

Ατομική Διπλωματική Εργασία

**MOTION CAPTURING, BEHAVIORAL ANALYSIS AND
MOTION SYNTHESIS OF CYPRIOT REPTILES**

Μιχαήλ Κωνσταντίνου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Δεκέμβριος 2020

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Motion capturing, behavioral analysis and motion synthesis of Cypriot reptiles

Μιχαήλ Κωνσταντίνου

Επιβλέπων Καθηγητής
Ανδρέας Αριστείδου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Δεκέμβριος 2020

Ευχαριστίες

Είμαι στην ευχάριστη θέση να οφείλω να αναφέρω πολλά ονόματα που με βοήθησαν για την ολοκλήρωση της διπλωματικής μου εργασίας καθώς στον δικό μου πηγαισμό για την Ιθάκη, απέκτησα πολλές «πραγμάτειες» και έμαθα πολλά «απ' τους σπουδασμένους».

Πρώτα από όλα θα ήθελα να κάνω μια ιδιαίτερη αναφορά στον καθηγητή μου «Ανδρέα Αριστείδου» ο οποίος με εμπιστεύθηκε για να φέρω εις πέρας αυτό το έργο και καθ' όλη τη διάρκεια της διπλωματικής μου εργασίας, δείχνοντας μου απόλυτη εμπιστοσύνη όπου έπρεπε και παρέχοντας μου βοήθεια εκεί που υπήρχε η ανάγκη. Από το ίδιο εργαστήριο, το Computer Graphics & Virtual Reality laboratory του Πανεπιστημίου Κύπρου, θα ήθελα να ευχαριστήσω τον Αναστάσιο Γιαννακίδη ο οποίος έχοντας περισσότερη εμπειρία από εμένα στα γραφικά υπολογιστών, στάθηκε δίπλα μου σαν δεύτερος καθοδηγητής βοηθώντας με να μάθω περισσότερα για το έργο, τα γραφικά υπολογιστών και την χρήση ορισμένων αλγορίθμων.

Εκτός από μέλη του Πανεπιστημίου Κύπρου θα ήθελα να ευχαριστήσω τον καθηγητή Δρ. Σάββα Ζώτο για την συνεισφορά του σε κάποια στάδια της διπλωματικής μου εργασίας και την άμεση ανταπόκριση του σε υλικό και πληροφορίες που χρειαστήκαμε για την ανάπτυξη κάποιων εφαρμογών στα πλαίσια αυτής της διπλωματικής εργασίας. Επιπρόσθετα θα ήθελα να ευχαριστήσω την εταιρεία Darkrony, η οποία σε κάθε μου αίτημα που αφορούσε έμμεσα ή άμεσα την διπλωματική μου εργασία, ανταποκρίθηκε θετικά.

Κλείνοντας θα ήθελα όπως ευχαριστήσω τους απόφοιτους φοιτητές Δημήτρη Κάιζερ και Ειρήνη Ποξυλάρη για την έμμεση τους παροχή βοήθειας στη διπλωματική εργασία, δείχνοντας μου σωστές πρακτικές και τεχνολογίες ανάπτυξης εφαρμογών οι οποίες χρησιμοποιήθηκαν σε μικρό ή μεγάλο βαθμό στις εφαρμογές αυτής της διπλωματικής εργασίας.

Περίληψη

Η χρήση των γραφικών και της τεχνικής νοημοσύνης, δεν έχει υλική υπόσταση μόνο στον κλάδο της Πληροφορικής αλλά σε όλες τις πτυχές της επιστήμης γενικότερα. Η ατομική διπλωματική εργασία, περιέχει την διαδικασία που ακολουθήθηκε από μια ομάδα επιστημόνων και φοιτητών του Πανεπιστημίου Κύπρου και όχι μόνο, για την μελέτη και την προσπάθεια ανάπτυξης μιας καινοτόμας τεχνολογίας, με απώτερο σκοπό να βοηθήσει στην μελέτη των ερπετών που ευρίσκονται στην Κύπρο, σε πραγματικό χρόνο. Τελικός στόχος του έργου είναι η δημιουργία ενός συστήματος όπου συναρτήσει ενός αισθητήρα επιτάχυνσης, το σύστημα να αναγνωρίζει και να καταγράφει τις κινήσεις των ερπετών. Πιο λαϊκευμένα, βάσει της ταχύτητας που δίνει ένας αισθητήρας κίνησης (επιταχυντής, ο οποίος βρίσκεται στο ερπετό), το σύστημα να αναγνωρίζει εάν το ερπετό κοιμάται, τρώει, τρέχει και άλλες κινήσεις που είναι ενδιαφέρουσες προς τους βιολόγους και παρατηρητές των ερπετών.

Αρχικά, περιγράφεται η διαδικασία της καταγραφής των κινήσεων των ερπετών χρησιμοποιώντας αισθητήρες και επιταχυντές στα σώματα των ερπετών. Στη συνέχεια αναφέρονται οι προσπάθειες που ακολουθήθηκαν για ψηφιοποίηση των δεδομένων που καταγράψαμε, τον καθαρισμό των δεδομένων μέσω του αλγορίθμου FABRIK και την εφαρμογή τρισδιάστατου μοντέλου. Γίνεται λεπτομερής αναφορά και σε άλλα προβλήματα που υπήρξαν και έπρεπε να διορθωθούν χρησιμοποιώντας αλγορίθμους και άλλα υπολογιστικά μέσα. Στη συνέχεια, περιγράφεται η διαδικασία ανάλυσης των καταγραφών και αποκοπής τους σε ξεχωριστές κινήσεις. Ολοκληρώνοντας την ψηφιακή καταγραφή των ερπετών σε αρχεία τα οποία μπορούν να τύχουν μελέτης και επεξεργασίας, γίνεται μια εκτενής αναφορά στην ανάπτυξη μιας εφαρμογής που σκοπό έχει την απεικόνιση σε πραγματικό χρόνο, των μοντέλων που εξετάζονται από το πρόγραμμα Daily Diary Multi Trace. Συνεχίζοντας την ιδέα της απεικόνισης μοντέλων, περιγράφεται η διαδικασία ανάπτυξης μιας ιστοσελίδας με σκοπό την κοινοποίηση του έργου με δυνατότητες παρουσίασης των μοντέλων που δημιουργήθηκαν. Κλείνοντας, περιγράφεται το πλάνο για το τελευταίο στάδιο που υπάρχει μετά το τέλος αυτής της διπλωματικής εργασίας, που είναι η εφαρμογή αλγορίθμων της μηχανικής μάθησης και γενικότερα της τεχνητής νοημοσύνης για την συσχέτιση της επιτάχυνσης ενός ερπετού και της κίνησης που διαπράττει

Περιεχόμενα

Κεφάλαιο 1 - Εισαγωγή	1
1.1 Σκοπός.....	1
1.2 Top-down approach.....	2
1.3 Δομή διπλωματικής εργασίας	4
1.4 Ανασκόπηση βιβλιογραφίας	5
Κεφάλαιο 2 – Καταγραφή και ετοιμασία δεδομένων	8
2.1 Καταγραφή δεδομένων στο εργαστήριο.....	8
2.2 Εφαρμογή ετικετών στους δείκτες (labeling)	9
Κεφάλαιο 3 – Επεξεργασία δεδομένων	12
3.1 Διόρθωση δεδομένων με εφαρμογή αλγορίθμου FABRIK	12
3.2 Σύγκριση αποτελεσμάτων μετά την υλοποίηση του αλγορίθμου.....	16
3.3 Διόρθωση δεδομένων στο MotionBuilder	17
3.4 Παραλλαγή σάυρας.....	21
3.5 Προβλήματα και διορθώσεις	21
3.6 Snake animation using mocap data.....	22
3.7 Διόρθωση κεφαλής και ουράς στο φίδι	28
Κεφάλαιο 4 – Ανάπτυξη εφαρμογής DDMT-Viewer	30
4.1 Πρόβλημα, προεργασία και καθορισμός των απαιτήσεων	30
4.2 Σχεδιασμός συστήματος	31
4.3 Υλοποίηση, έλεγχος και αναθεώρηση	34
4.4 Έλεγχος, αναθεώρηση και optimizations	40
4.5 Τελικό αποτέλεσμα, σφάλματα και συμπεράσματα	43
4.6 Μελλοντικά σχέδια	44
Κεφάλαιο 5 – Ανάλυση των δεδομένων και εξαγωγή κινήσεων.....	46
5.1 Διαχωρισμός δεδομένων από βιολόγο	46
5.2 Συσχέτιση χρόνων και frames.....	47
5.3 Αποκοπή κινήσεων στο Motion Builder.....	48

Κεφάλαιο 6 – Δημιουργία ιστοσελίδας	51
6.1 Ανασκόπηση	51
6.2 Εργαλεία ανάπτυξης ιστοσελίδας.....	51
6.2 Παρουσίαση Ιστοσελίδας	54
Κεφάλαιο 7 - Συμπεράσματα	62
7.1 Γενικά.....	62
7.2 Καταγραφή κινήσεων χρησιμοποιώντας αισθητήρες.....	62
7.3 Διόρθωση κινήσεων χρησιμοποιώντας ψηφιακή επεξεργασία.....	63
7.4 Απεικόνιση και χρήση κινήσεων	64
7.5 Μελλοντική επέκταση	65
Βιβλιογραφία	67
Κατάλογος εικόνων και γραφικών παραστάσεων.....	69
Παράρτημα Α – Κινήσεις φιδιού.....	A-1
Παράρτημα Β – Κινήσεις σάυρας.....	B-1
Παράρτημα Γ – Κώδικας και αρχεία κινήσεων.....	Γ-1

Κεφάλαιο 1

Εισαγωγή

1.1 Σκοπός	1
1.2 Top-down approach	2
1.3 Δομή διπλωματική εργασίας	4
1.4 Ανασκόπηση βιβλιογραφίας	5

1.1 Σκοπός

Από την αρχή της ανθρωπότητας, είχαμε την περιέργεια να μάθουμε ολοένα και περισσότερα για τον κόσμο που μας περιβάλλει. Είτε αυτό είναι κάποιο είδος φυτού, κάποιο είδος ζώου, κάποιο είδος ζωντανού οργανισμού ή απλά ο άνθρωπος και η ιστορία που τον διακατέχει. Ήδη από τους πρώτους πολιτισμούς, βλέπουμε πως οι αρχαίοι Έλληνες, δεν ήταν ικανοποιημένοι με μια απλή παρατήρηση ή απλά την γνώση, αλλά κυνηγούσαν κάτι περισσότερο: τη γνώση που ήταν εξακριβωμένη και τεκμηριωμένη. Σήμερα μας είναι εύκολο να ξεχωρίσουμε πως κάτι τέτοιο αποτελεί τον ορισμό της λέξης “επιστήμη”. Η πρώτη αναφορά της λέξης “επιστήμη” έγινε από τον Πλάτωνα, στο έργο του “*Θεαίτητος*” όπου και χαρακτηριστικά αναφέρει μεταξύ άλλων πως «*έστιν ουν επιστήμη δόξα αληθής μετά λόγου*», δηλαδή πως η επιστήμη είναι βεβαιωμένη με λογικά επιχειρήματα. Εκ τότε ο ορισμός της λέξης “επιστήμης” έχει αλλάξει και διάφορες μεθοδολογίες πλέον είναι βασισμένες στις μεσαιωνικές και όχι στις αρχαιοελληνικές. Παρόλα αυτά, συμφωνούν σε ένα: Πως η γνώση πρέπει να είναι βεβαιωμένη.

Μερικές χιλιετίες μετά, βλέπουμε πως ο άνθρωπος προσπαθεί ολοένα και περισσότερο να γνωρίσει καλύτερα τον κόσμο που τον περιβάλλει, να κάνει περισσότερες παρατηρήσεις σε βαθμό που το ανθρώπινο μάτι είναι αδύναμο και μετά από όλα αυτά να εξάγει τα κατάλληλα συμπεράσματα και να τεκμηριώσει όσα δεν μπόρεσε ποτέ να κατανοήσει ο ανθρώπινος νους. Λόγω των πιο πάνω, εύκολα μπορεί να αντιληφθεί κανείς την ανάγκη ανάπτυξης των κατάλληλων μέσων παρατήρησης και επεξεργασίας δεδομένων. Στις μέρες, ένα από τα βασικότερα μέσα επεξεργασίας δεδομένων, είναι ο ηλεκτρονικός υπολογιστής.

Ο στόχος της διπλωματικής αυτής εργασίας, ήταν να βοηθήσει στην καταγραφή των ερπετών της Κύπρου (των κινήσεων τους) για σκοπούς ανάλυσης της συμπεριφοράς τους με τη χρήση της τεχνολογίας. Έτσι, οι ερπετολόγοι να μπορούν να συλλέξουν περισσότερα δεδομένα (παρατηρήσεις) και να μάθουν περισσότερα για τη ζωή των ερπετών της Κύπρου. Ενώ υπάρχουν μερικά συστήματα εκεί έξω για την μελέτη των ερπετών, εντούτοις είτε παρουσιάζουν ελλείψεις όπως θα δούμε και στη συνέχεια, είτε δεν προσφέρουν τις λειτουργίες που επιθυμεί κάποιος ερπετολόγος. Κι αυτό γιατί δεν υπάρχει μια γνωστή μέθοδος για το πως να υλοποιηθεί κάποιο σύστημα το οποίο να μπορεί σε πραγματικό χρόνο να συλλέγει δεδομένα από ζωντανά ερπετά και να αναγνωρίζει ως επίσης την κίνηση που διαπράττουν. Αν μπούμε και σε μεγαλύτερο βάθος θα δούμε πως είναι ακόμη πιο δύσκολη η απεικόνιση των ερπετών σε συνδυασμό με την ψηφιακή τους ανάλυση και την εξαγωγή αποτελεσμάτων. Επιπρόσθετα, για να πετύχουμε το στόχο μας χρειαζόμαστε τεχνολογίες οι οποίες τα τελευταία 5 χρόνια έφτασαν σε ένα επίπεδο όπου μπορούν να μας φανούν χρήσιμες.

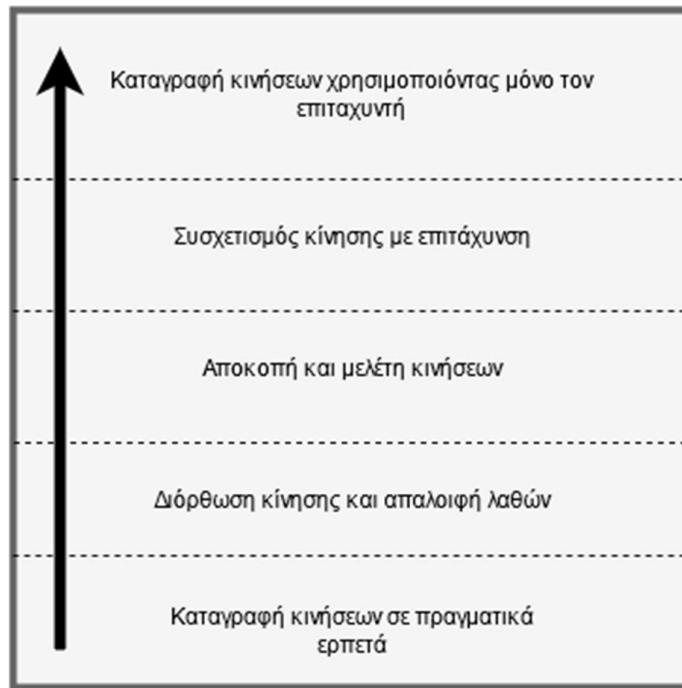
Η απουσία αυτής της τεχνολογικής χείρας βοηθείας προς τους ερπετολόγους ήταν το κίνητρο μας για την μελέτη, τη σχεδίαση και την ανάπτυξη ενός συστήματος που να μπορεί να δώσει λύση στα πιο πάνω προβλήματα. Δεν υπήρξε κάποιο σύστημα το οποίο να μπορεί σε πραγματικό χρόνο να παρακολουθεί κάποιο ερπετό και παράλληλα, να μπορεί να εξακριβώσει το είδος κίνησης το οποίο εκτελεί. Σε συνδυασμό με τον **βαθμό δυσκολίας** του έργου, η πρόκληση μετατράπηκε σε κίνητρο και το κίνητρο έγινε το έναυσμα για την επιλογή αυτού του θέματος ως διπλωματική εργασία.

1.2 Top-down approach

Επί της ουσίας, το τελικό αποτέλεσμα είναι η αναγνώριση της κίνησης χρησιμοποιώντας ένα επιταχυντή. **Επι του πρακτέου** όμως για να πετύχουμε αυτό το αποτέλεσμα έπρεπε να υπάρξει ένας συνδυασμός της τεχνητής νοημοσύνης, της επεξεργασίας γραφικών υπολογιστών καθώς και της καταγραφής των ερπετών.¹

Αρχίζοντας λοιπόν από τον τελικό στόχο (top level), πηγαίνοντας προς τα κάτω, μπορούμε να δούμε στην Εικόνα 1, τι χρειάστηκε να υλοποιηθεί σε κάθε επίπεδο για να φτάσουμε στον τελικό στόχο.

¹ Αν υπολογίσουμε ως επίσης τις τεχνικές απεικόνισης (visualization techniques) που χρειάστηκε να υλοποιήσουμε, εμπλέκεται και η τεχνολογία λογισμικού καθώς και οι τεχνολογίες διαδικτύου στο ευρύ φάσμα των πτυχών της πληροφορικής που καλύπτει η διπλωματική εργασία.



Εικόνα 1 Project, Top-down approach

Για να επιτευχθεί ο τελικός στόχος², δηλαδή η καταγραφή/αναγνώριση κινήσεων, θα πρέπει σε κάθε ερπετό να υπάρχει ένας επιταχυντής (accelerometer), ή αλλιώς αισθητήρας επιτάχυνσης/ταχύτητας ο οποίος να καταγράφει την επιτάχυνση του ερπετού. Στη συνέχεια, χρησιμοποιώντας αυτές της πληροφορίες θα πρέπει το σύστημα να είναι σε θέση να αναγνωρίσει την κίνηση του ερπετού.

Για να γίνει αυτό όμως, πρέπει να πάμε ένα σκαλί πιο κάτω όπου θα πρέπει να αναπτυχθούν ή να χρησιμοποιηθούν οι σωστοί αλγόριθμοι οι οποίοι θα συσχετίζουν την κίνηση του ερπετού με την επιτάχυνση του. Για να επιτευχθεί αυτό όμως θα πρέπει όπως υπάρχουν οι κινήσεις εκ των προταίρων. Επομένως προτού φθάσουμε σε αυτό το επίπεδο θα πρέπει όπως δημιουργηθούν οι κινήσεις.

Η δημιουργία των κινήσεων είναι μια διαδικασία η οποία περιλαμβάνει πραγματική καταγραφή των ερπετών στο εργαστήριο καθώς και την μετέπειτα διόρθωση των οποιονδήποτε λαθών προκύψουν κατά την διάρκεια της καταγραφής κάνοντας χρήση των κατάλληλων αλγορίθμων. Έτσι, φθάνουμε στην αρχή του έργου αυτού αλλά και της

² Εμπνευσμένος από το βιβλίο του Kross Ross, Computer Networking – A top down approach, βρίσκω πιο εύκολο να επεξηγηθεί πρώτα ποιος είναι ο απώτερος σκοπός, και στη συνέχεια όλα τα βήματα τα οποία είναι διαφανή προς τον «πελάτη». Εννοείται όμως πως εμείς πρώτα ξεκινήσαμε από κάτω προς τα πάνω. Αξίζει επίσης να αναφερθεί πως στην πορεία χρειάστηκαν να υλοποιηθούν κάποια άλλα βήματα παράλληλα (π.χ. απεικόνιση).

διπλωματικής εργασίας, η οποία ξεκινάει με την καταγραφή των κινήσεων στο εργαστήριο Computer Graphics & Virtual Reality laboratory του Πανεπιστημίου Κύπρου.

1.3 Δομή διπλωματικής εργασίας

Η δομή της διπλωματικής εργασίας ακολουθεί πιστά τα βήματα τα οποία αναφέρθηκαν προηγουμένως, αρχίζοντας από την κατώτατη βαθμίδα, πηγαίνοντας προς τα πάνω. Τα επόμενα κεφάλαια αφορούν κυρίως την μεθοδολογία ενώ στο τελευταίο κεφάλαιο γίνεται μια σύνοψη και αναφορά στην μελλοντική επέκταση του έργου. Αφ' υψηλού η μεθοδολογία ακολουθεί τα βήματα **data acquisition, labelling, cleaning, animation and visualization** και τέλος η δημιουργία ιστοσελίδας για την **παρουσίαση** του έργου.

Συνεπώς, το επόμενο κεφάλαιο αρχίζει με μια μικρή προεργασία που έγινε πριν να μπω στο έργο και αφορά την καταγραφή των ερπετών τοποθετώντας δείκτες στα σώματα τους. Στην συνέχεια γίνεται μια αναφορά στην πρώτη επεξεργασία που υπέστησαν τα αρχεία, χρησιμοποιώντας τεχνικές όπως “*labelling*” και εξαγωγής τους σε αρχεία τύπου C3D.

Στο τρίτο κεφάλαιο γίνεται μια εκτενής αναφορά στην επεξεργασία των αρχείων C3D, την εξαγωγή τους σε FBX και την μετέπειτα επεξεργασία τους στο Motion Builder στη νέα τους μορφή. Σε αυτό το κεφάλαιο δίνεται η εφαρμογή του αλγορίθμου *FABRIK*, η προσαρμογή του αλγορίθμου στα δικά μας δεδομένα και στην διόρθωση καινούργιων προβλημάτων που παρουσιάστηκαν, ο τρόπος με τον οποίο τα αρχεία C3D πήραν σάρκα και οστά (*rigging and skinning*), και με τον τρόπο με τον οποίο υπήρξε η διόρθωση των όποιων προβλημάτων παρουσιάστηκαν σε αυτή την διαδικασία. Κλείνοντας το κεφάλαιο, φτάνουμε στο σημείο όπου υπάρχουν δεδομένα έτοιμα για την πρώτη τους ανάλυση.

Στο τέταρτο κεφάλαιο περιγράφεται η διαδικασία της απεικόνισης πλέον (*visualization*). Δίνεται έμφαση στην πρώτη εφαρμογή η οποία αναπτύχθηκε επί της διπλωματικής αυτής εργασίας η οποία είναι το πρόγραμμα DDMT-Viewer. Παρουσιάζονται οι ελλείψεις υφιστάμενων εφαρμογών που υπάρχουν όπως το DDMT, και η προσπάθεια μας για επέκταση των υπηρεσιών που προσφέρονται και αφορούν το *visualization*. Με την βοήθεια αυτών των μέσων, έγινε και η πρώτη τους ανάλυση. Η πρώτη τους ανάλυση η οποία έγινε από τον Δρ. Σάββα Ζώτο αναφέρεται στο τέταρτο κεφάλαιο όπου και δίνεται έμφαση στον τρόπο με τον οποίο έγινε η κατάτμηση κινήσεων στο πρόγραμμα Motion Builder. Συνεχίζοντας στο έκτο και τελευταίο κεφάλαιο της μεθοδολογίας γίνεται μια αναφορά στον τρόπο με τον οποίο αναπτύξαμε την δεύτερη μας εφαρμογή, που είναι η ιστοσελίδα

παρουσίασης του έργου. Συνοψίζοντας, στο τελευταίο κεφάλαιο(κεφ. 7) δίνονται τα συμπεράσματα και η μελλοντική επέκταση του έργου.

1.4 Ανασκόπηση βιβλιογραφίας

Στο παρελθόν πολλοί ερευνητές κατόρθωσαν να μελετήσουν την κίνηση του ανθρώπινου σώματος, του σώματος των ζώων και την κίνηση των αντικειμένων. Οι Llyr ap Cenydd και Bill Teahan ανέπτυξαν ένα σύστημα για την δυναμική κίνηση, την κινητική συμπεριφορά των αρθροπόδων σε πραγματικό χρόνο διευκολύνοντας τη ρεαλιστική και αυτόνομη διέλευση σε ένα αυθαίρετο περιβάλλον. Συνδυάζοντας ένα μοντέλο συμπεριφοράς αποκεντρωμένο(decentralized reactive behavioral model) με μια υβριδική προσέγγιση στην κίνηση που εφαρμόζει τα συγκριτικά πλεονεκτήματα της φυσικής προσομοίωσης και του κινηματικού ελέγχου, το σύστημά έχει την δυνατότητα να δημιουργεί αυτόματα πολύπλοκη οργανική κίνηση πέρα από ένα μεγάλο εύρος από χαρακτηριστικά της επιφάνειας ανεξάρτητα από τη δομική πολυπλοκότητα. Η προσομοίωση μπορεί να παράγει φυσική εμφάνιση της κίνησης των αρθροπόδων που κινούνται σε αυθαίρετα περιβάλλοντα χωρίς να λαμβάνουν υπόψιν τη δομική πολυπλοκότητα. Η εξερεύνηση και η πλοήγηση του περιβάλλοντος είναι απολύτως αυτόνομη, χωρίς προεπεξεργασία ή ρητή επισήμανση συγκεκριμένων χαρακτηριστικά περιβάλλοντος [1].

Οι Sangchul Hwang και Young Choi ανέπτυξαν ένα σύστημα εκτίμησης συνεχής στάσης μικρών αρθρωτών ζώων χρησιμοποιώντας ένα τρισδιάστατο μοντέλο σκελετού του ζώου από μια ακολουθία βίντεο πολλαπλών προβολών. Περιλαμβάνει την εξαγωγή των άκρων και των ριζών κάθε ποδιού, καθώς και τον υπολογισμό της κοινής κινηματικής που χρησιμοποιεί τον αλγόριθμο FABRIK με τα εξαγόμενα άκρα και ρίζες. Αυτή είναι η πρώτη μελέτη που παρουσίασε την σημασία της εκτίμησης της κίνησης των μικρών αρθρωτών ζώων χρησιμοποιώντας πολλές εικόνες προβολής. [2] Επίσης, έχει παρατηρηθεί πως τα τελευταία χρόνια έχει αυξηθεί η τάση της χρήσης του επιταχυντή στα ζώα ούτως ώστε να μπορεί να γίνεται η παρακολούθηση της κίνησης τους. Στόχος είναι, η ανάλυση των δεδομένων που αντλούνται από την παρακολούθηση των ζώων να βοηθήσει στο σχεδιασμό του “conservation”, πρακτικές κτηνοτροφίας, στην μέτρηση υγείας των ζώων και την ανίχνευση ασθενειών. Οι επιταχυντές θεωρούνται αξιόπιστα εργαλεία για τον λόγο ότι έχουν την δυνατότητα να ανιχνεύουν την θέση του ζώου, την κίνηση του σώματος και παραμέτρους στάσης. Η ανάλυση αυτών των παραμέτρων επιτρέπει τον προσδιορισμό και ποσότητα συγκεκριμένων δραστηριοτήτων σε ζώα. Η αναγνώριση της δραστηριότητας των ζώων μέσω ροών δεδομένων επιταχυντή παίζει θεμελιώδη ρόλο στην ανακάλυψη της σχέσης μεταξύ της

κίνησης των ζώων, της ενέργειας που δαπανάτε και της συμπεριφοράς του ζώου (Halsey et al., 2011; Nathan et al., 2012) .] Taesoo Kwon, Hoimin Kim, και Yoonsang Lee έφτιαξαν ένα σύστημα για την προσομοίωση της κίνησης του ιγούανα γρηγορότερα από τον πραγματικό χρόνο αν και το ιγούανα θεωρείται πως έχει μαλακό σώμα. Χρησιμοποίησαν ένα μικρό αριθμό κορυφές στο σώμα του ιγούανα και αγνοώντας τον σκελετο πέτυχα την γρήγορη προσομοίωση [3].

Ερευνητές έχουν μελετήσει επιπροσθέτως, ελεγκτές για φυσικά προσομοιωμένους χαρακτήρες ούτως ώστε να υπάρχει φυσικός ρεαλισμός στις ενεργές κινήσεις των ζώων. Τέτοιες μελέτες έχουν επικεντρωθεί κυρίως στον έλεγχο δίποδων ή τετράποδων θηλαστικών ζώων όπως άνθρωποι ή σκύλοι. [4] [5] [6] [7] Ένα κοινό χαρακτηριστικό αυτών των θηλαστικών ζώων είναι η σχετικά υψηλή θέση σε σχέση με το κέντρο της μάζας στην περιοχή στήριξης των ποδιών. [8] Αυτό κάνει τη διατήρηση ισορροπία μιας από τις πιο σημαντικές προκλήσεις που πρέπει να ξεπεραστούν. Άλλα είδη ζώων όπως τα ερπετά μπορεί να παρουσιάζουν πολύ διαφορετικό τύπο κίνησης. Για παράδειγμα, η κίνηση της σαύρας είναι μάλλον διαφορετική από την κίνηση των θηλαστικών αφού λυγίζει τους κορμούς της πλευρικά σε κάθε βήμα [8] [9]. Ένα άλλο χαρακτηριστικό των σαυρών σχετικά με την κίνησή τους είναι ότι έχουν σχετικά χαμηλό κέντρο μάζας, το οποίο μειώνει τη σημασία της διατήρησης της ισορροπίας στον έλεγχο. Μερικοί ερευνητές έχουν μελετήσει τον έλεγχο δίποδων ζώων χρησιμοποιώντας μυοσκελετικά συστήματα [11].

Οι περισσότερες από αυτές τις μελέτες υποθέτουν ότι το προσομοιωμένο μοντέλο αποτελείται από ένα σύνολο από άκαμπτα σώματα και ενεργοποιείται από τις ροπές των αρθρώσεων. Ως εκ τούτου, δεν είναι σαφές εάν αυτές οι μέθοδοι μπορούν να χρησιμοποιηθούν για τη δημιουργία των ευέλικτων κινήσεων των σαυρών, που περιλαμβάνουν παραμόρφωση του σώματος λόγω εξωτερικών δυνάμεων από την επαφή με το έδαφος. Ο έλεγχος της κίνησης των αντικειμένων μαλακού σώματος ήταν μελετήθηκε για να κάνει τα ζώα να κινηθούν όπως επιθυμεί ο χρήστης. Ο Tan et al, πρότεινε ένα σύστημα ελέγχου για μαλακά σώματα χρησιμοποιώντας την έννοια της μυϊκής ίνας και FEM χωρίς τη χρήση σκελετού [12]. Ο Coros et al. [13] παρουσίασε μια μέθοδο για τον έλεγχο των κινήσεων των ενεργών παραμορφώσιμων χαρακτήρων από δυναμικά προσαρμόζοντας σχήματα ανάπαυσης. Ο Kim και ο Pollard [14] πρότειναν ένα γρήγορο φυσικό σύστημα προσομοίωσης για παραμορφώσιμους χαρακτήρες αμαξώματος με σκελετό. Ο Liu et al. [15] παρουσίασε μια μέθοδο προσομοίωσης και ελέγχου μαλακού σώματος με σκελετό συνδέοντας τον σκελετό και την μαλακή δυναμική αμαξώματος.

