



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ**  
**ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ**

**«Σχεδίαση καινοτόμου διεπαφής διάδρασης για ιατρικές εφαρμογές, μελέτη περίπτωσης: ανθρώπινος εγκέφαλος»**

**Σφίγγα Ειρήνη – Ευθαλία**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Υπεύθυνος**

**Κακαρούντας Αθανάσιος**

**Λαμία, 2018**





**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ**  
**ΒΙΟΙΑΤΡΙΚΗ**

**«Σχεδίαση καινοτόμου διεπαφής διάδρασης για ιατρικές εφαρμογές, μελέτη περίπτωσης: ανθρώπινος εγκέφαλος»**

**Σφίγγα Ειρήνη – Ευθαλία**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Υπεύθυνος**

**Κακαρούντας Αθανάσιος**

**Λαμία, 2018**

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις <sup>(1)</sup>, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία:07/02/2018

Η δηλούσα

Σφίγγα Ειρήνη Ευθαλία

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.

**«Σχεδίαση καινοτόμου διεπαφής διάδρασης για ιατρικές εφαρμογές, μελέτη περίπτωσης: ανθρώπινος εγκέφαλος»**

Σφίγγα Ειρήνη - Ευθαλία

**Τριμελής Επιτροπή:**

Αθανάσιος Κακαρούντας, Επίκουρος Καθηγητής (επιβλέπων)

Δημήτριος Ιακωβίδης, Αναπληρωτής Καθηγητής

Χαρίλαος Σανδαλίδης, Επίκουρος Καθηγητής



## Περίληψη

Η συνεχής και ραγδαία ανάπτυξη της τεχνολογίας είναι κύριο χαρακτηριστικό της εποχής μας. Σημαντικοί τομείς, όπως ο εκπαιδευτικός αλλά και ο ιατρικός τομέας έχουν εξελιχθεί και εκσυγχρονιστεί.

Μελετώντας υφιστάμενες τεχνολογίες και εφαρμογές διαπιστώθηκε ότι παρόλο που οι δυο αυτοί τομείς συνεχώς αναπτύσσονται δεν υπάρχει ταυτόχρονη διασύνδεση τους και ο χρήστης δεν έχει τη δυνατότητα να αλληλεπιδράσει φυσικά με το σύστημα. Για το λόγο αυτό δόθηκε ως κίνητρο η ανάπτυξη και η υλοποίηση μιας καινοτόμου διεπαφής μέσω της οποίας ο χρήστης θα μπορεί να αλληλεπιδράσει με το σύστημα μέσω των φυσικών κινήσεων του χεριού του. Για την διεπαφή μελετήθηκαν δύο τεχνολογίες αισθητήρων μια του Microsoft Kinect και μια του Leap Motion. Επιλέχθηκε ο αισθητήρας του Leap Motion καθώς ήταν ο καταλληλότερος για την αναγνώριση των χειρονομιών.

Παράλληλα με την υλοποίηση της διεπαφής αναπτύχθηκε και μια εφαρμογή στην πλατφόρμα του Unity, η οποία χωρίζεται στο εκπαιδευτικό και στο ιατρικό κομμάτι. Στο εκπαιδευτικό κομμάτι, υπάρχει ενσωματωμένο ένα τρισδιάστατο μοντέλο του εγκεφάλου, το οποίο είναι δομημένο σε οκτώ περιοχές οι οποίες έχουν αναγνωριστεί ξεχωριστά. Με ένα απλό πάτημα του κουμπιού ο χρήστης εισέρχεται σε όποια περιοχή επιθυμεί με στόχο να την μελετήσει ανατομικά αλλά και να διαβάσει κάποιες βασικές πληροφορίες για αυτή. Το ιατρικό κομμάτι δίνει τη δυνατότητα καταχώρησης ασθενών στο σύστημα και επιλογή τους από μια λίστα καταχωρήσεων. Παρέχεται η δυνατότητα προσθήκης, επεξεργασίας και ενημέρωσης του ιατρικού ιστορικού του ασθενούς.

**Λέξεις κλειδιά:** διεπαφή, αισθητήρας αναγνώρισης χειρονομιών, τρισδιάστατο μοντέλο εγκεφάλου, ιατρικό ιστορικό ασθενούς

## Abstract

The continuous and rapid development of technology is a major feature of our time. Important areas such as the academic and the medical sector have evolved and modernized.

By studying existing technologies and applications it has been found that although these two domains are constantly being developed, there is no simultaneous interconnection and the user is not able to interact with the system. For this reason, it was motivated to develop and implement an innovative interface through which the user can interact with the system through the natural movements of his hand. For the interface, two sensor technologies were studied: the Microsoft Kinect and the Leap Motion. The Leap Motion sensor was chosen as it was the most suitable for recognizing gestures.

Along with the implementation of the interface, an application was developed on the Unity platform, which is divided into the educational and medical parts. In the training part, there is a three-dimensional model of the brain, which is structured in eight areas that have been identified separately. With a simple click of a button, the user enters any region he wants to study anatomically, but also read some basic information about it. The medical part allows the patient to be registered and selected from a list of entries. It is possible to add, edit and update the medical history of the patient.

**Key words:** interface, gesture recognition sensor, three dimensional brain model, patient medical history



## Ευχαριστίες

Με την ολοκλήρωση της πτυχιακής μου εργασίας, ολοκληρώνεται και ο κύκλος των προπτυχιακών μου σπουδών. Για το λόγο αυτό θα ήθελα να εκφράσω την ευγνωμοσύνη μου σε όλους όσους με στήριξαν και μου έδωσαν το κίνητρο και τη δύναμη να συνεχίζω παρά τις εκάστοτε αντιξοότητες.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή και μέντορα μου, τον κύριο Αθανάσιο Κακαρούντα, τόσο για την στήριξη και την εμπιστοσύνη που μου έδειξε με την ανάθεση της παρούσας πτυχιακής εργασίας όσο και για την καθοδήγηση σε όλη τη διάρκεια της συνεργασίας μας αλλά και των ακαδημαϊκών μου χρόνων.

Επίσης, θα ήθελα να ευχαριστήσω την οικογένεια μου, για την υποστήριξη, την συμπαράσταση και την κατανόηση που έχει δείξει ειδικά κατά τη διάρκεια εκπόνησης της πτυχιακής μου εργασίας, αλλά και σε όλη τη διάρκεια της ζωής μου αποτελεί στήριγμα και αναπόσπαστο κομμάτι.

Τέλος, θα ήθελα να ευχαριστήσω τους φίλους μου για τη συμπαράσταση τους, αλλά και όλους τους καθηγητές του τμήματος Πληροφορικής με εφαρμογές στη βιοϊατρική, καθώς ο καθένας τους ξεχωριστά με ενέπνευσε, με τροφοδότησε με γνώσεις και εμπειρίες που θα με καθοδηγούν.

## Περιεχόμενα

Περίληψη .....	i
Abstract .....	ii
Ευχαριστίες .....	iii
Ευρετήριο Εικόνων .....	v
Ευρετήριο Πινάκων .....	viii
1. Εισαγωγή.....	1
1.1 Κίνητρο, σκοπός και στόχος πτυχιακής εργασίας .....	1
2. Έννοια διεπαφής .....	3
2.1 Έννοια φυσικής διεπαφής χρήστη (Natural User Interface, NUI).....	3
2.2 Έρευνα για διεπαφές σε υπάρχουσες ιατρικές εφαρμογές .....	3
2.2.1 Αποτελέσματα έρευνας και ανάλυση τους .....	4
2.2.2 Συγκεντρωτικός πίνακας χαρακτηριστικών .....	6
3. Τεχνολογίες ανίχνευσης της κίνησης.....	7
3.1 Τεχνολογία του Microsoft Kinect .....	7
3.1.1 Kinect Sensor Hardware .....	8
3.1.2 Skeleton Tracking .....	12
3.1.3 Απαιτήσεις συστήματος για χρήση του Kinect.....	13
3.2 Τεχνολογία του Leap Motion.....	14
3.2.1 Leap Motion Hardware .....	15
3.2.2 Λογισμικό Leap Motion.....	16
3.2.3 Ανίχνευση χειρονομιών από τον αισθητήρα Leap Motion.....	20
3.2.4 Απαιτήσεις συστήματος για τη χρήση Leap Motion .....	22
3.3 Σύγκριση αισθητήρων Microsoft Kinect – Leap Motion .....	22
4. Ανάλυση απαιτήσεων .....	25
4.1 Απαιτήσεις συστήματος.....	25
4.2 Απαιτήσεις οργάνωσης συστήματος.....	27
4.3 Διάγραμμα περιπτώσεων χρήσης.....	27
5. Υλοποίηση της διεπαφής .....	31
5.1 Πλατφόρμα του Unity.....	31
5.2 Βήματα υλοποίησης.....	32

5.2.1 Επιλογή τρισδιάστατου μοντέλου.....	32
5.2.2 Εισαγωγή μοντέλου στο Unity.....	34
5.2.3 Έννοια Game Object.....	36
5.2.4 Επικοινωνία Leap Motion – Unity.....	38
5.2.5 Βιβλιοθήκες κώδικα.....	40
5.2.6 Δημιουργία User Interface (UI) .....	41
5.2.7 CSV file (Comma – Separated Values) .....	45
5.2.8 Δημιουργία προσωπικού φακέλου ασθενούς.....	48
5.2.9 Προσωρινός φάκελος αποθήκευσης (Temporary File) .....	50
5.2.10 Τελική οργάνωση διεπαφής.....	51
6. Ανάλυση εφαρμογής.....	61
6.1 Αναγνωρισμένες χειρονομίες.....	62
7. Συμπεράσματα και μελλοντικές επεκτάσεις.....	64
7.1 Συμπεράσματα .....	64
7.2 Μελλοντικές επεκτάσεις.....	64
ΠΑΡΑΡΤΗΜΑ Α.....	66
ΠΑΡΑΡΤΗΜΑ Β.....	71
Βιβλιογραφία .....	<b>Error! Bookmark not defined.</b>

## Ευρετήριο Εικόνων

Εικόνα 1: Χρήση συστήματος TedCas	4
Εικόνα 2: Αλληλεπίδραση με το Cyber Science – Motion	5
Εικόνα 3: Παράδειγμα εφαρμογής	5
Εικόνα 4: Microsoft Kinect Sensor	8
Εικόνα 5: Εσωτερικό αισθητήρα Kinect	9
Εικόνα 6: Έγχρωμη κάμερα	10
Εικόνα 7: Παράδειγμα λειτουργίας αισθητήρα βάθους	11
Εικόνα 8: Σειρά μικροφώνων	11
Εικόνα 9: Μορφή επιταχυνσιόμετρου	12
Εικόνα 10: Μηχανισμός κλίσης και περιστροφής	12
Εικόνα 11: Αναγνώριση είκοσι σκελετικών αρθρώσεων σε όρθια στάση	13
Εικόνα 12: Αισθητήρας Leap Motion	14

Εικόνα 13: Χώρος αλληλεπίδρασης Leap Motion	15
Εικόνα 14: Παράδειγμα χειρονομίας μοντέλου χεριού	16
Εικόνα 15: Καρτεσιανό σύστημα συντεταγμένων	17
Εικόνα 16: Μοντέλο χεριού	18
Εικόνα 17: Μοντέλο δαχτύλων	19
Εικόνα 18: Σκελετικό σύστημα	19
Εικόνα 19: Circle gesture	20
Εικόνα 20: Swipe gesture	21
Εικόνα 21: Key tap gesture	21
Εικόνα 22: Screen tap gesture	22
Εικόνα 23: Σύμβολο περίπτωσης χρήσης	27
Εικόνα 24: Σύμβολο χειριστή	28
Εικόνα 25: Σχέση χειριστή με μια περίπτωση χρήσης	28
Εικόνα 26: Σύμβολο συμπερίληψης	29
Εικόνα 27: : Η περίπτωση χρήσης 2 επεκτείνεται από την περίπτωση χρήσης	29
Εικόνα 28: Περίπτωση χρήσης του χρήστη	30
Εικόνα 29: Σύνολο συσκευών και λογισμικών	32
Εικόνα 30: Πίνακας χαρακτηριστικών	34
Εικόνα 31: Δημιουργία πρότζεκτ στο Unity	35
Εικόνα 32: Σωστή τοποθέτηση μοντέλου απέναντι από την κάμερα(κόκκινο πλαίσιο)	35
Εικόνα 33: Εισαγωγή χρωματικού χάρτη σε κάθε δομή του μοντέλου ξεχωριστά	36
Εικόνα 34: Components του εγκεφάλου ως game object	37
Εικόνα 35: Επιλογή HandController	38
Εικόνα 36: Ενσωμάτωση και ρύθμιση του HandController στη σκηνή	39
Εικόνα 37: Τα χέρια αναγνωρίζονται επιτυχώς από τον αισθητήρα	40
Εικόνα 38: : Για διαχείριση εναλλαγής σκηνών	41
Εικόνα 39: Για πρόσβαση στο User Interface και τα στοιχεία του, διαχείριση εντολών συστήματος.	41
Εικόνα 40: Για πρόσβαση στις λειτουργίες του αισθητήρα	41
Εικόνα 41: : Εμφάνιση των UI στοιχείων	42
Εικόνα 42: Σκηνή εισόδου στην εφαρμογή	43
Εικόνα 43: Build Settings	44
Εικόνα 44: Καταχώρηση από την φόρμα της εφαρμογής στο αρχείο CSV	45
Εικόνα 45: Μορφή αρχείου CSV	47
Εικόνα 46: Εμφάνιση δεδομένων από το προσωπικό αρχείο CSV του ασθενούς	47
Εικόνα 47: Παρουσίαση εγγραφών μέσω της εφαρμογής στο CSV με την ανάλογη μορφή	48
Εικόνα 48: Με το «ADD» γίνεται η προσθήκη του ασθενούς στο σύστημα και η δημιουργία του φακέλου του. Μέσα στον φάκελο δημιουργείται το αρχείο για την καταγραφή του ιστορικού με όνομα «DateCSV» και το αρχείο name.txt.	49
Εικόνα 49: Εμφάνιση πληροφορίας από το name.txt	50

Εικόνα 50: Δημιουργία προσωρινού φακέλου	51
Εικόνα 51: Σχεδιάγραμμα σύνδεσης σκηνών	52
Εικόνα 52: Έναρξη με το πάτημα του κουμπιού Start	52
Εικόνα 53: Σε αυτό το σημείο ο χρήστης επιλέγει αν θα εξετάσει τη λειτουργικότητα του αισθητήρα ή όχι. (Scene 02)	53
Εικόνα 54: Σκηνή ελέγχου. Αν καταφέρει να αλληλοεπιδράσει με τους κύβους θα είναι ορθή η λειτουργικότητα του αισθητήρα αν όχι θα οδηγηθεί σε ιστοσελίδα υποστήριξης του προβλήματος. (Scene 15)	53
Εικόνα 55: : Μενού επιλογής για Medical ή Academic (Scene 03)	54
Εικόνα 56: Είσοδος στο Academic. Παρουσιάζεται το τρισδιάστατο μοντέλο του εγκεφάλου με τις αναγνωρισμένες περιοχές στις οποίες μπορεί να εισέλθει ο χρήστης. Υπάρχει και το εικονίδιο επιστροφής στο αρχικό μενού. (Scene 05)	54
Εικόνα 57: : Ενδεικτικά είσοδος στην περιοχή της παρεγκεφαλίδας. Δεξιά υπάρχουν οι πληροφορίες μαζί με το εικονίδιο «More...» για μετάβαση στην διαδικτυακή πηγή. Αριστερά περιστρέφεται μεμονωμένα το αντίστοιχο τμήμα και υπάρχει και το εικονίδιο επιστροφής στην προηγούμενη σκηνή επιλογής τμήματος του εγκεφάλου. (Scene 11)	55
Εικόνα 58: Ο χρήστης εισάγει είτε κάποιον νέο ασθενή είτε επιλέγει από τη λίστα (Scene 16)	55
Εικόνα 59: Προσθήκη νέου ασθενή και εμφάνιση φόρμας για την καταχώρηση του στο σύστημα. Με το «ADD» γίνεται η προσθήκη του στο σύστημα και η δημιουργία του φακέλου του. Μέσα στον φάκελο δημιουργείται το αρχείο για την καταγραφή του ιστορικού με όνομα «DateCSV» και το αρχείο name.txt. (Scene 17)	56
Εικόνα 60: Έχει τη δυνατότητα να επιλέξει ή να διαγράψει κάποιον ασθενή. Επίσης, μπορεί να επιστρέψει πίσω. (Scene 18)	56
Εικόνα 61: Το επόμενο βήμα είναι να εισάγει τον χρήστη στο χώρο επεξεργασίας του ιστορικού του ασθενούς. Λόγω ότι ο ασθενής είναι νέος δεν υπάρχει κάποια καταγραφή στο αρχείο των ημερομηνιών. (Scene 04)	57
Εικόνα 62: Εισαγωγή νέας εγγραφής στο ιστορικό όπως παρουσιάζονται σε βήματα. 1) επιλογή εικονιδίου για νέα προσθήκη, 2) είσοδος δεδομένων, 3) αποθήκευση εγγραφής.	57
Εικόνα 63: 4) Η λίστα των ημερομηνιών ανανεώθηκε αυτόματα και στα δεξιά εμφανίζονται τα περιεχόμενα της επιλεγμένης ημερομηνίας. Εφόσον υπάρχει αποθηκευμένη ημερομηνία ο χρήστης μπορεί να τροποποιήσει κάποια από τα σχόλια που τη συνοδεύουν.	58
Εικόνα 64: 5) Παρουσιάζεται το εικονίδιο για την επεξεργασία του σχόλιου της επιλεγμένης ημερομηνίας και 6) η νέα τροποποιημένη και αποθηκευμένη καταγραφή στο ιστορικό	58
Εικόνα 65: 1) Επιλογή ημερομηνίας, 2) επιλογή εικονιδίου διαγραφής, 3) ανανεωμένη λίστα των ημερομηνιών ιστορικού.	59

Εικόνα 66: 1) Επιστροφή στην προηγούμενη σκηνή, 2) ενημέρωση του αρχείου καταγραφών ημερομηνιών και άνοιγμα του για επαλήθευση της εργασίας που προηγήθηκε και τέλος συμβαίνει και η διαγραφή του TMP	60
Εικόνα 67: Μορφή αναγνωρισμένων χειρονομιών	63
Εικόνα 68: Χειρονομία pinch	63

## **Ευρετήριο Πινάκων**

Πίνακας 1: Δυνατότητες υφιστάμενων εφαρμογών	6
Πίνακας 2: Σύγκριση των αισθητήρων Microsoft Kinect – Leap Motion	22

## 1. Εισαγωγή

Κύριο χαρακτηριστικό της σύγχρονης εποχής στην οποία ζούμε είναι η ραγδαία ανάπτυξη και χρήση της τεχνολογίας. Παράλληλα με την ανάπτυξη της τεχνολογίας αναπτύσσονται οι υπηρεσίες και τα μέσα τα οποία χρησιμοποιεί, ώστε το αποτέλεσμα να είναι το βέλτιστο. Οι εξελιγμένες υπηρεσίες της τεχνολογίας έχουν οφέλη τόσο στην καθημερινή ζωή των πολιτών όσο και στους τομείς της εκπαίδευσης, της ιατρικής, της έρευνας και κυρίως σε όλη την επιστημονική κοινότητα.

Συγκεκριμένα στον τομέα της εκπαίδευσης έχουν γίνει σημαντικές και πρωτοποριακές εξελίξεις. Έχουν δημιουργηθεί πλατφόρμες για εκπαίδευση εξ ‘αποστάσεως, δυνατότητα διασύνδεσης με σχολεία και πανεπιστήμια σε όλο τον κόσμο αλλά με το πάτημα ενός κουμπιού καθώς και καινοτόμες εφαρμογές εκπαίδευσης.

Επιπλέον, η τεχνολογία έδωσε τη δυνατότητα στην ιατρική επιστήμη να κάνει πρωτοποριακά και σημαντικά άλματα. Χαρακτηριστικό παράδειγμα είναι η εξέλιξη και ο εκσυγχρονισμός των ιατρικών μηχανημάτων. Ακόμα, έχουν αναπτυχθεί τεχνολογίες όπως η τηλεϊατρική, η ρομποτική ιατρική, τεχνολογίες απεικόνισης και αναγνώρισης ασθενειών, καθώς και η μηχανική των ιστών και η αναγεννητική ιατρική.

### 1.1 Κίνητρο, σκοπός και στόχος πτυχιακής εργασίας

Παρόλα αυτά τα τεχνολογικά επιτεύγματα, έπειτα από μελέτη και έρευνα αγοράς, διαπιστώθηκε ότι υπάρχουν μερικά τεχνολογικά κενά στην εκπαίδευση και την ιατρική.

Αρχικά, ιδιαίτερα στον τομέα της ιατρικής εκπαίδευσης υπάρχει έλλειψη της τεχνολογίας. Οι φοιτητές ακόμα παρακολουθούν το μάθημα θεωρητικά δίχως παραστατικότητα, καθιστώντας την εκμάθηση της ανθρώπινης ανατομίας ασαφής και αφηρημένη. Ακόμα, γίνεται χρήση ανθρώπινων πτωμάτων για την πρώτη επαφή των φοιτητών της ιατρικής με το ανθρώπινο σώμα, φέρνοντας τους πολλές φορές σε άβολες καταστάσεις.

Επιπρόσθετα, το ιατρικό ιστορικό του ασθενούς αποτελεί αναπόσπαστο στοιχείο για τη μελέτη και το σχεδιασμό θεραπειών που θα ακολουθήσει ο θεράπωντας ιατρός. Έχουν γίνει βήματα με τη χρήση του ηλεκτρονικού ιατρικού φακέλου, όμως η ενημέρωση του και η χρήση του πολλές φορές είναι δύσχρηστη και χρονοβόρα διαδικασία για το ιατρικό προσωπικό. Ειδικά, οι ιδιωτικοί ιατροί δυσκολεύονται ιδιαίτερα στην καταχώρηση των στοιχείων του ιατρικού φακέλου, καθώς η διεπαφή

χρήστη είναι πολύπλοκη και πολλές φορές αδυνατούν να βρουν την επιθυμητή επιλογή.

Επομένως, υπάρχει ανάγκη για μια τεχνολογική δημιουργία, η οποία θα υποστηρίζει και την εκπαίδευση του χρήστη και την ενημέρωση στοιχείων του ασθενούς μέσω μιας διεπαφής χρήστη που πρέπει να είναι ευέλικτη, γρήγορη και να μην απαιτεί συγκεκριμένη γνώση και εξειδίκευση για τη χρήση της.

Μελετώντας και αναλύοντας τα προβλήματα που προαναφέρθηκαν, στόχος της συγκεκριμένης πτυχιακής εργασίας είναι η δημιουργία και η σχεδίαση μιας καινοτόμου διεπαφής για ιατρικές εφαρμογές. Ο ανθρώπινος εγκέφαλος και συγκεκριμένα ένα τρισδιάστατο μοντέλο του εγκεφάλου θα χρησιμοποιηθεί ως αντικείμενο αλληλεπίδρασης, μελέτης και ανάπτυξης της διεπαφής.

Η καινοτόμος διεπαφή που υλοποιείται στην συγκεκριμένη πτυχιακή εργασία θα επιτρέπει την αμφίδρομη επικοινωνία του συστήματος με τον χρήστη. Τόσο εκπαιδευτικοί, φοιτητές, απλό και μη εξειδικευμένο κοινό, όσο και ιατρικό προσωπικό θα έχουν τη δυνατότητα να κατανοήσουν μέσω μιας εικονικής επαφής την ανθρώπινη ανατομία να τη μελετήσουν και να εμβαθύνουν σε συγκεκριμένα σημεία. Ο σχεδιασμός της διεπαφής βασίζεται στο λογισμικό που έχει δημιουργηθεί με τη χρήση της πλατφόρμας του Unity και τη χρήση ενός εξωτερικού αισθητήρα αναγνώρισης χειρονομιών του Leap Motion Sensor.

Σκοπός είναι να αποτελέσει ένα εύχρηστο, ευέλικτο και προσιτό τεχνολογικό εργαλείο, ώστε να διευκολύνει τα σημεία που ακόμα αντιμετωπίζουν σημαντικές ελλείψεις.

Η πτυχιακή εργασία έχει δομηθεί σύμφωνα με τα παρακάτω. Στο κεφάλαιο 2 γίνεται μια σύντομη ανάλυση της έννοιας της διεπαφής και παρουσιάζεται η έρευνα που πραγματοποιήθηκε για διεπαφές σε υπάρχουσες ιατρικές εφαρμογές. Στο κεφάλαιο 3 αναλύονται και παρουσιάζονται τεχνολογίες ανίχνευσης της κίνησης. Στο κεφάλαιο 4 γίνεται ανάλυση των προδιαγραφών του συστήματος και τι ιδιότητες προσφέρει. Στο κεφάλαιο 5 παρουσιάζεται η ανάπτυξη και η υλοποίηση του συστήματος της πτυχιακής εργασίας. Στο κεφάλαιο 6 αναλύονται τα χαρακτηριστικά της εφαρμογής. Στο κεφάλαιο 7 γίνεται η διατύπωση των συμπερασμάτων και αναφορά στις μελλοντικές επεκτάσεις. Τέλος, στο παράρτημα Α αναλύεται εν συντομία ο εγκέφαλος, οι λειτουργίες του και τα μέρη που χρησιμοποιήθηκαν στην πτυχιακή εργασία.



## 2. Έννοια διεπαφής

Με τον όρο διεπαφή (interface) ορίζεται το σύνορο επικοινωνίας μιας οντότητας, για παράδειγμα ενός χρήστη ή ενός λογισμικού, με το περιβάλλον της. Κάθε οντότητα θεωρείται ότι μπορεί να εκτελέσει ένα σύνολο λειτουργιών. Συγκεκριμένα, η αμφίδρομη επικοινωνία που προσφέρει ένας υπολογιστής σε κάποιο χρήστη ονομάζεται διεπαφή χρήστη (User Interface, UI). Οι διεπαφές που πραγματοποιούνται μεταξύ ενός υπολογιστικού συστήματος και ενός ή πολλών χρηστών ονομάζονται φυσικές διεπαφές, ενώ εκείνες οι διεπαφές που γίνονται μεταξύ διαφορετικών τμημάτων λογισμικού ονομάζονται διεπαφές λογισμικού. [1]

### 2.1 Έννοια φυσικής διεπαφής χρήστη (Natural User Interface, NUI)

Για να γίνει κατανοητή η έννοια της φυσικής διεπαφής χρήστη είναι αξιόλογο να αναφερθεί η περιγραφή που δίνει ο Bill Buxton, ένας από τους πρωτοπόρους ερευνητές στην τεχνολογία και στο χώρο των φυσικών διεπαφών:

«Μια διεπαφή θεωρείται ως φυσική όταν αυτή εκμεταλλεύεται τις δεξιότητες που έχουν αποκτήσει οι άνθρωποι μέσα από την εμπειρία της ζωής τους στο φυσικό κόσμο.»

Ιδανικά οι διεπαφές ανθρώπου – υπολογιστή θα έπρεπε να είναι φυσικές διεπαφές, καθώς με αυτόν τον τρόπο θα μειωνόταν σημαντικά ο χρόνος αποδοχής από τον χρήστη, αλλά και ο χρόνος εκπαίδευσης του χρήστη. Βασικό εμπόδιο για την ανάπτυξη φυσικών διεπαφών είναι η απουσία αισθητήρων που ανιχνεύουν τα ανθρώπινα φυσιολογικά χαρακτηριστικά καθώς και τη συμπεριφορά. Παρόλα αυτά, πρόσφατες τεχνολογικές εφευρέσεις έχουν επιτρέψει την ανάπτυξη καινοτόμων χαμηλού κόστους αισθητήρων, οι οποίοι προσφέρουν τέτοιες δυνατότητες με παράλληλη διάθεση λογισμικού για ανάπτυξη κατάλληλων εφαρμογών.

### 2.2 Έρευνα για διεπαφές σε υπάρχουσες ιατρικές εφαρμογές

Προτού ξεκινήσει η ανάπτυξη και ο σχεδιασμός της διεπαφής που παρουσιάζεται στην παρούσα πτυχιακή εργασία, κρίθηκε σκόπιμο και αναγκαίο να πραγματοποιηθεί μια εκτενής έρευνα αγοράς, η οποία έχει συγκεντρώσει και παρουσιάζει αντίστοιχες υπάρχουσες διεπαφές για ιατρικές και εκπαιδευτικές εφαρμογές με χρήση του αισθητήρα Leap Motion.

### 2.2.1 Αποτελέσματα έρευνας και ανάλυση τους

Τα αποτελέσματα τα οποία προέκυψαν παρουσιάζονται παρακάτω εν συντομία με τα αντίστοιχα χαρακτηριστικά τους.

#### **TedCas**

Πρόκειται για μια ιατρική εφαρμογή, η οποία επιτρέπει την πρόσβαση και τη διαχείριση της ιατρικής πληροφορίας σε χώρους υγειονομικής περίθαλψης.

Το λογισμικό που αναπτύχθηκε επιτρέπει στο ιατρικό προσωπικό να αλληλοεπιδρά, μέσα σε ένα αποστειρωμένο περιβάλλον, με μια σειρά ιατρικών εικόνων και απεικονίσεων του ασθενή. Αυτό επιτυγχάνεται μόνο με τη χρήση χειρονομιών στον αέρα αναγνωρίσιμες από τον αισθητήρα του Leap Motion.

