ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΣΧΕΔΙΟ ΠΑΡΟΥΣΙΑΣΗΣ
ΑΤΟΜΙΚΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Μάιος 2015

Ατομική Διπλωματική Εργασία

Επαυξημένη πραγματικότητα με την χρήση Oculus Rift και Depth Sensor

Κωνσταντίνος Χαραλάμπους

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2015

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Επαυξημένη πραγματικότητα με την χρήση Oculus Rift και Depth Sensor

Κωνσταντίνος Χαραλάμπους

Επιβλέπων Καθηγητής
κ. Γιώργος Χρυσάνθου

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

Μάιος 2015

Ευχαριστίες

Στην παρούσα φάση θα ήθελα να ευχαριστήσω τον υπεύθυνο καθηγητή μου κ. Γιώργο Χρυσάνθου, που μου έδωσε την ευκαιρία να ασχοληθώ με το καινοτόμο αυτό θέμα, επεκτείνοντας την γνώση μου σε θέματα εικονικής και επαυξημένης πραγματικότητας, όπως και για την πολύτιμη βοήθεια του κατά τη διάρκεια της διπλωματικής εργασίας.

Ευχαριστίες θα ήθελα να δώσω και στον κ. Μάριο Κυριάκου μεταδιδακτορικό ερευνητή, για την παροχή βοήθειας κατά την υλοποίηση του συστήματος, όπως και στον θείο μου Μιχάλη Μιχαήλ για την στήριξη που μου παρείχε κατά τη διάρκεια της έρευνας και υλοποίησης της παρούσας διπλωματικής εργασίας.

Εξίσου σημαντική ήταν και η συμβολή του τμήματος Πληροφορικής του Πανεπιστημίο Κύπρου, για την άμεση παροχή των συσκευών που χρησιμοποιήθηκαν κατά την έρευνα, ενώ ευχαριστίες θα ήθελα να δώσω και στην φοιτήτρια Μηχανολογίας του Πανεπιστημίου Κύπρου Εβίτα Κουτέλη, για την βοήθεια της στη δημιουργία μιας βάσης για την στήριξη των καμερών στο Oculus.

Περίληψη

Η παρούσα διπλωματική εργασία χρησιμοποιεί τις σύγχρονες τεχνολογίες προκειμένου να προτείνει εναλλακτικές χρήσεις σε μερικές από τις προκλήσεις της εικονικής πραγματικότητας και κατ' επέκταση της επαυξημένης πραγματικότητας. (virtual reality). Η βασικότερη πρόκληση της εικονικής πραγματικότητας έγκειται στο γεγονός της εμβύθισης του χρήστη μέσω αλληλεπίδρασης του με τον κόσμο της εικονικής πραγματικότητας. Συνέπεια του γεγονότος αυτού ο χρήστης με την συνεχή χρήση της εικονικής πραγματικότητας, αποστασιοποιείται από τον πραγματικό κόσμο και αλληλεπιδρά με την εικονική πραγματικότητα.

Επομένως, η εναλλακτική επιλογή που προτείνεται στην έρευνα αυτή στηρίζεται στην χρήση των ανεπτυγμένων τεχνολογιών Oculus Rift και Kinect που επικεντρώνονται σε θέματα εικονικής πραγματικότητας. Για τον σκοπό αυτό αναπτύχθηκε μια εφαρμογή στην οποία οι τεχνολογίες Oculus Rift και Kinect θα επικοινωνούν μεταξύ τους προκειμένου να παρέχουν μοναδικά χαρακτηριστικά και νέες δυνατότητες στον χρήστη. Η συσκευή ανίχνευσης κίνησης – Kinect (Microsoft) χρησιμοποιήθηκε για την ανίχνευση του βάθους στον χώρο αλληλεπίδρασης χρησιμοποιώντας τον αισθητήρα που βρίσκεται στην συσκευή. Τα δεδομένα βάθους χρησιμοποιήθηκαν ως αναφορά για την τοποθέτηση εικονικών αντικειμένων, με την βοήθεια διαφόρων τεχνικών και αλγορίθμων. Επίσης ολόκληρη η σκηνή μετά από γραφική επεξεργασία (render) μπορεί να προβληθεί στην συσκευή προβολής εικονικής πραγματικότητας Oculus Rift, ώστε να δώσει στον χρήστη μια νέα εμπειρία σε πραγματικό χρόνο.

Στην παρούσα διπλωματική εργασία θα επεξηγηθεί αναλυτικά το υλικό και το λογισμικό που χρησιμοποιήθηκε κατά την έρευνα, την μεθοδολογία αλλά και τους αλγορίθμους που χρησιμοποιήθηκαν, τα αποτελέσματα της έρευνας συμπεριλαμβανομένων θετικών και αρνητικών στοιχείων, προβλημάτων αλλά και αδυναμιών που εντοπίστηκαν τόσο κατά την δοκιμή της εφαρμογής όσο και κατά την υλοποίησή της. Τέλος, θα αναφερθούν νέες τεχνικές και πιθανές μελλοντικές βελτιώσεις, οι οποίες θα εμβαθύνουν περισσότερο την έρευνα με απώτερο σκοπό η αλληλεπίδραση του χρήστη να γίνει αποτελεσματικότερη και πιο ρεαλιστική με την πραγματικότητα.

Περιεχόμενα

Κεφάλαιο 1 ... 1
Εισαγωγή ... 1
1.1 Κίνητρο ... 1
1.2 Σκοποί και Στόχοι ... 2
1.3 Δομή της Διπλωματικής ... 4
Κεφάλαιο 2 ... 7
Προηγούμενη Εργασία ... 7
2.1 Εισαγωγή ... 7
2.2 Ανακατασκευή χώρου με κινούμενη κάμερα σε πραγματικό χρόνο ... 8
2.3 Ανακατασκευή χώρου με επαύξηση εικονικών αντικειμένων ... 10
2.4 Επιτραπέζια επαυξημένη πραγματικότητα ... 12
2.5 Επαυξημένη πραγματικότητα μέσω Oculus Rift (AR-Rift) ... 14
2.6 Συμπεράσματα ... 17
Κεφάλαιο 3 ... 19
Επισκόπηση ... 19
3.1 Επισκόπηση του έργου ... 19
3.2 Συσκευές προβολής εικονικής πραγματικότητας ... 21
3.2.1 Oculus Rift ... 21
3.2.2 Εναλλακτικές Συσκευές ... 24
3.3 Συσκευές ανίχνευσης βάθους ... 28
3.3.1 Microsoft Kinect ... 28
3.3.2 Εναλλακτικές συσκευές ανίχνευσης βάθους ... 30
3.4 Κάμερες ... 31
3.4.1 Logitech Cameras C310 ... 31
3.4.2 Εναλλακτικές κάμερες ... 32
3.5 Επιπλέον εργαλεία που χρησιμοποιήθηκαν ... 33
3.5.1 Unity 3D ... 33
3.5.2 Custom Camera Rig ... 35
Κεφάλαιο 4 ... 36
Υλοποίηση ... 36
4.1 Διασύνδεση ... 36
4.2 Αντικείμενα που αποτελούν το σύστημα ... 38
4.3 Ανταλλαγή και επεξεργασία δεδομένων ... 39
4.4 Ανάλυση μεθοδολογίας και αλγόριθμοι ... 41

4.4.1 Ανίχνευση βάθους και δημιουργία πλέγματος ως αναφορά 41
4.4.2 Μεταφορά ζωντανής εικόνας από τις κάμερες στο Oculus Rift 43
4.4.3 Raycast και τοποθέτηση εικονικών αντικειμένων 43
4.4.4 Επιπρόσθετες διεργασίες για σωστή λειτουργία 45
Κεφάλαιο 5 47
Συμπεράσματα 47
5.1 Οθόνες αποτελεσμάτων 47
5.2 Γενικά συμπεράσματα 50
5.3 Προβλήματα που προέκυψαν και περιορισμοί 51
5.4 Μελλοντική εργασία και επέκταση συστήματος 52
Βιβλιογραφία 55
Παράρτημα Α 58
1. Ενδεικτικός Κώδικας (Unity3D – C#) 58
1.1 Raycast.cs 58
1.2 ObjectSpawner.cs 61
1.3 SpawnedObject.cs 62
1.4 WebcamTextureScript.cs 62
Παράρτημα Β 64
1. Απαιτούμενο Υλικό και Λογισμικό 64
1.1 Απαιτούμενο Υλικό (Hardware) 64
1.2 Απαιτούμενο Λογισμικό (Software) 64
2. Εγχειρίδιο Χρήστη 64
2.1 Οδηγίες σύνδεσης συσκευών 64
2.2 Οδηγίες εγκατάστασης απαιτούμενου λογισμικού 65
2.3 Οδηγίες χρήσης εφαρμογής 66

Κεφάλαιο 1

Εισαγωγή

1.1 Κίνητρο 1
1.2 Σκοποί και Στόχοι 2
1.3 Δομή της Διπλωματικής 4

1.1 Κίνητρο

Η εικονική πραγματικότητα θα μπορούσε να ορισθεί ως η «Απόλυτη Οθόνη», η οποία προσφέρεται σε κάθε χρήστη ξεχωριστά, ως μια καινούρια εμβυθιστική εμπειρία με καινοτομικά χαρακτηριστικά και νέες αλληλεπιδράσεις.

Κύριος στόχος της εικονικής πραγματικότητας είναι να μπορεί να δώσει την αίσθηση στον χρήστη ότι ο εικονικός κόσμος στον οποίο αλληλοεπιδρά είναι ή τουλάχιστον φαίνεται πραγματικός [1]. Επομένως ο χρήστης θα βιώσει την εμπειρία της εικονικής πραγματικότητας, ώστε να εμβυθιστεί στο εικονικό περιβάλλον μέσω αλληλεπίδρασης χωρίς όπως να απολέσει την επαφή του με τον πραγματικό κόσμο.

Αναρίθμητες εφαρμογές εικονικής πραγματικότητας έχουν δημιουργηθεί τα τελευταία χρόνια, ενώ ο αριθμός τους αναμένεται να μεταβάλλεται με αυξανόμενο ρυθμό, καθώς στον τομέα αυτό δεν υπάρχουν όρια, ενώ νέες ιδέες υλοποιούνται τεχνολογικά με σκοπό τη συνεχή βελτίωση της αποτελεσματικότητας στη χρήση της εικονικής πραγματικότητας. Οι εφαρμογές αυτές έχουν διαφορετικά τεχνολογικά χαρακτηριστικά και επικεντρώνονται σε ένα ευρύ φάσμα υπηρεσιών που αποσκοπεί σε διάφορες πληθυσμιακές κατηγορίες. Αυτό συμβαίνει διότι οι επιστήμονες προσεγγίζουν την εικονική πραγματικότητα ως μέσο έρευνας, εκπαίδευσης ή και ψυχαγωγίας. Συνεπώς, οι εφαρμογές αυτές αφορούν διάφορες ηλικιακές ομάδες όπως παιδιά, ενήλικους (παιχνίδια, ταινίες), μαθητές (εκπαιδευτικές εφαρμογές), επιστήμονες (έρευνα).

Το πρόβλημα όμως που μπορεί να πιστοποιήσει ακόμη και ο πιο φιλόδοξος ερευνητής, είναι ότι τα πλείστα μέσα προβολής και χρήσης εικονικής πραγματικότητας, αποξενώνουν τον χρήστη από το περιβάλλον στο οποίο βρίσκεται. Δηλαδή παύει να έχει την αίσθηση τόσο του χρόνου όσο και του χώρου όπου αλληλεπιδρά, καθώς το μέσο προβολής της εικονικής πραγματικότητας καλύπτει την όραση του στον εξωτερικό κόσμο, οπόταν και ο χρήστης επικεντρώνεται στην οθόνη της συσκευής που χειρίζεται. Αυτό για μερικούς ίσως να μην είναι ιδιαίτερο πρόβλημα αφού όπως προαναφέρθηκε κύριο μέλημα της εικονικής πραγματικότητας είναι η απόκτηση και αποστήθιση μιας διαφορετικής εμπειρίας σε σχέση με την πραγματικότητα (εμβύθιση). Εδώ υποστηρίζεται λοιπόν και η δική μας γνώμη σύμφωνα με την οποία ο χρήστης που χειρίζεται οποιεσδήποτε τεχνολογίες προβολής εικονικής πραγματικότητας, να μπορεί να αποκτήσει νέες δυνατότητες στον χώρο τον οποίο βρίσκεται χωρίς όμως να χάνει επαφή με το περιβάλλον στο οποίο αλληλεπιδρά.

Η αποστασιοποίηση του χρήστη είναι ένα από τα προβλήματα που μπορεί να αντιμετωπίσει η επαυξημένη πραγματικότητα. Κι αυτό γιατί η επαυξημένη πραγματικότητα χρησιμοποιείται ήδη σε μεγάλο βαθμό για την υλοποίηση εφαρμογών που επιλύουν διάφορα καθημερινά προβλήματα. Μεταξύ άλλων μερικά από αυτά είναι η δοκιμή επίπλων, ρούχων και γενικά καθημερινών αντικειμένων σε γνωστούς χώρους για τον χρήστη, GPS, παιχνίδια κλπ.

Κίνητρα λοιπόν αυτής της διπλωματικής αποτέλεσαν γενικότερα οι προκλήσεις της εικονικής πραγματικότητας όπως επίσης και η αποτελεσματικότερη χρήση των αναρίθμητων εφαρμογών της επαυξημένης πραγματικότητας στον χώρο της τεχνολογίας. Καθώς ο τομέας αυτός εξελίσσεται με ραγδαίο ρυθμό, πιστεύεται πως νέες δυνατότητες και νέες λειτουργίες εναπόκειται να δημιουργηθούν στο μέλλον με σκοπό να υποστηρίξουν και να βελτιώσουν την εμπειρία αυτή.

1.2 Σκοποί και Στόχοι

Έχοντας λάβει υπόψη τα πιο πάνω κίνητρα, η παρούσα διπλωματική επικεντρώνεται στο πως ένας απλός χρήστης θα έχει τη δυνατότητα να αλληλεπιδρά σε ένα ψηφιακό περιβάλλον όμοιο με το πραγματικό, έχοντας στη διάθεση του λειτουργίες οι οποίες δεν ήταν υπαρκτές και διαθέσιμες προηγουμένως.

Ένας από του σκοπούς της διπλωματικής είναι να μετατρέψει την συσκευή προβολής εικονικής πραγματικότητας Oculus Rift [2], σε οθόνη προβολής επαυξημένης πραγματικότητας. Αυτό συνεπάγεται ότι μέσω του Oculus Rift και των βοηθητικών συσκευών της διασύνδεσης, ο χρήστης θα μπορεί να τοποθετεί αντικείμενα στον πραγματικό χώρο, σε πραγματικό χρόνο. Επομένως, θα μελετηθούν υφιστάμενοι τρόποι που μπορούν να επιλύσουν το πρόβλημα αυτό και κατά πόσο είναι εφικτοί να πραγματοποιηθούν και να δώσουν τα ανάλογα αποτελέσματα στην παρούσα φάση. Η τεχνολογία όπως είναι γενικά παραδεκτό μεταβάλλεται συνεχώς με γοργούς ρυθμούς όπου οι νέες τεχνολογίες αναμένεται να συμβάλουν στη βέλτιστη και αποτελεσματικότερη χρήση της εικονικής πραγματικότητας.

Επιπρόσθετα θα αξιολογηθούν οι δυνατότητες των συσκευών που θα χρησιμοποιηθούν στην έρευνα αυτή (Oculus Rift, Kinect, Logitech Cameras) και κατά πόσο μπορούν να προσδώσουν την εμπειρία που αναφέρθηκε προηγουμένως. Μέσω πειραμάτων και ενδεικτικής χρήσης των συσκευών αυτών θα αναλυθεί η αλληλεπίδραση του χρήστη με το περιβάλλον σε πραγματικό χρόνο, τα προβλήματα που τυχόν θα προκύψουν από την χρήση των συσκευών όπως επίσης και τυχόν συνέπειες από τα προβλήματα αυτά.

Κύριο μέλημα της διπλωματικής είναι να χρησιμοποιηθούν με τον καλύτερο δυνατό τρόπο οι διαθέσιμες συσκευές ώστε να μπορούν να συνεργαστούν μεταξύ τους για την επίτευξη του καλύτερου αποτελέσματος. Συγκεκριμένα, θα τοποθετηθούν δύο κάμερες επάνω στην πρόσοψη του Oculus Rift, ώστε να μπορούν να δώσουν την εικόνα του χώρου σε πραγματικό χρόνο μέσα στο Oculus Rift. Οι δύο κάμερες θα έχουν κάποια απόσταση μεταξύ τους, δίνουν μια τρισδιάστατη απεικόνιση του χώρου (Stereoscopic 3D), εν αντιθέσει με την μια κάμερα η οποία δίνει επίπεδη εικόνα, κατά την οποία δεν μπορεί να γίνει αντιληπτή από τον χρήστη η απόσταση και το βάθος μεταξύ των αντικειμένων του χώρου.

Αφού εφαρμοστεί το πιο πάνω, στην συνέχεια θα τοποθετηθεί αισθητήρας βάθους ο οποίος είναι διαθέσιμος στην συσκευή Kinect [3], ώστε να αποκτηθεί η γεωμετρία του χώρου που είναι στο πεδίο εμβέλειας του αισθητήρα. Έχοντας και τις τρεις συσκευές ενωμένες και μέσω ενδιάμεσου λογισμικού, θα χρησιμοποιηθούν τα δεδομένα βάθους του παρόντος χώρου αλληλεπίδρασης ως αναφορά για τοποθέτηση εικονικών αντικειμένων, στατικών και δυναμικών, κατά μήκος του χώρου. Η τεχνική αυτή είναι γνωστή και ως επαυξημένη πραγματικότητα (Augmented Reality) και θα

χρησιμοποιηθεί στην παρούσα έρευνα. Αυτά τα εικονικά αντικείμενα θα είναι ορατά μέσω της εικόνας που θα προβάλλεται από το Oculus Rift σε πραγματικό χρόνο, οπόταν όπως γίνεται αντιληπτό θα δώσουν μια νέα αίσθηση του τρισδιάστατου χώρου.