Η δική μας έρευνα είναι παρόμοια με την παραπάνω, καθώς ελέγχει ένα μαλακό σώμα, την σαύρα. Παρόλα αυτά, είναι ένα διαφορετικό είδος από αυτό το οποίο θέλουμε να μελετήσουμε και ως επίσης δεν υπήρξε κάτι σχετικό για το φίδι. Και από αυτή την μελέτη, αλλά και ευρύτερα από όλες όσες αναφέρθηκαν, ξεχωρίζει η μακριά και δύσκολη διαδικασία που απαιτείται για την καταγραφή μερικών κινήσεων σε μικρά ερπετά. Ενώ έχουμε βρει πολλές βάσεις δεδομένων για κινήσεις με άνθρωπους, εντούτοις δεν έχουμε βρει κάτι αντίστοιχο το οποίο να περιλαμβάνει κινήσεις από τα είδη των ερπετών που θέλουμε να μελετήσουμε. Η μελέτη, του Taesoo Kwon, Hoimin Kim, και Yoonsang Lee σχετικά με το ιγκουάνα, είναι πολύ κοντά σε αυτό το οποίο θέλουμε να πετύχουμε. Χρησιμοποιεί δείκτες και ένα επιταχυντή για την καταγραφή των κινήσεων και της ταχύτητας/επιτάχυνσης. Παρόλα αυτά, δεν υπάρχει σχετική έρευνα όσο αφορά τον συσχετισμό που έχουν τα δύο αυτά μεγέθη ούτε υπήρξε το επόμενο στάδιο το οποίο χρησιμοποιεί όλα αυτά σε συνδυασμό με τα νευρωνικά δίκτυα. Είναι ένα δύσκολο εγχείρημα αυτό και γι' αυτό και δεν υπάρχουν άλλωστε τόσα δεδομένα όσο υπάρχουν για τις ανθρώπινες κινήσεις.

Κεφάλαιο 2

Καταγραφή και ετοιμασία δεδομένων

2.1 Καταγραφή δεδομένων στο εργαστήριο	8
2.2 Εφαρμογή ετικετών στους δείκτες (labeling)	9

2.1 Καταγραφή δεδομένων στο εργαστήριο

Στο εργαστήριο Computer Graphics & Virtual Reality Laboratory του τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου, δημιουργήθηκε ένας ειδικά μορφωμένος χώρος για την καταγραφή κινήσεων μιας σαύρας και ενός φιδιού, κυπριακής προέλευσης. Σε πρώτο στάδιο τοποθετήθηκαν αισθητήρες στα διάφορα μέρη του σώματος της σαύρας και του φιδιού. Ως επίσης τοποθετήθηκε ένας ανιχνευτής επιτάχυνσης (accelerometer) κοντά στο πίσω μέρος του λαιμού, λίγο πιο κάτω από το κεφάλι. Στη συνέχεια η σαύρα και το φίδι κινήθηκαν ελεύθερα στον ειδικά διαμορφωμένο χώρο πάντα υπό την επίβλεψη του καθηγητή βιολογίας Δρ. Σάββα Ζώτου και του καθηγητή του τμήματος Πληροφορικής Δρ. Ανδρέα Αριστείδου.

Η ελεύθερη κίνηση των ερπετών είχε στόχο την καταγραφή κίνησης τους κάτω από διάφορες συνθήκες όπως κίνδυνος, εντοπισμός κάποιου θηράματος, κατάποση τροφών και φυσικά οποιαδήποτε περιφορά στο χώρο. Για την επίτευξη αυτού, σε διάφορα σημεία του χώρου τοποθετηθήκαν εμπόδια και στοιχεία του περιβάλλοντος των ερπετών (π.χ. κλαδί).



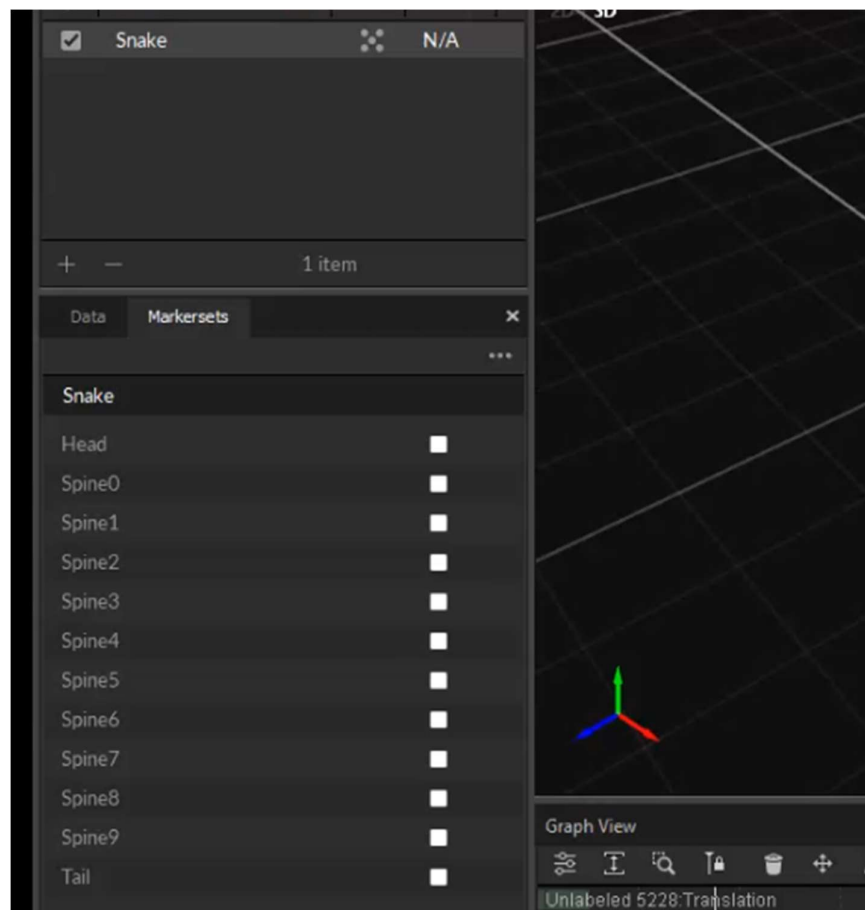
Εικόνα 2 Data acquisition

Τα αποτελέσματα από τους αισθητήρες καταγράφηκαν σε αρχεία τύπου .tak ενώ ολόκληρη η διαδικασία είναι βιντεοσκοπημένη από 3 διαφορετικές οπτικές γωνίες. Με την χρήση του προγράμματος *Motive* έγινε η επεξεργασία των αποτελεσμάτων από τους αισθητήρες.

2.2 Εφαρμογή ετικετών στους δείκτες (labeling)

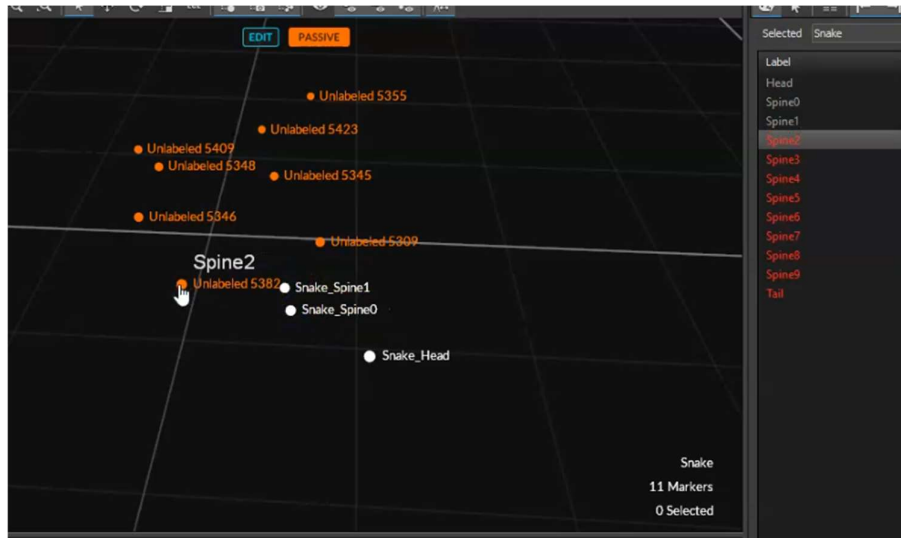
Μέσω του προγράμματος Motive μας δίνεται η δυνατότητα να ανοίξουμε τα αρχείου τύπου .tak [1], να δημιουργήσουμε και να προσθέσουμε ετικέτες στους δείκτες που καταγράψαμε. Με αυτό τον τρόπο ονομάζουμε τους αισθητήρες ψηφιακά και έτσι επιτυγχάνουμε να έχουμε μεγαλύτερη ανάγνωση των αρχείων τα οποία θα βοηθήσουν στην μετέπειτα επεξεργασία τους.

Αρχικά, δημιουργήσαμε ένα καινούργιο Markerset για κάθε ερπετό. Σε κάθε markerset προσθέσαμε μία ετικέτα για κάθε δείκτη που καταγράψαμε. Ένας δείκτης φυλάει τις τιμές ενός αισθητήρα. Μερικές ονομασίες που χρησιμοποιήθηκαν ήταν “head”, “right front leg”, “spine2”, “tail”, κ.α.



Εικόνα 3 Δημιουργία Markeset και ετικετών

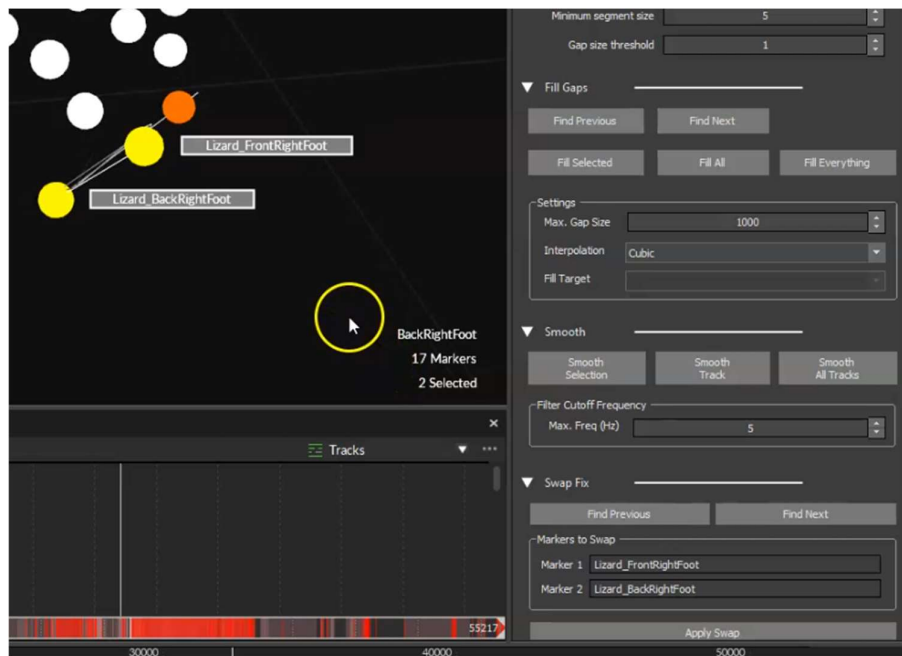
Στη συνέχεια για κάθε frame του αρχείου, επιλέγαμε ένα-ένα τους δείκτες και αντιστοιχούσαμε σε αυτούς μία από τις ετικέτες που δημιουργήσαμε. Το πρόγραμμα Motive λειτουργεί με κάποιο έξυπνο αλγόριθμο στο παρασκήνιο ο οποίος σε μερικές περιπτώσεις μπορεί να καταλάβει με βάση προηγούμενες τιμές, ποια ετικέτα αντιστοιχεί σε δείκτες που δεν περιέχουν ονομασία. Παρόλα αυτά σε ορισμένες περιπτώσεις το πρόγραμμα έκανε λάθη και απέδιδε αντίθετες ονομασίες στους δείκτες. Σε αυτή την περίπτωση έπρεπε να εφαρμόσουμε την ακόλουθη μέθοδο που ονομάζουμε “swap”



Εικόνα 4 Ονομασία δεικτών (labeling)

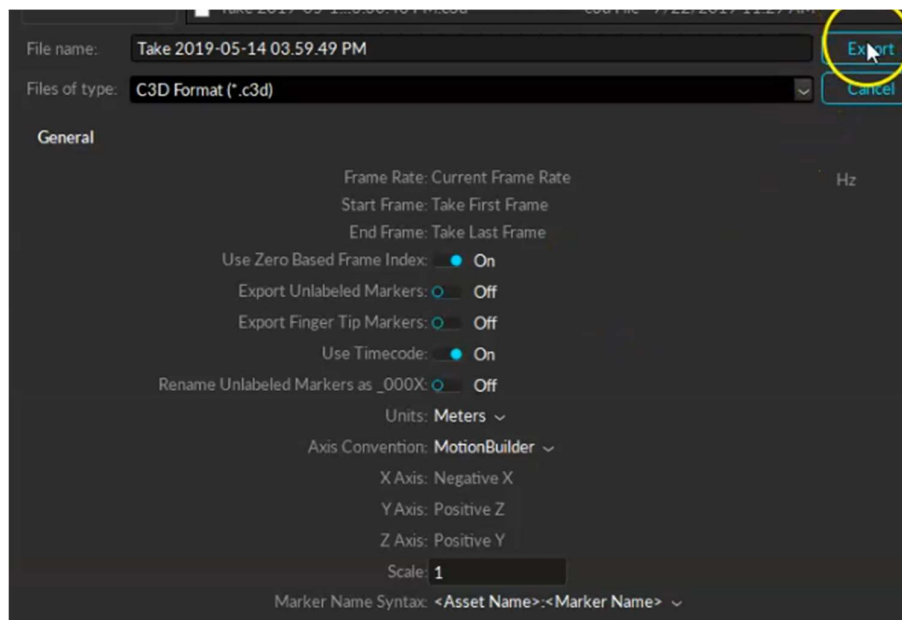
“Swap” ή “Ανταλλαγή ονομασιών” αν θα το ονομάζαμε στα ελληνικά, είναι η διαδικασία κατά την οποία διορθώνουμε κάποιο λάθος του αλγορίθμου του Motive, όπου 2 δείκτες πήραν τις αντίθετες ονομασίες. Κάτι τέτοιο συνήθως συμβαίνει όταν υπάρχει μετωπική σύγκρουση μεταξύ 2 δεικτών.

Όπως φαίνεται και στην πιο κάτω εικόνα, για να διορθώσουμε κάτι τέτοιο, επιλέγουμε τους δύο δείκτες και στην συνέχεια ανοίγουμε το παράθυρο με τα “Edit tools”. Κάτω υπάρχει η δυνατότητα για “Swap Fix” όπου πατώντας απλά το κουμπί “Apply swap”, το πρόγραμμα ανταλλάζει τις ονομασίες.



Εικόνα 5 Swap in Motive

Τελειώνοντας εξαγάγαμε τα αποτελέσματα μας σε C3D αρχεία. Στην πιο κάτω εικόνα απεικονίζονται οι ρυθμίσεις που χρησιμοποιούσαμε. Στις πλείστες περιπτώσεις επιλέξαμε τις προκαθορισμένες τιμές αλλά όσο αφορά την ιδιότητα “Export unlabeled markers” επιλέξαμε “off”. Οι δείκτες οι οποίοι δεν περιέχουν ετικέτα είναι θόρυβος (noise) γι’ αυτό και για να μην εξαγάγουμε θόρυβο στα αποτελέσματα μας αποφασίσαμε όπως απενεργοποιήσουμε αυτή την ιδιότητα.



Εικόνα 6 Εξαγωγή δεδομένων σε C3D file

Κεφάλαιο 3

Επεξεργασία δεδομένων

3.1 Διόρθωση δεδομένων με εφαρμογή αλγορίθμου FABRIK	12
3.2 Σύγκριση αποτελεσμάτων μετά την εφαρμογή του αλγορίθμου	16
3.3 Διόρθωση δεδομένων στο MotionBuilder	17
3.4 Παραλλαγή σαύρας	19
3.5 Προβλήματα και διορθώσεις	19
3.6 Snake animation using mocap data	20
3.7 Διόρθωση κεφαλής	28

Τα δεδομένα τα οποία πάρθηκαν στην προηγούμενη φάση, περιέχουν κάποια σφάλματα και κάποια κενά. Αυτό οφείλεται στο γεγονός ότι τα δεδομένα πάρθηκαν σε ένα φυσικό περιβάλλον, σε πραγματικά ζώα κάτι το οποίο δεν δίνει πάντα τις ιδανικές συνθήκες καταγραφής. Επιπρόσθετα, υπάρχουν πολλά εμπόδια τα οποία δεν μπορούσαν να προβλεφθούν. Για παράδειγμα, η σαύρα να κρύβεται κάτω από το δέντρο και η κάμερα να μην μπορεί να καταγράψει την σωστή τιμή του δείκτη. Ως επίσης, ένα άλλο πρόβλημα το οποίο υπάρχει, οφείλεται στην ανατομία των ζώων καθώς οι δείκτες δεν μπορούσαν να τοποθετηθούν σε κάποια κύρια σημεία-κλειδιά για εμάς. Ακόμη, οι δείκτες δεν μπορούν να τοποθετηθούν ακριβώς στο κόκκαλο, αλλά έχει ένα αναγκαστικό «κενό» (offset) το οποίο σπάζει κάπως την καμπυλότητα. Γι' αυτό τον λόγο, διορθώσεις έπρεπε να γίνουν και σε επίπεδο προγραμματιστικό σε Python αλλά και σε αφ' υψηλού επίπεδο στο Motion Builder [1].

3.1 Διόρθωση δεδομένων με εφαρμογή αλγορίθμου FABRIK

Ο αλγόριθμος Forward And Backwards Reaching Inverse Kinematics [3] είναι ένας εύκολος, γρήγορος και αποδοτικός αλγόριθμος που υλοποιήσαμε για να διορθώσουμε τυχόν λανθασμένες τιμές ή ελλειπές τιμές που προέκυψαν από τα προηγούμενα στάδια της καταγραφής των δεδομένων. Πιο γενικά, “Inverse Kinematics” ορίζεται ως το πρόβλημα κατά το οποίο πρέπει να βρεθούν οι κατάλληλες κινήσεις των joints έτσι ώστε να μετακινηθούν ομαλά, γρήγορα και με ακρίβεια στην σωστή θέση. Ο αλγόριθμος FABRIK αποτελεί μία σύγχρονη και αποδοτική λύση στο συγκεκριμένο πρόβλημα συνδυάζοντας όμως και την απλότητα που επιφέρει εύκολες υλοποιήσεις σε πρακτικό επίπεδο.

Χρησιμοποιώντας λοιπόν την γλώσσα προγραμματισμού Python και τη βιβλιοθήκη C3DServer όπως δίνεται από την επίσημη ιστοσελίδα c3d.org, και υλοποιώντας τον αλγόριθμο FABRIK, δημιουργήσαμε ένα μικρό πρόγραμμα(script) το οποίο διορθώνει την κίνηση του φιδιού.

Ο τρόπος με τον οποίο λειτουργεί ο αλγόριθμος FABRIK είναι με μια επαναλαμβανόμενη προσπέλεση των joints μπρος και πίσω. Κάθε φορά, έχοντας ως δείκτη αναφοράς ένα αρχικό joint το οποίο ονομάζουμε “κεφαλή” (head) και ένα τελικό joint το οποίο ονομάζουμε ουρά (tail), προσπαθεί να βρει τις τιμές που υπολοίπονται από τα joints που υπάρχουν ανάμεσα στα δύο σημεία αναφοράς.

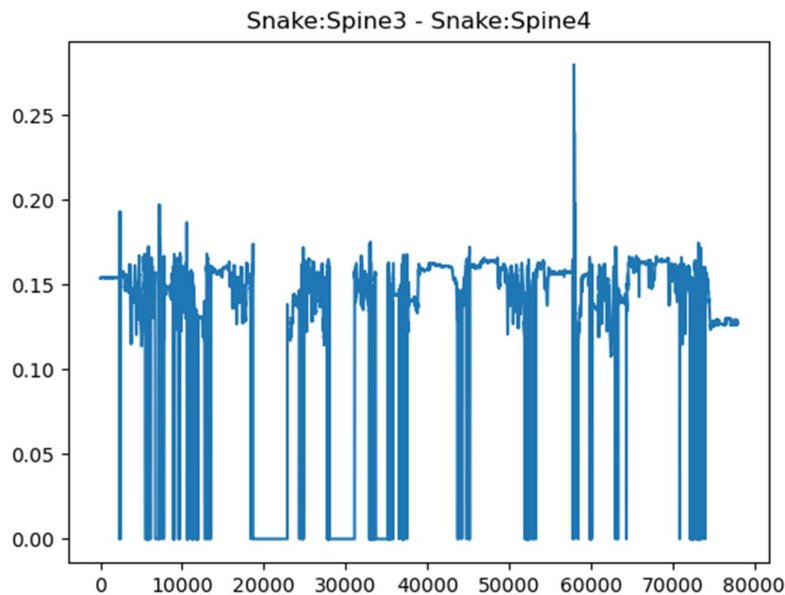
- Αρχικά, στην εμπρόσθια προσπέλαση ο αλγόριθμος υπολογίζει βάσει προηγούμενων τιμών την απόσταση μεταξύ των joints, ή του δίδονται αυτούσιες.
- Στη συνέχεια ο αλγόριθμος με βάση την απόσταση αλλά και τις προηγούμενες θέσεις των joints, έχοντας ένα σημείο αναφοράς, προσπαθεί να τοποθετήσει τα joints σε θέσεις κοντά στην προηγούμενη τιμή που να κρατούν όμως την σωστή απόσταση
 - Για να το επιτύχει αυτό ελέγχει ένα ένα τα joints και προσπαθεί να υπολογίσει τη θέση τους βάσει της απόστασης που πρέπει να έχουν από το σημείο αναφοράς

Προεργασία

Προτού όμως φτάσουμε σε αυτό το σημείο, μέλημα μας ήταν να ελέγξουμε κατά πόσο οι τιμές των δεικτών βρίσκονται σε ένα διορθώσιμο στάδιο. Για να γίνει αυτό, αποφασίσαμε όπως ελέγξουμε τις αποστάσεις μεταξύ των δεικτών μέσα από μια γραμμική γραφική παράσταση. Αυτό που θέλαμε να δούμε είναι κατά πόσο οι αποστάσεις που έχουν οι δείκτες είναι σταθερές. Εάν κάτι τέτοιο δεν ισχύει αυτό σημαίνει πως η ονομασία των δεικτών ή χειρότερα η καταγραφή των δεδομένων στο εργαστήριο δεν τελέστηκε με ακρίβεια. Εάν οι τιμές μας είχαν μεγάλη ανακρίβεια, ο αλγόριθμος δεν θα μπορούσε να διορθώσει τα λάθη στο επίπεδο που θέλουμε

Ευτυχώς στην περίπτωση μας, εξετάζοντας τις γραφικές παραστάσεις παρατηρήσαμε πως οι τιμές που καταγράψαμε για την απόσταση μεταξύ των δεικτών, ήταν σταθερές. Η πιο κάτω γραφική παράσταση δείχνει κάποιο παράδειγμα όπου παρουσιάζονται σε μια γραφική παράσταση οι τιμές της απόστασης όσο περνούν τα frames, μεταξύ 2 δεικτών. Ο αριστερός άξονας των Ψ, απεικονίζει τις τιμές της απόστασης ενώ ο άξονας των Χ απεικονίζει τον

αριθμό των frames. Όσες τιμές είναι μηδέν σημαίνει πως ένας από τους δύο δείκτες λείπει (δηλαδή δεν έχει έγκυρη τιμή)



Εικόνα 7 Απόσταση δύο διαδοχικών δεικτών με το πέρασμα των frames

Για να παράγουμε αυτά τα δεδομένα υλοποιήσαμε ένα μικρό πρόγραμμα(script) στην γλώσσα προγραμματισμού Python όπου υπολογίζεται η απόσταση μεταξύ των σημείων και παράγεται μια γραφική παράσταση χρησιμοποιώντας την βιβλιοθήκη **matplotlib**. Να αναφέρουμε επίσης, ότι στο τέλος του προγράμματος παράγεται ένα αρχείο τύπου .csv για κάθε γραφική παράσταση το οποίο περιέχει τις τιμές της απόστασης μεταξύ δύο δεικτών καθώς επίσης εμφανίζονται στην οθόνη οι median τιμές μεταξύ των αποστάσεων. Πιο κάτω παρουσιάζεται ένα παράδειγμα της εξόδου του προγράμματος. Μετά την επικεφαλίδα αναγράφονται οι median τιμές των αποστάσεων. Δεν αναγράφονται περισσότερες πληροφορίες όπως “περιγραφή” (description) , για να μπορούν να διαβαστούν τα δεδομένα πιο εύκολα από το επόμενο προγραμματάκι που ετοιμάσαμε, όπου υλοποιείται ο αλγόριθμος FABRIK.

```
=====
Evaluation C3Dserver
Version: 1.202
User: mkonst03
Organization: Motion Lab Systems, Inc.
=====
0.05563379393336929
0.06265978207466667
0.07363354394175148
0.12850383600120063|
0.12496188074264532
0.13446833119470128
0.12316518537757801
0.1361559865921411
0.17545062336661443
0.15602107264884152
0.13026038480506863
```

Εικόνα 8 Αποτελέσματα από πρόγραμμα-script

Υλοποίηση

Στην υλοποίηση μας, σπάσαμε το πρόγραμμα σε 4 διαφορετικές περιπτώσεις:

1. Υπάρχει κεφαλή και υπάρχει ουρά
2. Δεν υπάρχει κεφαλή αλλά υπάρχει ουρά
3. Δεν υπάρχει ουρά αλλά υπάρχει κεφαλή
4. Η απόσταση μεταξύ των δύο δεικτών είναι μεγαλύτερη από τις αποστάσεις που θα έπρεπε να έχουν

Κατά την πρώτη περίπτωση, υλοποιήσαμε τον αλγόριθμο, ως έχει και στο άρθρο που δημοσιεύτηκε το 2011 [3].

Για την δεύτερη και τρίτη περίπτωση έπρεπε να κάνουμε μια μικρή τροποποίηση. Αντί να γίνονται επαναλαμβανόμενες προσπελάσεις, στον αλγόριθμο γίνεται μόνο μία προσπέλαση [4] [5]. Σε περίπτωση όπου λείπει η κεφαλή αναγκαστικά μπορεί να γίνει μόνο η ανάποδη προσπέλαση, δηλαδή ξεκινώντας από την ουρά. Σε περίπτωση που λείπει η ουρά τότε αναγκαστικά εκτελείται η εμπρόσθια προσπέλαση δηλαδή ξεκινώντας από την κεφαλή

Για την τελευταία περίπτωση, ελέγχουμε αν το άθροισμα των αποστάσεων που πρέπει να έχουν οι διαδοχικοί δείκτες μεταξύ τους, είναι μεγαλύτερη της απόστασης μεταξύ της κεφαλής και της ουράς. Η σχέση δίδεται από τον πιο κάτω τύπο και σε περίπτωση που ισχύει κάτι τέτοιο τότε αυτό σημαίνει πως οι δείκτες δεν μπορούν να ταξινομηθούν σε οποιαδήποτε άλλη σειρά εκτός από την ευθεία μεταξύ της κεφαλής και της ουράς. Δηλαδή, ο μόνος

τρόπος για να μπορέσει να “δικαιολογηθεί” το αποτέλεσμα της ανίσωσης είναι εάν οι δείκτες μπουν σε μια ευθεία.

$$\sum_{i=1}^n \text{distance}(i, i-1) > \text{distance}(0, n)$$

Εικόνα 9 Μαθηματική συνθήκη κατά την οποία δηλώνει την ευθεία σειρά των δεικτών

$$\text{distance}(a, b) = \sqrt{(a.x - b.x)^2 + (a.y - b.y)^2 + (a.z - b.z)^2}$$

Εικόνα 10 Εξίσωση εύρεσης απόστασεως μεταξύ δύο δεικτών

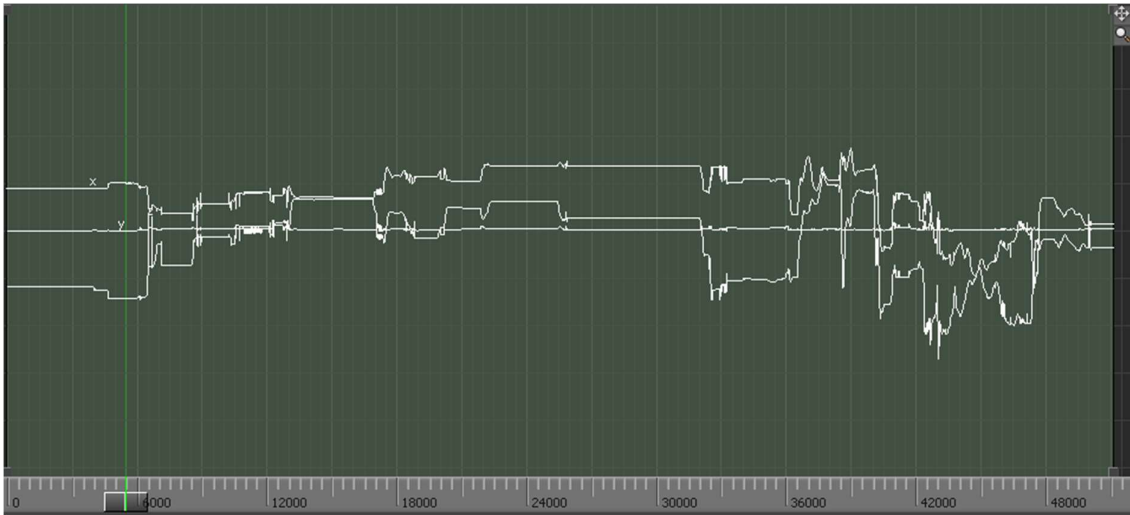
3.2 Σύγκριση αποτελεσμάτων μετά την υλοποίηση του αλγορίθμου

Πριν την εφαρμογή του αλγορίθμου, πολλοί δείκτες μπορεί να είχαν είτε εσφαλμένες τιμές είτε κενά σε διάφορα frames. Μετά το πέρασμα του αλγορίθμου στο αρχείο c3d όμως, οι τιμές αυτές διορθώθηκαν. Η διαφορά αυτή είναι περισσότερο εμφανές στο πρόγραμμα MotionBuilder μιας και για κάθε δείκτη, παρουσιάζονται οι μετατοπίσεις των συντεταγμένων του κάθε δείκτη σε κάθε frame ξεχωριστά. Πιο κάτω μπορούμε να δούμε τις τιμές χ,ψ,ζ και ένα τυχαία επιλεγμένο, αυθαίρετο δείκτη. Όπως μπορούμε να διαπιστώσουμε, υπάρχουν πολλά κενά σε διάφορα frames. Μερικά από αυτά, διορθώνονται από το πρόγραμμα Motion Builder όμως οι τιμές που παίρνουν δεν φαίνεται να είναι ρεαλιστικές. Ιδιαίτερη αναφορά κάνουμε μετά το frame 6000 όπου υπάρχει μια παραβολή για κάθε σημείο, και η κάθε παραβολή παίρνει μεγαλύτερες τιμές από όσο θα έπρεπε.



Εικόνα 11 Μετατόπιση συντεταγμένων πριν την εφαρμογή του FABRIK

Μετά την εφαρμογή του αλγορίθμου FABRIK όμως, μπορούμε να δούμε πως οι τιμές οι οποίες προηγουμένως ήταν κενές ή εσφαλμένες, είναι διορθωμένες με πολύ πιο προσιτές και λογικές τιμές. Ως επίσης, μπορούμε να δούμε πως δεν υπάρχουν κενά μεταξύ των frames κάτι που σημαίνει πως όλες οι εσφαλμένες ή κενές τιμές έχουν υποστεί επεξεργασία από τον αλγόριθμο.