Η εφαρμογή του γίνεται ξεχωριστή κατά τη διάρκεια μιας χειρουργικής επέμβασης, όπου παρέχεται στους χειρουργούς όλη η πληροφορία που χρειάζονται μέσα στον αποστειρωμένο χώρο. [2]



*Εικόνα 1: Χρήση συστήματος TedCas*

#### **Cyber Science – Motion**

Η αλληλεπίδραση του χρήστη με το συγκεκριμένο λογισμικό του επιτρέπει να εξερευνεί να αναλύει και να συναρμολογεί ένα ανθρώπινο κρανίο. Έχει εκπαιδευτικό αλλά και ψυχαγωγικό χαρακτήρα. [3]

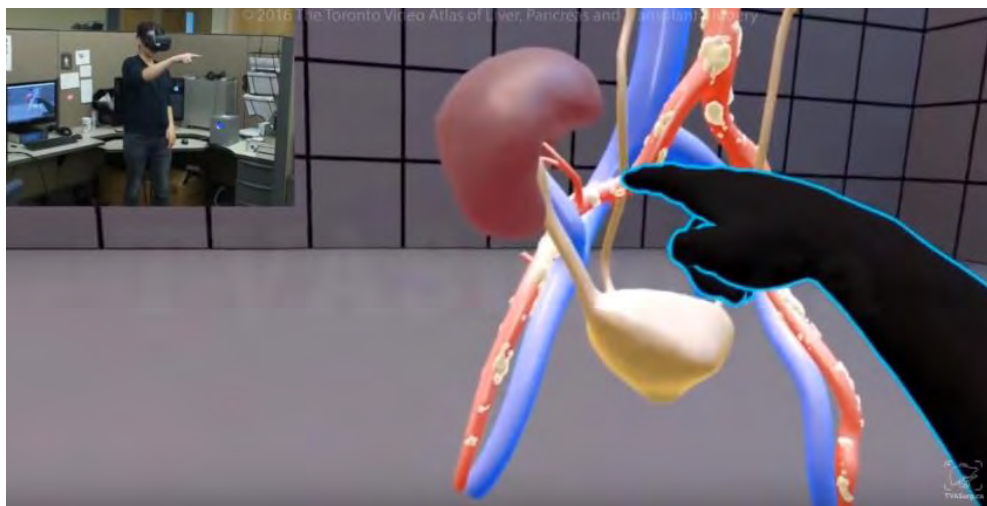


*Εικόνα 2: Αλληλεπίδραση με το Cyber Science – Motion*

### **TVASurg VR Surgical Anatomy Module**

Το συγκεκριμένο σύστημα έχει σχεδιαστεί ώστε να ανταποκρίνεται σε κάποια συγκεκριμένη και εξειδικευμένη περίπτωση χειρουργικής επέμβασης σε συνθήκες εικονικής πραγματικότητας.

Περιέχει τρισδιάστατα ανατομικά μοντέλα, σαρώσεις CT/MRI και αποσπάσματα βίντεο, επιτρέποντας στους χρήστες να διερευνήσουν το περιεχόμενο χρησιμοποιώντας μόνο τα χέρια τους μέσω του αισθητήρα Leap Motion. [4]



*Εικόνα 3: Παράδειγμα εφαρμογής*

### 2.2.2 Συγκεντρωτικός πίνακας χαρακτηριστικών

Στον παρακάτω πίνακα γίνεται η σύγκριση των χαρακτηριστικών της κάθε εφαρμογής που μελετήθηκε παραπάνω ως προς τις δυνατότητες και τα χαρακτηριστικά τους.

*Πίνακας 1: Δυνατότητες υφιστάμενων εφαρμογών*

	Εκπαιδευτικό Χαρακτήρα	Ιατρική Εφαρμογή	Αλληλεπίδραση Συστήματος – Χρήστη	Διαχείριση Αρχείων
TedCas	✗	✓	✓	✗
Cyber Science – Motion	✓	✗	✓	✗
TVASurg VR Surgical Anatomy Module	✓	✓	✓	✗

### 3. Τεχνολογίες ανίχνευσης της κίνησης

Η ανίχνευση της κίνησης είναι μια διαδικασία, η οποία αναγνωρίζεται με τη χρήση ειδικών αισθητήρων και βασίζεται στην αλλαγή της θέσης ενός αντικειμένου σε σχέση με το περιβάλλον του. Για την υλοποίηση της πτυχιακής εργασίας μελετήθηκαν δύο διαφορετικές τεχνολογίες ανίχνευσης της κίνησης. Η πρώτη βασίζεται στον αισθητήρα του Microsoft Kinect, ενώ η δεύτερη στον αισθητήρα του Leap Motion.

#### 3.1 Τεχνολογία του Microsoft Kinect

Ο αισθητήρας του Kinect είναι μια περιφερειακή συσκευή εισόδου ανίχνευσης της κίνησης, προϊόν της εταιρείας Microsoft που αναπτύχθηκε για τις κονσόλες βιντεοπαιχνιδιών Xbox 360 και Xbox One, καθώς και για υπολογιστές με λειτουργικό σύστημα Windows. Προσφέρει στον χρήστη μια φυσική διεπαφή αλληλεπίδρασης με την κονσόλα/υπολογιστή μόνο με την χρήση χειρονομιών, κινήσεων και προφορικών εντολών.

Η τεχνολογία του αισθητήρα Microsoft Kinect έγινε για πρώτη φορά ευρέως γνωστή στις 1 Ιουνίου 2009 με την κωδική ονομασία Project Natal. Ανέκαθεν η παράδοση της Microsoft είναι να δίνει ονόματα πόλεων ως κωδικά ονόματα σε μεγάλα πρότζεκτ. Για το λόγο αυτό και προς τιμήν του διευθυντή της Microsoft και εμπνευστή του συγκεκριμένου πρότζεκτ, Alex Kipman, δόθηκε το όνομα της πόλης Natal της Βραζιλίας όπου αποτελούσε και το τόπο καταγωγής του. Ένας ακόμα λόγος που χρησιμοποιήθηκε το όνομα natal είναι ότι σημαίνει και «γέννηση», δηλώνοντας την έναρξη μιας καινούργιας εποχής στον τομέα της ψυχαγωγίας. Το όνομα του τελικού προϊόντος που κυκλοφορεί στην αγορά μέχρι και σήμερα με το ίδιο όνομα είναι Kinect. Το όνομα αυτό έχει τις ρίζες του στις λέξεις kinetic (εκφράζει κινητικότητα) και nexus (ένα είδος σύνδεσης) ή connect (σύνδεση). [5]

Την επίσημη εμφάνιση στο αγοραστικό κοινό της Αμερικής έκανε το Kinect τον Νοέμβριο του 2010. Στις 17 Ιουνίου 2011 η Microsoft ανακοίνωσε την επίσημη κυκλοφορία του Kinect SDK για μη εμπορική χρήση. Το Kinect SDK για Windows είναι ένα σύνολο από «βιβλιοθήκες» που επιτρέπουν χάρη στα εργαλεία που περιλαμβάνουν, την ανάπτυξη και υλοποίηση ποικίλων εφαρμογών, χρησιμοποιώντας προγραμματιστικές πλατφόρμες της Microsoft. Η υλοποίηση των εφαρμογών πραγματοποιείται με τις προγραμματιστικές γλώσσες C++/CLI, C# ή Visual Basic.NET.

Την 31 Οκτωβρίου 2011, η Microsoft ανακοίνωσε την έναρξη της εμπορικής έκδοσης του SDK για το λειτουργικό σύστημα των Windows μόνο για τις τεχνολογικές εταιρείες.

Στις 1 Φεβρουαρίου 2012, κυκλοφόρησε η εμπορική έκδοση του Kinect SDK για Windows.

Τελικά, στις 16 Ιουνίου 2011, η Microsoft προχώρησε στην κυκλοφορία εμπορικού πακέτου ανάπτυξης λογισμικού (Kinect SDK) για τα Windows, παρέχοντας την δυνατότητα στους προγραμματιστές να έχουν πρόσβαση σε όλα τα εργαλεία που χρησιμοποιεί η ίδια η Microsoft για ανάπτυξη εφαρμογών που λαμβάνουν ως είσοδο τον αισθητήρα του Kinect. [6] [7]



*Εικόνα 4: Microsoft Kinect Sensor*

### **3.1.1 Kinect Sensor Hardware**

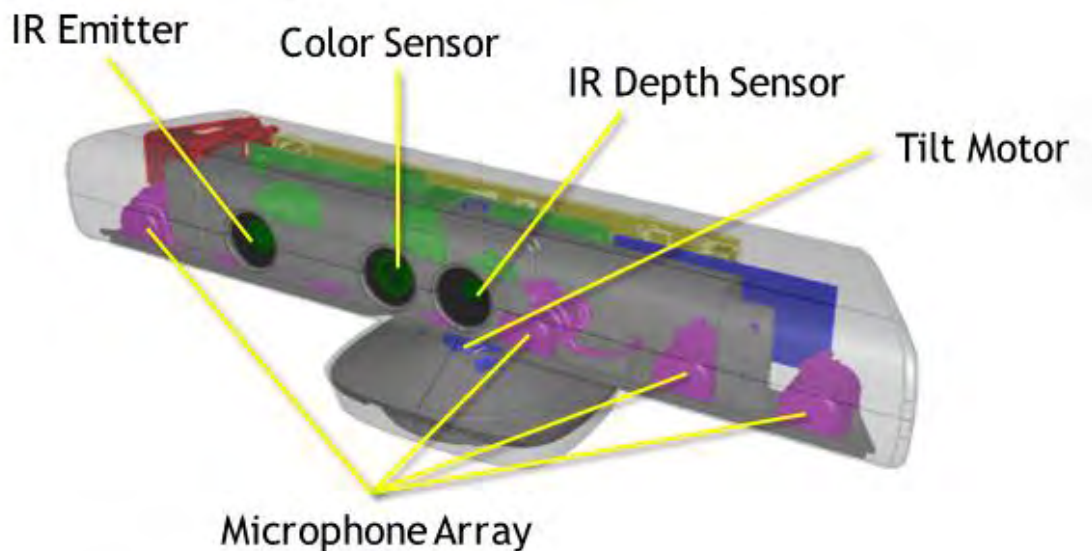
Όπως προαναφέρθηκε προηγουμένως ο συγκεκριμένος αισθητήρας αναγνώρισης κίνησης είναι μια περιφερειακή συσκευή, η οποία λειτουργεί παρόμοια με μια βιντεοκάμερα. Η συσκευή διαθέτει μια κάμερα χρώματος (RGB), έναν αισθητήρα βάθους και ένα μικρόφωνο πολυ-συστοιχίας. Ο συνδυασμός τους επιτρέπει τη σύλληψη της τρισδιάστατης κίνησης ολόκληρου του σώματος, την αναγνώριση προσώπου όπως και δυνατότητες αναγνώρισης φωνής.

Για κάθε pixel της εικόνας που ανιχνεύεται από τον αισθητήρα του Kinect γίνεται ταυτόχρονα και υπολογισμός της απόστασης του pixel που έχει ανιχνευτεί από τη θέση του αισθητήρα. Η λειτουργία αυτή βρίσκει πολλές εφαρμογές, όπως για παράδειγμα την ανίχνευση του σκελετού κίνησης (Skeleton Tracking) με χρήση εξωτερικών βιβλιοθηκών και προγραμμάτων.

Τα υλικά κομμάτια που συνθέτουν τον αισθητήρα του Kinect είναι:

- Αισθητήρας/Κάμερα χρώματος (Color Sensor).
- Ένας αισθητήρας βάθους υπέρυθρων (IR Depth Sensor) και ένας πομπός υπέρυθρων (IR Emitter).
- Σειρά μικροφώνων. (Multi-array microphone)

- Επιταχυνσιόμετρο (Accelerometer)
- Μηχανοκίνητη βάση (Tilt Motor)



*Εικόνα 5: Εσωτερικό αισθητήρα Kinect*

- **Αισθητήρας/Κάμερα χρώματος (Color Sensor).**

Η ενσωματωμένη κάμερα χρώματος σε συχνότητα 30Hz μπορεί να τραβήξει εικόνες με ανάλυση 640\*480 pixels έως μέγιστη ανάλυση 1280\*960Hz. Επιτυγχάνει την καλύτερη απόδοση στα 15 frames(καρέ) ανά δευτερόλεπτο. Η απόσταση στην οποία μπορεί να δουλέψει κυμαίνεται στα 1.2 μέτρα έως 3.5 μέτρα και μπορεί να σκανάρει μια σκηνή έως 3.8 μέτρα σε πλάτος.

Η ίδια η κάμερα διαθέτει χαρακτηριστικά, όπως η αυτόματη εξισορρόπηση των χρωμάτων και διόρθωση σφαλμάτων. Η εικόνα που παράγεται μπορεί να χρησιμοποιηθεί σε αλγόριθμους αναγνώρισης προσώπου, χειρονομιών ή σε οτιδήποτε άλλο.



*Εικόνα 6: Έγχρωμη κάμερα*

- **Αισθητήρας βάθους υπέρυθρων (IR Depth Sensor) & Πομπός υπέρυθρων (IR Emitter).**

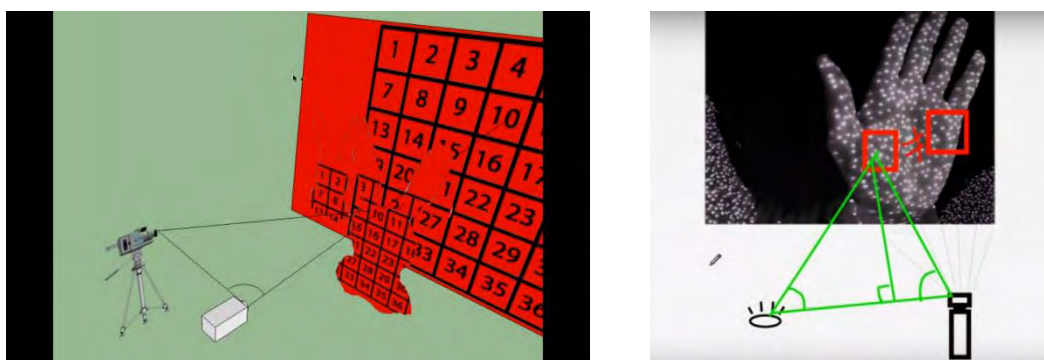
Ο αισθητήρας βάθους υπέρυθρων αποτελείται από έναν υπέρυθρο προβολέα λέιζερ σε συνδυασμό με έναν μονόχρωμο αισθητήρα CMOS, ο οποίος συλλέγει τα δεδομένα βίντεο σε τρισδιάστατη μορφή υπό οποιοδήποτε συνθήκες φωτισμού περιβάλλοντος, δημιουργώντας μια κάμερα βάθους. Μια κάμερα βάθους έχει τη δυνατότητα να δουλεύει σε συνθήκες χαμηλού φωτισμού κρατώντας αμετάβλητο το χρώμα και την υφή. Επίσης, χρησιμεύει στη διαδικασία αφαίρεσης του φόντου από ένα σκηνικό.

Αναλυτικότερα, σε μια τρισδιάστατη σκηνή προβάλλεται ένα τυχαίο σχέδιο υπέρυθρου φωτισμού, αποτελούμενο από φωτεινές κηλίδες, με τη χρήση ενός υπέρυθρου διαχυτή. Ο αισθητήρας βάθους παρατηρεί τη σκηνή. Η παραμόρφωση του σχεδίου φωτός που έχει δημιουργηθεί από την επιφάνεια στην οποία έχει προσπέσει επιτρέπει τη δυνατότητα κατασκευής και υπολογισμού μιας τρισδιάστατης δομής. Κάθε σημείο στη δομή που έχει προκύψει έχει μια μοναδική θέση στο χώρο.

Τα δεδομένα βάθους που προκύπτουν υπολογίζονται με τριγωνισμό κάθε φωτεινής κηλίδας μεταξύ του προβαλλόμενου στοιχείου και του παρατηρούμενου στοιχείου. Κάθε σημείο έχει μια φωτεινή κηλίδα. Έχοντας ένα βαθμονομημένο μοτίβο των φωτεινών κηλίδων, που δημιουργείται κατά την κατασκευή του υπέρυθρου σχεδίου φωτός, υπολογίζεται ο τρισδιάστατος χάρτης βάθους. Ο χάρτης βάθους χρησιμοποιείται για τη χαρτογράφηση τόσο κινούμενων όσο και σταθερών αντικειμένων.

Τα μεγέθη και το σχήμα των κηλίδων ποικίλλουν ανάλογα με την απόσταση και τον προσανατολισμό σε σχέση με τη θέση του αισθητήρα. Το Kinect χρησιμοποιεί τρία διαφορετικά μεγέθη κηλίδων και τρεις διαφορετικές περιοχές αποστάσεων.





*Εικόνα 7: Παράδειγμα λειτουργίας αισθητήρα βάθους*

- **Σειρά μικροφώνων (Multi-array microphone)**

Το Kinect αποτελείται από μια σειρά τεσσάρων μικροφώνων που δίνουν τη δυνατότητα να δέχεται ήχο και ταυτόχρονα να εντοπίζει και τη γωνία της πηγής. Ακόμα ελαχιστοποιεί τον θόρυβο. Κάθε ένα από τα τέσσερα κανάλια δέχονται ήχο 16-bit με συχνότητα δειγματοληψίας 16KHz.

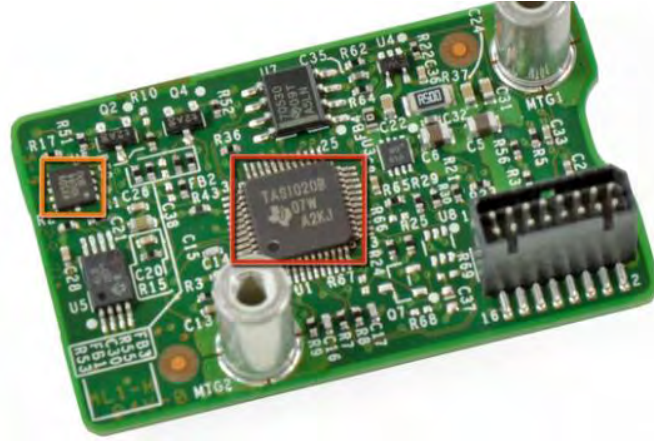


*Εικόνα 8: Σειρά μικροφώνων*

- **Επιταχυνσιόμετρο (Accelerometer)**

Είναι μια απλή συσκευή, η οποία μετρά την επιτάχυνση ενός σώματος. Στην περίπτωση του Kinect τα δεδομένα που συλλέγει ο αισθητήρας μπορούν να βοηθήσουν στην ανίχνευση λανθασμένου προσανατολισμού του αισθητήρα. Ακόμα, σε συνδυασμό με τα δεδομένα επίπεδου δαπέδου μπορεί να οδηγήσει σε πιο ακριβείς τρισδιάστατες δομές εικονικής πραγματικότητας.

Το επιταχυνσιόμετρο είναι τριών αξόνων (x,y,z) και παρέχει την πληροφορία για τη θέση της συσκευής. Οι τιμές των αξόνων x και y καθορίζουν την κύλιση και την κλίση, ενώ ο άξονας z καθορίζει αν ο αισθητήρας είναι σε ορθή ή ανάποδη θέση.



*Εικόνα 9: Μορφή επιταχυνσιόμετρου*

- **Μηχανοκίνητη βάση (Tilt Motor)**

Ο μηχανισμός κλίσης και περιστροφής του αισθητήρα του Kinect είναι αλληλένδετος με το επιταχυνσιόμετρο. Η λειτουργικότητα της βάσης έγκειται στο γεγονός ότι μπορεί να διατηρήσει τους χρήστες εντός του καρέ ακολουθώντας την κίνηση τους. Η μηχανοκίνητη βάση λειτουργεί με ένα σύνολο οδοντωτών τροχών, οι οποίοι όμως είναι ευαίσθητοι στη θερμότητα. [6] [7] [8] [9] [10] [11] [12]



*Εικόνα 10: Μηχανισμός κλίσης και περιστροφής*

### **3.1.2 Skeleton Tracking**

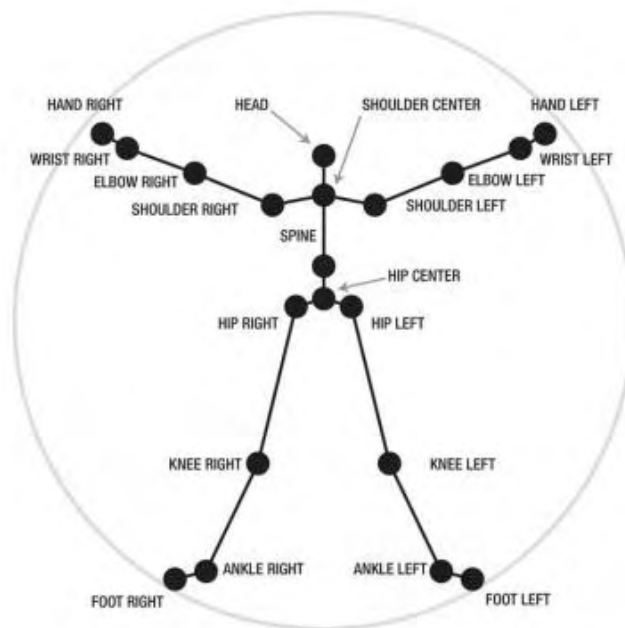
Έχοντας ως στόχο τη δημιουργία διαδραστικών, ψυχαγωγικών και μοναδικών εμπειριών με τη χρήση του αισθητήρα Kinect δεν είναι επαρκής μόνο οι πληροφορίες

---

που συλλέγονται από το βάθος και την απόσταση του κάθε pixel. Σε αυτό το σημείο γίνεται χρήση παρακολούθησης σκελετού (Skeleton Tracking), επιτρέποντας στο Kinect να αναγνωρίσει ανθρώπινα σώματα και να ανιχνεύσει και να παρακολουθήσει την κίνηση τους. Η παρακολούθηση σκελετού βασίζεται στην επεξεργασία των δεδομένων βάθους που έχουν προέλθει από τις θέσεις των διάφορων σκελετικών αρθρώσεων προερχόμενες από την ανθρώπινη ανατομική μορφή.

Για παράδειγμα, η παρακολούθηση σκελετού καθορίζει που είναι το κεφάλι, τα χέρια και το κέντρο μάζας του κάθε χρήστη. Παρέχει τιμές στους άξονες X, Y και Z για καθένα από το σύνολο των σημείων (skeleton positions) που αναγνωρίζονται.

Η λειτουργία του skeleton tracking έχει βελτιστοποιηθεί ώστε να μπορεί να αναγνωρίζει χρήστες σε όρθια στάση (αναγνώριση είκοσι αρθρώσεων) και σε καθιστή θέση (αναγνώριση 10 αρθρώσεων). Με τον τρόπο αυτό η χρήση των χειρονομιών λειτουργεί ως βασικό εργαλείο για τον καθορισμό εντολών του συστήματος από τους χρήστες. [6] [13] [14]



Εικόνα 11: Αναγνώριση είκοσι σκελετικών αρθρώσεων σε όρθια στάση

### 3.1.3 Απαιτήσεις συστήματος για χρήση του Kinect

- 64-bit (x64) επεξεργαστής
- USB 3.0 θήρα
- 4 GB RAM
- Κάρτα γραφικών που υποστηρίζει DirectX 11
- Windows 8 or 8.1, Windows Embedded 8, ή Windows 10 λειτουργικό σύστημα [15]

### 3.2 Τεχνολογία του Leap Motion

Ο αισθητήρας του Leap Motion είναι μια USB περιφερειακή συσκευή που υποστηρίζει την αναγνώριση και την παρακολούθηση των φυσικών κινήσεων των δαχτύλων και των χεριών στον τρισδιάστατο χώρο και σε πραγματικό χρόνο.

Η τεχνολογία του Leap Motion αναπτύχθηκε για πρώτη φορά το 2008 από τον David Holz. Στη συνέχεια σε συνεργασία με τον παιδικό του φίλο Michael Buckwald ίδρυσαν την εταιρεία παραγωγής της τεχνολογίας του Leap Motion. Ο αισθητήρας έκανε την πρώτη εμφάνιση του στη δημοσιότητα στις 21 Μαΐου 2012 με το όνομα The Leap. Το πρόγραμμα ανάπτυξης λογισμικού ξεκίνησε τον Οκτώβριο του 2012 με την παροχή 12,000 συσκευών από την εταιρεία σε προγραμματιστές που ενδιαφέρονταν να δημιουργήσουν εφαρμογές που να βασίζονται στη χρήση του αισθητήρα.

Επίσης, το Leap Motion κυκλοφόρησε για πρώτη φορά στην αγορά το Μάιο του 2013. Αντίστοιχα, το Μάιο του 2014 η εταιρεία ανέπτυξε την έκδοση 2.0 της διεπαφής προγραμματισμού εφαρμογών (application programming interface – API ) και υποστήριξη για εφαρμογές εικονικής πραγματικότητας. Τέλος, τον Φεβρουάριο του 2016 η εταιρεία έδωσε στη κυκλοφορία το νέο λογισμικό της με όνομα Orion, το οποίο είναι ειδικά διαμορφωμένο για εφαρμογές εικονικής πραγματικότητας.

Είναι αξιόλογο να αναφερθεί ότι η εταιρεία παραγωγής του Leap Motion έχει συνεργαστεί με την ASUS, καθώς και με την Hewlett Packard ώστε να ενσωματωθεί η τεχνολογία του αισθητήρα σε υπολογιστές HP. Η τεχνολογία ενσωματώθηκε τόσο σε υπολογιστές HP όσο και σε πληκτρολόγια και φορητούς υπολογιστές της ίδιας εταιρείας. [16]

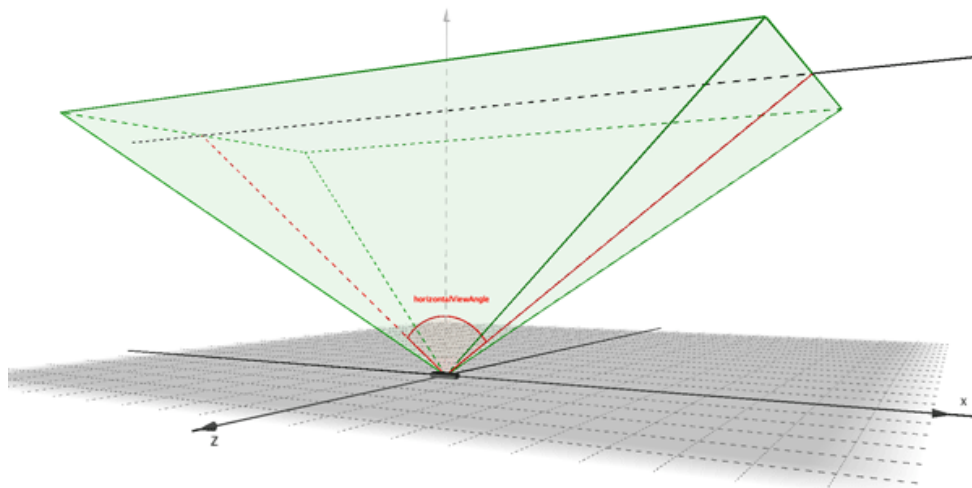


*Εικόνα 12: Αισθητήρας Leap Motion*

### 3.2.1 Leap Motion Hardware

Ο αισθητήρας του Leap Motion χάρη στο μικρό του μέγεθος μπορεί να τοποθετηθεί σε μια επίπεδη επιφάνεια μπροστά από τον χρήστη ή ακόμα να ενσωματωθεί σε κάσκα εικονικής πραγματικότητας.

Η υλική κατασκευή του αισθητήρα Leap Motion στην πραγματικότητα είναι αρκετά απλή. Η «καρδιά» της συσκευής αποτελείται από δύο υπέρυθρες μονοχρωματικές κάμερες και τρία υπέρυθρα λαμπάκια LED. Οι κάμερες εντοπίζουν το υπέρυθρο φως με μήκος κύματος 850 νανομέτρων, το οποίο βρίσκεται εκτός του ορατού φάσματος του φωτός. Χρησιμοποιώντας ευρυγώνιους φακούς, ο αισθητήρας έχει ένα μεγάλο χώρο αλληλεπίδρασης των οκτώ κυβικών ποδιών, ο οποίος έχει το σχήμα μιας ανεστραμμένης πυραμίδας. Το σχήμα αυτό προκύπτει από το σημείο τομής των οπτικών πεδίων από τις διοπτρικές κάμερες. Προηγουμένως, το εύρος θέασης του Leap Motion ήταν περιορισμένο στα δύο πόδια (60 εκατοστά) πάνω από την επιφάνεια της συσκευής. Με την αναβάθμιση του λογισμικού Orion, το εύρος θέασης έχει επεκταθεί στα 2,6 πόδια (80 εκατοστά). Το εύρος αλληλεπίδρασης επηρεάζεται και περιορίζεται από τον τρόπο διάδοσης του LED φωτισμού στο χώρο. Η ένταση του LED φωτισμού εξαρτάται από το μέγιστο ρεύμα που μπορεί να διοχετευτεί μέσα από μια σύνδεση USB.