Η διπλωματική εργασία στοχεύει να αναδείξει την παρουσία της εικονικής πραγματικότητας με προβολή εικόνας σε πραγματικό χρόνο, η οποία αναμένεται να προσφέρει καινοτόμα χαρακτηριστικά και να δώσει στον χρήστη νέες δυνατότητες οι οποίες δεν ήταν εφικτές ούτε μπορούσαν να εφαρμοστούν προηγουμένως. Ο σχεδιασμός θα αναδείξει τα χαρακτηριστικά και τις δυνατότητες του Oculus Rift αλλά και του Kinect, συσκευές οι οποίες ήδη χρησιμοποιούνται ευρέως ανά το παγκόσμιο. Ειδικότερα θα αναδείξει επιπρόσθετες πτυχές της συσκευής προβολής εικονικής πραγματικότητας Oculus Rift, η οποία αναμένεται να είναι διαθέσιμη στην αγορά το επόμενο έτος, καθώς ακόμη βρίσκεται σε δοκιμαστικό στάδιο.

Τέλος, ακόμη ένας στόχος που έχει τεθεί είναι να μπορεί κάποιος σε μεταγενέστερο στάδιο να βελτιώσει και να αναδείξει περεταίρω την έρευνα αυτή καθώς βρίσκεται σε δοκιμαστικό στάδιο. Οι δυνατότητες που μπορούν να προβληθούν είναι αναρίθμητες ενώ οι βελτιώσεις που μπορεί να γίνουν θα δείξουν ότι η εικονική πραγματικότητα είναι ένας τομέας ο οποίος θα συνεχίσει να αναπτύσσεται αλλά το κυριότερο να δύναται να αντικαταστήσει υφιστάμενες τεχνολογίες που βελτιώνουν την ανθρώπινη καθημερινότητα.

1.3 Δομή της Διπλωματικής

Τα στάδια που ακολουθήθηκαν κατά την διπλωματική εργασία και τα οποία αποτελούν τη δομή των κεφαλαίων της είναι τα ακόλουθα.

Στο **Κεφάλαιο 1** παρουσιάζεται η κύρια ιδέα της έρευνας, τα κίνητρα που ώθησαν σε αυτήν, όπως επίσης και οι σκοποί και στόχοι της διπλωματικής εργασίας. Ιδιαίτερη έμφαση δίνεται στον τρόπο σύνδεσης των διαθέσιμων τεχνολογικών συσκευών, ο οποίος με ενδιάμεσο λογισμικό θα δώσει το επιθυμητό αποτέλεσμα που αποτελεί τον στόχο της διπλωματικής εργασίας. Δηλαδή την προβολή εικονικών/μη πραγματικών αντικειμένων στο χώρο σε συνεργασία με προβολή εικόνας σε πραγματικό χρόνο.

Στο **Κεφάλαιο 2** θα μελετηθούν και θα αναλυθούν οι υφιστάμενοι τρόποι υλοποίησης του προβλήματος-στόχου της διπλωματικής εργασίας και συγκεκριμένα πώς και εάν πραγματοποιήθηκε το αναμενόμενο αποτέλεσμα. Θα μελετηθούν τα αποτελέσματα, τα

προβλήματα που προέκυψαν όπως επίσης και τυχόν δυσκολίες ή αδυναμίες που εμφανίστηκαν κατά την διάρκεια της έρευνας. Η μελέτη των υφιστάμενων τεχνικών θα βασιστεί σε διεθνή βιβλιογραφία η οποία εστιάζεται σε προηγούμενες έρευνες και άρθρα, όπως επίσης και άλλο σχετικό υλικό που είναι αναρτημένο στο διαδίκτυο, ενώ επικεντρώνεται πρωτίστως στον τρόπο υλοποίησης του προβλήματος αλλά και στα αποτελέσματα που είχε η κάθε έρευνα.

Το **Κεφάλαιο 3** ξεκινά με μια σύντομη επισκόπηση της εφαρμογής που έγινε κατά την έρευνα, ενώ έπειτα ακολουθούν τα υπόλοιπα μέρη του κεφαλαίου στα οποία γίνεται αναφορά στις συσκευές που χρησιμοποιήθηκαν κατά την υλοποίηση αλλά και σε εναλλακτικές συσκευές που δύνανται να χρησιμοποιηθούν εξίσου. Στην αρχή γίνεται αναφορά στο Oculus Rift, συσκευή η οποία χρησιμοποιήθηκε ως συσκευή προβολής εικονικής πραγματικότητας για τους σκοπούς της έρευνας, ενώ γίνεται αναλυτική περιγραφή των χαρακτηριστικών και δυνατοτήτων της συσκευής. Επίσης, για λόγους ενημέρωσης παρουσιάζονται οι συσκευές οι οποίες θα μπορούν να αντικαταστήσουν στο μέλλον το Oculus Rift, αναφέροντας μερικά από τα πλεονεκτήματα αλλά και μειονεκτήματά τους. Παρομοίως, γίνεται αναφορά στο Microsoft Kinect, συσκευή η οποία χρησιμοποιήθηκε ως συσκευή ανίχνευσης βάθους για τους σκοπούς της έρευνας, ενώ γίνεται εκτενής ανάλυση των χαρακτηριστικών της συσκευής. Έπειτα, γίνεται αναφορά σε άλλες συσκευές και αισθητήρες οι οποίοι θα μπορούν στο μέλλον να αντικαταστήσουν το Microsoft Kinect, ενώ γίνεται αναφορά στα χαρακτηριστικά τους. Στη συνέχεια γίνεται αναφορά στις κάμερες Logitech Cameras C310 που χρησιμοποιήθηκαν στην έρευνα [4]. Παράλληλα, παρουσιάζονται τα χαρακτηριστικά που έχουν σε σχέση με τα χαρακτηριστικά που θα έπρεπε να είχαν για αποφυγή σφαλμάτων και ομαλή λειτουργία σύμφωνα και με τα όσα αναφέρθηκαν στο κεφάλαιο 2, μέρος 2.5. Τέλος, στο τελευταίο μέρος του κεφαλαίου γίνεται αναφορά σε επιπλέον συσκευές που χρησιμοποιήθηκαν κατά την έρευνα όπως επίσης και διάφορες εφαρμογές που χρησιμοποιήθηκαν για την υλοποίηση του κώδικα.

Στο **Κεφάλαιο 4** θα γίνει ανάλυση του εργαλείου το οποίο χρησιμοποιήθηκε για την παραγωγή της ενδιάμεσης εφαρμογής η οποία υποστήριξε την λειτουργία των συσκευών που χρησιμοποιήθηκαν και βοήθησε στη σύνδεσή τους. Θα γίνει αναφορά στον τρόπο υλοποίησης της εφαρμογής, οι αλγόριθμοι που χρησιμοποιήθηκαν και τέλος ο τρόπος σύνδεσης των συσκευών και η αξιοποίηση των δεδομένων που ήταν

διαθέσιμα από τη κάθε συσκευή. Θα προβληθούν επίσης διάφορες τεχνικές που χρησιμοποιήθηκαν και βοήθησαν ώστε η χρήση της εφαρμογής να είναι όσο το δυνατό πιο απλή χωρίς να επισκιάζει τις δυνατότητες. Σε μεταγενέστερο στάδιο θα αναφερθούν τα προβλήματα που προέκυψαν κατά την υλοποίηση και τυχόν σφάλματα που υπάρχουν, ενώ θα μελετηθούν τα αποτελέσματα από την αλληλεπίδραση του χρήστη με τις συσκευές μέσω της εφαρμογής.

Στο **Κεφάλαιο 5** θα παρουσιαστούν τα αποτελέσματα της παρούσας διπλωματικής, όπως επίσης και κατά πόσο επιτεύχθηκε ο αρχικός στόχος που είχε τεθεί. Θα παρουσιαστούν επιπρόσθετα τα συμπεράσματα που εξάχθηκαν ενώ θα καταγραφεί και η συνεισφορά της διπλωματικής στην εικονική πραγματικότητα. Τέλος, το έβδομο κεφάλαιο κλείνει με τυχόν βελτιώσεις που μπορούν να υπάρξουν και αφορούν την υλοποίηση (τεχνικά θέματα, θέματα κώδικα), όπως επίσης και μελλοντικές πτυχές (επιπρόσθετες συσκευές, νέες τεχνικές) οι οποίες θα βελτιώσουν τις υφιστάμενες λειτουργίες ενώ θα παρέχουν και επιπρόσθετες δυνατότητες προς τους χρήστες.

Κεφάλαιο 2

Προηγούμενη Εργασία

2.1 Εισαγωγή 6
2.2 Ανακατασκευή χώρου με κινούμενη κάμερα σε πραγματικό χρόνο 6
2.3 Ανακατασκευή χώρου με επαύξηση εικονικών αντικειμένων 9
2.4 Επιτραπέζια επαυξημένη πραγματικότητα 10
2.5 Επαυξημένη πραγματικότητα μέσω Oculus Rift (AR-Rift) 12
2.6 Συμπεράσματα 15

2.1 Εισαγωγή

Η εικονική πραγματικότητα απασχολεί εδώ και πολλά χρόνια την επιστήμη της πληροφορικής όπως επίσης και άλλες δευτερεύουσες επιστήμες οι οποίες ασχολούνται με αυτό το συναφές θέμα. Το πρόβλημα της αποξένωσης του χρήστη υπάρχει από τον καιρό της ανακάλυψης της εικονικής πραγματικότητας οπόταν και πολλοί επιστήμονες προσπάθησαν να δώσουν λύση με διάφορες τεχνικές και μεθόδους. Στο παρόν κεφάλαιο θα αναλυθούν οι λύσεις ολικές ή μερικές, που προτάθηκαν από διάφορους επιστήμονες σε πλαίσιο επιστημονικής έρευνας. Συγκεκριμένα θα αναφερθούν οι τεχνικές που χρησιμοποιήθηκαν στην κάθε περίπτωση, τα προβλήματα που τυχόν υπήρξαν όπως επίσης και τα αποτελέσματα της κάθε έρευνας ξεχωριστά.

Οι έρευνες που μελετήθηκαν βασίζονται σε διεθνή βιβλιογραφία και ως επί το πλείστων αφορούν άρθρα, ενώ μια από αυτές αφορά πρακτική εφαρμογή του προβλήματος με βιντεοσκοπικό υλικό για υποστήριξη της λύσης. Στο μέρος της βιβλιογραφικής ανασκόπησης παρουσιάζονται αναλυτικά τα άρθρα τα οποία επικεντρώνονται στον τρόπο υλοποίησης των μεθόδων και τεχνικών που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία. Όλα τα άρθρα που επικεντρώνονται στον τρόπο υλοποίησης

είναι αναρτημένα στο διαδίκτυο, και στα οποία θα γίνει παραπομπή μέσω της βιβλιογραφίας της παρούσας διπλωματικής εργασίας.

2.2 Ανακατασκευή χώρου με κινούμενη κάμερα σε πραγματικό χρόνο

Ένα από τα άρθρα τα οποία μελετήθηκαν κατά την έρευνα αφορά την ζωντανή ανακατασκευή χώρου μέσω κινούμενης κάμερας [5]. Το συγκεκριμένο άρθρο παρουσιάζει μια μέθοδο για γρήγορη ανακατασκευή ενός χώρου ή μιας σκηνής χρησιμοποιώντας μια ζωντανή κάμερα. Συγκεκριμένα σκοπός της ανακατασκευής είναι η δημιουργία τρισδιάστατου πλέγματος (mesh) και έπειτα σε πίνακες βάθους (depth maps), ώστε να μπορεί να γίνει ανακατασκευή της αρχικής σκηνής η οποία να είναι όσο το δυνατόν πιο ακριβής. Έπειτα το δημιουργημένο πλέγμα ή ο πίνακας βάθους, σύμφωνα με τον συγγραφέα του άρθρου θα έχει εφαρμογή σε διάφορους τεχνολογικούς τομείς συμπεριλαμβανομένου της επαυξημένης πραγματικότητας (augmented reality) με επίγνωση της γεωμετρίας του χώρου, τομέα με τον οποίο ασχολείται η παρούσα διπλωματική εργασία.

Η μεθοδολογία και οι τεχνικές που χρησιμοποιούνται για την ανακατασκευή χώρου παρουσιάζονται αναλυτικά στο προαναφερθέν άρθρο. Περιληπτικά κατά την κίνηση της μίας κάμερας με ζωντανή εικόνα σε πραγματικό χρόνο, δημιουργείται μια τρισδιάστατη συλλογή από σημεία με συντεταγμένες των διαστάσεων x,y,z γνωστή και ως Point Cloud [6]. Τα σημεία αυτά αντιστοιχίζονται με τα σημεία του πραγματικού χώρου βάση της θέσης τους (x,y,z) και του βάθους σε σχέση με την κάμερα, ώστε αργότερα να μπορεί να γίνει εφικτή η ανακατασκευή του αρχικού χώρου. Η τεχνική της συλλογής σημείων είναι γνωστή ως SFM/SLAM [7] που χρησιμοποιείται και στην ρομποτική, και κατά την οποία πρώτα εκτιμάται και υπολογίζεται η θέση της κάμερας και έπειτα γίνεται υπολογισμός της συλλογής σημείων.

Σε μεταγενέστερο στάδιο μέσω της συλλογής σημείων γίνεται η ανακατασκευή της σκηνής σε τρισδιάστατο πλέγμα με την βοήθεια αλγορίθμων, ενώ στην συνέχεια γίνεται βελτιστοποίηση του πλέγματος ώστε να εξαλειφθούν τυχόν ατέλειες (τρύπες στο πλέγμα, επιπρόσθετα σημεία εκτός πλέγματος). Μεταξύ των τεχνικών ανακατασκευής της σκηνής από μια συλλογή σημείων, είναι και η τριγωνοποίηση του πίνακα βάθους κατά την οποία μέσω του τύπου που περιγράφεται αναλυτικά στο άρθρο [5], (vj = Djrj

+ tref), κάθε σημείο του πίνακα βάθους από τη συλλογή σημείων (depth pixel) μετατρέπεται σε κορυφή τριγώνου (vertex). Οι κορυφές τριγώνων ενώνονται μεταξύ τους δημιουργώντας τρίγωνα και έπειτα τα τρίγωνα ενώνονται μεταξύ τους δημιουργώντας το τρισδιάστατο πλέγμα. Η τεχνική αυτή ακολουθήθηκε κατά την μεθοδολογία της παρούσας διπλωματικής εργασίας και θα αναλυθεί περεταίρω στο έκτο κεφάλαιο, κεφάλαιο στο οποίο θα αναλυθούν επίσης οι όροι πλέγμα, τριγωνοποίηση κλπ.

Η προαναφερόμενη μέθοδος αν και δίνει λύση σε ένα από τα προβλήματα της έρευνας της παρούσας διπλωματικής εργασίας, εντούτοις έχει κάποια μειονεκτήματα. Ο χώρος καταγραφής και επομένως το τρισδιάστατο πλέγμα που δημιουργείται επηρεάζεται από το οπτικό πεδίο της κάμερας όπως και τον ρυθμό ανανέωσης και εγγραφής της. Συνεπώς αυτό έχει ως αποτέλεσμα μεγαλύτερο τρισδιάστατο πλέγμα να απαιτεί μεγαλύτερο οπτικό πεδίο κάμερας. Επομένως γίνεται αντιληπτό ότι η ανακατασκευή ενός ολόκληρου δωματίου και όχι ενός συγκεκριμένου χώρου του δωματίου, γίνεται δυσκολότερη και απαιτεί περεταίρω υπολογισμούς για την επίτευξή του.

Μειονέκτημα επίσης αποτελεί και η μονάδα επεξεργασίας γραφικών αλλά και η φυσική μνήμη του υπολογιστή με τον οποίο είναι συνδεδεμένη η κάμερα ζωντανής εικόνας, όπως επίσης και το καλώδιο σύνδεσης της κάμερας με τον υπολογιστή. Το καλώδιο USB 3.0 σε αντίθεση με το συνηθισμένο καλώδιο USB 2.0 παρέχει την δυνατότητα ροής πολύ περισσότερων δεδομένων σε πραγματικό χρόνο. Εντούτοις κάμερες με καλωδίωση USB 3.0 άρχισαν να κάνουν την εμφάνισή τους τον τελευταίο χρόνο, επομένως αναμένεται να βελτιώσουν την διαδικασία αυτή. Επίσης, λόγω του ότι ο υπολογισμός του πλέγματος, εξαρτάται από τη μονάδα επεξεργασίας γραφικών αλλά και την φυσική μνήμη του υπολογιστή, κατά την δημιουργία του πλέγματος μπορεί να παρατηρηθούν σφάλματα (τρύπες κατά σημεία του πλέγματος, επιπρόσθετα σημεία που είναι ανεξάρτητα του πλέγματος). Τα σφάλματα αυτά μπορούν να εξαλειφθούν αργότερα με την χρήση διάφορων τεχνικών οι οποίες αναφέρονται επίσης στο άρθρο.

Έχοντας υπόψη τα πιο πάνω, συνειρμικά μπορεί να εμφανιστεί το ερώτημα κατά πόσο αυτή η τεχνική μπορεί να βοηθήσει στην επαυξημένη πραγματικότητα και στην σύνδεση της με την εικονική πραγματικότητα. Κι όμως η εικόνα 7 στο άρθρο [5] δείχνει τις δυνατότητες μιας τέτοιας τεχνικής, όπου χρησιμοποιείτε μια ανοικοδόμηση ενός στατικού περιβάλλοντος για την τοποθεσία κίνησης ενός εικονικού αυτοκινήτου.

Το αποτέλεσμα είναι η κίνηση του εικονικού αυτοκινήτου να είναι τόσο φυσική όσο η κίνηση ενός αυτοκινήτου που κινείται στον πραγματικό χώρο.

Η πιο πάνω μεθοδολογία πιστοποιήθηκε και αναπτύχθηκε σε δύο πιο πρόσφατα άρθρα στα οποία η ανοικοδόμηση χώρου σε πραγματικό χρόνο γίνεται με την βοήθεια της συσκευής Microsoft Kinect και του ενδιάμεσου λογισμικού KinectFusion [8]. Το KinectFusion είναι ένα λογισμικό το οποίο αναπτύχθηκε από την Microsoft και είναι διαθέσιμο μέσω του SDK (Software Development Kit) τόσο του Kinect V1 (Windows, Xbox 360), όσο και του Kinect V2 (Windows, Xbox One). Μέσω του λογισμικού αυτού, μπορεί να γίνει ανακατασκευή του χώρου που είναι ορατός από την συσκευή Kinect σε πραγματικό χρόνο. Στις έρευνες που αναφέρονται στα συγκεκριμένα άρθρα χρησιμοποιήθηκε το Kinect V1. Αναλυτικά οι διαφορές μεταξύ των δύο θα αναφερθούν στο κεφάλαιο 4 το οποίο ασχολείται με τους αισθητήρες ανίχνευσης βάθους. Με την εμφάνιση του Kinect V2 τα σφάλματα που δημιουργούνται κατά την ανοικοδόμηση του χώρου μέσω KinectFusion μειώνονται, καθώς ο αισθητήρας της συσκευής είναι πιο εξελιγμένος, ενώ για την ροή και μεταφορά δεδομένων χρησιμοποιείται καλωδίωση με διεπαφή USB 3.0.