Εικόνα 12 Μετατόπιση συντεταγμένων μετά την εφαρμογή του FABRIK

3.3 Διόρθωση δεδομένων στο MotionBuilder

Σε αυτό το σημείο αξίζει να αναφέρουμε μια εναλλακτική προσέγγιση την οποία μετά από μια μικρή μελέτη, απορρίψαμε. Στο πρόγραμμα MotionBuilder, δίνεται η δυνατότητα δημιουργίας μικρών προγραμμάτων scripts σε γλώσσα Python.

Τρόπος χρήσης

Χρησιμοποιώντας την πιο κάτω εντολή, μας δίνεται η δυνατότητα να χρησιμοποιήσουμε όλες τις λειτουργίες που παρέχει το MotionBuilder.

```
from pyfbSDK import *
```

Για να δημιουργήσουμε κάποιο νέο αντικείμενο στο πλάνο, μπορούμε να χρησιμοποιήσουμε κάποια από τις κλάσεις της βιβλιοθήκης που παρέχονται. Πιο κάτω δοκιμάσαμε να δημιουργήσουμε κάποιο κύβο. Στον κύβο δώσαμε όνομα (testcube) και το μεγεθύνουμε κατά

10 προς όλες του τις συντεταγμένες χρησιμοποιώντας την κλάση για τρισδιάστατα διανύσματα, **FBVector3d**. Το πεδίο **Show**, αφορά την εμφάνιση του νέου μοντέλου την οποία θέσαμε σε αληθής για να μπορέσουμε να δούμε το αποτέλεσμα

```
cube = FBModelCube('testcube')
cube.Show = True
cube.Scaling = FBVector3d(10, 10, 10)
```

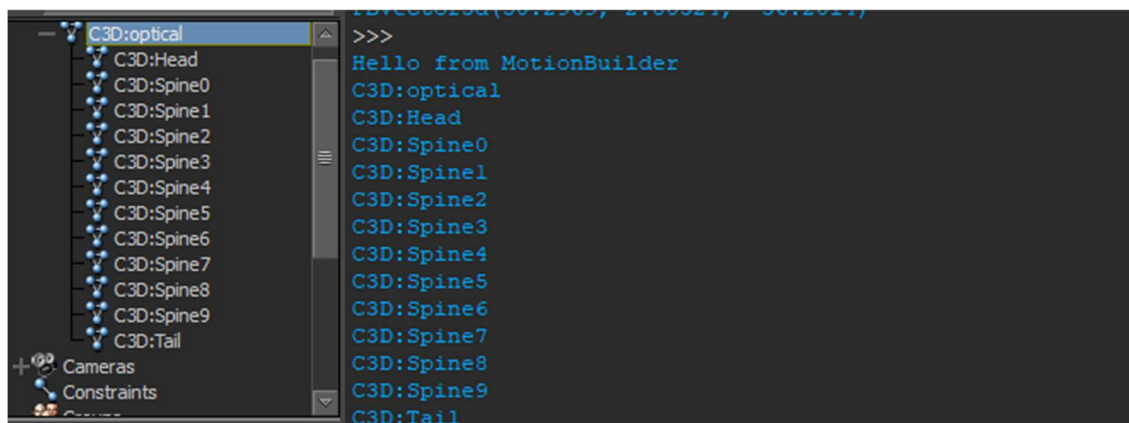
Κάθε κόμβος (node) μπορεί να επιλεγεί με την εντολή η οποία ψάχνει κάποιο κόμβο βάσει ονομασίας. Για να μπορέσουμε να φυλάξουμε στην μνήμη, δηλαδή σε κάποια μεταβλητή ένα κόμβο χρησιμοποιούμε την πιο κάτω εντολή. Στο πιο κάτω παράδειγμα βρίσκουμε και αποθηκεύουμε προσωπικά τον κόμβο που περιέχει όλους τους δείκτες

```
node = FBFindModelByLabelName('C3D:optical');
```

Στη συνέχεια, για επαλήθευση κυρίως, μπορούμε να τυπώσουμε τα ονόματα όλων των δεικτών ως έχουν

```
# Read every C3D optical child
for item in node.Children:
    print(item.LongName);
```

Στην πιο κάτω εικόνα, μπορούμε να δούμε δεξιά τους δείκτες όπως παρουσιάζονται από τον editor του MotionBuilder, ενώ στα αριστερά βλέπουμε ό,τι τύπωσε η κονσόλα που έτρεξε τον πιο πάνω κώδικα. Είναι εμφανές, πως ο κώδικας βρήκε όλους τους κόμβους με επιτυχία!

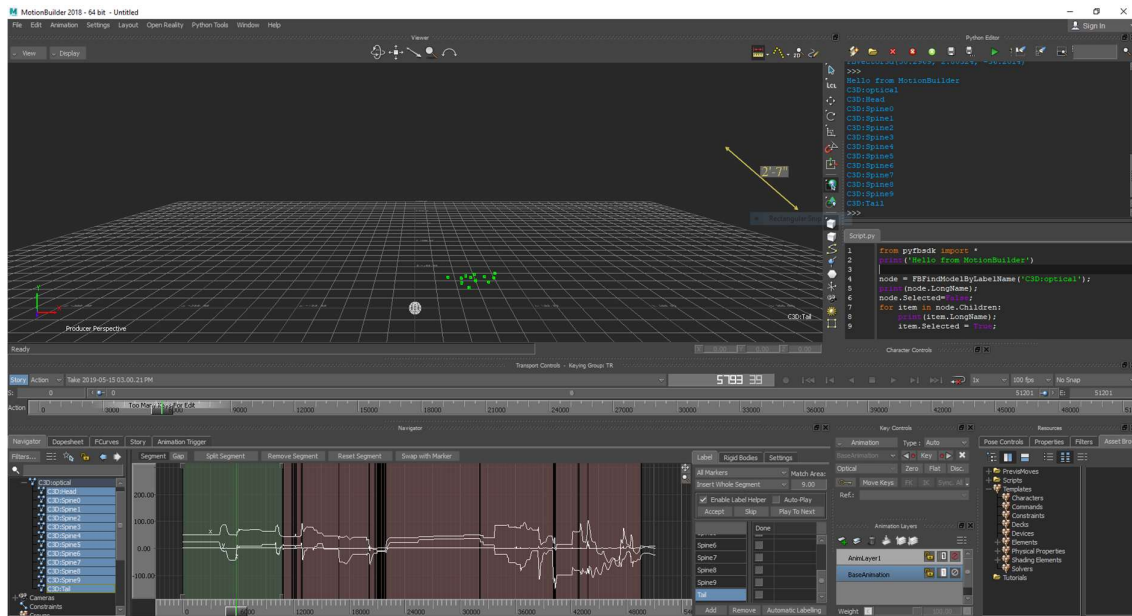


Εικόνα 13 Ονόματα των δεικτών από python script

Αν θέλουμε να επιλέξουμε κάποιο δείκτη, όπως ακριβώς θα τον επιλέγαμε και στο ποντίκι, θα προσθέταμε την εντολή

```
item.Selected = True
```

Όπως φαίνεται στην πιο κάτω εικόνα, στο MotionBuilder αφού τρέξαμε τον κώδικα, οι δείκτες επιλέχθηκαν. Αυτό φαίνεται και στην σκηνή αλλά και στα υπόλοιπα σχετικά παράθυρα



Εικόνα 14 Επιλογή δεικτών στο MotionBuilder μέσω Python script

Αυτό που μας απασχολεί στο πρόγραμμα είναι οι συντεταγμένες των δεικτών. Κάθε δείκτης περιέχει διάφορες ιδιότητες και πεδία στο πρόγραμμα MotionBuilder. Αρκετές από αυτές δεν ανήκουν στο αρχείο c3d αλλά ανήκουν στο αντικείμενο της βιβλιοθήκης pyfbSDK. Έτσι λοιπόν για να πάρουμε τις συντεταγμένες για το επιλεγμένο frame(και τον επιλεγμένο δείκτη φυσικά) πρέπει να χρησιμοποιήσουμε την πιο κάτω εντολή

```
item.PropertyList.Find('Translation (Lcl)').Data
```

Τυπώνοντας το περιεχόμενο, βλέπουμε στο αποτέλεσμα για κάθε δείκτη, την ακριβή τοποθεσία του

```

Hello from MotionBuilder
C3D:optical
C3D:Head
FBVector3d(59.9844, 3.66606, -67.611)
C3D:Spine0
FBVector3d(51.6339, 1.1391, -70.1253)
C3D:Spine1
FBVector3d(55.2214, 3.12816, -61.3242)
C3D:Spine2
FBVector3d(69.5678, 2.69292, -64.0624)
C3D:Spine3
FBVector3d(81.2017, 3.75244, -70.6832)
C3D:Spine4
FBVector3d(78.1912, 2.93283, -60.4291)
C3D:Spine5
FBVector3d(70.801, 3.2124, -48.61)
C3D:Spine6
FBVector3d(58.8145, 3.41905, -51.9392)
C3D:Spine7
FBVector3d(45.5606, 3.08702, -59.2213)
C3D:Spine8
FBVector3d(32.7077, 2.83038, -61.5938)
C3D:Spine9
FBVector3d(44.9759, 1.87562, -49.7543)
C3D:Tail
FBVector3d(50.2969, 2.80324, -36.2014)
>>>

```

Εικόνα 15 Δείκτες και ακριβή τοποθεσία στο MotionBuilder μέσω Python

Μειονεκτήματα

Αν και εκ πρώτης όψεως, δεν φαίνεται άσχημη πρακτική η χρήση της πιο πάνω εξειδικευμένης βιβλιοθήκης, η συγκεκριμένη προσέγγιση απορρίφθηκε. Πιο κάτω παρουσιάζονται οι κύριοι λόγοι

- **Παλαιότερη έκδοση της Python**

Ενώ εμείς χρησιμοποιούσαμε την τελευταία έκδοση του MotionBuilder, το MotionBuilder δεν χρησιμοποιούσε την τελευταία έκδοση της Python. Συγκεκριμένα, το πρόγραμμα χρησιμοποιούσε την έκδοση 2.7. Σύμφωνα με την ανακοίνωση που εκδόθηκε στην επίσημη ιστοσελίδα της Python, από το 2020 η έκδοση 2.7 παύει να υποστηρίζεται. Αυτό σημαίνει πως αργά ή γρήγορα ό,τι θα κτίζαμε με κόπο στην έκδοση της Python 2.7, πολύ πιθανόν να μην λειτουργεί τον επόμενο χρόνο.

- **Δυσκολία εύρεσης εκπαιδευτικού υλικού**

Μπορεί το πρόγραμμα MotionBuilder να είναι ένα ευρέως διαδεδομένο πρόγραμμα, εντούτοις πολύ λίγοι είναι οι προγραμματιστές εκεί έξω οι οποίοι ασχολούνται ενεργά με την δημιουργία προγραμμάτων τα οποία να χρησιμοποιούν την βιβλιοθήκη του MotionBuilder. Επιπρόσθετα, η χρήση αυτής της προσέγγισης στις τελευταίες εκδόσεις του MotionBuilder, και μάλιστα στη γλώσσα προγραμματισμού Python είναι ακόμα πιο περιορισμένη. Επομένως, όχι μόνο το εκπαιδευτικό υλικό του επίσημου παροχέα της βιβλιοθήκης δεν είναι

επαρκή, αλλά και το εκπαιδευτικό υλικό το οποίο παρέχεται από την κοινότητα των προγραμματιστών που χρησιμοποιούν Python στο MotionBuilder.

- **Περιορισμός συμβατότητας**

Το πρόβλημα με αυτή τη βιβλιοθήκη είναι πως μας περιορίζει ως προς τη χρήση της. Πιο συγκεκριμένα, όσα προγράμματα γραφτούν χρησιμοποιώντας τη συγκεκριμένη βιβλιοθήκη θα μπορούν να τρέξουν μόνο στο πρόγραμμα MotionBuilder και δεν θα μπορούν να χρησιμοποιηθούν ανεξάρτητα.

3.4 Παραλλαγή σαύρας

Η υλοποίηση του αλγορίθμου FABRIK στο φίδι ήταν αρκετά απλή καθώς το φίδι περιέχει ένα σκελετό στον οποίο οι δείκτες βρίσκονται σε μια σειρά. Όσο αφορά την σαύρα όμως, έπρεπε να γίνουν κάποιες διαφοροποιήσεις καθώς η σαύρα περιέχει και 4 πόδια: 2 πίσω και 2 μπροστά.

Αυτό σημαίνει πως για την σαύρα πρέπει να γίνει η υλοποίηση του αλγορίθμου για τα εξής:

- Σκελετός
- Μπροστινό πόδι
- Πίσω πόδι

Έτσι η υλοποίηση της σαύρας διαφοροποιήθηκε ελαφρώς έτσι ώστε μόλις διορθωθεί ο σκελετός, ο αλγόριθμος FABRIK να υλοποιηθεί για τα πόδια. Το κάθε πόδι περιλαμβάνει 3 δείκτες. Ο δείκτης που βρίσκεται στον σκελετό και οι δείκτες που βρίσκονται στο πόδι. Έτσι, έχοντας ήδη βρει την σωστή θέση των δεικτών στον σκελετό μπορούμε να καλέσουμε τον αλγόριθμο FABRIK για να βρούμε τις τιμές των 2 δεικτών που υπολείπονται σε κάθε πόδι (αν υπολείπονται). Σε αυτό το σημείο πλέον δεν υπάρχει κάποια αλλαγή στον αλγόριθμο παρά μόνο το γεγονός ότι ο έλεγχος αν υπάρχει η κεφαλή (δλδ ο αρχικός δείκτης) εξαιρείται κι αυτό διότι πλέον θεωρούμε ως κεφαλή τον δείκτη που βρίσκεται στο σημείο του σώματος. Ελέγχουμε μόνο εάν υπάρχουν οι 2 δείκτες, ή εάν υπάρχει η “ουρά” ή εάν υπάρχει ο δείκτης μεταξύ της κεφαλής και της ουράς.

3.5 Προβλήματα και διορθώσεις

Ο αλγόριθμος σε γενικές γραμμές δούλεψε χωρίς ιδιαίτερη δυσκολία. Παρόλα αυτά, ο αλγόριθμος δούλεψε στο δικό μας σενάριο επειδή τα δεδομένα μας είχαν μια συγκεκριμένη δομή. Μετά από περεταίρω έλεγχο του αλγορίθμου και των δεδομένων διαπιστώσαμε τα

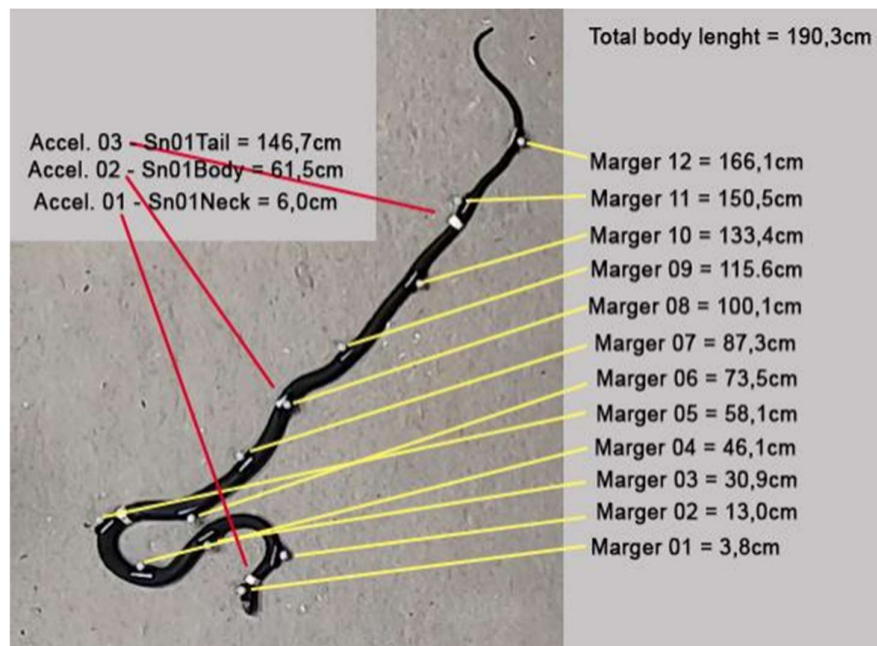
εξής προβλήματα. Κάποια από αυτά τα βρήκαμε σε κάποια αρχεία μας και κάποια άλλα όχι αλλά σκεφτήκαμε πως πιθανόν να τα συναντήσουμε σε άλλα αρχεία:

- Ο αλγόριθμος σε εμάς λειτουργεί διότι πάντα το πρώτο frame περιέχει κάποια τιμή. Αν δεν υπάρχει κάποια τιμή ο αλγόριθμος δεν θα δουλέψει
- Ο αλγόριθμος θεωρεί ως δεδομένο πως σε κάθε frame θα υπάρχει έστω και 1 δείκτης με τιμή. Όπως είδαμε όμως υπήρχαν κάποια συγκεκριμένα αρχεία στα οποία αυτή η συνθήκη δεν ίσχυε

3.6 Snake animation using mocap data

Σε αυτό το στάδιο τοποθετήσαμε ένα rigged μοντέλο στο φίδι βασισμένο στις τιμές που είχαν οι markers,. Χρησιμοποιώντας το c3d αρχείο δημιουργήσαμε animations. Πρώτα ετοιμάσαμε τον σκελετό εφαρμόζοντας κάποιους περιορισμούς (βλ. Εικόνα) και στη συνέχεια εισάξαμε στο Motion Builder τα δεδομένα mocap(οι τιμές που βρίσκονται στο c3d αρχείο) για να χειραγωγήσουμε αυτούς τους περιορισμούς.

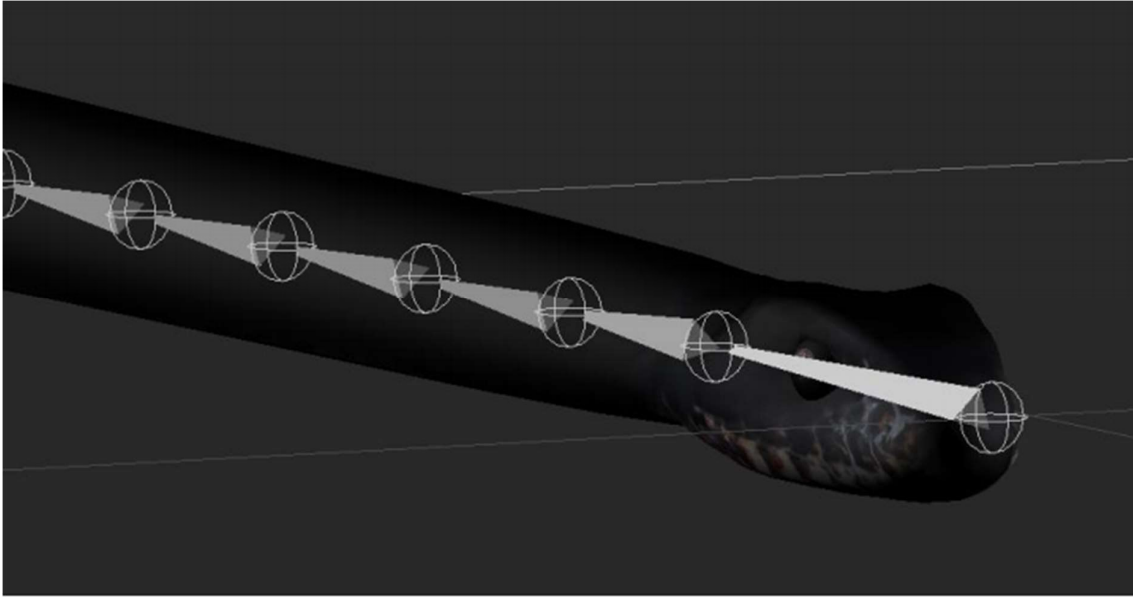
Στην πιο κάτω εικόνα διακρίνουμε πως τοποθετήθηκαν οι δείκτες στο φίδι. Για το φίδι έχουμε ένα σύνολο από 12 δείκτες ενώ το μέγεθος του φιδιού ήταν 190.3 εκατοστόμετρα.



Εικόνα 16 Ακριβής τοποθεσία δεικτών στο φίδι

Προεργασία

Εκτός από τα C3D αρχεία που δημιουργήθηκαν στο προηγούμενο στάδιο της διπλωματικής μου εργασίας, έγινε η δημιουργία ενός μοντέλου, πραγματικού φιδιού. Το μοντέλο είναι στημένο με ένα σκελετό που αποτελείται από μικρά ισομέγεθρα joints. Για το Motion Builder είναι σημαντικό όπως όλα τα joints να οριοθετηθούν σωστά στους άξονες (default axes configuration) αλλιώς οι περιορισμοί δεν θα εφαρμόσουν σωστά.

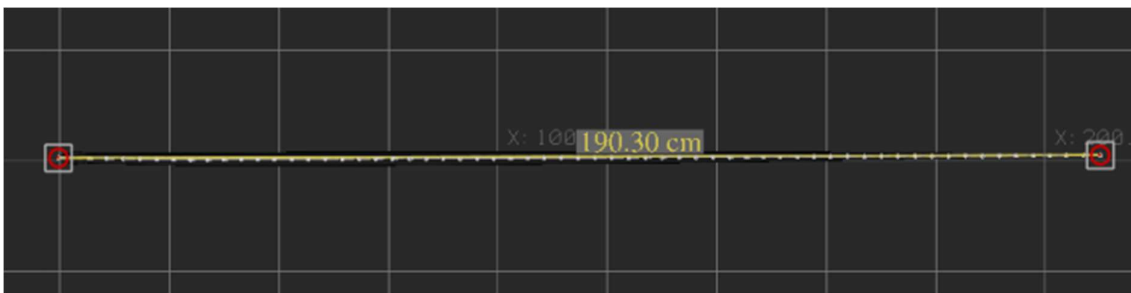


Εικόνα 17 Το μοντέλο του φιδιού που περιέχει σκελετό, mesh και texture

Στη συνέχεια προχωρήσαμε σε μια αντιστοιχία μεταξύ των joints και των markers. Όπως μπορείτε να δείτε πιο κάτω το πρώτο βήμα ήταν να θέσουμε το σωστό μήκος του φιδιού. Αντιστοιχίσαμε το μοντέλο μας πρώτα να είναι όσο μεγάλο όσο τα 190,30 εκατοστά.



Εικόνα 18 Το μοντέλο του φιδιού με τα joints



Εικόνα 19 Το μοντέλο του φιδιού scaled στα 190.30 εκατοστά

Ακολούθως, δημιουργήσαμε τον πιο κάτω πίνακα όπου κάναμε μια αντιστοιχία των πραγματικών αποστάσεων με τους δείκτες που ορίσαμε στα c3d αρχεία. Στην πιο πάνω εικόνα μπορούμε να δούμε που τοποθετήθηκαν οι markers. Στην προηγούμενη ενότητα αυτοί οι markers είχαν μια ετικέτα-ονομασία. (π.χ. spine2). Για να μην υπάρξει σύγχυση ονομάσαμε τα joints που προκύπτουν με το όνομα του αντίστοιχου δείκτη

<i>Marker ID</i>	Marker name	Joint name	Real-life distance from start(cm)
1	Head	(same)	3.8
2	Spine0		13.0
3	Spine1		30.9
4	Spine2		46.1
5	Spine3		58.1
6	Spine4		73.5
7	Spine5		87.3
8	Spine6		100.1
9	Spine7		115.6
10	Spine8		133.4
11	Spine9		150.5
12	Tail		166.1

Τελευταίο βήμα της προεργασίας ήταν όπως να ονομαστούν τα joints στο πρόγραμμα MotionBuilder και να χρωματιστούν καθαρά και μόνο για να μπορεί να είναι πιο εύκολο να τους ανιχνεύσουμε στην συνέχεια.

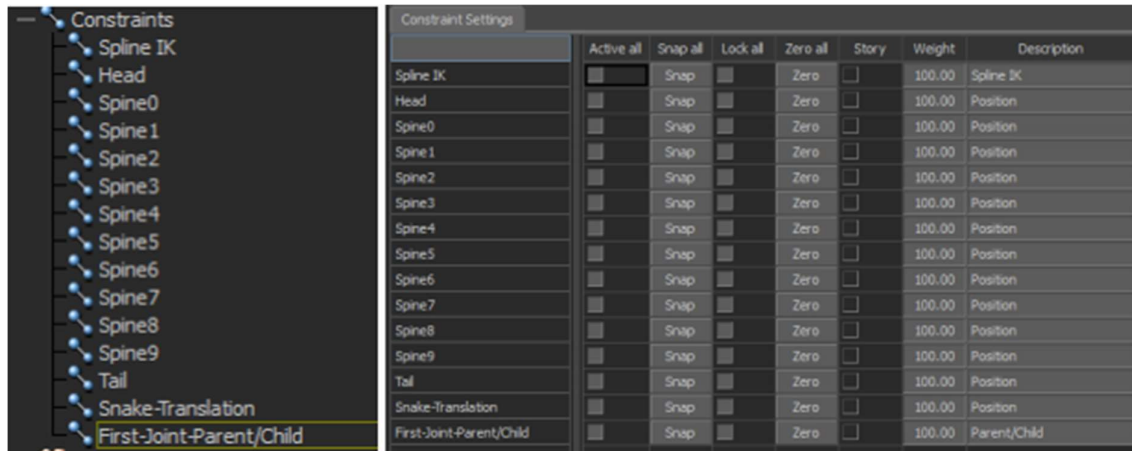


Εικόνα 20 Χρωματισμός των joints στο μοντέλο φιδιού

Εφαρμογή (applying constraints)

Εφαρμόζουμε ένα constraint για την προσομοίωση κίνησης, το οποίο αφήσαμε στην ονομασία Spline IK και στη συνέχεια εφαρμόσαμε ένα constraint για κάθε δείκτη. Στην εικόνα, μπορείτε να δείτε τις ονομασίες των constraints όπου για τους δείκτες, έχουμε τις

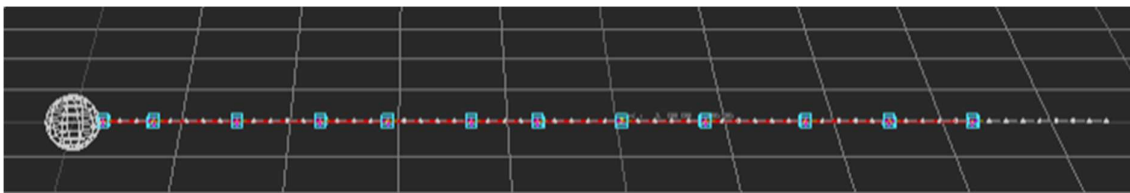
ίδιες ονομασίες με τις ετικέτες τους για να μη υπάρχει σύγχυση. Οι constraints που εφαρμόστηκαν στους δείκτες είναι υπεύθυνοι για την χειραγώγηση(manipulation) του φιδιού. Τέλος, δημιουργήσαμε 1 constraint θέσης με την ονομασία snake-translation και ένα constraint για το κεφάλι του φιδιού που όμως δεν ανήκει στο σώμα του.



Εικόνα 21 Ονομασίες constraints στο πρόγραμμα Motion Builder

Εισαγωγή των δεικτών από C3D αρχεία

Κατά την εισαγωγή του αρχείου εμφανίζονται 12 δείκτες για το φίδι. Κάθε δείκτης πρέπει να τοποθετηθεί στο σωστό σημείο του φιδιού. Για μεγαλύτερη ακρίβεια βάζουμε το φίδι σε μια ευθεία γραμμή και τους δείκτες στο σημείο που υπολογίσαμε κατά το προηγούμενο στάδιο, της προεργασίας.



Εικόνα 22 Αντιστοιχία των δεικτών στο φίδι

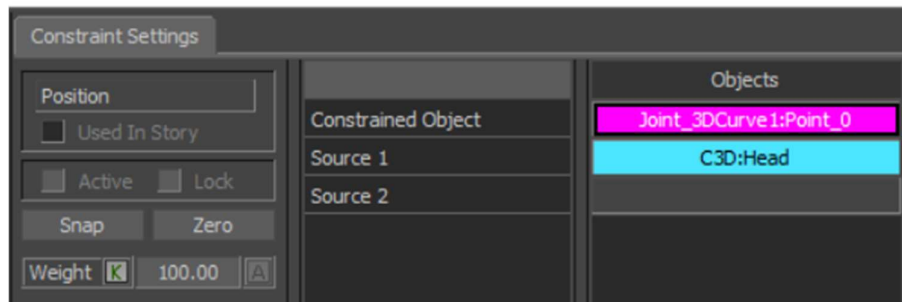
Στη συνέχεια, για κάθε δείκτη πρέπει να εισαχθεί ένα key-frame. Αυτό γίνεται ακολουθώντας τα πιο κάτω βήματα:

1. Επιλογή δείκτη
2. Βρίσκουμε το κλειδί που έχει αστεράκι. (key*)
 - α. Εάν δεν υπάρχει αστεράκι τότε πατάμε next key ή previous key

3. Επιλέγουμε στις ρυθμίσεις Base animation και optical
4. Θέτουμε το τύπο σε “Fixed”

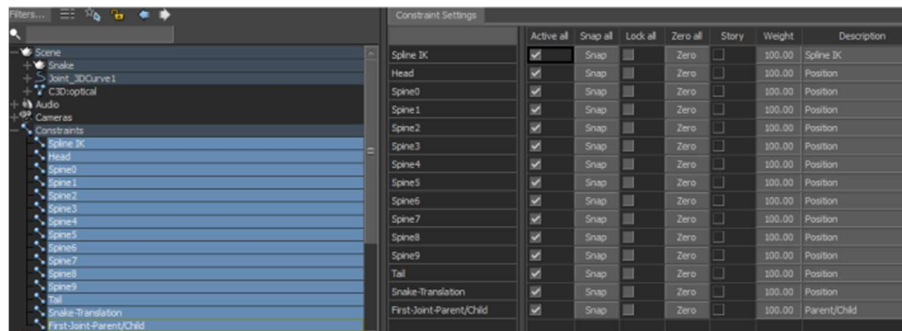
Ενεργοποίηση περιορισμών

Ο κάθε δείκτης πρέπει να είναι σε θέση να μετακινεί τον σκελετό. Επομένως, στις ρυθμίσεις του κάθε περιορισμού (constraint), επιλέξαμε ως πηγή να είναι ο αντίστοιχος δείκτης έτσι ώστε η θέση του να ορίζεται από την θέση του δείκτη. Η πιο κάτω εικόνα δείχνει τις ρυθμίσεις για τον δείκτη της κεφαλής (head marker)



Εικόνα 23 Ρυθμίσεις δείκτη

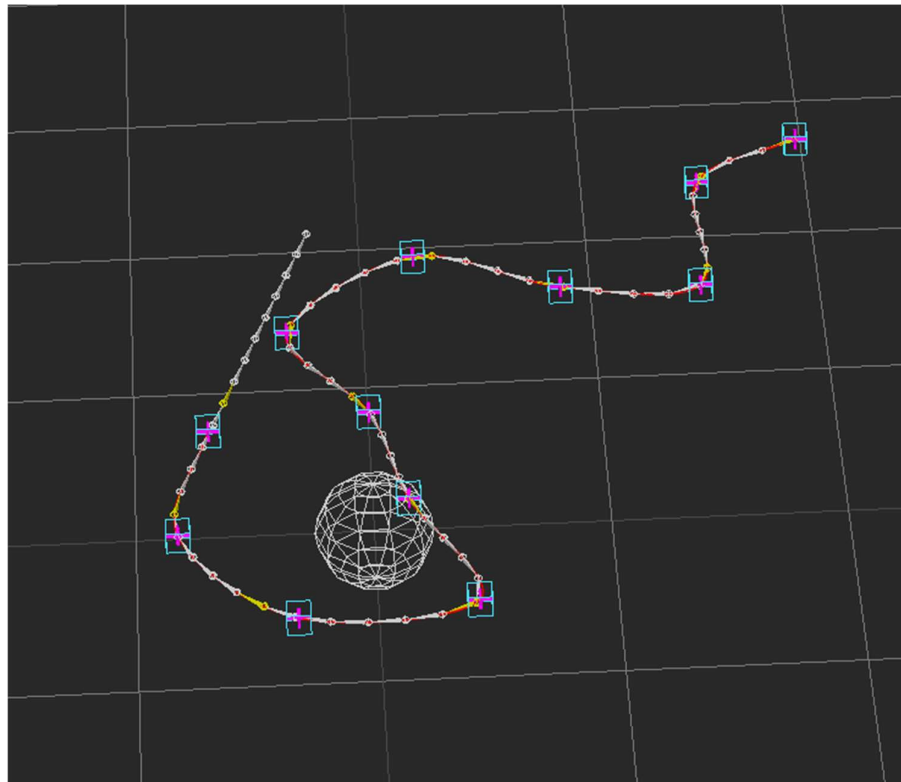
Δεν θα μας ήταν χρήσιμη όλη αυτή η διαδικασία εάν δεν ήταν ενεργοποιημένη. Έτσι στο τελευταίο στάδιο, επιλέξαμε όλους τους περιορισμούς που ρυθμίσαμε προηγουμένως και τους ενεργοποιήσαμε πατώντας το κουμπί “Active all” και στη συνέχεια “snap all”. Για να είμαστε πιο ασφαλείς όμως αποθηκεύουμε ένα αντίγραφο (backup) μήπως έχουμε ρυθμίσει κάτι λάθος.



