PySide

Όταν δημιουργούμε μία σύνθετη εφαρμογή μέσω του PySide, συναντάμε διάφορα προβλήματα με την διατήρηση γραφικών στοιχείων σε συντονισμό με τα δεδομένα της εφαρμογής. Τα δεδομένα που είναι αποθηκευμένα στο γραφικό στοιχείο (widget) δεν είναι άμεσα διαθέσιμα για χειρισμό από την Python. Οι αλλαγές απαιτούν την λήψη ενός αντικειμένου (item) και στην συνέχεια την επαναφορά του. Η πιο διαδεδομένη και πλέον προεπιλεγμένη λύση σε αυτό το πρόβλημα είναι η χρήση μίας εξωτερικής αναπαράστασης δεδομένων στην Python και στην συνέχεια η διπλή ενημέρωση των δεδομένων και γραφικού στοιχείου. Αυτό ωστόσο μπορεί να μην γίνει χρηστικό πολύ γρήγορα με το αποτέλεσμα να είναι η ύπαρξη πολλών αναπαραστάσεων απλώς για την εύρεση δεδομένων.

2.4.1.2 Η λύση στο πρόβλημα

Ευτυχώς το Qt έχει βρει την λύση για το παραπάνω πρόβλημα – ModelViews. Τα ModelViews είναι μία ισχυρή εναλλακτική λύση στα τυπικά γραφικά στοιχεία της οθόνης, τα οποία χρησιμοποιούν μία κανονική διεπαφή μοντέλου για αλληλεπίδραση με τις πηγές δεδομένων – από απλές δομές δεδομένων έως και εξωτερικές βάσεις δεδομένων. Αυτό απομονώνει τα δεδομένα επιτρέποντας τους να διατηρούνται σε οποιαδήποτε δομή, ενώ φροντίζει την εμφάνιση και τις ενημερώσεις.

2.4.1.3 Η αρχιτεκτονική του Qt

Το Model-View-Controller ή εν συντομία MVC είναι ένα αρχιτεκτονικό μοτίβο που χρησιμοποιείται για την ανάπτυξη διεπαφών χρήστη (user interface) που διαχωρίζει μία εφαρμογή σε τρία διασυνδεδεμένα μέρη. Διαχωρίζει τον τρόπο εσωτερικής αναπαράστασης από τον τρόπο παρουσίασης αυτών, και από την αποδοχή πληροφοριών από τον χρήστη. Το σχέδιο σχεδίασης MVC «αποσυνδέει» τρία κύρια στοιχεία με τον εξής τρόπο.

- Το Model διατηρεί τη δομή δεδομένων με την οποία λειτουργεί η εφαρμογή
- Το View είναι οποιαδήποτε αναπαράσταση πληροφοριών όπως φαίνεται στον χρήστη, είτε γραφική είτε είναι με πίνακες. Επιπλέον επιτρέπονται πολλές προβολές του ίδιου μοντέλου δεδομένων
- Το Controller δέχεται δεδομένα από το χρήστη, μετατρέποντας τα σε εντολές για το μοντέλο ή το View

Στο Qt και κατά συνέπεια στην PySide η διάκριση μεταξύ View και Controller γίνεται λιγότερο ευδιάκριτη. Το Qt δέχεται συμβάντα (events) εισόδου από τον χρήστη μέσω του λειτουργικού συστήματος και τα εκχωρεί στα γραφικά στοιχεία (controller) για το χειρισμό. Ωστόσο τα γραφικά στοιχεία χειρίζονται επίσης την παρουσίαση της τρέχουσας κατάστασης στο χρήστη τοποθετώντας τα στο View. Έτσι για να απλοποιηθεί η διαδικασία ενώνεται το View με το Controller δημιουργώντας μία Model/ViewController αρχιτεκτονική με απλοποιημένο όνομα «Model View». Σημαντικό είναι να αναφερθεί ότι διατηρείται η διάκριση μεταξύ δεδομένων και του τρόπου παρουσίασης τους.

2.4.1.4 Model View

Το Model View λειτουργεί ως διεπαφή μεταξύ του χώρου αποθήκευσης δεδομένων και του ViewController. Διατηρεί τα δεδομένα (ή μία αναφορά σε αυτά) και παρουσιάζει αυτά τα δεδομένα μέσω ενός τυποποιημένου API το οποίο χρησιμοποιεί το View για την εμφάνιση στο χρήστη. Τα δεδομένα μπορούν να χρησιμοποιηθούν από πολλαπλά Views όπου το καθένα από αυτά μπορεί να τα παρουσιάσει με τελείως διαφορετικούς τρόπους. Για ένα Model μπορεί να χρησιμοποιηθεί οποιαδήποτε «αποθήκη δεδομένων» (data store), συμπεριλαμβανομένης της τυπικής λίστας ή μιας ολόκληρης βάσης δεδομένων. Τα δύο μέρη είναι ουσιαστικά υπεύθυνα για:

- Το Model αποθηκεύει τα δεδομένα ή μία αναφορά σε αυτά και επιστρέφει μεμονωμένες και μη εγγραφές, καθώς και σχετικά μετα-δεδομένα ή οδηγίες εμφάνισης.
- Το View ζητά δεδομένα από το Model και εμφανίζει ότι επιστρέφεται από το γραφικό στοιχείο (widget).

2.4.2 Η βιβλιοθήκη Pandas

Η βιβλιοθήκη Pandas είναι μία βιβλιοθήκη ανοιχτού κώδικα που χρησιμοποιείται ευρέως για εργασίες επιστήμης και ανάλυσης δεδομένων και μηχανικής μάθησης. Έχει βασιστεί πάνω στην βιβλιοθήκη NumPy η οποία παρέχει υποστήριξη για πολυδιάστατους πίνακες. Ως μία από τις δημοφιλέστερες βιβλιοθήκες που χρησιμοποιούνται για διαχείριση δεδομένων η Pandas λειτουργεί ικανοποιητικά με πολλές άλλες βιβλιοθήκες επιστήμης δεδομένων.

2.4.2.1 Γιατί Pandas;

Τι μπορεί όμως να κάνει η βιβλιοθήκη αυτή; Η Pandas έχει μεγάλο εύρος λειτουργιών, κάποιες από τις σημαντικότερες είναι οι εξής:

- Καθαρισμός δεδομένων
- Συμπλήρωση δεδομένων
- Κανονικοποίηση δεδομένων
- Συγχωνεύσεις και ενώσεις δεδομένων
- Οπτικοποίηση δεδομένων
- Στατιστική ανάλυση
- Επιθεώρηση δεδομένων
- Φόρτωση και αποθήκευση δεδομένων

- Και πολλές άλλες λειτουργίες

Ο βασικός στόχος της βιβλιοθήκης Pandas είναι η ευρεία χρήση της από όλους τους επιστήμονες της ανάλυσης δεδομένων. Επιπλέον προσπαθεί να γίνει όσο το δυνατόν πιο ευέλικτο εργαλείο ανάλυσης και χειρισμού δεδομένων, το οποίο είναι ανοιχτού κώδικα και θα είναι διαθέσιμο σε οποιαδήποτε γλώσσα.

2.4.2.2 Πως λειτουργεί η Pandas;

Η βιβλιοθήκη έχει χτιστεί εξ' ολοκλήρου στον χειρισμό δεδομένων τα οποία βρίσκονται σε δομή τύπου πίνακα. Από πολλές απόψεις μπορεί να θεωρηθεί ως ένα ισχυρό υπολογιστικό φύλλο. Η καλύτερη χρήση του γίνεται κυρίως σε πίνακες δύο διαστάσεων, σε αντίθεση με την NumPy η οποία απευθύνεται σε πίνακες N διαστάσεων. Οι θεμελιώδεις δομές δεδομένων που χρησιμοποιεί η Pandas είναι τα Series και τα DataFrames. Τα Series είναι μία σειρά που περιέχει μία στήλη δεδομένων και ένα ευρετήριο (index). Το ευρετήριο είναι ένας τρόπος αναφοράς της σειράς που περιέχει τα δεδομένα στα Series. Ένα DataFrame μπορεί να θεωρηθεί ότι δημιουργείται συνδυάζοντας πολλαπλά Series που μοιράζονται σε ένα κοινό ευρετήριο. Έτσι ένας πίνακας με πολλές στήλες (columns) και κοινό ευρετήριο μπορεί να αποτελέσει ένα παράδειγμα ενός DataFrame.

2.4.3 Η βιβλιοθήκη OpenCV

Η OpenCV (Open Source Computer Vision Library) είναι μία βιβλιοθήκη ανοιχτού κώδικα που χρησιμοποιείται κατά κύριο λόγο σε εφαρμογές Computer Vision και μηχανικής μάθησης. Σκοπός της δημιουργίας της ήταν η παροχή μίας κοινής υποδομής σε εφαρμογές που έχουν λειτουργίες στις οποίες είναι απαραίτητη η «αίσθηση της όρασης», πράγμα που όπως είναι αναμενόμενο δεν έχει ένας υπολογιστής. Παρέχει περισσότερους από δύο χιλιάδες πεντακόσιους βελτιστοποιημένους αλγορίθμους οι οποίοι περιλαμβάνουν ένα ολοκληρωμένο σύνολο κλασικών και υπερσύγχρονων Computer Vision και Machine Learning αλγορίθμων. Αυτοί οι αλγόριθμοι μπορούν να χρησιμοποιηθούν για την ανίχνευση και την αναγνώριση προσώπων, την αναγνώριση αντικειμένων, την ταξινόμηση διάφορων ανθρώπινων ενεργειών σε βίντεο κ.α. Η OpenCV χρησιμοποιείται εκτενώς από εταιρείες, ερευνητικές ομάδες και από κυβερνητικούς φορείς.

2.4.3.1 Γιατί OpenCV;

Η συνεισφορά της OpenCV στους τομείς του Computer Vision και του Machine Learning είναι μεγάλη και κάποιος από τους λόγους που έχει προτιμηθεί από τους προγραμματιστές και από τις εταιρείες είναι οι εξής:

- Διαθέτει διεπαφές C++, Python, Java, MATLAB
- Υποστηρίζει τα βασικότερα λειτουργικά συστήματα (Windows, Linux, Android, MacOS)
- Κάνει χρήση των οδηγιών MMX και SSE όταν αυτές οι ενέργειες είναι διαθέσιμες
- Διεπαφές CUDA και OpenCL αναπτύσσονται ενεργά αυτή την στιγμή για καλύτερη υποστήριξη των καρτών γραφικών
- Ύπαρξη περισσότερων από πεντακοσίων αλγορίθμων και τουλάχιστον δέκα φορές περισσότερες συναρτήσεις που υποστηρίζουν αυτούς
- Άσφογη λειτουργία με Container STL μιας και η ίδια η βιβλιοθήκη έχει γραφτεί σε C++

2.4.4 Η βιβλιοθήκη PyTube

Η συγκεκριμένη βιβλιοθήκη έχει έναν και μοναδικό στόχο την λήψη βίντεο και ήχου από την δημοφιλή πλατφόρμα του YouTube.

2.4.4.1 Γιατί PyTube;

Οι βασικότεροι λόγοι που η PyTube ξεχωρίζει έναντι των άλλων βιβλιοθηκών της ίδιας λειτουργίας είναι οι εξής:

- Υποστήριξη Progressive & Dash ροών (streams)
- Εύκολη επιστροφή των κλήσεων on_download_progress & on_download_complete
- Ύπαρξη διεπαφής γραμμής εντολών
- Ύπαρξη track support
- Εξαγωγή υποτίτλων σε μορφή .stt (SubRip Subtitles)
- Δυνατότητα λήψης της κεντρικής εικόνας του βίντεο (thumbnail)
- Εκτενώς τεκμηριωμένος πηγαίος κώδικας
- Δεν γίνεται χρήση τρίτων βιβλιοθηκών

2.4.5 Το API Pyplot

Το PyPlot είναι ένα API (Application Programming Interface) για την matplotlib βιβλιοθήκη της Python που είναι ουσιαστικά μία απάντηση ανοιχτού κώδικα στην δημοφιλή γλώσσα της MATLAB. Το Matplotlib είναι μία βιβλιοθήκη για οπτικοποίηση δεδομένων, συνήθως με τη μορφή διάφορων γραφημάτων όπως plots, graphs και charts.

2.4.5.1 Τα θετικά του Pyplot

- Μία διεπαφή τύπου MATLAB η οποία επιτρέπει σε όσους γνωρίζουν MATLAB να προσαρμοστούν εύκολα
- Statefulness, που σημαίνει ότι το Pyplot αποθηκεύει την κατάσταση ενός αντικειμένου όταν σχεδιάζεται για πρώτη φορά. Αυτό είναι απαραίτητο για την χρήση στο ίδιο loop και σε άλλες περιπτώσεις όπου και θα τερματιστεί με την ανάλογη εντολή. Το State μπορεί επίσης να είναι σημαντικό όταν δημιουργούνται συνεχώς πολλά plots.

2.4.5.2 Η δομή του PyPlot

Το PyPlot αποτελείται από μία ιεραρχία αντικειμένων σε κώδικα Python και περιλαμβάνει πολυάριθμες συναρτήσεις που κορυφώνονται από το matplotlib.pyplot. Αυτή η στοίβα μπορεί να θεωρηθεί ότι έχει τρία αλληλεξαρτώμενα επίπεδα

- Scripting επίπεδο – που χρησιμοποιείται για τον καθορισμό ενός σχήματος το οποίο περιέχει ένα ή περισσότερα διαγράμματα τα οποία αποτελούνται από άξονες (δηλαδή άξονας x, y, και σε ορισμένες περιπτώσεις και z)
- Artist επίπεδο – που χρησιμοποιείται για τον χειρισμό στοιχείων μίας πλοκής όπως η σχεδίαση ετικετών (labels), γραμμών κ.α.
- Backend επίπεδο – χρησιμοποιείται για τη μορφοποίηση της γραφικής παράστασης για εμφάνιση σε μία συγκεκριμένη εφαρμογή όπως για παράδειγμα το Jupyter Notebook

2.4.6 Η βιβλιοθήκη NumPy

Η NumPy (Numerical Python) είναι μία βιβλιοθήκη που χρησιμοποιείται κατά βάση σε φόρτο εργασίας που απαιτεί πίνακες. Επιπλέον έχει συναρτήσεις στους τομείς της γραμμικής άλγεβρας, του fourier transformation και matrices. Δημιουργήθηκε το 2005 από τον Travis Oliphant και ο κώδικας διατίθεται ανοιχτά και μπορεί να χρησιμοποιηθεί ελεύθερα. Είναι ένα θεμελιώδες πακέτο για επιστημονικούς υπολογισμούς στην Python, συμπεριλαμβάνοντας την υποστήριξη για πίνακες N-διαστάσεων τύπου αντικειμένου. Επιτρέπει να εκτελούνται πολύπλοκες μαθηματικές πράξεις. Ένας πίνακας NumPy έχει

σχεδιαστεί για να λειτουργεί με μεγάλους πίνακες. Υπάρχουν πολλές υπάρχουσες συναρτήσεις Python που έχουν δημιουργηθεί για την επεξεργασία πινάκων NumPy, με τις πιο σημαντικές να παρέχονται στον επιστημονικό κόσμο μέσω του πακέτου SciPy.

2.4.6.1 Γιατί NumPy;

Στην Python έχουμε λίστες που εξυπηρετούν το σκοπό των πινάκων, αλλά είναι αρκετά αργές στο κομμάτι της επεξεργασίας. Η NumPy στοχεύει να παρέχει ένα αντικείμενο πίνακα που να είναι έως και 50 φορές ταχύτερο από τις παραδοσιακές λίστες Python. Οι πίνακες χρησιμοποιούνται πολύ συχνά στην επιστήμη δεδομένων όπου η ταχύτητα είναι ιδιαίτερα σημαντική. Άλλοι λόγοι που η NumPy έγινε τόσο διαδεδομένη είναι οι εξής:

- Ισχυρός N -διαστάσεων πίνακας
- Εξελιγμένες broadcasting λειτουργίες
- Βασικές συναρτήσεις γραμμικής άλγεβρας
- Βασικοί μετασχηματισμοί Fourier
- Εξελιγμένες δυνατότητες τυχαίων αριθμών
- Εργαλεία για την ενσωμάτωση κώδικα Fortran, C, C++

2.4.7 Η βιβλιοθήκη PyTorch

Η PyTorch (ή απλώς torch) είναι μία βιβλιοθήκη εκμάθησης ανοιχτού κώδικα που χρησιμοποιείται για την ανάπτυξη και εκπαίδευση μοντέλων βαθιάς μάθησης (deep learning) που βασίζονται σε νευρωνικά δίκτυα και αναπτύχθηκε κυρίως από την ερευνητική ομάδα AI του Facebook. Η PyTorch μπορεί να χρησιμοποιηθεί με Python καθώς και με C++, ωστόσο η διεπαφή στην Python είναι αρκετά πιο σωστά δομημένη. Υποστηρίζεται από μεγάλες εταιρείες (Facebook, Microsoft, Salesforce, Uber, κ.α.) είναι εξαιρετικά δημοφιλής στα ερευνητικά εργαστήρια αλλά όχι σε τόσο μεγάλο βαθμό σε επίπεδο παραγωγής (production level) σε αντίθεση με την TensorFlow (υποστηριζόμενο από την Google) που κυριαρχεί σε τέτοιες εφαρμογές. Η κύρια διαφορά της PyTorch σε σύγκριση με άλλα δημοφιλή deep learning framework όπως το TensorFlow που αναφέρθηκε πιο πάνω, είναι ότι δεν χρησιμοποιεί στατικά γραφήματα υπολογισμού (static computation graphs). Βασίζεται σε δυναμικό υπολογισμό (dynamic computation) που παρέχει μεγαλύτερη ευελιξία στη δημιουργία πολύπλοκων αρχιτεκτονικών. Η PyTorch χρησιμοποιεί τις βασικές έννοιες της Python όπως κλάσεις, δομές και βρόγχους υπό όρους που είναι πολύ οικείοι στα μάτια μας, επομένως είναι πολύ πιο εύκολη στην κατανόηση σε σύγκριση με άλλα framework που απαιτούν το δικό τους στυλ προγραμματισμού.

2.5 Επίλογος

Όπως μπορεί να παρατηρήσει κάποιος οι βιβλιοθήκες είναι ένας από τους πυρήνες της κάθε γλώσσας προγραμματισμού. Έτσι η δημιουργία και η συντήρηση τους είναι πολύ σημαντική. Παρακάτω θα γίνει επεξήγηση της χρήσης της κάθε βιβλιοθήκης στην εφαρμογή.

Κεφάλαιο 3ο: Το γραφικό περιβάλλον της εφαρμογής

3.1 Εισαγωγή

Το γραφικό περιβάλλον χρήστη (GUI) είναι μια διεπαφή μέσω της οποίας ένας χρήστης αλληλοεπιδρά με ηλεκτρονικές συσκευές όπως υπολογιστές και smartphone μέσω της χρήσης εικονιδίων, μενού και

άλλων οπτικών ενδείξεων ή αναπαραστάσεων (γραφικά). Τα GUI εμφανίζουν γραφικά πληροφορίες και σχετικά στοιχεία ελέγχου χρήστη, σε αντίθεση με τις διεπαφές που βασίζονται σε κείμενο, όπου τα δεδομένα και οι εντολές είναι αυστηρά σε κείμενο. Οι αναπαραστάσεις του γραφικού περιβάλλοντος χειρίζονται από μια συσκευή κατάδειξης όπως ποντίκι, trackball, γραφίδα ή με ένα δάχτυλο σε μια οθόνη αφής.

3.2 Ανάλυση του γραφικού περιβάλλοντος της εφαρμογής

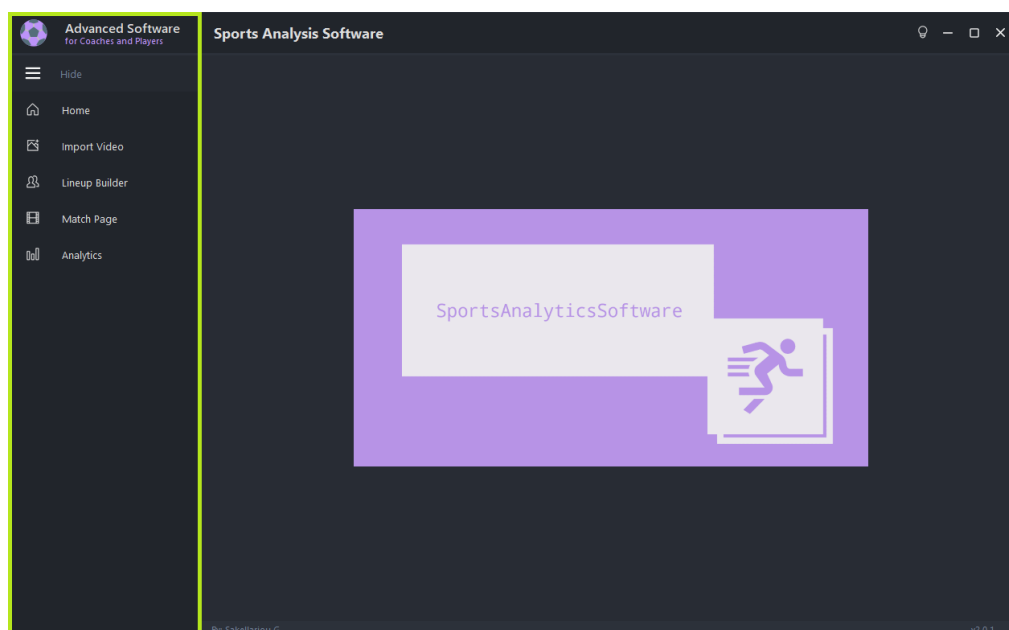
Το γραφικό περιβάλλον χρήστη της εφαρμογής μπορούμε να το χωρίσουμε σε τέσσερα διαφορετικά «εξαρτήματα» όπου το καθένα από αυτά έχει έναν δικό του διαφορετικό ρόλο στην εφαρμογή.