Κύριος στόχος της μεθοδολογίας που αναφέρθηκε πιο πάνω, ήταν η επαυξημένη πραγματικότητα λαμβάνοντας υπόψη την γεωμετρία του χώρου στον οποίο επιχειρείται το πείραμα. Η μεθοδολογία αυτή σε συνδυασμό με άλλες τεχνικές εφαρμόστηκε μερικώς και κατά την έρευνα της παρούσας διπλωματικής εργασίας, η οποία θα αναλυθεί εκτενέστερα στο τέταρτο κεφάλαιο το οποίο ασχολείται με τα τεχνικά χαρακτηριστικά και την υλοποίηση του απαιτούμενου κώδικα.

2.3 Ανακατασκευή χώρου με επαύξηση εικονικών αντικειμένων

Η αλληλεπίδραση με τρισδιάστατα μοντέλα σε επαυξημένη πραγματικότητα σε πραγματικό χρόνο περιγράφεται αναλυτικά στο άρθρο [9]. Κατά την έρευνα αυτή χρησιμοποιήθηκε μια καινοτόμα κάμερα Creative Senz3D, η οποία υποστήριζε RGB-D (RGB – χρώμα, D – βάθος) [10], το Oculus Rift, το Leap Motion (συσκευή ανίχνευσης κίνησης χεριών) [11], και δείκτες AR (Augmented Reality Markers). Το σύστημα που αναπτύχθηκε σε μετέπειτα στάδια μπορούσε να εμφανίσει τα χέρια του χρήστη σε

σχέση με τον ανιχνευτή βάθους που βρισκόταν πάνω στην κάμερα, ενώ οι δείκτες AR χρησιμοποιούνταν για να καθορίσουν την θέση της συσκευής Leap Motion.

Τα αποτελέσματα του συστήματος ήταν η ανίχνευση των κινήσεων της κεφαλής του χρήστη μέσω του Oculus Rift, και η ανίχνευση των κινήσεων των χεριών του μέσω του Leap Motion με βάση και το 6DoF – six degrees of freedom [12]. Τα πιο πάνω σε συνδυασμό με επιπρόσθετες τεχνικές βοήθησαν στην ακριβή αλληλεπίδραση του χρήστη με εικονικά αντικείμενα της σκηνής, καθώς ήταν εφικτή η έμφραξη (occlusion) μεταξύ των χεριών του χρήστη και των εικονικών αντικειμένων.

Κατά την υλοποίηση της πιο πάνω έρευνας χρησιμοποιήθηκε η κάμερα Creative Senz3D για τον υπολογισμό του πίνακα βάθους για το οποίον έγινε αναφορά και στο μέρος 2.2 του παρόντος κεφαλαίου. Έπειτα έγινε επιδιόρθωση του πίνακα βάθους κυρίως λόγω του θορύβου και των ελλιπών δεδομένων που οδηγούν στα σφάλματα που αναφέρθηκαν επίσης στο 2.2. Ο πίνακας βάθους χρησιμοποιήθηκε σε μετέπειτα στάδια για την ανακατασκευή των χεριών του χρήστη ώστε να είναι ορατά μέσω της συσκευής Oculus Rift. επίσης, ο πίνακας βάθους σημείων χρησιμοποιήθηκε για την ανακατασκευή πλέγματος ώστε ο δείκτης AR να είναι ορατός στο Oculus.

Το Leap Motion είναι μια συσκευή η οποία περιέχει αισθητήρα ανίχνευσης κίνησης χεριών. Στην πιο πάνω έρευνα χρησιμοποιήθηκε ταυτόχρονα με την βοήθεια που παρέχουν οι δείκτες AR με τη συμβολή των βιβλιοθηκών OpenCV και ArUco αναπτύχθηκε ο χειρισμός των εικονικών αντικειμένων μέσω των χεριών του χρήστη, χρησιμοποιώντας ακολουθίες χειρονομιών.

Τα αποτελέσματα της έρευνας που περιγράφεται στο άρθρο εμφανίζονται στις εικόνες 1,7 [9], όπου ο χρήστης αλληλεπιδρά με ένα τρισδιάστατο κύβο, ενώ η όλη διασύνδεση εμφανίζεται στην εικόνα 6 [9]. Οι ερευνητές κατόρθωσαν να ωθήσουν μια πιθανή λύση του προβλήματος που έχει τεθεί ως στόχος στην παρούσα διπλωματική εργασία, και ειδικότερα τον χειρισμό τρισδιάστατων αντικειμένων. Όμως η διαδικασία αυτή αντιμετωπίζει μερικά προβλήματα συμπεριλαμβανομένων διαφόρων σφαλμάτων κατά την ανίχνευση των χεριών τα οποία σύμφωνα με τον συγγραφέα του άρθρου αυτά αναμένεται θα επιλυθούν σε μεταγενέστερο στάδιο στο μέλλον.

2.4 Επιτραπέζια επαυξημένη πραγματικότητα

Η επιτραπέζια επαυξημένη πραγματικότητα βασίζεται στην παρουσίαση επαυξημένης πραγματικότητας με τη βοήθεια επίπεδων περιοχών (πχ. τραπέζι). Ουσιαστικά χρησιμοποιούνται διάφορες τεχνικές που προαναφέρθηκαν στα προηγούμενα μέρη του κεφαλαίου όπως δείκτες AR, υπολογισμός βάθους κλπ. οι οποίοι έχουν ως στόχο η αλληλεπίδραση μεταξύ πραγματικών και εικονικών αντικειμένων να γίνεται όσο το δυνατό με τον πιο φυσικό τρόπο.

Το πρώτο πλαίσιο (framework) που περιγράφεται στο σχετικό άρθρο περιλαμβάνει αλληλεπίδραση μεταξύ αντικειμένων βάσει φυσικής προσομοίωσης, η οποία μπορεί να εφαρμοστεί μέσω διάφορων μηχανών (engines) μεταξύ των Havok και Nvidia PhysX [13]. Αφού, γίνει η φυσική προσομοίωση απαιτείται να διασφαλισθεί η επίγνωση του περιβάλλοντος αλληλεπίδρασης προκειμένου τα εικονικά αντικείμενα να αναγνωρίζουν τυχόν συγκρούσεις με τα πραγματικά αντικείμενα και να συμπεριφέρονται αρμονικά, δηλαδή με φυσικό τρόπο. Για τον σκοπό αυτό χρησιμοποιήθηκε το Kinect, το οποίο ανιχνεύει το βάθος του χώρου οπόταν και μπορεί να συμβάλει στην ανακατασκευή του πραγματικού χώρου σε εικονικό. Το Kinect βρίσκεται σε κάποια απόσταση πάνω από το τραπέζι στο οποίο γίνονται όλες οι λειτουργίες και λαμβάνουν μέρος οι αλληλεπιδράσεις. Η διαδικασία αυτή απαιτεί όπως ο χώρος αλληλεπίδρασης να είναι σε συνεχή διαθεσιμότητα. Για να είναι αυτό εφικτό οι ερευνητές τοποθέτησαν και δεύτερη κάμερα η οποία θα βοηθούσε στην κίνηση του χρήστη, ώστε να μπορεί να βλέπει και να επεξεργάζεται τα εικονικά αντικείμενα που δημιουργούνται μέσω των δεικτών AR.

Αφού εφαρμοστούν τα πιο πάνω, τα επόμενα βήματα ακολουθούν την ανίχνευση των δεικτών AR, μέσα από τους οποίους ο χρήστης θα μπορεί να δει εικονικά αντικείμενα, όπως επίσης και την ανίχνευση του βάθους. Στη συνέχεια λαμβάνει μέρος η επεξεργασία του βάθους και η ευθυγράμμιση των συντεταγμένων των συσκευών, καθώς η θέση κάθε συσκευής σε σχέση με τον χώρο αλληλεπίδρασης έχει διαφορετικές συντεταγμένες. Μέσω της επεξεργασίας βάθους, γίνεται η τριγωνοποίηση του βάθους η οποία θα οδηγήσει στην κατασκευή τρισδιάστατου πλέγματος που θα βοηθήσει στην φυσική συμπεριφορά των αντικειμένων όπως και στην ανίχνευση συγκρούσεων. Τέλος, γίνεται το λεγόμενο rendering, όπου τα εικονικά αντικείμενα (στατικά και κινούμενα) εμφανίζονται στον χώρο αλληλεπίδρασης, όπου αυτά είναι ορατά μέσω της

δευτερεύουσας κάμερας, οπόταν και συμπεριφέρονται ανάλογα με το πλέγμα που δημιουργήθηκε προηγουμένως με την βοήθεια της ανίχνευσης βάθους του χώρου.

Για τον σκοπό της έρευνας για την επιτραπέζια επαυξημένη πραγματικότητα χρησιμοποιήθηκαν διάφορες βιβλιοθήκες οι οποίες υποστήριξαν τις λειτουργίες των συσκευών. Συγκεκριμένα χρησιμοποιήθηκε η βιβλιοθήκη OpenNI (Open Natural Interaction) προκειμένου να υπάρχει πρόσβαση στον αισθητήρα βάθους του Kinect για την ανάκτηση της έγχρωμης εικόνας και της εικόνας βάθους. Το OpenNI δεν υποστηρίζει το Kinect v2, που χρησιμοποιήθηκε στην έρευνα της παρούσας διπλωματικής εργασίας και προτιμήθηκε έναντι του προκατόχου του Kinect v1 για λόγους που θα επεξηγηθούν στην συνέχεια, επομένως δεν ήταν δυνατή η εφαρμογή του. Επίσης, χρησιμοποιήθηκε η βιβλιοθήκη OpenCV η οποία βοηθά στην επεξεργασία εικόνας μέσω ποικίλων αλγορίθμων, ενώ για σκοπούς ανίχνευσης δεικτών AR χρησιμοποιήθηκε η βιβλιοθήκη OPIRA. Τέλος, η φυσική συμπεριφορά των αντικειμένων πραγματοποιήθηκε με την βοήθεια της βιβλιοθήκης Bullet Physics, ενώ για τον διαχειρισμό της σκηνής αλληλεπίδρασης όπως και για σκοπούς ανατροφοδότησης (feedback) χρησιμοποιήθηκε το API γραφικών OpenSceneGraph.

Ένα από τα δείγματα που δημιουργήθηκαν για να υποστηρίξουν τις λειτουργίες της έρευνας ήταν ένα μικρό παιχνίδι με αυτοκίνητα, όπου ο παίκτης χρησιμοποιώντας ένα χειριστήριο οδηγούσε ένα εικονικό αυτοκίνητο, το οποίο συμπεριφερόταν ανάλογα με τα πραγματικά αντικείμενα που υπήρχαν στην σκηνή. Τα αποτελέσματα της έρευνας αυτής, ήταν αρκετά σημαντικά αφού η εφαρμογή αυτή σε πειραματικό στάδιο χρησιμοποιήθηκε από άτομα διαφορετικών ηλικιών προκειμένου να εξαχθούν αξιόπιστα συμπεράσματα. Η δυνατότητα αλληλεπίδρασης του φυσικού κόσμου με τον εικονικό ήταν εμφανής, ενώ οι ερευνητές έχουν ως στόχο στο μέλλον την περεταίρω ανάπτυξη του συστήματος για περισσότερη ακρίβεια σε φυσικές καταστάσεις.

Μια διαφορετική και συνάμα παρόμοια περίπτωση επαυξημένης πραγματικότητας σε επιτραπέζια βάση ήταν και το MirageTable που παρουσιάστηκε σε ένα ακόμη σχετικό άρθρο [14]. Η κύρια διαφορά με την προηγούμενη εφαρμογή είναι ότι τα εικονικά αντικείμενα ήταν ορατά χωρίς την χρήση κάμερας αλλά με την βοήθεια κυρτής οθόνης και γυαλιά με υποστήριξη τρισδιάστατης όρασης (Nvidia 3D Vision). Παράλληλα, όπως και στην προηγούμενη περίπτωση χρησιμοποιήθηκε το Kinect για σκοπούς ανίχνευσης βάθους.

Το σύστημα αυτό αν και είχε κάποιες ομοιότητες με το προηγούμενο, χρησιμοποίησε διαφορετικό τρόπο μεθοδολογίας αλλά και αλληλεπίδρασης του χρήστη. Ο χρήστης είχε τη δυνατότητα τοποθέτησης ενός πραγματικού αντικειμένου επάνω στην κυρτή οθόνη, και στη συνέχεια γινόταν σάρωση του αντικειμένου και προβολή του στην οθόνη. Έπειτα ο χρήστης μπορούσε με την βοήθεια των χεριών του να αλληλεπιδράσει με τα εικονικά αντικείμενα τα οποία συμπεριφέρονταν με φυσικό τρόπο. Τα θετικά της μεθοδολογίας αυτής, ήταν ότι δεν χρησιμοποιήθηκαν δείκτες AR, ενώ δεν ήταν απαραίτητη η χρήση γαντιών για σκοπούς ανίχνευσης κίνησης χεριών αφού αυτή γινόταν με την βοήθεια του Kinect.

2.5 Επαυξημένη πραγματικότητα μέσω Oculus Rift (AR-Rift)

Στόχος της συγκεκριμένης έρευνας ήταν να κατασκευαστεί μια βάση στην οποία ενσωματώθηκαν 2 κάμερες, από τις οποίες ο χρήστης θα μπορούσε να έχει ορατότητα σε τρισδιάστατη εικονική πραγματικότητα με την βοήθεια και του Oculus Rift. Κύριο μέλημα της έρευνας ήταν να ελαχιστοποιήσει το χάσμα ανάμεσα στην εικονική πραγματικότητα και την φυσική πραγματικότητα. Η έρευνα η οποία παρουσιάζεται πρακτικά στην ιστοσελίδα [15], έγινε από τον ερευνητή του University College of London, William Steptoe. Συγκεκριμένα, αναλύονται τα βήματα της έρευνας που οδήγησαν στην επίτευξη του στόχου, ενώ στο τέλος υπάρχει και βιντεοσκοπικό υλικό για την υποστήριξη των αποτελεσμάτων της έρευνας αυτής. Οι συσκευές που χρησιμοποιήθηκαν κατά την έρευνα όπως και τα βήματα που οδήγησαν στην επίτευξη του στόχου θα αναλυθούν στην συνέχεια.

Η συσκευή προβολής εικονικής πραγματικότητας η οποία χρησιμοποιήθηκε είναι το Oculus Rift Developer Kit 1 το οποίο δεν είναι διαθέσιμο στην αγορά παρά μόνο για προγραμματιστές και ερευνητές, οι οποίοι το χρησιμοποιούν για υποστήριξη της έρευνάς τους. Η επιλογή των καμερών βασίστηκε στα τεχνικά χαρακτηριστικά του Oculus Rift DK1, καθώς η όραση του χρήστη εξαρτάται από τα χαρακτηριστικά του Oculus Rift. Συγκεκριμένα κάθε κάμερα θα έπρεπε να έχει ανάλυση 800x600pixels η οποία είναι η υφιστάμενη ανάλυση του Oculus Rift για κάθε μάτι. Σε περίπτωση που η ανάλυση της κάμερας υπερβαίνει την ανάλυση του Oculus Rift, δεν θα υπάρχει πρόβλημα, χωρίς όμως να μπορεί να είναι εφικτό το αντίθετο. Το ίδιο ισχύει και για την αναλογία διαστάσεων του αισθητήρα της ανάλυσης της κάμερας (Aspect Ratio), η

οποία για σκοπούς ενημέρωσης υπολογίζεται ως εξής. Έστω η ανάλυση μιας κάμερας είναι 1080x720pixels, τότε το aspect ratio υπολογίζεται σύμφωνα με την ακόλουθο τύπο: Aspect Ratio = 1080/720 = 1.5:1. Επιπρόσθετα, το οπτικό πεδίο κάθε κάμερας θα έπρεπε να είναι 120° ενώ ο ρυθμός εγγραφής (capture rate) θα έπρεπε να είναι στα 60Hz, ώστε να αντιπαραβάλλεται τον ρυθμό ανανέωσης (refresh rate) του Oculus Rift DK1 ο οποίος μετριέται εξίσου στα 60Hz. Τέλος, για προγραμματιστικούς σκοπούς ήταν απαραίτητη η ύπαρξη διεπαφής USB μέσω από την οποία θα γινόταν η μεταφορά δεδομένων από την κάμερα στον υπολογιστή.

Επομένως σύμφωνα και με τα πιο πάνω ο ερευνητής κατάληξε στην επιλογή 2 καμερών Logitech C310. Οι κάμερες αυτές πληρούσαν ως επί το πλείστων τα απαιτούμενα χαρακτηριστικά. Συγκεκριμένα η ανάλυση όπως επίσης και η αναλογία του αισθητήρα ανάλυσης της κάμερας ξεπερνούσε τα ανάλογα χαρακτηριστικά του Oculus Rift κάτι που όπως αναφέρθηκε προηγουμένως ήταν ιδανικό, ενώ ο ρυθμός ανανέωσης, όπως επίσης και η διεπαφή USB για την μεταφορά των δεδομένων ήταν δεδομένα χαρακτηριστικά της κάμερας. Το οπτικό πεδίο της κάμερας όμως βρισκόταν στις 60° αντί στις 120°, επομένως για το σκοπό αυτό χρησιμοποιήθηκαν φακοί από 2 κάμερες Genius WideCam F100. Τέλος, κατασκευάστηκε μια βάση από τον ερευνητή, η οποία τυπώθηκε μέσω τρισδιάστατου εκτυπωτή υποστηρίζοντας έτσι την στήριξη των καμερών επάνω στο Oculus Rift.