Εικόνα 13: Χώρος αλληλεπίδρασης Leap Motion

Οι λυχνίες LED παράγουν υπέρυθρο φως το οποίο, ανάλογα με τη θέση των χεριών του χρήστη πάνω από τον αισθητήρα, αντανακλάται και επιστρέφει στις κάμερες της συσκευής. Ανάλογα με τον τύπο της USB θύρας αναγνωρίζονται και επιστρέφονται 50 έως 200 εικόνες ανά δευτερόλεπτο. Στη συνέχεια αυτά τα καρέ της εικόνας

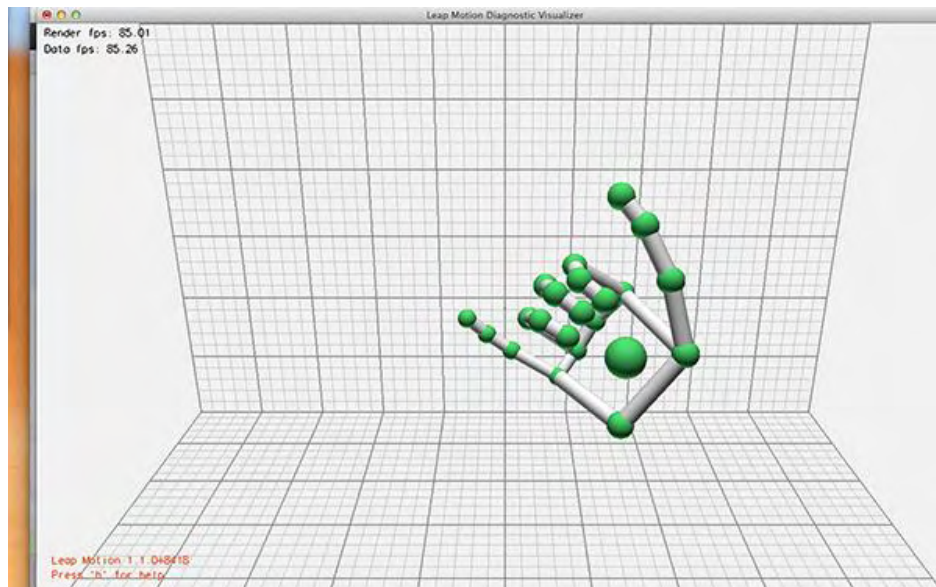


μεταδίδονται μέσω του USB καλωδίου στον υπολογιστή όπου γίνεται η ανάλυση τους με τη χρήση κατάλληλου λογισμικού και σύνθετων μαθηματικών μοντέλων. Μέσω του λογισμικού γίνεται αναγνώριση των σημείων αναφοράς και συντεταγμένων στα χέρια του χρήστη. [16] [17] [18]

### 3.2.2 Λογισμικό Leap Motion

Η τεχνολογία του αισθητήρα Leap Motion παρέχει ένα ολοκληρωμένο σύστημα διεπαφής προγραμματισμού (API) το οποίο είναι το Leap SDK API. Το σύστημα αυτό περιλαμβάνει για ένα μεγάλο σύνολο λειτουργιών βιβλιοθήκες προγραμματισμού, προσφέροντας επικοινωνία με το λειτουργικό σύστημα μέσω της συσκευής του αισθητήρα. Οι γλώσσες προγραμματισμού που υποστηρίζονται είναι η C++, C#, Java, Python, JavaScript και οι πλατφόρμες προγραμματισμού είναι κυρίως η Unity και η Unreal Engine. [19]

Το λογισμικό παρέχει ένα μοντέλο σκελετικού εντοπισμού και παρακολούθησης των χεριών και των δαχτύλων. Με τη μοντελοποίηση του ανθρώπινου χεριού μπορεί να γίνει καλύτερη και ορθότερη πρόβλεψη των θέσεων των δαχτύλων και των χεριών που η αναγνώριση τους από τη συσκευή δεν είναι πάντα ξεκάθαρη. Κατά τη διάρκεια ανίχνευσης της χειρονομίας του χεριού υπάρχουν πάντα πέντε δάχτυλα τα οποία μπορεί να είναι είτε εκτεταμένα είτε όχι. Ακόμα, τα δάχτυλα κατά την παρακολούθηση μπορεί να είναι διασταυρωμένα μεταξύ τους σχηματίζοντας διαφορετικούς συνδυασμούς.



**Εικόνα 14: Παράδειγμα χειρονομίας μοντέλου χεριού**

Το Leap SDK API περιλαμβάνει:

- Προσδιορισμό δεξιού ή αριστερού χεριού που αναγνωρίζεται.
- Αναγνώριση ψηφίων.
- Αναφορά της θέσης και προσανατολισμού κάθε δαχτύλου.
- Αναφορά παραγόντων πρόσφυσης που υποδεικνύουν αν ο χρήστης πιέζει ή αρπάζει κάτι.
- Αναφορά των πέντε δαχτύλων για κάθε χέρι.
- Αναφορά για το πότε ένα δάχτυλο επεκτείνεται ή όχι.

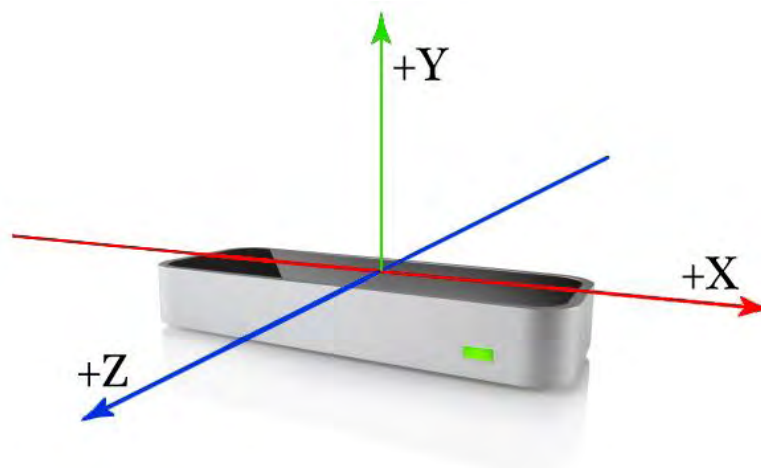
Επιπρόσθετα, χρησιμοποιεί κλάσεις, οι οποίες αναπαριστούν τα αντικείμενα που τις αποτελούν, όπως είναι οι χειρονομίες, τα δάχτυλα του χεριού, ολόκληρο το χέρι ή ακόμα κάποιο εργαλείο που μπορεί να χρησιμοποιηθεί.

Τέλος, η λειτουργία του αισθητήρα Leap Motion βασίζεται στο συνδυασμό ενός μοντέλου συντεταγμένων, ενός μοντέλου χεριών και δαχτύλων και σε ένα σκελετικό σύστημα οστών των δαχτύλων.

**Αναλυτικά:**

**1) Σύστημα συντεταγμένων:**

Η λειτουργία του αισθητήρα στηρίζεται σε ένα καρτεσιανό σύστημα συντεταγμένων. Οι άξονες X και Z βρίσκονται στο οριζόντιο επίπεδο, ενώ ο άξονας Y είναι παράλληλος προς το μεγάλο άκρο της συσκευής. Ο άξονας Y είναι κάθετος, με τις θετικές τιμές να αυξάνονται προς τα πάνω.



*Εικόνα 15: Καρτεσιανό σύστημα συντεταγμένων*

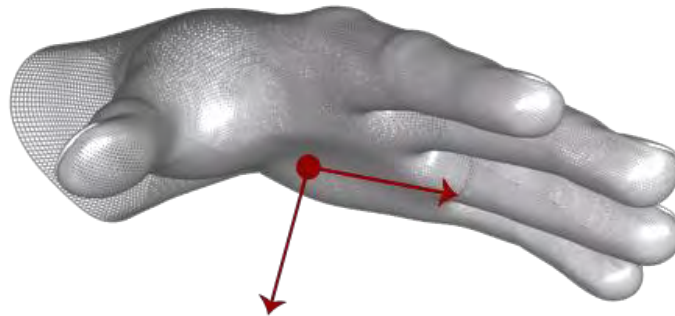
## 2) Μοντέλο χεριού:

Το μοντέλο του χεριού παρέχει πληροφορίες σχετικά με την θέση και τον προσανατολισμό του χεριού που έχει ανιχνευτεί από τον αισθητήρα καθώς και για τις κινήσεις των δαχτύλων του χεριού.

Γενικά, το λογισμικό του Leap Motion χρησιμοποιεί ένα εσωτερικό μοντέλο ανθρώπινου χεριού για την παρακολούθηση της κίνησης ακόμα και όταν δεν είναι ορατά όλα τα τμήματα ενός χεριού. Το μοντέλο παρέχει πάντα θέσεις και για τα πέντε δάχτυλα είτε είναι ορατά είτε όχι. Το λογισμικό χρησιμοποιεί τα ορατά μέρη του χεριού, ενώ το εσωτερικό μοντέλο που είναι ενσωματωμένο υπολογίζει τις πιθανές θέσεις των τμημάτων που δεν είναι ορατά από τον αισθητήρα.

Το μοντέλο χεριού αναπαρίσταται από την κλάση Hand με τις εξής παραμέτρους:

- Palm Normal: Διάνυσμα κάθετο στην παλάμη.
- Direction: Διάνυσμα παράλληλο στην παλάμη.



*Εικόνα 16: Μοντέλο χεριού*

## 3) Μοντέλο δαχτύλων:

Στο μοντέλο αυτό παρέχεται πληροφορία για κάθε δάχτυλο του χεριού. Αν όλα ή μερικά δάχτυλα δεν είναι ανιχνεύσιμα από τον αισθητήρα, τότε τα χαρακτηριστικά τους υπολογίζονται βάσει των πρόσφατων παρατηρήσεων που έχουν γίνει και βάσει του ανατομικού μοντέλου του χεριού. Τα δάχτυλα προσδιορίζονται σύμφωνα με το όνομα του τύπου τους όπως, αντίχειρα, δείκτη, μεσαίο, παράμεσο και μικρό.

Το μοντέλο των δαχτύλων αναπαρίσταται από την κλάση Finger με τις εξής παραμέτρους:

- TipPosition: Διάνυσμα της θέσης του δαχτύλου στο καρτεσιανό σύστημα.
- Direction: Διάνυσμα που δείχνει τη διεύθυνση του δαχτύλου.



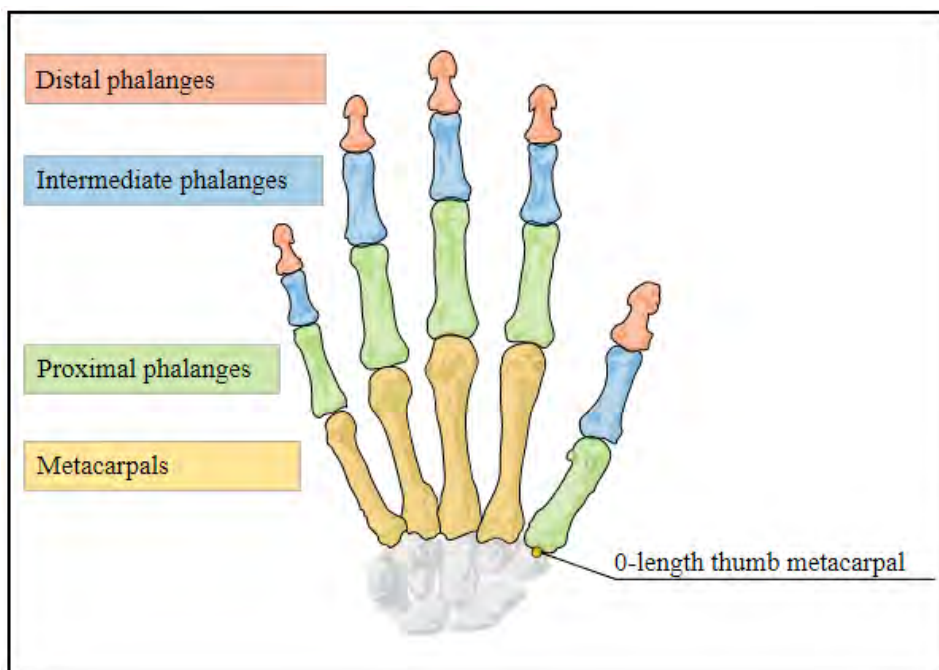


Εικόνα 17: Μοντέλο δαχτύλων

#### 4) Σύστημα οστών και δαχτύλων:

Το σκελετικό σύστημα έχει ως εξής:

- Μετακάρπιο (Metacarpal): είναι το οστό στο εσωτερικό του χεριού το οποίο συνδέει το δάχτυλο στον καρπό (εκτός του αντίχειρα).
- Εγγύς φάλαγγα (Proximal Phalanx): είναι το οστό στη βάση του δακτύλου, το οποίο συνδέεται με την παλάμη.
- Ενδιάμεση Φάλαγγα (Intermediate Phalanx): είναι το μεσαίο οστό του δακτύλου ανάμεσα στην κορυφή και τη βάση.
- Άπω Φάλαγγα (Distal Phalanx): είναι το τελικό οστό στην άκρη του δακτύλου



Εικόνα 18: Σκελετικό σύστημα

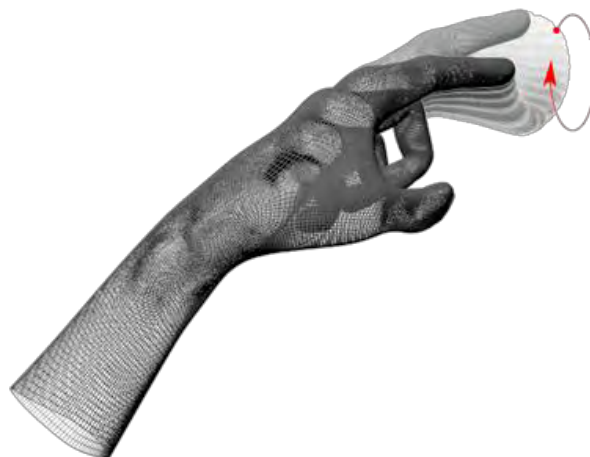
Ο αντίχειρας έχει ένα οστό λιγότερο οστό από τα υπόλοιπα δάχτυλα του χεριού, δηλαδή δεν έχει μετακάρπιο. Όμως για την ευκολία του προγραμματισμού με τη χρήση του αισθητήρα, το σκελετικό σύστημα προσδίδει ένα μετακάρπιο οστό μηδενικού μήκους στον αντίχειρα, ώστε όλα τα δάχτυλα να έχουν ίσο αριθμό οστών και ο προγραμματισμός και η αναγνώριση των χειρονομιών να είναι πιο εύκολη. [18] [20] [21]

### 3.2.3 Ανίχνευση χειρονομιών από τον αισθητήρα Leap Motion

Η αναγνώριση και ο προγραμματισμός των χειρονομιών είναι ένα από τα πιο πολύπλοκα αντικείμενα του Leap Motion SDK API. Το λογισμικό του Leap Motion αναγνωρίζει ορισμένα μοτίβα κίνησης τα οποία μπορεί να υποδηλώνουν μια πρόθεση ή μια εντολή από το χρήστη του συστήματος. Υπάρχουν κάποιοι συγκεκριμένοι τύποι χειρονομιών που αναγνωρίζονται από τον αισθητήρα ανάλογα με τη θέση, τη διεύθυνση και τη φορά του δαχτύλου.

Στο Leap Motion SDK API αναγνωρίζονται οι παρακάτω τύποι χειρονομιών:

**Circle gesture:** Το λογισμικό αναγνωρίζει την κυκλική κίνηση ενός δαχτύλου στο χώρο. Η κυκλική αυτή κίνηση πρέπει να είναι συνεχής (δεξιόστροφη ή αριστερόστροφη) και μπορεί να πραγματοποιηθεί από οποιοδήποτε δάχτυλο ή εργαλείο που μπορεί να τη σχηματίσει. Η κίνηση τερματίζεται όταν ο κύκλος που σχηματίζεται νοητά χαλάσει ή αν η κίνηση γίνεται πολύ αργά.



*Εικόνα 19: Circle gesture*

**Swipe gesture:** Το λογισμικό αναγνωρίζει μια απότομη γραμμική κίνηση πάνω από την επιφάνεια του αισθητήρα. Η κίνηση πρέπει να είναι συνεχής και μπορεί να υλοποιηθεί προς οποιαδήποτε κατεύθυνση. Η αναγνώριση της κίνησης θα τερματιστεί όταν τα δάχτυλα αλλάξουν κατευθύνσεις ή η κίνηση είναι πολύ αργή.



*Εικόνα 20: Swipe gesture*

**Key tap gesture:** Στη συγκεκριμένη περίπτωση το λογισμικό αναγνωρίζει μια γρήγορη και κατακόρυφη κίνηση που μοιάζει με το πάτημα ενός κουμπιού στο πληκτρολόγιο.



*Εικόνα 21: Key tap gesture*

**Screen tap gesture:** Το λογισμικό αναγνωρίζει μια γρήγορη και προς τα εμπρός κίνηση. Αυτή η κίνηση είναι παρόμοια με την κίνηση που κάνει κάποιος προς την οθόνη του υπολογιστή. [18] [22] [23]



*Εικόνα 22: Screen tap gesture*

### **3.2.4 Απαιτήσεις συστήματος για τη χρήση Leap Motion**

- Windows® 7+, Mac OS X 10.7 Lion λειτουργικό σύστημα
- AMD Phenom™ II ή Intel® Core™ i3/i5/i7 επεξεργαστής
- 2 GB RAM
- USB 2.0 θήρα
- Internet σύνδεση [24]

### **3.3 Σύγκριση αισθητήρων Microsoft Kinect – Leap Motion**

Έχοντας μελετήσει τις τεχνολογίες ανίχνευσης κίνησης τόσο του Microsoft Kinect αισθητήρα όσο και του αισθητήρα Leap Motion ως προς τα χαρακτηριστικά τους, τις δυνατότητες τους, το υλικό μέρος τους και το λογισμικό τους οδηγηθήκαμε σε κάποια συμπεράσματα.

Τα συμπεράσματα παρουσιάζονται στον παρακάτω πίνακα:

*Πίνακας 2: Σύγκριση των αισθητήρων Microsoft Kinect – Leap Motion*

	<b><u>Microsoft Kinect</u></b>	<b><u>Leap Motion</u></b>
<b>Αναγνώριση κίνησης</b>	Αναγνώριση σκελετού	Αναγνώριση δαχτύλων
<b>Προγραμματισμός</b>	Πολύπλοκος	Εύκολος
<b>Ακρίβεια ανίχνευσης</b>	Ανίχνευση κίνησης έως 3 μέτρα	Ανίχνευση κίνησης έως 80 εκατοστά
<b>Κόστος</b>	Υψηλό κόστος	Χαμηλό κόστος
<b>Συσκευή</b>	Μεγάλη συσκευή	Μικρή και εύκολη στη μεταφορά συσκευή

Το βασικό χαρακτηριστικό, το οποίο καθόρισε να χρησιμοποιηθεί η τεχνολογία του Leap Motion και όχι του Microsoft Kinect είναι το γεγονός ότι το πρώτο αναγνωρίζει και ξεχωρίζει τις κινήσεις και τους συνδυασμούς των δαχτύλων, ενώ το δεύτερο αναγνωρίζει ολόκληρο τον ανθρώπινο σκελετό από την κίνηση του μέχρι και τις εκφράσεις προσώπου. Για την υλοποίηση της εφαρμογής – διεπαφής της συγκεκριμένης πτυχιακής εργασίας το μόνο που κρίθηκε αναγκαίο ήταν να αναγνωριστούν οι κινήσεις των χεριών και ο συνδυασμός των δαχτύλων. Χάρη στις κατάλληλες βιβλιοθήκες που παρέχει το Leap Motion δίνει τη δυνατότητα να προγραμματιστούν έτσι ώστε να εκτελούν κάποια συγκεκριμένη λειτουργία. Σε αντίθεση το Microsoft Kinect έχει ένα μεγαλύτερο όγκο βιβλιοθηκών λόγω του μεγάλου όγκου δεδομένων που λαμβάνει ο αισθητήρας από την κίνηση του ατόμου στον χώρο γεγονός που καθιστά και τον προγραμματισμό οποιασδήποτε εφαρμογής πιο σύνθετο και πολύπλοκο.

Ένας ακόμη λόγος επιλογής του Leap Motion ήταν ότι είναι μια μικρή και οικονομική συσκευή, η οποία το μόνο που χρειάζεται είναι μια θήρα USB εν αντιθέσει με το Microsoft Kinect όπου το κόστος και το μέγεθος είναι εμφανώς μεγαλύτερα.



## 4. Ανάλυση απαιτήσεων

Η ανάλυση απαιτήσεων αποτελεί αναπόσπαστο κομμάτι για την ορθή ανάπτυξη ενός λογισμικού. Οι απαιτήσεις περιγράφουν τη συμπεριφορά ενός συστήματος. Πιο συγκεκριμένα, οι απαιτήσεις περιγράφουν τις δραστηριότητες του συστήματος και την κατάσταση της κάθε οντότητας του συστήματος πριν και μετά την εμφάνιση της δραστηριότητας. Γενικά, η ανάλυση των απαιτήσεων στοχεύει στην κατανόηση των αναγκών των χρηστών και της εργασίας που καλούνται να εκτελέσουν ώστε η διεπαφή που θα αναπτυχθεί να ανταποκρίνεται όσο το δυνατόν περισσότερο σε αυτές τις ανάγκες.

### 4.1 Απαιτήσεις συστήματος

Οι απαιτήσεις συστήματος αφορούν τις λειτουργίες, τις υπηρεσίες και τους λειτουργικούς περιορισμούς του συστήματος. Οι απαιτήσεις συστήματος χωρίζονται σε δύο κατηγορίες: τις λειτουργικές και τις μη λειτουργικές απαιτήσεις. Οι λειτουργικές απαιτήσεις περιγράφουν τις λειτουργίες του συστήματος και τον τρόπο συμπεριφοράς του συστήματος όταν αυτό δέχεται συγκεκριμένα ερεθίσματα. Οι μη λειτουργικές απαιτήσεις θέτουν περιορισμούς στο σύστημα όσο αφορά τα χαρακτηριστικά του και δεν σχετίζονται με την εκτέλεση κάποιας λειτουργίας. Αυτοί οι περιορισμοί μειώνουν συνήθως τις επιλογές μας όσο αφορά τη γλώσσα, την πλατφόρμα ή τις τεχνικές και τα εργαλεία υλοποίησης. [25] [26]

#### Λειτουργικές απαιτήσεις:

- Έλεγχος ορθής λειτουργίας του αισθητήρα Leap Motion.
- Μενού επιλογής για το χρήστη.
- Επιλογή ακαδημαϊκής λειτουργίας.
- Προβολή τρισδιάστατου μοντέλου του εγκεφάλου.
- Ταυτοποιημένες περιοχές του τρισδιάστατου μοντέλου του εγκεφάλου με δυνατότητα επιλογής τους.
- Πρόσβαση χρήστη σε κάθε περιοχή του μοντέλου του εγκεφάλου μεμονωμένα και στην πληροφορία που φέρει.
- Σύνδεση σε διαδικτυακή πηγή με το πάτημα ενός κουμπιού για την παροχή πληροφορίας.
- Επιστροφή στο αρχικό μενού επιλογής.
- Επιλογή λειτουργίας ιατρικού φακέλου.
- Εμφάνιση φόρμας για εισαγωγή και αποθήκευση προσωπικών δεδομένων του ασθενούς.
- Εισαγωγή νέου ασθενή στο σύστημα από το χρήστη.

- Διαγραφή ασθενή από το σύστημα.
- Επιλογή ασθενή από το σύστημα.
- Εμφάνιση φόρμας του συστήματος από το χρήστη για την ενημέρωση του ιατρικού φακέλου του ασθενή.
- Προβολή καταχωρήσεων στον ιατρικό φάκελο.
- Εισαγωγή καταχώρησης στον ιατρικό φάκελο.
- Διαγραφή καταχώρησης από τον ιατρικό φάκελο.
- Τροποποίηση καταχώρησης στον ιατρικό φάκελο.
- Μεμονωμένη εμφάνιση ιατρικού φακέλου.
- Αναίρεση μεμονωμένων αλλαγών από τον χρήστη.
- Δημιουργία προσωρινού φακέλου στον οποίο γίνεται η επεξεργασία ολόκληρου του ιστορικού.
- Ανανέωση των εγγραφών κατά την προβολή.
- Αποδοχή αποθήκευσης όλων των αλλαγών στο φάκελο του ασθενούς.
- Ακύρωση αλλαγών στον φάκελο του ασθενούς.
- Διαχείριση προσωρινού φακέλου για τη λειτουργία αναίρεσης.

#### **Μη λειτουργικές απαιτήσεις:**

- Επιλογή του κατάλληλου τρισδιάστατου μοντέλου του εγκεφάλου.
- Συμβατότητα με πλατφόρμα Unity.
- Ενσωμάτωση του τρισδιάστατου μοντέλου στην πλατφόρμα Unity.
- Συμβατότητα του Leap Motion με την έκδοση του Unity.
- Επικοινωνία της πλατφόρμας με τον αισθητήρα.
- Χρόνος απόκρισης του συστήματος σε κάθε περίπτωση αλληλεπίδρασης με το χρήστη λιγότερο από 3 δευτερόλεπτα.
- Δημιουργία πολλών σκηνών στην πλατφόρμα και ομαλή σύνδεση μεταξύ τους προκειμένου να υπάρχει ομαλή ροή και δυνατότητα πολλών επιλογών.
- Οι διεπαφές να είναι λιτές και με μονοσήμαντο αποτέλεσμα.
- Επιλογή του λευκού χρώματος στο φόντο ως ουδέτερο.
- Να μην υπάρχει χρωματική διακύμανση μεταξύ των σκηνών.
- Το περιβάλλον της εφαρμογής να είναι οικείο και να μη χρειάζεται κάποια εξειδικευμένη εκπαίδευση ο χρήστης.
- Σε κάθε διαφορετική σκηνή η σχεδίαση να παραμένει σταθερή ώστε να μην υπάρχει σύγχυση από την πλευρά του χρήστη.
- Εύκολη πλοήγηση μεταξύ των σκηνών.
- Επιλογή επιστροφής στην προηγούμενη σκηνή.
- Επιλογή επιστροφής στην αρχική σελίδα.



## 4.2 Απαιτήσεις οργάνωσης συστήματος

Αυτό το κομμάτι αφορά την κατανόηση των δυνατοτήτων οργάνωσης που προσφέρει το λειτουργικό σύστημα προκειμένου να μπορεί να γίνει η διαχείριση των καταχωρημένων ασθενών και των ιατρικών φακέλων τους.

- Δημιουργία φακέλου.
- Διαγραφή φακέλου.
- Μετονομασία φακέλου.
- Δημιουργία αρχείου.
- Διαγραφή αρχείου.
- Μετονομασία αρχείου.
- Προσάρτηση (append) σε αρχείο.
- Οργάνωση αρχείων σε φακέλους.
- Μεταφορά αρχείων.
- Φάκελος εφαρμογής.
- Φάκελος ασθενών.
- Πρότυπα αρχεία.

## 4.3 Διάγραμμα περιπτώσεων χρήσης

Το διάγραμμα περιπτώσεων χρήσης (Use Case Diagram) αποτυπώνει τις προδιαγραφές του υπό κατασκευήν συστήματος. Δίνει έμφαση στη λειτουργικότητα ενός συστήματος, όπως αυτή είναι ορατή στους εξωτερικούς χρήστες του.

Τα βασικά διαγραμματικά στοιχεία των διαγραμμάτων περιπτώσεων χρήσης είναι τα εξής:

### Περίπτωση χρήσης

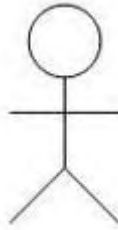
Αποτελεί ένα στόχο για έναν εξωτερικό χειριστή του συστήματος. Το σύμβολο το οποίο χρησιμοποιείται για μια περίπτωση χρήσης είναι η έλλειψη, μέσα στην οποία αναγράφεται το όνομα της περίπτωσης χρήσης.



*Εικόνα 23: Σύμβολο περίπτωσης χρήσης*

### Χειριστής

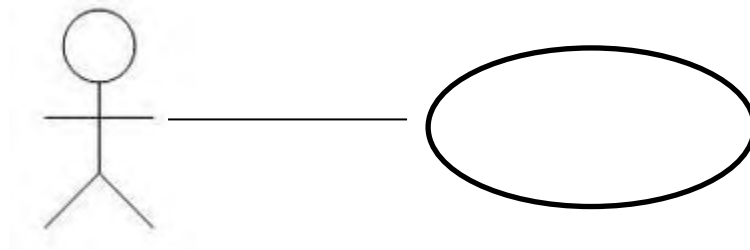
Ο χειριστής ενός συστήματος μπορεί να είναι άνθρωπος ή υποσύστημα. Το σύμβολο το οποίο χρησιμοποιείται για την αναπαράσταση του χειριστή είναι ένα ανθρώπακι. Οι χειριστές αποτελούν την πηγή από την οποία προκύπτουν οι προδιαγραφές του συστήματος.