3.2.1 Μενού μετακίνησης

Όπως μπορούμε να παρατηρήσουμε στην εικόνα 3.1 υπάρχει αριστερό μενού που παρέχει όλες τις σελίδες της εφαρμογής και επιτρέπει την μετακίνηση ανάμεσα στα διαφορετικά στοιχεία. Το αριστερό μενού της εφαρμογής που περιέχει όλες τις σελίδες της εφαρμογής και επιτρέπει την μετακίνηση ανάμεσα στα διαφορετικά γραφικά στοιχεία της εφαρμογής. Αναλυτικά έχουμε το πρώτο κουμπί από επάνω προς τα κάτω με στόχο την επέκταση και την συρρίκνωση του ίδιου του μενού. Σε περίπτωση που το μενού βρίσκεται σε κατάσταση διαστολής τότε φαίνονται τόσο το εικονίδιο, όσο και το όνομα της σελίδας στην οποία θα μεταβεί ο χρήστης κατά την επιλογή. Στην αντίθετη περίπτωση φαίνεται μονάχα το εικονίδιο, αλλά η λειτουργικότητα παραμένει ίδια.

Συγκεκριμένα το μενού έχει την δυνατότητα να μας κατευθύνει στα εξής διαφορετικά γραφικά στοιχεία:

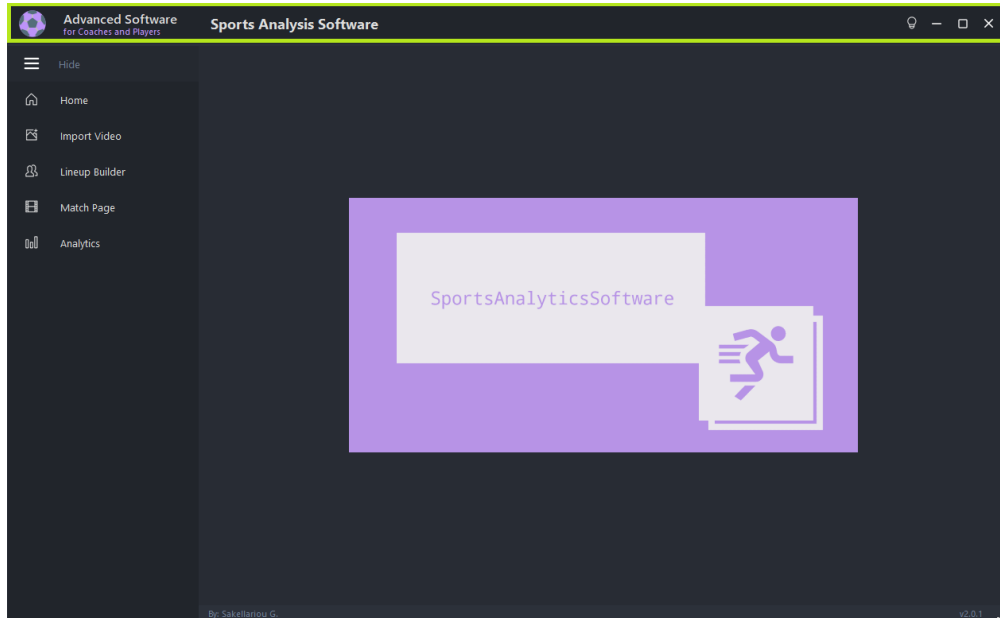
- Αρχική σελίδα
- Σελίδα εισαγωγής βίντεο
- Σελίδα δημιουργίας παράταξης παιχτών
- Σελίδα δημιουργίας στατιστικών για τους παίκτες
- Αναλυτικά δεδομένα για τον αγώνα



Εικόνα 3.1: Application's Navigation Menu

3.2.2 Η κεφαλίδα της εφαρμογής

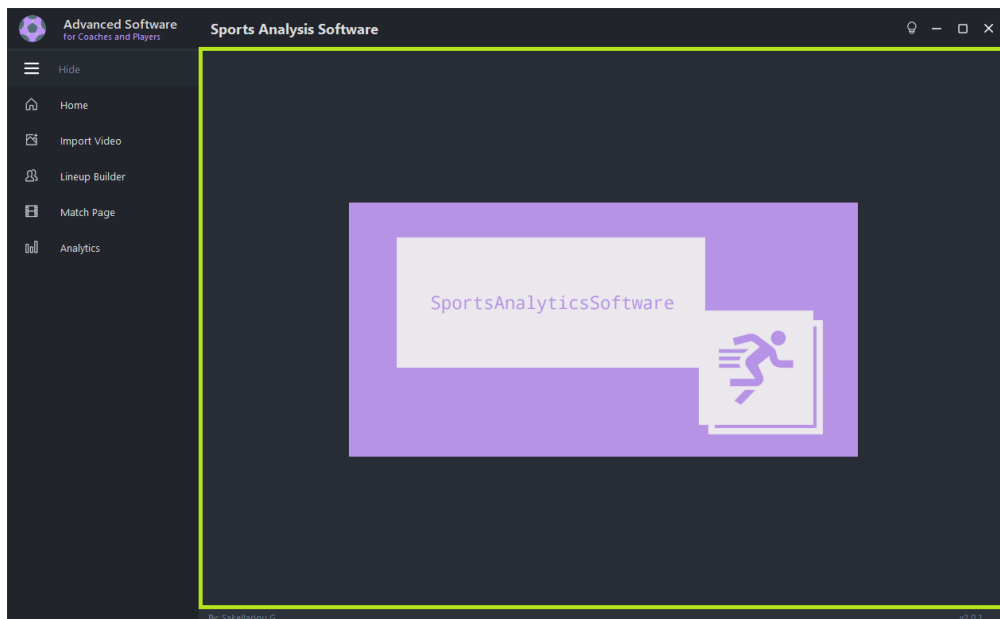
Επιπλέον υπάρχει κεφαλίδα που βρίσκεται στην κορυφή της εφαρμογής (βλ. 3.2) η οποία δίνει την δυνατότητα μεγιστοποίησης, ελαχιστοποίησης, κλεισίματος της εφαρμογής, καθώς και την αλλαγή χρωματικού συνδυασμού σε όλη την εφαρμογή από ανοιχτόχρωμο σε σκουρόχρωμο και το αντίθετο μέσω κουμπιού, επιπλέον αλλάζει και ο τίτλος ανάλογα με το κεντρικό γραφικό στοιχείο.



Εικόνα 3.2: Application's Header

3.2.3 Το κεντρικό γραφικό στοιχείο της εφαρμογής

Επίσης υπάρχει το κεντρικό γραφικό στοιχείο (βλ. 3.3) της εφαρμογής που παρέχει το γραφικό περιβάλλον για όλες τις λειτουργίες της εφαρμογής και εναλλάσσεται καθ' όλη την διάρκεια χρήσης της εφαρμογής.



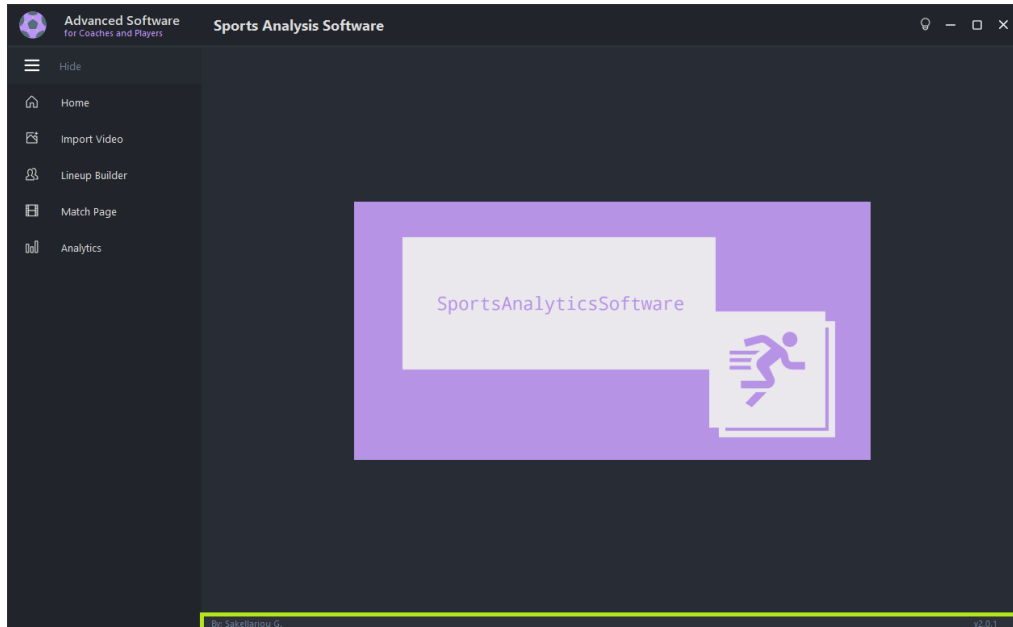
Εικόνα 3.3: Application's Main Widget

Το κεντρικό γραφικό στοιχείο όπως είναι λογικό αλλάζει κατά τη χρήση της εφαρμογής. Παρακάτω θα δούμε τις σελίδες που συναντά ο χρήστης.

3.2.4 Το υποσέλιδο της εφαρμογής

Παρακάτω γίνεται αναφορά στο σχήμα 3.4 όπου μας δείχνει το υποσέλιδο της εφαρμογής.

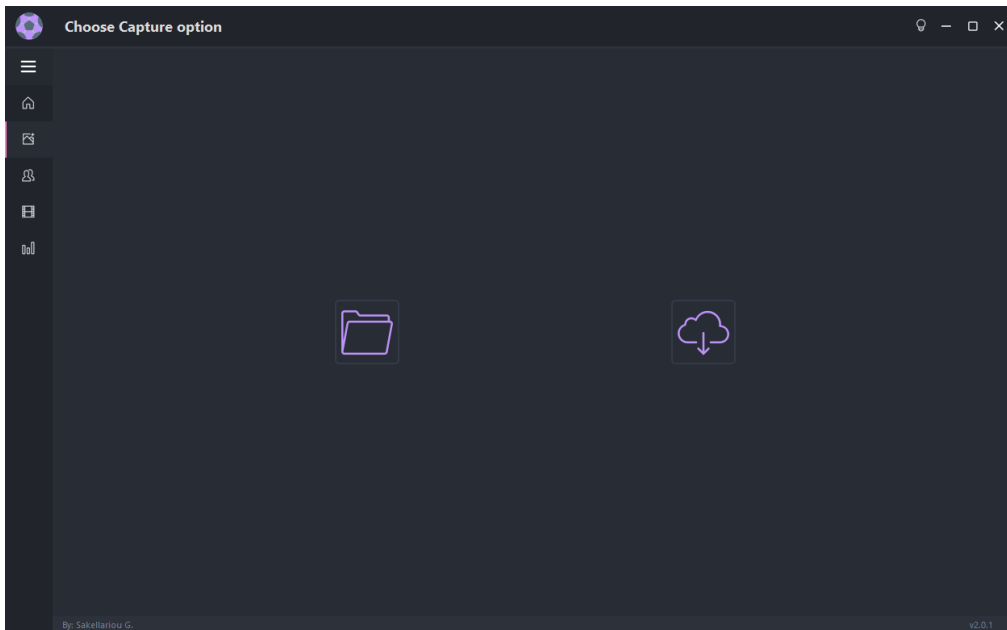
Τέλος υπάρχει το υποσέλιδο (footer) που παρέχει ελάχιστες πληροφορίες για το version και το όνομα του δημιουργού της εφαρμογής.



Εικόνα 3.4: Application's footer

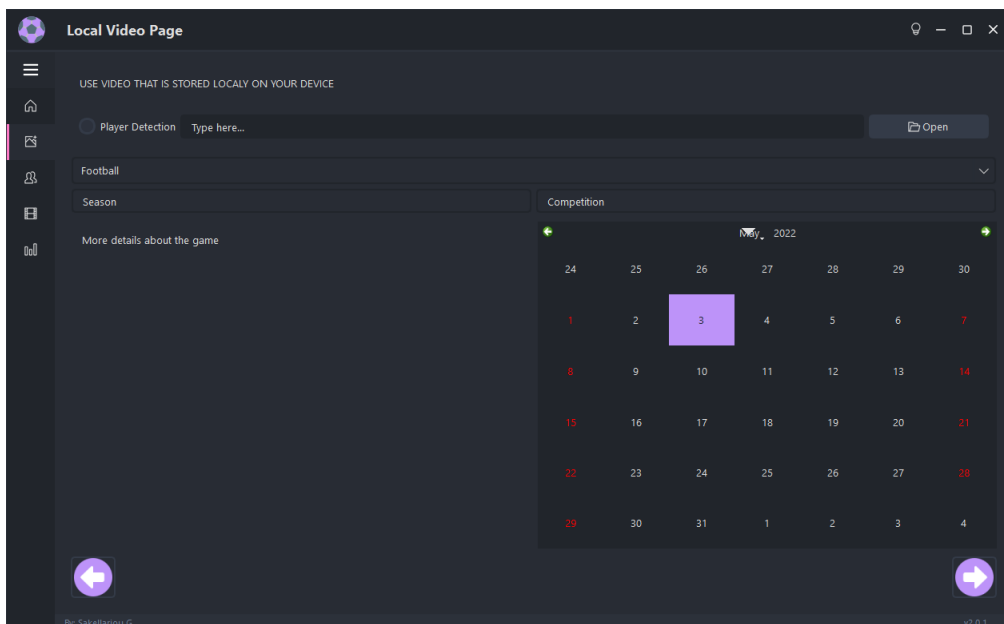
3.2.5 Γραφικά στοιχείο επισύναψης βίντεο

Κατά την εκκίνηση βρισκόμαστε στην αρχική σελίδα της εφαρμογής η οποία δεν παρέχει κάποιες ιδιαίτερες πληροφορίες. Ο χρήστης θα πρέπει να επιλέξει μέσω του μενού την κατηγορία του Import Video για να του προσφερθεί η δυνατότητα επισύναψης του βίντεο. Όταν το κεντρικό γραφικό περιβάλλον θα αλλάξει και θα του δοθεί η δυνατότητα να επισυνάψει βίντεο τοπικά από τον υπολογιστή του ή να προβεί στην λήψη ενός βίντεο από το YouTube. Ακολουθεί εικόνα για το γραφικό στοιχείο επισύναψης βίντεο.



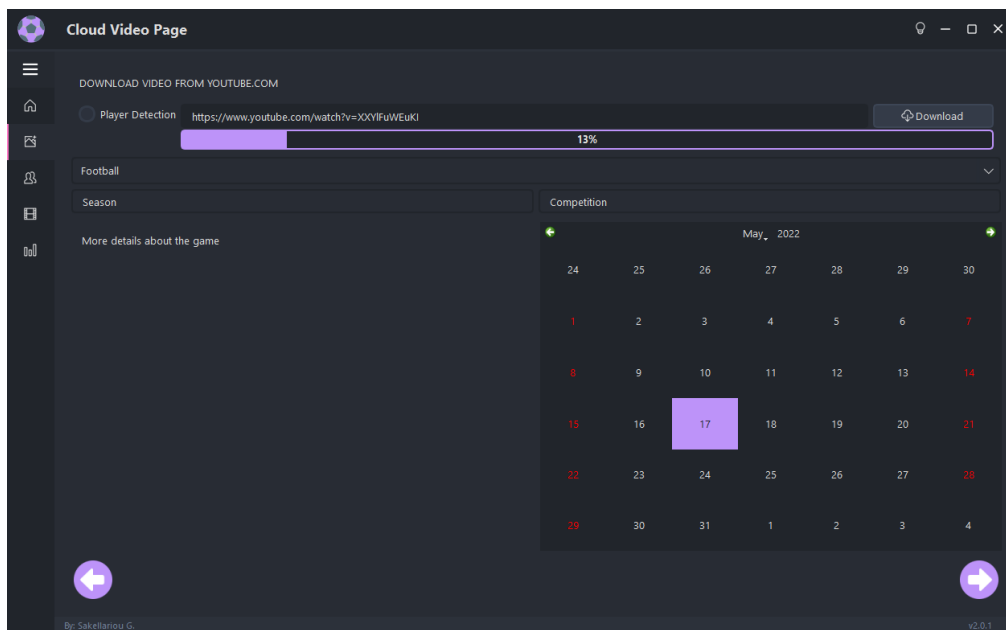
Εικόνα 3.5: Import Video Widget

Επιλέγοντας την πρώτη επιλογή της εικόνας 3.5, εμφανίζεται το γραφικό πλαίσιο της εικόνας 3.6 όπου μας δείχνει αναλυτικά το γραφικό στοιχείο επισύναψης βίντεο από τον υπολογιστή. Εδώ μέσω του κουμπιού Open ή μέσω επικόλλησης του τοπικού μονοπατιού του αρχείου (absolute system file path) γίνεται η επιλογή του βίντεο. Στην συνέχεια μπορεί κάποιος να επιλέξει αν τον ενδιαφέρει η αναγνώριση των παιχτών της ομάδας του, στον χώρο του γηπέδου μέσω ενός αλγορίθμου που θα αναλυθεί στην πορεία. Έπειτα μπορεί να πληκτρολογήσει την σεζόν του αγώνα (π.χ. 2^η σεζόν), το είδος του αγώνα (π.χ. Champions League), κάποιες λεπτομέρειες για τον αγώνα (π.χ. εντός έδρας, απώλεια κάποιου βασικού παίκτη) και τέλος την ημερομηνία που γίνεται ή έγινε ο αγώνας. Από τα παραπάνω στοιχεία είναι υποχρεωτικό να συμπληρωθεί μονάχα η επιλογή του βίντεο, και τα υπόλοιπα μπορούν να μην συμπληρωθούν ποτέ.



Εικόνα 3.6: Import Video Locally

Εάν αντί για επισύναψη βίντεο με τοπικό τρόπο επιλέξουμε την επισύναψη βίντεο μέσω λήψης από το YouTube, τότε εμφανίζεται ένα ελαφρώς διαφορετικό γραφικό περιβάλλον όπως μπορεί να διαπιστωθεί μέσω της εικόνας 3.7. Η μόνη διαφορά που υπάρχει σε περίπτωση όπου ο χρήστης επιλέξει την λήψη βίντεο από το YouTube είναι η ύπαρξη μίας μπάρας προόδου (progress bar) που δείχνει το ποσοστό ολοκλήρωσης της λήψης του βίντεο. Μόλις ολοκληρωθεί αυτή, τότε αντικαθίσταται ο σύνδεσμος που επικόλλησε ο χρήστης από την λέξη “Completed”. Σε περίπτωση που ο σύνδεσμος δεν είναι αποδεκτός (π.χ. δεν παραπέμπει σε κάποιο βίντεο, δεν είναι εφικτή η λήψη λόγω Firewall του υπολογιστή, κ.α.) τότε διαγράφεται και ο χρήστης πρέπει να κάνει επικόλληση εκ νέου.



Εικόνα 3.7: Import Cloud Video

Ανεξάρτητα από τον τρόπο επισύναψης του βίντεο, αποθηκεύονται οι ίδιες πληροφορίες σε ένα αρχείο για μελλοντική χρήση από την εφαρμογή.