Η σύνδεση του Oculus Rift όπως επίσης και των καμερών επάνω στον υπολογιστή, ο οποίος θα υποστήριζε την αμφίδρομη λειτουργία μεταξύ τους, απαιτεί πολλές καλωδιώσεις οπόταν και η κίνηση/μεταφορά κάποιου χρήστη στον χώρο ήταν δύσκολη έως αδύνατη. Ειδικότερα για σκοπούς καλύτερης πρακτικής και καλύτερης αλληλεπίδρασης του χρήστη με τις συσκευές, ο ερευνητής χρησιμοποίησε τις συσκευές ASUS Wavi, και την επαναφορτιζόμενη μπαταρία EasyAcc 15000mAh. Η συσκευή ASUS Wavi χρησιμοποιήθηκε για την ασύρματη μεταφορά της εικόνας HDMI από τον υπολογιστή στο Oculus Rift, επομένως αποφεύχθηκε η χρήση καλωδίου HDMI, καλώδιο το οποίο είναι γνωστό από την σημερινή ευρεία χρήση του από πολλές συσκευές για την μεταφορά εικόνας συνήθως από την συσκευή σε μια οθόνη προβολής (τηλεόραση).

Ακόμη μια πρόκληση ήταν ότι το Oculus Rift DK1 για την ομαλή του λειτουργία απαιτεί την τροφοδότηση του μέσω συνεχούς παροχής ρεύματος, καθώς η τροφοδότηση

μέσω του καλωδίου USB, που υποστηρίζει την μεταφορά δεδομένων, δεν ήταν αρκετή. Επομένως, ακολουθήθηκε η τεχνική του Michael J Moffitt, ο οποίος υποστήριξε ότι το Oculus Rift θα μπορούσε να τροφοδοτηθεί μέσω καλωδίου USB [16]. Ειδικότερα στην περίπτωση αυτή το Oculus Rift συνδέθηκε με την συσκευή ASUS Wavi με καλώδιο USB για σκοπούς τροφοδότησης, ενώ η μπαταρία EasyAcc θα βοηθούσε για επιπλέον χρόνο μπαταρίας. Όλες οι επιπλέον συσκευές τοποθετήθηκαν σε ένα σακίδιο, οπόταν η μεταφορά του χρήστη στον χώρο σε μεγάλο βαθμό ήταν εφικτή. Ένα από τα προβλήματα που εμφανίστηκαν, ήταν ότι η συσκευή ASUS Wavi αντιμετώπιζε προβλήματα υπερθέρμανσης κατά διαστήματα, οπόταν και έπρεπε να γίνει διακοπή της σύνδεσης για περίπου 15 λεπτά.

Για σκοπούς εντοπισμού της θέσης του χρήστη, όπως και για την ανίχνευση των κινήσεων του ανά πάσα στιγμή στον χώρο αλληλεπίδρασης, γνωστό και ως 6DOF (6 degrees of freedom) tracking [12], χρησιμοποιήθηκε το σύστημα καταγραφής οπτικής κίνησης OptiTrack που βρισκόταν στον χώρο έρευνας. Το σύστημα αποτελείτο από 12 κάμερες με ρυθμό εγγραφής 100FPS (100 καρέ ανά δευτερόλεπτο) και με την βοήθεια του λογισμικού Motive, έγινε εφικτό να υπολογισθεί η θέση του χρήστη βάση δεικτών που ήταν τοποθετημένοι επάνω στον χρήστη.

Η πιο σημαντική πρόκληση στην έρευνα υπήρξε η ευθυγράμμιση των συντεταγμένων του συστήματος ανίχνευσης του δωματίου αλληλεπίδρασης του χρήστη με την στερεοσκοπική εικόνα των 2 καμερών [35]. Η ευθυγράμμιση έγινε με τέτοιο τρόπο ώστε τα εικονικά αντικείμενα να προσαρμόζονται ανάλογα με την συμπεριφορά των πραγματικών αντικειμένων που υπάρχουν στην σκηνή, ώστε το αποτέλεσμα να προσομοιάζει περισσότερο με το φυσικό και αυτό να γίνεται αντιληπτό από το χρήστη. Η ευθυγράμμιση αποτελείται από πολύπλοκες και χρονοβόρες ενέργειες οι οποίες επεξηγούνται εκτενέστερα στο 5^{ο} μέρος της έρευνας.

Τέλος για την αλληλεπίδραση του χρήστη με τις διασύνδεση των συσκευών που χρησιμοποιήθηκαν, δημιουργήθηκε διεπαφή χρήστη η οποία ήταν ορατή στο Oculus Rift σταθμίζοντας επίσης και την τοποθεσία των δεικτών που βρίσκονταν επάνω στον χρήστη. Η διεπαφή δημιουργήθηκε με την βοήθεια του εργαλείου Unity3D και υποστήριζε μεταξύ άλλων τη δημιουργία εικονικών αντικειμένων και εικονικών ειδώλων (avatar) σε πραγματικό χρόνο, όπως επίσης και την προβολή ψηφιακών παραθύρων του υπολογιστή.

Όπως αναφέρθηκε και προηγουμένως τα αποτελέσματα της έρευνας εμφανίζονται σε βιντεοσκοπημένο υλικό στην ιστοσελίδα παραπομπής. Τα αποτελέσματα είναι άκρως εντυπωσιακά και πραγματικά δημιουργούν αίσθηση στον ψηφιακό κόσμο της εικονικής πραγματικότητας επιτυγχάνοντας την ελαχιστοποίηση του χάσματος μεταξύ πραγματικού και εικονικού κόσμου. Οι προηγμένες δυνατότητες που έχει ο χρήστης ανοίγουν τον δρόμο για νέες αλληλεπιδράσεις και νέες λειτουργίες, συμπεριλαμβανομένης της παρουσίας εικονικών αντικειμένων στην εικόνα πραγματικού χρόνου της διεπαφής, λειτουργία που οδηγεί τον χρήστη την ψευδαίσθηση ύπαρξης περισσότερων αντικειμένων στον πραγματικό χώρο αλληλεπίδρασης. Αν και ο ερευνητής προσπάθησε να δώσει την ελευθερία κίνησης στον χρήστη μέσω των δεικτών και του συστήματος καταγραφής οπτικής κίνησης, εντούτοις η αλληλεπίδραση περιορίζεται στο δωμάτιο αλληλεπίδρασης και ως εκ τούτου, αυτό αποτελεί μειονέκτημα για τον χρήστη. Η παρούσα διπλωματική εργασία εμπνεύστηκε από την πιο πάνω έρευνα και σε συνδυασμό με νέες τεχνικές και νέες συσκευές επιχειρεί να δώσει μια διαφορετική τεχνική λύση η οποία θα αναλυθεί εκτενέστερα στο έκτο κεφάλαιο.

2.6 Συμπεράσματα

Μελετώντας τα άρθρα που αναφέρθηκαν προηγουμένως παρατηρούμε ότι κύριος σκοπός και πρόκληση που είχαν οι συγγραφείς και ερευνητές των άρθρων ήταν η μείωση του χάσματος μεταξύ φυσικού και εικονικού κόσμου. Αυτό συνεπάγεται στην μείωση της αποξένωσης του ανθρώπου από την εικονική πραγματικότητα αλληλοεπιδρώντας σε περιβάλλον γνωστό για τον ίδιο.

Οι έρευνες συνδυάζουν παρόμοιες τεχνικές χρησιμοποιώντας διαφορετικές υλοποιήσεις και συνδυάζοντας τεχνολογικές συσκευές με σκοπό να επιτύχουν τον σκοπό τους. Στις έρευνες αυτές χρησιμοποιήθηκε σε μεγάλο βαθμό το Kinect της Microsoft για σκοπούς ανίχνευσης βάθους, ενώ στην περίπτωση της υφιστάμενης έρευνας της παρούσας διπλωματικής εργασίας, χρησιμοποιήθηκε η έκδοση του Kinect V2, για λόγους που θα αναφερθούν σε επόμενα κεφάλαια.

Η έρευνα επίσης εμπνεύστηκε από το παράρτημα 2.5 όπου τα αποτελέσματα ήταν εμφανή και πειστικά μέσω βιντεοσκοπικού υλικού, επομένως η επιλογή των καμερών

που χρησιμοποιήθηκαν έγινε με παρόμοιο τρόπο. Μερικά από τα άρθρα δεν παρουσίαζαν στην λύση τους την συσκευή προβολής εικονικής πραγματικότητας Oculus Rift, η οποία χρησιμοποιήθηκε επίσης στην παρούσα έρευνα για λόγους που θα αναφερθούν στην συνέχεια.

Παρατηρώντας όλες τις πιθανές λύσεις του προβλήματος, τα αποτελέσματα και τα προβλήματα που προέκυψαν στην κάθε περίπτωση, η έρευνα της διπλωματικής εργασίας προσπαθεί να δώσει τη δική της λύση στο πρόβλημα, χρησιμοποιώντας στοιχεία από τις προαναφερόμενες μελέτες. Λόγω της ανάπτυξης της τεχνολογίας και της εμφάνισης νέων συσκευών οι οποίες χρησιμοποιήθηκαν, πολλές βιβλιοθήκες που αναφέρονταν στα άρθρα δεν ήταν διαθέσιμες οπόταν και προτιμήθηκαν διαφορετικοί τρόποι υλοποίησης όπως και διαφορετικοί αλγόριθμοι, θέματα τα οποία θα συζητηθούν εκτενέστερα στο κεφάλαιο 4 που αφορά την υλοποίηση.

Κεφάλαιο 3

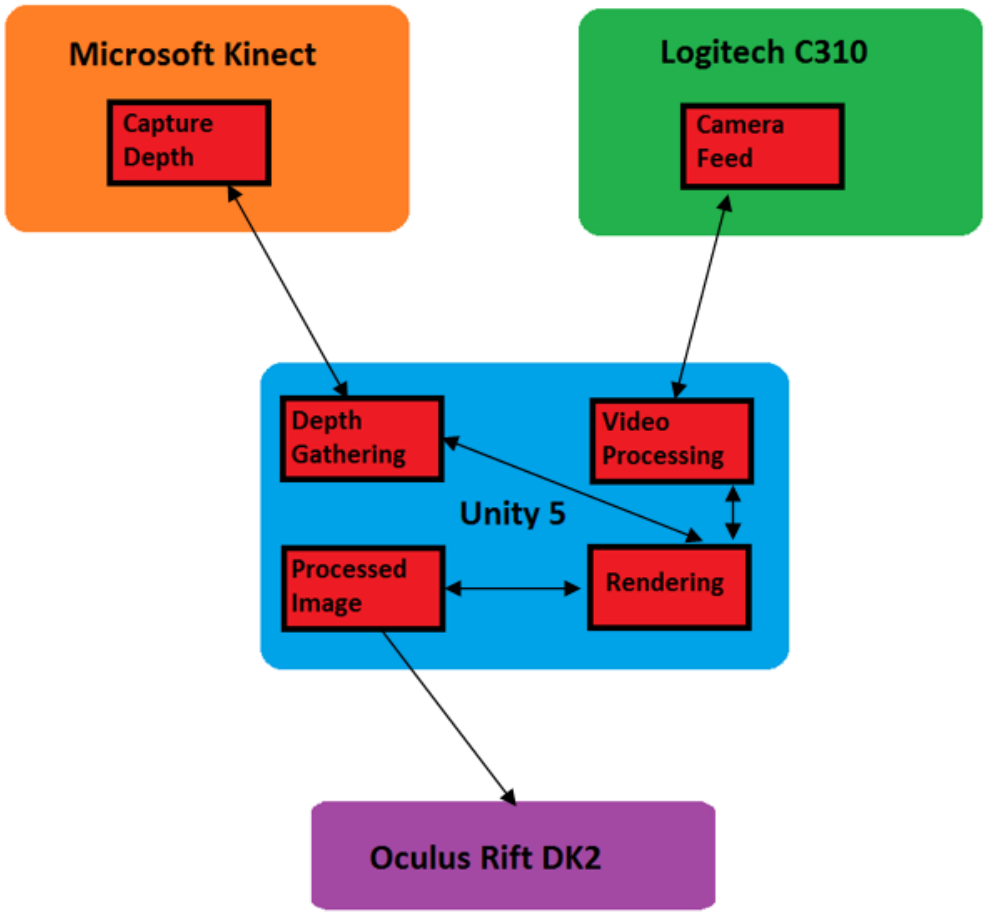
Επισκόπηση

3.1 Επισκόπηση του έργου 19
3.2 Συσκευές προβολής εικονικής πραγματικότητας 21
3.2.1 Oculus Rift 21
3.2.2 Εναλλακτικές συσκευές εικονικής πραγματικότητας 24
3.3 Συσκευές ανίχνευσης βάθους 28
3.3.1 Microsoft Kinect V2 28
3.3.2 Εναλλακτικές συσκευές ανίχνευσης βάθους 30
3.4 Κάμερες 31
3.4.1 Logitech Cameras C310 31
3.4.2 Εναλλακτικές κάμερες 32
3.5 Επιπλέον εργαλεία που χρησιμοποιήθηκαν 33
3.5.1 Unity 3D 33
3.5.2 Custom Camera Rig 35

3.1 Επισκόπηση του έργου

Γενικός στόχος της έρευνας αποτελεί η δημιουργία μιας εφαρμογής μέσω της οποίας ο χρήστης να μπορεί να τοποθετεί/αλληλεπιδρά με αντικείμενα στον πραγματικό χώρο αλληλεπίδρασης σε πραγματικό χρόνο. Συγκεκριμένα, ο χρήστης με τη βοήθεια της συσκευής προβολής εικονικής πραγματικότητας Oculus Rift θα έχει τη δυνατότητα να βλέπει τον πραγματικό χώρο αλληλεπίδρασης και επομένως να τοποθετεί εικονικά αντικείμενα επαυξάνοντας τον χώρο, δηλαδή να έχει την εμπειρία της επαυξημένης πραγματικότητας. Η εικόνα του χώρου αλληλεπίδρασης θα μεταφέρεται σε πραγματικό χρόνο από τις κάμερες, που θα βρίσκονται στην πρόσοψη του Oculus Rift σε

συγκεκριμένη απόσταση μεταξύ τους, προκειμένου να δώσουν στερεοσκοπική εικόνα, στην οθόνη του Oculus Rift [35]. Η επαύξηση των αντικειμένων στον πραγματικό χώρο θα επιτυγχάνεται επίσης με τη βοήθεια της συσκευής ανίχνευσης βάθους Kinect η οποία βοηθά στην επαυξημένη πραγματικότητα αφού αυτή παρέχει επίγνωση της γεωμετρίας του χώρου. Πιο κάτω παρουσιάζονται στο Σχήμα 1, αναλυτικά τα μέρη τα οποία αποτελούν το σύστημα όπως και ο τρόπος λειτουργίας με τον οποίο οι συσκευές ανταλλάσσουν δεδομένα σε πραγματικό χρόνο.



Σχήμα 3.1: Ανταλλαγή δεδομένων μεταξύ συσκευών

3.2 Συσκευές προβολής εικονικής πραγματικότητας

3.2.1 Oculus Rift

Το Oculus Rift [2] είναι μια συσκευή μέσω της οποίας είναι δυνατή η προβολή εικονικής πραγματικότητας συμπεριλαμβανομένου παιχνιδιών αλλά και εικονικών κόσμων. Η συσκευή τοποθετείται φορητά στο κεφάλι από τον χρήστη και λειτουργεί κυρίως ως οθόνη προβολής, καθώς καλύπτει το οπτικό πεδίο του χρήστη με τους δύο φακούς που την αποτελούν. Κύριο χαρακτηριστικό του είναι η στερεοσκοπική τρισδιάστατη προβολή εικόνας, η οποία προσδίδει στον χρήστη μια διαφορετική και συνάμα μοναδική εμπειρία. Η στερεοσκοπική εικόνα επιτυγχάνεται με παράλληλες εικόνες που στέλνονται στο οπτικό πεδίο του χρήστη σε πραγματικό χρόνο [35].

Το Oculus Rift κατασκευάστηκε από την εταιρία Oculus και ειδικότερα από τον Palmer Luckey, ο οποίος ήταν πρωτοπόρος και ένθερμος υποστηρικτής της εικονικής πραγματικότητας. Η εταιρεία αυτή αρχικά άντλησε τη χρηματοδότηση της μέσω εκστρατείας προβολής του Kickstarter προκειμένου να λανσάρει την συσκευή στην αγορά. Τελικά, με την στήριξη αρκετά γνωστών εταιρειών όπως: Valve, Epic Games και Unity, το Oculus είχε τεράστια απήχηση και επομένως οι ιδρυτές του κατόρθωσαν να έχουν επαρκή χρηματοδότηση και να προχωρήσουν στην κατασκευή του [17].

Στο σημείο αυτό αξίζει να αναφερθεί ότι τον Μάρτιο του 2014, το γνωστό κοινωνικό δίκτυο Facebook ανακοίνωσε την επίτευξη συμφωνίας για αγορά της εταιρείας Oculus, με συνεπακόλουθο τεράστια οφέλη για την εταιρεία και περαιτέρω προβολή της εικονική πραγματικότητας προς το κοινό.



Η εταιρεία Oculus κατασκεύασε πρώτα το εργαλείο Oculus Rift Developer Kit 1 τον Αύγουστο του 2012, το οποίο ήταν διαθέσιμο αρχικά σε προγραμματιστές που είχαν την θέληση να ενασχοληθούν με την παραγωγή εφαρμογών και παιχνιδιών για την εν λόγω συσκευή.



Τον Ιανουάριο του 2014, η εταιρεία προχώρησε στη δημιουργία του πρωτότυπου Crystal Cove, το οποίο έκανε την εμφάνιση του στην έκθεση Consumer Electronic Show στο Las Vegas. Η κύρια διαφορά με την προηγούμενη έκδοση ήταν ότι η συσκευή αυτή χρησιμοποιούσε μια πρόσθετη συσκευή, δηλαδή μια κάμερα θέσεως η οποία μπορούσε να ανιχνεύσει δυναμικά τις κινήσεις του χρήστη (σκύψιμο, κλίσεις κλπ.) [17].

Εμπνευσμένη από το πρωτότυπο Crystal Cove, η Oculus προχώρησε τον Μάρτιο του 2014, στην κατασκευή του Developer Kit 2, το οποίο υπερτερούσε έναντι του προκατόχου του, και το οποίο παρουσιάστηκε στη έκθεση τεχνολογίας GDC, ενώ η συσκευή ήταν επίσης διαθέσιμη για προγραμματιστές. Μερικές βελτιώσεις αποτέλεσαν η καλύτερη ποιότητα εικόνας, υψηλότερος ρυθμός ανανέωσης εικόνας, ανίχνευση κινήσεων κεφαλής χρήστη ενώ έγινε παράλειψη του εξωτερικού κουτιού που υπήρχε στο Developer Kit 1, το οποίο περιείχε κουμπιά για μεταβολή φωτεινότητας, αντίθεσης, ενώ χρησίμευε και για την ενεργοποίησης της συσκευής [17]. Αξίζει να αναφέρουμε ότι η κάμερα θέσεως του Oculus έχει εμβέλεια από 50 εκατοστά μέχρι και 2.5 μέτρα.