*Εικόνα 24: Σύμβολο χειριστή*

### Σχέση

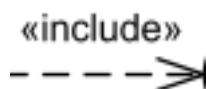
Δηλώνει τη σύνδεση ενός χειριστή με μια περίπτωση χρήσης. Ένας ή περισσότεροι χειριστές μπορεί να σχετίζονται με μια περίπτωση χρήσης. Η σχέση συμβολίζεται με μια ευθεία γραμμή.



*Εικόνα 25: Σχέση χειριστή με μια περίπτωση χρήσης*

### Συμπερίληψη (include)

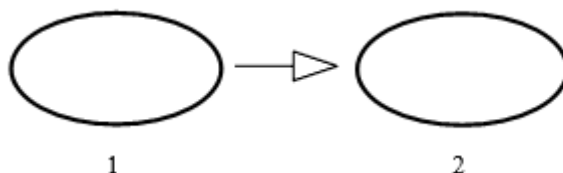
Είναι μια ειδική περίπτωση σχέσης με την οποία συσχετίζονται μεταξύ τους δύο περιπτώσεις χρήσης. Η μία περίπτωση χρήσης συμπεριλαμβάνει την άλλη. Η συμπερίληψη είναι υποχρεωτική και χρησιμοποιείται όταν η ίδια λειτουργικότητα περιλαμβάνεται σε περισσότερες από μια περιπτώσεις χρήσης. Το σύμβολο που χρησιμοποιείται είναι ένα κατευθυνόμενο διακεκομμένο βέλος, όπου η φορά του βέλους έχει κατεύθυνση από την περίπτωση χρήσης που συμπεριλαμβάνει προς αυτή που συμπεριλαμβάνεται.



*Εικόνα 26: Σύμβολο συμπερίληψης*

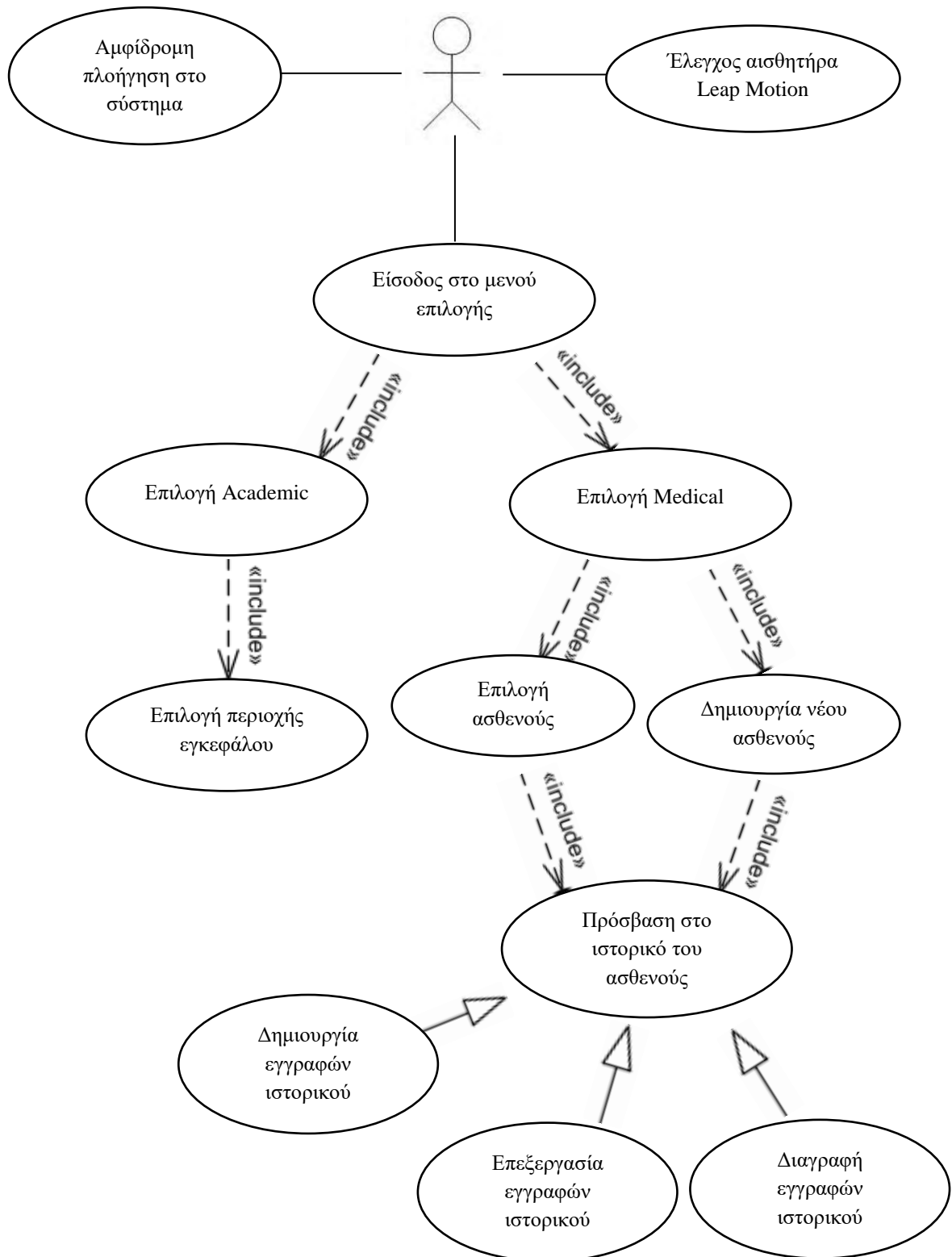
### **Επέκταση (extend)**

Κατά την επέκταση μια περίπτωση χρήσης επεκτείνεται προαιρετικά από μια άλλη. Η σχέση αυτή μπορεί να προκύψει σε συγκεκριμένα σημεία της λειτουργικότητας της βασικής ροής της περίπτωσης χρήσης. [27]



*Εικόνα 27: : Η περίπτωση χρήσης 2 επεκτείνεται από την περίπτωση χρήσης*

Στη συνέχεια ακολουθεί το διάγραμμα περίπτωσης χρήσης για το χρήστη του συστήματος.



Εικόνα 28: Περίπτωση χρήσης του χρήστη

## 5. Υλοποίηση της διεπαφής

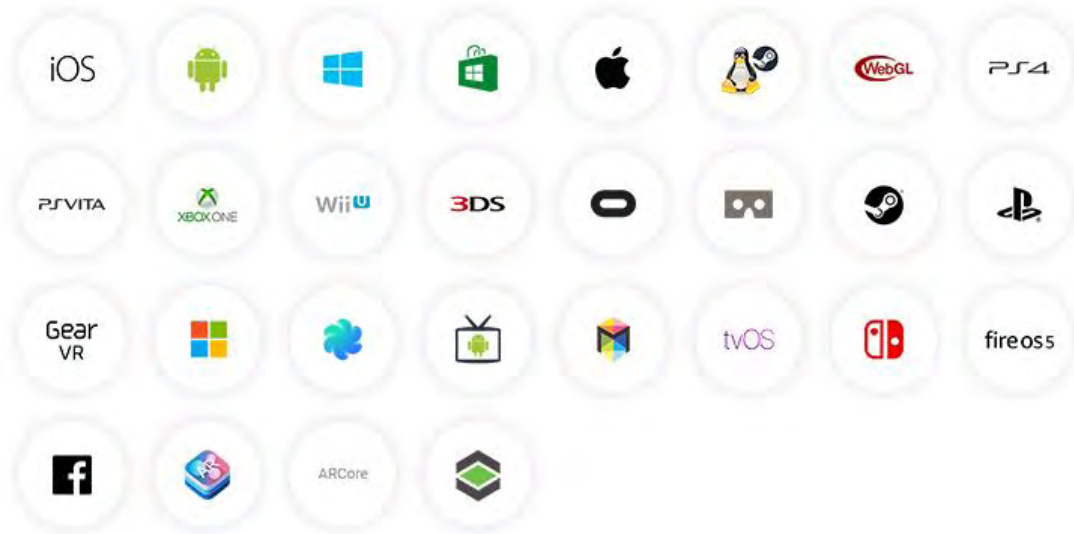
Η εφαρμογή που παρουσιάζεται στην πτυχιακή εργασία αναλύθηκε και σχεδιάστηκε στην πλατφόρμα του Unity. Παρακάτω θα δοθούν πληροφορίες σχετικά με το Unity και όλων των εργαλείων που χρησιμοποιήθηκαν ώστε να οδηγηθούμε στο τελικό αποτέλεσμα.

### 5.1 Πλατφόρμα του Unity

Το Unity είναι μια μηχανή κατασκευής δισδιάστατων και τρισδιάστατων βιντεοπαιχνιδιών και προσομοιώσεων για υπολογιστές, κονσόλες και κινητές συσκευές. Η εξέλιξη του είναι τόσο μεγάλη που μέχρι σήμερα υποστηρίζεται σε 28 διαφορετικά λογισμικά και συσκευές. Η λειτουργικότητα του Unity στηρίζεται κυρίως στη μέθοδο drag-and-drop συνοδευόμενη από τον προγραμματισμό των ενεργειών που επιθυμεί ο χρήστης να πραγματοποιούνται μέσω της γλώσσας προγραμματισμού C#. Ακόμη μια γλώσσα προγραμματισμού που υποστηρίζονταν από την πλατφόρμα μέχρι πρόσφατα ήταν αυτή της JavaScript, αλλά από τον Αύγουστο του 2017 έχει αρχίσει σταδιακά η διαδικασία της απομάκρυνσης της. Το Unity οργανώνεται σε σκηνές (scenes). Η κάθε σκηνή έχει απαραίτητα μια κάμερα προκειμένου να περιεχόμενα της να εμφανιστούν στη λειτουργία «play». Η κάθε σκηνή μπορεί να επικοινωνήσει με όποια σκηνή επιθυμεί ο χρήστης αρκεί να κάνει σωστά τη συσχέτιση μεταξύ του μέσω του κώδικα (script). Επιπλέον, ο χρήστης κατά τη διάρκεια της δημιουργίας ενός νέου πρότζεκτ μπορεί να ενσωματώσει εξωτερικά πακέτα με στοιχεία και εργαλεία διάδρασης (assets) στα ήδη υπάρχοντα του έργου.

Το Unity παρέχει τρεις διαφορετικές εκδόσεις τις οποίες ο χρήστης/προγραμματιστής μπορεί να κατεβάσει στον υπολογιστή του και να αρχίσει αμέσως να το χρησιμοποιεί. Η μία είναι η Personal έκδοση, η οποία απευθύνεται κυρίως σε φοιτητές και αρχάριους προγραμματιστές. Είναι δωρεάν και παρέχει πρόσβαση σε όλα τα εργαλεία της πλατφόρμας με συνεχή αναβάθμιση σε αυτά. Οι υπόλοιπες δύο είναι η Unity Plus έκδοση και η Unity Pro, οι οποίες είναι επι πληρωμή και απευθύνονται σε πιο εξειδικευμένο κοινό προγραμματιστών, καθώς και οι λειτουργίες και τα εργαλεία που προσφέρουν είναι επαγγελματικά και πιο εξειδικευμένα.

Οι πλατφόρμες που υποστηρίζουν το Unity φαίνονται στην παρακάτω εικόνα:



*Εικόνα 29: Σύνολο συσκευών και λογισμικών*

Ένα ακόμα χαρακτηριστικό που συνοδεύει τη μηχανή ανάπτυξης του Unity είναι ότι παρέχει ένα ολοκληρωμένο σύστημα υποστήριξης για το χρήστη με έτοιμα πακέτα assets, τα οποία θα τον βοηθήσουν να υλοποιήσει και να ολοκληρώσει το προσωπικό του δημιούργημα. Το σύστημα αυτό ονομάζεται Unity Asset Store και παρέχει έναν ολοκληρωμένο κατάλογο δωρεάν ή επι πληρωμής περιεχόμενου. Περιλαμβάνει τρισδιάστατα μοντέλα, ολοκληρωμένα πακέτα με UI στοιχεία, έτοιμους κώδικες, πολυμεσικά στοιχεία όπως ήχοι και εφέ σωματιδίων (particles). Το Unity είναι μια εύκολη και έξυπνη μηχανή παιχνιδιού. Μέχρι και σήμερα βρίσκεται στην κορυφή όσο αφορά το πλήθος των επιλογών. Προσφέρει άριστα γραφικά, εργαλεία User Interface για αλληλεπίδραση και υποστήριξη για λογισμικό καταγραφής κίνησης. [28] [29]

## **5.2 Βήματα υλοποίησης**

Στη συγκεκριμένη ενότητα θα παρουσιαστούν αναλυτικά τα βήματα που πραγματοποιήθηκαν για την υλοποίηση της διεπαφής. Θα γίνει αναλυτική αναφορά σε όλα τα στάδια της υλοποίησης, στα προβλήματα που παρουσιάστηκαν και στον τρόπο που αντιμετωπίστηκαν.

### **5.2.1 Επιλογή τρισδιάστατου μοντέλου**

Αρχικά, το πρώτο βήμα ήταν η εύρεση του κατάλληλου τρισδιάστατου ανατομικού μοντέλου του εγκεφάλου. Η σωστή επιλογή του ήταν πολύ σημαντική, καθώς το σύστημα αλληλεπίδρασης βασίζεται στον χρήστη και στο τρισδιάστατο μοντέλο. Το μοντέλο έπρεπε να πληρεί τις κατάλληλες προϋποθέσεις και να διαθέτει τα

κατάλληλα χαρακτηριστικά ώστε να ικανοποιεί το τελικό επιθυμητό αποτέλεσμα της διεπαφής.

Κάποια από τα κύρια χαρακτηριστικά που έπρεπε να συνοδεύουν το τρισδιάστατο μοντέλου του εγκεφάλου και τα οποία αποτελούσαν αναπόσπαστο κομμάτι για την υλοποίηση της εφαρμογής είναι τα εξής:

- Ρεαλιστική και πλήρης απεικόνιση του εγκεφάλου.
- Η δημιουργία του τρισδιάστατου μοντέλου να μην είναι τυχαία αλλά να είναι από κάποιον γνώστη του αντικειμένου, για παράδειγμα κάποιον ιατρό ή βιολόγο.
- Υψηλή ανάλυση στις δομές του εγκεφάλου.
- Δυνατότητα διαχωρισμού των περιοχών του εγκεφάλου σε μεμονωμένες οντότητες.
- Χαμηλό κόστος.

Έπειτα από εκτενή έρευνα που πραγματοποιήθηκε στο διαδίκτυο και σε ιστοσελίδες που παρείχαν έτοιμα ανατομικά μοντέλα του εγκεφάλου, οδηγηθήκαμε στην επιλογή του τρισδιάστατου μοντέλου που παρουσιάζεται στην εφαρμογή. Το μοντέλο αυτό ικανοποιεί όλα τα κριτήρια που τέθηκαν παραπάνω. Η απεικόνιση του είναι πλήρης και ρεαλιστική. Η μία πλευρά του εγκεφάλου φαίνεται συμπαγής αλλά η άλλη παρουσιάζεται διχοτομημένη ώστε να φαίνονται αναλυτικά οι δομές του. Το μοντέλο έχει δημιουργηθεί υπό την επίβλεψη και καθοδήγηση ενός καθηγητή της νευρολογίας και η κλίμακα των διαστάσεων είναι προσεκτικά ορισμένη ώστε να αντιστοιχεί στην πραγματικότητα. Επίσης, οι δομές του είναι υψηλής ανάλυσης και συνοδεύεται από χρωματικούς χάρτες. Το μοντέλο μπορεί να διαχωριστεί σε οκτώ διαφορετικές οντότητες και αποτελείται από τα αντίστοιχα νεύρα του εγκεφάλου. [30]

Η αγορά του πραγματοποιήθηκε από την ιστοσελίδα TURBOSQUID (<https://www.turbosquid.com/>), όπου το μοντέλο είναι ακόμα διαθέσιμο προς αγορά. Πιο αναλυτικά τα χαρακτηριστικά του μοντέλου παρουσιάζονται στην παρακάτω εικόνα:

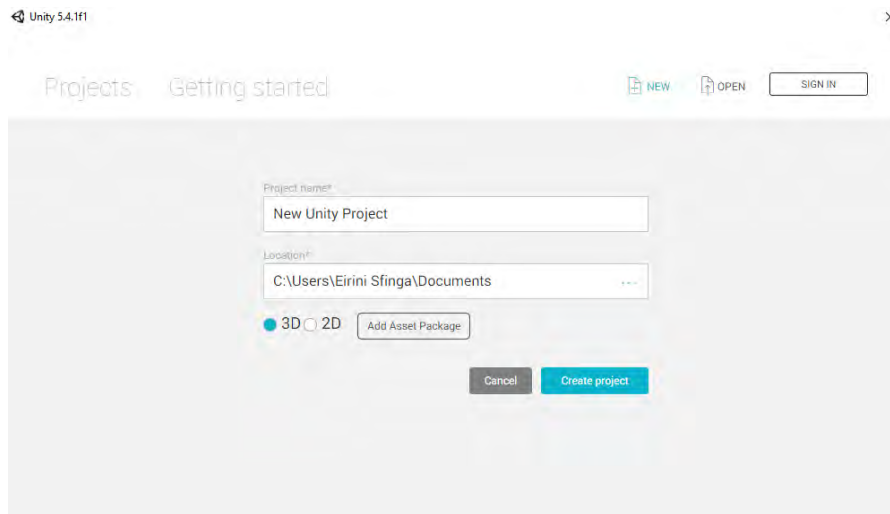
3D Model Specifications	
Product ID:	815426
Published:	Apr 18, 2014
Geometry:	Polygonal Quads/Tris
Polygons:	49,795
Vertices:	99,746
Textures:	Yes
Materials:	Yes
Rigged:	No
Animated:	No
UV Mapped:	Yes
Unwrapped UVs:	Yes, overlapping

*Εικόνα 30: Πίνακας χαρακτηριστικών*

### **5.2.2 Εισαγωγή μοντέλου στο Unity**

Το επόμενο βήμα μετά την απόκτηση του τρισδιάστατου μοντέλου ήταν να γίνει η ενσωμάτωση του στο Unity. Εφόσον, το μοντέλο είναι τρισδιάστατο και όλη η εφαρμογή θα περιλάμβανε στοιχεία του σχεδόν σε κάθε σκηνή, από το μενού του Unity για δημιουργία νέου έργου επιλέχθηκε η σχεδίαση να γίνει σε 3D περιβάλλον. Όπως φαίνεται και στην εικόνα που ακολουθεί κατά τη δημιουργία ενός νέου έργου (project) ο χρήστης μπορεί να επιλέξει αν θα χρησιμοποιήσει ένα 2D ή ένα 3D περιβάλλον ανάπτυξης. Στο περιβάλλον αυτό μπορούν να τοποθετηθούν όλα τα πακέτα των στοιχείων που επιθυμεί (assets package). Ως assets μπορούν να θεωρηθούν όλα τα γραφικά στοιχεία, βιβλιοθήκες προγραμματισμού, τρισδιάστατα μοντέλα, κώδικες (scripts), εικόνες και ούτω καθεξής.

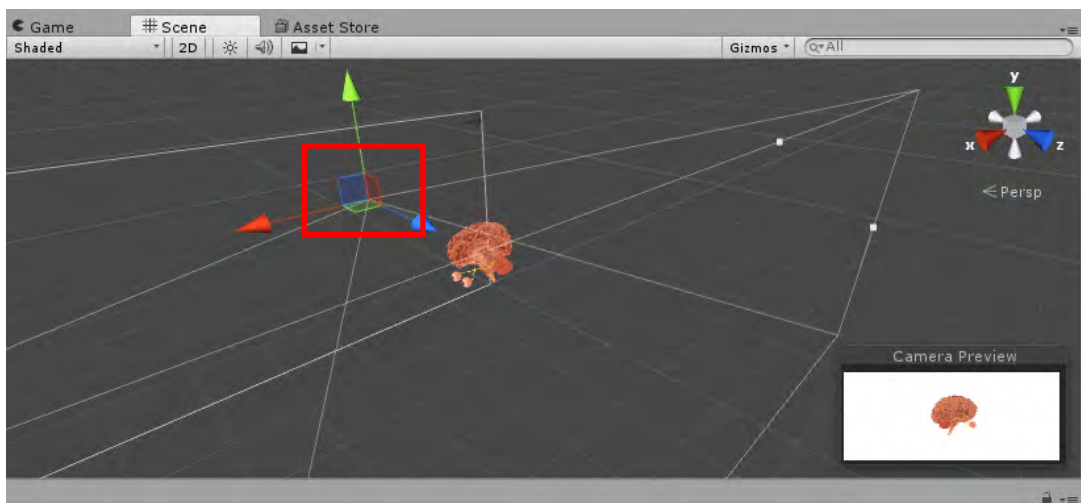




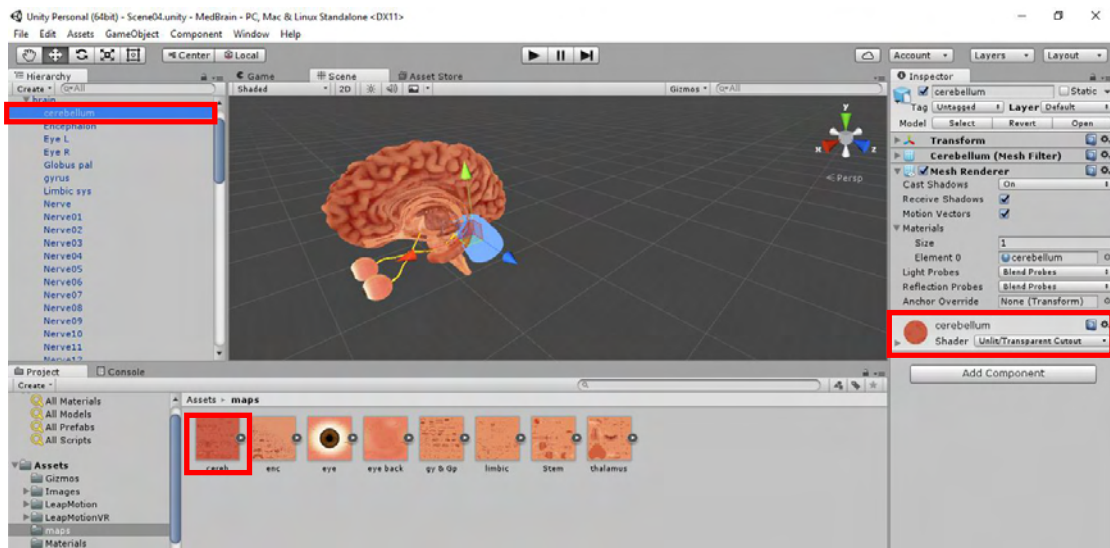
**Εικόνα 31: Δημιουργία πρότζεκτ στο Unity**

Αφού πραγματοποιήθηκε η διαδικασία της δημιουργίας του πρότζεκτ έχει δημιουργηθεί σε αυτό ο φάκελος με το όνομα assets. Για να ενσωματωθεί το τρισδιάστατο μοντέλο στο Unity το μόνο που χρειάζεται να γίνει είναι drag-and-drop διαδικασία των περιεχομένων του φακέλου με το όνομα Brain 3D στο φάκελο assets του έργου.

Έπειτα πάλι με τη διαδικασία drag-and-drop έγινε ενσωμάτωση του μοντέλου στην σκηνή του Unity. Επιλέγοντας κάθε οντότητα του μοντέλου ξεχωριστά εισήγαγα τους χρωματικούς χάρτες που συνόδευαν το μοντέλο. Το πρώτο πρόβλημα που παρουσιάστηκε ήταν ότι το μοντέλο δεν εμφανιζόταν στην λειτουργία «play scene». Αυτό διορθώθηκε τοποθετώντας το μοντέλο σε θέση τέτοια ώστε να ανιχνεύεται από την κάμερα της σκηνής και ορίζοντας το ως αντικείμενο game object.



**Εικόνα 32: Σωστή τοποθέτηση μοντέλου απέναντι από την κάμερα(κόκκινο πλαίσιο)**



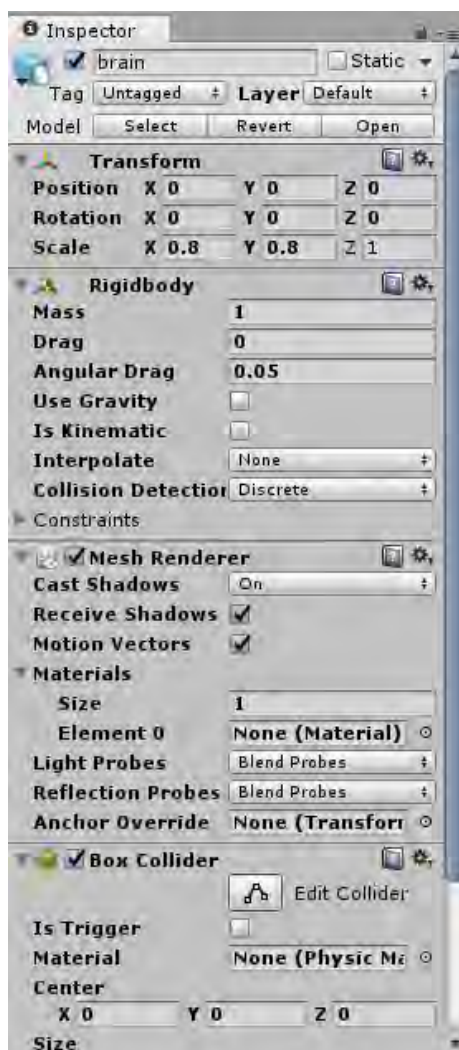
Εικόνα 33: Εισαγωγή χρωματικού χάρτη σε κάθε δομή του μοντέλου ξεχωριστά

### 5.2.3 Έννοια Game Object

Η έννοια game object όπου προ λίγου αναφέρθηκε, έλυσε πολλαπλά προβλήματα που παρουσιάστηκαν στη συνέχεια του σχεδιασμού της διεπαφής. Τα προβλήματα αυτά δεν αφορούσαν μόνο το τρισδιάστατο μοντέλο αλλά και προβλήματα που παρουσιάστηκαν στη δημιουργία User Interface και των λειτουργιών των στοιχείων UI, στην επικοινωνία των σκηνών μεταξύ τους καθώς και στην επικοινωνία του αισθητήρα με τις αντίστοιχες σκηνές.

Τα game objects είναι τα θεμελιώδη αντικείμενα του Unity. Αντιπροσωπεύουν χαρακτήρες, τρισδιάστατα μοντέλα, σκηνικά και λειτουργίες. Από μόνα τους δεν επιτυγχάνουν πολλά αλλά λειτουργούν ως φορείς για «εξαρτήματα» (components), τα οποία υλοποιούν την πραγματική λειτουργικότητα. Με λίγα λόγια λειτουργούν ως χειριστήριο.

Στην ακόλουθη εικόνα παρουσιάζεται το τρισδιάστατο μοντέλο ορισμένο ως game object για να γίνει κατανοητή η λειτουργία του.



Εικόνα 34: Components του εγκεφάλου ως game object

Στα components που το ορίζουν περιλαμβάνονται:

- Mesh Renderer: αφορά την εμφάνιση του μοντέλου
- Box Collider: αναπαριστά το μοντέλο ως σύνολο και επιτρέπει την αλληλεπίδραση σύμφωνα με τους νόμους της φυσικής.
- Transform: αφορά τη θέση και τον προσανατολισμό του μοντέλου.
- Rigidbody: δέχεται ορίσματα για την κίνηση του μοντέλου σε ρεαλιστικές συνθήκες, για παράδειγμα, επηρεασμός από βαρύτητα.

Επίσης, στα components περνούν και τα scripts που έχει δημιουργήσει ο προγραμματιστής και είναι εκείνα τα οποία περιέχουν τις εντολές για την επιθυμητή λειτουργία.

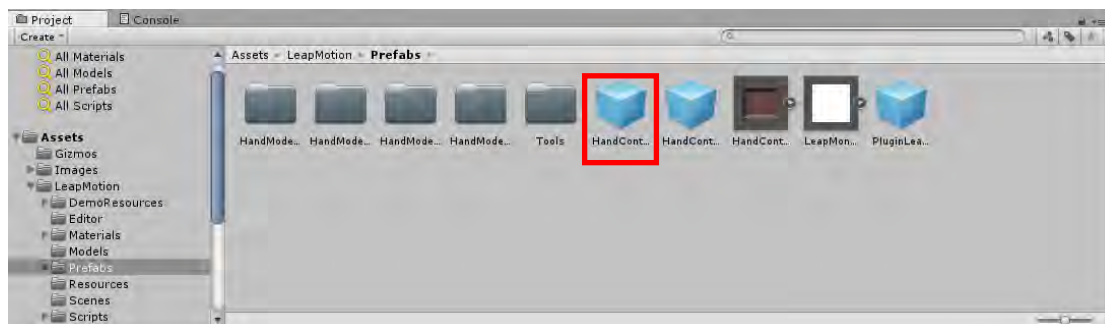
Τέλος, η χρήση του game object επεκτείνεται και όσο αφορά το περιεχόμενο των scripts. Το game object μπορεί να οριστεί ως μια κλάση μέσα στον κώδικα. Μπορεί να λάβει παραμέτρους όπως για παράδειγμα να κάνει κάποια σύγκριση, να ενεργοποιήσει κάποια λειτουργία και άλλα πολλά. [31] [32]

#### 5.2.4 Επικοινωνία Leap Motion – Unity

Έχοντας επιλέξει ότι ο σχεδιασμός και η υλοποίηση της διεπαφής θα πραγματοποιηθεί στην πλατφόρμα του Unity, το επόμενο βήμα ήταν να βρεθεί ένας τρόπος ώστε ο αισθητήρας του Leap Motion να επικοινωνήσει με την πλατφόρμα. Το θέμα αυτό μας απασχόλησε αρκετά, καθώς έπρεπε να βρεθεί τόσο η κατάλληλη έκδοση λογισμικού του Unity όσο και η κατάλληλη έκδοση του πακέτου SDK που συνοδεύει τον αισθητήρα του Leap Motion και τον κάνει λειτουργικό.