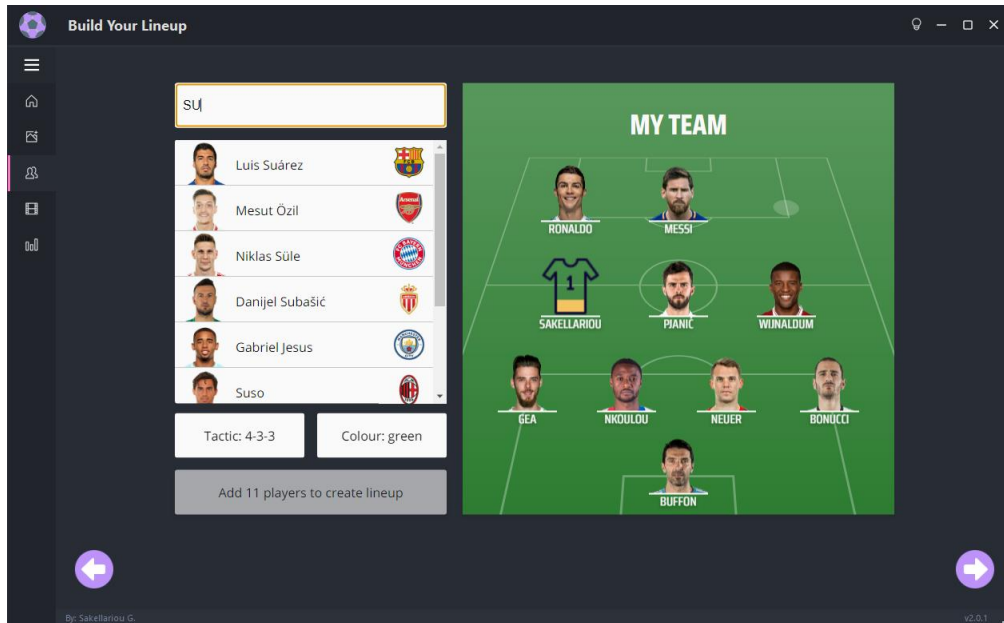
3.2.6 Δημιουργία στρατηγικής – Lineup Builder

Εάν ο χρήστης πατήσει στο βέλος της επόμενης σελίδας το γραφικό στοιχείο θα αλλάξει και θα εμφανιστεί αυτό της διαμόρφωσης ομάδας το οποίο φαίνεται στην εικόνα 3.8. Όπως μπορεί να παρατηρηθεί, στα αριστερά υπάρχει ένα σημείο εισαγωγής κειμένου που επιστρέφει τους διάφορους παίχτες ανάλογα με το κείμενο που πληκτρολογήθηκε μέσα σε αυτό. Εάν κάποιος παίχτης δεν υπάρχει ο χρήστης μπορεί να γράψει ένα οποιοδήποτε όνομα και να τοποθετηθεί στο γήπεδο χωρίς κανένα πρόβλημα με μία τυχαία εικόνα. Επιπλέον γίνεται η αλλαγή της τακτικής της ομάδας σε διαφορετικά δημοφιλή συστήματα (π.χ. 4-3-3, 3-5-2 κ.α.). Ο κάθε παίχτης τοποθετείται ιεραρχικά στην κάθε θέση, δηλαδή αν υπάρχει θέση επίθεσης η οποία είναι κενή, και ο παίχτης είναι επιθετικός τότε κατατάσσεται εκεί. Σε περίπτωση που δεν υπάρχει τέτοια θέση, τοποθετείται στην αμέσως επόμενη θέση της επιλογής του, ενώ μπορεί να μπει και σε θέση που δεν είναι στις προτιμήσεις του σε περίπτωση που δεν υπάρχει τέτοια κενή θέση και τοποθετείται στην πρώτη θέση που θα βρεθεί διαθέσιμη ιεραρχικά από επάνω προς τα κάτω.

Σε περίπτωση που πρέπει να αλλάξει η θέση ενός παίχτη με έναν άλλον, απλώς μπορεί να γίνει χρήση του κέρσορα του υπολογιστή και να μετακινηθεί σε οποιαδήποτε θέση ανταλλάσσοντας την θέση του με την θέση ενός άλλου.

Είναι υποχρεωτικό να συμπληρωθούν έντεκα διαφορετικοί παίκτες για την συμπλήρωση μίας ομάδας. Αφού γίνει αυτό, ο χρήστης μπορεί να πιάσει το κουμπί “Get my lineup” και στην συνέχεια “Done” για να αποθηκευτεί η τακτική του για μελλοντική χρήση.

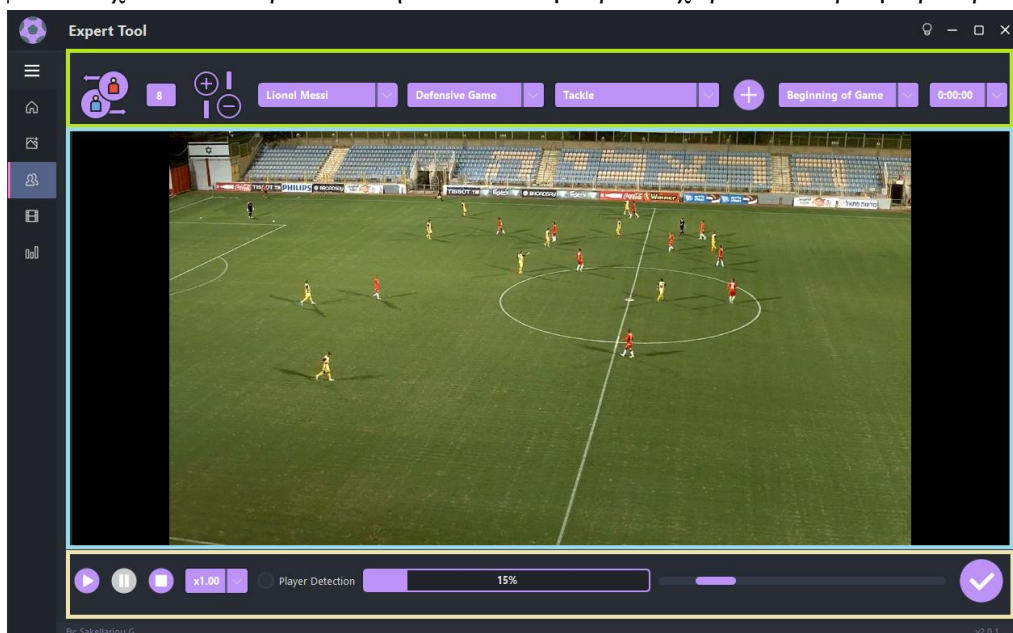
Σε περίπτωση που πατηθεί το αριστερό βέλος το γραφικό στοιχείο αλλάζει σε αυτό της επισύναψης βίντεο, ενώ σε περίπτωση που πατηθεί το δεξί τότε το γραφικό στοιχείο αλλάζει και εμφανίζεται το επόμενο.



Εικόνα 3.8: Lineup Builder

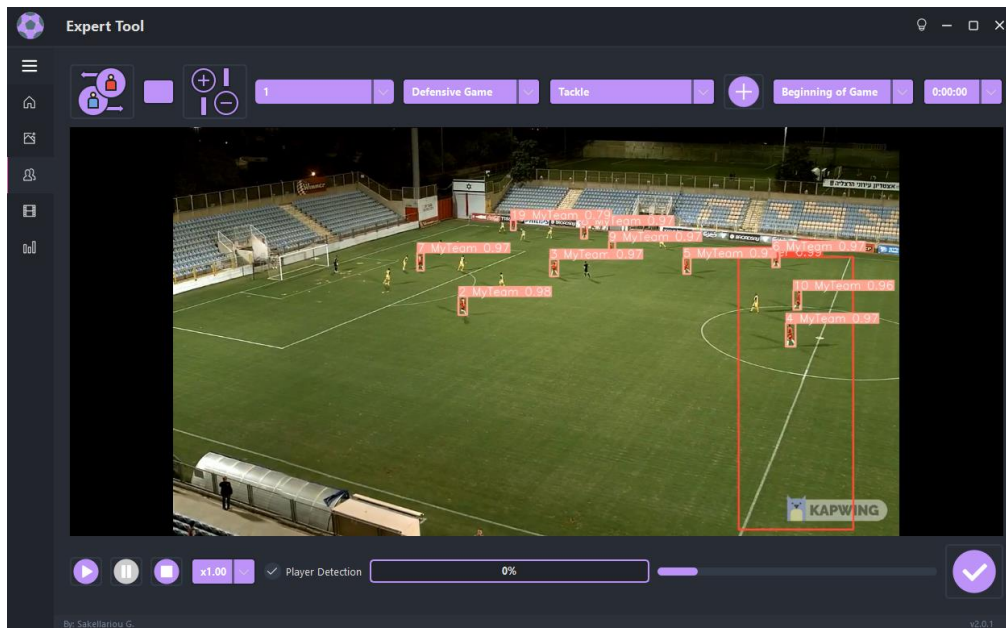
3.2.7 Δημιουργία στατιστικών

Το γραφικό στοιχείο που διακρίνεται στην εικόνα 3.9 μπορεί να χωριστεί σε τρία μικρότερα στοιχεία.



Εικόνα 3.9: Statistics Generator

Ξεκινώντας από κάτω προς τα πάνω έχουμε τα βασικά κουμπιά αναπαραγωγής του βίντεο, δηλαδή αυτά της εκκίνησης, της παύσης, και του τερματισμού αναπαραγωγής. Δίπλα μπορεί κανείς να βρει την ταχύτητα με την οποία πραγματοποιείται η αναπαραγωγή του βίντεο, η οποία είναι αρχικοποιημένη με την τιμή 1, και μπορεί να πάρει τιμές από 0.25 έως και 2 ώστε να παίζει 2 φορές πιο αργά ή πιο γρήγορα το βίντεο αντίστοιχα. Έπειτα υπάρχει ένα κουμπί με κείμενο Player Detection που δείχνει στον χρήστη το βίντεο που ανέλυσε ο υπολογιστής όταν ο χρήστης επέλεξε την αντίστοιχη ρύθμιση κατά την επισύναψη του βίντεο (βλ. 3.10). Σε περίπτωση που δεν είχε επιλεγθεί η συγκεκριμένη ρύθμιση τότε το κουμπί δεν κάνει απολύτως τίποτα.



Εικόνα 3.10: Player Detection Analyzed Video

Στην συνέχεια μπορούμε να παρατηρήσουμε το ποσοστό του βίντεο που έχει αναπαραχθεί, καθώς και μία μπάρα για να μετακινήσουμε το βίντεο σε οποιαδήποτε στιγμή του αγώνα θέλουμε. Τέλος της κάτω μεριάς υπάρχει ένα σημάδι ελέγχου που οριστικοποιεί τις επιλογές, και μας μεταφέρει στην επόμενη σελίδα.

Στο κέντρο πραγματοποιείται μονάχα η αναπαραγωγή του βίντεο και δεν δίνεται η δυνατότητα στον χρήστη να προσαρμόσει οτιδήποτε σε εκείνο το σημείο.

Το επάνω κομμάτι είναι και το σημαντικότερο για την ανάλυση του αγώνα. Μέσω του πρώτου κουμπιού μπορεί ο χρήστης να μεταβεί άμεσα στο γραφικό στοιχείο διαμόρφωσης ομάδας που αναφέρθηκε και προηγουμένως. Αυτό είναι λογικό καθώς ο αγώνας και ο τρόπος που παίζει η κάθε ομάδα καθορίζεται καθ' όλη τη διάρκεια του αγώνα, ενώ μπορεί να υπάρχει κάποιος απρόοπτος τραυματισμός, κάποια κόκκινη κάρτα κ.α. Αξίζει να σημειωθεί πως η προηγούμενη τακτική παραμένει αποθηκευμένη, οπότε δεν χρειάζεται ο προπονητής να την δημιουργήσει από την αρχή αντιθέτως αρκεί να πραγματοποιήσει μονάχα τις αλλαγές που θέλει. Στην συνέχεια παρατηρείται ένας εισαγωγέας κειμένου όπου σε συνδυασμό με το αμέσως επόμενο κουμπί επιτρέπει την εστίαση και μεγέθυνση του πλάνου σε έναν συγκεκριμένο παίκτη. Αυτό μπορεί να πραγματοποιηθεί μονάχα σε περίπτωση όπου ο χρήστης έχει επιλέξει την επιλογή του Player Detection κατά την επισύναψη του βίντεο και το αποτέλεσμα φαίνεται στην εικόνα 3.10.

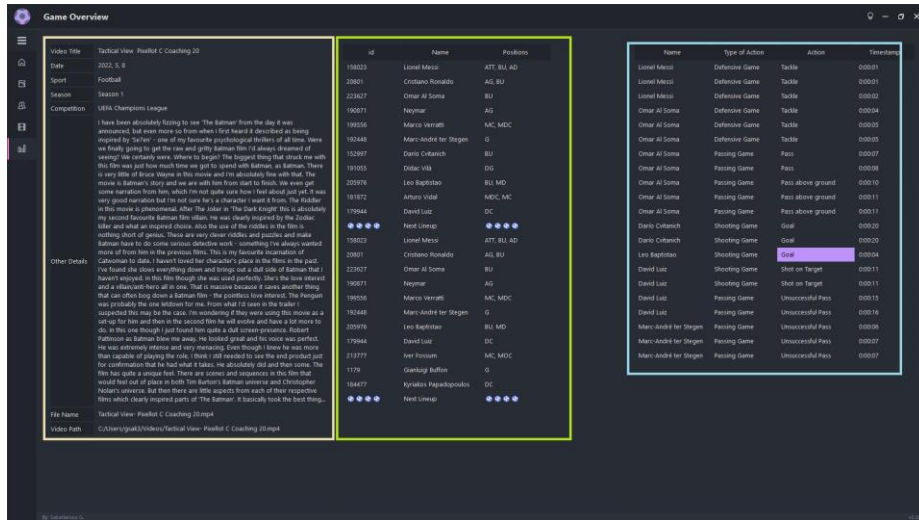


Εικόνα 3.11: Zoom Into Player

Τα τρία επόμενα αναπτυσσόμενα κελιά είναι για την επιλογή του παίχτη και για την ενέργεια του. Για παράδειγμα εάν ο προπονητής θέλει να αποθηκεύσει ότι ο παίχτης Lionel Messi έβαλε γκολ πρέπει να επιλέξει το όνομα του στο πρώτο κελί, την κατηγορία της ενέργειας που πραγματοποίησε, δηλαδή Shooting Game, και τέλος την ενέργεια του γκολ. Πατώντας τον σταυρό αποθηκεύεται η ενέργεια καθώς και η ώρα που πραγματοποιήθηκε αυτή βάση του χρονικού στιγμιότυπου του βίντεο για μελλοντική χρήση. Ακολουθούν άλλα δύο κελιά, τα οποία λειτουργούν και αυτά μονάχα εάν έχει επιλεγθεί η αναγνώριση παίχτη στον χώρο του γηπέδου. Σκοπός αυτών είναι η επιτυχής αναγνώριση όλων των σκηνών επίθεσης, άμυνας, και όλων των ενδιάμεσων κατά την διάρκεια του αγώνα ώστε να είναι ευκολότερος ο εντοπισμός τους και ως εκ τούτου ευκολότερη η δημιουργία στατιστικών από τον διαιτητή. Όταν ο χρήστης επιλέξει για παράδειγμα το κομμάτι της επίθεσης, θα αλλάξουν τα πεδία του τελευταίου κελιού και θα εμφανίζονται όλους τους χρόνους όπου η ομάδα έχει καταφέρει να κάνει επίθεση και από εκεί μπορεί να αναπαραχθεί το βίντεο σε εκείνη την χρονική στιγμή. Έτσι γίνεται η διαδικασία ανάλυσης του αγώνα πολύ γρηγορότερη, μιας και δεν χρειάζεται να αναζητά μία προς μία τις φάσεις που θέλει.

3.2.8 Εμφάνιση στατιστικών

Το τελευταίο γραφικό στοιχείο είναι αυτό της εμφάνισης των στατιστικών, το οποίο περιέχει τρεις πίνακες ο καθένας από τους οποίους παρέχει διαφορετικές πληροφορίες όπως διακρίνεται στην εικόνα 3.12.

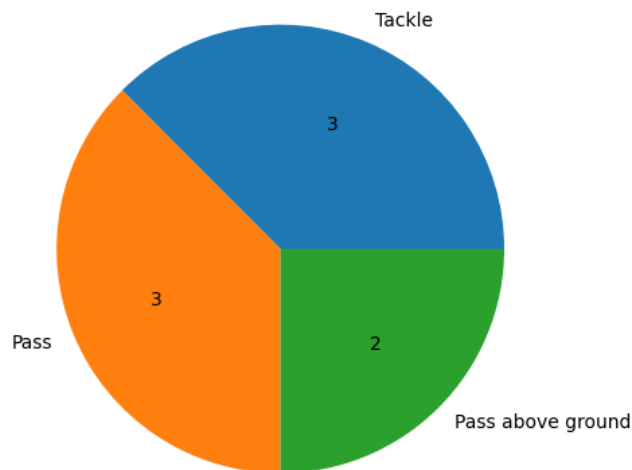


Εικόνα 3.12: Statistics Showcase

Ο πρώτος πίνακας δείχνει στον χρήστη γενικές πληροφορίες οι οποίες αφορούν κατά βάση το αρχείο του βίντεο και όχι τον αγώνα. Τέτοιες πληροφορίες είναι ο τίτλος του βίντεο, η ημερομηνία που επέλεξε ο χρήστης κατά την εισαγωγή του βίντεο, το άθλημα, την σεζόν, τον τύπο του διαγωνισμού του βίντεο, περαιτέρω πληροφορίες που έδωσε ο χρήστης σαν σχόλιο στο βίντεο, το όνομα του αρχείου του βίντεο, και τέλος το σημείο από το οποίο έγινε η εισαγωγή του βίντεο (file path).

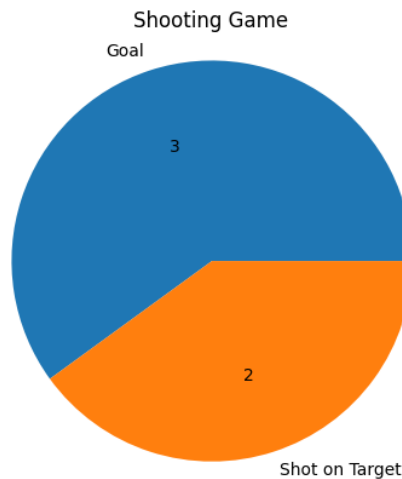
Στον δεύτερο πίνακα ο χρήστης βλέπει αναλυτικά τις τακτικές που είχε δημιουργήσει κατά την διάρκεια του αγώνα. Συγκεκριμένα βλέπουμε το id του παίχτη, το όνομα του, και τέλος τις θέσεις στις οποίες είναι ικανός να παίξει. Όπως αναφέρθηκε και προηγουμένως η τακτική μίας ομάδας μπορεί να αλλάξει πολλές φορές κατά την διάρκεια ενός αγώνα, έτσι υπάρχει ένα διαχωριστικό για κάθε τέτοια αλλαγή. Πατώντας διπλό κλικ σε οποιοδήποτε κελί αυτού του πίνακα εμφανίζεται ένα σχετικό γράφημα σε μορφή πίτας με τις ενέργειες του παίχτη στον αγώνα (βλ. 3.13) ή σε περίπτωση που δεν έχει πραγματοποιήσει καμία ενέργεια εμφανίζεται ένα κενό γράφημα.

Omar Al Soma actions during the entire game



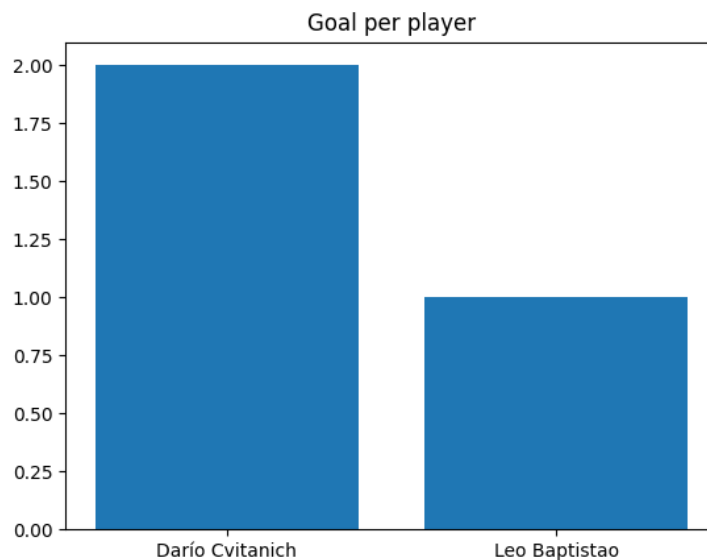
Εικόνα 3.13: Player Statistics Chart

Ο τρίτος πίνακας είναι υπεύθυνος για την ανάδειξη ενεργειών του κάθε παίχτη την κάθε χρονική στιγμή. Στο πρώτο κελί φαίνεται το όνομα του παίχτη, ακολουθεί η ομάδα της ενέργειας που πραγματοποιήσε, η ενέργεια, και τέλος η χρονική στιγμή που έγινε αυτή. Κάνοντας διπλό κλικ πάνω στο όνομα του παίχτη εμφανίζεται όπως και προηγουμένως ένα διάγραμμα σε μορφή πίτας (βλ. 3.13). Σε περίπτωση που γίνει διπλό κλικ στο κελί της οικογένειας μίας ενέργειας τότε εμφανίζονται όλες οι ενέργειες που ανήκουν σε εκείνη την οικογένεια σε μορφή γραφήματος πίτας (βλ. 3.14).



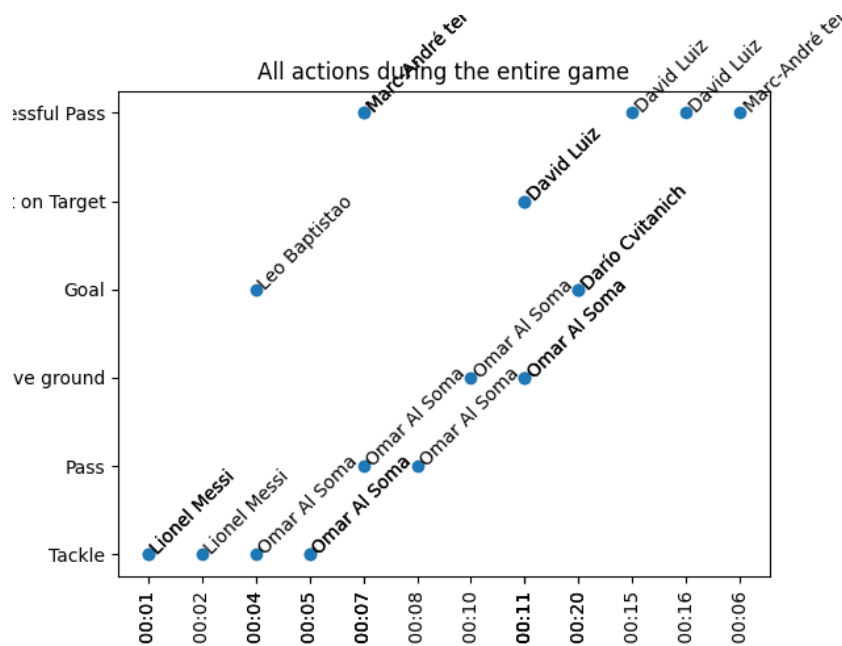
Εικόνα 3.14: Action Family Chart

Κάνοντας διπλό κλικ σε μία ενέργεια εμφανίζεται ο αριθμός των φορών που έχει πραγματοποιηθεί η συγκεκριμένη ενέργεια από τον κάθε παίχτη σε μορφή γραφήματος μπάρας (βλ. Εικόνα 3.15).