Τον Σεπτέμβριο του 2014 η εταιρεία προχώρησε στη δημιουργία του πρωτότυπου Crescent Bay το οποίο παρουσιάστηκε στο συνέδριο Oculus Connect στο Los Angeles. Σημαντικές βελτιώσεις έναντι του Developer Kit 2, ήταν η καλύτερη ανάλυση οθόνης, λιγότερο βάρος, ενσωματωμένος ήχος, όπως επίσης και ανίχνευση κίνησης 360 μοιρών η οποία επιτεύχθηκε με την ενσωμάτωση σε αυτή, λαμπάκια ανίχνευσης LED που τοποθετήθηκαν στο πίσω μέρος της συσκευής [17].

Ακόμη δεν έχει κάνει την εμφάνιση της μια έκδοση Oculus διαθέσιμη προς το κοινό, κάτι που περιλαμβάνεται στα μελλοντικά πλάνα της εταιρείας καθώς τόσο τα Developer Kit 1 και 2, όσο και τα πρωτότυπα παρουσίασαν σημαντική επιτυχία. Σύμφωνα με την εταιρεία Oculus, μια έκδοση αποκλειστικά για το κοινό είναι ήδη στο στάδιο κατασκευής περιλαμβάνοντας καλύτερη ανίχνευση κίνησης, υψηλότερη ανάλυση οθόνης αλλά και ασύρματης μεταφοράς εικόνας στην συσκευή αφαιρώντας τα περιττά καλώδια που συνδέουν την συσκευή με άλλες τεχνολογικές συσκευές.

Για τους σκοπούς της έρευνας προτιμήθηκε η χρήση του Developer Kit 2 έναντι του 1. Στα αρχικά στάδια χρησιμοποιήθηκε το Oculus Rift Developer Kit 1, αλλά στην συνέχεια προτιμήθηκε το Developer Kit 2 για τους λόγους που θα αναφερθούν πιο κάτω. Η ανάλυση του DK2 είναι στα 1920x1080 pixels έναντι των 1280x800pixels του DK1, ενώ η οθόνη είναι του τύπου OLED [18] αν και είναι μικρότερη 5,7’’ έναντι 7’’ κάτι που συνεπάγει σε καλύτερη ποιότητα εικόνας. Επίσης, ακόμη ένας σημαντικός παράγοντας είναι ο ρυθμός ανανέωσης εικόνας ο οποίος βρίσκεται στα 75Hz έναντι

60Hz. Τέλος, στο DK2 η παρουσία της κάμερας ανίχνευσης θέσεως που απουσιάζει από το DK1, αποτελεί σημαντικό παράγοντα για μελλοντική ανάπτυξη του συστήματος καθώς δίνει την δυνατότητα για νέες λειτουργίες [19, 20].

3.2.2 Εναλλακτικές Συσκευές

Ο κύριος λόγος επιλογής της συσκευής Oculus Rift έναντι εναλλακτικών αποτέλεσε και το γεγονός ότι το SDK (Software Development Kit), το οποίο βοηθά στην υλοποίηση κώδικα αλλά και στην κατανόηση των λειτουργιών του Oculus, ήταν διαθέσιμο σε ανοιχτό κώδικα. Εντούτοις, στο σημείο αυτό θα παρουσιαστούν ορισμένες εναλλακτικές συσκευές που μπορούν να αντικαταστήσουν σε μελλοντικά στάδια το Oculus Rift.



Μια από τις αναμενόμενες συσκευές είναι το Project Morpheus. Η συσκευή αυτή έκανε την πρώτη της εμφάνιση στην έκθεση GDC το 2015 και αναπτύσσεται αποκλειστικά για PlayStation 4 και PlayStation Vita. Επομένως, δεν θα αποτελέσει ισχυρό κίνητρο για προγραμματιστές εφαρμογών διαφορετικών συσκευών (υπολογιστές, κινητά τηλέφωνα, λοιπές κονσόλες), αλλά αναμένεται να κάνει αίσθηση στον κόσμο των παιχνιδιών. Μερικά από τα χαρακτηριστικά της συσκευής αυτής η οποία βρίσκεται ακόμη σε στάδιο κατασκευής και συνεχούς ελέγχου είναι: η ανάλυση της εικόνας που βρίσκεται στα 1920x1080pixels, η παρουσία ενσωματωμένου ήχου, ενώ το κυριότερο πλεονέκτημα της έναντι του Oculus Rift αποτελεί ο ρυθμός ανανέωσης εικόνας που βρίσκεται στα 120fps (frames per second / καρέ ανά δευτερόλεπτο) αντί των 75fps που έχει το Oculus Rift. Η συσκευή αναμένεται να κυκλοφορήσει εκτός απροόπτου γύρω στις αρχές του 2016 [21, 26].

Η συνεργασία Oculus Rift και Samsung επέφερε στο κοινό την ανάπτυξη συσκευής προβολής εικονικής πραγματικότητας με όνομα Gear VR. Κύριο μειονέκτημα της συσκευής είναι η υποχρεωτική παρουσία κινητού τηλεφώνου



Samsung και συγκεκριμένα Samsung Galaxy Note 4 ή Samsung Galaxy S6, το οποίο μπαίνει σε μια βάση στην πρόσοψη της συσκευής και βοηθά στην παρουσίαση της εικόνας προς τον χρήστη. Η συσκευή αποτελεί μια λύση για τους ανυπόμονους πελάτες οι οποίοι περιμένουν την κυκλοφορία του Oculus Rift, έχει χαρακτηριστικά παρόμοια με το Oculus Rift Developer Kit 1 που αναφέρθηκε προηγουμένως και είναι ήδη διαθέσιμη στο κοινό [22, 26].



Από τον χώρο της εικονικής πραγματικότητας δεν θα μπορούσε να λείπει και η γνωστή εταιρεία Google, η οποία τον Ιούνιο του 2014 στο συνέδριο Google I/O παρουσίασε τη δική της εκδοχή στον χώρο με το Google Cardboard. Η συσκευή μετατρέπεται σε οθόνη προβολής εικονικής πραγματικότητας με την τοποθέτηση κινητού τηλεφώνου στην πρόσοψη της συσκευής. Το θετικό του Google Cardboard είναι ότι μπορεί να κατασκευαστεί πολύ εύκολα από τον κάθε ενδιαφερόμενο εάν ακολουθήσει τον σχετικό οδηγό που βρίσκεται στην ιστοσελίδα της google [23, 26].



Η Microsoft στην πρόσφατη ανακοίνωση της για την νέα έκδοση του λειτουργικού συστήματος Windows 10, προχώρησε στην ανακοίνωση μιας πρωτοποριακής συσκευής προβολής εικονικής πραγματικότητας η οποία θα συνοδεύει τα Windows 10. Πρόκειται για την συσκευή Microsoft HoloLens η οποία δίνει στον χρήστη νέες δυνατότητες και νέες αλληλεπιδράσεις με την τεχνική ολογραμμάτων [24].



Όσον αφορά τις εναλλακτικές συσκευές προβολής εικονικής πραγματικότητας, μια ακόμα συσκευή η οποία αναμένεται να κάνει αίσθηση είναι το SteamVR ή αλλιώς HTC

Vive. Βρίσκεται σε ανάπτυξη από την εταιρεία κινητών τηλεφώνων HTC και την εταιρεία ανάπτυξης παιχνιδιών Valve, γνωστή στο κοινό από τα παιχνίδια Half-Life, Portal και Left 4 Dead. Τα χαρακτηριστικά της συσκευής αυτής επικεντρώνονται κυρίως στο ότι αυτή επιχειρεί να προσφέρει νέες εμπειρίες στον χώρο εικονικής πραγματικότητας, συμπεριλαμβανομένης της κίνησης του χρήστη σε μεγάλους χώρους. Για τον σκοπό αυτό η συσκευή θα διατίθεται με κάμερες οι οποίες θα βοηθούν την ανίχνευση κίνησης, ενώ θα διατίθενται και δύο χειριστήρια για καλύτερη αλληλεπίδραση και νέες δυνατότητες. Σημαντική πληροφορία αποτελεί ότι η συσκευή θα είναι διαθέσιμη δωρεάν για εγκεκριμένους προγραμματιστές προκειμένου να αναπτύξουν διάφορες εφαρμογές για την εν λόγω συσκευή, ενώ και το API του SteamVR θα είναι διαθέσιμο σε ανοιχτό κώδικα, σημαντικό στοιχείο του Oculus Rift [25, 26].

Τελειώνοντας την αναφορά στις συσκευές προβολής εικονικής πραγματικότητας, αξίζει να αναφέρουμε την συσκευή FOVE, η οποία είναι η πρώτη συσκευή προβολής εικονικής πραγματικότητας με ανίχνευση κίνησης ματιών. Η επαναστατική αυτή συσκευή αντλεί τη χρηματοδότηση της μέσω εκστρατείας προβολής του Kickstarter, ώστε να βγει στην αγορά. Όπως γίνεται αντιληπτό, με την ανίχνευση κίνησης ματιών, η εικονική πραγματικότητα γίνεται αυτομάτως προσβάσιμη σε ανθρώπους με κινητικές δυσλειτουργίες, αναπηρίες ή ακόμη και με ειδικές ικανότητες [36].



3.3 Συσκευές ανίχνευσης βάθους

3.3.1 Microsoft Kinect



Το Kinect for Windows είναι μια καινοτόμα συσκευή, η οποία έκανε την πρώτη της εμφάνιση στην αγορά το 2010. Το Kinect είναι μια συσκευή ανίχνευσης βάθους η οποία χρησιμοποιείται στις κονσόλες Xbox 360 και Xbox One, αλλά και σε ηλεκτρονικούς υπολογιστές, κυρίως για σκοπούς παιχνιδιών. Ωστόσο, κατά τη διάρκεια των τελευταίων χρόνων, στην οποία μελετήθηκαν περεταίρω οι δυνατότητες του αισθητήρα του Kinect, αρκετοί προγραμματιστές χρησιμοποίησαν την συσκευή για σκοπούς έρευνας ή για ανάπτυξη εφαρμογών για σκοπούς ψυχαγωγίας ή ακόμη και για ιατρικούς αλλά και εκπαιδευτικούς σκοπούς.

Η εταιρεία Microsoft κυκλοφόρησε το Kinect για πρώτη φορά το 2010 για την κονσόλα Xbox 360, αφού πρώτα το ανακοίνωσε στην έκθεση παιχνιδιών E3 το 2009, με την κωδική ονομασία Project Natal. Η συσκευή δημιουργήθηκε με κύριο σκοπό την χρησιμοποίηση της σε παιχνίδια της κονσόλας Xbox 360. Παρ' όλα αυτά όμως, μετά από περεταίρω έρευνα που έγινε σχετικά με τις δυνατότητες του αισθητήρα της συσκευής, η Microsoft προχώρησε στην ανάπτυξη νέας συσκευής ειδικά για ηλεκτρονικούς υπολογιστές Windows, έχοντας ως κύριο στόχο να επιταχύνουν την ανάπτυξη εφαρμογών. Η νέα συσκευή η οποία είχε παρόμοια χαρακτηριστικά με την προηγούμενη, με εξαίρεση μικρών διαφορών, συνοδευόταν με SDK (Software Development Kit) το οποίο βοηθούσε τους προγραμματιστές παρέχοντας τους τις δυνατότητες του αισθητήρα της συσκευής με μορφή κώδικα γραμμένου σε γλώσσες προγραμματισμού C#, C++ και Visual Basic. Μερικά από τα χαρακτηριστικά της πρώτης έκδοσης του SDK ήταν η ανάκτηση έγχρωμης εικόνας και εικόνας βάθους από τον αισθητήρα, ανίχνευση σκελετού και προηγμένες δυνατότητες ήχου [27].

Λόγω της μεγάλης επιτυχίας του Kinect για Xbox 360 και ειδικότερα λόγω του ενδιαφέροντος που έδειξαν οι προγραμματιστές, κυκλοφόρησαν νέες αναβαθμισμένες εκδόσεις του SDK οι οποίες έδιναν στους χρήστες νέες λειτουργίες αλλά παράλληλα βελτίωναν τις υφιστάμενες. Μία σημαντική συμβολή της Microsoft απετέλεσε η ανακοίνωση του νέου αισθητήρα Kinect, γνωστό και ως Kinect for Xbox One (V2) το οποίο κυκλοφόρησε στην αγορά στα τέλη του 2014. Μερικές από τις βελτιώσεις σε σχέση με τον προκάτοχο του ήταν η βελτιωμένη ανίχνευση σκελετού, όπως και η ανεπτυγμένη ανίχνευση κινήσεων και μορφασμών προσώπου κυρίως λόγω της αναβαθμισμένης ανάλυσης του αισθητήρα.



Για τους σκοπούς της έρευνας χρησιμοποιήσαμε το Kinect για Xbox One, για λόγους που θα επεξηγηθούν πιο κάτω. Αν και το Kinect για Xbox 360 ήταν ελαφρύτερο από το Kinect για Xbox One και η διασύνδεση ήταν πιο πρακτική, εντούτοις ο νέος αισθητήρας υπερτερούσε του προηγούμενου. Η ανάλυση της κάμερας βάθους ήταν επίσης μεγαλύτερη (512x424 pixels έναντι 320x240 pixels), ενώ και η ανάλυση της έγχρωμης κάμερας ήταν επίσης μεγαλύτερη (1920x1080 pixels – HD στα 30fps έναντι 640x480 pixels στα 30fps). Αξίζει να αναφέρουμε ότι η εμβέλεια της ανίχνευσης βάθους από τον αισθητήρα κυμαίνεται από 40 εκατοστά μέχρι και 4.5 μέτρα. Επίσης, σημαντικές βελτιώσεις παρατηρήθηκαν στην αύξηση του οπτικού πεδίου του αισθητήρα (οριζόντια και κάθετα) όσο και στην μετάβαση από το καλώδιο USB 2.0 σε USB 3.0 το οποίο συμβάλει στην μεταφορά περισσότερων δεδομένων σε πραγματικό χρόνο [28].

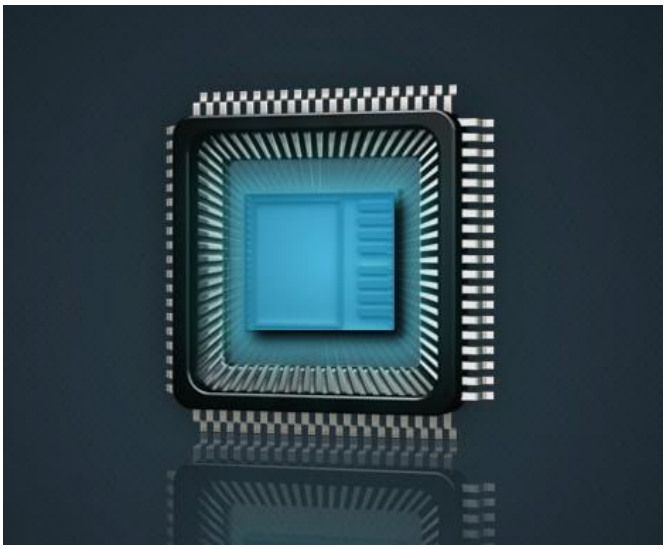
Ένα από τα κυριότερα χαρακτηριστικά του αισθητήρα Kinect, ήταν η παροχή άμεσης υποστήριξης για τις δυνατότητες του αισθητήρα μέσω του Microsoft Kinect SDK με την βοήθεια της μηχανής Unity 3D και διαφόρων plugins. Αυτό δεν καθίστατο δυνατό με την προηγούμενη έκδοση του Kinect, καθώς απαιτούσε την χρήση δευτερευόντων βιβλιοθηκών και διαφόρων πλαισίων (OpenNI) προκειμένου ο προγραμματιστής να έχει πρόσβαση στον αισθητήρα μέσω κώδικα με την βοήθεια του Unity3D.

3.3.2 Εναλλακτικές συσκευές ανίχνευσης βάθους



Ο αισθητήρας βάθους SENTIS TOF – P510 κατασκευάστηκε από την εταιρεία BlueTechnix Products και αποτελεί συνδυασμό τρισδιάστατου και δυσδιάστατου αισθητήρα βάθους. Η ανάλυση του αισθητήρα βρίσκεται στα 160x120 pixels ενώ η ανάλυση της εικόνας βρίσκεται στα 1080p (5 MP). Ο αισθητήρας μεταφέρει πληροφορίες βάθους και ασπρόμαυρη εικόνα για κάθε pixel του χώρου καταγραφής μέσω των δύο αισθητήρων. Επομένως η ανάλυση των δεδομένων βάθους γίνεται σε συνδυασμό με τα δεδομένα του δυσδιάστατου αισθητήρα [29].

Η εταιρεία PMD Intelligence, παρουσίασε τη δική της εκδοχή στον χώρο των αισθητήρων βάθους, με την τεχνολογία Time-of-Flight (ToF). Ο αισθητήρας PhotonlCs 19k-S3 είναι ένα τσιπ που μπορεί να τοποθετηθεί σε οποιαδήποτε κάμερα προκειμένου να δώσει δεδομένα βάθους αλλά και ασπρόμαυρη εικόνα για κάθε pixel του χώρου καταγραφής. Η ανάλυση του αισθητήρα βρίσκεται επίσης στα 160x120 pixels, ενώ η ποιότητα ανίχνευσης βάθους δεν επηρεάζεται από τον φωτισμό του χώρου στον οποίο γίνεται ο υπολογισμός του βάθους [30].



Αισθητήρες βάθους κατασκευάστηκαν και από άλλες εταιρείες με σχετικά παραδείγματα τους αισθητήρες OPT8140 και MLX75023 από την εταιρεία SoftKinetic, αλλά και τον mvBlueFox που μπορεί να χρησιμοποιηθεί με τον μίνι-υπολογιστή με λειτουργικό σύστημα Linux, γνωστό ως Raspberry Pi [31, 32].

Όμως μια συσκευή η οποία μπορεί όντως να αντικαταστήσει το Microsoft Kinect, είναι η κάμερα Creative Senz3D, η οποία χρησιμοποιήθηκε και κατά την ανακατασκευή χώρου με επαύξηση εικονικών αντικειμένων που αναφέρθηκε στο κεφάλαιο 2 στο μέρος 2.3. Η κάμερα υποστηρίζει μεταξύ άλλων τρισδιάστατη ανίχνευση κινήσεων χεριού αλλά και ανίχνευση προσώπου ενώ η ανάλυση της βρίσκεται στα 1280x720 pixels με ανάλυση βίντεο στα 720p στα 30fps με διασύνδεση USB 2.0 [10].