Έπειτα από πολλές δοκιμές και προσπάθειες καταλήξαμε στην έκδοση του Unity 5.4.1f1 (64-bit) και όσο αφορά το SDK του Leap Motion στην έκδοση Leap Motion Orion 3.2.1 που είναι διαθέσιμο προς τους προγραμματιστές από τις 13 Ιουνίου 2016. Το συγκεκριμένο πακέτο περιέχει βελτιώσεις στην αναγνώριση των χεριών, σε θέματα αναγνώρισης της συσκευής από το σύστημα καθώς και στην ποιότητα της αναγνώρισης των χειρονομιών. [33]

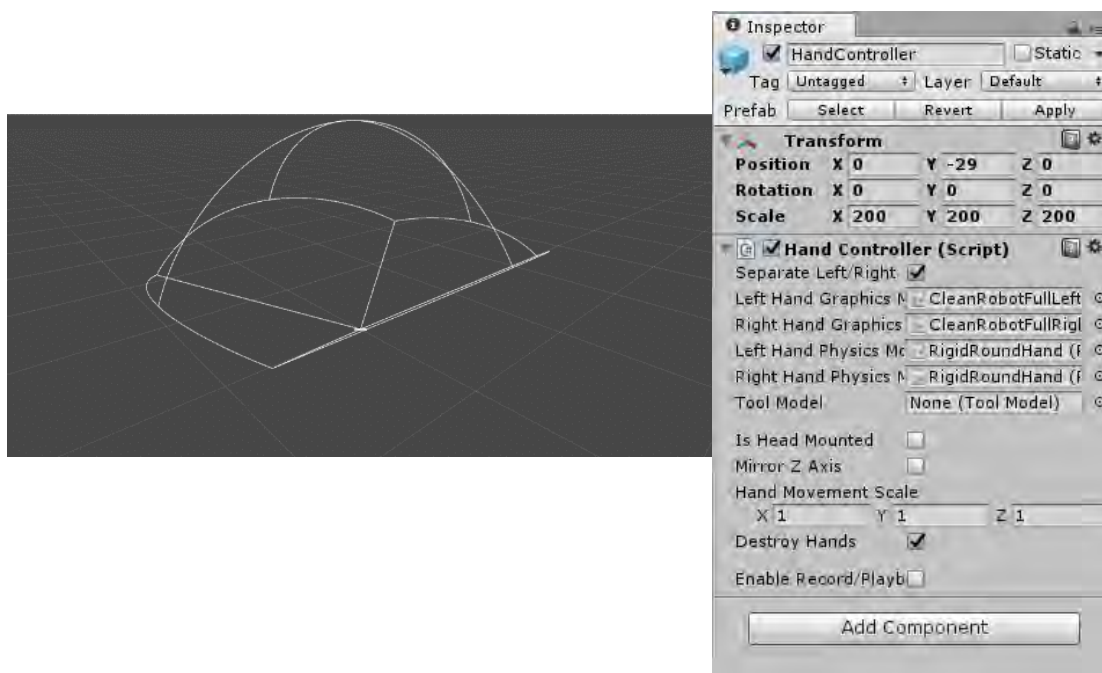
Μέσω της διαδικασίας ενσωμάτωσης πακέτων στο Unity, ενσωματώθηκαν στο φάκελο assets οι φάκελοι του Leap Motion SDK. Για να μπορέσει ο αισθητήρας να ενσωματωθεί στην σκηνή του Unity και να είναι λειτουργικός έγινε η εξής διαδικασία: Assets -> LeapMotion -> Prefabs -> HandController



Εικόνα 35: Επιλογή HandController

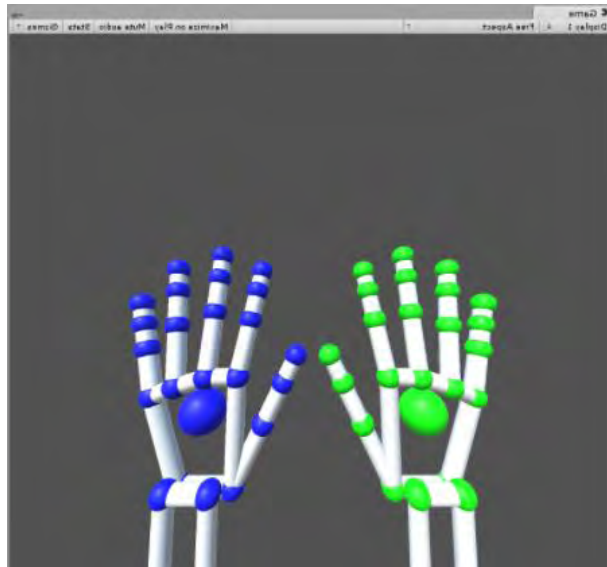
Ενσωματώνοντας στη σκηνή το HandController και εφόσον έχει τοποθετηθεί σε θέση τέτοια ώστε να αναγνωρίζεται από την κάμερα της σκηνής παρουσιάζεται στην μορφή της εικόνας που ακολουθεί. Υπάρχει στο χώρο της σκηνής μια μικρογραφία της συσκευής του Leap Motion και συνοδεύεται από το σχήμα μια ανεστραμμένης

πυραμίδας, όπου υποδηλώνεται ο χώρος στον οποίο μπορεί να αλληλοεπιδράσει ο χρήστης. Στα δεξιά υπάρχει στον Inspector η επιλογή Scale στην οποία ο χρήστης ρυθμίζει την κλίμακα του χώρου που επιθυμεί να είναι αναγνωρίσιμος από τον αισθητήρα.



**Εικόνα 36:** Ενσωμάτωση και ρύθμιση του *HandController* στη σκηνή

Έπειτα, στην λειτουργία play mode τα χέρια αναγνωρίζονται και παρουσιάζονται με την εξής μορφή.



*Εικόνα 37: Τα χέρια αναγνωρίζονται επιτυχώς από τον αισθητήρα*

Έπειτα από τα βήματα αυτά και το παραπάνω αποτέλεσμα ο αισθητήρας έχει επικοινωνήσει επιτυχώς με την πλατφόρμα του Unity και είναι έτοιμος προς οποιαδήποτε χρήση και προγραμματισμένη λειτουργία.

### **5.2.5 Βιβλιοθήκες κώδικα**

Στο επόμενο στάδιο, έπρεπε να γραφτούν οι ανάλογοι προγραμματιστικοί κώδικες προκειμένου να υλοποιούνται οι επιθυμητές ενέργειες από το χρήστη στην εφαρμογή της διεπαφής αλλά και οι ανάλογες ενέργειες από την πλευρά διαχείρισης του συστήματος.

Η πλατφόρμα του Unity χρησιμοποιεί το Microsoft Visual Studio 2015 για τη συγγραφή κώδικα σε C#. Το μυστικό σε αυτό το σημείο ήταν να οριστούν οι κατάλληλες βιβλιοθήκες στην αρχή του κώδικα ώστε να ενεργοποιηθούν στο περιεχόμενο του οι επιπλέον επιλογές – εντολές για τα αντικείμενα που υπάρχουν στο Unity και τις λειτουργίες που θέλουμε να επιτελούν. Επίσης, στον ορισμό των βιβλιοθηκών έπρεπε να ενσωματωθούν και οι κατάλληλες βιβλιοθήκες για τον αισθητήρα του Leap Motion ώστε γίνει πρόσβαση στις ενσωματωμένες συναρτήσεις του και στο API που το συνοδεύει ώστε η κάθε χειρονομία να μεταφραστεί σε κάποια συγκεκριμένη εντολή.

Ακολουθούν μερικά παραδείγματα ορισμένων βιβλιοθηκών πριν το κύριο σώμα του κώδικα.

```
using UnityEngine;
using System.Collections;
using UnityEngine.SceneManagement;
```

*Εικόνα 38: : Για διαχείριση εναλλαγής σκηνών*

```
using UnityEngine;
using UnityEditor;
using System.Collections;
using UnityEngine.UI;
using System.IO;
```

*Εικόνα 39: Για πρόσβαση στο User Interface και τα στοιχεία του, διαχείριση εντολών συστήματος.*

```
using UnityEngine;
using System.Collections;
using Leap; //για prosvasi sto API tou leap
motion
```

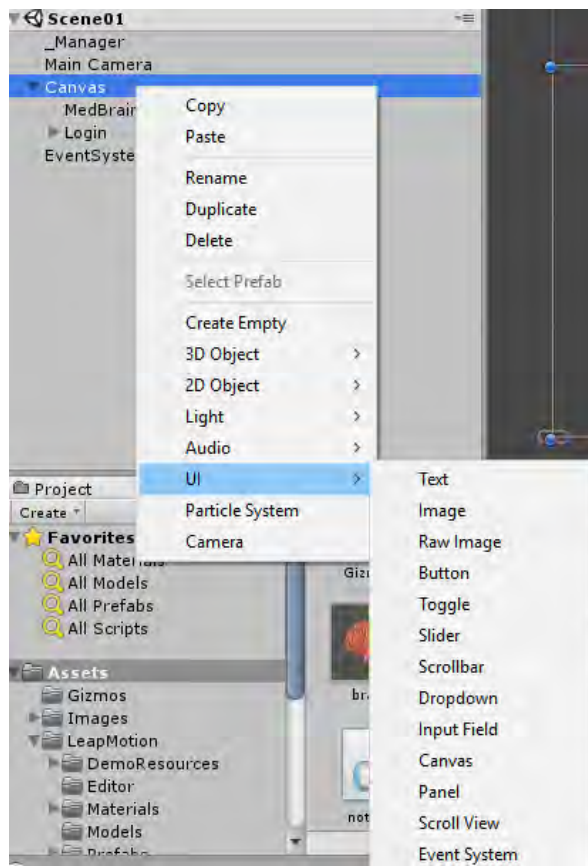
*Εικόνα 40: Για πρόσβαση στις λειτουργίες του αισθητήρα*

### **5.2.6 Δημιουργία User Interface (UI)**

Το User Interface που μπορεί να σχεδιάσει ένας χρήστης για την εφαρμογή του μπορεί να κυμανθεί από απλό έως σύνθετο και να διαμορφωθεί με βάση το προσωπικό του γούστο και τα κριτήρια που έχει θέσει για την εφαρμογή που επιθυμεί να υλοποιήσει.

Το πρώτο βήμα για το σχεδιασμό του UI της διεπαφής μας ήταν να δημιουργηθεί ένας καμβάς (canvas) και να υπάρχει και μια κάμερα (Main Camera) στη σκηνή. Ο καμβάς οριοθετεί έναν κενό χώρο στον οποίο μπορούν να τοποθετηθούν τα περαιτέρω στοιχεία του UI. Στον καμβά περιλαμβάνονται μια σειρά από στοιχεία UI που μπορούν να ενσωματωθούν στην σκηνή (scene). Τα στοιχεία αυτά μπορεί να είναι κάποιο κείμενο (text), μια εικόνα (image), ένα κουμπί (button), μια ροδέλα κύλισης (slider), ένα πάνελ (panel), ένα πλαίσιο κειμένου (Input Field) και άλλα πολλά.





*Εικόνα 41: : Εμφάνιση των UI στοιχείων*

Η εφαρμογή μας περιλαμβάνει σχεδόν όλα τα διαθέσιμα UI στοιχεία. Αν γίνει κατανοητός ο τρόπος συσχέτισης και λειτουργίας των στοιχείων αυτών η διαδικασία υλοποίησης του User Interface της εφαρμογής απλοποιείται σημαντικά.

Για παράδειγμα το κουμπί (button) είναι ένα από τα πιο σημαντικά στοιχεία καθώς ανταποκρίνεται στο «κλικ» που θα γίνει από τον χρήστη ώστε να πυροδοτήσει ή να επιβεβαιώσει μια εντολή. Αρχικά μόλις εισαχθεί στον καμβά έχει τη μορφή:



Όπως κάθε στοιχείο έχει το δικό του πίνακα (Inspector) ώστε να ρυθμιστούν τα χαρακτηριστικά του.

Στην εφαρμογή μας όλα τα κουμπιά τροποποιήθηκαν σύμφωνα με τις προσωπικές προτιμήσεις. Για παράδειγμα θα παρουσιαστεί η πρώτη σκηνή της εφαρμογής, την οποία αντικρίζει ο χρήστης κατά την είσοδο του στην εφαρμογή.





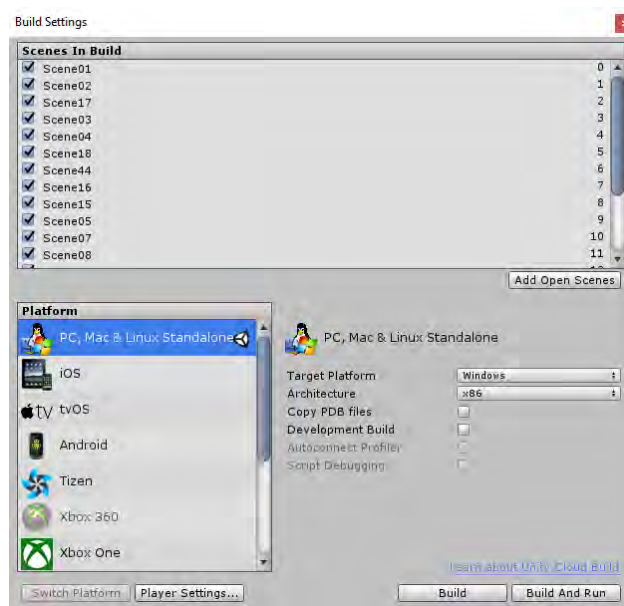
*Εικόνα 42: Σκηνή εισόδου στην εφαρμογή*

Το κουμπί Start έχει μορφοποιηθεί ώστε να αλλάξει η εικόνα του σε κάτι το οποίο είναι πιο παιχνιδιάρικο και όμορφο στα μάτια του χρήστη και να γίνει η είσοδος στην εφαρμογή. Δεξιά παρουσιάζονται κάποιες δυνατότητες του κουμπιού:

- Target Graphic: Εισαγωγή εικόνας που επιθυμούμε να εμφανίζεται στο κουμπί, οπότε χάνει την αρχική του μορφή.
- Interactable: Ενεργοποίηση/Απενεργοποίηση λειτουργικότητας του.
- On Click(): Λειτουργεί σαν πίνακας ελέγχου για το κουμπί. Στο συγκεκριμένο έχει ενσωματωθεί το game object με όνομα \_Manager το οποίο περιλαμβάνει τον κώδικα για την εναλλαγή της επόμενης σκηνής. Δίνεται πρόσβαση στις κλάσεις του κώδικα και αυτό είναι εμφανές καθώς επιλέγουμε από μια λίστα με τις κλάσεις του κώδικα αυτή που θέλουμε να πραγματοποιείται. Σε αυτή την περίπτωση είναι η εντολή LoadScene από την κλάση ChangingScenes. Επίσης, πληκτρολογούμε σε ποια σκηνή επιθυμούμε να μεταβούμε.

Αφού γίνει η διαδικασία στο On Click() το τελικό βήμα ώστε να πραγματοποιηθεί η επιτυχής εναλλαγή σκηνής είναι να γίνει και το «χτίσιμο» (build).

Ο προγραμματιστής της εφαρμογής πρέπει να επιλέξει: File ➔ Build Settings. Εμφανίζεται η εξής εικόνα:



**Εικόνα 43: Build Settings**

Στο σημείο αυτό καθορίζεται η σειρά προτίμησης για την εναλλαγή σκηνών και η πλατφόρμα στην οποία επιθυμούμε να «χτίσουμε» την εφαρμογή. Μόλις ολοκληρωθούν επιτυχώς όλα αυτά τα βήματα η μετάβαση των σκηνών γίνεται επιτυχώς. Παρόμοια διαδικασία ακολουθείται για την εναλλαγή σκηνών σε όλη την εφαρμογή.

Συμπερασματικά, ο σχεδιασμός του User Interface είναι πολύ σημαντικός καθώς αφορά όλη την εμφάνιση της εφαρμογής. Οι γραμματοσειρές, οι εικόνες και όλο το σύνολο των στοιχείων που θα χρησιμοποιηθούν πρέπει να επιλεγθούν προσεκτικά από τον προγραμματιστή.

Στόχος της πτυχιακής εργασίας ήταν το UI που συνοδεύει την εφαρμογή να είναι απλό, κατανοητό από την πλευρά του χρήστη, ξεκούραστο και εύκολο στην διαδραστικότητα και την πλοήγηση.

### 5.2.7 CSV file (Comma – Separated Values)

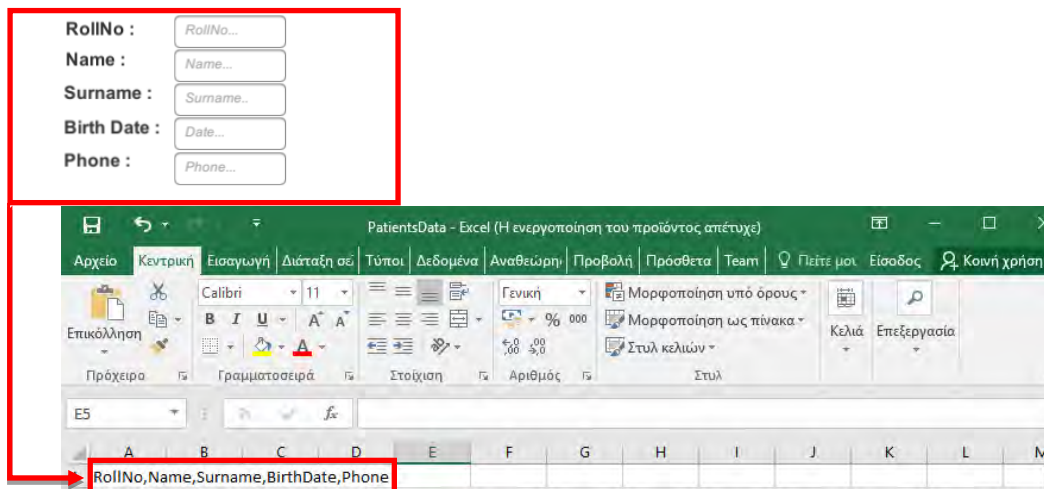
Κατά τη διάρκεια της υλοποίησης της εφαρμογής και εφόσον είχαν αντιμετωπιστεί αρκετά προβλήματα που εμφανίστηκαν, παρουσιάστηκε ακόμα ένα σημαντικό πρόβλημα το οποίο αφορούσε:

- 1) την καταχώρηση των στοιχείων ενός ασθενή κατά την εισαγωγή στο σύστημα,
- 2) την ανάκτηση ιατρικού ιστορικού από το σύστημα για οποιονδήποτε ασθενή και την εμφάνιση των δεδομένων του.

Έγιναν αρκετές προσπάθειες με αρχεία τύπου .txt ή excel αλλά το αποτέλεσμα δεν ήταν το επιθυμητό. Υπήρχαν περιπτώσεις όπου δεν γινόταν καθόλου η καταχώρηση των στοιχείων στο σύστημα αλλά ούτε και η ανάγνωση των δεδομένων από το σύστημα παρόλο που δοκιμαστικά είχαν εισαχθεί χειροκίνητα.

Στο σημείο αυτό βοήθησε η χρήση των αρχείων τύπου CSV. Όπως δηλώνει και το πλήρες όνομα του comma – separated values, δηλαδή τιμές χωρισμένες μεταξύ τους με κόμμα, η μορφή αυτού του αρχείου επιτρέπει τα δεδομένα να αποθηκεύονται με τη μορφή ενός δομημένου πίνακα. Κάθε γραμμή του πίνακα αποτελεί μια εγγραφή. Κάθε εγγραφή αποτελείται από ένα ή περισσότερα πεδία διαχωρισμένα μεταξύ του με κόμμα. Τα αρχεία αυτού του τύπου μπορούν κυρίως να χρησιμοποιηθούν και να τροποποιηθούν μέσω υπολογιστικών προγραμμάτων όπως το Microsoft Excel και άλλων που αφορούν διαχείριση βάσεων δεδομένων.

Η χρήση του ήταν απαραίτητη για την εφαρμογή μας, διότι μετά την ολοκλήρωση της συμπλήρωσης των δεδομένων ενός νέου ασθενή στη φόρμα εγγραφής, τα δεδομένα αυτά έπρεπε να αποθηκευτούν ως μια ενιαία εγγραφή στο σύστημα ώστε να ανακτηθεί στο σύστημα αργότερα όλη η εγγραφή. Το αρχείο που δημιουργήθηκε για αυτή τη λειτουργία ονομάζεται PatientsData.csv και περιέχει όλους τους εγγεγραμμένους ασθενείς στο σύστημα με τις πληροφορίες που συμπληρώθηκαν κατά τη συμπλήρωση της φόρμας εγγραφής από την εφαρμογή.



**Εικόνα 44: Καταχώρηση από την φόρμα της εφαρμογής στο αρχείο CSV**

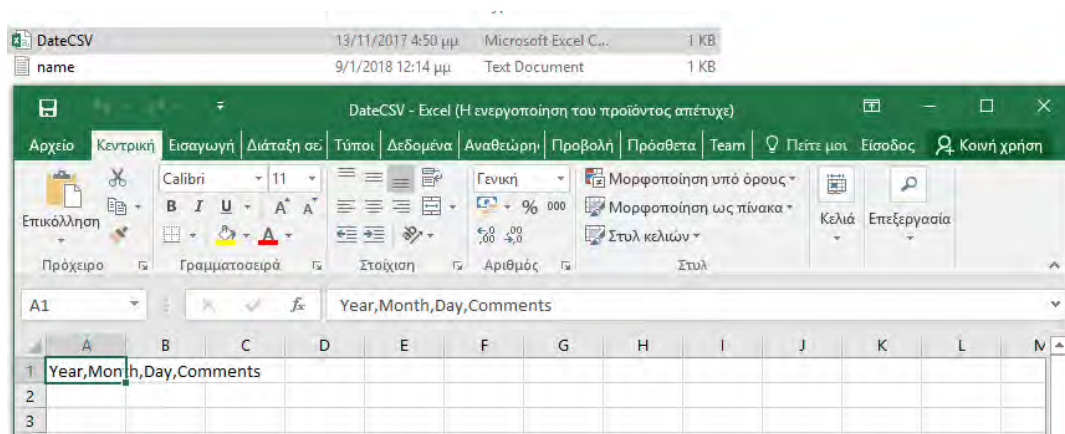
Επιπλέον, η μορφή αυτών των αρχείων μας έλυσε τα χέρια διότι προγραμματιστικά μπορούσαμε να καλέσουμε οποιοδήποτε πεδίο του CSV. Αυτό βοήθησε για την αυτόματη δημιουργία των φακέλων των ασθενών με το όνομα τους και την ημερομηνία γέννησης τους. Ακόμα, οι πληροφορίες που περιλαμβάνονται στο αρχείο name.txt που δημιουργείται αυτόματα με τη δημιουργία του ασθενούς προέρχονται από τις εντολές που δόθηκαν προγραμματιστικά για τα περιεχόμενα των πεδίων του CSV.

Ωστόσο, κατά τη διάρκεια της χρήσης αρχείων αυτού του τύπου πρόκυπτε συνεχώς ένα συγκεκριμένο σφάλμα, το οποίο δεν επέτρεπε την προσπέλαση και την ορθή εγγραφή των δεδομένων στο σύστημα. Το σφάλμα αυτό αφορούσε την δημιουργία μια κενής γραμμής είτε μετά από κάποια εγγραφή είτε στο τέλος του αρχείου. Αυτό είχε ως αποτέλεσμα να μη γίνεται σωστά η εργασία που είχαμε ορίσει προγραμματιστικά. Μετά από πολλές δοκιμές και αλλαγές στον κώδικα, το πρόβλημα αντιμετωπίστηκε αφαιρώντας προγραμματιστικά την κάθε κενή γραμμή που δημιουργούνταν αυτόματα από το αρχείο και το σύστημα.

Το επόμενο που έπρεπε να λυθεί, ήταν ο τρόπος με τον οποίο θα γινόταν η ανάκτηση πληροφορίας για το ιατρικό ιστορικό του κάθε ασθενούς στην οθόνη, με ποιον τρόπο θα γινόταν νέα καταχώρηση σε αυτό, κάποια τροποποίηση ή κάποια διαγραφή εγγραφής. Η χρήση του CSV κρίθηκε απαραίτητη για ακόμα μια φορά. Το πρόβλημα επιλύθηκε μετά από πολλές προσπάθειες και δοκιμές επιτυχώς.

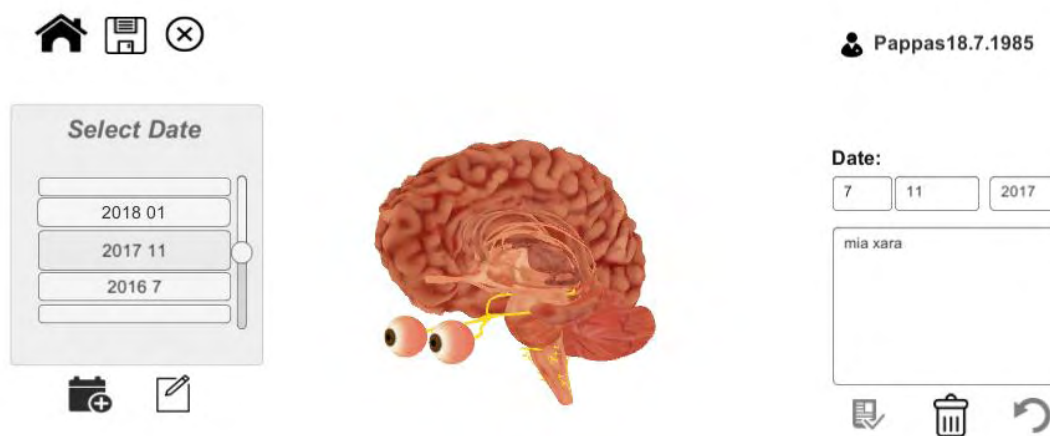
Ακόμα, η διαχείριση των αρχείων CSV αποδείχθηκε ιδιαίτερα δύσκολη καθώς η χρήση έτοιμων εντολών για αυτό το σκοπό παρουσίαζαν πρόβλημα κατά την ανανέωση προβολής των στοιχείων που περιείχαν. Αυτό οφειλόταν στην αρχική δέσμευση μνήμης του συστήματος στην οποία δεν ήταν δυνατή οποιαδήποτε ενημέρωση. Επιλέχθηκε έτσι η δημιουργία υπορουτινών, οι οποίες βασίστηκαν σε εντολές συστήματος προκειμένου να μπορεί να γίνει διαχείριση φακέλων και αρχείων, κατ' επέκταση και των αρχείων CSV.

Αρχικά, μέσα στον φάκελο του ασθενούς που δημιουργείται αυτόματα από το σύστημα αποφασίσαμε να δημιουργείται και ένα κενό αρχείο τύπου CSV και με όνομα DateCSV.csv. Το κενό αυτό αρχείο από εγγραφές περιέχει μόνο το όνομα των πεδίων που μπορούν να συμπληρωθούν μέσω της εφαρμογής. Στα περιεχόμενα του υπάρχουν διαχωρισμένα με κόμμα η ημερομηνία της καταχώρησης (Year, Month, Day) και κάποιο σχόλιο για την καταχώρηση (Comments).

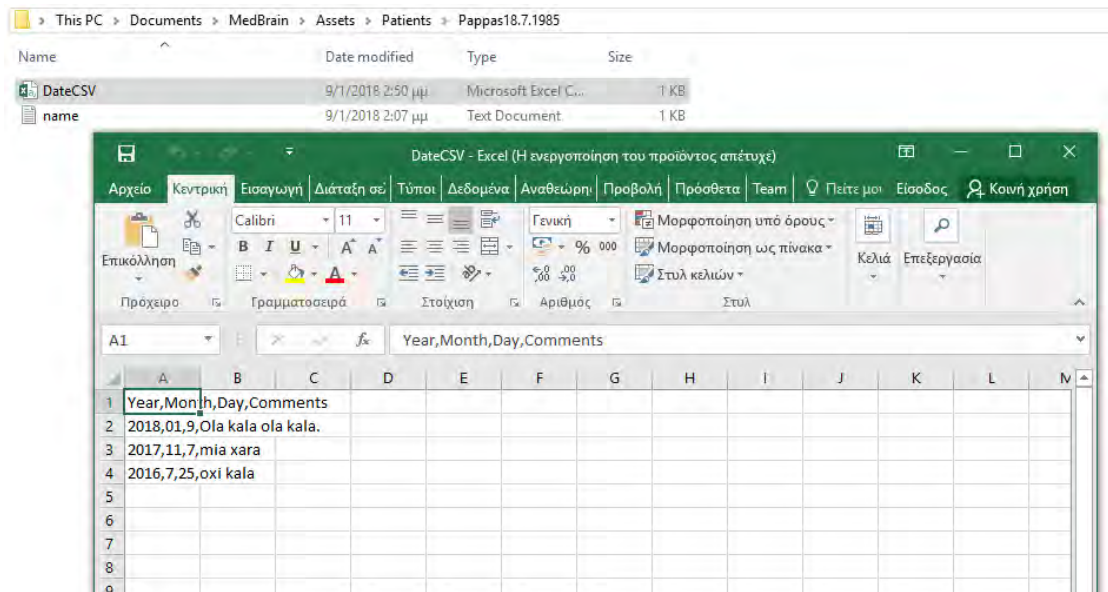


**Εικόνα 45: Μορφή αρχείου CSV**

Μέσω της εφαρμογής πραγματοποιείται η προσθήκη δεδομένων σε αυτά τα πεδία. Μετά την αποθήκευση των αλλαγών, τα περιεχόμενα του DateCSV.csv ανανεώνονται με αποτέλεσμα στην σκηνή της εφαρμογής να παρουσιάζονται και να μπορούν να επεξεργαστούν.