Εικόνα 3.15: Action Per Player

Τέλος κατά το διπλό κλικ σε οποιοδήποτε κελί της τελευταίας στήλης εμφανίζεται ένα γράφημα με όλες τις ενέργειες όλων των παιχτών, καθώς και οι χρονικές στιγμές που έχουν γίνει. Όπως είναι λογικό όσοι περισσότερες ενέργειες γίνονται σε έναν αγώνα τόσο πιο δυσνόητο θα γίνεται το συγκεκριμένο γράφημα (βλ. Εικόνα 3.16).



Εικόνα 3.16: All Actions Chart

Κεφάλαιο 4ο: Οδηγίες εγκατάστασης

4.1 Εισαγωγή

Ο κώδικας της συγκεκριμένης εργασίας φιλοξενείται στο [GitHub.com](https://github.com). Για την εκκίνηση της εφαρμογής χρειάζεται αρχικά η εγκατάσταση της Python και προαιρετικά του git στον υπολογιστή.

4.2 Εγκατάσταση Python

Για την εγκατάσταση της Python αρκεί η λήψη και η εκτέλεση του κατάλληλου αρχείου από την επίσημη σελίδα της Python. Αφού μεταβούμε στη σελίδα <https://www.python.org/downloads/> μέσω του φυλλομετρητή πατώντας στο ανάλογο εκτελέσιμο αρχείο γίνεται η λήψη του αρχείου που θα εγκαταστήσει την Python στον υπολογιστή. Αξίζει να σημειωθεί πως πολλά λειτουργικά συστήματα την έχουν ήδη εγκατεστημένη και αυτή η ενέργεια δεν είναι απαραίτητη.

4.3 Εγκατάσταση Git

Η εγκατάσταση του Git έχει αρκετά αντίστοιχο τρόπο με την εγκατάσταση της Python που αναφέρθηκε προηγουμένως. Αφού μεταβούμε στη σελίδα <https://git-scm.com/downloads> και κατεβάσουμε το κατάλληλο αρχείο, πραγματοποιούμε την εγκατάσταση του git.

4.4 Λήψη του κώδικα της εφαρμογής

Η ανάπτυξη της συγκεκριμένης εφαρμογής συνεχίζει να προχωράει μέχρι και θα συνεχίσει στο άμεσο μέλλον, οπότε για την λήψη του κώδικα που απευθύνεται στην συγκεκριμένη πτυχιακή εργασία θα πρέπει να κάνουμε τις εξής εντολές. Αρχικά θα πρέπει μέσω του git να κλωνοποιήσουμε το repository της εφαρμογής. Αυτό γίνεται ανοίγοντας το τερματικό και γράφοντας την εντολή 'git clone

`https://github.com/Gsak31/Sports-Analysis-Software.git`. Με αυτή την εντολή πραγματοποιούμε την λήψη της τελευταίας έκδοσης της εφαρμογής η οποία όμως έχει κάποιες διαφοροποιήσεις σε σχέση με αυτή που αναλύθηκε και θα αναλυθεί στην πορεία. Έτσι μέσω των εντολών `cd 'Sports-Analysis-Software'` και της εντολής `'git checkout no-db'` οι οποίες θα πληκτρολογηθούν πάλι στο τερματικό, αλλάζουμε την έκδοση της εφαρμογής σε αυτή που παρουσιάζεται στη συγκεκριμένη εργασία.

4.5 Λήψη βιβλιοθηκών και εκκίνηση της εφαρμογής

Επίσης απαραίτητη είναι η εγκατάσταση των βιβλιοθηκών, έτσι πρέπει μέσω του τερματικού και της εντολής `'pip install -r requirements.txt'` να εγκαταστήσουμε όλες τις απαραίτητες βιβλιοθήκες που χρειάζονται για την εφαρμογή. Στη συνέχεια αρκεί η εκτέλεση της εντολής `'python main.py'` για την εκκίνηση της.

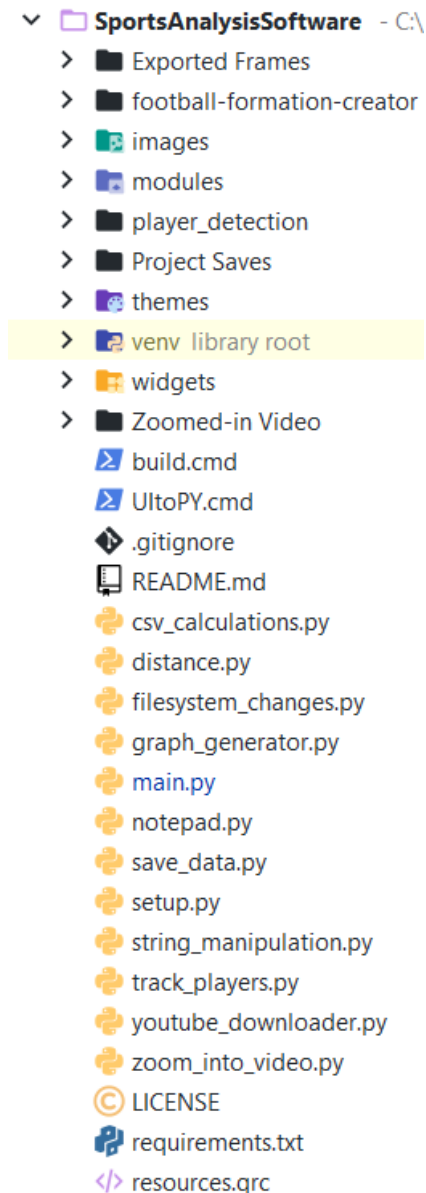
Κεφάλαιο 5ο: Ανάλυση Κώδικα

5.1 Εισαγωγή

Όλες οι παραπάνω λειτουργίες πραγματοποιήθηκαν με την χρήση των βιβλιοθηκών οι οποίες αναφέρθηκαν προηγουμένως. Παρακάτω θα αναλύσουμε τον τρόπο με τον οποίο επιτεύχθηκε αυτό σε επίπεδο κώδικα.

5.2 Η δομή του Project

Στην εικόνα 5.1 μπορεί κανείς να παρατηρήσει την δομή του συγκεκριμένου Project στο επίπεδο της ρίζας (root) του.



Εικόνα 5.1: Project Content Root

Περιληπτικά δίνεται παρακάτω ο λόγος ύπαρξης του κάθε αρχείου:

- Αρχικά το αρχείο **setup.py** μετατρέπει το Project σε εκτελέσιμη μορφή ώστε να μην χρειαστεί η εγκατάσταση όλων των βιβλιοθηκών που αναφέρθηκαν προηγουμένως. Αξίζει να σημειωθεί πως η συγκεκριμένη μέθοδος είναι χρήσιμη μονάχα για την δημιουργία εκτελέσιμου προγράμματος στο περιβάλλον των Windows, μιας η εκτέλεση θα δημιουργήσει ένα αρχείο τύπου .exe.
- Το **main.py** είναι ο σκελετός της εφαρμογής, είναι υπεύθυνο για την αλληλεπίδραση του γραφικού περιβάλλοντος και των υπόλοιπων στοιχείων της εφαρμογής.
- Η λειτουργία του **string_manipulation.py** είναι να μετατρέπει και να προσαρμόζει τα δεδομένα σε διαφορετικούς τύπους (π.χ. integer σε string), να αλλοιώνει κείμενα, κ.α.
- Στόχος του **filesystem_changes.py** είναι η επίτευξη διάφορων αλλαγών στα αρχεία του υπολογιστή, και πιο συγκεκριμένα στα αρχεία που βρίσκονται στον φάκελο της εφαρμογής
- Μέσω του **youtube_downloader.py** γίνεται η εφικτή η λήψη βίντεο από την πλατφόρμα του YouTube.
- Με το **csv_calculations.py** γίνονται διάφορες διεργασίες σε αρχεία μορφής csv.

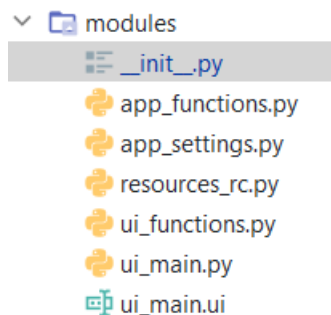
- Εξαιτίας του **save_data.py** είναι δυνατή η αποθήκευση των δεδομένων που παράγονται κατά την χρήση της εφαρμογής, σε csv και json αρχεία.
- Το αρχείο **track.py** είναι υπεύθυνο για την εκκίνηση της εύρεσης των παιχτών στον χώρο του γηπέδου.
- Μέσω του **distance.py** βρίσκεται η απόσταση των παιχτών από διάφορα σημεία του γηπέδου και από την κάμερα.
- Με το **graph_generator.py** δημιουργούνται διάφορα γραφήματα των στατιστικών των παιχτών αφού τα έχει προσθέσει ο χρήστης.
- Στον φάκελο **themes** βρίσκεται ο QSS κώδικας για το αισθητικό κομμάτι της εφαρμογής.
- Στον φάκελο **modules** βρίσκονται αρχεία που είναι υπεύθυνα για το γραφικό περιβάλλον της εφαρμογής και τις λειτουργίες του
- Στον φάκελο **images** υπάρχουν οι εικόνες που χρησιμοποιούνται στην εφαρμογή και έχουν χωριστεί σε υποφακέλους ανάλογα με την χρήση τους.
- Στον φάκελο **football-formation-creator** βρίσκεται το κομμάτι της εφαρμογής που είναι υπεύθυνο για την παράταξη των παιχτών στην εφαρμογή, γραμμένο στο Framework της React.

5.3 Δημιουργία του User Interface

Για την δημιουργία του γραφικού περιβάλλοντος χρήστη, χρησιμοποιήθηκε η εφαρμογή Qt Creator. Μέσω αυτής της εφαρμογής επιτρέπεται η δημιουργία ενός αρχείου τύπου ui, το ui_main.ui. Μέσα σε αυτό το αρχείο βρίσκεται κώδικας γραμμένος σε xml, και περιέχει όλα τα στοιχεία του γραφικού περιβάλλοντος, όπως για παράδειγμα χρώματα, σημεία στα οποία βρίσκονται τα κουμπιά, εικόνες, πίνακες κ.α. Επιπλέον μέσω του αρχείου resources.qrc μεταφορτώνουμε τα διάφορα εξωτερικά στοιχεία που θέλουμε να χρησιμοποιήσουμε στην εφαρμογή του Qt Creator όπως για παράδειγμα τις εικόνες ή βίντεο.

5.3.1 Ο φάκελος Modules

Έτσι το γραφικό περιβάλλον απαρτίζεται από τα αρχεία του φακέλου Modules, ο οποίος με την σειρά του περιέχει τα αρχεία της εικόνας 5.2.



Εικόνα 5.2: Modules Directory

Στο αρχείο **app_settings.py** αρχικοποιούνται κάποιες βασικές γενικές ιδιότητες οι οποίες παραμένουν ίδιες καθ' όλη την χρήση της εφαρμογής όπως τα βασικά χρώματα, το μέγεθος εκκίνησης της εφαρμογής, ο κεντρικός τίτλος κ.α.

Στο αρχείο **ui_functions.py** βρίσκονται οι ενέργειες που πραγματοποιεί η εφαρμογή στο κομμάτι του περιβάλλοντος χρήστη. Μέσω αυτού γίνονται οι ενέργειες της ελαχιστοποίησης και μεγιστοποίησης, η εμφάνιση του μενού στα αριστερά, καθώς και κάποιες επιπλέον λειτουργίες για την βέλτιστη αισθητική της εφαρμογής όπως animations.

Επιπλέον υπάρχει και το αρχείο **app_functions.py** με μοναδικό στόχο την αλλαγή θέματος από ανοιχτόχρωμο σε σκουρόχρωμο με την χρήση του κατάλληλου κουμπιού από τον χρήστη.

Το **resources_rc.py** περιέχει όλα τα εξωτερικά στοιχεία που συνδέονται με το γραφικό περιβάλλον όπως εικόνες.

Το σημαντικότερο στοιχείο για το User Interface είναι το **ui_main.py** που περιέχει όλα τα αντικείμενα του γραφικού περιβάλλοντος. Κάθε κουμπί, label, εισαγωγέας κειμένου και οτιδήποτε άλλο υπάρχει στην εφαρμογή είναι δηλωμένο σε αυτό το αρχείο μαζί με το id του.

Αξίζει να σημειωθεί πως τα **ui_main.py** και **ui_resources.py** δημιουργούνται αυτόματα μέσω του αρχείου **UItoPY.cmd**, το οποίο ουσιαστικά καλεί τις μεθόδους `pyside6-uic` και `pyside6-rcc` οι οποίες εγκαταστάθηκαν μαζί με την βιβλιοθήκη της PySide6 και μετατρέπουν τα αρχεία **ui_main.ui** και **resources.qrc** στα αντίστοιχα αρχεία γραμμένα σε Python.

5.4 Ανάλυση του `save_data.py`

5.4.1 Εισαγωγή

Το αρχείο `save_data.py` είναι υπεύθυνο για την αποθήκευση των δεδομένων που μας δίνει ο χρήστης σε διαφορετικές μορφές ανάλογα με την χρήση που είναι απαραίτητο να γίνει. Χρησιμοποιεί τις βιβλιοθήκες `json` και `pandas`, ενώ επιπλέον κάνει χρήση κάποιων μεθόδων από τα αρχεία `string_manipulation.py`, `filesystem_changes.py` και `csv_calculations.py`.

5.4.2 Ανάλυση Μεθόδων

Παρακάτω περιγράφονται οι ενέργειες της κάθε μεθόδου του αρχείου `save_data.py`:

- Η πρώτη μέθοδος που εμφανίζεται είναι η `save_pre_local_video_data()` η οποία δέχεται ως ορίσματα μεταβλητές ημερομηνίας, του τύπου αθλήματος, της σεζόν, του τύπου του διαγωνισμού, κάποιων λεπτομερειών για τον συγκεκριμένο αγώνα, και τέλος το μονοπάτι του βίντεο που έχει επισυνάψει στο γραφικό στοιχείο επισύναψης τοπικού βίντεο ο χρήστης. Στην συνέχεια δημιουργεί μία μεταβλητή τύπου λεξικού (`dictionary`) η οποία περιέχει όλες αυτές τις πληροφορίες και καλώντας την μέθοδο `save_readable_json()` η οποία βρίσκεται στο ίδιο αρχείο αποθηκεύει αυτές τις πληροφορίες.
- Η μέθοδος που ακολουθεί είναι η `save_pre_cloud_video_data()` η οποία και αυτή δέχεται ως ορίσματα μεταβλητές ημερομηνίας, του τύπου αθλήματος, της σεζόν, του τύπου του διαγωνισμού, κάποιων λεπτομερειών για τον συγκεκριμένο αγώνα, και τέλος το μονοπάτι του βίντεο που έχει επισυνάψει στο γραφικό στοιχείο επισύναψης μέσω YouTube ο χρήστης. Στην συνέχεια και αυτή δημιουργεί μία μεταβλητή τύπου λεξικού (`dictionary`) η οποία περιέχει όλες αυτές τις πληροφορίες και καλώντας την μέθοδο `save_readable_json()` η οποία όπως αναφέρθηκε προηγουμένως βρίσκεται στο ίδιο αρχείο, αποθηκεύει τις πληροφορίες αυτές.
- Η μέθοδος `save_readable_json()` δέχεται το μονοπάτι του φακέλου του αρχείου και το όνομα του αρχείου ως ορίσματα καλεί την μέθοδο `find_last_created_folder()` του αρχείου `filesystem_changes.py` για να εντοπίσει τον τελευταίο φάκελο όπου δημιουργήθηκε, και να αποθηκεύσει το αρχείο `json` σε εκείνο τον φάκελο. Μέσω του `json.dumps()` είναι εφικτή η σωστή στοίχιση του αρχείου ώστε να είναι εύκολα αναγνώσιμο.
- Η μέθοδος `fix_one_line_json()` δέχεται και αυτή ως ορίσματα το μονοπάτι του φακέλου του αρχείου και το όνομα του αρχείου το οποίο είναι σε μορφή `json`. Αρχικά εξαιτίας του τρόπου όπου λειτουργούν κάποια λειτουργικά συστήματα στο κομμάτι των μονοπατιών (`path`) αρχείων, καλείται η μέθοδος `double_backslash_to_slash()` του αρχείου `string_manipulation.py` ώστε να μετατρέψει τις διπλές καθέτους σε μονές (π.χ. από `C:\\my_data.json` σε

C:/my_data.json). Σε περίπτωση που δεν καλούνταν η συγκεκριμένη μέθοδος η εφαρμογή δεν μπορούσε να διαβάσει το αρχείο σωστά. Αφού κληθεί η συγκεκριμένη μέθοδος, τότε μέσω του `json.dumb()` μετατρέπεται το μη αναγνώσιμο json αρχείο σε ένα φιλικό προς τον χρήστη αρχείο.

- Η μέθοδος `json_data_cleanup()` δέχεται ως ορίσματα το μονοπάτι του αρχείου και το όνομα του, και στην συνέχεια καθαρίζει τα άχρηστα δεδομένα από αυτό. Το αρχείο που δέχεται είναι αυτό της τακτικής της ομάδας, και περιέχει περιττές πληροφορίες, όπως για παράδειγμα το κλαμπ του παίκτη. Αφού διαγράψει το συγκεκριμένο πεδίο, αποθηκεύει τις αλλαγές σε ένα αρχείο με όνομα το οποίο περιέχει την ώρα που δημιουργήθηκε η συγκεκριμένη τακτική.
- Η μέθοδος `json_to_csv()` δέχεται ως ορίσματα το μονοπάτι και το όνομα του json αρχείου, αφού καλέσει την μέθοδο `get_file_name()` από το αρχείο `string_manipulation.py` ώστε να πάρει μονάχα το όνομα του αρχείου και όχι την κατάληξη, το μετατρέπεται σε csv και το αποθηκεύει στον ίδιο φάκελο όπου υπήρχε και το αντίστοιχο json αρχείο.
- Η μέθοδος `save_cleaned_data()` δέχεται ως ορίσματα το μονοπάτι και το όνομα του csv αρχείου, και καλώντας την `cleanup_csv_lineup()` κρατάει μονάχα τα σημαντικά κελιά. Στην συνέχεια αποθηκεύει το νέο αρχείο στον ίδιο φάκελο που υπήρχε και το προηγούμενο μη «τακτοποιημένο» αρχείο.
- Η μέθοδος `lineup_manager()` είναι υπεύθυνη για την κλήση των παραπάνω μεθόδων που έχουν άμεση σχέση με τα δεδομένα του lineup της εφαρμογής. Σκοπός της είναι η κλήση των μεθόδων `fix_one_line_json()`, `json_to_csv()`, `save_cleaned_csv()` με την σωστή σειρά ενώ επιστρέφει τα ονόματα των παιχτών μέσω της μεθόδου `get_player_names_from_csv()` του αρχείου `csv_calculations.py`

5.5 Ανάλυση του `csv_calculations.py`

5.5.1 Εισαγωγή

Το αρχείο `csv_calculations.py` έχει ως σκοπό ύπαρξης την δημιουργία και τον καθαρισμό αρχείων τύπου csv, ενώ σε ορισμένες περιπτώσεις την μετατροπή τους σε κάποιου διαφορετικού είδους αρχείου ή μεταβλητής. Χρησιμοποιεί τις βιβλιοθήκες `re`, `pandas`, και `datetime`, ενώ κάνει χρήση των αρχείων `string_manipulation.py` και `filesystem_changes.py`.