Το κύριο πλεονέκτημα των προαναφερθέντων αισθητήρων βάθους είναι το πρακτικό τους μέγεθος, ενώ μπορούν να αποδώσουν ανάλογα αποτελέσματα με το Microsoft Kinect. Εντούτοις η συσκευή Creative Senz3D υπερτερεί έναντι των υπολοίπων καθώς χρησιμοποιήθηκε κατά την έρευνα που παρουσιάσθηκε στο δεύτερο κεφάλαιο, οπόταν και τα αποτελέσματα που προέκυψαν ήταν αρκετά ακριβή.

3.4 Κάμερες



3.4.1 Logitech Cameras C310

Όπως αναφέρθηκε και προηγουμένως, για τους σκοπούς της έρευνας τοποθετήσαμε στην πρόσοψη της συσκευής Oculus Rift τις κάμερες C310 από την Logitech [4]. Οι κάμερες χρησιμοποιούνται για την μεταφορά εικόνας σε πραγματικό χρόνο στην οθόνη του Oculus Rift μέσω διασύνδεσης USB που χρησιμοποιείται για την μεταφορά δεδομένων από τις κάμερες στον υπολογιστή. Τα δεδομένα μεταφέρονται στην συνέχεια από τον υπολογιστή στο Oculus Rift μέσω της εφαρμογής που δημιουργήθηκε.

Οι κάμερες τοποθετήθηκαν στην πρόσοψη του Oculus Rift σε απόσταση μεταξύ τους προκειμένου να αποδώσουν μια στερεοσκοπική τρισδιάστατη εικόνα στον χρήστη,

ώστε η εικόνα να είναι όσο το δυνατό ρεαλιστική [35]. Παράλληλα η τρισδιάστατη εικόνα θα πρέπει να αντιπαραβάλλεται με την οθόνη του Oculus Rift ώστε η όραση του χρήστη να είναι όσο το δυνατό ομαλή. Επομένως, τα τεχνικά χαρακτηριστικά των καμερών θα πρέπει να ταιριάζουν στα τεχνικά χαρακτηριστικά του Oculus Rift ώστε να δώσουν την καλύτερη εμπειρία, όπως αναφέρθηκε στο κεφάλαιο 2 (μέρος 2.5).

Η επιλογή των καμερών έγινε βάσει της έρευνας που αναλύθηκε στο μέρος 2.5 του δεύτερου κεφαλαίου της διπλωματικής εργασίας. Επομένως, τα τεχνικά χαρακτηριστικά των καμερών έπρεπε να ταυτίζονται ή να ξεπερνούν μερικώς τα τεχνικά χαρακτηριστικά του Oculus Rift. Μερικά από τα τεχνικά χαρακτηριστικά των καμερών που χρησιμοποιήθηκαν ήταν η ανάλυση εικόνας στα 1280x720 pixels αλλά και η διεπαφή USB 2.0 για υποστήριξη μεταφοράς δεδομένων, μιας και η διεπαφή USB 3.0 δεν παρουσιάστηκε ακόμη στις κάμερες. Τα υπόλοιπα χαρακτηριστικά τα οποία πρέπει να έχουν οι κάμερες σε σχέση με την συσκευή προβολής εικονικής πραγματικότητας αναφέρθηκαν αναλυτικά στο 5^{ο} μέρος του 2^{ου} κεφαλαίου.

3.4.2 Εναλλακτικές κάμερες

Η επιλογή των καμερών που θα τοποθετηθούν στο Oculus Rift, θα πρέπει να γίνει βάσει των χαρακτηριστικών της οθόνης του Oculus Rift, για την ομαλότερη αλληλεπίδραση του χρήστη με την συσκευή, έτσι ώστε να μην προκύψουν προβλήματα κατά την όραση. Επομένως, σύμφωνα με την έρευνα που έγινε στην συγκεκριμένη διπλωματική εργασία, κατά την οποία επιλέχθηκε το Oculus Rift DK2, τα τεχνικά χαρακτηριστικά των καμερών θα πρέπει να αντιπαραβάλλουν τα τεχνικά χαρακτηριστικά του Oculus Rift.

Ως εκ τούτου, οποιαδήποτε κάμερα με ανάλυση μεγαλύτερη ή ίση της ανάλυσης της οθόνης του Oculus Rift, όπως και με μεγαλύτερο ή ίσο οπτικό πεδίο (field of view) μπορεί να χρησιμοποιηθεί για τον σκοπό που αναφέραμε προηγουμένως. Επιπρόσθετα σημαντικά χαρακτηριστικά αποτελούν τόσο ο ρυθμός ανανέωσης εικόνας όσο και η υποστήριξη διεπαφής USB 2.0/3.0 για την μεταφορά δεδομένων σε πραγματικό χρόνο.



Συνοπτικά αναφέρουμε μερικές από τις κάμερες που μπορούν να χρησιμοποιηθούν για να αντικαταστήσουν σε μελλοντικό χρόνο τις κάμερες που χρησιμοποιήθηκαν ήδη. Μια εξ' αυτών είναι η IDS uEye USB 3.0 camera, η οποία ανάλογα με το μοντέλο μπορεί να προσφέρει ρυθμό εγγραφής μέχρι και 169 fps (frames per second/ καρέ ανά δευτερόλεπτο). Άλλη επιλογή για κάμερα αποτελεί η PS3 Eye Camera η οποία μπορεί να προσφέρει ρυθμό εγγραφής στα 60 fps, αλλά με πολύ λιγότερη ανάλυση εικόνας. Κάθε κάμερα έχει θετικά αλλά και αρνητικά στοιχεία, επομένως η επιλογή της κάμερας θα πρέπει να γίνει με γνώμονα την αποφυγή οποιονδήποτε προβλημάτων όρασης κατά την αλληλεπίδραση του χρήστη με την εφαρμογή.



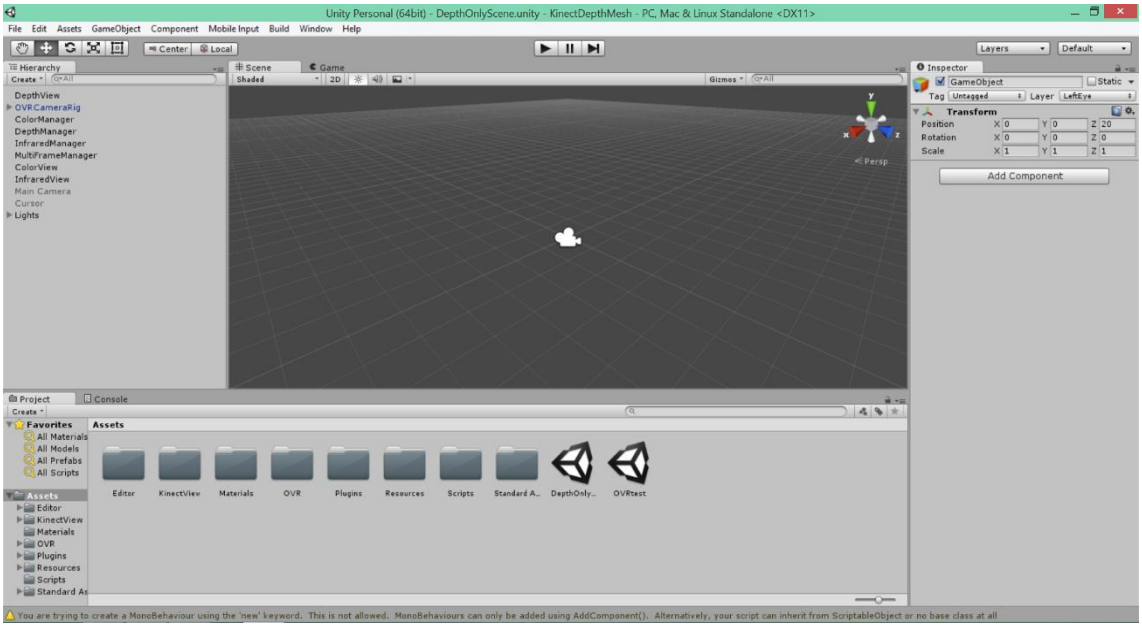
3.5 Επιπλέον εργαλεία που χρησιμοποιήθηκαν

3.5.1 Unity 3D

Η εφαρμογή που χρησιμοποιήθηκε για να υποστηρίξει την λειτουργικότητα του συστήματος, αναπτύχθηκε στο λογισμικό Unity 3D. Το Unity αποτελεί μια μηχανή ανάπτυξης παιχνιδιών για διάφορες πλατφόρμες και δημιουργήθηκε από την Unity Technologies. Μερικές από τις πλατφόρμες ανάπτυξης είναι ηλεκτρονικοί υπολογιστές, κονσόλες παιχνιδιών, έξυπνα κινητά τηλέφωνα αλλά και ιστοσελίδες. Η μηχανή ανάπτυξης παιχνιδιών υποστηρίζει προγραμματισμό σε γλώσσες προγραμματισμού JavaScript, C# και Boo [33, 34].

Ένα από τα μειονεκτήματα του Unity ήταν ότι προκειμένου να γίνει αξιοποίηση όλων των προχωρημένων λειτουργιών από τους προγραμματιστές, η premium έκδοση του Unity (Unity pro) θα έπρεπε να αγοραστεί, λόγω των μειωμένων δυνατοτήτων της δωρεάν έκδοσης. Αυτό όμως άλλαξε άρδην τον περασμένο Μάρτιο του 2015, καθώς η εταιρεία αποφάσισε η δωρεάν $5^{η}$ έκδοση της μηχανής ανάπτυξης (Unity 5.0), να περιλαμβάνει αντίστοιχα όλα τα χαρακτηριστικά και λειτουργίες που ενσωματώνονται στην premium έκδοση. Η νέα έκδοση παρέχει νέες δυνατότητες στον χώρο ανάπτυξης παιχνιδιών και εφαρμογών υποστηρίζοντας 21 διαφορετικές πλατφόρμες. Επομένως,

προκειμένου να αξιοποιηθούν οι δυνατότητες της νέας έκδοσης της μηχανής ανάπτυξης, αλλά κυρίως της άμεσης υποστήριξης μεταξύ Unity και Oculus Rift, αλλά και Unity και Kinect, η ανάπτυξη της εφαρμογής της παρούσας διπλωματικής εργασίας έγινε στο Unity 5.0.

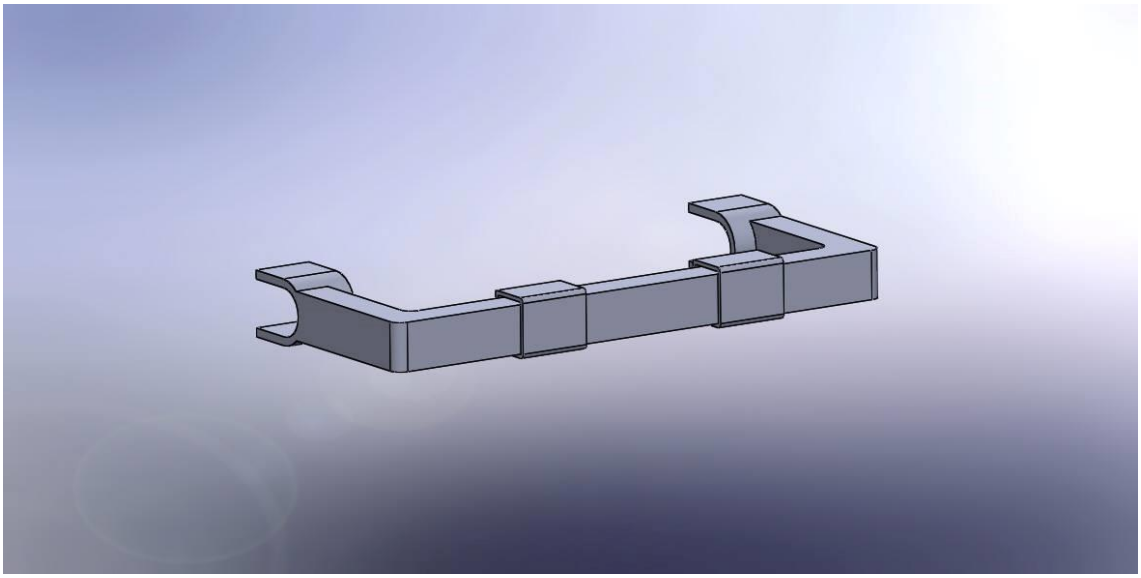


Σχήμα 3.1: Unity 5 Interface

3.5.2 Custom Camera Rig

Για την στήριξη των δύο καμερών Logitech, αναπτύχθηκε στο λογισμικό Solidworks ένα custom camera rig, δηλαδή μια βάση για δύο κάμερες. Όπως φαίνεται στο πιο κάτω σχήμα η βάση των καμερών περιλαμβάνει το βασικό στήριγμα όπως και δύο μεταβλητά στηρίγματα, ένα για κάθε κάμερα. Τα μεταβλητά στηρίγματα χρησιμοποιήθηκαν έτσι ώστε ο κάθε χρήστης να μπορεί να μεταβάλει την απόσταση μεταξύ των καμερών, καθώς η απόσταση των ματιών κάθε ανθρώπου είναι διαφορετική.

Με την προσαρμογή της σωστής απόστασης των καμερών, ο χρήστης θα μπορεί να δει την στερεοσκοπική εικόνα χωρίς ανεπιθύμητα αποτελέσματα. Οι κάμερες που τοποθετήθηκαν στα μεταβλητά στηρίγματα έγιναν stripped-down δηλαδή από τις κάμερες πήραμε μόνο τις πλακέτες αφαιρώντας το εξωτερικό κάλυμμα και την βάση της κάμερας, για σκοπούς εργονομίας αλλά και λιγότερου βάρους.



Σχήμα 3.2: Custom-made Camera Rig

Κεφάλαιο 4

Υλοποίηση

4.1 Διασύνδεση 36
4.2 Αντικείμενα που αποτελούν το σύστημα 38
4.3 Ανταλλαγή και επεξεργασία δεδομένων 39
4.4 Ανάλυση μεθοδολογίας και αλγόριθμοι 41
4.4.1 Ανίχνευση βάθους και δημιουργία πλέγματος ως αναφορά 41
4.4.2 Μεταφορά ζωντανής εικόνας από τις κάμερες στο Oculus Rift 43
4.4.3 Raycast και τοποθέτηση εικονικών αντικειμένων 44
4.4.4 Επιπρόσθετες διεργασίες για σωστή λειτουργία 46

4.1 Διασύνδεση

Η διασύνδεση που δημιουργήθηκε κατά την υλοποίηση εμφανίζεται στην πιο κάτω εικόνα και περιγράφεται αναλυτικά ως εξής: Ο χρήστης αλληλεπιδρά μέσω ενός υπολογιστή με μια εφαρμογή που δημιουργήθηκε στην πλατφόρμα Unity3D. Οι συσκευές Oculus Rift, Logitech Cameras και Microsoft Kinect πρέπει να είναι συνδεδεμένες στον υπολογιστή του χρήστη, μέσω διασύνδεσης USB, η οποία χρησιμοποιείται τόσο για την τροφοδότηση των συσκευών όσο και για την μεταφορά δεδομένων από τις συσκευές στον υπολογιστή.

Στη συνέχεια ο χρήστης τοποθετεί στο κεφάλι του το Oculus Rift, στην πρόσοψη του οποίου βρίσκονται τοποθετημένες οι δύο κάμερες. Επίσης, επάνω στο Oculus βρίσκεται ήδη τοποθετημένος ο αισθητήρας Microsoft Kinect. Λόγω του ότι η διασύνδεση που εφαρμόστηκε είναι λιγότερο πρακτική κυρίως λόγω του βάρους αλλά και μεγέθους του Kinect, εντούτοις η έρευνα που έγινε δεν στόχευε στην ευχρηστία της διασύνδεσης αλλά στην αλληλεπίδραση των συσκευών μεταξύ τους, ώστε να επιτευχθεί ο αρχικός

στόχος που τέθηκε στη διπλωματική. Γι’ αυτό το λόγο στο κεφάλαιο 3 αναφέρθηκαν εναλλακτικές συσκευές που θα μπορούσαν να αντικαταστήσουν τις υφιστάμενες ώστε η διασύνδεση να γίνει πιο εργονομική και πιο πρακτική, για την καλύτερη αλληλεπίδραση του χρήστη.