**Εικόνα 46: Εμφάνιση δεδομένων από το προσωπικό αρχείο CSV του ασθενούς**



Εικόνα 47: Παρουσίαση εγγραφών μέσω της εφαρμογής στο CSV με την ανάλογη μορφή

### 5.2.8 Δημιουργία προσωπικού φακέλου ασθενούς

Στη συνέχεια της υλοποίησης και εφόσον ξεπεράστηκε το πρόβλημα της διαχείρισης των δεδομένων με τη χρήση των αρχείων CSV, όπως προαναφέρθηκε, έπρεπε να λυθεί το ζήτημα της δημιουργίας προσωπικού φακέλου για κάθε ασθενή ξεχωριστά. Ο προσωπικός φάκελος είναι πολύ σημαντικός, καθώς μέσα σε αυτόν βρίσκεται όλη η πληροφορία που έχει καταχωρηθεί για έναν και μόνο ασθενή και τον προσδιορίζει. Για το λόγο αυτό, το στάδιο δημιουργίας του προσωπικού φακέλου αποφασίστηκε να είναι εκείνο κατά το οποίο θα γίνεται η εισαγωγή ενός νέου ασθενούς στο σύστημα μέσω της εφαρμογής.

Όσο αφορά την εισαγωγή νέου ασθενή στο σύστημα υπάρχει ειδική φόρμα καταχώρησης του, στην οποία αναγράφονται όλα τα προσωπικά στοιχεία που πρέπει να γνωρίζει ο χρήστης για έναν ασθενή. Μόλις ολοκληρωθεί η συμπλήρωση της φόρμας για τον νέο ασθενή το μόνο που πρέπει να γίνει από τον χρήστη είναι να πατήσει το κουμπί με το όνομα «ADD» ώστε να γίνει προσθήκη του στο σύστημα και να δημιουργηθεί ο αντίστοιχος ιατρικός φάκελος.

Ο ιατρικός φάκελος που δημιουργείται φέρει ως όνομα το επίθετο του ασθενούς και την ημερομηνία γέννησης του, ώστε αν υπάρχει στο σύστημα κάποιος ασθενής με το ίδιο ονοματεπώνυμο να μη δημιουργηθεί σύγχυση αλλά να γίνει η σωστή επιλογή. Εκτός από τον ιατρικό φάκελο όπου έχει οριστεί προγραμματιστικά να δημιουργείται, μέσα σε αυτόν δημιουργούνται ένα κενό αρχείο για την καταχώρηση των δεδομένων του ασθενούς με όνομα DateCSV και ένα αρχείο κειμένου μορφής name.txt. Το αρχείο name.txt περιέχει επίσης το επίθετο και η ημερομηνία γέννησης του ασθενούς και βρίσκει εφαρμογή στη σκηνή της επεξεργασίας του ιατρικού



ιστορικού του ασθενούς. Είναι υπεύθυνο για την εμφάνιση τίνος ασθενούς το ιστορικό επεξεργάζεται.

Πιο αναπαραστατικά φαίνονται οι διαδικασίες που περιεγράφηκαν στις ακόλουθες εικόνες.

## Patient Register

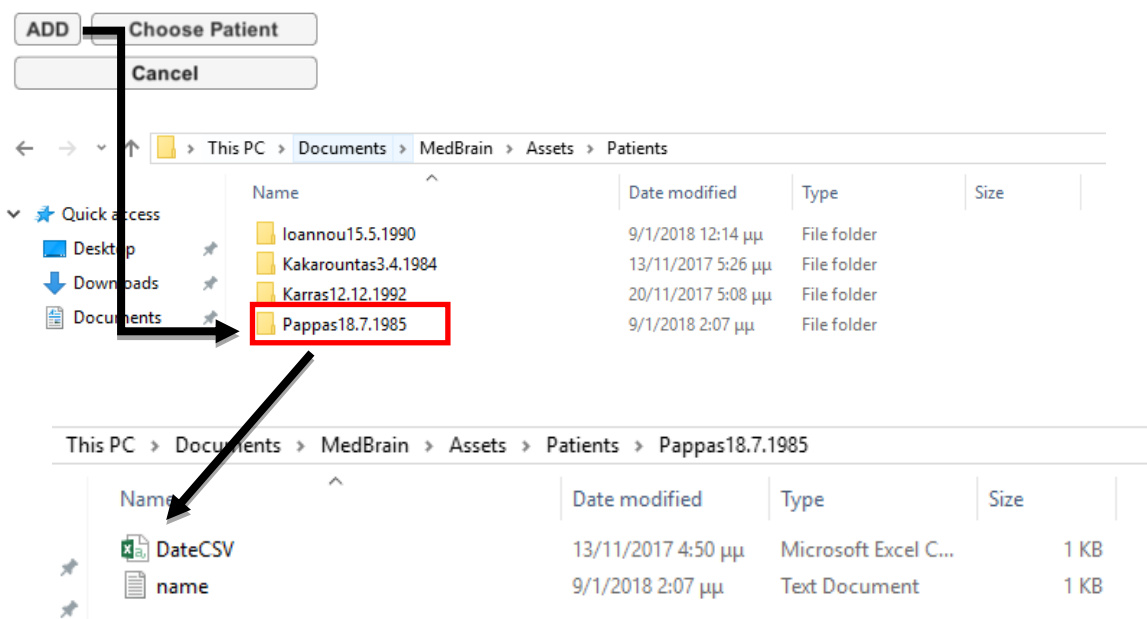
RollNo :

Name :

Surname :

Birth Date :

Phone :



**Εικόνα 48:** Με το «ADD» γίνεται η προσθήκη του ασθενούς στο σύστημα και η δημιουργία του φακέλου του. Μέσα στον φάκελο δημιουργείται το αρχείο για την καταγραφή του ιστορικού με όνομα «DateCSV» και το αρχείο name.txt.



Εικόνα 49: Εμφάνιση πληροφορίας από το name.txt

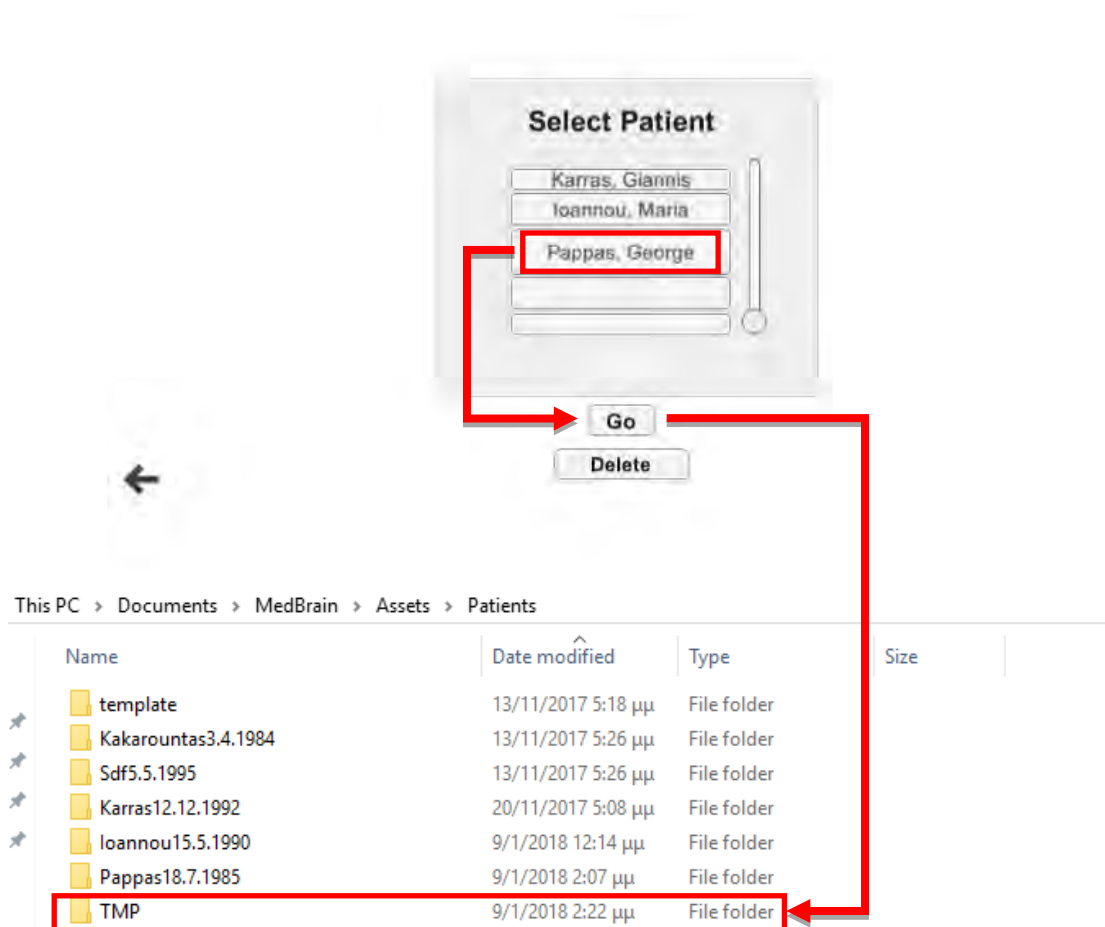
### 5.2.9 Προσωρινός φάκελος αποθήκευσης (Temporary File)

Ένα επιπλέον ζήτημα που προέκυψε κατά τον σχεδιασμό και την υλοποίηση της διεπαφής αφορούσε την ορθή διαχείριση των αρχείων. Για τον λόγο αυτό, επιλέχθηκε να δημιουργείται αυτόματα ένας προσωρινός φάκελος (TMP) μετά την επιλογή ενός ασθενή από τη λίστα των καταχωρημένων ασθενών.

Η χρήση του προσωρινού φακέλου κρίθηκε πολύ σημαντική, καθώς τα κύρια αρχεία δεν μπορούν να επηρεαστούν αν πραγματοποιηθεί οποιαδήποτε τροποποίηση κατά τη διάρκεια της επεξεργασίας του ιστορικού. Αν ο χρήστης επιθυμεί να αποθηκεύσει τις αλλαγές που πραγματοποίησε πρώτα θα γίνει αποθήκευση στον TMP, ο οποίος με τη σειρά του θα αντιγράψει τα νέα δεδομένα στον κύριο ιατρικό φάκελο του συγκεκριμένου ασθενούς και έπειτα θα διαγραφεί αυτόματα από το σύστημα. Αν ο χρήστης πατήσει το κουμπί της ακύρωσης, οι αλλαγές που πραγματοποιήθηκαν δεν επηρεάζουν τις κύριες καταχωρήσεις και αυτόματα ο φάκελος TMP διαγράφεται. Με τον τρόπο αυτό αποφεύγεται οποιοδήποτε σφάλμα μπορεί να συμβεί κατά την επεξεργασία του ιατρικού ιστορικού είτε είναι η διαγραφή κάποιας καταχώρησης από το σύστημα, είτε κάποια διπλοεγγραφή κατά τη διαδικασία της αποθήκευσης είτε στο γενικό σύνολο να αλλοιωθούν και να χαθούν δεδομένα.

Στην παρακάτω εικόνα παρουσιάζεται η αυτοποιημένη διαδικασία που έχουμε ορίσει προγραμματιστικά να συμβαίνει κατά την επιλογή ενός συγκεκριμένου ασθενούς από τη λίστα των καταχωρήσεων.



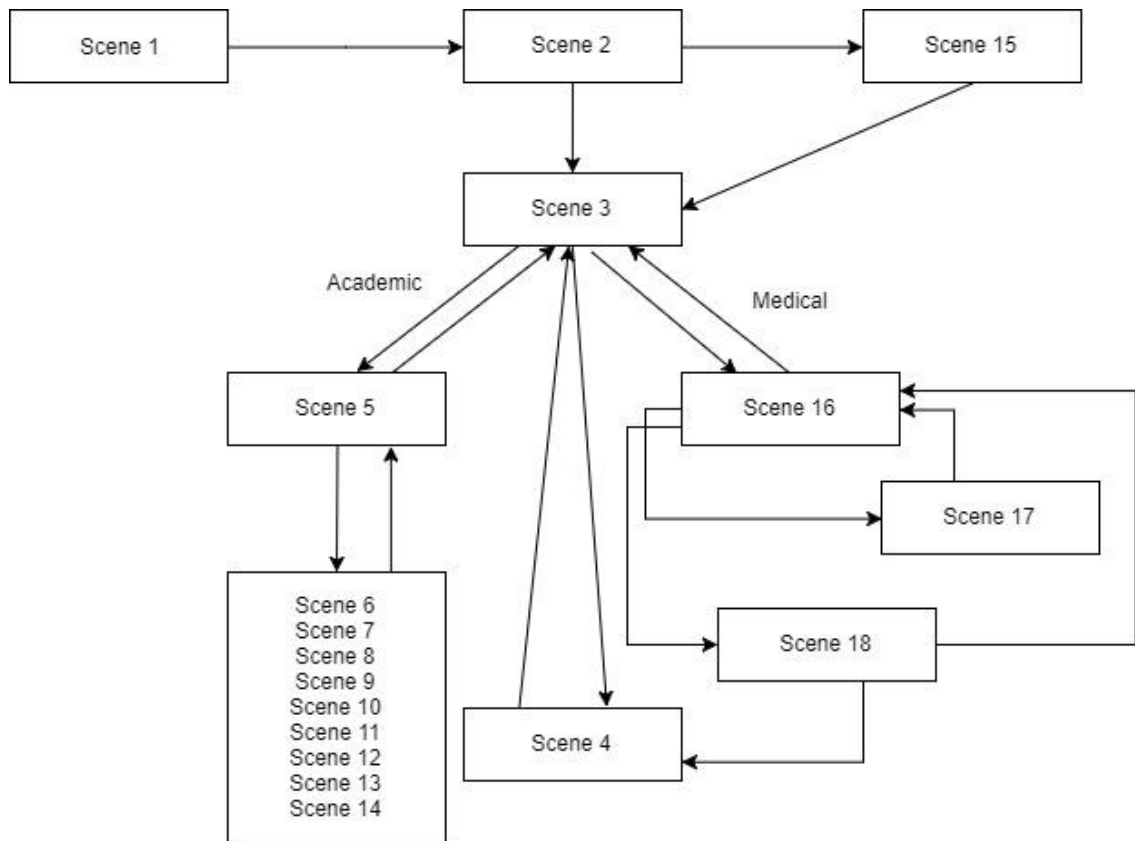


*Εικόνα 50: Δημιουργία προσωρινού φακέλου*

### 5.2.10 Τελική οργάνωση διεπαφής

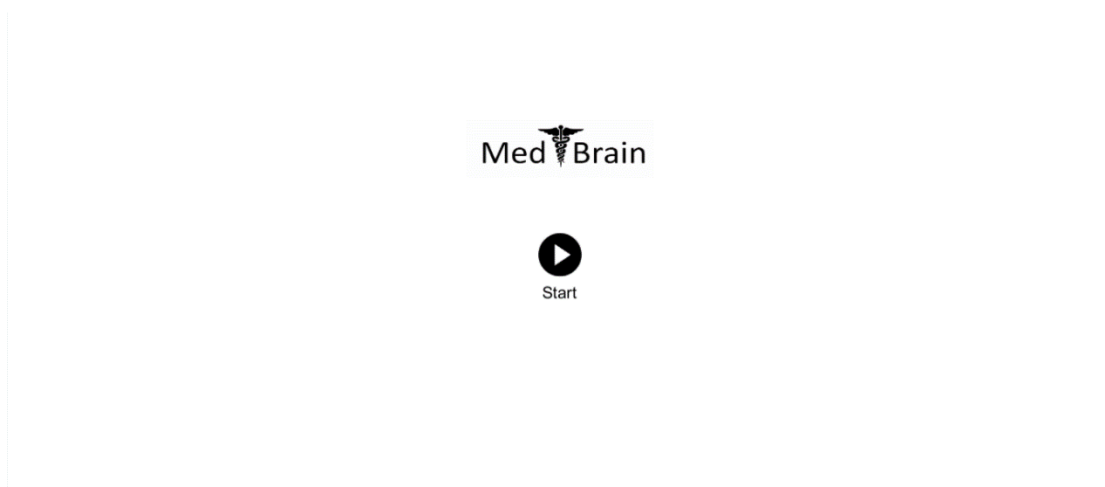
Συνολικά η διεπαφή που υλοποιήθηκε αποτελείται από δεκαοχτώ σκηνές οι οποίες είναι διαδοχικά συνδεδεμένες μεταξύ τους ώστε η εναλλαγή να γίνεται ομαλά και να οδηγεί στοχευμένα στην διεργασία που επιθυμούμε να συμβεί κάθε χρονική στιγμή.

Στο παρακάτω διάγραμμα παρουσιάζεται η σύνδεση τους.



**Εικόνα 51: Σχεδιάγραμμα σύνδεσης σκηνών**

Για να παρουσιαστεί με τον καλύτερο και πιο κατανοητό δυνατό τρόπο η λειτουργία της ακολουθούν μερικά στιγμιότυπα κατά τη διάρκεια της χρήσης της με μια σύντομη περιγραφή.



**Εικόνα 52: Έναρξη με το πάτημα του κουμπιού Start**

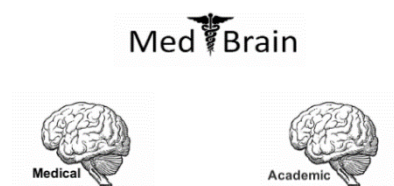
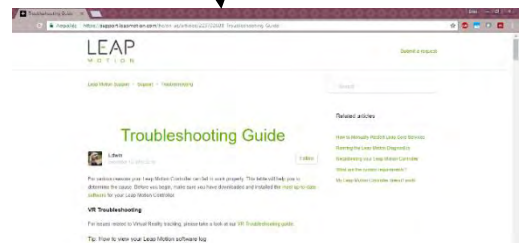


**Εικόνα 53:** Σε αυτό το σημείο ο χρήστης επιλέγει αν θα εξετάσει τη λειτουργικότητα του αισθητήρα ή όχι. (Scene 02)



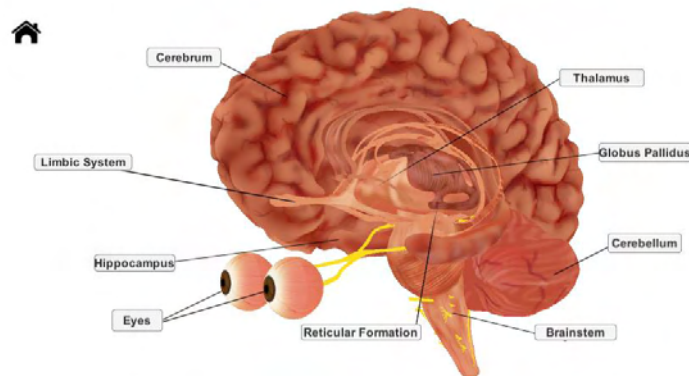
Did it work?

**Εικόνα 54:** Σκηνή ελέγχου. Αν καταφέρει να αλληλοεπιδράσει με τους κύβους θα είναι ορθή η λειτουργικότητα του αισθητήρα αν όχι θα οδηγηθεί σε ιστοσελίδα υποστήριξης του προβλήματος. (Scene 15)





*Εικόνα 55: : Μενού επιλογής για Medical ή Academic (Scene 03)*



*Εικόνα 56: Είσοδος στο Academic. Παρουσιάζεται το τρισδιάστατο μοντέλο του εγκεφάλου με τις αναγνωρισμένες περιοχές στις οποίες μπορεί να εισέλθει ο χρήστης. Υπάρχει και το εικονίδιο επιστροφής στο αρχικό μενού. (Scene 05)*



**Εικόνα 57:** : Ενδεικτικά είσοδος στην περιοχή της παρεγκεφαλίδας. Δεξιά υπάρχουν οι πληροφορίες μαζί με το εικονίδιο «More...» για μετάβαση στην διαδικτυακή πηγή. Αριστερά περιστρέφεται μεμονωμένα το αντίστοιχο τμήμα και υπάρχει και το εικονίδιο επιστροφής στην προηγούμενη σκηνή επιλογής τμήματος του εγκεφάλου. (Scene 11)

Αφού εξέλθει από το Academic κομμάτι και εισέρχεται στο Medical, η εικόνα που εμφανίζεται στο χρήστη είναι η ακόλουθη:



**Εικόνα 58:** Ο χρήστης εισάγει είτε κάποιον νέο ασθενή είτε επιλέγει από τη λίστα (Scene 16)

### Patient Register

RollNo :	<input type="text" value="5"/>
Name :	<input type="text" value="George"/>
Surname :	<input type="text" value="Pappas"/>
Birth Date :	<input type="text" value="18.7.1985"/>
Phone :	<input type="text" value="6945214789"/>

<input type="button" value="ADD"/>	<input type="button" value="Choose Patient"/>
<input type="button" value="Cancel"/>	

*Εικόνα 59: Προσθήκη νέου ασθενή και εμφάνιση φόρμας για την καταχώρηση του στο σύστημα. Με το «ADD» γίνεται η προσθήκη του στο σύστημα και η δημιουργία του φακέλου του. Μέσα στον φάκελο δημιουργείται το αρχείο για την καταγραφή του ιστορικού με όνομα «DateCSV» και το αρχείο name.txt. (Scene 17)*

Εάν ο χρήστης εισέλθει στη λίστα επιλογής ασθενή του παρουσιάζεται το εξής στιγμιότυπο:



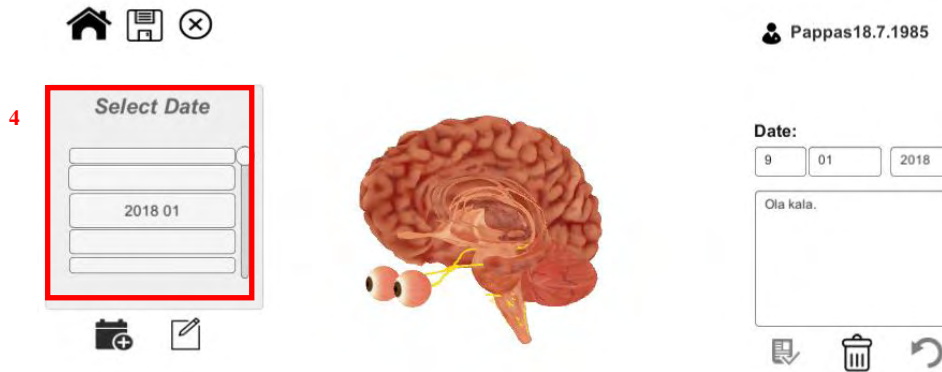
*Εικόνα 60: Έχει τη δυνατότητα να επιλέξει ή να διαγράψει κάποιον ασθενή. Επίσης, μπορεί να επιστρέψει πίσω. (Scene 18)*



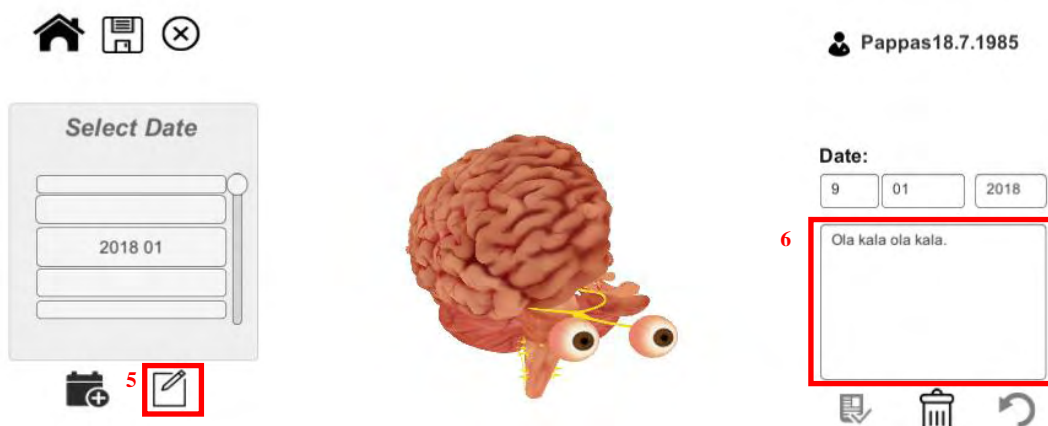
*Εικόνα 61: Το επόμενο βήμα είναι να εισάγει τον χρήστη στο χώρο επεξεργασίας του ιστορικού του ασθενούς. Λόγω ότι ο ασθενής είναι νέος δεν υπάρχει κάποια καταγραφή στο αρχείο των ημερομηνιών. (Scene 04)*



*Εικόνα 62: Εισαγωγή νέας εγγραφής στο ιστορικό όπως παρουσιάζονται σε βήματα. 1) επιλογή εικονιδίου για νέα προσθήκη, 2) είσοδος δεδομένων, 3) αποθήκευση εγγραφής.*



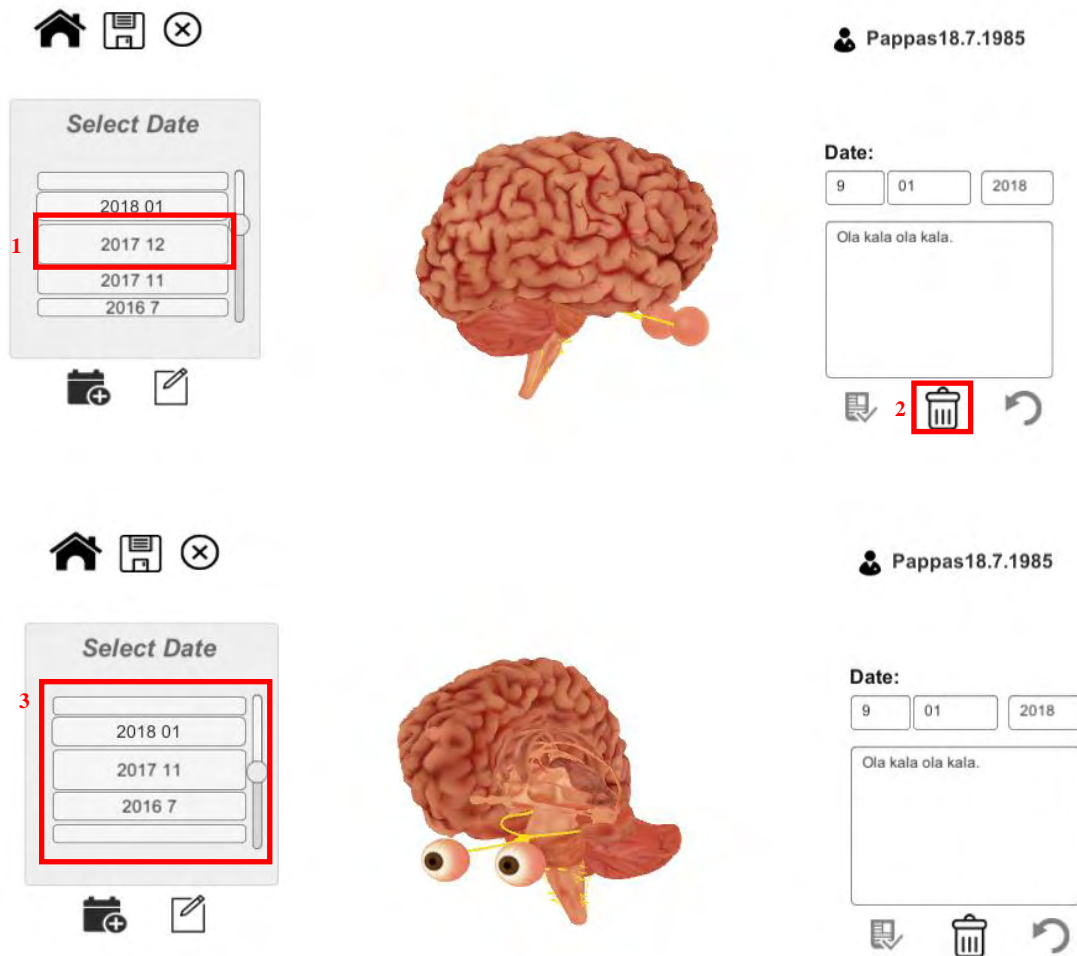
*Εικόνα 63: 4) Η λίστα των ημερομηνιών ανανεώθηκε αυτόματα και στα δεξιά εμφανίζονται τα περιεχόμενα της επιλεγμένης ημερομηνίας. Εφόσον υπάρχει αποθηκευμένη ημερομηνία ο χρήστης μπορεί να τροποποιήσει κάποια από τα σχόλια που τη συνοδεύουν.*



*Εικόνα 64: 5) Παρουσιάζεται το εικονίδιο για την επεξεργασία του σχόλιου της επιλεγμένης ημερομηνίας και 6) η νέα τροποποιημένη και αποθηκευμένη καταγραφή στο ιστορικό*



Παρακάτω ακολουθεί διαγραφή κάποιας εγγραφής σε βήματα.