5.5.2 Ανάλυση μεθόδων

Παρακάτω περιγράφονται οι ενέργειες της κάθε μεθόδου του αρχείου `csv_calculations.py`:

- Η μέθοδος `get_player_names_from_csv()` δέχεται το μονοπάτι και το όνομα ενός αρχείου json, αναζητά το αντίστοιχο αρχείο csv με το ίδιο όνομα μέσω της `get_file_name()` του αρχείου `string_manipulation.py`, και στην συνέχεια μέσω της `list_to_string()` του αρχείου `string_manipulation.py` επιστρέφει τα ονόματα των παιχτών ως ένα string αφού τα ξεχωρίσει μέσω της `pandas`.
- Η μέθοδος `add_to_csv()` δέχεται ως ορίσματα το όνομα ενός παίκτη, το είδος της ενέργειας που έκανε, την ενέργεια που έκανε, και την χρονική στιγμή σε δευτερόλεπτα του βίντεο. Στην συνέχεια μετατρέπεται τα δευτερόλεπτα σε ώρες, λεπτά, και δευτερόλεπτα μέσω της μεθόδου `timedelta()` της βιβλιοθήκης `datetime`. Στην συνέχεια δημιουργεί μία μεταβλητή τύπου λεξικού (`dictionary`) με πεδία τις μεταβλητές που δόθηκαν. Αφού γίνει ένας έλεγχος μέσω της μεθόδου `check_if_file_exists()` σε συνδυασμό με την `find_last_created_folder()` που βρίσκονται στο αρχείο `filesystem_changes.py` και το όνομα του αρχείου «`actions.csv`» και βεβαιωθεί ο κώδικας πως υπάρχει αυτό το αρχείο, τότε ουσιαστικά γίνεται επέκταση του αρχείου αυτού προσθέτοντας την νέα αυτή ενέργεια που έκανε ο παίκτης. Σε περίπτωση που ο

έλεγχος έχει ως αποτέλεσμα ψευδές, τότε δημιουργείται ένα νέο αρχείο με πρώτο κλειδί την ενέργεια αυτή.

- Η μέθοδος `create_csv()` έχει ως μοναδικό στόχο την δημιουργία ενός κενού csv στον κατάλληλο φάκελο, μέσω της `find_last_created_folder()` του αρχείου `filesystem_changes()`. Στην συνέχεια απλώς δημιουργεί το αρχείο και σε περίπτωση όπου δεν τα καταφέρει επιστρέφει στην έξοδο το πρόβλημα που υπήρξε.
- Η μέθοδος `read_and_clean()` δέχεται ως μοναδικό όρισμα το txt αρχείο που περιέχει τα σημεία που κατάφερε ο αλγόριθμος να εντοπίσει τον κάθε παίχτη την κάθε χρονική στιγμή. Αφού διαβάσει το αρχείο που δίνεται, στην συνέχεια αφαιρεί τα πεδία που δεν είναι απαραίτητα για τις παρακάτω λειτουργίες, και ονομάζει τους τίτλους αυτών που μένουν για ευκολότερη διαχείριση, σε `Frame`, `ID`, `x`, `y`, `w`, `h` όπου `Frame` είναι το καρέ του βίντεο, `ID` είναι το μοναδικό κλειδί του συγκεκριμένου παίχτη στο χώρο του γηπέδου, `x` και `y` είναι το σημείο που βρέθηκε, και `w` και `h` είναι συντομογραφίες για το `width` και `height`, για το μέγεθος της αναγνώρισης του παίχτη. Αφού κάνει όλες τις παραπάνω ενέργειες, επιστρέφει το dataframe με τα δεδομένα στο σημείο όπου κλήθηκε.
- Η μέθοδος `cleanup_csv_lineup()` δέχεται στην είσοδο ένα dataframe με όλες τις πληροφορίες που δημιουργούνται από το γραφικό στοιχείο της δημιουργίας τακτικής, και έπειτα διαγράφει τα πεδία `photoFolderIndex`, `photo`, `shortName` που περιέχουν φωτογραφίες και την συντομογραφία ονόματος του κάθε παίχτη, και δεν είναι απαραίτητα για την ανάλυση του αγώνα. Σε περίπτωση όπου δεν τα καταφέρει, τότε εμφανίζει το πρόβλημα που παρουσιάστηκε στο τερματικό. Στην συνέχεια μετατρέπει τις θέσεις που παίζει ο κάθε παίχτης από την μορφή λίστας στην απλούστερη μορφή `string` με διαχωριστικό κόμμα (,).

5.6 Ανάλυση του `youtube_downloader.py`

5.6.1 Εισαγωγή

Το αρχείο `youtube_downloader.py` έχει ως μοναδικό στόχο την λήψη βίντεο από την πλατφόρμα του YouTube. Αυτό είναι απαραίτητο καθώς ο χρήστης δεν θα μπορεί να βιντεοσκοπεί τον αγώνα την οποιαδήποτε χρονική στιγμή και σίγουρα όχι σε ανάλυση που να επιτρέπουν οι κάμερες παρακολούθησης ευρείας γωνίας. Γίνεται χρήση της βιβλιοθήκης PyTube και του αρχείου `filesystem_changes.py`

Παρακάτω δίνεται η περιγραφή των ενεργειών της μοναδικής μεθόδου του αρχείου `youtube_downloader.py`:

- Η μέθοδος `save_video_to_downloads()` είναι η μόνη μέθοδος του συγκεκριμένου αρχείου. Δέχεται ως όρισμα ένα `string` με το `url` του βίντεο που χρειάζεται να ληφθεί. Έπειτα παίρνει τον τίτλο του βίντεο που αντιστοιχεί στο συγκεκριμένο `url`, και πραγματοποιεί την λήψη του βίντεο στην υψηλότερη δυνατή ανάλυση και αποθηκεύεται στον φάκελο λήψεων του υπολογιστή ο οποίος εντοπίζεται μέσω της μεθόδου `downloads_path()` του αρχείου `filesystem_changes.py`. Ο λόγος που έχει επιλεγθεί η υψηλότερη δυνατή ανάλυση και όχι μία μέτρια είναι επειδή είναι απαραίτητο να είναι ευδιάκριτοι οι παίχτες στο γήπεδο, ακόμη και αν αυτό στοιχίσει σε αποθηκευτικό χώρο.

5.7 Ανάλυση του `string_manipulation.py`

Το συγκεκριμένο αρχείο έχει ως κύριο λόγο ύπαρξης την μετατροπή και αλλαγή μεταβλητών σε διαφορετικούς τύπους ή και σε ίδιους τύπους απλώς με διαφορετική μορφή. Χρησιμοποιεί τις βιβλιοθήκες `re` και `datetime`.

5.7.1 Ανάλυση Μεθόδων

Παρακάτω περιγράφονται οι ενέργειες της κάθε μεθόδου του αρχείου `string_manipulation.py`:

- Η μέθοδος `path_to_video_title()` δέχεται ως όρισμα ένα `string` με το απόλυτο μονοπάτι του βίντεο του αγώνα και κρατάει μονάχα το όνομα του βίντεο (π.χ. `C:/video_name.mp4` σε `video name`). Αυτό το επιτυγχάνει με ένα αρχικό `regular expression` το οποίο κρατάει οτιδήποτε μετά την τελευταία κάθετη παύλα το οποίο δίνει το όνομα του βίντεο μαζί με την κατάληξη του. Στη συνέχεια εντοπίζει όλους τους χαρακτήρες που δεν είναι αλφαριθμητικοί ή τελεία «.» και τους διαγράφει. Τέλος εντοπίζει την τελευταία τελεία στο όνομα του βίντεο, και διαγράφει αυτή και ότι ακολουθεί.
- Η μέθοδος `path_to_video_name()` δέχεται όπως και η προηγούμενη το απόλυτο μονοπάτι του βίντεο του αγώνα και κρατάει μονάχα το όνομα του βίντεο μαζί με την κατάληξή του (π.χ. `C:/video_name.mp4` σε `video_name.mp4`). Αυτό το επιτυγχάνει με μονάχα μία εντολή, η οποία βρίσκει την τελευταία κάθετη παύλα, και κρατάει ότι ακολουθεί μετά από αυτή. Δεν διαγράφονται σύμβολα όπως στη προηγούμενη μέθοδο, και αυτό γιατί χρειάζεται το ακριβές όνομα του αρχείου.
- Η μέθοδος `qdate_to_date()` δέχεται ως είσοδο μία μεταβλητή τύπου `qdate` που είναι αντικείμενο του γραφικού περιβάλλοντος του PySide6 και το μετατρέπει σε μία επιτρεπτή μορφή (π.χ. `PySide6.QtCore.QDate(2022, 5, 26)` σε `2022, 5, 26`). Αυτό επιτυγχάνεται με την εύρεση των παρενθέσεων και την διαγραφή όλων των χαρακτήρων που δεν βρίσκονται στο εσωτερικό αυτών.
- Η μέθοδος `double_backslash_to_slash()` λύνει το πρόβλημα που παρουσιάζεται με την διαφορετική διαχείριση του συστήματος διαχείρισης αρχείων. Τα Windows χρησιμοποιούν διαφορετική κάτω παύλα από τα συστήματα Linux ή MacOS. Έτσι αυτή η μέθοδος μετατρέπει το μονοπάτι ενός αρχείου σε ένα αποδεκτό που έχει οριστεί για την εφαρμογή μετατρέποντας τις διπλές και μονές ανάποδες καθέτους σε κανονικές (π.χ. `C:\\video.mp4` `C:/video.mp4` σε `C:/video.mp4`). Αυτό επιτυγχάνεται με δύο απλές εντολές που βρίσκουν τις διπλές ή μονές ανάποδες καθέτους και τις αντικαθιστούν με μονές κανονικής κατεύθυνσης.
- Η μέθοδος `contains_letters()` ελέγχει εάν σε ένα `string` που δέχεται στην είσοδο της, υπάρχουν μονάδα γράμματα με την χρήση μίας απλής εντολής `regex`.
- Η μέθοδος `count_numbers_in_string()` μετράει το πλήθος των αριθμών που υπάρχουν στο `string` που δέχεται ως είσοδο ελέγχοντας έναν προς έναν τους χαρακτήρες του `string`.
- Η μέθοδος `sum_digits_string()` δέχεται στην είσοδο της ένα `string` και αφού βεβαιωθεί πως είναι αριθμός τον προσθέτει με τους προηγούμενους του ίδιου `string` και επιστρέφει το συνολικό άθροισμα αυτών των αριθμών.
- Η μέθοδος `dash_after_char()` δέχεται έναν χαρακτήρα και προσθέτει στο τέλος αυτού μία παύλα και στην συνέχεια τον επιστρέφει.
- Η μέθοδος `allow_dash_number()` δέχεται ένα `string` ως όρισμα, και στη συνέχεια ελέγχει με μία εντολή της `regex` εάν περιέχει μονάχα παύλες και αριθμούς. Σε περίπτωση όπου αυτό ισχύει, τότε επιστρέφει αληθές, αλλιώς ψευδές.
- Η μέθοδος `get_file_name()` αυτή δέχεται το όνομα ενός αρχείου στην είσοδο της, και μέσω της εντολής `rsplit` επιστρέφει μονάχα το όνομα του αρχείου χωρίς την κατάληξη του.
- Η μέθοδος `list_to_string()` δέχεται στην είσοδο της μία λίστα της βιβλιοθήκης `pandas` και την μετατρέπει σε ένα `string`. Αυτό γίνεται επειδή τα ονόματα των παιχτών βρίσκονται σε αρχεία τύπου `csv` μετά από κάποια επεξεργασία, και η μετατροπή είναι απαραίτητη για την εμφάνιση στα διάφορα γραφικά στοιχεία.
- Η μέθοδος `frame_to_time_list()` έχει ως στόχο την εύρεση χρονικών στιγμιότυπων βάση των καρτέ του βίντεο, και των καρτέ ανά δευτερόλεπτο. Δέχεται στην είσοδο ένα `pandas dataframe` και τα καρτέ ανά δευτερόλεπτο του βίντεο (`fps`). Έπειτα δημιουργεί μία λίστα με το δευτερόλεπτα διαιρώντας το καρτέ με τα `fps` (το καρτέ με αριθμό 48 σε ένα βίντεο των 24^{ov}

καρέ ανά δευτερόλεπτο δείχνει στο 2^ο δευτερόλεπτο). Έπειτα μετατρέπει την πιθανή float μορφή της τιμής, σε integer για να πάρει ακριβώς δευτερόλεπτα και όχι και κλάσματα του δευτερολέπτου. Τέλος τα μετατρέπει σε μία λίστα που περιέχονται μέσα όλες οι πληροφορίες σε ώρες, λεπτά και δευτερόλεπτα και τις επιστρέφει στο σημείο από το οποίο κλήθηκε.

- Η μέθοδος `date_to_seconds()` δέχεται ως είσοδο ένα string της μορφής 01:35:17, ώρες/λεπτά/δευτερόλεπτα, και το μετατρέπει σε δευτερόλεπτα και επιστρέφει την τιμή. Αυτό χρησιμεύει στο να ορίσει το πρόγραμμα ποιο δευτερόλεπτο του βίντεο θέλει να παίξει και αυτό δεν γίνεται με την παραπάνω μορφή.
- Η μέθοδος `int_list_to_string_list()` δέχεται μία λίστα της Python η οποία περιέχει μονάχα αριθμούς και τους μετατρέπει σε string. Ο λόγος που πρέπει να γίνει αυτό είναι ότι το PySide6 δεν μπορεί να εμφανίσει αριθμούς αλλά μπορεί να εμφανίσει string αριθμών.
- Η μέθοδος `keep_numbers()` δέχεται ένα string, και επιστρέφει μονάχα τους αριθμούς που περιέχει το συγκεκριμένο string.
- Η μέθοδος `string_to_int_or_pass()` αυτή δέχεται ένα string το οποίο σε περίπτωση όπου αποτελείται μονάχα από αριθμούς, το μετατρέπει σε integer. Σε περίπτωση που δεν αποτελείται μονάχα από αριθμούς τότε επιστρέφει το string που δόθηκε χωρίς να αλλάξει τίποτα.
- Η μέθοδος `keep_till_second_dot()` χρησιμεύει στον καθαρισμό της έκδοσης Python που τρέχει ο χρήστης. Έτσι δέχεται ένα string που έχει την μορφή `python3.y.z (tags/v.3.y.z.....)` και επιστρέφει το 3.y.
- Η μέθοδος `get_date_time()` επιστρέφει την τωρινή ημέρα και ώρα με την μορφή μέρα/μήνας/χρόνος και ώρα/λεπτό/δευτερόλεπτο αντίστοιχα, μέσω της μεθόδου `strftime`.
- Κλείνοντας με το αρχείο `string_manipulation.py` η μέθοδος `get_after_last_slash()` δέχεται ένα string σαν όρισμα, και στην συνέχεια το οποίο έχει την μορφή ενός μονοπατιού ενός φακέλου και επιστρέφει τον τελευταίο φάκελο (π.χ. `Documents/folder` σε `folder`).

5.8 Ανάλυση του `filesystem_changes.py`

5.8.1 Εισαγωγή

Το συγκεκριμένο αρχείο ευθύνεται για όλες τις αλλαγές που γίνονται στο σύστημα διαχείρισης αρχείων του φακέλου της εφαρμογής. Οποιαδήποτε δημιουργία ή διαγραφή αρχείων γίνεται από τις μεθόδους του συγκεκριμένου αρχείου. Χρησιμοποιεί τις βιβλιοθήκες `re`, `os`, `glob` και `shutil`, ενώ χρησιμοποιεί και κάποιες από τις μεθόδους του αρχείου `string_manipulation.py`.

5.8.2 Ανάλυση Μεθόδων

Παρακάτω περιγράφονται οι ενέργειες της κάθε μεθόδου του αρχείου `filesystem_changes.py`:

- Η μέθοδος `create_root_save_directory()` δημιουργεί τον πρώτο φάκελο όπου θα αποθηκεύονται οι υποφάκελοι που θα περιέχουν στο εσωτερικό τους τα διάφορα αρχεία που δημιουργούνται κατά την εκτέλεση και λειτουργία της εφαρμογής δημιουργείται δηλαδή ο φάκελος «Project Saves» στο root των αρχείων.
- Η μέθοδος `create_sub_save_folder()` δημιουργεί έναν υποφάκελο μέσα στον φάκελο «Project Saves» με όνομα «Date day.month.year» όπου `day`, `month` και `year` η αντίστοιχη μέρα, μήνας και χρόνος που αντιστοιχεί στην ημερομηνία εκτέλεσης της εφαρμογής και αφού δημιουργήσει αυτόν δημιουργεί στο εσωτερικό του έναν ακόμη φάκελο με όνομα «Time hours.minutes.seconds» όπου `hours`, `minutes` και `seconds` αντιστοιχούν στην αντίστοιχη ώρα, λεπτά και δευτερόλεπτα εκκίνησης της εφαρμογής.
- Η μέθοδος `find_last_folder_lineups()` βρίσκει τον τελευταίο φάκελο που περιέχει μέσα τις τακτικές που δημιούργησε ο χρήστης, και επιστρέφει αυτές ως μία λίστα που περιέχει τα

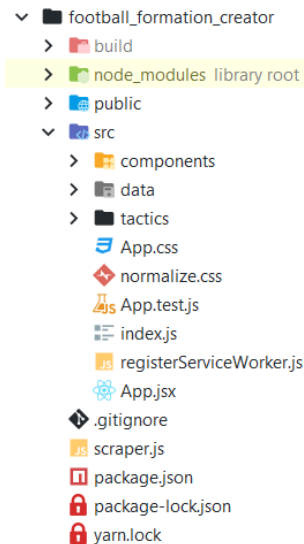
ονόματα τους. Σημαντικό είναι να αναφερθεί πως επιστρέφει μόνο τα αρχεία με lineup στο όνομα τους δηλαδή όχι τις ενέργειες των παιχτών.

- Η μέθοδος `check_if_file_exists()` ελέγχει για την ύπαρξη ενός αρχείου, έτσι ώστε να σιγουρευτεί πως δεν προσπαθεί το πρόγραμμα να διαβάσει ένα αρχείο το οποίο δεν υπάρχει, και επιστρέφει την κατάλληλη λογική τιμή σε περίπτωση που υπάρχει.
- Η μέθοδος `find_last_created_folder()` ψάχνει να βρει τον φάκελο που δημιουργήθηκε τελευταίος μέσα στον φάκελο «Project Saves». Σε περίπτωση που δεν υπήρχε αυτή η μέθοδος και επιλέγαμε φάκελο με βάση της ημερομηνία που βρίσκεται στο όνομα του υποφακέλου θα επιλεγόταν πολλές φορές ο λάθος φάκελος.
- Η μέθοδος `find_yolo_weight_folder()` βρίσκει το αρχείο το οποίο περιέχει τα weights δηλαδή ένα αρχείο που περιέχει το μοντέλο της αναγνώρισης των παιχτών στο χώρο του γηπέδου. Όταν βρει το συγκεκριμένο αρχείο επιστρέφει το μονοπάτι από το root στο οποίο βρίσκεται.
- Η μέθοδος `downloads_path()` επιστρέφει το μονοπάτι του που βρίσκεται ο φάκελος των λήψεων του υπολογιστή σχετικά με το που βρίσκεται ο φάκελος της εφαρμογής έτσι ώστε να μπορεί να αποθηκεύσει το βίντεο κατά την λήψη στον συγκεκριμένο φάκελο.
- Η μέθοδος `delete_file()` αυτή δέχεται ως ορίσματα δύο string ένα με το μονοπάτι του αρχείου που πρόκειται να διαγράψει και ένα με το όνομα του αρχείου αυτού και στην συνέχεια το διαγράφει.
- Η μέθοδος `delete_files_and_folder()` δέχεται στην είσοδο της το μονοπάτι ενός φακέλου, διαγράφει οτιδήποτε βρίσκεται στο εσωτερικό του, από αρχεία μέχρι φακέλους, και στην συνέχεια διαγράφει και τον ίδιον τον φάκελο. Αυτό γίνεται έτσι ώστε η εφαρμογή να μην καταναλώνει μεγάλο ποσοστό του χώρου του υπολογιστή και διαγράφει ότι δεν είναι απαραίτητο αφού το χρησιμοποιήσει (π.χ. διαγραφή φωτογραφιών με ζουμ μετά την προβολή του μεγεθυμένου βίντεο).
- Η μέθοδος `create_zoom_video_folder()` δημιουργεί έναν φάκελο με όνομα «Zoomed-in Video» ο οποίος θα περιέχει το βίντεο στην μεγεθυμένη μορφή όταν το ζητήσει ο χρήστης. Αυτός ο φάκελος βρίσκεται στη ρίζα της εφαρμογής.
- Η μέθοδος `find_last_detection_video()` εντοπίζει το τελευταίο βίντεο που δημιουργήθηκε με αναγνώριση παιχτών έτσι ώστε να το εμφανίσει όταν ο χρήστης ζητήσει κάτι τέτοιο. Αυτή η μέθοδος επιστρέφει δηλαδή το μονοπάτι του τελευταίου βίντεο που έχει εφαρμοστεί αναγνώριση παιχτών.
- Εκτός από βίντεο κατά την αναγνώριση παιχτών δημιουργείται και ένα txt αρχείο το οποίο περιέχει την θέση του κάθε παίκτη σε κάθε καρέ. Έτσι η μέθοδος `find_last_detection_text_file()` επιστρέφει το μονοπάτι του txt αρχείου.
- Η μέθοδος `chdir_to_content_root()` ουσιαστικά επαναφέρει το μονοπάτι στην ρίζα του project μιας και αυτό αλλάζει ανάλογα όταν χρειαστεί να τρέξουμε πολλά αρχεία Python που βρίσκονται σε διαφορετικό φάκελο από αυτόν τις ρίζας.

5.9 Ανάλυση του φακέλου `football_formation_creator`

5.9.1 Εισαγωγή

Ο συγκεκριμένος φάκελος είναι υπεύθυνος για την εμφάνιση του γραφικού στοιχείου για τη δημιουργία τακτικής. Ο κώδικας είναι γραμμένος σε ReactJS, το οποίο είναι framework βασισμένο στην JavaScript. Στην εικόνα που ακολουθεί μπορεί να φανεί η δομή του συγκεκριμένου φακέλου (βλ. Εικόνα 5.3).