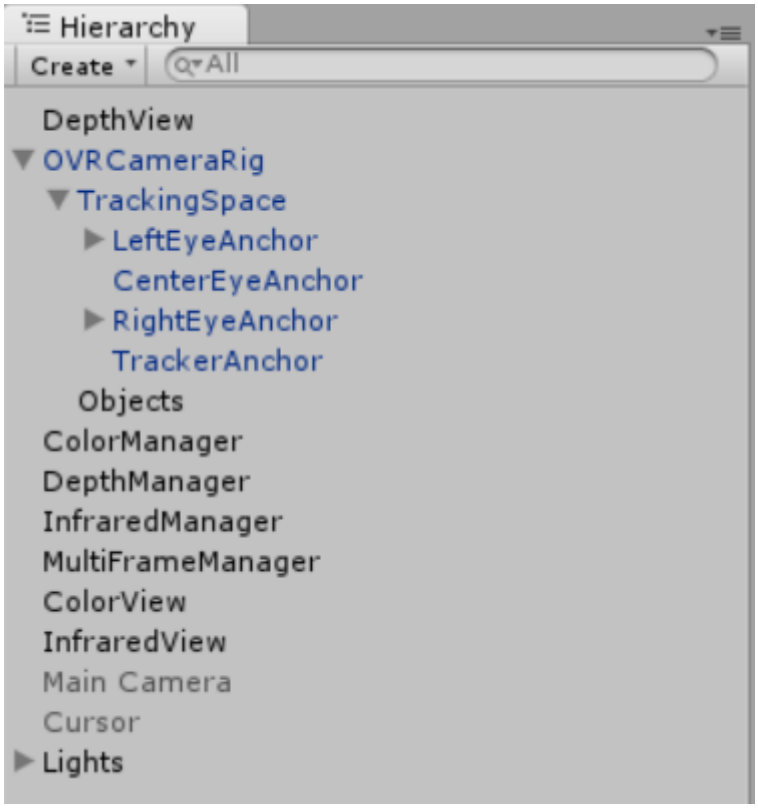
Εικόνα 4.1 Διασύνδεση

Επισημαίνεται ότι οι συσκευές που αποτελούν τη διασύνδεση έχουν αριθμηθεί από το 1 μέχρι 4 και είναι οι ακόλουθες:

1) Kinect for Windows v2
2) IR Positional Camera
3) Oculus Rift DK2
4) Logitech Cameras C310 (stripped down)

4.2 Αντικείμενα που αποτελούν το σύστημα

Όπως φαίνεται στο πιο κάτω σχήμα τα μέρη που αποτελούν το σύστημα το οποίο δημιουργήθηκε στην μηχανή ανάπτυξης Unity 3D, είναι τα ακόλουθα:



Σχήμα 4.1: Ιεραρχία αντικειμένων που αποτελούν το σύστημα

Το αντικείμενο `OVRCameraRig` αναφέρεται στην οθόνη του Oculus Rift. Με την αναβάθμιση του Oculus Rift SDK για Unity στην έκδοση 0.6.0.0, ο τρόπος εντοπισμού των κινήσεων της κεφαλής του χρήστη (tracking), αλλά και η τροφοδότηση εικόνας στο Oculus Rift έχει διαφοροποιηθεί. Το αντικείμενο που χρησιμοποιείται για το tracking, είναι το `TrackingSpace` το οποίο περιλαμβάνει 5 αντικείμενα ως παιδιά του. Τα `LeftEyeAnchor` και `RightEyeAnchor` αναφέρονται στην τοποθέτηση του αριστερού και δεξιού ματιού αντίστοιχα, ενώ το `CenterEyeAnchor` συμπίπτει με τον μέσο όρο της στάσης του αριστερού και του δεξιού ματιού. Το `TrackerAnchor` αφορά την τοποθεσία της συσκευής του Oculus σε σχέση με την κάμερα θέσεως που είναι διαθέσιμη μόνο στο Oculus Rift DK2 (IR Positional Camera). Τέλος το αντικείμενο `Objects` είναι υπεύθυνο για την τοποθέτηση και ομαδοποίηση των εικονικών αντικειμένων που τοποθετούνται δυναμικά στην σκηνή.

Το DepthView είναι το πλέγμα (mesh), το οποίο θα χρησιμοποιηθεί για την ανίχνευση της γεωμετρίας του χώρου. Το πλέγμα δημιουργείται κατά τον χρόνο εκτέλεσης του προγράμματος με την βοήθεια των scripts που θα αναφερθούν εκτενέστερα στο μέρος 4 του κεφαλαίου αυτού. Για την ομαλή λειτουργία του raycast το οποίο τοποθετεί εικονικά αντικείμενα στο πλέγμα, τα συστατικά στοιχεία MeshCollider και MeshFilter χρειάζονται, ώστε να αναγνωρίζονται οι συγκρούσεις μεταξύ του πλέγματος και των εικονικών αντικειμένων που τοποθετούνται σε αυτό.

Τα αντικείμενα ColorManager, DepthManager, InfraredManager και MultiFrameManager χρησιμοποιούνται για την μεταφορά εικόνων σε πραγματικό χρόνο (έγχρωμη εικόνα, εικόνα βάθους, υπέρυθρη εικόνα κλπ.), από το Microsoft Kinect στο σύστημα. Οι εικόνες αυτές χρησιμοποιούνται έπειτα για σκοπούς ελέγχου όπου και η έγχρωμη εικόνα αλλά και η υπέρυθρη εικόνα (μαυρόασπρη) εμφανίζονται αντίστοιχα στα αντικείμενα ColorView και InfraredView.

Τα αντικείμενα LeftQuad και RightQuad αφορούν την μεταφορά εικόνας σε πραγματικό χρόνο από τις δύο κάμερες που είναι τοποθετημένες στο Oculus Rift, στο ίδιο το Oculus Rift υπό την μορφή στερεοσκοπικής εικόνας. Συγκεκριμένα τα δύο αντικείμενα (GameObjects) είναι παιδιά των LeftEyeAnchor και RightEyeAnchor αντίστοιχα, λόγω του ότι γίνονται render σε κάθε μάτι ξεχωριστά. Επομένως το CenterEyeAnchor αποδίδει ι την ένωση των δύο όπως αναφέρθηκε προηγουμένως, άρα η εικόνα θα είναι στερεοσκοπική [35].

4.3 Ανταλλαγή και επεξεργασία δεδομένων

Η ανταλλαγή και επεξεργασία δεδομένων επιτυγχάνεται με την βοήθεια της μηχανής ανάπτυξης Unity3D. Τα δεδομένα μεταφέρονται από την κάθε συσκευή στον υπολογιστή μέσω της διασύνδεσης με USB, όπου τυγχάνουν περεταίρω επεξεργασίας προκειμένου να χρησιμοποιηθούν αργότερα για την λύση του προβλήματος. Η διαδικασία επεξεργασίας των δεδομένων που λαμβάνονται περιγράφεται στην συνέχεια, όπως επίσης και ο τρόπος αλληλεπίδρασης των συσκευών μεταξύ τους. Παράλληλα ακολουθεί ο πίνακας 4.1 στον οποίο παρατίθενται τα βήματα που εκτελούνται μέσω της

εφαρμογής σε πραγματικό χρόνο, όπως και τα scripts που απαιτούνται για την εκτέλεσή τους.

Πίνακας 4.1: Βήματα εκτέλεσης σε πραγματικό χρόνο

Βήματα εκτέλεσης	**Scripts που χρησιμοποιούνται για την εκτέλεση**	**Συσκευές που χρησιμοποιούνται**
1. Depth Capture	DepthSourceManager.cs DepthSourceView.cs	Kinect for Windows v2/ Microsoft Kinect v2
2. Mesh Creation using Depth	DepthSourceView.cs	-
3. Render Camera Feed on Quads	WebcamTextureScript.cs	Logitech Cameras C310
4. Display Quads on Oculus with the created mesh	-	Oculus Rift DK2
5. RayCast on created mesh	Raycast.cs	Oculus Rift DK2
6. Spawn Objects on hit vector	ObjectSpawner.cs SpawnedObject.cs PositionalCamera.cs	-

Το Microsoft Kinect V2 χρησιμοποιείται όπως αναφέρθηκε προηγουμένως για την ανίχνευση βάθους στον χώρο αλληλεπίδρασης. Η ανίχνευση βάθους γίνεται ώστε να επιτευχθεί η επίγνωση του χώρου αλληλεπίδρασης από την εφαρμογή. Τα δεδομένα βάθους υπολογίζονται σε πραγματικό χρόνο και λαμβάνονται από την εφαρμογή χρησιμοποιώντας διάφορες μεθόδους από το Microsoft Kinect SDK το οποίο υποστηρίζεται από το Unity3D.

Τα δεδομένα βάθους μετατρέπονται στην συνέχεια σε τρισδιάστατο πλέγμα (mesh) μέσω των αλγορίθμων που θα παρουσιαστούν αναλυτικά στο επόμενο μέρος του κεφαλαίου. Το πλέγμα που δημιουργείται αντικατοπτρίζει τον χώρο αλληλεπίδρασης

που είναι ορατό από την συσκευή και κατ' επέκταση από τον χρήστη. Επομένως, το πλέγμα θα χρησιμοποιηθεί ως αναφορά για την τοποθέτηση εικονικών αντικειμένων στον χώρο. Λόγω του ότι ο χρήστης έχει τη δυνατότητα κίνησης μέσα στον χώρο αλληλεπίδρασης, στρέφοντας το κεφάλι του προς όλες τις επιθυμητές κατευθύνσεις, τα δεδομένα βάθους και επομένως το πλέγμα που δημιουργείται αλλάζουν συνεχώς σε πραγματικό χρόνο οπόταν και επιτυγχάνεται η επίγνωση ολόκληρου του χώρου αλληλεπίδρασης.

Οι δύο κάμερες που είναι τοποθετημένες στην πρόσοψη του Oculus Rift μεταφέρουν την εικόνα του χώρου αλληλεπίδρασης στην οθόνη του Oculus Rift. Οι δύο κάμερες που είναι τοποθετημένες σε μια δεδομένη απόσταση μεταξύ τους μπορούν να τύχουν προσαρμογής, ανάλογα με την απόσταση των ματιών του χρήστη. Η ύπαρξη δύο καμερών σε συνδυασμό με μεθόδους και τεχνικές του Oculus Rift που θα αναφερθούν στο επόμενο μέρος του κεφαλαίου, δημιουργούν μια στερεοσκοπική εικόνα η οποία είναι ορατή μέσω του Oculus, η οποία είναι όσο το δυνατό ρεαλιστική [35].

4.4 Ανάλυση μεθοδολογίας και αλγόριθμοι

4.4.1 Ανίχνευση βάθους και δημιουργία πλέγματος ως αναφορά

Η δημιουργία του τρισδιάστατου πλέγματος από τα δεδομένα βάθους του Kinect, έγινε με την μέθοδο της τριγωνοποίησης. Συνοπτικά τα δεδομένα βάθους λαμβάνονται από τον αισθητήρα του Kinect όπου και φυλάσσονται υπό την μορφή μονοδιάστατου πίνακα. Έπειτα τα δεδομένα του μονοδιάστατου πίνακα αναλύονται όπου με την βοήθεια της τριγωνοποίησης του πίνακα δημιουργείται το τρισδιάστατο πλέγμα.

Η διαδικασία της τριγωνοποίησης περιλαμβάνει υπολογισμό των κορυφών των τριγώνων του πλέγματος (vertices – x,y,z), οι οποίες θα αντικατοπτρίζουν την τοποθεσία κάθε τριγώνου στο πλέγμα, όπως και υπολογισμό των UVs (άξονες των textures του τριγώνου). Τέλος, γίνεται ο υπολογισμός του κάθε τριγώνου που περιλαμβάνεται στο πλέγμα, οπόταν και η ένωση όλων των τριγώνων μεταξύ τους δίνει το τελικό πλέγμα.

Κατά την υλοποίηση του πιο πάνω χρησιμοποιήθηκαν δύο scripts, `DepthSourceManager.cs` και `DepthSourceView.cs`, που παρουσιάζονται αναλυτικά στο Παράρτημα Α.



Σχήμα 4.2: Scripts για δημιουργία δυναμικού πλέγματος

Το πρώτο script `DepthSourceManager.cs` χρησιμοποιεί το δεύτερο ως συμπληρωματικό για την διεκπεραίωση των εργασιών που γίνονται σε αυτό. Περιέχει την `public` μεταβλητή `ViewMode` που αντιπροσωπεύει τον τρόπο με τον οποίο τα δεδομένα συλλέγονται από τον αισθητήρα του Kinect. Οι επιλογές είναι το `SeperateSourceReaders` όπου τα δεδομένα συλλέγονται σε διαφορετικούς `readers` και το `MultiSourceReader` όπου τα δεδομένα συλλέγονται από ένα μόνο `reader`. Οι επιλογές αυτές είναι προσβάσιμες μέσω του αντικειμένου `DepthViewMode`.

Κατά την αρχικοποίηση στο script `DepthSourceManager` που γίνεται με την μέθοδο `void Start()` όπως γίνεται και σε κάθε unity script, καλείται η μέθοδος `CreateMesh(width, height)`, όπου γίνεται η αρχικοποίηση του πλέγματος με την βοήθεια παραμέτρων width και height, που αποτελούν το μέγεθος του πλέγματος. Κατά την αρχικοποίηση του πλέγματος επιτυγχάνεται και η προγραμματιστική τοποθέτηση των `MeshCollider` και `MeshFilter`, στοιχεία που όπως προαναφέραμε σε προηγούμενο μέρος του κεφαλαίου είναι απαραίτητα για την ανίχνευση συγκρούσεων. Με το πέρας της αρχικοποίησης του πλέγματος, γίνεται η εκκίνηση του αισθητήρα της συσκευής Kinect με την βοήθεια του `_Sensor.Open()`.

Με την μέθοδο `void Update()`, η οποία εκτελείται σε κάθε frame κατά τον χρόνο εκτέλεσης όπως γίνεται σε όλα τα unity scripts, λαμβάνονται τα δεδομένα από το Kinect. Τα δεδομένα βάθους όπως και οι εικόνες σε πραγματικό χρόνο από την κάμερα και τον αισθητήρα του Kinect, λαμβάνονται με την βοήθεια του δεύτερου script που σημειώθηκε πιο πάνω `DepthSourceView.cs`. Έπειτα καλείται η μέθοδος

`RefreshData()` στην οποία τα δεδομένα βάθους αποθηκεύονται σε ένα μονοδιάστατο πίνακα τύπου `ushort`, όπου κάθε πεδίο του πίνακα αποτελεί και ένα pixel από τον χώρο καταγραφής. Παράλληλα, το πλέγμα ανανεώνεται σε κάθε frame κατά τη διάρκεια του χρόνου εκτέλεσης με βάση τα δεδομένα βάθους που λαμβάνονται. Η δημιουργία του πλέγματος κατά τη διάρκεια του χρόνου εκτέλεσης γίνεται με την μέθοδο της τριγωνοποίησης που σημειώθηκε πιο πάνω. Η διαδικασία δημιουργίας του πλέγματος καθώς και της ανανέωσης του σε κάθε frame, εμφανίζεται στο Παράρτημα Α, όπου σε αυτό περιέχεται και ο πηγαίος κώδικας.

4.4.2 Μεταφορά ζωντανής εικόνας από τις κάμερες στο Oculus Rift

Η μεταφορά της ζωντανής εικόνας από τις δύο κάμερες που είναι τοποθετημένες στην πρόσοψη του Oculus Rift, γίνεται όπως αναφέρθηκε στο πρώτο μέρος αυτού του κεφαλαίου μέσω των αντικειμένων `RightQuad` και `LeftQuad` με την βοήθεια του `WebcamTextureScript`. Συγκεκριμένα κάθε `Quad` (texture), τοποθετείται σε κάθε `EyeAnchor` του Oculus, `left` και `right`, ώστε να γίνεται render σε κάθε κάμερα ξεχωριστά. Το `CenterEyeAncor` κατά την απόδοση θα ενώσει την εικόνα των δύο καμερών και επομένως αφού η φυσική θέση των δύο καμερών είναι σε κοντινή απόσταση, η τελική εικόνα που θα εμφανιστεί στο Oculus θα είναι στερεοσκοπική [35].

Κατά την αρχικοποίηση η οποία εκτελείται στο πιο πάνω script, αρχικοποιούνται τα στοιχεία των δύο καμερών στα οποία συμπεριλαμβάνονται το μήκος, πλάτος και το όνομα. Παράλληλα για λόγους ελαχιστοποίησης απωλειών και για επίτευξη ομαλότερης εικόνας με τη καλύτερη δυνατή ευκρίνεια απαιτήθηκε όπως τα απαιτούμενα fps(frames per second) τέθηκαν στο 100.. Τέλος, γίνεται περιστροφή των καμερών κατά 90° ώστε να επιτυγχάνεται μέγιστος βαθμός ανταπόκρισης τους στη φυσική θέση των καμερών η οποία τυγχάνει επίσης περιστροφής κατά 90° λόγω περιορισμένου χώρου.

4.4.3 Raycast και τοποθέτηση εικονικών αντικειμένων

Για την τοποθέτηση εικονικών αντικειμένων στο πλέγμα που δημιουργείται και κατά



συνέπεια στον εικονικό χώρο που είναι ορατός από το Kinect, χρησιμοποιήθηκαν τα scripts `ObjectSpawner.cs` και `Raycast.cs`.

Σχήμα 4.3: Scripts για τοποθέτηση αντικείμενων στον εικονικό χώρο

α). Το πρώτο script χρησιμοποιείται για τη δημιουργία των αντικειμένων σε πραγματικό χρόνο, ενώ το δεύτερο επικεντρώνεται με την ανίχνευση συγκρούσεων αλλά και την ορθή τοποθέτηση των αντικειμένων στον χώρο. Αναλυτικά με το πρώτο script `ObjectSpawner.cs`, δημιουργείται το ανάλογο αντικείμενο το οποίο πρόκειται να τοποθετηθεί στο εικονικό περιβάλλον ενώ προσδιορίζεται και η θέση του στον χώρο υπό την μορφή Vector3 (x,y,z). Παράλληλα με βάση την παραμετροποίηση και τις τιμές των παραμέτρων που λαμβάνονται από το script `Raycast.cs`, με το κάλεσμα του προηγούμενου script, προσδιορίζεται και η συμπεριφορά του δημιουργημένου αντικειμένου. Συγκεκριμένα ανάλογα με τις παραμέτρους, το αντικείμενο μπορεί να παραμένει σταθερό στην θέση που τοποθετείται ή σε αντίθετη περίπτωση αποκτά βαρύτητα οπόταν αυτό μετατρέπεται σε δυναμικό.

β). Το δεύτερο script `PostionalCamera.cs,` χρησιμοποιείται για την σωστή συμπεριφορά του Oculus σε σχέση με την positional camera, η οποία ανιχνεύει τις κινήσεις του Oculus, και συγκεκριμένα κινήσεις μπροστά, πίσω, πάνω, κάτω και δεξιά ή αριστερά. Το script αυτό βρίσκει την θέση του Oculus σε σχέση με την κάμερα όπως και την θέση των εικονικών αντικειμένων στο χώρο. Έπειτα μεταβάλλει την θέση των αντικειμένων έτσι ώστε να είναι ανάλογη με την θέση του Oculus. Επομένως υπάρχει η έννοια του βάθους, όπως θα φανεί και στις οθόνες αποτελεσμάτων 5.1, καθώς με την μετακίνηση του Oculus κατά μήκος της κάμερας θέσεως, ο χρήστης αντιλαμβάνεται ότι πλησιάζει στην θέση των εικονικών αντικειμένων ή αντίστοιχα απομακρύνεται από αυτά.

γ). Το τρίτο script που ονομάζεται `Raycast.cs`, έχει την αποκλειστική ευθύνη για την ομαλή λειτουργία του προγράμματος αφού μέσα από αυτό εκτελούνται οι βασικές λειτουργίες του προγράμματος. Στο script χρησιμοποιείται η μέθοδος `raycast` κατά την οποία υποστηρίζεται η δυνατότητα αναγνώρισης συγκρούσεων μεταξύ δύο αντικειμένων που σημειώνονται ως αντικείμενα αναφοράς. Ειδικότερα στην περίπτωση αυτή τα σημεία που σημειώνονται ως αντικείμενα αναφοράς αυτά είναι το

`CenterEyeAnchor`, που όπως έχει λεχθεί προηγουμένως αντικατοπτρίζει τον μέσο όρο της τοποθέτησης των δύο ματιών του χρήστη, και του πλέγματος που δημιουργείται.