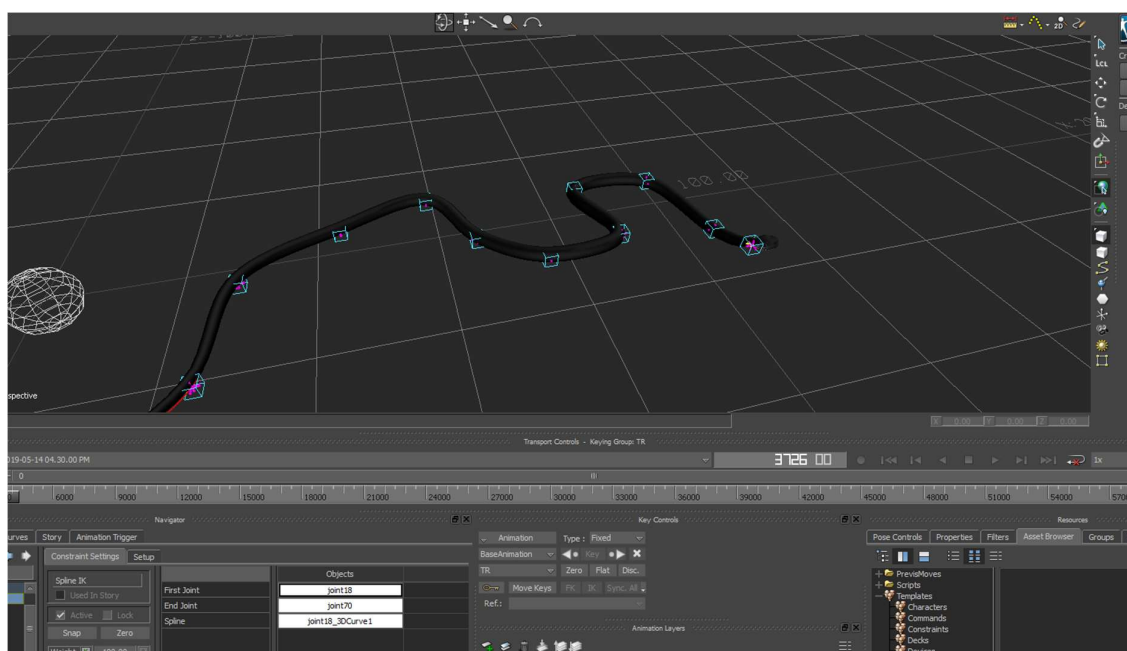
Εικόνα 24 Ενεργοποίηση δεικτών

Τα αποτελέσματα στο Motion Builder. Όπως μπορούμε να δούμε το φίδι κινείται σωστά βάσει των δεικτών. Παρόλα αυτά το splineIK δεν είναι τόσο ακριβής αν χρησιμοποιήσουμε το αρχείο το οποίο δεν υπέστη επεξεργασία από τον αλγόριθμο FABRIK. Αν προσέξουμε τα

χρώματα (το κίτρινο) όπου τοποθετήσαμε αρχικά τους δείκτες, υπό κανονικές συνθήκες θα έπρεπε να είχε κάποιο δείκτη επάνω του. Αντί αυτού βλέπουμε πως ξεφεύγει ελαφρώς σε κάποιες περιπτώσεις. Ως επίσης υπάρχουν αρκετές περιπτώσεις όπου κάποιοι δείκτες απουσιάζουν με αποτέλεσμα να πρέπει να γίνει κάποια διόρθωση. Γι' αυτό χρειαστήκαμε να υλοποιήσουμε κάποιο αλγόριθμο διόρθωσης των τιμών.



Εικόνα 25 Αποτελέσματα εφαρμογής περιορισμών 1 – Χωρίς διόρθωση FABRIK



Εικόνα 26 Αποτελέσματα εφαρμογής περιορισμών 2 - Με διόρθωση FABRIK και Mesh

3.7 Διόρθωση κεφαλής και ουράς στο φίδι

Μετά από την εφαρμογή περιορισμών, εντοπίσαμε 3 προβλήματα τα οποία έπρεπε να διορθωθούν, αυτή τη φορά στο Motion Builder, και όχι στο ίδιο το αρχείο c3d.

1. Το κεφάλι δεν ακολουθεί σωστά το σώμα
2. Η ουρά δεν έχει μια φυσιολογική καμπή στο τέλος
3. Σε κάποια σημεία, η ουρά του φιδιού περνάει πάνω από το υπόλοιπο σώμα του, αλλά στο μοντέλο μας αντί να υπάρχει αυτό το αποτέλεσμα, η ουρά μπαίνει μέσα στο σώμα του δημιουργώντας ένα μη-ρεαλιστικό σαφώς αποτέλεσμα.

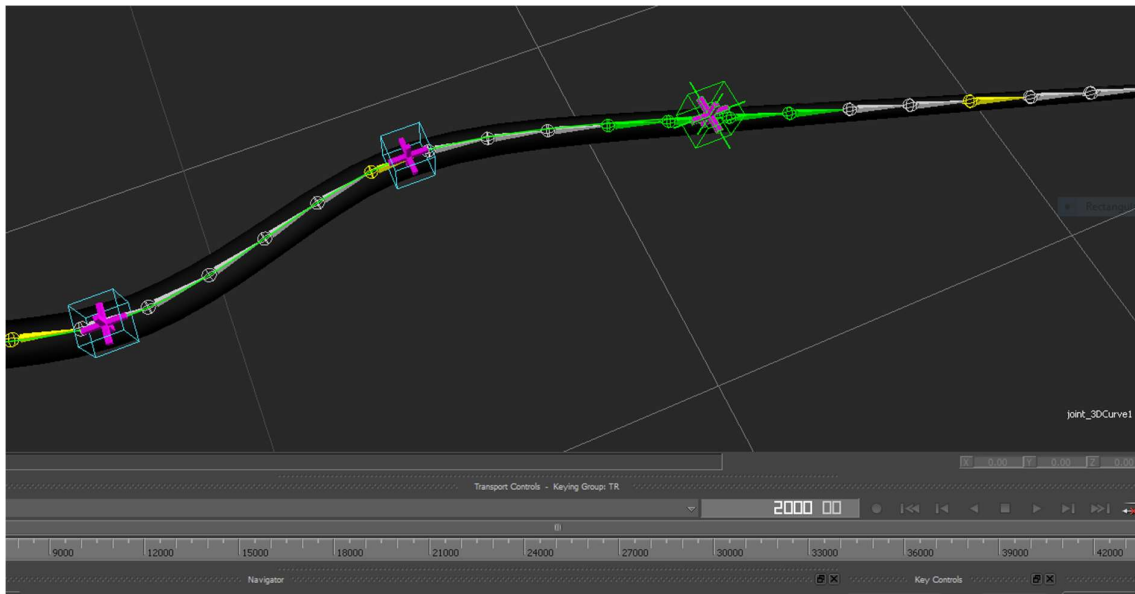
Όσο αφορά το τρίτο πρόβλημα, είναι κάτι το οποίο μπορεί να διορθωθεί από κάποιο animator. Έτσι, διορθώσαμε τα υπόλοιπα δύο προβλήματα που χρειάζονταν καθαρά προγραμματιστική προσέγγιση.

Για το πρόβλημα της κεφαλής, αποφασίσαμε όπως ξαναακολουθήσουμε την διαδικασία που περιγράφεται στην προηγούμενη φάση (2.6) με μία βασική αλλαγή. Αντί ο δείκτης της κεφαλής (Head) να ακολουθεί το 2^ο joint, όπως ακολουθεί το πρώτο. Πράγματι, το πρόβλημα διορθώθηκε. Όσο αφορά την ουρά, η λύση ήταν πιο σύνθετη καθώς απαιτούσε αλλαγές στο splineIK που έπρεπε να γίνουν σε Python.

Η λύση ήταν όπως τοποθετηθεί ένας επιπλέον δείκτης στο τέλος, ο “end” marker ή αν προτιμάτε μια καινούργια ουρά. Ο τελευταίος δείκτης, χρησιμοποιώντας την βιβλιοθήκη της Autodesk “fbxsdk”, ορίστηκε έτσι ώστε να είναι ίσος με την τοποθεσία του δείκτη “tail” (που ήταν και ο τελευταίος στο φίδι που καταγράφηκε), στο προηγούμενο του frame. Δλδ,

$$End_{position} = Tail_{position-1}$$

Κρίνοντας από τα αποτελέσματα, μπορούμε να δούμε πως ο καινούργιος δείκτης επηρεάζει θετικά ως ένα βαθμό το μοντέλο καθώς η ουρά παίρνει ένα πιο καμπυλωτό σχήμα



Εικόνα 27 Η προσθήκη του τελευταίου δείκτη

Κεφάλαιο 4

Δημιουργία εφαρμογής DDMT-Viewer

4.1. Πρόβλημα, προεργασία και καθορισμός των απαιτήσεων	30
4.2. Σχεδίαση συστήματος	31
4.3. Υλοποίηση, έλεγχος και αναθεώρηση	34
4.4. Έλεγχος, αναθεώρηση και optimizations	42
4.5. Τελικό αποτέλεσμα, σφάλματα και συμπεράσματα	43
4.6 Μελλοντικά σχέδια	44

4.1 Πρόβλημα, προεργασία και καθορισμός των απαιτήσεων

Το πανεπιστήμιο του Swansea, στα πλαίσια της έρευνας του, δημιούργησε ένα λογισμικό το οποίο παρέχει αρκετά από τα εργαλεία που χρειάζεται η ερευνητική ομάδα του Terra Cypria. Το συγκεκριμένο πρόγραμμα ονομάζεται *Daily Diary Multi Trace* [17] (από εδώ και στο εξής θα αναφέρεται ως DDMT) και χρησιμοποιήθηκε στα πλαίσια της πτυχιακής αυτής εργασίας, με κύριο χρήστη την ομάδα του κύριου Σάββα Ζότου, για την επεξεργασία ορισμένων κινήσεων.

Στο DDMT, μεταξύ πολλών λειτουργιών, μπορεί κανείς να ορίσει τις κινήσεις που παρουσιάζονται στο αρχείο με βάση κάποιο χρονικό διάστημα. Ακολουθώντας το πρόγραμμα, παρουσιάζει στην οθόνη μερικά στοιχεία, όπως τον κωδικό της κίνησης, μερικά εικονίδια έτσι ώστε ο χρήστης του συστήματος να είναι σε θέση να κατανοήσει ποια κίνηση είναι ενεργή στο χρονικό διάστημα που επέλεξε.

Το πρόβλημα με το συγκεκριμένο πρόγραμμα, είναι πως ενώ οι κινήσεις μπορούν να ορισθούν και να τύχουν περεταίρω επεξεργασίας μέσω του προγράμματος DDMT, εντούτοις δεν υφίσταται η δυνατότητα απεικόνισης (visualization), της ίδιας της κίνησης. Δηλαδή, ενώ το πρόγραμμα είναι σε θέση να ξεχωρίσει ποια κίνηση είναι ενεργή σε κάποιο χρονικό διάστημα, επί της ουσίας ο χρήστης δεν μπορεί να δει στην οθόνη του κάποιο απόσπασμα της κίνησης παρά μόνο μερικά στατικά εικονίδια και κάποιους κωδικούς. Επομένως, στα πρόθυρα της διπλωματικής αυτής εργασίας, ορίστηκε ως αναγκαία η αναβάθμιση του συγκεκριμένου προγράμματος έτσι ώστε να είναι σε θέση να καλύψει τις ανάγκες της ερευνητικής ομάδας Terra Cypria

Εφ' όσων το πρόγραμμα DDMT, είναι ένα εξωτερικό πρόγραμμα κλειστού πηγαίου κώδικα, είναι αναμενόμενο πως εκτός κι αν υπάρξει κάποια συνεργασία αλλά και συμφωνία με το Πανεπιστήμιο Κύπρου, η πρόσβαση στον κώδικα του προγράμματος DDMT αλλά και η τροποποίηση του, δεν καθίσταται εφικτή. Το καλύτερο που μπορούσαμε να πετύχουμε (και το πετύχαμε) είναι όπως ζητήσουμε από την ομάδα του πανεπιστημίου του Swansea που είναι υπεύθυνη για το συγκεκριμένο πρόγραμμα, να τροποποιήσει το πρόγραμμα έτσι ώστε να μπορούμε να διαχειριστούμε τα αποτελέσματα που εξάγονται από κάποιο δικό μας λογισμικό. Για να μπορέσουμε να το κάνουμε αυτό σε πραγματικό χρόνο όμως, απαραίτητο θα ήταν όπως εδραιωθεί μια επικοινωνία μεταξύ των δύο λογισμικών. Έτσι για κάθε αλλαγή που γίνεται στο DDMT, το δικό μας λογισμικό θα την αναγνωρίζει και θα δημιουργεί κάποιο visualization

Με την ομάδα της Terra Cypria του Ανοικτού Πανεπιστημίου Κύπρου, έχοντας ως επικεφαλή τον Δρ. Σάββα Ζότο, θέσαμε κάποια κριτήρια τα οποία θα έπρεπε να πληροί η νέα εφαρμογή που αναπτύξαμε. Αποφασίσαμε όπως πρώτα απ' όλα:

- Η εφαρμογή να μπορεί σε πραγματικό χρόνο και σε “συνεργασία” με το πρόγραμμα DDMT, να εντοπίζει της αλλαγές που γίνονται στο DDMT και να τις παρουσιάζει
- Η εφαρμογή να καταγράφει κάπου τις κινήσεις που σημειώθηκαν
- Να είναι ευέλικτο σε ρυθμίσεις έτσι ώστε να μην χρειάζεται η συνδρομή του προγραμματιστή για κάθε αλλαγή.
 - ο π.χ. Αν αλλάξει το socket, η αλλαγή να μην χρειάζεται προγραμματιστικές αλλαγές

4.2 Σχεδιασμός συστήματος

Όπως και σε κάθε μεγάλο έργο, προτού καταλήξει μια ομάδα σε ένα τελικό σχεδιασμό του συστήματος, πολλά άλλα προσχέδια μελετώνται καθώς και διαφορετικές τεχνολογίες δοκιμάζονται. Τόσο σε ευκολία αλλά και αποτελεσματικότητα. Το συγκεκριμένο έργο δεν αποτελεί εξαίρεση και πολλοί άλλοι σχεδιασμοί προτάθηκαν. Πιο κάτω αναγράφονται οι διαφορετικές μέθοδοι που σκεφτήκαμε να ακολουθήσουμε, τα πλεονεκτήματα και μειονεκτήματα κάθε τρόπου αλλά και την τελική απόφαση

Δημιουργία λογισμικού ως Web Application

Εφ' όσων το πρόγραμμα ήδη χρησιμοποιεί socket programming [1], μία από τις λύσεις που προτάθηκαν ήταν όπως δημιουργηθεί μια διαδικτυακή εφαρμογή η οποία θα ήταν

εγκατεστημένη στο δίκτυο του Πανεπιστημίου Κύπρου, και θα ερχόταν σε απευθείας επικοινωνία με τον χρήστη που θα χρησιμοποιούσε το σύστημα μέσω κάποιου φυλλομετρητή (browser). Το πρόβλημα όμως ήταν πως μια τέτοια εφαρμογή θα έφερνε άλλου είδους προβλήματα μεταξύ των οποίων

- Δυσκολία στην χρήση καθώς θα έπρεπε για κάθε χρήση να γίνεται διαφορετικό setup. Με την λέξη setup, εννοούμε τη διαδικασία όπου ο χρήστης θα έπρεπε να ενώσει την εφαρμογή DDMT που τρέχει στον υπολογιστή του με την διαδικτυακή εφαρμογή. Κάτι τέτοιο θα ήταν εξαιρετικά δύσκολο για κάποιο χρήστη που δεν έχει το βασικό υπόβαθρο στα δίκτυα καθώς εμπλέκονται έννοιες όπως IP address, port number, socket κλπ.
- Θέματα ασφάλειας στον server του Πανεπιστημίου Κύπρου
- Δυσκολία εγκατάστασης στον server του πανεπιστημίου

Δημιουργία χρησιμοποιώντας τα προγράμματα της Microsoft και C#

Όπως φάνηκε πιο πάνω, η καταλληλότερη λύση θα ήταν η δημιουργία ενός desktop application, αφού ούτως ή άλλως και το DDMT, είναι μια εφαρμογή τύπου desktop. Μετά από μια μικρή έρευνα που ακολούθησε όσο αφορά την δημιουργία desktop application, ήταν η χρήση των διαφόρων προγραμμάτων που παρέχει η Microsoft . Πιο συγκεκριμένα η χρήση των Windows Forms [2] με το Visual Studio Community edition.

Τα συγκεκριμένα εργαλεία που παρέχει η Microsoft είναι δωρεάν, και εύκολα στην χρήση. Με απλές κινήσεις drag-n-drop που παρέχονται στο visual studio, εύκολα μπορεί κάποιος να σχεδιάσει το πρόγραμμα οπτικά. Κατ'επέκταση, δίνεται η δυνατότητα χρήσης της γλώσσας προγραμματισμού C# η οποία είναι ευρέως διαδεομένη και πολύ κοντά στη γλώσσα JAVA που διδαχθήκαμε στο πανεπιστήμιο.

Ο λόγος που απορρίφθηκε η συγκεκριμένη πρόταση είναι για τον λόγο πως αυτά τα εργαλεία είναι περιορισμένα μόνο για λειτουργικά συστήματα τύπου Windows. Εμείς όμως στις μηχανές του εργαστηρίου ήδη χρησιμοποιούμε εκτός από Windows και Linux τύπου CentOS οπότε προτιμήσαμε κάποιες άλλες εναλλακτικές μεθόδους.

Δημιουργία λογισμικού χρησιμοποιώντας Qt framework και C++

Το Qt Framework είναι ένα ευρέως διαδεμένο framework για την υλοποίηση εφαρμογών τύπου desktop το οποίο ακόμη και η ίδια η Microsoft χρησιμοποιεί για την ανάπτυξη των δικών της

εφαρμογών. Χαρακτηριστικό παράδειγμα το πρόγραμμα RedisDesktopManager. Δημιουργοί του θεωρούνται οι Haavard Nord και Eirik Chambe-Eng ενώ η εταιρεία που έχει τα δικαιώματα του framework είναι η “The Qt Company”. Ιστορικά, οι εταιρείες που είχαν τα δικαιώματα του framework σε αύξουσα χρονολογική σειρά:

- Trolltech (1991 – 2008)
- Nokia (2008 – 2011)
- Qt Project (2011 - σήμερα)
- Digia (2012 – 2014)
- The Qt company (2014 - σήμερα)

Είναι σχετικά εύκολο, παρέχει πολλά βοηθητικά εργαλεία, υπάρχουν πολλά βοηθητικά μαθήματα στο διαδίκτυο και ένα πολύ αναλυτικό documentation. Δίνει την δυνατότητα προγραμματισμού σε 2 βασικές γλώσσες προγραμματισμού. C++ και Python

Προτιμήθηκε το Qt framework αλλά όχι η γλώσσα προγραμματισμού C++ . Ο λόγος είναι επειδή θα χρειαζόμουν περισσότερο χρόνο υλοποίησης του προγράμματος μέχρι να μάθω C++. Ως επίσης, τη γλώσσα προγραμματισμού Python, την είχαμε ξαναχρησιμοποιήσει για άλλα προγράμματα ή scripts κατά την διάρκεια αυτής της πτυχιακής εργασίας αλλά ως επίσης, η γλώσσα Python είναι πιο εύκολη

Δημιουργία λογισμικού χρησιμοποιώντας PySide2

Για την χρήση του Qt framework, εκτός από C++, υπάρχουν δύο διάσημες βιβλιοθήκες στη γλώσσα προγραμματισμού Python:

1. PyQt5
2. PySide2

Και οι δύο βιβλιοθήκες έχουν παρόμοια δομή και σε πολλές περιπτώσεις τις ίδιες συναρτήσεις. Η διαφορά είναι πως αρχικά, το Qt framework δεν ήταν διαθέσιμο σε python. Έτσι, ο Phil Thompson της εταιρείας Riverbank Computing Ltd αποφάσισε όπως δημιουργήσει μια βιβλιοθήκη σε python η οποία θα δίνει την δυνατότητα χρήσης των C++ modules που υπάρχουν για το Qt framework. Υπήρξαν διάφορες εκδόσεις με την 5η να είναι η τελευταία και πιο ανανεωμένη.

Η εταιρεία που είχε τα δικαιώματα του framework το 2009, Nokia, δεν άργησε να παρατηρήσει πόσο διαδεδομένη ήταν η χρήση της Python με το Qt toolkit, κι έτσι αποφάσισε όπως

δημιουργήσει τα δικά της bindings για Python. Έτσι δημιουργήθηκε η νέα βιβλιοθήκη PySide η οποία είναι και η επίσημη βιβλιοθήκη που υποστηρίζει το Qt framework όσο αφορά την Python.

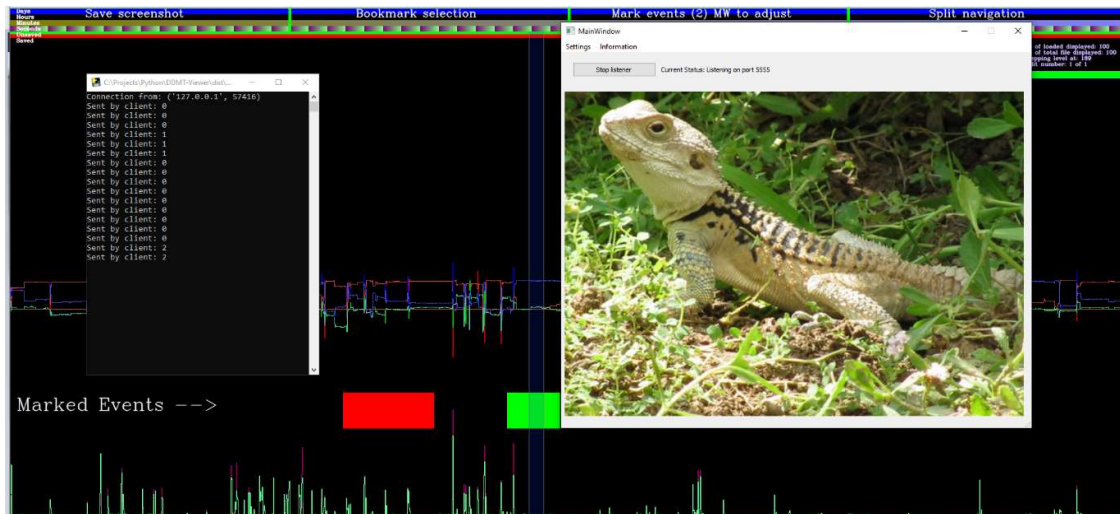
Για τους πιο πάνω λόγους, θεωρήσαμε πως αν και είναι λίγο πιο διαδεδομένη η βιβλιοθήκη PyQt5, όπως χρησιμοποιήσουμε την βιβλιοθήκη PySide2 καθώς είναι η τελευταία, πιο ανανεωμένη και η βιβλιοθήκη που υποστηρίζεται επίσημα.

4.3 Υλοποίηση, έλεγχος και αναθεώρηση

Αρχικά, ο Δρ. Σάββας Ζότος, επικοινωνήσε με τον υπεύθυνο του λογισμικού DDMT, Mark D. Holton για να γίνει μια τροποποίηση του προγράμματος έτσι ώστε να μπορεί να εξάγει τα αποτελέσματα του σε κάποιο socket. Έτσι, ο Mark D. Holton όρισε όπως το πρόγραμμα να εξάγει τα αποτελέσματα του στη διεύθυνση IP 127.0.0.1 και στο Port number 5555.

Ακολουθως, έγινε κάποιος ελαφρύς σχεδιασμός της φόρμας μέσω του προγράμματος Qt designer , υλοποιήθηκε η σύνδεση των δύο εφαρμογών και πλέον ο έλεγχος της ροής του προγράμματος μετατέθηκε σε εμάς.

Socket programming - Σύνδεση δύο εφαρμογών



Εικόνα 28 Πρώτοι έλεγχοι επικοινωνίας με το DDMT

Στο socket programming, υπάρχουν δύο βασικά τερματικά σημεία(end points). Βασισμένο στην αρχιτεκτονική client-server [3], απαραίτητη προϋπόθεση αποτελεί όπως υπάρξει κάποιο πρόγραμμα το οποίο να περιμένει σύνδεση στο socket: 127.0.0.1:5555 ,δηλαδή ο server αλλά και ένα πρόγραμμα το οποίο θα πάρει τον ρόλο του πελάτη (client). Το DDMT πήρε τον

ρόλο του πελάτη. Δηλαδή, κάθε φορά που ενεργοποιείται η κατάλληλη διεργασία, το DDMT συνδέεται σε κάποιο process και στέλνει τα σήματα. Έτσι, αυτό που έπρεπε να υλοποιήσουμε εμείς είναι τον server.

Χρησιμοποιώντας τη γλώσσα προγραμματισμού Python σε συνδυασμό με τις βιβλιοθήκες os και socket, δημιουργήσαμε ένα server ο οποίος περιμένει κάποιον πελάτη (client) να συνδεθεί στο socket 127.0.0.1:5555. Αν και δεν χρησιμοποιήθηκε στο τελικό προϊόν, δημιουργήσαμε ως επίσης ένα client πρόγραμμα μέσω python, το οποίο μιμείται τον τρόπο λειτουργίας του DDMT. Χρησιμοποιήθηκε κυρίως για σκοπούς ελέγχου.

Αφού είδαμε πως ο server λειτουργεί κανονικά, θα έπρεπε να τοποθετηθεί με κάποιο φιλικό προς τον χρήστη τρόπο, στο κυρίως πρόγραμμα που αναπτύσσουμε. Αποφασίσαμε όπως τοποθετήσουμε ένα μεγάλο κουμπί το οποίο υπεύθυνο θα ήταν για να ανοίγει και να κλείνει τον server. Η επιλογή έγινε βάσει των καλών πρακτικών που χρησιμοποιούνται σε εφαρμογές κτισμένες στο User Experience (UX). Με αυτό τον τρόπο, το κουμπί θα ήταν το πρώτο πράγμα που θα έβλεπε ο χρήστης. Αν δεν χρειάζεται κάποια εξειδικευμένη ρύθμιση τότε είναι το μόνο που χρειάζεται να πατήσει ο χρήστης.

Σε πιο τεχνικό επίπεδο, λόγω του ότι υπάρχει ένα κυρίως πρόγραμμα και μια δευτερεύουσα διεργασία η οποία ακούει στα σήματα κάποιας άλλης διεργασίας, θα έπρεπε όπως ο server να τρέχει παράλληλα με το κυρίως πρόγραμμα. Γι' αυτό τον λόγο χρησιμοποιήθηκε παράλληλος προγραμματισμός κάνοντας χρήση των νημάτων που προσφέρει η Python. Προς το τέλος όμως αντικαταστάθηκαν τα νήματα με τη κλάση που προσφέρει το Qt framework, το οποίο χειρίζεται τα νήματα με πιο αποτελεσματικό τρόπο ως προς την εφαρμογή.

Στην εικόνα της προηγούμενης σελίδας μπορείτε να δείτε στο παρασκήνιο το πρόγραμμα του Swansea και μια πρώτη έκδοση της εφαρμογής DDMT – Viewer. Στο πρώτο παράθυρο (αριστερά), παρουσιάζεται η κονσόλα όπου για σκοπούς ελέγχου καταγράφονται σε πραγματικό χρόνο τα σήματα που δέχτηκε από το πρόγραμμα DDMT. Πάνω ψηλά, στην πρώτη γραμμή εμφανίζεται το “Connection connected” το οποίο υποδηλώνει πως η διεργασία του DDMT συνδέθηκε κανονικά στον server που δημιουργήσαμε. Στο δεύτερο παράθυρο (δεξιά) εμφανίζεται η διαπροσωπεία της εφαρμογής η οποία περιέχει τις ελάχιστες επιλογές. Ένα κουμπί και ένα χώρο όπου εμφανίζονται εικόνες. Για σκοπούς οπτικού ελέγχου και μόνο, εκτός από την κονσόλα αποφασίσαμε όπως παρουσιάσουμε μερικές εικόνες ανάλογα με τα σήματα τα οποία δεχόταν το πρόγραμμα. Παράλληλα, ήταν μια καλή προετοιμασία για το

επόμενο βήμα, που ήταν η υλοποίηση μιας τρίτης εφαρμογής η οποία θα ήταν υπεύθυνη για την απεικόνιση της κίνησης.