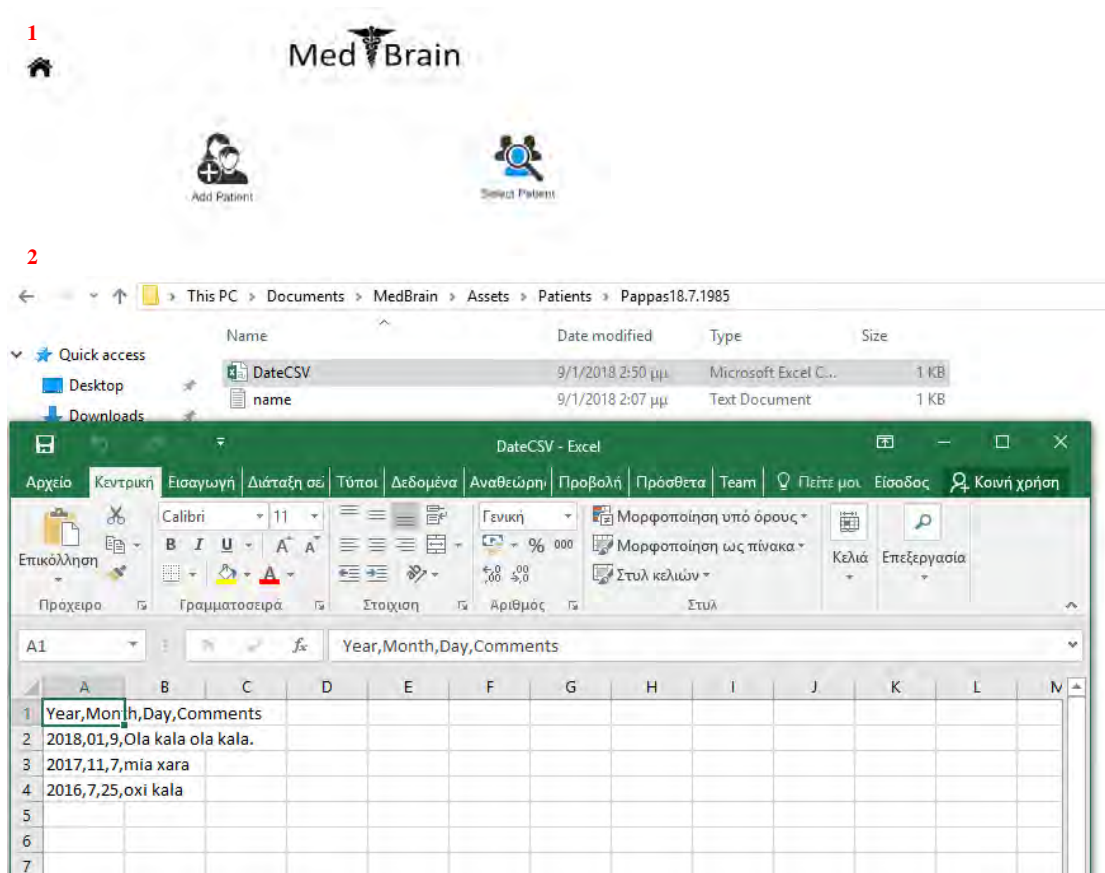


**Εικόνα 65: 1)Επιλογή ημερομηνίας, 2) επιλογή εικονιδίου διαγραφής, 3) ανανεωμένη λίστα των ημερομηνιών ιστορικού.**

Σε αυτό το σημείο και εφόσον έχει ολοκληρωθεί η εργασία που επιθυμεί ο χρήστης μπορεί είτε να αποθηκεύσει όλες τις αλλαγές από τον TMP στον αυθεντικό ιατρικό φάκελο του ασθενούς πατώντας το εικονίδιο της αποθήκευσης πάνω αριστερά, είτε να μην πραγματοποιήσει καμία αλλαγή πατώντας το εικονίδιο της ακύρωσης πάνω αριστερά.

Έστω επιλέγει να αποθηκεύσει τις αλλαγές:

Σχεδίαση καινοτόμου διεπαφής διάδρασης για ιατρικές εφαρμογές, μελέτη περίπτωσης: ανθρώπινος εγκέφαλος



**Εικόνα 66: 1) Επιστροφή στην προηγούμενη σκηνή, 2) ενημέρωση του αρχείου καταγραφών ημερομηνιών και άνοιγμα του για επαλήθευση της εργασίας που προηγήθηκε και τέλος συμβαίνει και η διαγραφή του TMP**

## 6. Ανάλυση εφαρμογής

Η εφαρμογή που αναπτύχθηκε στο πλαίσιο εκπόνησης της πτυχιακής εργασίας, ξεχωρίζει σε μεγάλο βαθμό από πληθώρα εφαρμογών που υπάρχουν στο εμπόριο, καθώς περιλαμβάνει χαρακτηριστικά και δυνατότητες που μέχρι στιγμής είναι μοναδικά.

Αρχικά, η εφαρμογή σχεδιάστηκε ώστε να εξυπηρετεί ένα μεγάλο πλήθος χρηστών και όχι μόνο ένα στοχευμένο κοινό. Οι επιλογές λειτουργίας που προσφέρονται στον χρήστη είναι δύο. Η μία επιλογή που δίνεται μέσα από το μενού που εμπεριέχεται είναι η εκπαιδευτική («Academic»). Στο κομμάτι αυτό είναι δυνατόν να έχουν πρόσβαση φοιτητές, εκπαιδευτικό προσωπικό αλλά και απλό κοινό χρηστών που επιθυμούν να μελετήσουν και να εμβαθύνουν στην ανατομία του ανθρώπινου εγκεφάλου. Μέσω του τρισδιάστατου ανατομικού μοντέλου που παρέχεται, οι χρήστες μπορούν να εξερευνήσουν μεμονωμένα περιοχές του εγκεφάλου. Το μοντέλο του εγκεφάλου είναι χωρισμένο σε οκτώ κομμάτια και κάθε περιοχή συνοδεύεται από το όνομα της, πληροφορίες σχετικά με την λειτουργία της καθώς και απεικονίζεται η μορφή της ξεχωριστά από το ολικό μοντέλο. Επιπλέον, προσφέρεται η δυνατότητα σύνδεσης σε ηλεκτρονική εξωτερική πηγή. Η πηγή αυτή επιλέχθηκε τυχαία και είναι μια ιστοσελίδα με ιατρικό εκπαιδευτικό χαρακτήρα. Περιλαμβάνει τρισδιάστατα όργανα του ανθρώπινου σώματος με αναλυτική περιγραφή και επιπλέον εμβάθυνση σε αυτά. Το όνομα της είναι : «<http://www.innerbody.com>».

Η άλλη επιλογή που προσφέρεται στους χρήστες είναι αυτή με ιατρικό χαρακτήρα («Medical»). Το κομμάτι αυτό σχεδιάστηκε και αναπτύχθηκε κυρίως για κάποιον ιατρό, κάποια νοσοκομειακή μονάδα ή για όλο το ιατρικό προσωπικό. Παρέχεται η δυνατότητα διαχείρισης όλων των ασθενών που έχουν καταχωρηθεί στο σύστημα και επεξεργασία του προσωπικού ιατρικού ιστορικού τους. Ένα από τα πλεονεκτήματα της εφαρμογής έγκειται στο γεγονός ότι η διαδικασία αυτή δεν απαιτεί κάποια τεχνολογική εκπαίδευση ή εξειδίκευση παρά μόνο τις βασικές γνώσεις χειρισμού ενός υπολογιστικού συστήματος. Η εφαρμογή σχεδιάστηκε με λιτές διεπαφές κατανοητές προς τη χρήση και τη λειτουργία τους. Η ροή της εφαρμογής είναι ομαλή και σε κάθε στάδιο δίνεται η δυνατότητα της επιστροφής στην προηγούμενη σκηνή ή στο αρχικό μενού επιλογής. Στο ιατρικό αυτό κομμάτι της εφαρμογής μπορεί να γίνει εκ νέου εισαγωγή ενός ασθενή στο σύστημα, μέσω της φόρμας εγγραφής που υπάρχει. Επίσης, αν ο ασθενής είναι ήδη εγγεγραμμένος στο σύστημα μπορεί να γίνει η επιλογή του από τη λίστα των καταγεγραμμένων ασθενών. Τέλος, εμφανίζεται στην οθόνη η σκηνή με το προσωπικό ιστορικό του ασθενούς. Δίνεται η δυνατότητα για προσθήκη νέας εγγραφής, τροποποίησης κάποιας υπάρχουσας εγγραφής ή επιλογή για ολική διαγραφή κάποιας εγγραφής από το ιστορικό.

Μέχρι και σήμερα δεν υπάρχει κάποια αντίστοιχη εφαρμογή που να συνδυάζει εκπαιδευτική χρήση αλλά και επαγγελματική από το χώρο της υγείας. Το γεγονός

όμως ότι ξεχωρίζει από όλες τις άλλες οφείλεται λόγω της καινοτόμου διεπαφής χρήστη που αναπτύχθηκε συγχρόνως. Η διεπαφή αυτή βασίζεται και έχει συσχετιστεί με τον αισθητήρα αναγνώρισης χειρονομιών, Leap Motion.

Η σκέψη ήταν να σχεδιαστεί μια εφαρμογή που θα επιτρέπει στους χρήστες να αλληλοεπιδρούν μόνο με τη φυσική κίνηση του χεριού τους. Συνδυάζοντας απλές ή πιο σύνθετες χειρονομίες θα πραγματοποιούν και κάποια λειτουργία. Συγκεκριμένες λειτουργίες έχουν προγραμματιστεί να μεταφράζονται σε ενέργειες. Για παράδειγμα, η κίνηση που θα κάνει ο χρήστης και είναι όμοια με το πάτημα ενός κουμπιού στο πληκτρολόγιο του υπολογιστή, έχει μεταφραστεί στην διεπαφή να πραγματοποιεί το πάτημα ενός κουμπιού από τα στοιχεία UI που την απαρτίζουν. Η διεπαφή αυτή προσφέρει μια μοναδική εμπειρία, διότι ο χρήστης είναι μέρος του συστήματος και υπεύθυνος για τον έλεγχο του. Δε χρειάζεται να χρησιμοποιήσει επιπλέον περιφερειακές συσκευές παρά μόνο τα χέρια του. Ο χειριστής του αισθητήρα δε χρειάζεται να λάβει κάποια εκπαίδευση για τη λειτουργία του. Το μόνο που χρειάζεται είναι να εξοικειωθεί με τη συσκευή του Leap Motion μέσω κατάλληλων εφαρμογών που διαθέτει το leap motion app store.

Με λίγα λόγια αναπτύχθηκε ένα κατάλληλο και ευέλικτο User Interface περιβάλλον που επιτρέπει την επικοινωνία του Unity με το Leap Motion, το οποίο είναι στοχευμένο στο χρήστη. Η διεπαφή που υλοποιήθηκε αποτελεί ένα κεντρικό άξονα και είναι πολύ σημαντική για περαιτέρω εξέλιξη και αναβάθμιση της εφαρμογής.

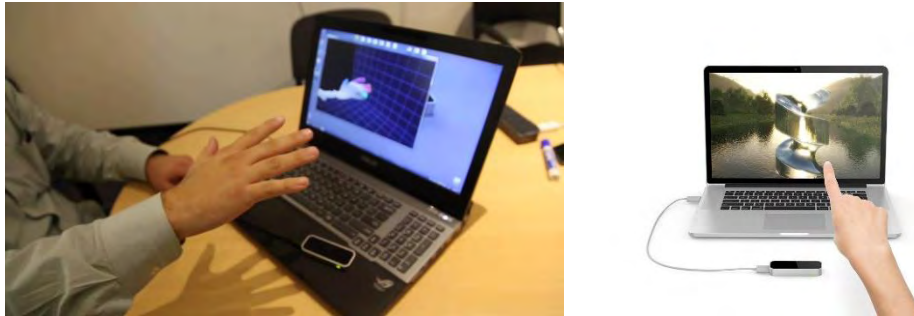
Νέα τρισδιάστατα ανατομικά μοντέλα του ανθρώπινου σώματος δύνανται να προστεθούν στην ήδη υπάρχουσα διεπαφή. Ακόμα, νέες συνθέσεις χειρονομιών μπορούν να ενσωματωθούν και να εμπλουτίσουν τις λειτουργίες που μπορεί να πραγματοποιήσει ο χρήστης στο σύστημα.

## **6.1 Αναγνωρισμένες χειρονομίες**

Μέχρι στιγμής έχουν οριστεί να αναγνωρίζονται από τον αισθητήρα του Leap Motion συγκεκριμένες χειρονομίες και να μεταφράζονται σε συγκεκριμένες λειτουργίες στο σύστημα.

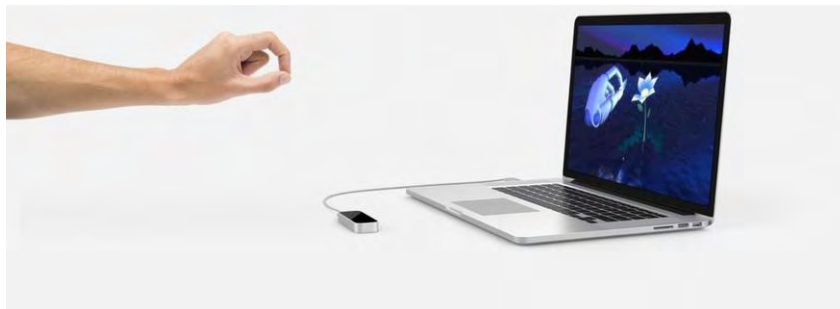
Αρχικά, μια από τις πιο βασικές αναγνωρισμένες κινήσεις είναι η πλοήγηση του χεριού μέσα στην περιοχή αναγνώρισης της συσκευής του Leap Motion. Ανάλογα τη φορά και την κατεύθυνση του χεριού αναγνωρίζεται και η κίνηση σε ένα σύστημα συντεταγμένων (x,y,z). Η πλοήγηση του χεριού στο χώρο μεταφράζεται σαν την κίνηση του βέλους από την περιφερειακή συσκευή του ποντικιού. Τα εκτεταμένα πέντε δάκτυλα είτε του δεξιού είτε του αριστερού χεριού ελέγχουν την κίνηση. Επίσης, για μεγαλύτερη ευκολία επιλέχθηκε να ελέγχει την κίνηση και ο δείκτης του κάθε χεριού.

Ενδεικτικά παρουσιάζονται οι δύο χειρονομίες στην ακόλουθη εικόνα.



**Εικόνα 67: Μορφή αναγνωρισμένων χειρονομιών**

Ακόμα στη διεπαφή έπρεπε να οριστεί η κίνηση, η οποία όταν θα αναγνωριζόταν από τον αισθητήρα να μεταφράζεται ως πάτημα του αριστερού κουμπιού της περιφερειακής συσκευής του ποντικιού. Η κίνηση που επιλέχθηκε για αυτή τη λειτουργία είναι η ένωση του αντίχειρα και του δείκτη (pinch). Μόλις ο χρήστης σχηματίσει αυτή τη χειρονομία αυτόματα θα πραγματοποιήσει και την επιλογή οποιουδήποτε στοιχείου της εφαρμογής.



**Εικόνα 68: Χειρονομία pinch**

Οι χειρονομίες ορίστηκαν να αναγνωρίζονται τόσο αν πραγματοποιούνται από το δεξί χέρι όσο και αν πραγματοποιούνται από το αριστερό δίνοντας ευελιξία στους χρήστες.

Μελλοντικά, στόχος είναι να αναγνωριστούν και άλλες χειρονομίες και να μεταφραστούν σε συγκεκριμένες εντολές και λειτουργίες. Για παράδειγμα, η περιστροφή του εγκεφάλου που πραγματοποιείται στην εφαρμογή αυτόματα και με μια συγκεκριμένη συχνότητα να γίνεται από τον χρήστη στο επίπεδο που εκείνος επιθυμεί. Η χειρονομία που θα πραγματοποιεί την περιστροφή θα σχετίζεται με την περιστροφή του καρπού του χεριού και συγχρόνως των δακτύλων.

## 7. Συμπεράσματα και μελλοντικές επεκτάσεις

Στο κεφάλαιο αυτό θα παρουσιαστούν τα συμπεράσματα που προέκυψαν από τον σχεδιασμό και την υλοποίηση της πτυχιακής εργασίας. Επιπλέον, θα διατυπωθούν κάποιες σκέψεις και στόχοι για μελλοντική επέκταση της λειτουργίας της εφαρμογής.

### 7.1 Συμπεράσματα

Ακολουθώντας παρουσιάζονται τα κυριότερα συμπεράσματα που προέκυψαν από την υλοποίηση της διεπαφής σε συνδυασμό με την εφαρμογή που αναπτύχθηκε.

Αρχικά, ο σχεδιασμός και η υλοποίηση πραγματοποιήθηκε ώστε ο χρήστης να μπορεί να αλληλοεπιδρά με το σύστημα με τρόπο απλό, εύκολο και ευέλικτο, δίχως να απαιτείται κάποια εκπαίδευση ή εξειδίκευση. Για το λόγο αυτό η ηλικία των χρηστών που χρησιμοποιούν τη διεπαφή μπορεί να ποικίλλει. Επιπλέον, λόγω της τεχνολογίας του αισθητήρα που ήδη υπάρχει αλλά και της πλατφόρμας του Unity, η οποία συνεχώς ανανεώνεται, υπάρχει η δυνατότητα και το πλαίσιο της αναβάθμισης του συστήματος διεπαφής - εφαρμογής. Γενικά, προσφέρεται η δυνατότητα ενσωμάτωσης και αλληλεπίδρασης με άλλα τρισδιάστατα μοντέλα του ανθρώπινου σώματος. Ακόμα, η διεπαφή έχει προσαρμοστεί ώστε να μπορεί να χρησιμοποιούν οι χρήστες τόσο το δεξί όσο και το αριστερό χέρι. Οι χειρονομίες που έχουν οριστεί να εκτελούν τις επιμέρους λειτουργίες είναι εύκολες στο σχηματισμό. Τέλος, υπάρχει η δυνατότητα μετατροπής της εφαρμογής σε περιβάλλον εικονικής πραγματικότητας μέσω της πλατφόρμας του Unity και του ειδικού λογισμικού του Leap Motion Virtual Reality.

### 7.2 Μελλοντικές επεκτάσεις

Η διεπαφή που δημιουργήθηκε στο πλαίσιο της πτυχιακής εργασίας καθώς και η εφαρμογή που έχει αναπτυχθεί αποτελούν βάση και κύριο άξονα για επιπλέον εμπλοτύιση και επέκταση της εφαρμογής. Μελλοντικός στόχος είναι να προστεθούν επιμέρους τρισδιάστατα ανατομικά μοντέλα του ανθρώπινου οργανισμού στην εφαρμογή. Για κάθε επιμέρους τρισδιάστατο όργανο θα υπάρχει τόσο η λειτουργία του «Academic» όσο και η λειτουργία του «Medical». Στο τέλος, όλα αυτά τα τρισδιάστατα ανατομικά μοντέλα θα προστεθούν σε έναν εικονικό σκελετό αναπαράστασης ανθρώπινου σώματος ώστε να αποτελούν ένα ολοκληρωμένο σύστημα αλληλεπίδρασης του χρήστη με την ανατομία και τη φυσιολογία του ανθρώπινου οργανισμού.

Επιπλέον, στόχος είναι και ο εμπλουτισμός των δύο ενοτήτων που αποτελούν την εφαρμογή. Στο «Academic» κομμάτι πρόκειται να προστεθεί επιπλέον οπτικοακουστικό υλικό και ίσως να το συνοδεύει κάποιο quiz με ερωτήσεις πάνω στο περιεχόμενο που επέλεξε να μελετήσει ο χρήστης. Σκοπός είναι η εφαρμογή να

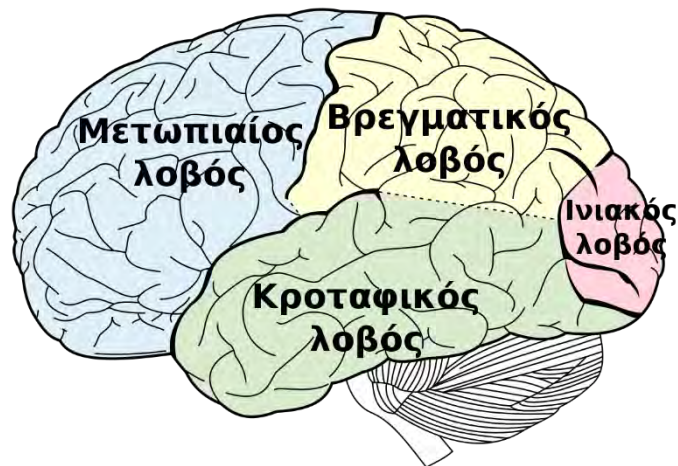
προσφέρει μια ολοκληρωμένη, διαδραστική και διασκεδαστική εκπαιδευτική εμπειρία.

Τέλος, όσο αφορά το «Medical» μέρος της εφαρμογής μια μελλοντική προσθήκη είναι να μπορεί να γίνεται σηματοδότηση των περιοχών (pin annotation) του τρισδιάστατου μοντέλου του εγκεφάλου που υπάρχει. Ο χρήστης θα μπορεί να μαρκάρει τον τομέα του εγκεφάλου που επιθυμεί με ένα σημάδι (pin), στο οποίο θα εμπεριέχονται επιπρόσθετες πληροφορίες και δεδομένα που έχει εισάγει ο χρήστης. Τα pins θα μπορούν να εμπλουτίζονται, να τροποποιούνται και να διαγράφονται. Επιπλέον, τα στοιχεία του ιατρικού φακέλου να είναι πλήρη υποστηρίζοντας αρχεία εικόνας και ήχου.

## ΠΑΡΑΡΤΗΜΑ Α

### ΕΓΚΕΦΑΛΟΣ

Ο ανθρώπινος εγκέφαλος είναι το κεντρικό όργανο του ανθρώπινου νευρικού συστήματος και μαζί με το νωτιαίο μυελό αποτελεί το κεντρικό νευρικό σύστημα. Είναι υπεύθυνος για τις περισσότερες λειτουργίες του οργανισμού, επεξεργάζεται, ενσωματώνει και συντονίζει τις πληροφορίες που λαμβάνει από τα αισθητήρια όργανα αποφασίζοντας για τις οδηγίες που θα στείλει στο σώμα. Ο εγκέφαλος εμπεριέχεται και προστατεύεται από τα οστά του κρανίου της κεφαλής. Το εγκεφαλικό στέλεχος (cerebrum) είναι το μεγαλύτερο μέρος του ανθρώπινου εγκεφάλου και διαιρείται σε δύο εγκεφαλικά στέλεχη. Κάθε ημισφαίριο χωρίζεται κατά κανόνα σε τέσσερις λοβούς: τον μετωπιαίο λοβό, τον κροταφικό λοβό, τον βρεγματικό λοβό και τον ινιακό λοβό. Παρόλο που το δεξί και το αριστερό ημισφαίριο έχουν γενικά παρόμοια μορφή, ορισμένες λειτουργίες συνδέονται μόνο με τη μια πλευρά, όπως για παράδειγμα η γλωσσική ικανότητα στην αριστερή ενώ η οπτική ικανότητα στη δεξιά. Τα ημισφαίρια συνδέονται μεταξύ τους μέσω νευρικών οδών. [34]



Εικόνα Α: Λοβοί εγκεφάλου

- **Μετωπιαίος λοβός**

Είναι ο μεγαλύτερος από όλους τους υπόλοιπους λοβούς του εγκεφάλου και βρίσκεται μπροστά από τον βρεγματικό λοβό με τον οποίο χωρίζεται από την κεντρική αύλακα και μπροστά και πάνω από τον κροταφικό λοβό, με τον οποίο χωρίζεται από την πλάγια σχισμή. Μερικές από τις πολλές λειτουργίες του μετωπιαίου λοβού περιλαμβάνουν τον συντονισμό των εθελοντικών κινήσεων



όπως το περπάτημα. Επίσης, είναι υπεύθυνος για τη διατήρηση της μνήμης. Παίζει κύριο ρόλο στην κατανόηση της γλώσσας και της ομιλίας. Ακόμα, σε αυτόν στηρίζεται η ανάπτυξη της προσωπικότητας του κάθε ανθρώπου. [35] [36]

- **Βρεγματικός λοβός**

Ο συγκεκριμένος λοβός εκτείνεται προς τα εμπρός και προς τα κάτω μέχρι τον κροταφικό λοβό. Είναι υπεύθυνος για την εκδήλωση ηθελημένων κινήσεων. Στον συγκεκριμένο λοβό βρίσκεται η θέση ελέγχου για την οπτική ικανότητα και για την αντίληψη της αφής. Σε περίπτωση βλάβης εμφανίζεται αδυναμία ονομασίας αντικειμένων, δυσκολία στην εκτέλεση μαθηματικών πράξεων καθώς και αδυναμία συγκέντρωσης της οπτικής προσοχής. [37]

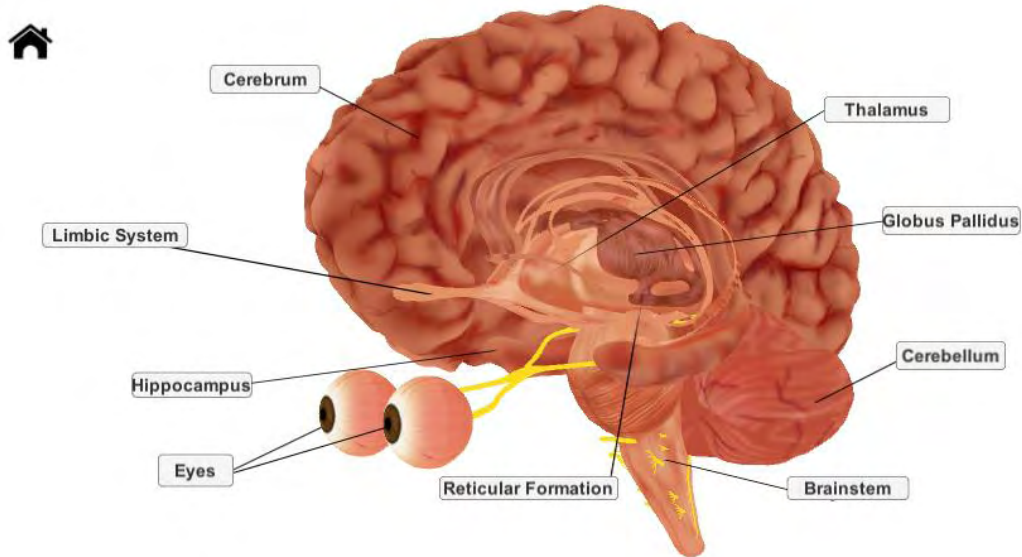
- **Ινιακός λοβός**

Εντοπίζεται στο πίσω τμήμα του εγκεφάλου και είναι υπεύθυνος για την όραση. Το αριστερό ημισφαίριο βλέπει το δεξί ήμισυ του οπτικού πεδίου και το δεξί ημισφαίριο στο αριστερό ήμισυ. Βλάβη στους ινιακού λοβούς μπορούν να προκαλέσουν, εκτός από προβλήματα στην όραση, δημιουργία παραισθήσεων και αδυναμία στην αναγνώριση κινήσεων και ζωγραφισμένων αντικειμένων. [38]

- **Κροταφικός λοβός**

Βρίσκεται στον πλάγιο τμήμα κάθε ημισφαιρίου, κοντά στους κροτάφους. Είναι υπεύθυνος για την ακοή, την αντίληψη σύνθετων εικόνων, την κατανόηση της ομιλίας (στο αριστερό ημισφαίριο) καθώς και για το συναίσθημα. Μετά από βλάβες σε περιοχές του κροταφικού λοβού παρουσιάζονται δυσλειτουργίες που έχουν σχέση με την αναγνώριση προσώπων, επιθετική συμπεριφορά, επίμονη ομιλία και συναισθήματα πανικού. [39]

Στο τρισδιάστατο μοντέλο της πτυχιακής εργασίας εντοπίζονται και αναλύονται συνοπτικά και ανάλογα με τις λειτουργίες τους τα παρακάτω μέρη του εγκεφάλου.



Εικόνα Β: Τρισδιάστατο μοντέλο εγκεφάλου της διεπαφής μας

### 1) Cerebrum (Εγκεφαλικό στέλεχος)

Είναι το μεγαλύτερο μέρος του εγκεφάλου και αποτελείται από το δεξί και αριστερό ημισφαίριο. Εκτελεί τις πιο σημαντικές λειτουργίες όπως για παράδειγμα την ερμηνεία της αφής, της όρασης και της ακοής. Ακόμα ελέγχει τον λόγο, τα συναισθήματα, την εκμάθηση και τον έλεγχο της κίνησης.