Εικόνα 5.3: Lineup Builder Directory

Ο φάκελος που μας ενδιαφέρει είναι ο φάκελος `src`. Μέσα σε αυτόν βρίσκονται οι φάκελοι `components`, `data`, `tactics`, το αρχείο `normalize.css` και το αρχείο `App.jsx`.

5.9.2 Ανάλυση Αρχείων

Παρακάτω θα δοθεί η επεξήγηση αυτών που αναφέρθηκαν:

- Ο φάκελος `components` περιέχει στο εσωτερικό του έξι αρχεία `jsx` που είναι υπεύθυνα για διαφορετικές λειτουργίες.
- Το `Customize.jsx` έχει σχέση με τις διαφορετικές τακτικές, και χρώμα του γηπέδου κατά την χρήση του `lineup` όπου ο χρήστης μπορεί να τα αλλάξει.
- Το `Pitch.jsx` πραγματοποιεί την μετακίνηση μέσω κέρσορα των παιχτών.
- Το `PlayerCard.jsx` είναι η καρτέλα του παίχτη στο `lineup` και περιέχει πληροφορίες για την θέση που βρίσκεται στο γήπεδο στο `lineup`, τις θέσεις που παίζει, το όνομα του.
- Το `PositionIndicator.jsx` είναι η θέση του γηπέδου (επίθεση, άμυνα κ.λπ.)
- Το `Search.jsx` πραγματοποιεί την αναζήτηση των παιχτών με `shuffle` λειτουργία έτσι ώστε να μην χρειάζεται να γραφεί ολόκληρο το όνομα του παίχτη ή ακριβώς.
- Το `SearchResult.jsx` επιστρέφει απλώς το όνομα του παίχτη ή των παιχτών που βρέθηκαν από την αναζήτηση και τα εμφανίζει.
- Ο φάκελος `data` περιέχει μία τοπική «βάση δεδομένων» που στην πραγματικότητα είναι ένας φάκελος με `json` αρχεία για τον κάθε παίχτη που περιέχουν πληροφορίες για το όνομα του, την θέση του, το μονοπάτι στο οποίο βρίσκεται η εικόνα του, η ομάδα του κ.α.
- Το τελευταίο αρχείο που μας απασχολεί είναι το `App.jsx` που καλεί όλα τα αρχεία του φακέλου `components` και επιτρέπει την αλληλεπίδραση του χρήστη με το γραφικό περιβάλλον.

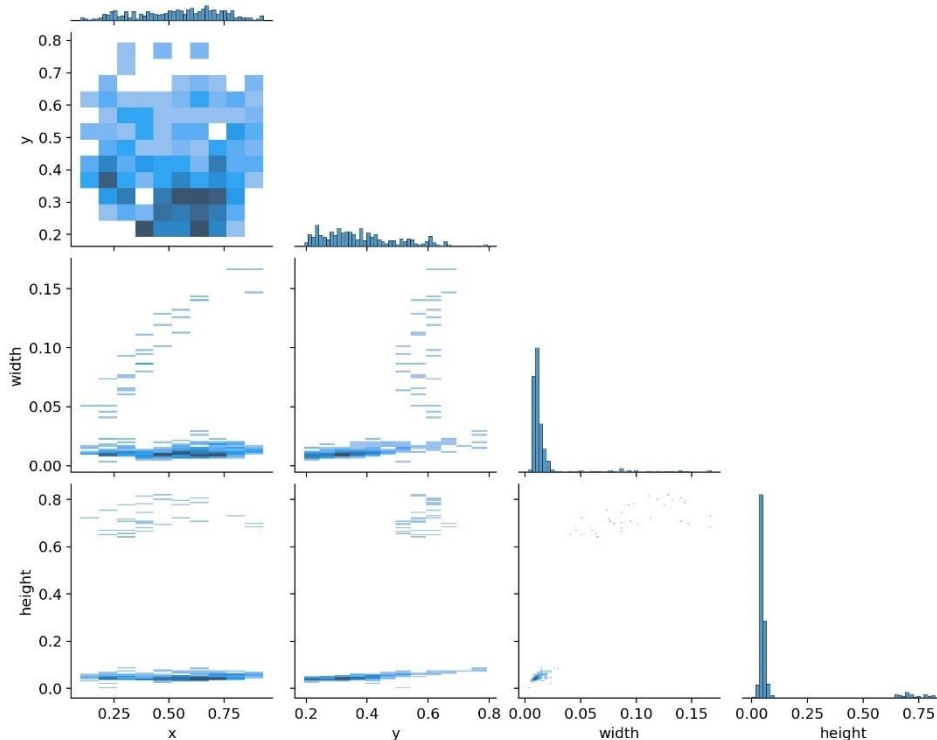
5.10 Ανάλυση του φακέλου `player_detection`

5.10.1 Εισαγωγή

Ο φάκελος αυτός είναι υπεύθυνος για την αναγνώριση παιχτών στο χώρο του γηπέδου. Χρησιμοποιούνται δύο μεγάλοι αλγόριθμοι για την ανίχνευση αυτή, ο `yoion5` και ο `deep sort`.

5.10.2 Λεπτομέρειες για την Αναγνώριση Παιχτών

Ουσιαστικά ο αλγόριθμος yolov5 εντοπίζει ένα προεκπαιδευμένο μοντέλο δεδομένων COCO, και στην συνέχεια το μεταβιβάζει στον αλγόριθμο deep sort που συνδυάζει πληροφορίες κίνησης και εμφάνισης με βάση το OSNet για την παρακολούθηση αντικειμένων. Ο αλγόριθμος yolov5 περιέχει κάποια μοντέλα για παράδειγμα αναγνώριση αυτοκινήτων, ανθρώπων κ.λπ. ωστόσο για την εύρεση ανθρώπων σε τόσο μεγάλη απόσταση ήταν απαραίτητη η εκπαίδευση ενός καινούριου μοντέλου. Ο λόγος που χρειάστηκε ο συνδυασμός yolov5 και deep sort είναι πως ο yolov5 υστερεί στο κομμάτι ταχύτητας, και πολλές φορές ακρίβειας όταν πρόκειται για εντοπισμός σε βίντεο και ιδιαίτερα όταν πρόκειται για αναγνώριση σε πραγματικό χρόνο. Το μοντέλο το οποίο εκπαιδεύτηκε βρίσκεται στον φάκελο `player_detection/yolov5/runs/train/exp55/weights/last.pt`. Για την εκπαίδευση χρησιμοποιήθηκαν οι εικόνες που βρίσκονται στον φάκελο `player_detection/yolov5/train_data` και χωρίζονται σε `images` και `labels`. Η δημιουργία των `label` έγινε μέσω του `makesense.ai` ωστόσο υπάρχουν πολλά ακόμη παρόμοια εργαλεία που βοηθούν στο `labeling` εικόνων. Η διαδικασία έγινε χειροκίνητα, και ο αλγόριθμος εκπαιδεύτηκε με 60 εικόνες για το `train` ενώ χρησιμοποιήθηκαν και 30 εικόνες για το `validation`, δηλαδή για την επιβεβαίωση του που βρίσκονται οι παίκτες σε συγκεκριμένες χρονικές στιγμές. Επιπλέον χρειάστηκε η δημιουργία του αρχείου `player_detection/yolov5/custom_data.yaml` που δίνει τις πληροφορίες για το που βρίσκονται οι εικόνες και τα `label`, δηλαδή το μονοπάτι τους και το πόσες κλάσεις δηλαδή πόσα διαφορετικά στοιχεία ψάχνει ο αλγόριθμος. Το συγκεκριμένο μοντέλο αναζητάει το κέντρο του γηπέδου και τους παίκτες της μίας από των δύο ομάδων. Στην εικόνα 5.4 μπορεί να διακριθεί το αποτέλεσμα τις αναγνώρισης παιχτών και κέντρου σε μορφή κορρολογράμματος που είναι ουσιαστικά μία ομάδα 2d ιστογραμμάτων που δείχνει κάθε άξονα των δεδομένων έναντι των υπόλοιπων αξόνων. Τα `labels` στην εικόνα είναι τα `x`, `y`, `width`, `height`.



Εικόνα 5.4: Correlogram Image of Detection

5.11 Ανάλυση του `track_players.py`

5.11.1 Εισαγωγή

Το συγκεκριμένο αρχείο καλεί την αναγνώριση παιχτών περνώντας σαν ορίσματα τις διάφορες μεταβλητές που χρειάζονται. Χρησιμοποιεί τις βιβλιοθήκες `os`, και `sys`, ενώ χρησιμοποιεί τα αρχεία `string_manipulation.py` και `filesystem_changes.py`.

5.11.2 Ανάλυση Μεθόδων

Παρακάτω θα εξηγήσουμε τις μεθόδους του συγκεκριμένου αρχείου:

- Η μέθοδος `get_python_version()` βρίσκει την έκδοση της Python που είναι εγκατεστημένη στον υπολογιστή ή στο περιβάλλον από το οποίο καλείται μέσω της εντολής `sys.version` και την απλοποιεί μέσω της μεθόδου `keep_till_second_dot()` του αρχείου `string_manipulation.py` και επιστρέφει το αποτέλεσμα.
- Η μέθοδος `get_yolo_mode()` βρίσκει τον τελευταίο φάκελο με yolo μοντέλα μέσα, και επιστρέφει το μονοπάτι του μοντέλου (π.χ. `yolon5/runs/train/exp55/weights/last.pt`). Για να έχει το συγκεκριμένο αποτέλεσμα χρησιμοποιεί τις μεθόδους `find_yolo_weight_folder()` από το αρχείο `filesystem_changes.py` και `double_backslash_to_slash()` του αρχείου `string_manipulation.py` και επιστρέφει το αποτέλεσμα.

- Η μέθοδος `track_players_given_the_weights()` δέχεται στην είσοδο της το μονοπάτι του βίντεο, και στην συνέχεια καλεί μέσω του `os.system()` την εντολή όπου εκτελεί την αναγνώριση των παιχτών. Χρησιμοποιεί τις δύο παραπάνω μεθόδους `get_python_version()` και `get_yolo_mode()` και αφού αλλάξει μονοπάτι στο οποίο βρίσκεται το εικονικό περιβάλλον μέσω του `os.chdir()` καλεί την μέθοδο `player_detection/track.py` περνώντας τα `arguments` του μοντέλου (`--yolo_mode`), του μονοπατιού του βίντεο (`--source`) των κλάσεων για την αναγνώριση (`--classes`), την επιβεβαίωση αποθήκευσης του αποτελέσματος σε μορφή βίντεο και txt (`--save-vid`, `--save-text`).

Η αναγνώριση κατά την εκτέλεση της εφαρμογής μπορεί να διαρκέσει από λίγα δευτερόλεπτα μέχρι ώρες, ανάλογα με την διάρκεια του βίντεο, την υπολογιστική δύναμη, εάν αξιοποιείται η τεχνολογία CUDA που υποστηρίζει η βιβλιοθήκη `torch` πάνω στην οποία πατάνε οι αλγόριθμοι `yoloV5` και `deepsort` κ.α.

5.12 Το αρχείο `distance.py`

5.12.1 Εισαγωγή

Το συγκεκριμένο αρχείο συγκρίνει τις θέσεις των παιχτών μεταξύ τους. Κάποιες από τις μεθόδους στο συγκεκριμένο αρχείο λειτουργούν μονάχα σε θεωρητικό επίπεδο, μιας και δεν υπήρχαν κάπου διαθέσιμα τα πραγματικά δεδομένα για σύγκριση με αυτά που βρίσκει το πρόγραμμα. Στο συγκεκριμένο αρχείο χρησιμοποιούνται οι βιβλιοθήκες `numpy`, `pandas`, `math`, ενώ χρησιμοποιείται και το αρχείο `csv_calculations.py`.

5.12.2 Ανάλυση Μεθόδων

Παρακάτω θα αναλυθούν οι μέθοδοι του συγκεκριμένου αρχείου. Το συγκεκριμένο αρχείο δέχεται το αποτέλεσμα του συνδυασμού των αλγορίθμων `deep sort` και `yoloV5` και πραγματοποιεί διάφορες αλλαγές σε αυτά για να βρει τα ζητούμενα. Στην εικόνα 5.5 φαίνεται ένα μέρος του txt αρχείου που αποθηκεύεται μετά την εύρεση των παιχτών.

```

          3  1  903  204  190  515  -1  -1  -1  -1
0         3  2  628  308  19  33  -1  -1  -1  -1
1         3  3  1068  326  15  42  -1  -1  -1  -1
2         3  4  686  233  13  32  -1  -1  -1  -1
3         3  5  1011  204  12  27  -1  -1  -1  -1
4         3  6  857  208  12  29  -1  -1  -1  -1
...
          ...

```

Εικόνα 5.5: Deepsort&YoloV5 text file

Μέσω της μεθόδου `read_and_clean()` του αρχείου `csv_calculations()` αφαιρούνται τα πεδία που περιέχουν `-1` μιας και δεν είναι απαραίτητα για τις επόμενες λειτουργίες, και δίνονται οι τίτλοι στα πεδία που παραμένουν. Τα δεδομένα πλέον έχουν την μορφή της εικόνας 5.6.

	Frame	ID	x	y	w	h
0	3	1	903	204	190	515
1	3	2	628	308	19	33
2	3	3	1068	326	15	42
3	3	4	686	233	13	32
4	3	5	1011	204	12	27
...

Εικόνα 5.6: Cleaned Deepsort & YoloV5 Data

Παρακάτω γίνεται επεξήγηση των μεθόδων που δέχονται το επεξεργασμένο πλέον dataframe για την εύρεση αποτελεσμάτων:

- Η μέθοδος `Euclidean_distance_pixels()` δέχεται το dataframe που αναφέρθηκε προηγουμένως, και επιστρέφει την απόσταση που έχει ο κάθε παίκτης από τον άλλον μεταφρασμένο σε pixels. Αρχικά δημιουργούνται δύο κενές λίστες, στην συνέχεια γίνεται σύγκριση του κάθε παίκτη με τον κάθε άλλο παίκτη. Ο τύπος που χρησιμοποιείται είναι αυτός της Ευκλείδειας μετρική (Euclidean distance), $d(p, q)^2 = (q1 - p1)^2 + (q2 - p2)^2$, όπου p και q στην εφαρμογή χρησιμοποιείται x και y. Έτσι το αποτέλεσμα της συγκεκριμένης μεθόδου είναι ένας δυσδιάστατος με τις αποστάσεις παίκτη από παίκτη (βλ. Εικόνα 5.7)

	0	1	2	3	4	5	6	7	8	9	10
0	0	423	169	286	80	75	529	270	185	139	394
1	423	0	536	145	502	349	140	195	253	540	141
2	169	536	0	418	130	226	660	419	335	57	540
3	286	145	418	0	366	211	243	63	108	413	127
4	80	502	130	366	0	155	609	348	264	80	473
5	75	349	226	211	155	0	454	197	112	207	321
6	529	140	660	243	609	454	0	266	345	656	149
7	270	195	419	63	348	197	266	0	85	405	124
8	185	253	335	108	264	112	345	85	0	319	209
9	139	540	57	413	80	207	656	405	319	0	528
10	394	141	540	127	473	321	149	124	209	528	0

Εικόνα 5.7: Distance Between players

Όπως είναι λογικό, ο κάθε παίκτης έχει μηδενική απόσταση από τον εαυτό του, και ίδια μεταξύ τους (π.χ. ο παίκτης 4 από τον παίκτη 10 έχει την ίδια απόσταση που έχει ο παίκτης 10 από τον παίκτη 4).

- Η επόμενη μέθοδος είναι η `euclidean_distance_meters()` η οποία δέχεται το καθαρισμένο dataframe καθώς και την απόσταση σε pixel που βρέθηκε προηγουμένως, και στη συνέχεια βγάζει την απόσταση των παιχτών σε μέτρα. Η συγκεκριμένη μέθοδος δεν χρησιμοποιήθηκε στην εφαρμογή σε τελικό στάδιο, μιας και δεν λειτούργησε σωστά σε πολλές από τις περιπτώσεις, και υπάρχει στον κώδικα για μελλοντική βελτίωση.
- Η μέθοδος `players_from_camera_meters()` υπολογίζει κατά προσέγγιση την απόσταση των παιχτών που αναγνώρισε ο αλγόριθμος, από την κάμερα. Χρησιμοποιήθηκε ο μαθηματικός τύπος $distance\ to\ object(mm) = \frac{f(mm) * real\ height(mm) * image\ height(px)}{object\ height(pixels) * sensor\ height(mm)}$. Όπως βλέπετε είναι απαραίτητη η γνώση του ύψους το κάθε παίκτη, το ύψος της εικόνας σε pixels, το focal length και το ύψος του φακού με τον οποίο βγαίνει το βίντεο.
- Η μέθοδος `calculate_offense_defense()` βρίσκει σε ποια χρονικά στιγμιότυπα η ομάδα πραγματοποιεί επίθεση, άμυνα και σε ποια δεν γίνεται κάτι το ενδιαφέρον στον αγώνα,

δηλαδή βρίσκεται η μπάλα στο κέντρο. Ο τρόπος που το κάνει αυτό είναι με την εύρεση του κέντρου του γηπέδου. Εάν το γήπεδο βρίσκεται αριστερά ή δεξιά τότε η ομάδα βρίσκεται στην αντίστοιχη κατάσταση επίθεσης ή άμυνας, ενώ σε περίπτωση που βρίσκεται κεντραρισμένο σε σχέση με την κάμερα τότε δεν γίνεται κάποια σημαντική φάση στον αγώνα.

- Η μέθοδος `calculate_running_distance()` είναι εξαιρετικά και αυτή από την εφαρμογή μιας και είναι εξαιρετικά δύσκολη η μετατροπή pixels σε μέτρα όταν πρόκειται για τόσο μακρινή απόσταση, και υπάρχει στον κώδικα για μελλοντική βελτιστοποίηση. Η εύρεση του πόσο έτρεξε ο παίχτης σε μονάδα μέτρησης pixel είναι εφικτή, αρκεί να συγκρίνουμε την απόσταση του παίχτη από το κέντρο του γηπέδου σε κάθε χρονική στιγμή.
- Τέλος η `manager()` και `manager_2()` δέχεται το dataframe στην είσοδο της, και καλούν τις ανάλογες μεθόδους. Η `manager_2()` δεν χρησιμοποιείται και υπάρχει για μελλοντική χρήση σε βελτιστοποιημένο version της εφαρμογής.

5.13 Το αρχείο `zoom_into_video.py`

5.13.1 Εισαγωγή

Το συγκεκριμένο αρχείο χρησιμεύει στην δημιουργία ενός βίντεο, το οποίο εστιάζει πάνω σε έναν συγκεκριμένο παίχτη στο γήπεδο. Περιέχει μεθόδους οι οποίες έχουν άμεση σχέση με το βίντεο, την διάρκεια του κ.λπ. Χρησιμοποιεί τις βιβλιοθήκες `glob`, `cv2` και `json`, ενώ χρησιμοποιεί και διάφορες μεθόδους από τα αρχεία `filesystem_changes.py`, `string_manipulation.py` και `csv_calculations.py`.

5.13.2 Ανάλυση Μεθόδων

Παρακάτω γίνεται επεξήγηση των λειτουργιών των μεθόδων του συγκεκριμένου αρχείου:

- Η μέθοδος `get_video_fps()` δέχεται ως είσοδο το μονοπάτι του βίντεο, και μέσω της μεθόδου `cv2.VideoCapture()` και `cv2.CAP_PROP_FPS` επιστρέφει τα καρέ ανά δευτερόλεπτο του βίντεο.
- Η μέθοδος `find_video_sec_length()` επιστρέφει την διάρκεια του βίντεο σε δευτερόλεπτα. Διαβάζει το αρχείο «`game_details.json`» που περιέχει διάφορα στοιχεία για τον συγκεκριμένο αγώνα που αποθηκεύτηκαν σε προηγούμενος και βρίσκει το μονοπάτι του βίντεο. Έπειτα έχοντας τα καρέ ανά δευτερόλεπτο με τον ίδιο τρόπο που εξηγήσαμε προηγούμενος, μπορεί να βρεθεί η διάρκεια του βίντεο με την διαίρεση των συνολικών καρέ διά των καρέ ανά δευτερόλεπτο, $duration = frame\ count / fps$.
- Η μέθοδος `resize_image()` κόβει (crop) μία εικόνα σε ένα ανάλογο μέγεθος. Δέχεται ως είσοδο την εικόνα και ένας αριθμός που δηλώνει το πόσο πρέπει να κοπεί η εικόνα, και μέσω της μεθόδου `cv2.resize` επιστρέφει την νέα αποκομμένη εικόνα.
- Η μέθοδος `create_zoom_version()` δέχεται στην είσοδο της την εικόνα, το που βρίσκεται στους άξονες `x` και `y`. Στην συνέχεια την διαβάζει, και εστιάζει στην εικόνα βάση του τύπου `img[y - 20: y + 50, x - 20: x + 50]`, δηλαδή εστιάζει 20px αριστερά του `x` και `y` έως και 50px δεξιά του `x` και του `y`. Στη συνέχεια καλεί την μέθοδο `resize_image()` που αναφέρθηκε προηγούμενος για να επιτύχει το κόψιμο της εικόνας.
- Η μέθοδος `zoom_frame_player()` δέχεται ως είσοδο το καρέ, το `id` του παίχτη, και το αρχείο `txt` που δημιούργησε ο αλγόριθμος αναγνώρισης και μέσω των δύο προηγούμενων μεθόδων δημιουργεί την νέα εικόνα που είναι εστιασμένη στον παίχτη για το συγκεκριμένο καρέ και την αποθηκεύει στον φάκελο `Exported Frames/frame.jpg`.
- Η μέθοδος `export_all_frames()` εξάγει τα καρέ του βίντεο ως εικόνες. Δέχεται ως είσοδο το μονοπάτι του βίντεο. Αρχικά διαγράφει τον φάκελο `Exported Frames` εάν αυτός υπάρχει, μιας και είναι πιθανό να περιέχει τις εικόνες από προηγούμενη εκτέλεση της εφαρμογής, και τον ξανά δημιουργεί. Στη συνέχεια διαβάζει καρέ προς καρέ το βίντεο, και για όσο το καρέ αυτό

υπάρχει, το αποθηκεύει στον φάκελο Exported Frames/framex.jpg όπου x ο αριθμός του frame.