SketchFab integration

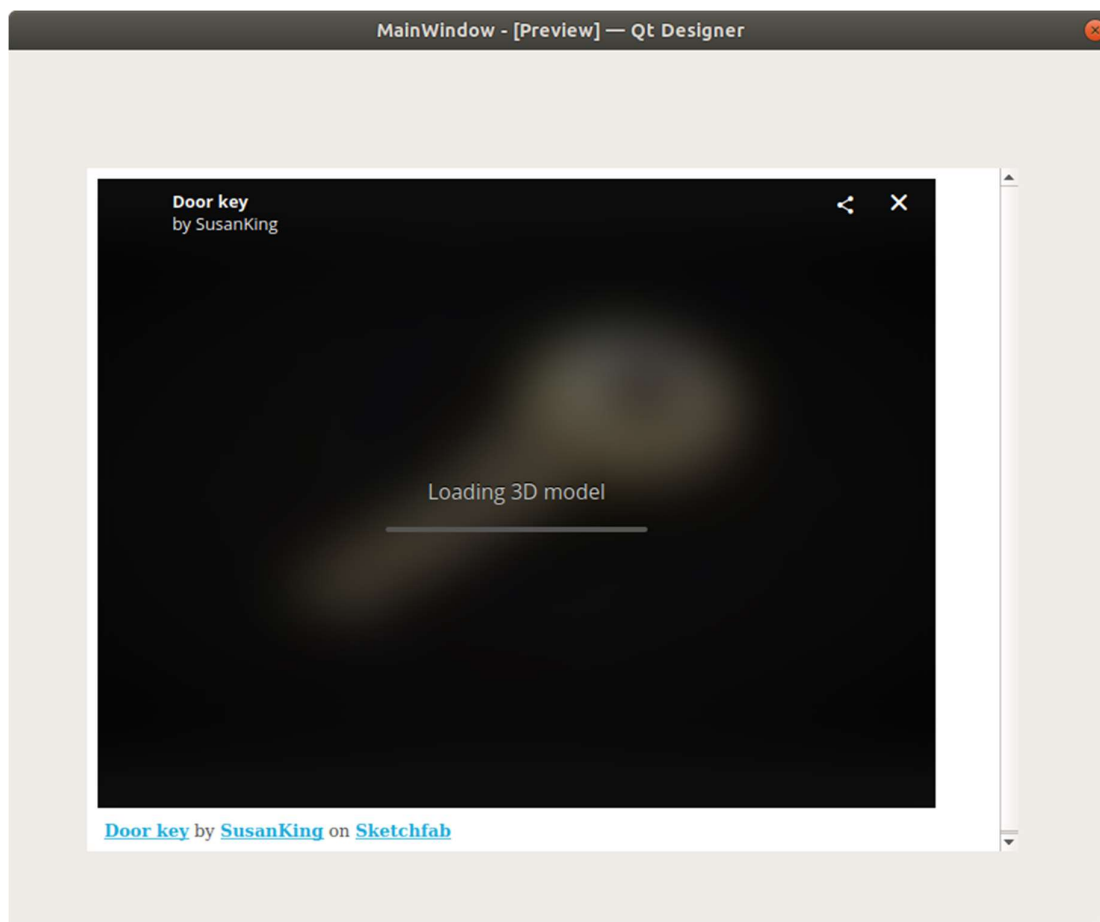
Σκοπός της εφαρμογής όμως δεν είναι η εμφάνιση στατικών εικόνων αλλά η εμφάνιση των τρισδιάστατων μοντέλων και των κινήσεων τους σε ένα πλαίσιο που επιτρέπει την αλληλεπίδραση με το μοντέλο. Η πλατφόρμα SketchFab, δίνει την δυνατότητα παρουσίασης των μοντέλων που είναι αποθηκευμένα στην βάση δεδομένων τους, σε εξωτερικές εφαρμογές. Έτσι αποφασίσαμε όπως ανεβάσουμε τα μοντέλα στην πλατφόρμα sketchfab και στην συνέχεια να παρουσιάσουμε τα μοντέλα χρησιμοποιώντας τα εργαλεία της πλατφόρμας.

Πως το αξιοποιήσαμε;

Η πλατφόρμα SketchFab δίνει ένα μικρό κομμάτι “frame” σε html κώδικα το οποίο μπορεί να εισάξει κάποιος στην ιστοσελίδα του και να μπορεί να παρουσιάσει στην ιστοσελίδα του το μοντέλο που ανέβασε καθώς και μερικά εργαλεία αλληλεπίδρασης. Όπως φαίνεται και στην εικόνα, το SketchFab δίνει ένα πλαίσιο αλληλεπίδρασης.

Το framework Qt δίνει την δυνατότητα ενσωμάτωσης φυλλομετρητή (browser). Ο φυλλομετρητής μπορεί να ορισθεί έτσι ώστε να παρουσιάζει είτε μια σελίδα .html η οποία είναι αποθηκευμένη τοπικά, είτε την παρουσίαση μιας ιστοσελίδας η οποία είναι διαθέσιμη στο διαδίκτυο. Έτσι κι εμείς αποφασίσαμε όπως δημιουργήσουμε μια μικρή ιστοσελίδα αποθηκευμένη τοπικά η οποία θα περιλαμβάνει αυτό το πλαίσιο αλληλεπίδρασης που προσφέρεται από την πλατφόρμα sketchFab.

Πιο κάτω, στην εικόνα 29 διαφαίνεται ένα μοντέλο το οποίο ανέβηκε στο sketchFab και εμφανίζεται σε ένα αρχείο τύπου html (html document), το οποίο προσθήσαμε σε μια εφαρμογή χτισμένη με Qt framework.



Εικόνα 29 Εμφάνιση μοντέλου από sketchfab σε Qt application

Μειονεκτήματα

Παρόλο που το πλαίσιο που δίνεται από το sketchfab είναι ό,τι χρειαζόμαστε, εντούτοις η ταχύτητα με την οποία το επιτυγχάνει αυτό δεν είναι ιδανική. Πρώτα απ' όλα το σύστημα αυτό θα στηρίζεται σε μεγάλο βαθμό από την σύνδεση στο διαδίκτυο. Εάν δεν υπάρχει σύνδεση στο διαδίκτυο τότε το σύστημα αυτό δεν θα μπορεί να χρησιμοποιηθεί. Ως επίσης, πάντα θα υπάρχει μια μικρή καθυστέρηση ασχέτως της ποιότητας της σύνδεσης του διαδικτύου καθώς η επικοινωνία του προγράμματος με την πλατφόρμα της sketchfab είναι υπεράκτια.

Ακόμη όμως κι αν όλα αυτά δεν επηρεάζουν αρνητικά τον βιολόγο που θα χρησιμοποιήσει αυτό το σύστημα, η εφαρμογή βασίζεται σε τριτογενείς παράγοντες οι οποίοι εάν αλλάξουν στον μέλλον ή σταματήσουν να παρέχουν τις υπηρεσίες τους, τότε το σύστημα θα είναι άχρηστο.

Autodesk FBX SDK integration

Εφ' όσων τα αρχεία μας είναι στο format FBX, το οποίο δημιούργησε η εταιρία Autodesk για τις εφαρμογές της, το πρώτο μας βήμα ήταν να ανατρέξουμε στην σελίδα της και να

μελετήσουμε τα APIs που προσφέρονται για κάτι τέτοιο. Αυτό το οποίο βρήκαμε, ήταν ένα SDK (software development kit) το οποίο υποστηρίζει τις γλώσσες C++ και Python.

Στα εκπαιδευτικά μαθήματα της Autodesk υπάρχουν πολλά παραδείγματα για την δημιουργία μιας σκηνής, δημιουργία τρισδιάστατων αντικειμένων στον χώρο προγραμματιστικά, κινήσεις (animations) και πολλά άλλα. Παρόλα αυτά τα περισσότερα μαθήματα αφορούσαν την εφαρμογή των πιο πάνω στη γλώσσα C++. Όσο αφορά την python δεν υπήρχαν πολλά μαθήματα αλλά υπήρχε μια λίστα η οποία για κάθε κλάση της C++, δίνεται η αντίστοιχη ονομασία στη Python. Έτσι με λίγη προσοχή θεωρητικά θα ήταν εφικτή η “μετάφραση” των παραδειγμάτων που δίδονται από την Autodesk σε C++, στη γλώσσα Python που χρησιμοποιούμε.

Μειονεκτήματα

Όπως παρατηρήθηκε μετά από ψάξιμο στο documentation της Autodesk αλλά και στο εκπαιδευτικό υλικό που δίνεται από το κοινό(community resources), ήταν ξεκάθαρο πως υπήρχε μεγαλύτερη έμφαση στη γλώσσα C++ ενώ αυτά που αφορούσαν την Python ήταν outdated και obsolete.

Οι τελευταίες κλάσεις που δίνονταν σε python(δλδ το τελευταίο python sdk), ήταν για την έκδοση 3.3 ενώ εμείς χρησιμοποιήσαμε την έκδοση της Python 3.8! Για να μπορέσει να λειτουργήσει το πρόγραμμα μας με βάσει τις προδιαγραφές στο documentation, θα έπρεπε είτε να αναθεωρήσουμε ολόκληρο τον κώδικο για την έκδοση 3.3 είτε να δημιουργήσουμε τα δικά μας bindings από C++ κώδικα σε python 3.8. Κάτι τέτοιο χρειάζεται χρόνο, μελέτη και καλή κατανόηση και των δύο γλωσσών σε ένα πιο βαθύ επίπεδο.

Στο documentation της Autodesk όμως δεν ήταν το μοναδικό πρόβλημα που εντοπίσαμε. Η λίστα με τις κλάσεις και τις αντίστοιχες μεθόδους που μπορούν να χρησιμοποιηθούν σε python, δεν μεταφράζονται με τις ίδιες παραμέτρους. Για παράδειγμα, ενώ μια συνάρτηση στην C++ μπορεί ήταν τύπου FBXObject και να δέχεται 3 παραμέτρους, η ίδια συνάρτηση στην Python μπορεί να περιείχε 2 παραμέτρους.

Όλα αυτά ενώ θεωρητικά δίνουν τα εφόδια για υλοποίηση του συστήματος, δηλαδή να μπορέσει να γίνει ένα σωστό FBX integration με τα εργαλεία της εταιρείας που δημιούργησε τον συγκεκριμένο τύπο αρχείου, εντούτοις όμως όπως συμπεράναμε μετά την έρευνα που έγινε, ήταν προφανές πως μια τέτοια υλοποίηση θα ήταν δύσκολη και αρκετά χρονοβόρα. Ο συνδυασμός των πιο πάνω, μας ώθησε στο ψάξιμο κάποιας άλλης λύσης

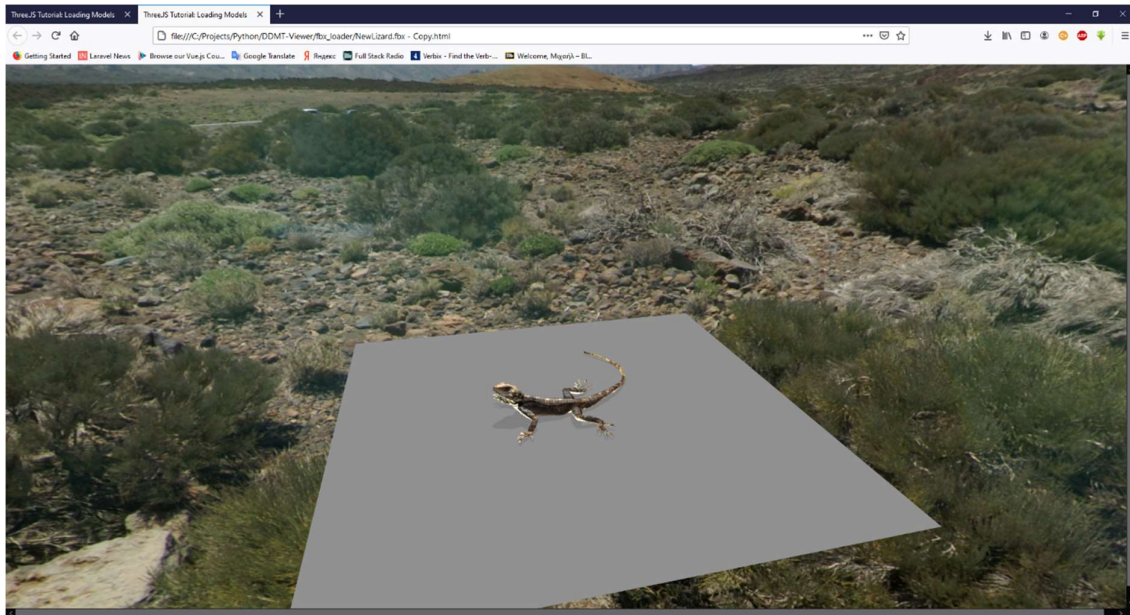
Three.js library integration

Η βιβλιοθήκη three.js [5] [6] είναι μια βιβλιοθήκη γραμμένη σε JavaScript από τον Ricardo Cabello η οποία μπορεί να χρησιμοποιηθεί σε όλους τους φυλλομετρητές για την παρουσίαση τρισδιάστατων γραφικών. Η three.js χρησιμοποιεί WebGL, μια άλλη βιβλιοθήκη γραμμένη σε JavaScript η οποία εμφανίστηκε για πρώτη φορά στην Mozilla από τον Vladimir Vukićević [7].

Η συγκεκριμένη βιβλιοθήκη περιείχε πολλά έτοιμα παραδείγματα για την παρουσίαση μοντέλων(συμπεριλαμβανομένου και για FBX), τα οποία δεν ήταν δύσκολο να εντάξουμε στο δικό μας πρόγραμμα. Η διαδικασία με την οποία το υλοποιήσαμε αυτό ήταν με τον ίδιο ακριβώς τρόπο που εντάξαμε το SketchFab. Η διαφορά ήταν πως αντί να φορτώνουμε μία ιστοσελίδα η οποία βρίσκεται στο διαδίκτυο, τώρα θα φορτώνουμε μια σελίδα η οποία θα βρίσκεται τοπικά.

Δημιουργήσαμε δηλαδή μια σελίδα html η οποία περιείχε σε ένα tag, την ονομασία του FBX αρχείου που θέλαμε να προβάλουμε, και στο τέλος, φορτώσαμε το αρχείο *main.js* το οποίο είναι υπεύθυνο για το rendering.

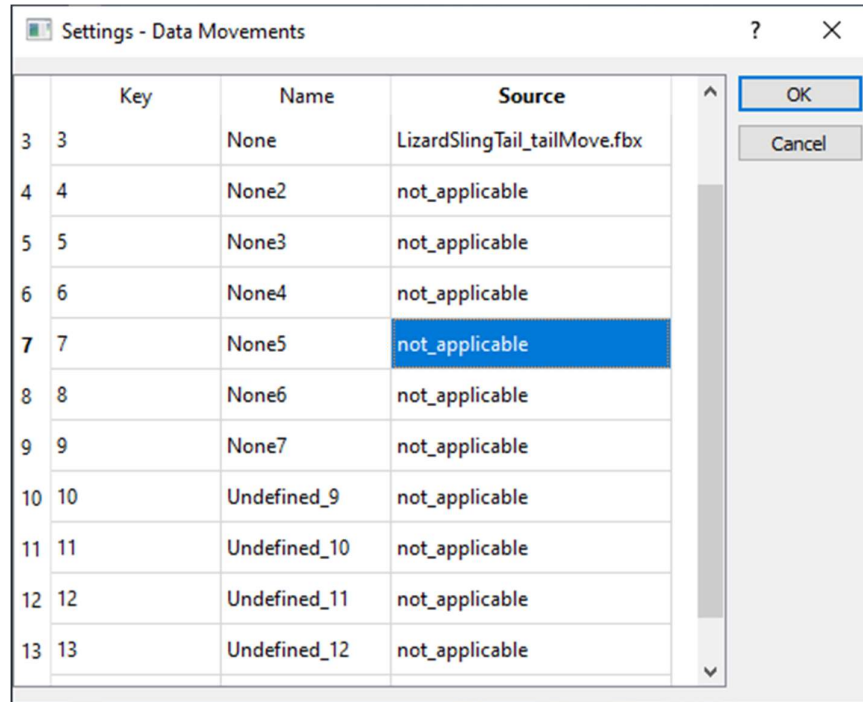
Το αποτέλεσμα της σελίδας που δημιουργήσαμε αρχικά ήταν το πιο κάτω.



Εικόνα 30 Three.js integration - Initial html page

Αφού έγινε αυτό, τώρα έπρεπε να βρεθεί κάποιος τρόπος αυτή η διαδικασία από χειρονακτική να γίνει αυτοματοποιημένη. Ας μη ξεχνάμε, πως μια από τις βασικές προδιαγραφές που θέσαμε αρχικά, ήταν όπως ο χρήστης να μην χρειάζεται την βοήθεια του προγραμματιστή για απλά ζητήματα όπως αλλαγή αρχείου, μοντέλου κα.

Αυτό το οποίο έγινε, ήταν όπως κάθε φορά που γίνεται αλλαγή στις ρυθμίσεις των κινήσεων(settings movements, βλ εικόνα), να γίνεται η δημιουργία των html σελίδων όπου μέσα θα αναγράφεται το όνομα της κίνησης που θέλουμε να φορτώσουμε. Έτσι κάθε φορά που το πρόγραμμα έπρεπε να εμφανίσει στην οθόνη κάποια κίνηση, το μόνο που έπρεπε να κάνουμε ήταν να αλλάζουμε την html σελίδα.



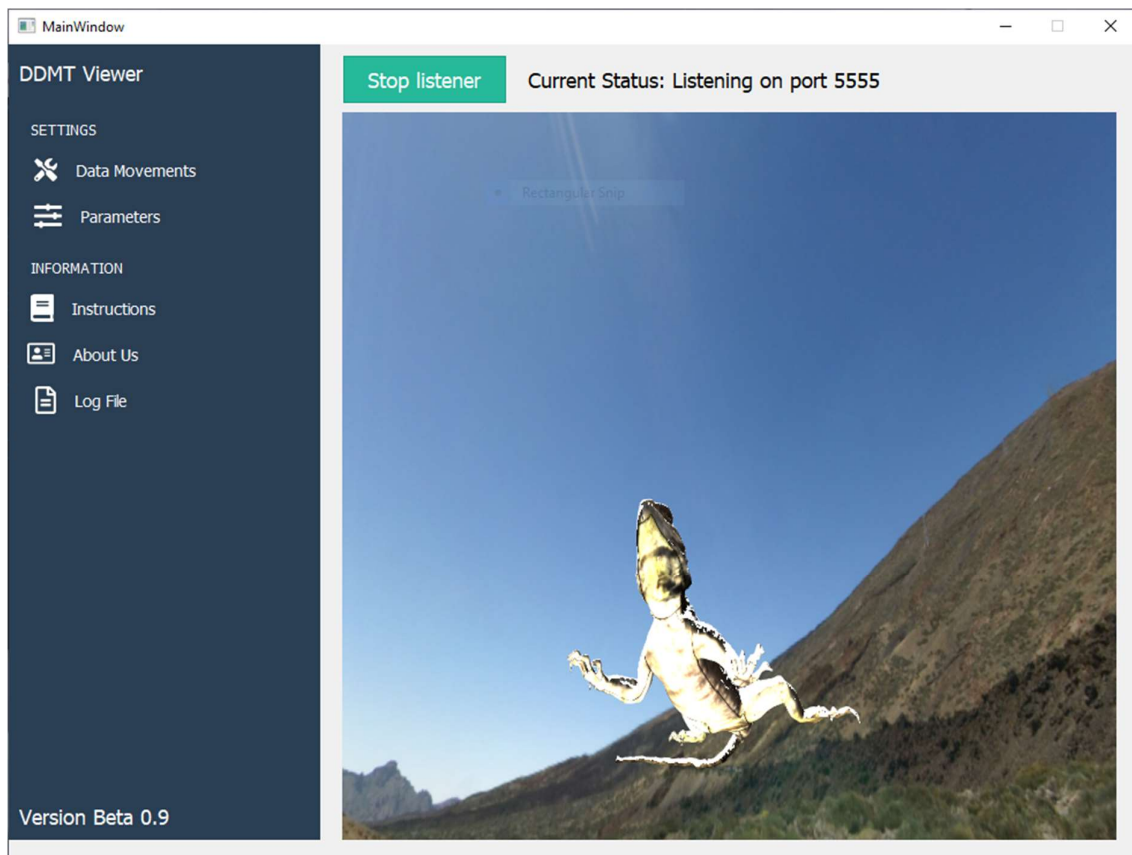
Εικόνα 31 Settings - Data movements with FBX source

```
fbx_loader/LizardSlingTail_tailMove.fbx.html
fbx_loader/LizardSlingTail_tailMove.fbx.html
fbx_loader/LizardSlingTail_tailMove.fbx.html
fbx_loader/not_applicable.html
fbx_loader/not_applicable.html
fbx_loader/not_applicable.html
fbx_loader/not_applicable.html
fbx_loader/not_applicable.html
fbx_loader/not_applicable.html
fbx_loader/not_applicable.html
fbx_loader/not_applicable.html
fbx_loader/not_applicable.html
fbx_loader/not_applicable.html
fbx_loader/not_applicable.html
```

Εικόνα 32 Console output - Δημιουργία των html σελίδων δυναμικά

4.4 Έλεγχος, αναθεώρηση και optimizations

Σε γενικές γραμμές, με την βιβλιοθήκη three.js τα αποτελέσματα ήταν ικανοποιητικά από τις πρώτες προσπάθειες ωστόσο υπήρχαν κάποια σημεία στα οποία έπρεπε να δοθεί προσοχή.



Εικόνα 33 Πρώτη έκδοση με το three.js integration

1. Όλοι οι φυλλομετρητές σήμερα περιέχουν αρκετά πρωτόκολλα ασφαλείας από επιθέσεις. Κάποιες από τις εντολές της JavaScript, δεν μπορούν να χρησιμοποιηθούν στο client-side για αποφυγή επιθέσεων όπως XSS, Cross origin policy κ.α. Στην περίπτωση μας η βιβλιοθήκη που χρησιμοποιήσαμε, η three.js, φορτώνει τα αρχεία με τέτοιο τρόπο όπου πολλοί φυλλομετρητές δεν επιτρέπουν την εκτέλεση ορισμένων συναρτήσεων με αποτέλεσμα να χρειάζονται ειδικές ρυθμίσεις για να μπορέσει να δουλέψει κάτι τέτοιο.
 - a. Για να λυθεί αυτό έπρεπε τα αρχεία αυτά να τρέξουν στο server-side. Έτσι έπρεπε να δημιουργήσουμε κάτι σαν μικρο-server.
 - b. Παράλληλα έπρεπε να κρατηθεί ένα απλό για τον χρήστη περιβάλλον.
 - c. Έτσι αποφασίσαμε όπως δημιουργήσουμε ένα πολύ μικρό script, το οποίο να μπορεί να τρέξει ο χρήστης, το οποίο το μόνο που θα έκανε θα ήταν δημιουργήσει ένα server, δηλαδή να ανοίξει ένα καινούργιο socket όπως προηγουμένως αλλά αυτή τη φορά στο port 8080.
 - d. Το πρόβλημα επιδιορθώθηκε και ο χρήστης δεν φάνηκε να δυσκολεύεται με αυτή την αλλαγή

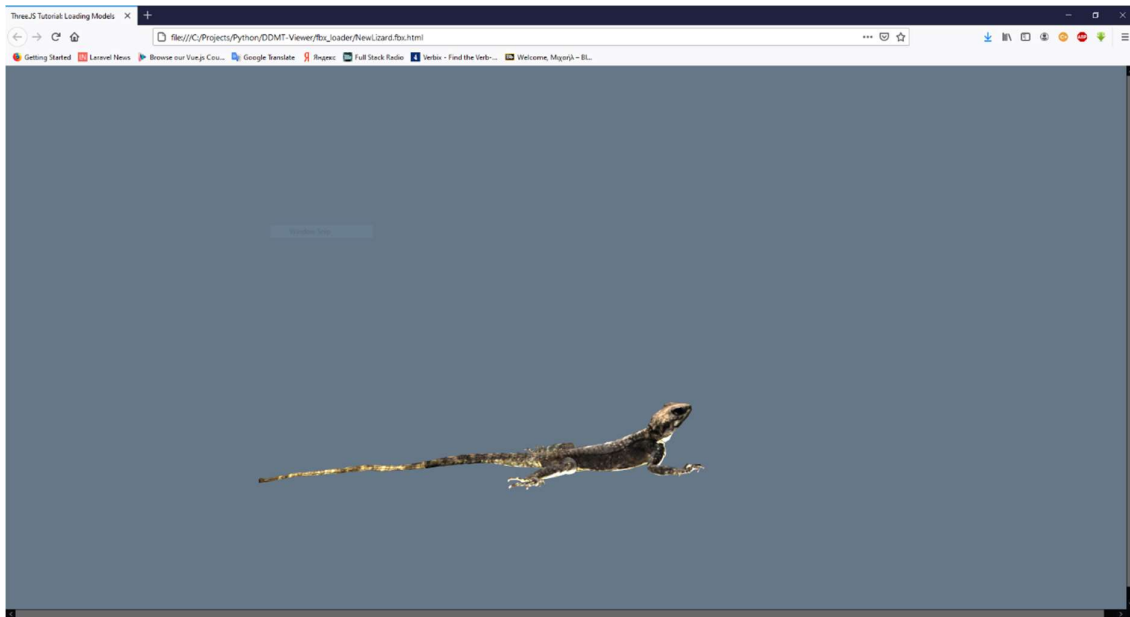
2. Επειδή έχουμε να κάνουμε με επεξεργασία γραφικών μοντέλων, έπρεπε να είμαστε σίγουροι πως φορτώνουμε με τον πιο βέλτιστο τρόπο τα δεδομένα μας. Μετά από συζήτηση σκεφτήκαμε ποιες θα ήταν οι πιθανές βελτιστοποιήσεις που θα μπορούσαμε να κάνουμε, και καταλήξαμε στις ακόλουθες επιλογές. Για κάθε επιλογή σκεφτήκαμε και τα πλεονεκτήματα/μειονεκτήματα που πιθανόν να υπάρξουν
- a. Αντί να φορτώνεται το σωστό μοντέλο στην οθόνη και σε κάθε αλλαγή, να αλλάζει το αρχείο που εμφανίζεται, ας φορτώσουν όλα τα μοντέλα στην αρχή και να είναι “ορατό” μόνο το μοντέλο που πρέπει
 - i. Κάτι τέτοιο θα αποτελεί πρόβλημα στην αρχική εκκίνηση του προγράμματος καθώς θα είναι πιο χρονοβόρο
 - ii. Θα απαιτεί μεγαλύτερη μνήμη RAM καθώς θα είναι σαν να έχουμε πολλά tabs ανοικτά σε κάποιο φυλλομετρητή.
 - b. Να γίνει αφαίρεση πολλών λεπτομερειών που υπάρχουν στην απεικόνιση της κίνησης.
 - i. Εμφανισιακά δεν θα είναι το ίδιο καθώς δεν θα υπάρχουν τόσες πολλές λεπτομέρειες
 - ii. Θα γίνει σίγουρα πιο γρήγορο το rendering

Η πρώτη αλλαγή λοιπόν ήταν η δημιουργία ενός μικρού script η οποία θα ξεκινούσε ένα μικρό server. Πιο κάτω μπορείτε να δείτε τις εντολές που τοποθετήσαμε σε ένα batch file με την ονομασία start_server.bat.

```
1 cd fbx_loader
2 echo Starting Http Server
3 py -m http.server
```

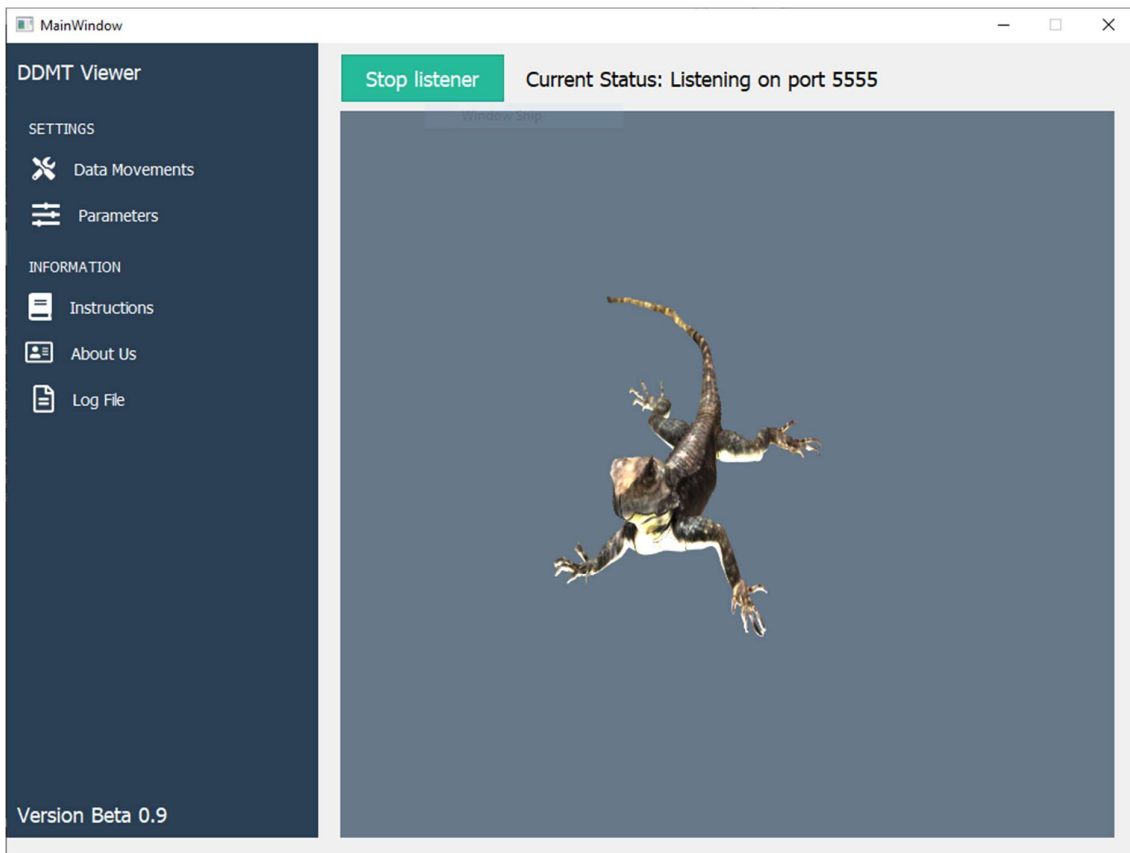
Εικόνα 34 Εντολές για έναρξη του mini-server

Η δεύτερη αλλαγή ήταν λοιπόν η αφαίρεση πολλών λεπτομερειών στο παρασκήνιο. Η τελική σελίδα φαινόταν πλέον ως εξής:



Εικόνα 35 Σελίδα html χωρίς λεπτομέρειες

4.5 Τελικό αποτέλεσμα, σφάλματα και συμπεράσματα



Εικόνα 36 Τελικό αποτέλεσμα

Όπως φαίνεται και στην εικόνα, το τελικό αποτέλεσμα είναι ρυθμισμένο να επικοινωνεί με το πρόγραμμα DDMT στο port 5555 και τα δεδομένα εμφανίζονται στην οθόνη χωρίς λεπτομέρειες για γρήγορες αλλαγές στην κίνηση. Ως επίσης το πρόγραμμα υποστηρίζει όλες τις προδιαγραφές που θέσαμε αρχικά και ίσως μερικές επιπλέον λειτουργίες που δεν ορίσαμε

στην αρχή. Για παράδειγμα, έχουμε τοποθετήσει το Log file το οποίο καταγράφει τις κινήσεις που υπήρξαν.

Το πρόγραμμα αρχικά ξεκίνησε ως ένα μέσω επικοινωνίας χρησιμοποιώντας socket programming σε python, αλλά στη συνέχεια προσπαθήσαμε σε ό,τι αφορά την γραφική απεικόνιση, να χρησιμοποιηθούν τεχνολογίες οι οποίες να μπορούν να επικοινωνούν με το λειτουργικό σύστημα(OS) αντί να γίνονται χρήσεις τεχνολογιών διαδικτύου. Παρόλα αυτά, όπως φάνηκε και στην πορεία ήταν πιο εύκολο να χρησιμοποιηθούν τεχνολογίες διαδικτύου και το υλικό που βρήκαμε ήταν περισσότερο και πιο ανανεωμένο.