### 2) Thalamus (Θάλαμος)

Ο θάλαμος λειτουργεί ως σταθμός αναμετάδοσης σχεδόν για όλες τις πληροφορίες που έρχονται και πηγαίνουν στον εγκεφαλικό φλοιό. Παίζει ρόλο στην αίσθηση του πόνου, την προσοχή, την εγρήγορση και τη μνήμη. [40] [41]

### 3) Globus Pallidus (Ωχρά σφαίρα)

Είναι ένας από τους πυρήνες εξόδου των βασικών γαγγλίων. Τα βασικά γάγγλια είναι τεράστιες συστάδες νευρώνων που ευθύνονται για τις ακούσιες κινήσεις. Η ωχρά σφαίρα αποτελείται από νευρώνες που περιέχουν γ-αμινοβουτυρικό οξύ γνωστό ως GABA. Το GABA είναι ένας νευροδιαβιβαστής ο οποίος μεταδίδει σήματα από τον ένα νευρώνα στον άλλο. [42]

#### **4) Cerebellum (Παρεγκεφαλίδα)**

Η παρεγκεφαλίδα βρίσκεται πίσω από το πάνω μέρος του εγκεφαλικού στελέχους, όπου ο νωτιαίος μυελός συναντά τον εγκέφαλο και αποτελείται από δύο ημισφαίρια. Είναι υπεύθυνη για τη ρύθμιση του κινητικού συστήματος. Συντονίζει τις εθελοντικές κινήσεις όπως είναι η στάση του σώματος, η ισορροπία και ο συντονισμός με αποτέλεσμα την ομαλή μυϊκή δραστηριότητα. Μπορεί να αποτελεί περίπου το δέκα τοις εκατό του συνολικού βάρους του εγκεφάλου, αλλά περιέχει σχεδόν το ήμισυ των νευρώνων του εγκεφάλου. Βλάβες στην παρεγκεφαλίδα διαταράσσουν την ισορροπία, την ακρίβεια των κινήσεων στο χώρο, το συντονισμό των κινήσεων και των οφθαλμών. [43] [44]

#### **5) Limbic System (Μεταχιακό σύστημα)**

Είναι ένα σύνθετο σύνολο δομών που βρίσκεται και στις δύο πλευρές του θαλάμου, ακριβώς κάτω από το εγκεφαλικό σύστημα. Περιλαμβάνει τον υποθάλαμο, τον ιππόκαμπο, την αμυγδαλή και άλλες κοντινές περιοχές. Φαίνεται να είναι κυρίως υπεύθυνο για τη συναισθηματική ζωή του κάθε ανθρώπου και έχει συσχετιστεί με τη δημιουργία των αναμνήσεων. [45]

#### **6) Hippocampus (Ιππόκαμπος)**

Ο ιππόκαμπος συμμετέχει στην αποθήκευση της μακροχρόνια μνήμης. Φαίνεται να διαδραματίζει σημαντικό ρόλο στη δηλωτική μνήμη, η οποία είναι ένας τύπος μνήμης που περιλαμβάνει πράγματα τα οποία μπορούν να ανακληθούν σκόπιμα από το άτομο όπως γεγονότα που συνέβησαν στη ζωή του. Ο ιππόκαμπος δεν ασχολείται με βραχυπρόθεσμους τύπους μνήμης. Όσοι έχουν χάσει κάποια λειτουργία ή έχουν αφαιρέσει σημαντικά τμήματα του μεταχιακού συστήματος, αλλά έχουν ακόμα τον ιππόκαμπο, έχουν μόνο μακροχρόνια μνήμη και δε μπορούν να καταγράψουν νέες μήμες ή λειτουργίες. [46]

#### **7) Reticular formation (Δικτυωτός σχηματισμός)**

Αποτελείται από ένα δίκτυο διάχυτων συσσωμάτων νευρώνων που κατανέμονται σε όλα τα κεντρικά τμήματα του μυελού και του μεσεγκεφάλου. Οι νευρώνες λαμβάνουν πληροφορίες από το νωτιαίο μυελό, τους πυρήνες του κρανιακού νεύρου, την παρεγκεφαλίδα και το εγκεφαλικό στέλεχος και στέλνουν μηνύματα στις ίδιες δομές. Οι κύριες λειτουργίες είναι:

- Ενεργοποίηση του εγκεφάλου για συμπεριφορική διέγερση και διαφορετικά επίπεδα συνειδητοποίησης
- Διαμόρφωση τμηματικών αντανακλαστικών και μυϊκού τόνου για τον έλεγχο της κινητικής λειτουργίας
- Συντονισμός των αυτόνομων λειτουργιών, όπως ο έλεγχος της αναπνοής και οι καρδιαγγειακές δραστηριότητες
- Διαμόρφωση της αντίληψης του πόνου [47]

### **8) Brain Stem (Στέλεχος εγκεφάλου)**

Ανατομικά συνδέει τον εγκέφαλο με το νωτιαίο μυελό. Οι αισθητήριοι νευρώνες ταξιδεύουν μέσα από στο στέλεχος του εγκεφάλου επιτρέποντας τη μετάδοση σημάτων μεταξύ του εγκεφάλου και του νωτιαίου μυελού. Τα περισσότερα κраниακά νεύρα βρίσκονται στο στέλεχος του εγκεφάλου. Είναι υπεύθυνο για διάφορες σημαντικές λειτουργίες του σώματος, οι οποίες περιλαμβάνουν την αναπνοή, την πέψη, τον παλμό της καρδιάς, τον έλεγχο της πίεσης του αίματος και άλλες αυτόνομες λειτουργίες. Τραυματισμός στο εγκεφαλικό στέλεχος μπορεί να προκληθεί από κάποιο τραύμα ή εγκεφαλικό επεισόδιο και μπορεί να οδηγήσει σε δυσκολίες κινητικότητας και συντονισμού της κίνησης. [48]

### **9) Eyes (Μάτια)**

Τα μάτια λειτουργούνε όπως μια κάμερα. Οι ακτίνες φωτός που λαμβάνονται στην επιφάνεια του ματιού, κατευθύνονται στο πίσω μέρος του ματιού, στον αμφιβληστροειδή, ο οποίος λειτουργεί όπως το φιλμ της κάμερας. Τα κύτταρα του αμφιβληστροειδή απορροφούν και μετατρέπουν το φως σε ηλεκτροχημικούς παλμούς όπου μεταφέρονται κατά μήκος του οπτικού νεύρου στον εγκέφαλο. Ο εγκέφαλος στη συνέχεια μεταφράζει την εικόνα ώστε να μπορέσουμε να καταλάβουμε τι βλέπουμε. [49]

## ΠΑΡΑΡΤΗΜΑ Β

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;
using System.IO;

public class CSVParsing1 : MonoBehaviour
{
    public TextAsset csvFile; // Reference of CSV file
    public Slider mainSlider;
    public Button Button0;
    public Button Button1;
    public Button Button2;
    public Button Button3;
    public Button Button4;
    public Button del;

    private char lineSeperator = '\n'; // It defines line separate character
    private char fieldSeparator = ','; // It defines field separate chracter

    void Start()
    {
        Button obj = Button2.GetComponent<Button>();
        obj.onClick.AddListener(TaskOnClick);
        readData();
    }

    // Read data from CSV file
    private void readData()
    {
        mainSlider.minValue = 1;
        mainSlider.maxValue = 0;
    }
}
```

```
        string[] records = File.ReadAllLines(getPath() +
"/Patients/PatientsData.csv");

        mainSlider.maxValue = records.Length - 1;

        if (mainSlider.maxValue < mainSlider.minValue)
        {
            mainSlider.minValue = 0;
            mainSlider.maxValue = 0;
            mainSlider.value = 0;
        }

        if (mainSlider.value > 0)
        {
            readButtons();
        }
    }

    // Read buttons from CSV file
    private void readButtons()
    {
        GameObject.Find("btn").GetComponentInChildren<Text>().text = "";
        GameObject.Find("gob").GetComponentInChildren<Text>().text = "";
        GameObject.Find("obj").GetComponentInChildren<Text>().text = "";
        GameObject.Find("but").GetComponentInChildren<Text>().text = "";
        GameObject.Find("asd").GetComponentInChildren<Text>().text = "";

        string[] records = File.ReadAllLines(getPath() +
"/Patients/PatientsData.csv");

        if (records.Length > 1)
        {
            //string[] records = csvFile.text.Split(lineSeperator);

```

```

        if ((int)mainSlider.value >= 3)
        {
            string[] gobrec = records[(int)mainSlider.value -
2].Split(fieldSeperator);
            GameObject.Find("gob").GetComponentInChildren<Text>().text =
gobrec[2] + ", " + gobrec[1];
        }
        if ((int)mainSlider.value >= 2)
        {
            string[] btnrec = records[(int)mainSlider.value -
1].Split(fieldSeperator);
            GameObject.Find("btn").GetComponentInChildren<Text>().text =
btnrec[2] + ", " + btnrec[1];
        }
        if ((int)mainSlider.value >= 1)
        {
            string[] objrec =
records[(int)mainSlider.value].Split(fieldSeperator);
            GameObject.Find("obj").GetComponentInChildren<Text>().text =
objrec[2] + ", " + objrec[1];
            Button2.name = objrec[2] + objrec[3];
        }
        if (((int)mainSlider.maxValue - (int)mainSlider.value) > 0)
        {
            string[] butrec = records[(int)mainSlider.value +
1].Split(fieldSeperator);
            GameObject.Find("but").GetComponentInChildren<Text>().text =
butrec[2] + ", " + butrec[1];
        }
        if (((int)mainSlider.maxValue - (int)mainSlider.value) > 1)
        {
            string[] asdrec = records[(int)mainSlider.value +
2].Split(fieldSeperator);
            GameObject.Find("asd").GetComponentInChildren<Text>().text =
asdrec[2] + ", " + asdrec[1];
        }
    }

```

```
    }

}

public void DeleteRec()
{

    string[] lines = File.ReadAllLines(getPath() +
"/Patients/PatientsData.csv");

    string lineSeparator = ((char)0x2028).ToString();
    string paragraphSeparator = ((char)0x2029).ToString();

    File.Delete(getPath() + "/Patients/PatientsData.csv");

    using (var stream = File.OpenWrite(getPath() +
"/Patients/PatientsData.csv"))
    using (StreamWriter writer = new StreamWriter(stream))
    {
        if (lines.Length > 0)
        {
            if (lines.Length < 3)
            {
                writer.Write(lines[0]);
            }
            if (lines.Length > 2)
            {
                if ((int)mainSlider.value < lines.Length - 1)
                {
                    for (int i = 0; i < lines.Length - 1; i++)
                    {
                        if (i != mainSlider.value)
                        { writer.WriteLine(lines[i]); }
                    }
                    writer.Write(lines[lines.Length - 1]);
                }
            }
        }
    }
}
```



```
    }

    if ((int)mainSlider.value == lines.Length - 1)
    {
        for (int i = 0; i < lines.Length - 2; i++)
        {
            writer.WriteLine(lines[i]);
        }
        writer.Write(lines[lines.Length - 2]);
    }

}

}

readData();
readButtons();
}

void TaskOnClick()
{
    Debug.Log("You have clicked the button!");
    Debug.Log(Button2.name);

    // Create the TMP Folder
    if (Directory.Exists(getPath() + "/Patients/TMP"))
    {
        string[] files = Directory.GetFiles(getPath() + "/Patients/TMP");
        string[] dirs = Directory.GetDirectories(getPath() +
"/Patients/TMP");

        foreach (string file in files)
        {
```

```
        File.SetAttributes(file, FileAttributes.Normal);
        File.Delete(file);
    }

    //foreach (string dir in dirs)
    //{
    //    Directory.Delete(getPath() + "/Patients/TMP/"+dir);
    //}
    Directory.Delete(getPath() + "/Patients/TMP");

    //        Directory.Delete(getPath() + "/Patients/TMP");
}
Directory.CreateDirectory(getPath() + "/Patients/TMP");

// Create the owner folder text file

DirectoryInfo dir = new DirectoryInfo(getPath() + "/Patients/" +
Button2.name);
DirectoryInfo[] dirs2 = dir.GetDirectories();

FileInfo[] files2 = dir.GetFiles();
foreach (FileInfo file in files2)
{
    string temppath = Path.Combine(getPath() + "/Patients/TMP",
file.Name);
    file.CopyTo(temppath, false);
}

if (!File.Exists(getPath() + "/Patients/TMP/name.txt"))
{
    File.Create(getPath() + "/Patients/TMP/name.txt").Dispose();
    using (TextWriter tw = new StreamWriter(getPath() +
"/Patients/TMP/name.txt"))
    {
        tw.WriteLine(Button2.name);
        tw.Close();
    }
}
```

```
    }

    // Copy files from the Owner Folder to the TMP
    //foreach (string fileName in Directory.GetFiles(getPath() +
"/Patients/" + Button2.name))
    //{
    //    File.Copy(
    //        Path.Combine(getPath() + "/Patients/" + Button2.name,
fileName),
    //        Path.Combine(getPath() + "/Patients/TMP/", fileName), true);
    //}

}

public void SubmitSliderSetting()
{
    if (mainSlider.value > 0)
    {
        readButtons();
    }
}

// Get path for given CSV file
private static string getPath()
{
#if UNITY_EDITOR
    return Application.dataPath;
#elif UNITY_ANDROID
    return Application.persistentDataPath; // +fileName;
#elif UNITY_IPHONE
    return GetiPhoneDocumentsPath(); // +"/"+fileName;
#else
    return Application.dataPath; // +"/"+ fileName;
#endif
}
```

#endif

}

}

## Βιβλιογραφία

- [1] [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Interface\\_\(computing\)](https://en.wikipedia.org/wiki/Interface_(computing)). [Πρόσβαση 30 11 2017].
- [2] TedCas, «TedCas,» [Ηλεκτρονικό]. Available: <http://www.tedcas.com/en>. [Πρόσβαση 8 11 2017].
- [3] «Leap Motion Gallery,» [Ηλεκτρονικό]. Available: <https://gallery.leapmotion.com/cyber-science-motion/>. [Πρόσβαση 8 11 2017].
- [4] «PIE,» [Ηλεκτρονικό]. Available: <http://pie.med.utoronto.ca/TVASurg/project/vr-surgical-anatomy-module/>. [Πρόσβαση 8 11 2017].
- [5] «Dictionary.com,» [Ηλεκτρονικό]. Available: <http://www.dictionary.com/e/the-kinect/>. [Πρόσβαση 11 11 2017].
- [6] J. W. a. J. Ashley, Beginning Kinect Programming with the, New York: Apress, 2012.
- [7] «Wikipedia,» [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/Kinect#Launch>. [Πρόσβαση 11 11 2017].
- [8] [Ηλεκτρονικό]. Available: <https://msdn.microsoft.com/en-us/library/jj131033.aspx>. [Πρόσβαση 12 11 2017].
- [9] N. B. F. E. D. H. C. M. P. Jeff Kramer, Hacking the Kinect, New York: Apress, 2012.
- [10] [Ηλεκτρονικό]. Available: <http://gmv.cast.uark.edu/scanning/hardware/microsoft-kinect-resourceshardware/>. [Πρόσβαση 12 11 2017].
- [11] [Ηλεκτρονικό]. Available: <https://fabrizio89.wordpress.com/kinect-1-generation/>. [Πρόσβαση 12 11 2017].
- [12] [Ηλεκτρονικό]. Available: <https://msdn.microsoft.com/en-us/library/jj663790.aspx>. [Πρόσβαση 12 11 2017].
- [13] [Ηλεκτρονικό]. Available: <https://msdn.microsoft.com/en-us/library/jj131025.aspx>. [Πρόσβαση 13 11 2017].
- [14] A. A. a. P. D. Georgios Th. Papadopoulos, «CERTH,» [Ηλεκτρονικό]. Available: <https://www.iti.gr/iti/files/document/publications/mmm-preprint.pdf>. [Πρόσβαση 13 11 2017].
- [15] [Ηλεκτρονικό]. Available: <https://developer.microsoft.com/en-us/windows/kinect/hardware-setup>. [Πρόσβαση 13 11 2017].

- [16] [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Leap\\_Motion](https://en.wikipedia.org/wiki/Leap_Motion). [Πρόσβαση 28 11 2017].
- [17] [Ηλεκτρονικό]. Available: <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>. [Πρόσβαση 28 11 2017].
- [18] B. Sanders, Mastering Leap Motion, PACKT.
- [19] [Ηλεκτρονικό]. Available: <https://developer.leapmotion.com/documentation/index.html?proglang=current>. [Πρόσβαση 28 11 2017].
- [20] «Introducing the Skeletal Tracking Model,» [Ηλεκτρονικό]. Available: [https://developer.leapmotion.com/documentation/csharp/devguide/Intro\\_Skeleton\\_API.html](https://developer.leapmotion.com/documentation/csharp/devguide/Intro_Skeleton_API.html). [Πρόσβαση 29 11 2017].
- [21] «API Overview,» [Ηλεκτρονικό]. Available: [https://developer.leapmotion.com/documentation/csharp/devguide/Leap\\_Overview.html](https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Overview.html). [Πρόσβαση 29 11 2017].
- [22] «Leap motion gestures,» [Ηλεκτρονικό]. Available: [https://developer.leapmotion.com/documentation/csharp/devguide/Leap\\_Gestures.html](https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Gestures.html). [Πρόσβαση 29 11 2017].
- [23] L. Shao, «Hand movement and gesture recognition using Leap Motion Controller,» [Ηλεκτρονικό]. Available: [https://stanford.edu/class/ee267/Spring2016/report\\_lin.pdf](https://stanford.edu/class/ee267/Spring2016/report_lin.pdf). [Πρόσβαση 29 11 2017].
- [24] [Ηλεκτρονικό]. Available: <http://store-eur.leapmotion.com/products/leap-motion-controller>. [Πρόσβαση 29 11 2017].
- [25] S. L. Pfleeger, Software Engineering: Theory and Practice, Pearson Education Inc., 2001.
- [26] M. Νικολαΐδου, «Χαροκόπειο Πανεπιστήμιο - Τμήμα πληροφορικής και τηλεματικής,» [Ηλεκτρονικό]. Available: [https://eclass.hua.gr/modules/document/file.php/DIT187/%CE%A0%CE%91%CE%A1%CE%9F%CE%A5%CE%A3%CE%99%CE%91%CE%A3%CE%95%CE%99%CE%A3/se02\\_requirements.pdf](https://eclass.hua.gr/modules/document/file.php/DIT187/%CE%A0%CE%91%CE%A1%CE%9F%CE%A5%CE%A3%CE%99%CE%91%CE%A3%CE%95%CE%99%CE%A3/se02_requirements.pdf). [Πρόσβαση 5 01 2018].
- [27] Β. Γερογιάννης, Γ. Κακαρόντζας, Α. Καμέας, Γ. Σταμέλος και Π. Φιτσιλής, Αντικειμενοστραφής ανάπτυξη λογισμικού με τη UML, Αθήνα: Κλειδάριθμος, 2006.
- [28] [Ηλεκτρονικό]. Available: <https://unity3d.com/unity>. [Πρόσβαση 12 1 2018].
- [29] «Wikipedia,» [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)). [Πρόσβαση 12 1 2018].
- [30] [Ηλεκτρονικό]. Available: <https://www.turbosquid.com/3d-models/max-brain-medical-nerves/815426>. [Πρόσβαση 14 1 2018].

- [31] [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Manual/class-GameObject.html>. [Πρόσβαση 14 1 2018].
- [32] [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/ScriptReference/GameObject.html>. [Πρόσβαση 14 1 2018].
- [33] [Ηλεκτρονικό]. Available: <https://developer.unity.com/releases/leap-motion-orion-313>. [Πρόσβαση 14 1 2018].
- [34] «Wikipedia,» [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Human\\_brain](https://en.wikipedia.org/wiki/Human_brain). [Πρόσβαση 20 12 2017].
- [35] «Wikipedia,» [Ηλεκτρονικό]. Available: [https://el.wikipedia.org/wiki/Μετωπιαίος\\_λοβός](https://el.wikipedia.org/wiki/Μετωπιαίος_λοβός). [Πρόσβαση 20 12 2017].
- [36] [Ηλεκτρονικό]. Available: <https://www.spinalcord.com/frontal-lobe>. [Πρόσβαση 20 12 2017].
- [37] [Ηλεκτρονικό]. Available: [https://el.wikipedia.org/wiki/%CE%92%CF%81%CE%B5%CE%B3%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CF%8C%CF%82\\_%CE%BB%CE%BF%CE%B2%CF%8C%CF%82](https://el.wikipedia.org/wiki/%CE%92%CF%81%CE%B5%CE%B3%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CF%8C%CF%82_%CE%BB%CE%BF%CE%B2%CF%8C%CF%82). [Πρόσβαση 20 12 2017].
- [38] «Wikipedia,» [Ηλεκτρονικό]. Available: [https://el.wikipedia.org/wiki/%CE%99%CE%BD%CE%B9%CE%B1%CE%BA%CF%8C%CF%82\\_%CE%BB%CE%BF%CE%B2%CF%8C%CF%82](https://el.wikipedia.org/wiki/%CE%99%CE%BD%CE%B9%CE%B1%CE%BA%CF%8C%CF%82_%CE%BB%CE%BF%CE%B2%CF%8C%CF%82). [Πρόσβαση 20 12 2017].
- [39] «Wikipedia,» [Ηλεκτρονικό]. Available: [https://el.wikipedia.org/wiki/%CE%9A%CF%81%CE%BF%CF%84%CE%B1%CF%86%CE%B9%CE%BA%CF%8C%CF%82\\_%CE%BB%CE%BF%CE%B2%CF%8C%CF%82](https://el.wikipedia.org/wiki/%CE%9A%CF%81%CE%BF%CF%84%CE%B1%CF%86%CE%B9%CE%BA%CF%8C%CF%82_%CE%BB%CE%BF%CE%B2%CF%8C%CF%82). [Πρόσβαση 20 12 2017].
- [40] T. Hines, «Mayfield,» 4 2016. [Ηλεκτρονικό]. Available: <https://www.mayfieldclinic.com/PE-AnatBrain.htm>. [Πρόσβαση 21 12 2017].
- [41] H. R. K. T. 1. E. Eric P. Widmaier, Vander's Human Physiology: The Mechanisms of Body Function, New York: McGraw-Hill, 2014.
- [42] «Healthline,» [Ηλεκτρονικό]. Available: <https://www.healthline.com/human-body-maps/medial-globus-pallidus>. [Πρόσβαση 21 12 2017].
- [43] «Healthline,» [Ηλεκτρονικό]. Available: <https://www.healthline.com/human-body-maps/cerebellum>. [Πρόσβαση 26 12 2017].
- [44] «ΠΜΣ 'Εγκέφαλος και Νους',» [Ηλεκτρονικό]. Available: <http://brain-mind.med.uoc.gr/sites/default/files/Cerebellum2012for%20students.pdf>. [Πρόσβαση 26 12 2017].

[45] D. C. G. Boeree. [Ηλεκτρονικό]. Available: <https://webpace.ship.edu/cgboer/limbicsystem.html>. [Πρόσβαση 26 12 2017].

[46] «Healthline,» [Ηλεκτρονικό]. Available: <https://www.healthline.com/human-body-maps/hippocampus>. [Πρόσβαση 26 12 2017].

[47] M. Loren A. Rolak, «Chapter 9 - Brain Stem Disease,» σε Neurology Secrets (Fifth Edition), MOSBY ELSEVIER, 2011, p. 447.

[48] R. Bailey, 24 08 2017. [Ηλεκτρονικό]. Available: <https://www.thoughtco.com/brainstem-anatomy-373212>. [Πρόσβαση 27 12 2017].

[49] S. DeRemer, 3 03 2016. [Ηλεκτρονικό]. Available: <http://discoveryeye.org/the-brain-and-the-eye/>. [Πρόσβαση 27 12 2017].

## Πηγές Εικόνων

Εικόνα 1: Χρήση συστήματος TedCas [1]

Εικόνα 2: Αλληλεπίδραση με το Cyber Science – Motion:

[http://www.prweb.com/releases/science\\_software/leap\\_motion/prweb10961515.htm](http://www.prweb.com/releases/science_software/leap_motion/prweb10961515.htm)

Εικόνα 3: Παράδειγμα εφαρμογής:

[https://www.youtube.com/watch?v=1S3\\_ZLqFu80](https://www.youtube.com/watch?v=1S3_ZLqFu80)

Εικόνα 4: Microsoft Kinect Sensor:

<https://msdn.microsoft.com/en-us/library/hh855355.aspx>

Εικόνα 5: Εσωτερικό αισθητήρα Kinect:

<https://msdn.microsoft.com/en-us/library/jj131033.aspx>

Εικόνα 6: Έγχρωμη κάμερα:

<http://gmv.cast.uark.edu/scanning/hardware/microsoft-kinect-resourceshardware/>

Εικόνες 7: Παράδειγμα λειτουργίας αισθητήρα βάθους:

<https://www.youtube.com/watch?v=uq9SEJxZiUg>

Εικόνα 8: Σειρά μικροφώνων:

<http://gmv.cast.uark.edu/scanning/hardware/microsoft-kinect-resourceshardware/>

Εικόνα 9: Μορφή επιταχυνσιόμετρου:

<https://www.ifixit.com/Teardown/Xbox+360+Kinect+Teardown/4066>

Εικόνα 10: Μηχανισμός κλίσης και περιστροφής:

<https://social.msdn.microsoft.com/Forums/en-US/954197e5-a0df-44d3-84f4-9bf3885deead/how-to-rotate-kinect-sensor?forum=kinectsdk>

Εικόνα 11: Αναγνώριση είκοσι σκελετικών αρθρώσεων σε όρθια στάση [5]



Εικόνα 12: Αισθητήρας Leap Motion:

[https://cnet1.cbsistatic.com/img/PIV\\_du5LCv5Yi-R3jlZcvJE9Rxw=/770x433/2013/07/18/747120a6-67c2-11e3-a665-14feb5ca9861/LeapMotionController\\_35823002\\_09.jpg](https://cnet1.cbsistatic.com/img/PIV_du5LCv5Yi-R3jlZcvJE9Rxw=/770x433/2013/07/18/747120a6-67c2-11e3-a665-14feb5ca9861/LeapMotionController_35823002_09.jpg)

Εικόνα 13: Χώρος αλληλεπίδρασης Leap Motion:

[https://cms-assets.tutsplus.com/uploads/users/289/posts/20455/image/Leap\\_horizontalViewAngle.png](https://cms-assets.tutsplus.com/uploads/users/289/posts/20455/image/Leap_horizontalViewAngle.png)

Εικόνα 14: Παράδειγμα χειρονομίας μοντέλου χεριού:

[https://developer.leapmotion.com/documentation/csharp/devguide/Intro\\_Skeleton\\_API.html](https://developer.leapmotion.com/documentation/csharp/devguide/Intro_Skeleton_API.html)

Εικόνα 15: Καρτεσιανό σύστημα συντεταγμένων:

[https://developer.leapmotion.com/documentation/csharp/devguide/Leap\\_Overview.html](https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Overview.html)

Εικόνα 16: Μοντέλο χεριού:

[https://developer.leapmotion.com/documentation/csharp/devguide/Leap\\_Overview.html](https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Overview.html)

Εικόνα 17: Μοντέλο δαχτύλων:

[https://developer.leapmotion.com/documentation/csharp/devguide/Leap\\_Overview.html](https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Overview.html)

Εικόνα 18: Σκελετικό σύστημα:

[https://developer.leapmotion.com/documentation/csharp/devguide/Intro\\_Skeleton\\_API.html](https://developer.leapmotion.com/documentation/csharp/devguide/Intro_Skeleton_API.html)

Εικόνα 19: Circle gesture:

[https://developer.leapmotion.com/documentation/csharp/devguide/Leap\\_Gestures.html](https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Gestures.html)

Εικόνα 20: Swipe gesture:

[https://developer.leapmotion.com/documentation/csharp/devguide/Leap\\_Gestures.html](https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Gestures.html)

Εικόνα 21: Key tap gesture:

[https://developer.leapmotion.com/documentation/csharp/devguide/Leap\\_Gestures.html](https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Gestures.html)

Εικόνα 22 Screen tap gesture:

[https://developer.leapmotion.com/documentation/csharp/devguide/Leap\\_Gestures.html](https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Gestures.html)  
1

Εικόνα 29: Σύνολο συσκευών και λογισμικών:

<https://unity3d.com/unity>

Εικόνα 30: Πίνακας χαρακτηριστικών:

<https://www.turbosquid.com/3d-models/max-brain-medical-nerves/815426>

Εικόνα 67: [https://i.ytimg.com/vi/vukBCXLG\\_u0/maxresdefault.jpg](https://i.ytimg.com/vi/vukBCXLG_u0/maxresdefault.jpg)

Εικόνα 68:

<http://macandmecom.weebly.com/uploads/1/4/9/2/14921094/44048.jpg?880>

Εικόνα Α: : [https://el.wikipedia.org/wiki/Μετωπιαίος\\_λοβός](https://el.wikipedia.org/wiki/Μετωπιαίος_λοβός)

Εικόνα Β: προσωπικά δικαιώματα

Όλες οι υπόλοιπες εικόνες αποτελούν προσωπικά δικαιώματα.