- Η μέθοδος `export_10_second_frames()` εξάγει τα καρέ του βίντεο ως εικόνες για 10 δευτερόλεπτα. Αυτό γίνεται έτσι ώστε να μπορεί να εστιάσει ο διαιτητής μόνο στις ενέργειες του παίκτη στη συγκεκριμένη φάση του αγώνα. Η εξαγωγή γίνεται από 3 δευτερόλεπτα πριν το καρέ που επισήμανε ο χρήστης ότι θέλει να γίνει η μεγέθυνση του βίντεο, έως και 7 δευτερόλεπτα μετά, ώστε να έχει μία σφαιρική άποψη για την συγκεκριμένη φάση. Όπως και η προηγούμενη μέθοδος αποθηκεύει καρέ προς καρέ στον φάκελο Exported Frames/framex.jpg όπου x ο αριθμός του frame.
- Κλείνοντας με το συγκεκριμένο αρχείο, η μέθοδος `zoom_player()` πραγματοποιεί την μεγέθυνση του βίντεο πάνω σε έναν παίκτη και δημιουργεί το νέο βίντεο αποθηκεύοντας το. Αρχικά δέχεται το txt αποτέλεσμα του αλγορίθμου αναγνώρισης παιχτών, και το διαβάζει καθαρίζοντας το μέσω της μεθόδου `read_and_clean()` του αρχείου `csv_calculations.py`. Στη συνέχεια αποθηκεύει τις τιμές για τα καρέ και την τοποθεσία του συγκεκριμένου παίκτη σε λίστες. Έπειτα ελέγχει ποιες εικόνες έχουν εξαχθεί από το βίντεο. Μιας και η μέθοδος εξαγωγής εικόνων εκτελείται πριν από αυτή την μέθοδο η μέθοδος θεωρεί δεδομένο πως πρέπει να εφαρμόσει την εστίαση – μεγέθυνση στα συγκεκριμένα καρέ. Έτσι δημιουργεί μία λίστα με το μονοπάτι των καρέ αυτών και τα ταξινομεί. Η επόμενη ενέργεια που πραγματοποιεί είναι το να προσπεράσει εικόνα προς εικόνα και χρησιμοποιώντας την μέθοδο `create_zoom_version()` να δημιουργήσει την εικόνα η οποία εστιάζει στον παίκτη. Έπειτα την αποθηκεύει στον φάκελο Exported Frames/zoomed images/zoomx.jpg όπου x ο αριθμός του καρέ. Αφού τελειώσει με την δημιουργία και την αποθήκευση των εικόνων, τις μετατρέπει σε βίντεο χρησιμοποιώντας μία λούπα, και την μέθοδο `cv2.VideoWriter()`, δημιουργεί τον φάκελο Zoomed-in Video μέσω της μεθόδου `create_zoom_video_folder()` του αρχείου `filesystem_changes.py` και τέλος αποθηκεύει το βίντεο στον φάκελο αυτό.

5.14 Το αρχείο `graph_generator.py`

5.14.1 Εισαγωγή

Το συγκεκριμένο αρχείο δημιουργεί τα διάφορα γραφήματα που εμφανίζονται κατά την εκτέλεση του προγράμματος όταν ο χρήστης πατήσει σε κάποιο δεδομένο στο γραφικό στοιχείο εμφάνισης ενεργειών των παιχτών. Χρησιμοποιεί τις βιβλιοθήκες `pandas`, `matplotlib.pyplot` και `numpy`, ενώ κάνει χρήση και του αρχείου `string_manipulation.py`.

5.14.2 Ανάλυση Μεθόδων

Οι μέθοδοι που περιέχει το αρχείο αυτό είναι οι εξής:

- Η μέθοδος `all_game_all_player_actions()` δέχεται στην είσοδο της το csv αρχείο με τις ενέργειες όλων των παιχτών. Έπειτα ξεχωρίζει το όνομα του παίκτη, την ενέργεια, και την χρονική στιγμή που την πραγματοποίησε και την αποθηκεύει σε τρεις διαφορετικές μεταβλητές. Στη συνέχεια δημιουργεί ένα γράφημα τύπου διασποράς (plot), με τον άξονα x να περιλαμβάνει τις ενέργειες και τον άξονα y να περιλαμβάνει τις χρονικές στιγμές αυτών. Επιπλέον για την κάθε ενέργεια μέσω της `plt.annotate()` εμφανίζεται και το όνομα του παίκτη πάνω στην ενέργεια. Το γράφημα στη συνέχεια εμφανίζεται στον χρήστη.
- Η μέθοδος `all_game_single_player_actions()` δέχεται στην είσοδο της το αρχείο με όλες τις ενέργειες από όλους τους παίκτες και το όνομα ενός συγκεκριμένου παίκτη. Έπειτα ξεχωρίζει όλες τις ενέργειες που έχει πραγματοποιήσει, και τις κρατάει διαγράφοντας από το dataframe τις υπόλοιπες. Στη συνέχεια μετράει πόσες φορές έχει γίνει η κάθε ενέργεια και δημιουργεί

ένα γράφημα πίτας (pie) με τις ενέργειες αυτές. Το γράφημα στη συνέχεια εμφανίζεται στον χρήστη.

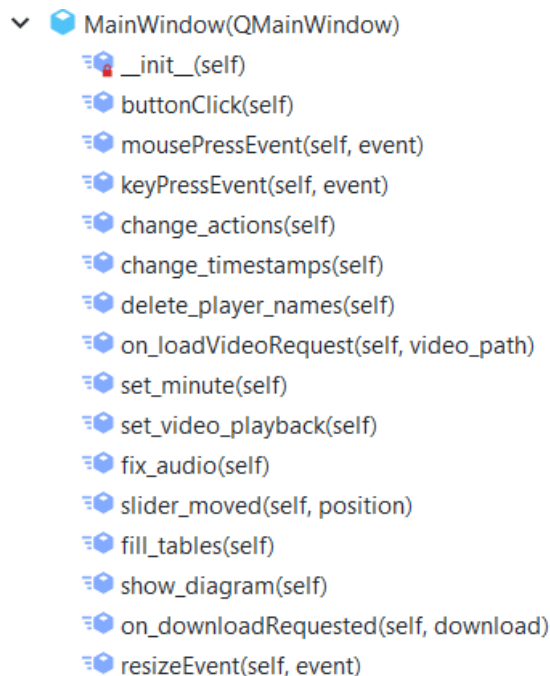
- Η μέθοδος `all_game_specific_action()` δέχεται στην είσοδο της το αρχείο με όλες τις ενέργειες που έχουν γίνει από όλους τους παίκτες και το όνομα μίας ενέργειας και κρατάει στο `dataframe` μονάχα την ενέργεια αυτή. Έπειτα την ξεχωρίζει τα ονόματα και τις φορές των παιχτών που έχουν πραγματοποιήσει την συγκεκριμένη ενέργεια. Τέλος δημιουργεί ένα ραβδόγραμμα (bar) με τις φορές που εμφανίστηκε αυτή η ενέργεια και το εμφανίζει το γράφημα στο χρήστη.
- Η μέθοδος `all_game_action_family()` εκτελεί παρόμοιες ενέργειες με την προηγούμενη μέθοδο. Η διαφορά είναι πως αντί για μία συγκεκριμένη ενέργεια δέχεται μία οικογένεια ενεργειών όπως είναι π.χ. η άμυνα. Αφού ξεχωρίσει όλες τις ενέργειες αυτής της οικογένειας μετράει πόσες φορές εμφανίστηκε η κάθε ενέργεια που ανήκει σε αυτή (π.χ. πόσες φορές πραγματοποιήθηκε τάκλιν, κεφαλιά κ.λπ.). Έχοντας συγκεντρώσει τα ονόματα των ενεργειών και τις φορές που έγιναν, δημιουργεί ένα γράφημα τύπου πίτας και το εμφανίζει στον χρήστη.

Έτσι ο χρήστης μπορεί να έχει σε διαγραμματική μορφή τις ενέργειες που πραγματοποίησε συνολικά η ομάδα και ατομικά ο κάθε παίκτης για να μπορεί να αποφασίσει τις αλλαγές που θα κάνει στους μετέπειτα αγώνες.

5.15 Ανάλυση της `main.py`

5.15.1 Εισαγωγή

Η `main.py` όπως αναφέρθηκε και προηγουμένως είναι ο συνδετικός κρίκος ανάμεσα στην λειτουργικότητα της εφαρμογής και στο περιβάλλον με το οποίο μπορεί να αλληλοεπιδράσει ο χρήστης. Οτιδήποτε πληκτρολογεί, ή πιέζει με τον κέρσορα είναι προσβάσιμο από την `main.py` και συγκεκριμένα από την κλάση `MainWindow()`. Στην εικόνα 5.8 παρουσιάζεται η δομή της κλάσης `MainWindow()`.



Εικόνα 5.8: `MainWindow()` Structure

5.15.2 Ανάλυση των import

Τα import στην main.py μπορούν να χωριστούν στις εξωτερικές βιβλιοθήκες είτε αυτές είναι ήδη εγκατεστημένες στην Python είτε όχι, στα τοπικά αρχεία, και στα import για το γραφικό περιβάλλον.

- Οι βιβλιοθήκες που χρησιμοποιούνται είναι οι sys, os, time και json που υπάρχουν ήδη στην python και η pandas που εγκαταστάθηκε.
- Τα τοπικά αρχεία που χρησιμοποιούνται είναι τα save_data.py, string_manipulation.py, youtube_downloader.py, filesystem_changes.py, csv_calculations.py, zoom_into_video.py, graph_generator.py, track_players.py, distance.py. Από αυτά χρησιμοποιούνται κάποιες αν όχι όλες οι μέθοδοι.
- Για το γραφικό περιβάλλον γίνεται import των αρχείων του τοπικού φακέλου modules, και των αρχείων του τοπικού φακέλου widgets. Επίσης από την βιβλιοθήκη PySide6 χρησιμοποιείται το QtCore, QtWidgets, QtWebEngineWidgets, από το PySide6.QtWebEngineCore το QWebEngineProfile, από το PySide6.QtMultimediaWidgets το QVideoWidget, και από το PySide6.QtMultimedia τα QMediaPlayer και QAudioOutput.

5.15.3 Οι global μεταβλητές

Οι global μεταβλητές είναι μεταβλητές που η ύπαρξη τους ορίζεται εκτός της MainWindow(), και αρχικοποιούνται στην μέθοδο __init__(). Ο λόγος που υπάρχουν είναι η χρήση τους από οποιαδήποτε μέθοδο της MainWindow(). Αυτές οι μεταβλητές στη συγκεκριμένη περίπτωση είναι οι widgets που αρχικοποιείται ως το ui της εφαρμογής, το media_player μία κενή λίστα με όνομα names η οποία προορίζεται για να περιέχει τα ονόματα των παιχτών, μία παρόμοια λίστα με όνομα actions στην οποία κατά την εκκίνηση του προγράμματος δίνονται οι διάφορες ενέργειες, και δύο ακόμη μεταβλητές οι date_stamps και timestamps.

5.15.4 Ανάλυση της μεθόδου __init__()

Η μέθοδος __init__() είναι η μέθοδος που καλείται με το που κληθεί η κλάση MainWindow() δηλαδή με το που ζητηθεί η φόρτωση του γραφικού περιβάλλοντος. Σε αυτή δίνονται οι τιμές κάποιων από των global μεταβλητών, δημιουργείται ο φάκελος όπου θα αποθηκευτούν τα διάφορα αρχεία κατά την χρήση της εφαρμογής, δίνεται ο τίτλος και η περιγραφή της εφαρμογής, ενεργοποιείται η κύλιση του αριστερού μενού, ενώ ενεργοποιούνται οι μέθοδοι που θα εκτελεστούν κατά το πάτημα ή την πληκτρολόγηση αντίστοιχων αντικειμένων. Επιπλέον πολύ σημαντική ενέργεια που γίνεται με την φόρτωση της __init__() είναι η τιμή που δίνεται στην μεταβλητή widgets και αυτή είναι το self.ui δηλαδή πλέον η widgets ελέγχει όλα τα αντικείμενα του γραφικού περιβάλλοντος και η κλήση σε αυτά γίνεται μέσω της μεταβλητής αυτής.

Παρακάτω δίνεται περιγραφή του πως καλείται μία μέθοδος με την ανάλογη ενέργεια, για καθένα από τα αντικείμενα του γραφικού περιβάλλοντος:

- Για την σύνδεση ενός κουμπιού με την εφαρμογή χρησιμοποιείται η εντολή widgets.btn_id.clicked.connect(self.method). Όπου btn_id είναι το id του κουμπιού, και method είναι η μέθοδος η οποία θα κληθεί με το πάτημα του κουμπιού.
- Αντίστοιχα για την αρχικοποίηση του μονοπατιού που αποθηκεύεται ένα αρχείο χρησιμοποιείται η εντολή widgets.formation.page.profile().setDownloadPath(path). Όπου formation είναι το γραφικό στοιχείο από το οποίο θα αποθηκευτεί το αρχείο, και path είναι το μονοπάτι του που θα αποθηκευτεί.

- Για την φόρτωση του της σελίδας δημιουργίας τακτικής χρησιμοποιείται η εντολή `widgets.formation.load(QUrl(path))` όπου `path` το μονοπάτι που βρίσκεται το αρχείο εκκίνησης της εφαρμογής.
- Τα Combo Boxes ενεργοποιούνται με την αλλαγή του `index` τους, δηλαδή με την αλλαγή της τιμής που έχει το επιλεγμένο στοιχείο. Έτσι χρειάζεται η εντολή `widgets.combobox_id.currentIndexChanged.connect(self.method)` όπου `combobox_id` το `id` του combo box, και όπου `method` η μέθοδος που θα κληθεί με την αλλαγή αυτού.

Τέλος δίνονται τα απαραίτητα αισθητικά στοιχεία στην εφαρμογή και πραγματοποιείται η εμφάνιση του γραφικού περιβάλλοντος.

5.15.5 Ανάλυση της μεθόδου `buttonClick()`

Οποιοδήποτε κουμπί και αν πατηθεί θα κληθεί η μέθοδος αυτή. Ωστόσο δεν μπορεί να καταλάβει ποιο από τα κουμπιά την καλεί, έτσι μέσω της μεθόδου `self.sender()` σε συνδυασμό με την `objectName()` μπορεί να αναγνωρίσει ποιο κουμπί την κάλεσε και να βρει το `id` του. Έτσι η επιλογή του ποια ενέργεια θα εκτελεστεί με το κλικ ενός κουμπιού γίνεται με την εντολή `if`. Παρακάτω δίνονται συνοπτικά οι λειτουργίες που εκτελούνται ανάλογα με το `id` του κουμπιού:

- Σε περίπτωση που το κουμπί έχει `id` `'themeBtn'` εφαρμόζεται ένα ανοιχτόχρωμο ή σκουρόχρωμο θέμα ανάλογα με το τι θέμα υπήρχε προηγουμένως. Δηλαδή αν ήταν σκουρόχρωμο εφαρμόζεται ανοιχτόχρωμο και το αντίθετο. Αυτό γίνεται μέσω της εντολής `UIFunctions.theme(self, path, True)` όπου `path` το μονοπάτι του `.qss` αρχείου με το θέμα.
- Τα επόμενα κουμπιά έχουν πολύ παρόμοιες λειτουργίες γι' αυτό δεν θα αναλυθούν ξεχωριστά. Αυτά τα κουμπιά αφορούν το αριστερό μενού της εφαρμογής που ευθύνεται για την πλοήγηση του χρήστη. Συγκεκριμένα εάν το κουμπί έχει `id` `'btn_home'`, `'btn_import_video'`, `'btn_formation'`, `'btn_video_player'` ή `'btn_post_game_details'`, τότε οι ενέργειες που γίνονται είναι οι εξής. Αρχικά αλλάζει ο τίτλος του γραφικού στοιχείου με έναν νέο που υποδηλώνει το γραφικό στοιχείο που βλέπουμε μέσω της εντολής `widgets.titleRightInfo.setText(text)` όπου `text` το αντίστοιχο κείμενο, αλλάζει το κεντρικό widget σε αυτό το οποίο θέλουμε να εμφανίσει με την εντολή `widgets.stackedWidget.setCurrentWidget(widgets.id)` όπου `id` το `id` του widget που θέλουμε να εμφανιστεί, επαναφέρεται το στυλ όλων των κουμπιών του μενού μέσω της εντολής `UIFunctions.resetStyle(self, id)` και αυτό για να διαγραφεί η σκίαση που έχει το προηγούμενο κουμπί, και εφαρμόζεται η σκίαση στο κουμπί που πατήθηκε μέσω της εντολής `btn.setStyleSheed(UIFunctions.selectMenu(btn.styleSheet()))`.
- Το επόμενο κουμπί `'btn_local_footage'` ανήκει στην σελίδα επιλογής τρόπου επισύναψης βίντεο εμφανίζει το γραφικό στοιχείο επισύναψης βίντεο τοπικά από τον υπολογιστή, και αλλάζει τον τίτλο του γραφικού στοιχείου. Ο τρόπος που γίνεται αυτό είναι ίδιος με τον τρόπο που αναφέρθηκε προηγουμένως.
- Ακολουθεί το κουμπί `'local_video_file_button'` που ανήκει στο γραφικό στοιχείο τοπικής επισύναψης βίντεο και εμφανίζει ένα αναδυόμενο παράθυρο του αποθηκευτικού χώρου του χρήστη για την επισύναψη του βίντεο. Αρχικά δημιουργείται μία μεταβλητή που περιέχει τους υποστηριζόμενους τύπους αρχείων (`avi`, `mp4`, `mpg`), και μέσω της εντολής `QFileDialog.getOpenFileName(self, 'Open Video', fc.downloads_path(), supported_formats)` εμφανίζεται το αναδυόμενο παράθυρο έχοντας ως αρχικό μονοπάτι τον φάκελο των λήψεων.
- Το επόμενο κουμπί είναι το `'local_next_page_button'`, ανήκει στο γραφικό στοιχείο τοπικής επισύναψης βίντεο. Αρχικά ελέγχει εάν το αρχείο που υπάρχει στο μονοπάτι, δηλαδή το αρχείο που έχει επιλέξει ο χρήστης είναι υπαρκτό μέσω της μεθόδου `check_if_file_exists()` του αρχείου `filesystem_changes.py`, και σε περίπτωση όπου αυτό είναι αληθές αποθηκεύει τα δεδομένα που έχει συμπληρώσει ο χρήστης στην συγκεκριμένη σελίδα μέσω της μεθόδου

`save_pre_local_video_data()` του αρχείου `save_data.py`. Έπειτα αλλάζει τον τίτλο και το κεντρικό γραφικό στοιχείο εμφανίζοντας το επόμενο που είναι αυτό της δημιουργίας τακτικής. Σε περίπτωση που το κουμπί `'local_player_detection_button'` είναι επιλεγμένο, τότε θέτει σε εκτέλεση την αναγνώριση παιχτών στο γήπεδο, χρησιμοποιώντας την μέθοδο `track_players_given_the_weights()` του αρχείου `track_players.py` και αλλάζει τον πίνακα των χρονικών στιγμών καλώντας την μέθοδο `change_timestamps()` του ίδιου αρχείου. Αντίθετα σε περίπτωση το μονοπάτι του αρχείου που έχει δώσει ο χρήστης δεν είναι σωστό τότε εμφανίζεται το κατάλληλο μήνυμα στο σημείο που θα εμφανιζόταν κανονικά το μονοπάτι αυτό.