Τέλος, χρησιμοποιήθηκε σε πολλές περιοχές παράλληλη επεξεργασία χρησιμοποιώντας πολυνηματικές βιβλιοθήκες παρόλα αυτά, οι πολυνηματικές βιβλιοθήκες δεν χρησιμοποιήθηκαν για βελτίωση της απόδοσης του προγράμματος αλλά για την ομαλή σύνδεση τεχνολογιών διαδικτύου και του υπόλοιπου προγράμματος.

Τελική δομή προγράμματος:

- Config – Used by the system
 - Constant
 - Movements
- Icons – Εικονίδια στο GUI του προγράμματος
- Fbx_loader – Υπεύθυνο για το loading των μοντέλων
 - Resources
 - Custom – Η τοποθεσία των μοντέλων
- Views – Used by the system
- Main.exe – Το κυρίως πρόγραμμα
- Start_server.bat – Έναρξη server

4.6 Μελλοντικά σχέδια

Η πρώτη βασική αλλαγή που θα μπορούσε να γίνει είναι όπως να τρέχουν τα πάντα μέσω του κυρίως προγράμματος. Για παράδειγμα το script που δημιουργήσαμε θα ήταν πιο εύκολο για τον χρήστη, εάν έτρεχε δια μέσω του προγράμματος που υλοποιήσαμε.

Όπως είδαμε και στις προηγούμενες ενότητες όμως, έγινε χρήση τεχνολογιών διαδικτύου για την υλοποίηση ενός **Desktop Application**. Αν και θέλαμε να αποφύγουμε κάτι τέτοιο, στο

τέλος της ημέρας αποδείχτηκε η ιδανικότερη λύση. Επομένως, στα μελλοντικά σχέδια είναι ένας καλύτερος σχεδιασμός αυτού του προγράμματος με πρωταρχικό στόχο την υλοποίηση ενός Desktop application χρησιμοποιώντας κατά κόρον τεχνολογίες διαδικτύου. Κάτι τέτοιο υλοποιήσαμε κι εμείς αλλά έχοντας στο μυαλό μας το ανάποδο. Στην κοινότητα των προγραμματιστών υπάρχει μια ιδιαίτερη προσέγγιση αυτού του θέματος την οποία θα θέλαμε να χρησιμοποιήσουμε όταν δωθεί το πράσινο φως για ανανέωση του υφιστάμενου προγράμματος.

Η νέα προσέγγιση του θέματος θα ήταν χρησιμοποιώντας το framework Electron [5] το οποίο είναι ένας συνδυασμός server + web browser ως desktop application. Κάτι τέτοιο χρησιμοποιεί τις τεχνολογίες JavaScript, Node.js [6] και Chromium ως web browser. Στο τέλος της διπλωματικής μου εργασίας δημιουργήσαμε κάτι παρόμοιο. Δηλαδή δημιουργήσαμε ένα πρόγραμμα το οποίο χρησιμοποιεί python ως server αντί node.js, qt browser αντί chromium και τις γλώσσες javascript και python. Το πλεονέκτημα της χρήσης του Electron όμως είναι πως υπάρχουν ήδη πολλές υλοποιημένες συναρτήσεις για την επικοινωνία του web browser με το server-side κομμάτι του κώδικα, το οποίο είναι κάτι που δεν υλοποιήσαμε.

Κεφάλαιο 5

Ανάλυση των δεδομένων και εξαγωγή κινήσεων

5.1 Διαχωρισμός δεδομένων από βιολόγο	46
5.2 Συσχέτιση χρόνων και frames	47
5.3 Αποκοπή κινήσεων στο Motion Builder	48

5.1 Διαχωρισμός δεδομένων από βιολόγο

Σε αυτό το σημείο πλέον ο Δρ. Σαββας Ζώτος, που είναι και ο βιολόγος με τον οποίο συνεργαστήκαμε, ήταν σε θέση να αναλύσει τα βίντεο που καταγράφηκαν. Ο Δρ. Ζώτος, για κάθε βίντεο, βρήκε τις κινήσεις που έγιναν στον χώρο, και κατέγραψε χειρακτικά τους χρόνους. Αυτή η μέθοδος ονομάζεται “labeling”. Δηλαδή για κάθε στιγμιότυπο που μας ήταν ενδιαφέρον, δηλαδή χρήσιμο για περαιτέρω ανάλυση, ο Δρ. Ζώτος έγραψε το όνομα της κίνησης.

Οι κινήσεις που ορίσθηκαν για τη σαύρα:

- Running slow
- Turning right
- Turning left
- Try to climb on glossy surface
- Running
- Eating insect
- Fast run short
- Walk
- Jump on log
- Jump off the log
- Rapid turn left
- Rapid turn right
- Climb log
- Descent from log
- Run log to log
- Walk stop walk stop
- Rolling head left
- Rolling head right
- Walk upward on slope
- Pitch head up (slight)
- Climb vertical on log

Οι κινήσεις που ορίσθηκαν για το φίδι:

- Throw forward
- Slow slithering (along a specific direction)

- Head right/left turn
- Head up/down on wall
- Move right/left 180 degrees
- Head defensive retraction
- Head right/left/up/down turn 90/180 degrees
- Head shift
- Head small straight forward
- Rapid small head movements
- Slow straight slithering of first half
- Head rapid heave up/down
- Change slithering direction
- Rapid forward movement
- Raising head to climb wall
- Right/left/up/down sway
- Rapid defense retraction
- Short biting attack and retraction
- Slow slithering along corner
- Turn right/left
- Short attack
- Short attack to scare with retraction
- Short head retraction
- Left/right yaw
- Forward extension
- Rapid forward extension
- Open right turn (yaw + sway) of head
- Eject half first forward
- Large head sway

5.2 Συσχέτιση χρόνων και frames

Ο Δρ. Ζώτος συσχέτισε τις κινήσεις με τους χρόνους. Παρόλα αυτά, για να μπορέσουμε να επεξεργαστούμε τα δεδομένα ψηφιακά, έπρεπε να συσχετίσουμε τους χρόνους με τα frames. Πιο κάτω μπορείτε να δείτε ένα πίνακα όπου συμπληρώθηκε αρχικά από τον Δρ. Ζώτο(rgb time in, duration and behaviour) και στη συνέχεια συμπληρώσαμε εμείς με τα αντίστοιχα frames (motives time in).

RGB time in	Motives time in	Duration (sec)	Behaviour
15.04.34.00	4332 - 4440	1	Running slow
15.04.37.00	4670 - 4777	1	Turning right
15.04.38.00	4777 - 4903	1	Running slow
15.04.41.00	5063 - 5151	1	Turning right
15.04.48.00	5699 - 5799	1	Turning left
15.04.54.00	6300 - 6432	1	Running slow
15.04.57.00	6648 - 6790	1	Running slow
15.05.13.00		1	Try to climb on glossy surface
15.06.10.00		5	Try to climb on glossy surface
15.06.20.00	14949 - 15160	3	Running
15.11.36.00	43309	74	Eating insect

* Με κόκκινο χρώμα σημαίνει πως οι markers χάνονται

** Με πράσινο χρώμα σημαίνει πως υπάρχουν οι markers αλλά δεν φαίνεται αλλαγή κίνησης(π.χ. η σαύρα τρώει χωρίς να κουνάει το κεφάλι της)

Για να γίνει κάτι τέτοιο έπρεπε να βλέπουμε παράλληλα το βίντεο καταγραφής και στο κάθε RGB time in χρόνο να επαληθεύουμε την κίνηση. Παράλληλα, έχοντας ανοικτό το πρόγραμμα Motion Builder ή Motive, ορίσαμε την εμφάνιση των frames για ένα c3d αρχείο στα 100fps και μπορούσαμε να καταγράψουμε τα ακριβή frames των κινήσεων.

Η καταγραφή των markers δεν ήταν η ιδανική εξού και σε κάποια σημεία χάθηκαν. Ο λόγος ήταν κυρίως διότι η σαύρα ή το φίδι απομακρύνθηκε από τον χώρο όπου έγραφε η κάμερα. Δεν ήταν πολλά τα σημεία όπου χάθηκαν οι markers αλλά τα σημεία όπου έγινε αυτό καταγράφηκαν με κόκκινο χρώμα. Στη σπάνια περίπτωση όπου το ερπετό κινήθηκε αλλά δεν ήταν εμφανής η διαφορά στους markers, ορίστηκε με πράσινο χρώμα.

Η πιο κάτω εικόνα παρουσιάζει το βίντεο που χρησιμοποιήσαμε εμείς για να μπορέσουμε να συσχετίσουμε τις κινήσεις στο c3d αρχείο. Διαφαίνεται ο χρόνος κάτω αριστερά (RGB time in) και μια πιο κοντινή προβολή της σαύρας.



Εικόνα 37 Συσχετισμός κινήσεων και frames - Original video

5.3 Αποκοπή κινήσεων στο Motion Builder

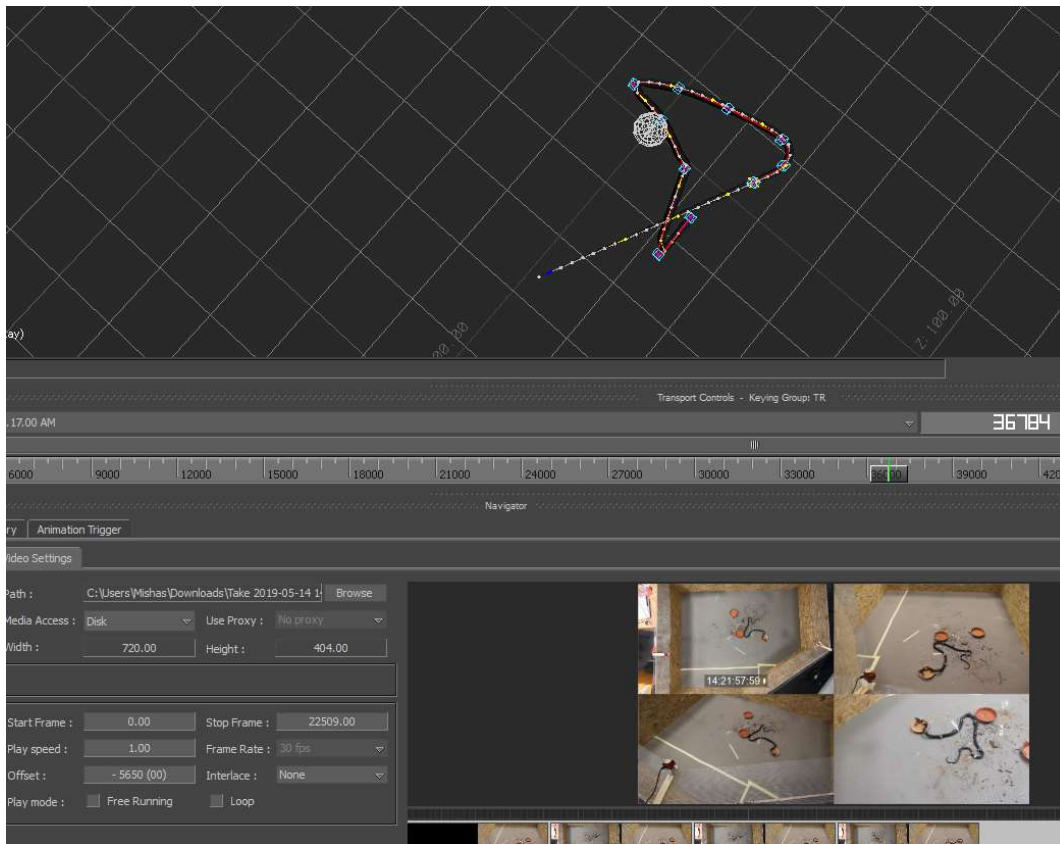
Για να μπορέσει να αποκοπεί και να αποθηκευτεί ξεχωριστά η κάθε κίνηση απαραίτητο ήταν όπως πρώτα φορτωθεί το διορθωμένο αρχείο c3d με το fbx μοντέλο (βλ. Κεφάλαιο 3).

Ακολουθώς, σε κάποια αρχεία προτιμήσαμε όπως συσχετίσουμε το βίντεο που

χρησιμοποιήσαμε προηγουμένως με το αρχείο που φορτώσαμε έτσι ώστε να είμαστε σίγουροι πως κινήθηκε το φίδι σε πραγματικό περιβάλλον και πως κινήθηκε το φίδι στο αρχείο μας. Έτσι αποφεύγονται τα όποια πιθανά λάθη μπορεί να υπήρξαν στη «συσχέτιση χρόνων και frames» που εξηγείτε πιο πάνω.

Εισαγωγή βίντεο και συγχρονισμός με το αρχείο που ετοιμάσαμε

Η διαδικασία εισαγωγής του βίντεο απαιτούσε από εμάς όπως πρώτα επιλέξουμε File -> Import video, ενώ στη συνέχεια έπρεπε να συσχετίσουμε το βίντεο με το αρχείο μας έτσι ώστε να συμβαδίζουν. Δηλαδή, καθώς κινείται το φίδι που ετοιμάσαμε, η ίδια κίνηση να παρουσιάζεται και στο βίντεο. Γι' αυτό έπρεπε όπως βρούμε με το μάτι κάποια χαρακτηριστική κίνηση (π.χ απότομη κίνηση φιδιού, επίθεση ερπετού, κτλ) και να υπολογίσουμε την διαφορά που υπάρχει μεταξύ των frames.



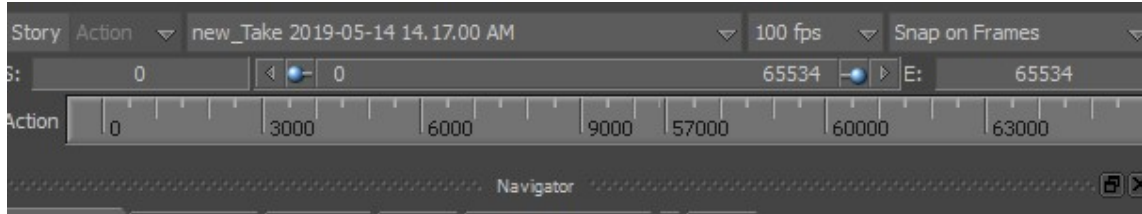
Εικόνα 38 Applying offset to imported video

Για παράδειγμα, εάν το φίδι που ετοιμάσαμε στο frame 320 επιτέθηκε, πρέπει να βρούμε σε ποιο frame του βίντεο παρουσιάζεται αυτό. Έστω πως αυτό έγινε 200 frames μετά, τότε αλλάζουμε το offset του βίντεο έτσι ώστε να μην υπάρχει αυτή η διαφορά. (Στην εικόνα το offset που επέλεξα για αυτό το βίντεο είναι -5650)

Αποκοπή κίνησης και αποθήκευσης

Κάτω από το “Display”, υπάρχει το toolbar (μενού επιλογών), “Story” στο οποίο μπορούμε να περιηγηθούμε στο πλάνο αλλά και να αποκόψουμε ένα μέρος του αρχείου.

Χρησιμοποιώντας τις υποδείξεις «Start» και «End» που συμβολίζονται με τα ονόματα “S” και “E” αντίστοιχα επιλέξαμε από ποιο frame αρχίζει μια κίνηση και σε ποιο frame τελειώνει.



Εικόνα 39 Αποκοπή κίνησης χρησιμοποιώντας τους δείκτες Start και End

Κεφάλαιο 6

Δημιουργία ιστοσελίδας για προβολή των κινήσεων και άλλων πληροφοριών

6.1 Ανασκόπηση	51
6.2 Τεχνολογίες ανάπτυξης ιστοσελίδας	51
6.3 Παρουσίαση ιστοσελίδας	54

6.1 Ανασκόπηση

Όπως και σε κάθε άλλη εφαρμογή που αναπτύξαμε, έτσι και σε αυτό το έργο ακολουθήσαμε πιστά τον κύκλο της ανάπτυξης λογισμικού, κοινώς με την αγγλική ορολογία *Software engineering cycle*. Επιπρόσθετα, ακολουθήσαμε την μεθοδολογία Agile [5] [6]. όπου δηλαδή ο ίδιος κύκλος επαναλαμβάνονταν για μικρά κομμάτια της ιστοσελίδας καταλήγοντας σε ανατροφοδότηση και αλλαγές βάσει αυτής.

Σκοπός της σελίδας ήταν να παρέχει της απαραίτητες πληροφορίες σχετικά με το έργο που επιτελέσαμε μέχρι σήμερα, καθώς επίσης να μπορεί κανείς να δει ζωντανά τις κινήσεις που δημιουργήσαμε αλλά και μερικές άλλες πληροφορίες που αφορούν τον τρόπο με τον οποία εργαστήκαμε, τα αποτελέσματα που πετύχαμε, δημοσιεύσεις σχετικά με το έργο καθώς και μερικές πληροφορίες που αφορούν τα ερπετά με τα οποία εργαστήκαμε. Στο παρών κεφάλαιο, γίνεται κυρίως η παρουσίαση της ιστοσελίδας.

6.2 Εργαλεία ανάπτυξης ιστοσελίδας

Για την ανάπτυξη της ιστοσελίδας έγινε η χρήση των παρακάτω εργαλείων.

HTML

Η HTML είναι το ακρωνύμιο των λέξεων **HyperText Markup Language** που στα ελληνικά σημαίνει **Γλώσσα Σήμανσης Υπερκειμένου**. Βασίζεται στη γλώσσα SGML, Standard Generalized Markup Language, που είναι ένα πολύ μεγαλύτερο σύστημα επεξεργασίας εγγράφων και είναι η βασική γλώσσα με την οποία πραγματοποιείται η δόμηση σελίδων του Παγκόσμιου Ιστού. Η HTML αποτελείται από ένα σύνολο κοινών στυλ για τις

Web σελίδες, όπως τίτλοι (titles), επικεφαλίδες (headings), παράγραφοι (paragraphs), λίστες (lists) και πίνακες (tables). Κάθε στοιχείο έχει ένα όνομα και περιέχεται μέσα στα σύμβολα < >, που αποκαλούνται ετικέτες (tags). Δεν θεωρείται γλώσσα προγραμματισμού αλλά ένας ειδικός τρόπος γραφής κειμένου.

CSS

Η CSS είναι το ακρόνυμο των λέξεων Cascading Style Sheets που στα ελληνικά σημαίνει αλληλουχίες φύλλων στυλ. Είναι μια γλώσσα υπολογιστή που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης HTML και XHTML. Καθορίζει τη εμφάνιση και μορφή μιας ιστοσελίδας και γενικότερα ενός ιστό τύπου. Συγκριτικά με την HTML η CSS προσφέρει περισσότερες δυνατότητες όσο αφορά τα στιλιστικά χαρακτηριστικά μιας ιστοσελίδας δηλαδή χρώματα, στοίχιση. Η CSS είναι απαραίτητη για την ανάπτυξη μιας εμπλουτισμένης στιλιστικά ιστοσελίδας.

Javascript

Η JavaScript είναι μια γλώσσα προγραμματισμού που ενισχύει τη δυναμική συμπεριφορά στους περισσότερους ιστότοπους [16]. Παράλληλα με το HTML και το CSS, είναι μια βασική τεχνολογία που κάνει τον ιστό να λειτουργεί. Η σύνταξη της είναι επηρεασμένη από την γλώσσα προγραμματισμού C. Αν και η Javascript αντιγράφει πολλά ονόματα και συμβάσεις ονοματολογίας από τη Java, οι δυο αυτές γλώσσες δεν σχετίζονται μεταξύ τους, αφού η Javascript είναι script language. Είναι βασισμένη, στα αντικείμενα και ο κώδικας JavaScript, συνήθως ενσωματώνεται στον HTML κώδικα.

jQuery

Το jQuery είναι μια γρήγορη, μικρή και πλούσια σε χαρακτηριστικά βιβλιοθήκη JavaScript. έχει σχεδιαστεί για να διευκολύνει την πλοήγηση ενός εγγράφου, την επιλογή στοιχείων DOM, τη δημιουργία κινούμενων σχεδίων, τη διαχείριση συμβάντων και την ανάπτυξη εφαρμογών Ajax, πολύ πιο εύκολα με ένα εύχρηστο API που λειτουργεί σε πολλά προγράμματα περιήγησης [17]. Με ένα συνδυασμό ευελιξίας και επεκτασιμότητας, το jQuery άλλαξε τον τρόπο με τον οποίο εκατομμύρια άνθρωποι γράφουν JavaScript. Το jQuery παρέχει επίσης δυνατότητες στους προγραμματιστές να δημιουργούν προσθήκες πάνω από τη βιβλιοθήκη JavaScript. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν αφαιρετικότητα για χαμηλού επιπέδου αλληλεπίδραση και animation, προηγμένα εφέ και υψηλού επιπέδου, θεματικά widget.

Alpine.js

Η Alpine.js είναι μια μικρή Javascript βιβλιοθήκη γραμμένη από τον Caleb Porzio, που έχει σχεδιαστεί ούτως ώστε ο προγραμματιστής να γράφει markup-driven client-side JavaScript. [18] Η σύνταξη της βασίζεται σε μεγάλα frameworks όπως Vue.js και React.js με την διαφορά πως το Alpine.js είναι πιο μικρό σε μέγεθος, χρειάζεται λιγότερο χρόνο εκμάθησης και είναι αρκετά γρήγορο. Χτίστηκε χρησιμοποιώντας την βιβλιοθήκη που περιγράφεται πιο πάνω (jQuery) και έτσι η χρήση αυτής της βιβλιοθήκης έναντι της jQuery είναι σαφώς πιο εύκολη.

PHP

Η PHP (PHP: Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργασθεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script. Η PHP αποτελεί μια από τις πιο διαδεδομένες τεχνολογίες στο Παγκόσμιο Ιστό, καθώς χρησιμοποιείται από πληθώρα εφαρμογών και ισότοπων. Η ευρύτητα στη χρήση της είναι απορροεή από την ευκολία που παρουσιάζει ο προγραμματισμός με αυτή αλλά και στο γεγονός πως είναι μια γλώσσα η οποία βρίσκεται σχεδόν σε κάθε διακομιστή.

Laravel

Είναι ένα framework για την δημιουργία εφαρμογών διαδικτύου με εύκολη και δομημένη σύνταξη [19]. Είναι γραμμένο σε γλώσσα προγραμματισμού PHP και δημιουργήθηκε από τον **Taylor Otwell** βασιζόμενος στην αρχιτεκτονική **MVC**, δηλαδή Model-View-Controller [20].

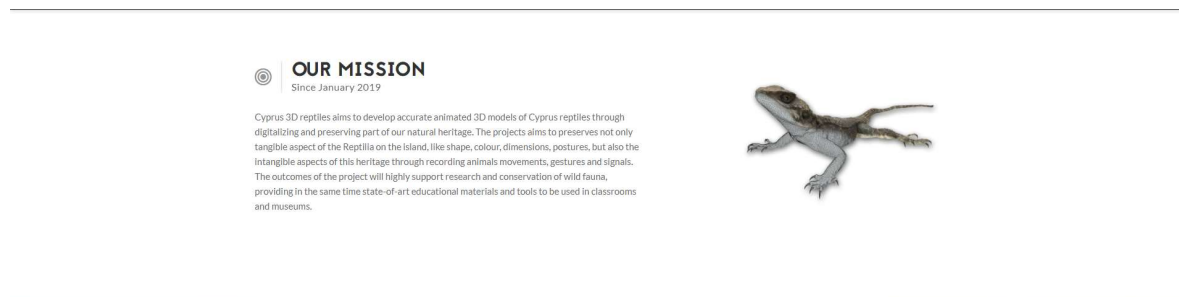
6.2 Παρουσιαση Ιστοσελίδας

Αρχική σελίδα



Εικόνα 40 Αρχική σελίδα ιστοσελίδας

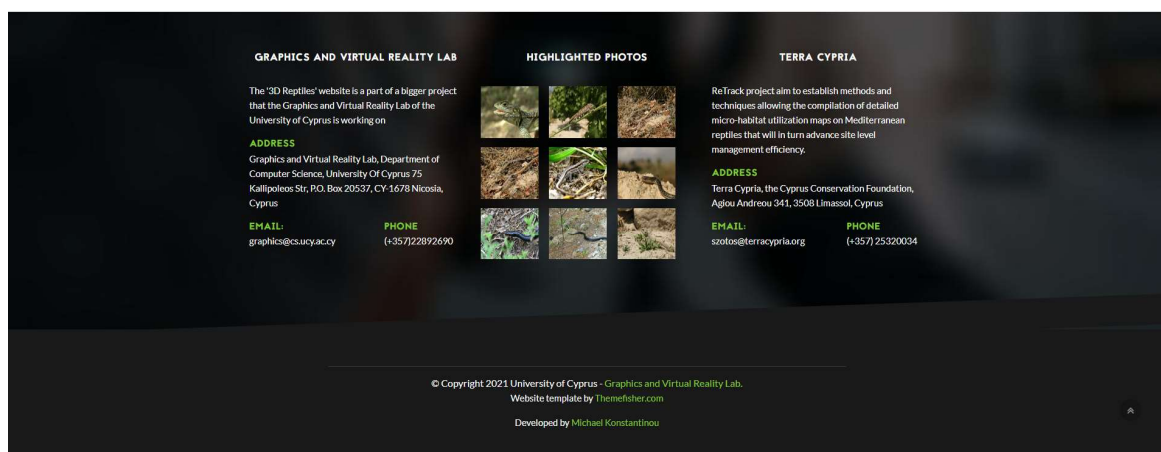
Στην εικόνα 40 φαίνεται η αρχική σελίδα της ιστοσελίδας. Όταν ο χρήστης μετακινηθεί προς το κάτω θα είναι σε θέση να διαβάσει μια μικρή περιγραφή για την μελέτη μας.



Εικόνα 41 Περιγραφή μελέτης

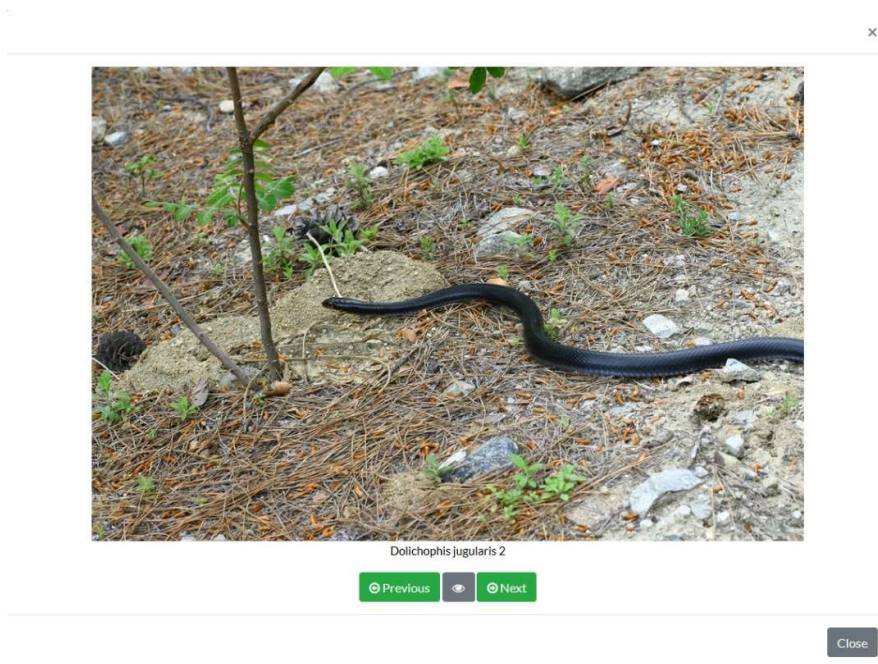
Προχωρώντας πιο κάτω στην αρχική σελίδα ο επισκέπτης του ισότοπου μας μπορεί να δει κάποιες βασικές πληροφορίες για τον στόχο του έργου καθώς και παραπομπή στην σελίδα της TerraCypria με την οποία συνεργαστήκαμε.

Στην εικόνα 42 απεικονίζεται το υποσέλιδο (footer) της ιστοσελίδας το οποίο είναι κοινό και εμφανίζεται σε όλες τις κατηγορίες της ιστοσελίδας. Περιέχει κάποιες βασικές πληροφορίες για το εργαστήριο Graphics and Virtual Reality του πανεπιστημίου Κύπρου καθώς και του TerraCypria του ανοικτού πανεπιστημίου Κύπρου.



Εικόνα 42 Υποσέλιδο (footer)

Κάνοντας κλικ σε μια φωτογραφία, ανοίγει ένας πλοηγητής των φωτογραφιών τον οποίο αναπτύξαμε σε alpine.js



Εικόνα 43 Πλοηγητής φωτογραφιών

Στο πάνω δεξιό μέρος της αρχικής σελίδας φαίνεται το η μπάρα πλοήγησης με τις διάφορες δυνατότητες που προσφέρει η ιστοσελίδα.



Εικόνα 44 Μενού πλοήγησης

Ο χρήστης μπορεί να επιλέξει οποιαδήποτε από τις επιλογές που φαίνονται στην εικόνα 44 και στην οθόνη του θα εμφανιστούν πληροφορίες σχετικά με την κατηγορία που επέλεξε.

Our Reptiles

Όταν ο χρήστη επιλέξει την κατηγορία Our reptiles θα έχει την δυνατότητα να δει τα πειράματα που έχουν πραγματοποιηθεί για τον σκοπό του έργου. Επιλέγοντας την απεικόνιση της σαύρας που φαίνεται στην Εικόνα 45 θα είναι σε θέση να ενημερωθεί σχετικά με το πείραμα της σαύρας αλλά και να επεξεργαστεί την σαύρα που παρουσιάζεται σε τρισδιάστατη μορφή. Οι αντίστοιχες δυνατότητες θα παρέχονται από την ιστοσελίδα για τα πειράματα που έγιναν με το φίδι.



SLING TAILED AGAMA
Learn more about reptile



CYPRUS RACER SNAKE
Learn more about reptile

Εικόνα 45 Επιλογή πειράματος

Model 3D - Slingshaped Agama by DigitalArt3D

Moving tail

Moving head

PHOTOS

Xrullacore

SLING TAILED AGAMA

The Slingshaped Agama is a very hardy lizard, growing up to a total length of 200mm. It has flattened relatively short body and a triangular head. The neck is well defined, and the legs are prominent. Noticeably spiny body with most of tubercles on the back. The tail is stout, covered with rings of spiny scales. The body is blackish brown with a series of yellow blotches along the back and on the forelegs. Dark rhomboid patches are scattered on the back, particularly on the anterior part. The underside is creamy beige. Young specimens are similarly patterned but their drawing is highly contrasted.

General information

Specific for the studied individual

Conservation status

Acquisition information

GENERAL INFORMATION

Scientific name:

Stellagama stellio (Linnaeus, 1758)

Subspecies:

Stellagama stellio cyprica (Dean, 1957)

English common name:

Slingshaped Agama

Greek common name:

Κουρουνόξυ (Kourounoxi)

Order:

Agamidae

Global distribution:

Greece, Turkey, Syria, Lebanon, Egypt, Iraq, Saudi Arabia, Cyprus, Jordan, Israel

Species distribution:

Widespread. From sea level up to the highest peak of Troodos mountain (1,930m altitude)

Sexual dimorphism:

There is sex-specific difference due to the presence of callosus glandular scales on males. Those scales are absent from female individuals. Additionally, the tail base of the males bears the hemipenes and is thus vaulted.

Ecology:

The Slingshaped Agama inhabits a wide variety of habitats. It can be found in dry habitats on stone walls, rocks, beaches, habitats with bushes and shrubs, forests, and fields. It also lives in olive and carob trees as well as other hollow trees. It is a synanthropic species than can be found in cities climbing walls and living inside roofs of houses.

Diet:

The Slingshaped Agama is mainly insectivorous feeding with a large variety of insect upon availability. It can also feed on small vertebrate (eg. young mice or sparrows) and plant material (eg. fruits, petals)

Reproduction:

Oviparous. Reproductive period starts early on spring. Lay clutches of 3-10 eggs. First clutches are laid in May and June. Two clutches are typically laid in a year. Eggs are deposited in rock gaps and holes. First hatchlings appear late August to early September. Embryos temperature-dependent sex determination.

Behaviour:

The Slingshaped Agama is a great heliotherm and thermophilic lizard and can frequently be seen sunbathing active during the very hot midday hours basking on rocks. The Cyprus population seems to remain active on the hottest and remain in this posture while running away. When on trees individuals will run up the trunk on the side opposite to the observer. The Slingshaped Agama is a highly social species. Intraspecific communications involve numerous characteristic movements such as the vertical bobbing of the head (head bobbing), a very commonly observed movement. Head bobbing seems to be a threatening posture against conspecifics during territorial defence and a mating signal for attracting the opposite sex.

Close

Εικόνα 46 Πείραμα σαύρας

Στο πάνω δεξιό μέρος φαίνεται μια μικρή περιγραφή της σαύρας και του. Στην συνέχεια απεικονίζονται τα δύο εικονίδια υπερσυνδέσμοι που φαίνονται στην Εικόνα 47 που εάν ο

56

χρήστης τα επιλέξει θα μεταβεί στο The reptile database [18] ή στο IUCN Red List [19] όπου αναφέραμε κάποιες πληροφορίες για τα στοιχεία των ερπετών που χρησιμοποιήσαμε.



Εικόνα 47 Εικονίδια Υπερσυνδέσμοι

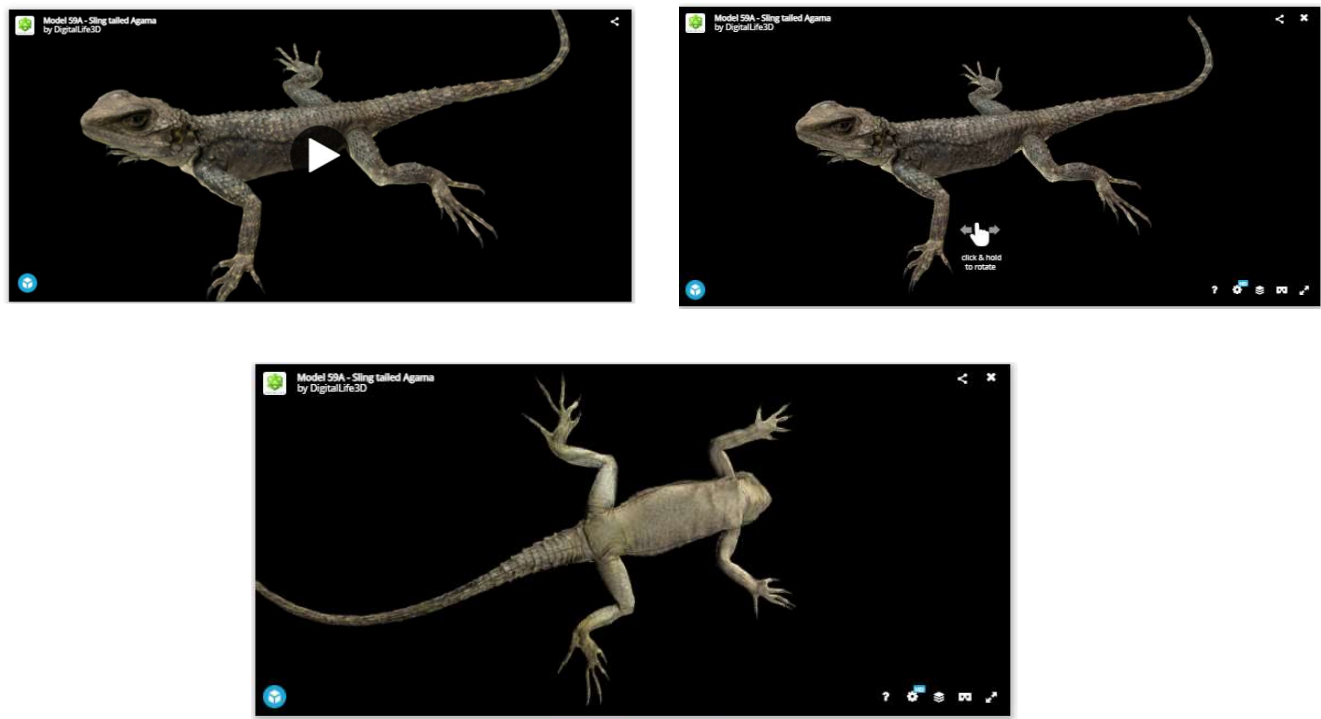
Στην συνέχεια απεικονίζονται κάποιες πληροφορίες για την σαύρα που χρησιμοποιήθηκε για το πείραμα. Οι πληροφορίες είναι χωρισμένες σε κατηγορίες και ο χρήστης έχει την δυνατότητα να πλοηγηθεί σε κάθε μια από τις κατηγορίες κάνοντας κλικ στο όνομα της κατηγορίας που φαίνεται η καρτέλα όπως φαίνεται στην Εικόνα 48.

General information	Specific for the studied individual	Conservation status	Acquisition Information
GENERAL INFORMATION			
Scientific name: <i>Stellagama stellio</i> (Linnaeus, 1758)			
Subspecies: <i>Stellagama stellio cypriaca</i> (Daan, 1967)			
English common name: Starred Agama			
Greek common name: Κουρκουτάς (Kirkutás)			
Higher Taxa (Family): Agamidae			
Global distribution: Greece, Turkey, Syria, Lebanon, Egypt, Iraq, Saudi Arabia, Cyprus, Jordan, Israel			
Cyprus distribution: Widespread. From sea level up to the highest peak of Troodos mountain (1.930m altitude)			
Sexual dimorphism: There is sex-specific difference due to the presence of callous glandular scales on males. Those scales are absent from female individuals. Additionally, the tail base of the males bears the hemipenes and is thus vaulted.			
Ecology: The Starred Agama inhabit a wide variety of habitats. It can be found in dry habitats on stone walls, rocks, beaches, habitats with bushes and shrubs, forests, and fields. It also lives in olive and carob trees as well as other hollow trees. It is a synanthropic species than can be found in cities climbing walls and living inside roofs of houses.			
Diet: The Starred Agama is mainly insectivorous feeding with a large variety of insect upon availability. It can also feed on small vertebrate (eg. young mice or sparrows) and plant material (eg. fruits, petals)			
Reproduction: Oviparous. Reproductive period starts early on spring. Lay clutches of 3-10 eggs. First clutches are laid in May and June. Two clutches are typically laid in a year. Eggs are deposits in rock gaps and holes. First hatchlings appear late August to early September. Exhibits temperature-dependent sex determination.			
Behaviour: The Starred Agama is a great heliophile and thermophily lizard and can frequently be seen remaining active during the very hot midday hours basking on rocks. The Cyprus population seems to remain active throughout the year. The Starred Agamas is a very shy species and will quickly disappear into the nearest rock crevices if disturbed. When in open spaces, individuals sometimes raise themselves on the hindfeet and remain in this posture while running away. When on trees individuals will run up the trunk on the side opposite to the observer. The Starred Agamas is a highly social species. Intraspecific communications involve numerous characteristics movements such as the vertical bobbing of the head (head bobbing), a very commonly observe movement. Head bobbing seems to be a threatening posture against conspecifics during territorial defence and a mating signal for attracting the opposite sex.			

General information	Specific for the studied individual	Conservation status	Acquisition Information
SPECIFIC FOR THE STUDIED INDIVIDUAL			
Gender: Male			
Age group: Adult			
Snout-Vent Length (cm): 10.7			
Tail size (cm): 19.4			
Total size (cm): 30.1			
Morphometric data (cm): Body length: 6.8; Forelimb to hindlimb distance: 4.2; Head length: 2.7; Head width: 2.8; Head depth: 1.8; Body width: 3.0; Pelvic width: 1.8; Tail width: 1.7; Pelvic height: 1.4; Humerus Length: 1.9; Ulna Length: 2.3; Forefoot Length: 1.6; Femur Length: 2.3; Tibia Length: 3.0; Hindfoot Length: 3.1			
Captured Location (name): Parekklisia village, Limassol district, Cyprus			
Captured date: 14.04.2019			

Εικόνα 48 Πληροφορίες για την σαύρα

Στο πάνω αριστερό μέρος της σελίδας φαίνεται μια τρισδιάστατη μορφή της σαύρας. Ο χρήστης μπορεί να δει την σαύρα από όλες τις πλευρές της κάνοντας κλικ πάνω στην εικόνα και μετακινώντας το mouse του.



CYPRUS RACER SNAKE

info

PHOTOS

CYPRUS RACER SNAKE

The Large whip snake is probably the largest snake found in Cyprus, growing up to a total length of almost 3m. The pupils are round. It is a non-venomous snake and thus has aplocheilous dentition. The species displays an ontogenetic colour change. During the first one and a half years, the snake has an olive-green colour with small brown-flecks that cover about two thirds of the body. The underside of juveniles has an iridescent sheen. On reaching sexual maturity, the ground colour darkens and the pattern disappears. The snake gets a shiny bluish-black colour, with only the underside of the head being a whitish, yellowish to red colour. Usually, specimens of more than 1 m are exclusively black.

THE REPTILE DATABASE
www.reptile-database.org

YouTube

General information **Specify for the studied individual** **Conservation status** **Acquisition information**

GENERAL INFORMATION

Scientific name: *Dolichophis jugularis* (Sonnens, 1758)

Subspecies: *Dolichophis jugularis jugularis* (Sonnens, 1758)

English common name: Large whip snake, black snake

Greek common name: Μαύρο φίλι (Μαύρη φίλη), Θάφι (Θάφι), Τυπολόλας (Περσική)

Higher taxa (taxonomy): Colubridae

Global distribution: Turkey, Greece, Cyprus, Israel, Syria, Iran, Jordan, Lebanon, Kuwait

Cyprus distribution: Widespread. From sea level up to 1700m altitude.

Sexual dimorphism: There is a slight sex-specific difference especially in the number of subcaudal scales: males have 107 to 118 subcaudals, females 101 to 113. Additionally, the tail base of the males bears the hemipenes and is thus vaulted. Males sometimes have a rusty brown background colouration.

Ecology: The Large whip snake is the most common of all snakes species in Cyprus. It is widespread and does not appear to need a particular habitat for survival. It has been observed in towns, gardens, fields, meadows, low and high shrubs, and in forests. It often hides under flat stones, logs, shrub piles, rock piles, and stone walls.

Diet: Young individuals of the Large whip snake feeds on insects and small lizards. Adults prey on mice, rats, birds, lizards and other snakes. Cannibalism has been observed.

Reproduction: Oviparous. Reproductive period starts early in spring. Lay clutches of 6-11 eggs early on summer (June/July). First hatchlings appear late August to early September. Exhibits temperature-dependent sex determination.

Behavior: The Large whip snake is a strong, agile and fast-moving snake. Although it is normally found on the ground, it is an excellent climber and has been repeatedly observed climbing in trees and old houses in searching for food. Different types of hunting techniques have been observed depending on the type of prey (e.g. rodents, birds, snakes, lizards). The species seems to remain active throughout the year (not hibernating) but reduce its activity during the cold period of the year. In captivity, these snakes usually get used to being handled though they are rather unpredictable, and may strike or bite at first.

Close

Εικόνα 49 Σελίδα φιδιού

Process

Με την επιλογή Process θα εμφανιστεί στην οθόνη η διαδικασία που ακολουθήσαμε για να πετύχουμε το έργο. Η διαδικασία έχει χωριστεί σε διάφορα κεφάλαια, πρώτο κεφάλαιο είναι το **Overview** όπου εδώ γίνεται μια εισαγωγή στο έργο καθώς και ποιες μεθοδολογίες χρειάστηκαν για τον σκοπό του έργου. Ακολούθως στο κεφάλαιο **Scanning** περιγράφει την τεχνολογία σάρωσης των ερπετών μέσω της τεχνολογίας Beastcam. Στο κεφάλαιο **Rigging, skinning and texturing** που έπεται υπάρχει μια αναφορά στον τρόπο με τον οποίο αναπτύχθηκαν τα αρχεία texture χρησιμοποιώντας γραφιστικά εργαλεία όπως Blender, Meshroom και Meshlab (δεν ήταν μέρος της διπλωματικής αυτό αλλά χρησιμοποιήθηκε, περισσότερα στο κεφάλαιο 3 για τον τρόπο που χρησιμοποιήθηκε). Στην συνέχεια στο κεφάλαιο **Data Capture** περιγράφεται η διαδικασία καταγραφής κινήσεων της σαύρας και του φιδιού. Έπειτα στο κεφάλαιο **Cleaning** γίνεται μια μικρή περιγραφή του αλγορίθμου Forward And Backwards Reaching Inverse Kinematics, εξηγείται γιατί χρειάστηκε να υλοποιηθεί και τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του. Ακολουθεί η περιγραφή της απεικόνισης του φιδιού στο κεφάλαιο **Visualization**. Τέλος στο κεφάλαιο **Data Analysis** γίνεται μια αναφορά για το πως ο Δρ. Σαββας Ζώτος ανέλυσε τα διάφορα δεδομένα που παράχθηκαν κατά την διάρκεια της κίνησης των ερπετών, καθώς επίσης πως η ομάδα της Πληροφορικής επεξεργάστηκε ψηφιακά τα δεδομένα.

In order to motion capture our subjects, we conducted in two wooden enclosures (2.5m x 3m for the snake in the facilities of the Graphics Virtual Reality Lab, University of Cyprus and 1.5m x 1.5m for the lizard in the facilities of Terrestrial Ecosystem Management Lab, Open University of Cyprus). The purpose of building these enclosures was to prevent escape, while simultaneously allowing the free and unobstructed movement of the reptiles in the available space. The enclosures were open from above to allow a direct view of the animals' movement by a set of cameras surrounding the enclosure. For the acquisition of accurate and highly detailed information on animal motion, we combined three different methods using numerous recording devices.



A. OPTICAL MOTION CAPTURE

Aim to achieve high precision motion acquisition allowing the creation of high quality visual 3D patterns of our animal movements.

A passive motion capture system of twelve Optitrack Flex 3 cameras (resolution of 640x480) was used to capture the 3D motion of articulated subjects. The cameras operate at high frequency (in our experiments, we capture movements at 100Hz), each of which can capture the position of any number of bright spots from the reflective markers. For the lizard, we used a total of 17 facial markers (3 mm diameter), while for the snake we used 12 medium-sized markers (1.4 mm diameter). The animals were released in the developed enclosure setup and moved freely. The position of the markers in 3D space was estimated using triangulation, assuming that at least three cameras have direct view of their reflected light. The markers are tracked over time in high frequency, and used to reconstruct a complete three-dimensional pose of the reptile's body.



B. RGB VISION RECORDS

Aim to assist with the visual identification of behaviour

One top view, two fixed side views and one movable hand camera, recording at 24-30 fps (resolution of 1920x1080), were used as to provide an overview of the animals' motion, allowing researchers to identify, label, and categorize their movements.

C. ACCELEROMETERS

Aim to acquiring acceleration signatures for further analysis of the recorded behaviours

The ADXL-4 tri-axial accelerometers of Technomat, weighing no more than 3g, were used to record and analyse acceleration produced by the animals' movements. Accelerometers were glued in the dorsal area near the neck of the two species using non epoxy glue. The position of placement was carefully selected as to allow the device to record even the smallest body movement minimizing at the same time the discomfort of the animal allowing an unobstructed behaviour. Accelerometers were configured to a sample rate of 100 Hz, sensitivity of 2g and a resolution of 8 bits. Prior to attachment to the animals, accelerometers were calibrated to the three-axes.



Εικόνα 50 Περιγραφή Διαδικασίας

FAQ

Στην κατηγορία FAQ υπάρχουν διάφορες ερωτήσεις απαντημένες τις οποίες πιστεύουμε θα έχει ο κάθε χρήστης.

 **QUESTIONS ANSWERED AT A GLANCE**

Q. HOW IS THE DATA MOTION CAPTURED IN YOUR LAB?

We have a Phasespace Impulse X2 motion capture system consisting of 8 double array cameras that are able to capture 3D motion using modulated LEDs. These cameras contain a pair of linear scanner arrays operating at high frequency (480Hz) each of which can capture the position of any number of bright spots of light as generated by the LEDs. Motions are captured in a working volume of approximately 4m x 6m. The capture subject wears a mocap suit of 38 markers and a stylish black garment.

Q. WHICH FORMATS DO YOU USE IN YOUR LAB?

Video format: Dance recording using an HD camera. Data saved in .MP4 (MPEG-4 part 10 H.264) or .FLV (flash) format. C3D format: stored 3D coordinate information, analog data and associated information as it is recorded from the motion capture system. Actor (.FBX) format: the motion of the performer is saved as an actor. Any virtual character (avatar) can be incorporated to perform the dance. This format is compatible with Adobe MotionBuilder, 3Ds Max, Maya, Unity3D etc. Character (.FBX) format: A virtual character has been incorporated to the actor. The character cannot be removed. This format is compatible with Adobe MotionBuilder, 3Ds Max, Maya, Unity3D etc. .RPD format: The PhaseSpace .RPD format that is stored 3D coordinate information, analog data and associated information (including cameras, rigid bodies etc) as it is recorded from the motion capture system. The .RPD files are available upon request.

Q. WHAT KIND OF RGB CAMERAS DID YOU USED?

Q. HOW DID YOU SYNC ALL OPTICAL CAMERAS TOGETHER?

Q. HOW DID YOU SYNC CAMERAS WITH THE ACCELEROMETER?

Q. WHAT WERE THE MAIN CHALLENGES YOU FACED WHEN USING MO-CAP FOR THIS ANIMALS?

Q. WERE ANIMAL STRESS UNDER THIS PROCEDURE?

Q. DID YOU ADD OBSTACLES IN THE ENCLOSURE?

Q. DID YOU ALSO RECORD SOCIAL INTERACTIONS?

Q. HOW DID YOU GLUE THE MARKERS ON THE REPTILES?

Q. HOW DID YOU SYNC THE OPTICAL MOTION CAPTURE DATA WITH THE RGB CAMERAS?

Εικόνα 51 FAQ

Publications

Με την επιλογή Publications ο χρήστης θα μπορεί να ενημερωθεί για διάφορες επιστημονικές δημοσιεύσεις που έγιναν από τα μέλη της ομάδας μας αλλά και να διαβάσει τα σχετικά άρθρα της κάθε δημοσίευσης πατώντας στον σχετικό σύνδεσμο.

PUBLICATIONS

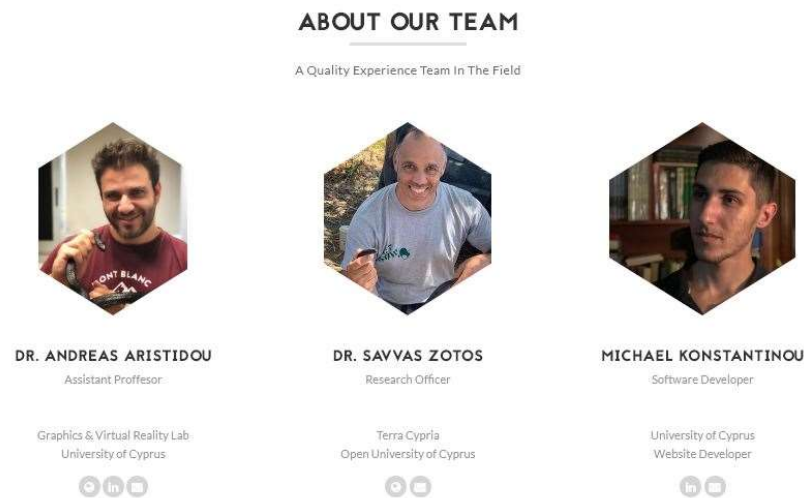


DEVICES AND METHODS FOR RAPID 3D PHOTO-CAPTURE AND PHOTOGRAMMETRY OF SMALL REPTILES AND AMPHIBIANS IN THE LABORATORY AND THE FIELD
Duncan Jirschick, Zachary Corriveau, Trevor Mayhan, Cameron Siler, Mark Mandica, Tony Gamble, Johnson Martin, Jer Bot, Savvas Zotos
Herpetological Review, 51(4), 716-725, December 2020
Because of the importance of specimen identification, and for establishing protocols for new species boundaries, novel methods and tools for identifying and sharing specimen data for vertebrate organisms, particularly amphibians and reptiles, is an important aim for taxonomists. Here, we describe novel tools and techniques for the creation of 3D models of live reptiles and amphibians, both in wild settings in the field and in the laboratory.
[\[Paper\]](#)

Εικόνα 52 Publications

About us

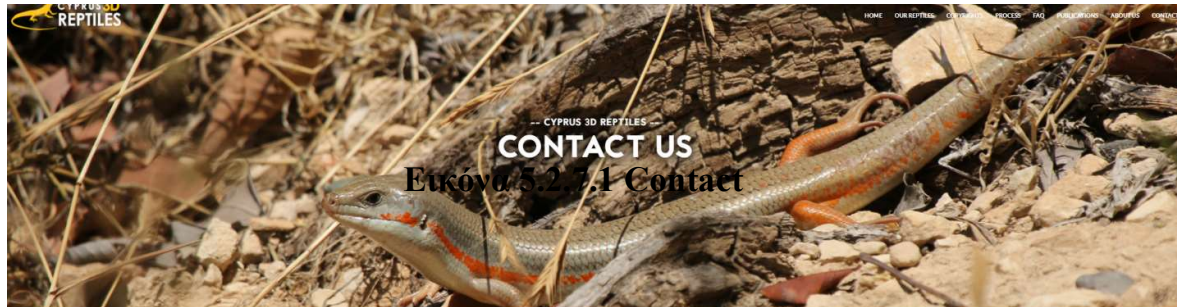
Στην παρούσα κατηγορία ο χρήστης μπορεί να βρει λίγες πληροφορίες για το κάθε ένα από τα μέλη της ομάδας καθώς επίσης και στοιχεία επικοινωνίας για τον καθένα.



Εικόνα 53 About us

Contact

Στην τελευταία κατηγορία από την μπάρα πλοήγησης ο χρήστης επιλέγοντας την θα έχει την δυνατότητα να βρει ακριβώς την θέση των εργαστηρίων μας με την βοήθεια του χάρτη που έχει τοποθετηθεί και είναι συνδεδεμένος με τους χάρτες της Google. Επίσης στην κατηγορία αυτή παρέχονται κι άλλες πληροφορίες επικοινωνίας όπως το τηλέφωνο και η ηλεκτρονική διεύθυνση.



Εικόνα 54 Contact us

Κεφάλαιο 7

Συμπεράσματα

7.1 Γενικά	62
7.2 Καταγραφή κινήσεων χρησιμοποιώντας αισθητήρες	62
7.3 Διόρθωση κινήσεων χρησιμοποιώντας ψηφιακή επεξεργασία	63
7.4 Απεικόνιση και χρήση κινήσεων	64
7.5 Μελλοντική επέκταση	65

7.1 Γενικά

Ήταν μια διπλωματική εργασία οι οποία συνδυάζει πολλά στάδια από διαφορετικές πτυχές της πληροφορικής. Επομένως, σε κάθε φάση της διπλωματικής εργασίας, υπήρξαν διαφορετικά αποτελέσματα, διαφορετικά προβλήματα και διαφορετική προσέγγιση. Αυτό το οποίο έχουν όλα κοινό όμως, είναι το πως μπορούν να συνδυαστούν και να τύχουν ψηφιακής επεξεργασίας είτε οπτικά, είτε μέσω κώδικα.

Σε γενικές γραμμές αυτό που παρατηρήθηκε είναι πως εάν σε κάποιο στάδιο της διπλωματικής εργασίας το αποτέλεσμα ήταν ελλειπτικό ή με σφάλματα, αυτό δεν σήμαινε πως το αποτέλεσμα ήταν κατ' ανάγκη μη διορθώσιμο. Σε κάθε φάση της εργασίας είχε υπολογιστεί πως θα υπήρχε ένα μικρό περιθώριο λάθους το οποίο μπορούσε να διορθωθεί χρησιμοποιώντας τους κατάλληλους αλγορίθμους. Παρόλα αυτά, πρέπει να αναφερθεί πως τα προβλήματα μπορεί να είναι διορθώσιμα, ωστόσο αυτό δεν σημαίνει πως η διαδικασία της διόρθωσης είναι εύκολη.

7.2 Καταγραφή κινήσεων χρησιμοποιώντας αισθητήρες

Εάν υπήρχε μια βάση δεδομένων με τις κινήσεις των ερπετών που χρειαζόμαστε, δεν θα χρειαζόταν να μπούμε σε αυτή τη διαδικασία. Παρόλα αυτά, δεν βρήκαμε κάποια βάση, κι έτσι η καταγραφή των κινήσεων των ερπετών ήταν αναγκαία για το έργο αυτό.

Η καταγραφή παρουσίασε κάποιες δυσκολίες οι οποίες επηρέασαν την ακρίβεια των κινήσεων που παράξαμε. Το πρώτο πρόβλημα είναι πως εμείς θέλαμε όπως οι δείκτες

τοποθετηθούν όσο πιο κοντά στα οστά. Αυτό φυσικά δεν είναι εφικτό καθώς υπάρχει δέρμα, μυς και λίπος ενδιάμεσα. Έτσι αναγκαστικά βλέπουμε κάποιο μικρό κενό (offset).

Ως επίσης, θέλαμε όπως τα ερπετά να αλληλεπιδρούν με το φυσικό τους περιβάλλον κι έτσι στον ειδικά διαμορφωμένο χώρο του εργαστηρίου, μπήκαν αντικείμενα όπως κλαδί και κορμός. Κατά την αλληλεπίδραση των ερπετών με τα αντικείμενα όμως, σε μερικές περιπτώσεις η κάμερα δεν μπορούσε να καταγράψει την θέση κάποιου δείκτη. Ως αποτέλεσμα, έπρεπε να χρησιμοποιηθεί κάποιος αλγόριθμος για την διόρθωση αυτών των τιμών, σε μεταγενέστερο στάδιο.

Το άλλο πρόβλημα που παρατηρήθηκε, είναι πως σε κάποιες καταγραφές ακολουθήθηκε ένας διαφορετικός τρόπος διάταξης των δεικτών. Για παράδειγμα, σε μια καταγραφή η σαύρα να είχε 3 δείκτες στο κεφάλι ενώ σε άλλη καταγραφή να υπάρχει μόνο ένας. Σε μερικές περιπτώσεις αυτό έδωσε μεγαλύτερη ακρίβεια στις κινήσεις, ενώ σε άλλες περιπτώσεις υπήρξαν τα αντίθετα αποτελέσματα.

Σε γενικές γραμμές, προβλήματα στην ακριβή καταγραφή της τοποθεσίας όλων των δεικτών στο ερπετό θα υπάρξουν. Αυτό το οποίο μπορούμε να περιορίσουμε είναι το σφάλμα της καταγραφής έτσι ώστε το επόμενο στάδιο, δηλαδή η διόρθωση των σφαλμάτων να είναι πιο εύκολη.

7.3 Διόρθωση κινήσεων χρησιμοποιώντας ψηφιακή επεξεργασία

Σε αυτό το στάδιο υπήρχουν διάφοροι αλγόριθμοι οι οποίοι είτε χρησιμοποιήθηκαν είτε θα χρησιμοποιηθούν για την επίλυση των προβλημάτων που αναφέρονται στο προηγούμενο στάδιο. Το σημαντικότερο βήμα όμως σε αυτή την φάση, είναι να εξακριβώσουμε εάν τα σφάλματα είναι διορθώσιμα. Στη δική μας περίπτωση ελέγξαμε τις τιμές και βρήκαμε πως κάποια αρχεία είχαν διορθώσιμες τιμές. Έτσι, με την χρήση του κατάλληλου αλγορίθμου διορθώσαμε το πρόβλημα.

Όσο αφορά την σαύρα όμως υπήρξαν κάποια αρχεία τα οποία όπως είδαμε εκ των υστέρων, ήταν δύσκολο να διορθωθούν με την χρήση ενός αλγορίθμου και μόνο. Γι' αυτό, θα πρέπει στη συνέχεια αυτής της διπλωματικής εργασίας, να χρησιμοποιηθούν τεχνικές που έχουν να κάνουν με **υπολογιστική όραση** (computer vision).

Επιπροσθέτως, υπήρχαν και προβλήματα τα οποία δεν είχαν να κάνουν με σφάλματα κατά την καταγραφή, αλλά με την βελτιστοποίηση κάποιων αδυναμιών που παρατηρήθηκαν στο

προηγούμενο στάδιο. Μια από αυτές είναι το γεγονός πως σε κάποια σημεία του ερπετού δεν μπορούσαν να μουν δείκτες. Πιο συγκεκριμένα, στην καταγραφή του φιδιού, δεν μπορούσε να μπει κάποιος δείκτης από ένα σημείο και έπειτα επειδή η ουρά γινόταν ολοένα και πιο λεπτή. Έτσι, έπρεπε να γίνει κάποια διόρθωση υπολογιστικά για να πετύχουμε την πραγματική κίνηση της ουράς. Μπορεί η τιμή να βασίστηκε στην πραγματική καταγραφή αλλά επί της ουσίας δεν ήταν η πραγματική θέση(κίνηση) της ουράς. Ήταν μια πολύ καλή προσέγγιση.

Μιλώντας για προσεγγίσεις, υπήρξε ακόμα μια προσέγγιση στην δημιουργία της κίνησης η οποία βασίζεται στους δείκτες και αφορούσε το σώμα των ερπετών. Στο φίδι το αποτέλεσμα ήταν κάπως πιο χαρακτηριστικό διότι τα άκρα της σάυρας ήταν πιο σταθερά. Ο κορμός της σάυρας είναι ευθύς. Εν εναντία περιπτώσει, ο σκελετός του φιδιού παρουσιάζει καμπυλώματα. Ως εκ τούτου, χρησιμοποιήθηκε ο κατάλληλος αλγόριθμος για την εκτίμηση του σώματος του φιδιού εν συναρτήσει των πραγματικών διαστάσεων του ερπετού. Παρόλα αυτά, σε κάποια σημεία το αποτέλεσμα δεν ήταν φυσιολογικό, καθώς παρουσίαζε σε κάποια σημεία απότομες εναλλαγές.