- Σε περίπτωση όπου το `id` του κουμπιού είναι το `'local_previous_page_button'` τότε αλλάζει το γραφικό στοιχείο εμφανίζοντας αυτό της επιλογής επισύναψης βίντεο.
- Αν το κουμπί έχει όνομα `'btn_cloud_footaget'` τότε εμφανίζεται το γραφικό στοιχείο λήψης και επισύναψης βίντεο από το YouTube.
- Αν το κουμπί έχει όνομα `'cloud_video_file_button'` που είναι κουμπί που βρίσκεται στο γραφικό στοιχείο λήψης και επισύναψης βίντεο από το YouTube τότε σηματοδοτεί την λήψη του URL που δόθηκε και μέσω της μεθόδου `save_video_to_downloads()` του αρχείου `youtube_downloader.py` ξεκινάει η λήψη του βίντεο. Επιπλέον υπάρχει ένα progress bar, το οποίο μόλις φτάσει στο 100% αλλάζει το κείμενο του σε "Completed". Το progress λειτουργεί καθαρά για αισθητικούς καθησυχαστικούς λόγους μιας και δεν αντικατοπτρίζει την πραγματική λήψη του βίντεο. Σε περίπτωση που δεν καταφέρει να πραγματοποιηθεί η λήψη του βίντεο τότε εμφανίζεται το ανάλογο μήνυμα στον χρήστη.
- Το επόμενο κουμπί είναι το `'cloud_next_page_button'` που είναι κουμπί που ανήκει στο ίδιο γραφικό στοιχείο με τα προηγούμενα κουμπιά, τότε γίνεται ο απαραίτητος έλεγχος για την επιβεβαίωση ότι το αρχείο έχει ληφθεί μέσω της μεθόδου `check_if_file_exists()` του αρχείου `filesystem_changes.py` και αποθηκεύει τα δεδομένα που έχει τοποθετήσει προσθέσει ο χρήστης σε αυτό το γραφικό στοιχείο μέσω της μεθόδου `save_pre_cloud_video_data()` του αρχείου `save_data.py`. Στη συνέχεια αλλάζει τον τίτλο και το κεντρικό γραφικό στοιχείο, εμφανίζοντας αυτό της δημιουργίας τακτικής, και σε περίπτωση όπου το κουμπί `'cloud_player_detection_button'` είναι επιλεγμένο τότε εκκινεί την αναγνώριση των παιχτών στο βίντεο που δόθηκε. Αν το βίντεο δεν υπάρχει τότε εμφανίζεται το ανάλογο μήνυμα.
- Σε περίπτωση όπου το κουμπί έχει `id` `'cloud_previous_page_button'` τότε αλλάζει το γραφικό στοιχείο στο προηγούμενο δηλαδή αυτό της επιλογής επισύναψης βίντεο είτε τοπικά είτε μέσω YouTube.
- Αν το κουμπί έχει `id` `'formation_next_page_button'` που ανήκει στη σελίδα της δημιουργίας τακτικής τότε γίνεται η προσπάθεια του να προστεθούν τα ονόματα στην global λίστα `names`. Αυτό γίνεται με συνδυασμό τριών μεθόδων, της `lineup_manager()` του `save_data.py`, του `double_backslash_to_slash()` του `string_manipulation.py` και του `find_last_created_folder()` του `filesystem_changes.py` σε συνδυασμό με το string `'lineup.json'`. Έτσι το πρόγραμμα βρίσκει το json αρχείο το οποίο έχει την τακτική, διαβάσει τα δεδομένα και τα επιστρέφει μέσω των μεθόδων που καλεί η `lineup_manager()`. Αφού βρει τα ονόματα, διαγράφει τα ήδη υπάρχοντα ονόματα εάν αυτά υπάρχουν μέσω της `delete_player_names()` και προσθέτει τα νέα ονόματα. Επιπλέον αλλάζει ο τίτλος του γραφικού στοιχείου και το κεντρικό γραφικό στοιχείο, και φορτώνει το βίντεο που πρόκειται να παίξει στο γραφικό στοιχείο αναπαραγωγής βίντεο μέσω της εντολής `on_load_video_request()`. Επίσης απενεργοποιείται το κουμπί εκκίνησης του βίντεο και ενεργοποιούνται τα κουμπιά παύσης και τερματισμού μέσω της εντολής `widgets.button_id.setDisabled(True/False)`.
- Σε περίπτωση που το κουμπί έχει `id` `'formation_previous_page_button'` τότε επιστρέφει στο γραφικό στοιχείο επιλογής τρόπου επισύναψης βίντεο, αλλάζει ο τίτλος και το αισθητικό κομμάτι των κουμπιών.

- Εάν το id του κουμπιού είναι 'play_video_button', 'pause_video_button' ή stop_video_button εκτελούνται παρόμοιες λειτουργίες, Αυτά είναι τα κουμπιά αναπαραγωγής, παύσης, και τερματισμού του βίντεο αντίστοιχα, έτσι κατά το πάτημα οποιουδήποτε εξ' αυτών απενεργοποιείται το κουμπί που πατήθηκε, και ενεργοποιούνται τα άλλα δύο μιας και δεν γίνεται να αναπαράγει βίντεο που έχει ήδη ξεκινήσει η αναπαραγωγή του ή να το σταματήσει ενώ είναι ήδη σταματημένο.
- Αν το id του κουμπιού είναι 'player_detection_button' της σελίδας αναπαραγωγής βίντεο είναι επιλεγμένο τότε δίνεται η εντολή για αναπαραγωγή του βίντεο που έχει επεξεργαστεί ο υπολογιστής εάν αυτό υπάρχει τότε ξεκινάει η αναπαραγωγή του βίντεο που έχει επεξεργαστεί ο υπολογιστής. Αν αντίστοιχα δεν είναι επιλεγμένο δίνεται η εντολή αναπαραγωγής του κανονικού βίντεο.
- Σε περίπτωση που το κουμπί έχει id 'add_action' που είναι η προσθήκη ενέργειας που πραγματοποίησε ένας παίχτης, τότε μέσω της εντολής widgets.combobox_id.currentText() παίρνει το πρόγραμμα το κείμενο για το όνομα του παίχτη, την οικογένεια της ενέργειας, το όνομα της ενέργειας και τη χρονική στιγμή που έγινε βάση των δευτερολέπτων που έχουν αναπαραχθεί στο βίντεο και τα αποθηκεύει σε ένα csv αρχείο μέσω της μεθόδου add_to_csv() του αρχείου csv_calculations.py.
- Αν το id του κουμπιού είναι 'return_to_lineup_builder' που είναι το κουμπί για την επιστροφή στο γραφικό στοιχείο δημιουργίας τακτικής τότε αλλάζει το γραφικό στοιχείο για να μπορέσει ο χρήστης να αλλάξει την τακτική του.
- Εάν το κουμπί έχει id 'zoom_into_player_button' που είναι το κουμπί για την εστίαση σε έναν παίχτη, τότε δημιουργούνται μεταβλητές για το τοπικό βίντεο, για τον παίχτη στον οποίο θέλουμε να εστιάσουμε και για το txt αρχείο που ανέλυσε ο υπολογιστής κατά τον εντοπισμό παιχτών. Στη συνέχεια γίνεται ο έλεγχος για το αν η τιμή του παίχτη στον οποίο πρέπει να γίνει η εστίαση είναι κενή και αν δεν είναι καλούνται οι μέθοδοι export_10_seconds_frames() και zoom_player() του αρχείου zoom_into_video.py, δημιουργείται το ανάλογο βίντεο και τέλος ξεκινάει η αναπαραγωγή του. Σε περίπτωση που δεν έχει δοθεί κάτι στο input του παίχτη για εστίαση τότε ξεκινάει η αναπαραγωγή του αρχικού βίντεο. Σε όλες αυτές τις ενέργειες προσαρμόζονται και ανάλογα τα κουμπιά αναπαραγωγής του βίντεο.
- Σε περίπτωση που το κουμπί έχει id 'show_post_game_button' τότε εμφανίζεται το γραφικό στοιχείο παρουσίασης ενεργειών που έγιναν κατά την διάρκεια του αγώνα και καλείται η μέθοδος fill_tables() που γεμίζει τις λίστες του γραφικού στοιχείο με δεδομένα.

5.15.6 Ανάλυση των μεθόδων mousePressEvent() και keyPressEvent()

Οι συγκεκριμένες μέθοδοι είναι υπεύθυνες για τις ενέργειες που πραγματοποιούνται κατά την το πάτημα του ποντικιού ή κατά την πληκτρολόγηση. Μέσω του mousePressEvent() γίνεται εφικτή η μετακίνηση του παραθύρου κατά το συνεχόμενο πάτημα σε μία νέα τοποθεσία που ορίζει ο χρήστης ενώ μέσω του πληκτρολογίου μπορούν να υπάρξουν διάφορες συντομεύσεις πληκτρολογίου.

5.15.7 Οι Μέθοδοι change_actions(), change_timestamps(), delete_player_names()

Οι δύο αυτοί μέθοδοι έχουν παρόμοιες λειτουργίες που λειτουργούν με παρόμοιο τρόπο. Και οι δύο μέθοδοι έχουν φτιαχτεί για την αλλαγή των διάφορων στοιχείων που υπάρχουν μέσα σε διαφορετικά combo boxes. Έτσι η change_actions() καλείται όταν αλλάξει το combo box της οικογένειας μίας ενέργειας, διαγράφονται όλα τα στοιχεία που είχε το combo box των ενεργειών μέσω της εντολής widgets.combobox_id.removeItem() όπου combobox_id το id του combo box και προστίθενται οι ενέργειες που ανήκουν στη συγκεκριμένη ενέργεια οικογενειών. Αντίστοιχα λειτουργεί και η change_timestamps() η οποία έχει ως στόχο την αλλαγή των χρονικών στιγμών στο αριστερό combo box ανάλογα με την περίπτωση επίθεσης, άμυνας ή στην περίπτωση που δεν γίνεται κάποια φάση.

Συλλέγει δηλαδή τις χρονικές στιγμές που έχουν γίνει οι αντίστοιχες φάσεις στον αγώνα μέσω της μεθόδου `manager()` του αρχείου `distance.py`, και αφού κάνει αυτό διαγράφει τις παλιές και προσθέτει τις ανάλογες χρονικές σε σχέση με το τι είναι επιλεγμένο στο διπλανό `combo box`. Η `delete_player_names()` απλώς διαγράφει τα ονόματα των παιχτών και δεν προσθέτει κάτι.

5.15.8 Η μέθοδος `on_load_video_request()`

Η μέθοδος αυτή είναι απαραίτητη για την αναπαραγωγή των διάφορων βίντεο. Δέχεται στην είσοδο της μονάχα μία μεταβλητή με το μονοπάτι του βίντεο. Αρχικά επιλέγει τις `global` μεταβλητές `audio_output` και `video_player` ως έξοδο ήχου και εικόνας μέσω των εντολών `self.media_player.setAudioOutput()` και `self.media_player.setVideoOutput()`. Έπειτα επιλέγει την πηγή του βίντεο μέσω της εντολής `self.media_player.setSource(QUrl(path))` όπου `path` το μονοπάτι του βίντεο που δέχτηκε στην είσοδο. Έπειτα επιλέγει την ταχύτητα αναπαραγωγής του βίντεο που την αρχικοποιεί με την τιμή 1 μέσω της εντολής `self.media_player.setPlaybackRate(1)` και ξεκινάει την αναπαραγωγή του βίντεο ενώ προσαρμόζει ανάλογα τα κουμπιά αναπαραγωγής βίντεο.

5.15.9 Η μέθοδος `set_minute()`

Μέσω της μεθόδου `set_minute()` αναπαράγεται το βίντεο σε μία συγκεκριμένη χρονική στιγμή. Αυτό γίνεται με την επιλογή που κάνει ο χρήστης στα κουμπιά των επιθέσεων, άμυνων και άλλων ενεργειών. Έτσι μέσω της μεθόδου `find_video_sec_length()` του αρχείου `zoom_into_video.py` παίρνει τη συνολική διάρκεια του βίντεο, μέσω του `widgets.od_timestamps_combobox.currentText()` παίρνει την χρονική στιγμή που θέλει να δει ο χρήστης, την μετατρέπει σε δευτερόλεπτα και αφαιρεί 10 από αυτά ώστε να ξεκινήσει το βίντεο λίγο νωρίτερα, και μέσω της εντολής `self.media_player.setPosition(seconds%)` αναπαράγει το βίντεο. Το πρόβλημα είναι ότι πρέπει να γίνει χρήση ποσοστού για την εντολή `setPosition()` έτσι όπου `seconds%` ο τύπος $int\left(\frac{duration*seconds}{max\ game\ duration}\right)$.

5.15.10 Η μέθοδος `fix_audio()`

Η μέθοδος αυτή ελέγχει την ταχύτητα αναπαραγωγής βίντεο και αν το βίντεο έχει ήχο μέσω της εντολής `self.audio_output.isMuted()` και αν το `index` της ταχύτητας αναπαραγωγής αντιστοιχεί στην ταχύτητα 1 τότε επιστρέφει τον ήχο στο βίντεο μέσω της εντολής `self.audio_output.setMuted(False)` και `self.media_player.setAudioOutput(self.audio_output)`. Σε περίπτωση που ο παραπάνω έλεγχος είναι ψευδής τότε γίνεται `mute` στον ήχο και αυτό για να αποφύγουμε την αλλοίωση του ήχου.

5.15.11 Η μέθοδος `slider_moved()`

Η μέθοδος αυτή είναι υπεύθυνη για την αλλαγή της διάρκειας του βίντεο μέσω του `slider`. Συγκεκριμένα δέχεται το ποσοστό που έχει μετακινηθεί το `slider`, και εφαρμόζει το αντίστοιχο ποσοστό στην αναπαραγωγή του βίντεο. Επιπλέον εμφανίζει το ποσοστό αυτό στην μπάρα προόδου.

5.15.12 Η μέθοδος `fill_tables()`

Η μέθοδος αυτή γεμίζει τις λίστες των στατιστικών με τις απαραίτητες πληροφορίες.

- Για τον πρώτο πίνακα δηλαδή αυτόν με τις πληροφορίες που προστέθηκαν πριν τον αγώνα διαβάζει το αρχείο `'game_details.json'` και περνάει τις πληροφορίες του σε μία μεταβλητή τύπου λεξικού. Έπειτα την χωρίζει αυτή σε δύο λίστες, με την μία να περιέχει τους τίτλους και την άλλη να περιέχει τα περιεχόμενα των τίτλων αυτών. Έπειτα μέσω των εντολών `setColumnCount()` και `setRowCount()` επιλέγει τις στήλες και γραμμές που θα έχει ο

συγκεκριμένος πίνακας, και μέσω ενός loop και της εντολής `setItem()` προσθέτει τις πληροφορίες μία προς μία σε αυτόν.

- Με αντίστοιχο τρόπο λειτουργεί και η συμπλήρωση του πίνακα των τακτικών. Αρχικά βρίσκει τον φάκελο που περιέχει μέσα τα αρχεία με τις τακτικές, και στην συνέχεια βρίσκει τα αρχεία με τις διαφορετικές τακτικές εάν υπάρχουν περισσότερα από ένα. Επιλέγει το μέγεθος του πίνακα σε γραμμές και στήλες, επιλέγει τον τίτλο που θα έχει η κάθε στήλη, και τοποθετεί το `id`, το όνομα και τις θέσεις του παίκτη στις αντίστοιχες στήλες. Εάν υπάρχουν περισσότερα από ένα αρχεία, τότε η τοποθέτηση γίνεται από επάνω προς τα κάτω και χωρίζεται η κάθε τακτική με το ανάλογο μήνυμα.
- Ο τελευταίος πίνακας περιέχει τις ενέργειες των παιχτών. Βρίσκει το κατάλληλο αρχείο με όνομα `'actions.csv'` και στην συνέχεια επιλέγει στήλες και γραμμές. Έπειτα τοποθετεί τον ανάλογο τίτλο στις στήλες, και τοποθετεί ένα προς ένα τα δεδομένα με δύο λούπες.

5.15.13 Η μέθοδος `show_diagram()`

Αυτή η μέθοδος εμφανίζει τα διαγράμματα των στατιστικών ανάλογα με το διπλό πάτημα στα κελιά των κατάλληλων πινάκων. Αρχικά μέσω των `sender()` και `objectName()` εντοπίζει ποιος από τους πίνακες έχει πατηθεί.

- Εάν το `id` του πίνακα είναι `'lineup_table'` τότε παίρνει μέσω της εντολής `widgets.lineup_table.currentIndex().row()` την επιλεγμένη γραμμή και μέσω της εντολής `widgets.lineup_table.item(row, 1).text()` την επιλεγμένη στήλη. Έπειτα επιλέγει το κατάλληλο αρχείο με τις ενέργειες του παίκτη, και καλώντας την μέθοδο `all_game_single_player_actions()` εμφανίζει το κατάλληλο γράφημα. Αυτή η μέθοδος καλείται σε περίπτωση που πατηθεί οποιοδήποτε πεδίο στον πίνακα τακτικής των παιχτών.
- Σε περίπτωση που το `id` είναι `'actions_table'` παίρνουμε με τον ίδιο τρόπο όπως και προηγουμένως την γραμμή και την στήλη και καλούνται οι μέθοδοι `all_game_single_player_actions()`, `all_game_action_family()`, `all_game_specific_action()`, `all_game_all_player_actions()` αν η στήλη που έχει επιλεγθεί είναι η 1^η, 2^η, 3^η, ή 4^η αντίστοιχα.

5.15.14 Η μέθοδος `on_download_requested()` και `resizeEvent()`

Η μέθοδος αυτή δέχεται στην είσοδο της ένα αρχείο μέσω του `lineup builder` και αποδέχεται την λήψη μέσω της εντολής `download.accept()`. Ενώ η μέθοδος `resizeEvent()` καλεί την μέθοδο `UIFunctions.resize_grips()` για την αλλαγή μεγέθους του προγράμματος μέσω του κέρσορα, σέρνοντας δηλαδή την άκρη.

Κεφάλαιο 6ο: Συμπεράσματα ή/και προτάσεις βελτίωσης

Όπως είναι λογικό ένα τέτοιο λογισμικό δεν μπορεί να αντικαταστήσει πλήρως εργαλεία που κοστίζουν μεγάλο χρηματικό ποσό και αναπτύσσονται εδώ και πολλά χρόνια. Ωστόσο είναι μία καλή πρώτη λύση για μικρές ομάδες. Παρόλα αυτά ορισμένες βελτιώσεις θα μεγιστοποιήσουν την χρηστικότητα της. Συγκεκριμένα η παράλληλη χρήση μίας δεύτερης κάμερας στην ακριβώς αντίθετη πλευρά του γηπέδου για παράλληλο εντοπισμό των παικτών από πολλαπλές κάμερες θα αποτρέψει το πρόβλημα που αντιμετωπίζεται όταν ένας παίκτης καλύπτεται πίσω από έναν άλλον. Ταυτόχρονα η αθλητική ομοσπονδία επέτρεψε πρόσφατα την χρήση εξοπλισμού όπως π.χ. γιλέκων τα οποία φέρουν επάνω τους σένσορα τύπου `gps`. Η χρήση αυτού του σένσορα θα απογειώσει την χρηστικότητα της εφαρμογής μιας

Κεφάλαιο 6

και θα υπάρχει ακόμα μεγαλύτερη ακρίβεια στον εντοπισμό παικτών και στην απόσταση που έχουν ο ένας από τον άλλο σε πραγματικό χρόνο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία

- [1] Jaganmohan, Gopinath; Loganathan, Venkateshwaran. Series: Community Experience Distilled *PySide GUI Application Development - Second Edition*. Packt Publishing, 2016.
- [2] Petrou, Maria., *Image processing: dealing with texture* Chichester ; Hoboken, NJ : John Wiley & Sons Inc., c2006.

Πατέντες

- [3] Ultralytics. YoloV5 object detection algorithm 2021.

Data Sheet

- [4] Tsung-Yi Lin. Common Objects in Context, *COCO Dataset*. 2006-2022.

Internet Site

- [5] Graphical User Interface (GUI). Available: <https://www.techopedia.com/definition/5435/graphical-user-interface-gui>.
- [6] What Is Pyplot In Matplotlib? Available: <https://www.activestate.com/resources/quick-reads/what-is-pyplot-in-matplotlib/>
- [7] OPEN-CV About. Available: <https://opencv.org/about/>
- [8] Pandas. Available: <https://mode.com/python-tutorial/libraries/pandas/>
- [9] What is Pandas: Available: <https://pandas.pydata.org/about/index.html>
- [10] The ModelView Architecture: Available <https://www.pythonguis.com/tutorials/pyside-modelview-architecture/>
- [11] NumPy in ArcGIS. Available: <https://pro.arcgis.com/de/pro-app/latest/arcpy/get-started/working-with-numpy-in-arcgis.htm>
- [12] PyQt vs Pyside. Available: <https://dev.to/amigosmaker/pyqt-vs-pyside-13b>
- [13] What is Scrum. Available: <https://www.atlassian.com/agile/scrum>
- [14] Scrum: A better way of building products. Available: A Better Way Of Building Products.
- [15] NumPy Introduction: Available: https://www.w3schools.com/python/numpy/numpy_intro.asp
- [16] About NumPy. Available: <https://numpy.org/doc/1.16/about.html>

Paper in Conference Proceedings

[17] Bueno, Murilo José de Oliveira; Silva, Maisa; Cunha, Sergio Augusto; Torres, Ricardo da Silva; Moura, Felipe Arruda. *Multiscale fractal dimension applied to tactical analysis in football: A novel approach to evaluate the shapes of team organization on the pitch*. PLoS ONE. 9/1/2021, Vol. 16 Issue 9, p1-14. 14p. DOI: 10.1371/journal.pone.0256771.

[18] Scott, Atom; Fujii, Keisuke; Onishi. *How does AI play football? An analysis of RL and real-world football strategies*. Masaki. 11/24/2021.

Journal Articles

[19] Eric Hofesmann “How to work with object detection datasets in COCO format” *Towards Data Science*. Feb. 2021.

[20] Vikas Solegaonkar “Introduction to PyTorch” *Towards Data Science*. Sep. 2019.

[21] Candu Kuan “Football players tracking — identifying players’ team based on their jersey colors using OpenCV” *Towards Data Science* Aug. 2020