7.4 Απεικόνιση και χρήση κινήσεων

Εν ολίγοις η απεικόνιση βασίστηκε στις θεμελιώθεις αξίες της τεχνολογίας λογισμικού. Η προσέγγιση μας επί του θέματος ήταν η ίδια προσέγγιση με την οποία θα αναπτύσσαμε μια εφαρμογή. Ωστόσο αξίζει να γίνει ιδιαίτερη αναφορά στις δυσκολίες που συναντήσαμε αλλά και στις λύσεις που προέκυψαν, όσο αφορά την απεικόνιση αρχείων .FBX

Υπάρχουν 2 τρόποι να παρουσιάσει κανείς στην εφαρμογή του τα αρχεία αυτού του τύπου. Ο πρώτο είναι χρησιμοποιώντας έτοιμα προγράμματα επί πληρωμής τα οποία είναι εύκολα προς χρήση. Ο δεύτερος είναι η ανάπτυξη κάποιου “Viewer” (δηλαδή οθόνη αναπαραγωγής πολυμέσων). Έχοντας χρησιμοποιήσει και τα δύο, παρατηρήθηκε πως είναι πιο εύκολο και πιο ανανεωμένο να χρησιμοποιήσει κανείς όποιο τρόπο θέλει, φτάνει να αξιοποιήσει τις τεχνολογίες διαδικτύου. Τόσο έτοιμες υπηρεσίες σε τεχνολογίες διαδικτύου είναι πιο εύκολες προς χρήση όσο και η ανάπτυξη ειδικής οθόνης αναπαραγωγής πολυμέσων.

Το μεγαλύτερο πρόβλημα που διαπιστώθηκε, είναι πως η απεικόνιση των αρχείων τύπου .fbx, χωρίς την χρήση τεχνολογιών διαδικτύου, παρουσιάζει προβλήματα συμβατότητας καθώς η ανάπτυξη μιας οθόνης αναπαραγωγής πολυμέσων βασίζεται καθαρά και μόνο στις υπηρεσίες που προσφέρει η εταιρεία Autodesk (δηλαδή η εταιρεία που συντηρεί τον τύπο

.fbx). Αυτό απαιτεί χρήση της γλώσσας προγραμματισμού C++ η οποία μπορεί να είναι δύσκολη για αρχάριους. Αν και προσφέρονται βιβλιοθήκες για την γλώσσα προγραμματισμού Python, η Autodesk ενθαρρύνει τους προγραμματιστές της όπως δημιουργήσουν τις δικές τους χρησιμοποιώντας τον υφιστάμενο κώδικα σε C++ (γνωστό και ως python bindings). Είναι επίπονη διαδικασία η οποία απαιτεί προχωρημένο επίπεδο γνώσεων στις δύο γλώσσες προγραμματισμού ενώ παράλληλα παρουσιάζονται προβλήματα συμβατότητας από συσκευή σε συσκευή μιας και η Autodesk, δεν υποστηρίζει Linux (τουλάχιστον σε αυτό που δουλέψαμε εμείς).

Η καλύτερη προσέγγιση από πλευράς ευκολίας ήταν η προσέγγιση του αρχείου ως ένα γενικό αρχείο το οποίο πρέπει να φορτωθεί χρησιμοποιώντας το WebGL API. Πιθανόν, με παρόμοιο σκεπτικό η προσέγγιση του θέματος της απεικόνισης χρησιμοποιώντας την βιβλιοθήκη OpenGL η οποία δεν αφορά τους φυλλομετρητές, να είχε παρόμοια ευκολία και παρόμοια αποτελέσματα. Κάτι τέτοιο όμως δεν δοκιμάστηκε για σκοπούς επεκτασιμότητας και χρήσης στην ιστοσελίδα που αναπτύξαμε.

7.5 Μελλοντική επέκταση

Μέχρι αυτό το σημείο, έχουν ήδη ολοκληρωθεί κάποια από τα στάδια του έργου αλλά δεν αποτελούν ολόκληρο το έργο. Επομένως, για να ολοκληρωθεί το έργο θα πρέπει όπως – χρησιμοποιώντας ως βάση το παρόν στάδιο – να συνεχίσει κανείς με την ανάλυση των κινήσεων σε επίπεδο μηχανικής μάθησης και βαθιών νευρωνικών δικτύων (machine learning and deep learning) για συσχετισμό με την επιτάχυνση.

Η διπλωματική εργασία, έχει καταφέρει να παράξει σε αυτό το σημείο, μια πληθώρα από κινήσεις οι οποίες είναι διορθωμένες, ελεγμένες και έτοιμες προς χρήση. Συνεπώς, η συνέχεια προβλέπει όπως γίνει η εκπαίδευση ενός deep learning δικτύου για εντοπισμό πρότυπων συμπεριφερών. Εδώ θα πρέπει να χρησιμοποιηθούν τα δεδομένα του επιταχυντή. Το δίκτυο θα πρέπει να εκπαιδευτεί έτσι ώστε να σαρώνει τα δεδομένα του επιταχυντή και να βρίσκει ξανά τις ίδιες επαναλαμβανόμενες συμπεριφορές

Έπειτα, θα πρέπει να γίνει μια σύνδεση των δεδομένων αυτών, με τις κινήσεις οι οποίες αποκοπήκαν κατά την διάρκεια της διπλωματικής αυτής εργασίας. Επί της ουσίας, εδώ τίθεται το ερώτημα κατά πόσο μπορεί να γίνει μια ανακατασκευή ή δημιουργία της τρισδιάστασης κίνησης του ερπετού μόνο με την χρήση από επιταχυντές. Για να γίνει κάτι

τέτοιο θα πρέπει να γίνει η τροφοδότηση ενός δικτύου με μεγάλα δεδομένα από επιταχυντές και mosaic τα οποία θα είναι συγχρονισμένα.

Βιβλιογραφία

- [1] B. T. Llyr ap Cenydd, "An embodied approach to antropod animation".
- [2] S. H. a. Y. Choi, "Pose Estimation of Small-articulated Animals using Multiple View Images".
- [3] H. K. a. Y. L. Taesoo Kwon, "Control of an Iguana Character Using Soft-Body Simulation".
- [4] W. L. W. D. C. B. a. J. F. O. J. K. Hodgins, "Animating human athletics," *Proc. ACM SIGGRAPH*, pp. 71-78.
- [5] K. L. a. M. v. d. P. K. Yin, "SIMBICON: Simple biped locomotion control," *ACM Trans. Graph.*, vol. 26, no. 3, p. 105, 2007.
- [6] T. K. a. J. K. Hodgins, "Control systems for human running using an inverted pendulum model and a reference motion capture sequence," *Proc. Symp. Comput. Animation*, p. 129–138, 2010.
- [7] Y. Y. a. C. K. Liu, "Optimal feedback control for character animation using an abstract model," *ACM Trans. Graph.*, vol. 29, no. 4, p. 74, 2012.
- [8] S. K. a. J. L. Y. Lee, "Data-driven biped control," vol. 29, no. 4, p. 129, 2010.
- [9] R. Rittep, "Lateral bending during lizard locomotion," *J. Exp. Biol.*, vol. 173, no. 1, pp. 1-10, 1992.
- [10] C. T. F. a. T. C. Ko, "Mechanics of locomotion in lizards," *J. Exp. Biol.*, vol. 200, no. 16, pp. 2177-2188, 1997.
- [11] M. v. d. P. a. A. F. v. d. S. T. Geijtenbeek, "Flexible muscle-based locomotion for bipedal creatures," *ACM Trans. Graph.*, vol. 32, no. 6, p. 206, 2013.
- [12] G. T. a. C. K. L. J. Tan, "Soft body locomotion," *ACM Trans.*, vol. 31, no. 4, p. 26, 2012.
- [13] S. M. B. T. C. S. R. S. M. G. S. Coros, "Deformable objects alive," vol. 31, no. 4, p. 69.
- [14] J. K. a. N. S. Pollard, "Fast simulation of skeleton-driven deformable," *ACM Transactions on Graphics*, vol. 30, no. 5, p. 121.
- [15] K. K. Y. B. W. a. B. G. L. Liu, "Simulation and control of skeleton-driven soft body characters," *ACM Transactions on Graphics*, vol. 32, no. 6, p. 215, 2013.
- [16] NaturalPoint, "Optitrack - Motive Basics," Optitrack, 31 10 2019. [Online]. Available: https://v22.wiki.optitrack.com/index.php?title=Motive_Basics.
- [17] Š. a. K. G. a. O. F. a. B. R. a. S. E. a. J. H. a. P. M. Obdržálek, "Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population," *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1188-1193, 2012.
- [18] J. L. Andreas Aristidou, "FABRIK: A fast, iterative solver for the Inverse Kinematics problem," *Graphical Models*, 2011.
- [19] Y. C. a. J. L. Andreas Aristidou, "Extending FABRIK with model constraints," *COMPUTER ANIMATION AND VIRTUAL WORLDS*, 2015.
- [20] J. Y. C. a. A. S. A. Aristidou, "Inverse Kinematics Techniques in Computer Graphics: A Survey," *COMPUTER GRAPHICS forum*, vol. 00, pp. 1-24, 2017.
- [21] WildByte Technologies, DDMT Manual, 2020.
- [22] N. Jennings, "Real Python," 2018. [Online]. Available: <https://realpython.com/python-sockets/>. [Accessed 19 11 2020].
- [23] Microsoft, "Windows Forms overview," Microsoft, 30 3 2017. [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/windows-forms-overview?view=netframeworkdesktop-4.8>. [Accessed 20 11 2020].

- [24] W3Schools, "W3Schools," [Online]. Available: <https://www.w3schools.in/what-is-client-server-architecture/>. [Accessed 12 06 2020].
- [25] B. Danchilla, Three.js Framework. In: *Beginning WebGL for HTML5*, Apress, Berkeley, CA, 2012.
- [26] J. Dirksen, Three.js Cookbook, Packt Publishing Ltd, 2015.
- [27] D. Cantor, WebGL beginner's guide, Packt Publishing Ltd, 2012.
- [28] M. Jasim, Building cross-platform desktop applications with electron, Packt Publishing Ltd, 2017.
- [29] S. a. S. V. Tilkov, "Node.js: Using JavaScript to build high-performance network programs," *IEEE Internet Computing*, pp. 80-83, 2010.
- [30] J. A. Livermore, "Factors that impact implementing an agile software development methodology," *Proceedings 2007 IEEE SoutheastCon*, pp. 82-86, 2007.
- [31] M. R. J. Qureshi, "Agile software development methodology for medium and large projects," *IET software* 6.4, Vols. 358-363, 2012.
- [32] "CodeAcademy," [Online]. Available: <https://www.codecademy.com/learn/introduction-to-javascript/modules/learn-javascript-introduction>. [Accessed 15 05 2021].
- [33] "Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/JQuery>. [Accessed 13 05 2021].
- [34] C. Porzio, "Github - Alpine Js," [Online]. Available: <https://github.com/alpinejs/alpine/>. [Accessed 1 November 2020].
- [35] "Laravel - The php Framework," Laravel, 2011. [Online]. Available: <https://laravel.com/>. [Accessed 21 November 2020].
- [36] "Wikipedia - Laravel," [Online]. Available: <https://en.wikipedia.org/wiki/Laravel>. [Accessed 23 May 2021].
- [37] "THE REPTILE DATABASE," 10 November 1995. [Online]. Available: <http://reptile-database.org/>. [Accessed 23 April 2021].
- [38] "The IUCN Red List of threatened species," [Online]. Available: <https://www.iucnredlist.org/>. [Accessed 21 April 2021].
- [39] [Online]. Available: <https://www.codecademy.com/learn/introduction-to-javascript/modules/learn-javascript-introduction>.

Κατάλογος εικόνων και γραφικών παραστάσεων

Εικόνα 1 Project, Top-down approach	3
Εικόνα 2 Data acquisition	8
Εικόνα 3 Δημιουργία Markeset και ετικετών	9
Εικόνα 4 Ονομασία δεικτών (labeling)	10
Εικόνα 5 Swap in Motive	11
Εικόνα 6 Εξαγωγή δεδομένων σε C3D file	11
Εικόνα 7 Απόσταση δύο διαδοχικών δεικτών με το πέρασμα των frames	14
Εικόνα 8 Αποτελέσματα από πρόγραμμα-script	15
Εικόνα 9 Μαθηματική συνθήκη κατά την οποία δηλώνει την ευθεία σειρά των δεικτών	16
Εικόνα 10 Εξίσωση εύρεσης αποστάσεως μεταξύ δύο δεικτών	16
Εικόνα 11 Μετατόπιση συντεταγμένων πριν την εφαρμογή του FABRIK	16
Εικόνα 12 Μετατόπιση συντεταγμένων μετά την εφαρμογή του FABRIK	17
Εικόνα 13 Ονόματα των δεικτών από python script	18
Εικόνα 14 Επιλογή δεικτών στο MotionBuilder μέσω Python script	19
Εικόνα 15 Δείκτες και ακριβή τοποθεσία στο MotionBuilder μέσω Python	20
Εικόνα 16 Ακριβής τοποθεσία δεικτών στο φίδι	22
Εικόνα 17 Το μοντέλο του φιδιού που περιέχει σκελετό, mesh και texture	23
Εικόνα 18 Το μοντέλο του φιδιού με τα joints	23
Εικόνα 19 Το μοντέλο του φιδιού scaled στα 190.30 εκατοστά	23
Εικόνα 20 Χρωματισμός των joints στο μοντέλο φιδιού	24
Εικόνα 21 Ονομασίες constraints στο πρόγραμμα Motion Builder	25
Εικόνα 22 Αντιστοιχία των δεικτών στο φίδι	25
Εικόνα 23 Ρυθμίσεις δείκτη	26
Εικόνα 24 Ενεργοποίηση δεικτών	26
Εικόνα 25 Αποτελέσματα εφαρμογής περιορισμών 1 – Χωρίς διόρθωση FABRIK	27
Εικόνα 26 Αποτελέσματα εφαρμογής περιορισμών 2 - Με διόρθωση FABRIK και Mesh	27
Εικόνα 27 Η προσθήκη του τελευταίου δείκτη	29
Εικόνα 28 Πρώτοι έλεγχοι επικοινωνίας με το DDMT	34
Εικόνα 29 Εμφάνιση μοντέλου από sketchfab σε Qt application	37
Εικόνα 30 Three.js integration - Initial html page	39
Εικόνα 31 Settings - Data movements with FBX source	40
Εικόνα 32 Console output - Δημιουργία των html σελίδων δυναμικά	40
Εικόνα 33 Πρώτη έκδοση με το three.js integration	41
Εικόνα 34 Εντολές για έναρξη του mini-server	42

Εικόνα 35 Σελίδα html χωρίς λεπτομέρειες	43
Εικόνα 36 Τελικό αποτέλεσμα	43
Εικόνα 37 Συσχετισμός κινήσεων και frames - Original video	48
Εικόνα 38 Applying offset to imported video	49
Εικόνα 39 Αποκοπή κίνησης χρησιμοποιώντας τους δείκτες Start και End	50
Εικόνα 40 Αρχική σελίδα ιστοσελίδας	54
Εικόνα 41 Περιγραφή μελέτης	54
Εικόνα 42 Υποσέλιδο (footer)	55
Εικόνα 43 Πλοηγητής φωτογραφιών	55
Εικόνα 44 Μενού πλοήγησης	55
Εικόνα 45 Επιλογή πειράματος	56
Εικόνα 46 Πείραμα σαύρας	56
Εικόνα 47 Εικονίδια Υπερσυνδέσμοι	57
Εικόνα 48 Πληροφορίες για την σαύρα	57
Εικόνα 49 Σελίδα φιδιού	58
Εικόνα 50 Περιγραφή Διαδικασίας	59
Εικόνα 51 FAQ	60
Εικόνα 52 Publications	60
Εικόνα 53 About us	61
Εικόνα 54 Contact us	61

Παράρτημα Α – Κινήσεις φιδιού

Πιο κάτω παρουσιάζονται οι πίνακες με τις κινήσεις έτσι όπως ακριβώς ο Δρ. Σάββας Ζώτος τις ανέλυσε χρησιμοποιώντας το βοηθητικό προγράμμα DDMT και τα βίντεο. Όπου οι εγγραφές είναι με κόκκινο χρώμα σημαίνει πως είτε η κίνηση δεν υπάρχει στο αρχείο c3d/fbx είτε πως η κίνηση δεν είναι καθαρή στο αρχείο c3d/fbx.

Motive code: Take 2019-05-14 13:22:00 AM / Take 01 / My clock 13:20

Syncing RGB video with Motive: 13:22:24:00 (RGB) = 16:000 (Motive)

Time in	Frame start	Frame end	Duration (sec)	Behaviour
13:22:24:00	1538	1838	3 sec	Throw forward first half
13:22:27:01	1838	2838	10 sec	Slow slithering forward
13:22:37:01	2838	5338	25 sec	Slow slithering along corner of right wall
13:23:02:01	5600	6100	5 sec	Head right turn - up on wall – move right 180 – down wall
13:25:33:00	6100	6400	3 sec	Defensive head and first half retraction – head left – up – back
13:25:45:00	7300	7800	5 sec	Head defensive retraction – slithering back – head down
13:25:53:00	7800	8050	2,5 sec	Head defensive retraction – slithering back – head down
13:25:55:31	8050	8200	1,5 sec	Head defensive retraction – slithering back – head down
13:25:57:01	8200	8800	6 sec	Head defensive retraction – head up

Syncing RGB video with Motive: 14:20:00:00 (RGB) = 4:09:000 (Motive)

Syncing RGB video with Accelerometers: 14:15:42:00 (RGB) = 14:16:10

Time in (RGB)	Start	End	Duration (sec)	Behaviour
14:15:50			0:03:20	Rapid small head movements right/left and up/down - searching were to go.
14:15:50			1 sec	Head left turn 180 – Rapid
14:15:52			1 sec	Head left turn 180 – Rapid
14:16:00			3 sec	Head right turn 180 + 180 – in steps
14:16:50	5958	6090	1 sec	Head right turn 90
14:17:00	7126	7277	1 sec	Head left turn 90
14:17:30	9855		1 sec	Head right turn 90
14:17:42	11114	11229	1 sec	Head right shift
14:17:49	11870	12212	3 sec	Head small straight forward move
14:18:05	13465		3 sec	Head slow left shift (sway) retraction – left turn 90 + 90 in steps
14:18:17	14670	14770	1 sec	Head left turn 180 – Rapid
14:18:26	15570	15670	1 sec	Head left turn 180
14:18:40	16965	17065	1 sec	Head left turn 180
14:19:00	18970	19070	1 sec	Head up – climbing above small obstacle
14:20:00	25031	25985	10 sec	Slow straight slithering of first half
14:20:18	26780	26980	2 sec	Head open left turn 180 (yaw + sway)
14:21:12	32216	32642	3 sec	Head up – turn right 180 – climb small obstacle
14:21:45			0:00:05	Head fast left shift (sway) retraction
14:21:55	36500	36596	0:00:30	Turning left – raising head – rapid motion – circular movement around obstacle
14:22:02	37140	37350	2 sec	Head right shift (sway) retraction
14:22:10	38000	38200	2 sec	Head right shift (sway) retraction
14:22:14	38400	38600	2 sec	Head right shift (sway) retraction
14:22:18	38800	39000	2 sec	Slow straight slithering of first half
14:22:19	38900	39000	1 sec	Head rapid heave up (5cm)
14:22:24	39390	39490	1 sec	Head rapid heave up (5cm)
14:22:56	42700	42954	2 sec	Rapid forward movement
14:23:00	42997	44733	17 sec	Slow slithering forward - large head sway
14:23:17	44750	45708	10 sec	Change slithering direction – left 90 – continue slithering

14:23:54	48400	50006	15 sec	Slow slithering forward - large head sway (GOOD)
14:24:07	49700	51200	15 sec	Raising the head to climb wall (surge and pitch) (10 cm) – right sway – left sway – right say – lowering head (surge and pitch) (GOOD)
14:24:45	53500	56510	30 sec	Slow slithering forward - large head sway -sway move left to change slithering direction and then right again to go back on initial direction – continue slithering. (GOOD)
14:25:15	56510	57010	5 sec	Raising the heat to climb wall (surge and pitch) (10 cm) – right sway – lowering head (surge and pitch) (GOOD)
14:25:25	57500	58000	5 sec	Raising the head to climb wall (surge and pitch) (10 cm) – left sway – lowering head (surge and pitch) (GOOD)
14:25:35	58510	59010	5 sec	Raising the head to climb wall (surge and pitch) (10 cm) – right/left/right sway
14:26:10	62010	64010	20 sec	Slow slithering forward - large head sway. (GOOD)

Motive code: Take 2019-05-14 14:17:00 AM_001 / Take 03 / My clock 14:33

Syncing RGB video with Motive: 14:33:55:30 (RGB) = 1:46:000 (Motive)

Time in	Motive's timestamp	Duration (sec)	Behaviour
14:35:22		1 sec	Rapid defence retraction – right turn 90 – heave up – retract
14:36:14		1 sec	Short biting attack and retraction
14:36:55		1 sec	Rapid right head turn

Syncing RGB video with Motive: 16:08:19:50 (RGB) = 31:000 (Motive)

Two cameras – GoPro from above and a second one closeup within the cage. Nice shots.

Time in	Start	End	Duration (sec)	Behaviour
16:08:22	3340	5340	20 sec	Slow slithering forward - large head sway.
16:08:42	5340	7146	20 sec	Slow slithering forward along corner of left wall
16:09:03	7450	7950	5 sec	Short slow slithering forward - large head sway
16:09:19	9039	9239	2 sec	Rapid head turn right – 180 – short surge forward
16:10:45	17658	17871	2 sec	Head turn right 90 – attack to scare. No retraction
16:11:41	23248	23389	2 sec	attack to scare with retraction (Good in close-up camera)
16:11:54	24511	24608	1 sec	Short head retraction
16:11:59	25167	25356	2 sec	Short attack to scare with retraction and right yaw
16:12:24	27380	27880	5 sec	Turn right 90 – up head to pass mall obstacle and retract
16:12:35	37250	37750	5 sec	Rapid retract and head up (heave and pitch) – turn right 180
16:16:53	42783	42983	2 sec	Rapid head turn left - 180
16:16:55	42983	45483	25 sec	Slow slithering forward - large head sway - move left to change slithering direction 180 (GOOD)
16:17:40	50000	53500	35 sec	Slow slithering forward - large head sway (VERY GOOD)
16:18:16			20 sec	Straight head retraction – Head extension front – right open turn 180 (sway and yaw) – head turn left 90 – right shift - head turn right 180
16:19:38			20 sec	Rapid left turn – head up – defensive – biting attach ejecting front half forward – rapid retraction turns right – rapid retraction turn right
16:19:58			2 sec	biting attach ejecting front half forward – retraction – left swift

Syncing RGB video with Motive: 16:28:55:30 (RGB) = 29:000 (Motive)

Time in	Start	End	Duration (sec)	Behaviour
16:29:02	3431	5431	20 sec	Slow slithering forward - large head sway (VERY GOOD)
16:29:56	8970	10502	20 sec	Rapid forward extension followed by slow slithering forward
16:31:10	16750	19250	25 sec	Slow slithering forward - large head sway (VERY GOOD) – climb on small obstacle – head right turn 180
16:32:36	25342	29342	40 sec	Slow slithering forward - large head sway (VERY GOOD)
16:33:45	32242	36242	40 sec	Short slow slithering forward - large head sway – Raising the head to climb wall (surge and pitch) (10 cm) – right sway – lowering head (surge and pitch) – start slow slithering along corner of left wall.
16:35:17	41442	41642	2 sec	Open right turn (yaw + sway) of head 90
16:36:33	49042	49242	2 sec	Open right turn (yaw + sway) of head 90
16:36:40	49743	52241	25 sec	Slow slithering forward - large head sway (VERY GOOD)
16:37:10	52742	54746	20 sec	slow slithering along corner of right wall
16:38:38	61540	63540	20 sec	Slow slithering forward - large head sway – elevating head to climb small obstacle – turn head right 180
16:39:42	63947	65543	20 sec	Rapid turn head right 180 – left 90 – lower head to the ground – lifting head up (10cm) (Pitch 60) – shift left – turn left 90 -
16:40:09			2 sec	Turn right 180 – eject half first forward
16:40:40			10 sec	Short slow slithering forward - large head sway

Παράρτημα Β – Κινήσεις σαύρας

Πιο κάτω παρουσιάζονται οι πίνακες με τις κινήσεις έτσι όπως ακριβώς ο Δρ. Σάββας Ζώτος τις ανέλυσε χρησιμοποιώντας το βοηθητικό προγράμμα DDMT και τα βίντεο. Όπου οι εγγραφές είναι με κόκκινο χρώμα σημαίνει πως είτε η κίνηση δεν υπάρχει στο αρχείο c3d/fbx είτε πως η κίνηση δεν είναι καθαρή στο αρχείο c3d/fbx. Με πράσινο χρώμα παρουσιάζονται οι κινήσεις όπου δεν φαίνεται η αλλαγή της κίνησης (π.χ. η σαύρα τρώει χωρίς να κουνάει το κεφάλι της)

Motive code: Take 2019-06-20 03:05:38 PM / Take 02 / My clock 15:05:25

Syncing RGB video with Motive: 15:04:34:07 (RGB) = 43:050 (Motive)

Syncing RGB video with Accelerometers:

RGB time in	Frames (100fps)	Duration (sec)	Behaviour
15.04.34.00	4332 - 4440	1	Running slow
15.04.37.00	4670 - 4777	1	Turning right
15.04.38.00	4777 - 4903	1	Running slow
15.04.41.00	5063 - 5151	1	Turning right
15.04.48.00	5699 - 5799	1	Turning left
15.04.54.00	6300 - 6432	1	Running slow
15.04.57.00	6648 - 6790	1	Running slow
15.05.13.00		1	Try to climb on glossy surface
15.06.10.00		5	Try to climb on glossy surface
15.06.20.00	14949 - 15160	3	Running
15.11.36.00	43309	74	Eating insect

Motive code: Take 2019-06-20 03:30:01 PM / Take 03 / My clock 15:27:50

Syncing RGB video with Motive: 15:28:36:35 (RGB) = 25:080 (Motive)

Syncing RGB video with Accelerometers: 15:28:11:28 = 15:30:00 (My clock)

RGB time in	Frames	Duration (sec)	Behaviour
15.28.36.00	2560 - 2830	3	Turn right and run slow
15.29.27.00	7623 - 8090	5	Run slow
15.29.45.00	9480 - 9948	5	Turn right and run slow
15.32.40.00	27049 - 27279	4	run slow
15.37.27.00	55700 - 55861	3	run slow

Motive code: Take 2019-06-21 11.15.32 AM / Take 01 / My clock 11:15:30
17 markers on animal (one on head)

Syncing RGB video with Motive: 11:15:33:19 (RGB) = 1:19:080 (Motive)
Syncing RGB video with Acceler.: 11:14:41:00 (RGB) = 11:16:00 (My clock)

RGB time in	Motives time in	Duration (sec)	Behaviour
11:15:33:00	7949 - 8037	1	Fast run short
11:16:19:00	12498 - 12793	4	Walk and jump on log (Good posture on log)
11:17:44:00	21207 - 21317	2	Jump off the log (down)
11:17:49:00	21654 - 21745	1	walk
11:18:11:00	24102 - 24387	1	Rapid turn left
11:18:25:00	25400 - 25625	3	Jump on and jump off log
11:21:23:00	43000 - 43293	3	Turn left and walk
11:23:08:00	53622 - 53740	3	Jump on log and climb
11:23:58:00	58650	4	Turn left on log small climb
11:24:22:00	60883 - 60973	2	Jump off the log

Motive code: Take 2019-06-21 11.30.06 AM / Take 02 / My clock 11:30:00
17 markers on animal (one on head)

Syncing RGB video with Motive: 11:29:32:23 (RGB) = 46:040 (Motive)

RGB time in	Motives time in	Duration (sec)	Behaviour
11:29:31:00	4616 - 4829	2	Walk and turn right
11:29:37:00	5182 - 5275	1	Jump on log
11:31:41:00	17600 - 17728	1	Jump on log (Good posture on log)
11:33:23:00	27780 - 27934	1	Jump log to log
11:33:38:00	29273 - 29448	2	Descent from log
11:33:47:00	30090 - 30214	1	Walk
11:34:04:00	31814 - 31963	2	Walk and climb log
11:34:08:00	32277 - 32434	2	Walk and descent from log

11:34:12:00	32571 - 32646	1	Rapid turn right
11:34:14:00	32850 - 33162	2	Walk, turn right, jump on log
11:34:56:00	37133 - 37247	2	Turn left on log
11:35:18:00	39538 - 40186	5	Walk and jump on log
11:36:04:00	43889 - 44138	1	Turn left on log
11:36:31:00	46543 - 46637	2	Run log to log (Good posture on log)

Motive code: Take 2019-06-21 12.37.27 PM / Take 03 / My clock 12:37:20
19 markers on animal (3 on head close together)

Syncing RGB video with Motive: 12:37:02:20 (RGB) = 1:05:050 (Motive)

RGB time in	Motives time in	Duration (sec)	Behaviour
12:37:02:00	6473 - 6685	2	Turn left and walk
12:37:04:00	6765 - 6886	2	Turn right and walk
12:37:33:00	9613 - 9726	1	Turn left
12:37:35:00	10167 - XAΘHKE	20	Walk stop walk stop

Motive code: Take 2019-06-21 03.29.48 PM / Take 05 / My clock 15:29:20
19 markers on animal (3 on head further apart)

BAD QUALITY

Syncing RGB video with Motive: 05:14:55:16 (RGB) = 30:000 (Motive)

I think there is a problem with syncing. We must sync again.

Syncing RGB video with Acceler.: 15:29:33:00 (RGB) = 15:29:23 (My clock)

RGB time in	Motives time in	Duration (sec)	Behaviour
15:30:25:00	5405 - 5600	2	Run (good)
15:31:03:00	6700 - 6920	5	Turn right, climb on log, slope
15:31:36:00	9893 - 10089	3	Turn left, walk on log,
15:32:24:00	14816 - 15035	3	Turn right (180), walk descent from log,

Παράρτημα Γ – Κώδικας και αρχεία κινήσεων

Ο κώδικας και τα αρχεία της διπλωματικής αυτής εργασίας έχουν ανεβεί είτε σε ανοικτές προς το πλήθος πλατφόρμες κοινοποίησης κώδικα (π.χ. GitLab) είτε στον διακομιστή (server) του πανεπιστημίου. Κάποια από τα έργα αυτά ωστόσο δεν είναι διαθέσιμα προς το κοινό παρά μόνο σε όσους έχουν δικαιώματα (υπεύθυνοι καθηγητές της διπλωματικής εργασίας)

Υλοποίηση αλγορίθμων

- Fabrik implementation
 - <https://gitlab.com/mikhailkonstan/fabrik-implementation-for-reptiles-compgraphics>
- Tail fix for full and half marker set
 - <https://gitlab.com/mikhailkonstan/ucy-tail-fix-implementations>

Εφαρμογές

- DDMT-Viewer
 - <https://gitlab.com/mikhailkonstan/ddmt-viewer>
- Website
 - <https://gitlab.com/mikhailkonstan/3d-reptiles>

Αρχεία κινήσεων

- C3D, διορθωμένα αρχεία για την σαύρα
 - https://www.dropbox.com/s/k0hgvej93x5jsd9/fixed_lizard.zip?dl=0
- C3D, διορθωμένα αρχεία για το φίδι
 - https://www.dropbox.com/s/s13xi0gnm3dpmi1/fixed_snake.zip?dl=0
- FBX, κινήσεις
 - <https://drive.google.com/drive/folders/1jpEpHIEI2rIVdaAB72hYVos-WXjzwQ1R?usp=sharing>