Η σύγκρουση είναι εφικτή με τον χρήστη να εστιάζει με το κεφάλι στο σημείο όπου θέλει να τοποθετήσει το εικονικό αντικείμενο. Ο χρήστης έπειτα πρέπει να επιλέξει ένα καθορισμένο πλήκτρο από το πληκτρολόγιο, ή να επιλέξει ένα εκ των κουμπιών του ποντικιού του υπολογιστή. Έπειτα με την βοήθεια του πλέγματος που χρησιμοποιείται ως αναφορά για το πραγματικό περιβάλλον το οποίο είναι ορατό από τον χρήστη, ελέγχεται η πιθανότητα σύγκρουσης με την μέθοδο `checkCollision(hit)`. Συγκεκριμένα εάν εμφανιστεί σύγκρουση, επιστρέφεται στο πρόγραμμα η θέση του τριγώνου του πλέγματος στο οποίο εμφανίζεται η ανάλογη σύγκρουση υπό την μορφή Vector3 (x,y,z).

Έπειτα λόγω του ότι η θέση του τριγώνου αποτελείται από τρία Vector3 όπου το καθένα αντικατοπτρίζει και την κάθε κορυφή του τριγώνου, υπολογίζεται ο μέσος όρος των τριών κορυφών που αποτελεί το κεντρικό σημείο του τριγώνου (`centroid`). Το σημείο αυτό περνά ως παράμετρος κατά την εκτέλεση του script `ObjectSpawner.cs`, όπου και χρησιμοποιείται ως αναφορά για την τοποθεσία του εικονικού αντικειμένου που δημιουργείται.

δ). Το τελετυταίο script υποστηρίζει την δυνατότητα για την καταστροφή του αντικειμένου που τοποθετείται σύμφωνα με όσα λέχθηκαν προηγουμένως. Συγκεκριμένα στο script αυτό υπάρχει η μεταβλητή age, η οποία αποθηκεύει ένα μοναδικό αριθμό για κάθε νέο αντικείμενο. Έπειτα στο script `Raycast.cs`, ελέγχεται το πάτημα του κουμπιού πληκτρολογίου R, το οποίο υποστηρίζει τη δυνατότητα διαγραφής/καταστροφής αντικειμένου (Remove). Επομένως, με τη βοήθεια και της μεταβλητής age που αναφέρθηκε προηγουμένως το αντικείμενο καταστρέφεται, οπόταν και εξαφανίζεται από την σκηνή. Αυτό είναι βοηθητικό, εάν κάποιο από τα αντικείμενα τοποθετήθηκε σε λάθος σημείο έπειτα από σφάλμα του χρήστη, οπόταν υπάρχει η δυνατότητα καταστροφής.

4.4.4 Επιπρόσθετες διεργασίες για σωστή λειτουργία

Λόγω ασυμβατότητας ταυτοποίησης των συντεταγμένων του Kinect και των συντεταγμένων του Oculus, το πλέγμα που δημιουργήθηκε από τα δεδομένα βάθους

του Kinect είχε διαφορετική κλίμακα σε σχέση με την μετακίνηση του Oculus κατά μήκος της κάμερας θέσεως. Επομένως το πλέγμα έγινε scale down δηλαδή έτυχε παραμετροποίηση κλίμακας με scale factor = 0.1. Με τον τρόπο αυτό οι αποστάσεις μεταξύ Oculus και εικονικού πλέγματος είχαν τις σωστές αναλογίες, επομένως και οι αποστάσεις μεταξύ Oculus και εικονικών αντικειμένων ήταν οι αναμενόμενες.

Κεφάλαιο 5

Συμπεράσματα

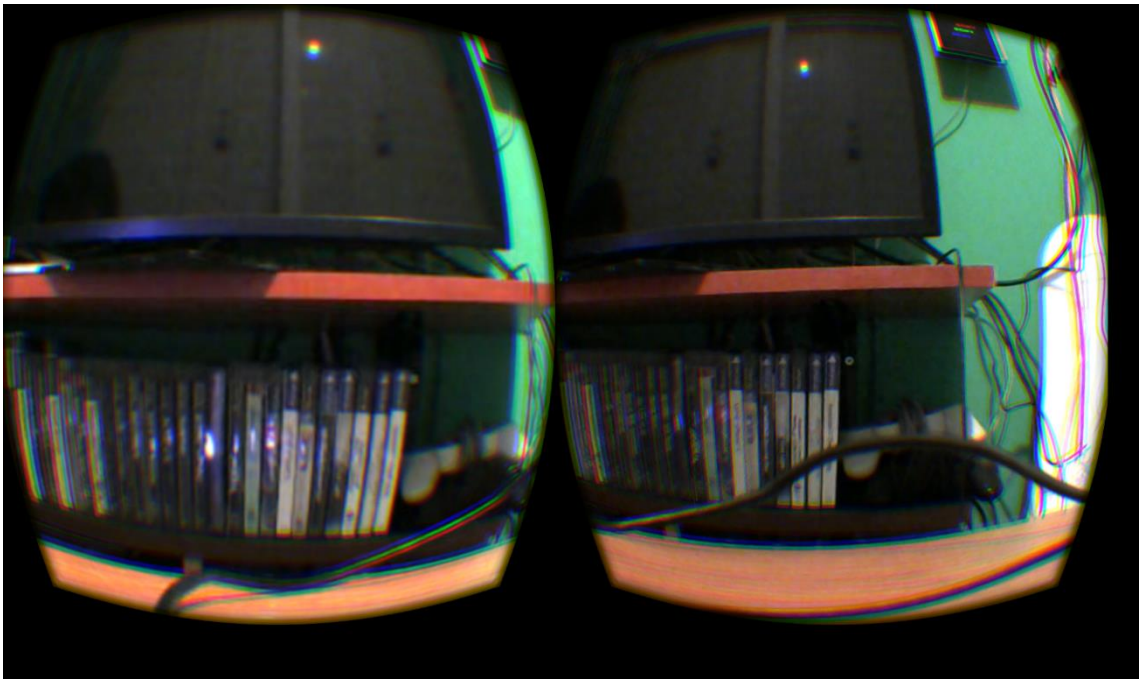
5.1 Οθόνες αποτελεσμάτων 47

5.2 Γενικά συμπεράσματα 50

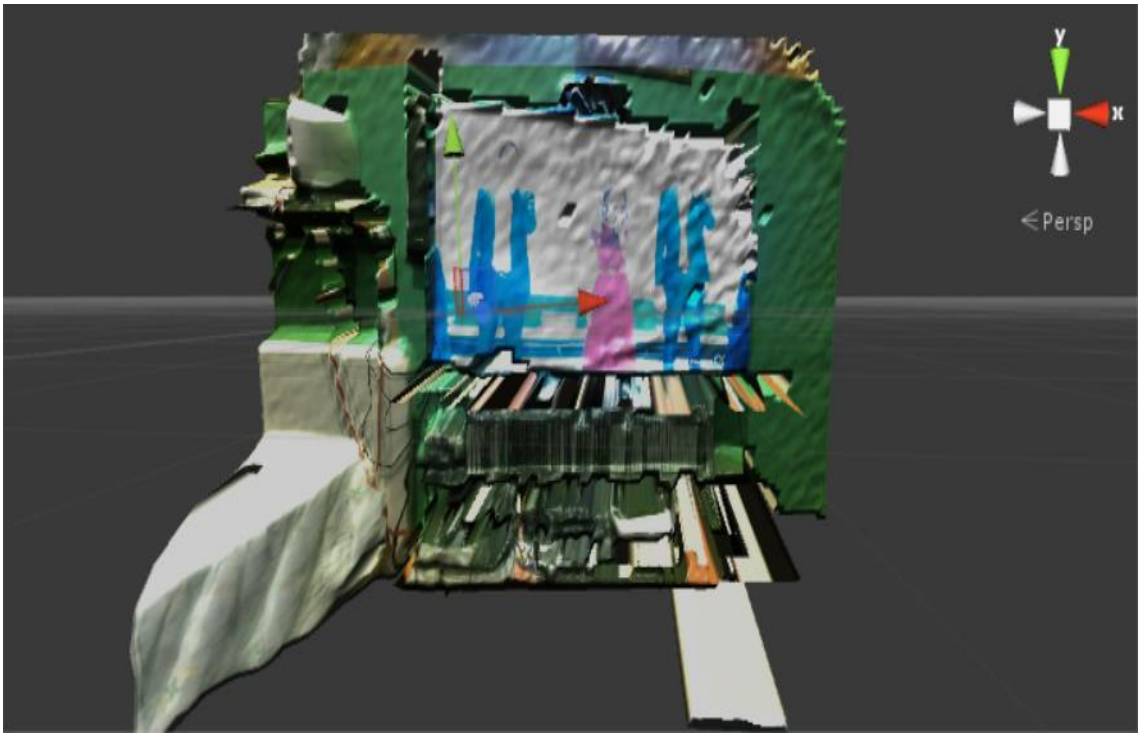
5.3 Προβλήματα που προέκυψαν και γνωστά σφάλματα 51

5.4 Μελλοντική εργασία και επέκταση συστήματος 52

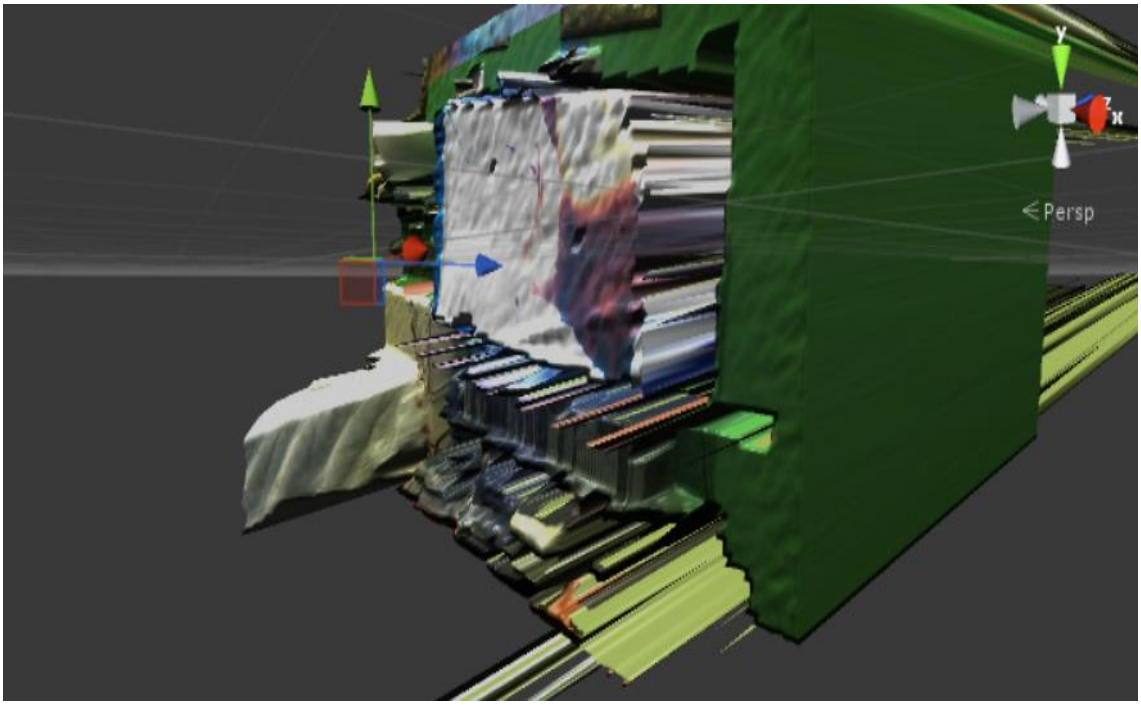
5.1 Οθόνες αποτελεσμάτων



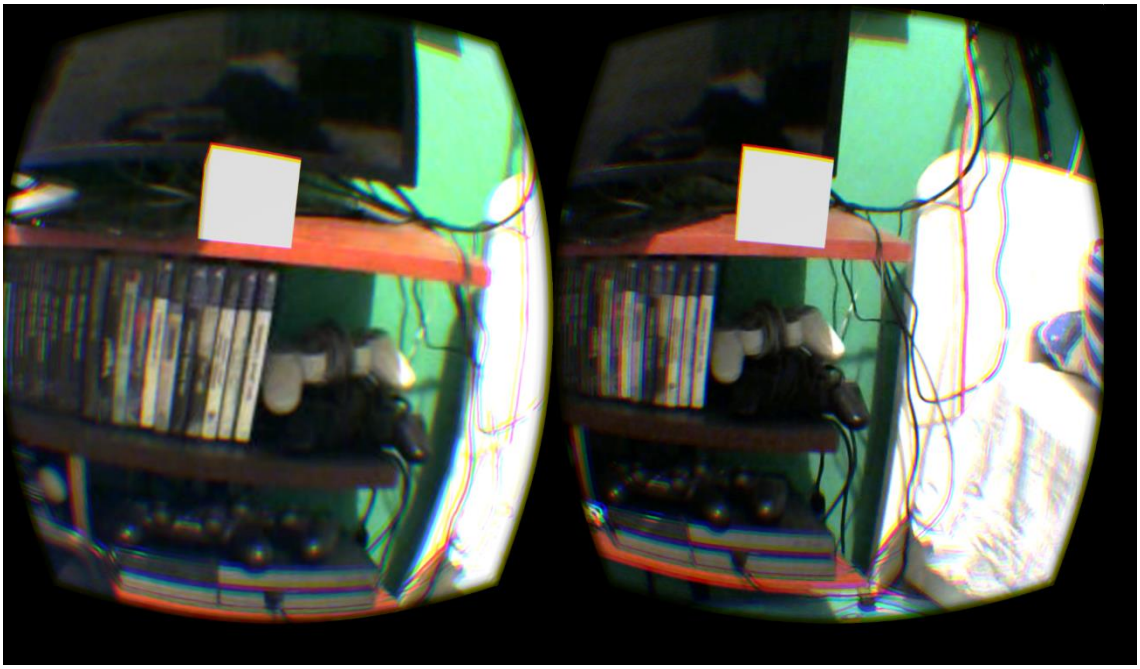
Σχήμα 5.1 Live Camera Feed Screen



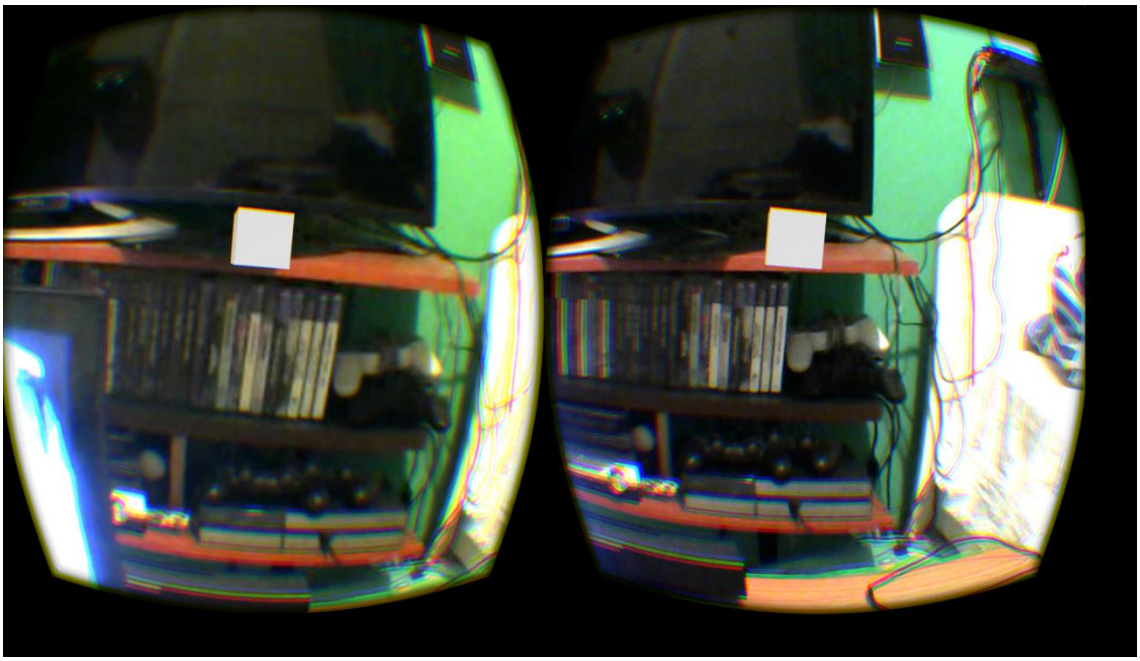
Σχήμα 5.2 Dynamic Mesh Screen - Front



Σχήμα 5.2.1 Dynamic Mesh Screen - Side



Σχήμα 5.3.1 Combined Scene with object (Near)



Σχήμα 5.3.2 Combined Scene with objects (Far)

5.2 Γενικά συμπεράσματα

Το τελικό σύστημα παρουσιάζει την τοποθέτηση εικονικών αντικειμένων στον χώρο που αποδίδεται από την εικόνα των δύο καμερών σε πραγματικό χρόνο. Τα αντικείμενα τοποθετούνται στον χώρο με βάση την απόσταση του Oculus από το δυναμικό πλέγμα που δημιουργείται από τον αισθητήρα του Kinect. Επομένως οι εικονικές αποστάσεις είναι ανάλογες με τις αποστάσεις του πραγματικού χώρου. Με την κίνηση του κατά μήκος της κάμερας ανίχνευσης του Oculus (IR positional camera), ο χρήστης αντιλαμβάνεται το βάθος των αντικειμένων καθώς πλησιάζει ή απομακρύνεται από αυτήν, όπου και τα αντικείμενα φαίνονται πιο κοντά στο οπτικό του πεδίο και αντίστοιχα σε πιο μακρινή απόσταση.

Η τοποθέτηση των εικονικών αντικειμένων λόγω του ότι γίνεται με βάση τις πραγματικές αποστάσεις μεταξύ Oculus και κάμερας θέσεως, όπως και τις αντίστοιχες αποστάσεις από τον αισθητήρα του Kinect, δημιουργεί την αίσθηση στον χρήστη ότι τα εικονικά αντικείμενα υπάρχουν ήδη στην σκηνή ως πραγματικά. Αυτό έχει ως επακόλουθο η κίνηση του χρήστη κατά μήκος της κάμερας θέσεως να μεταβάλλει την θέση των αντικειμένων στο οπτικό του πεδίο, με αποτέλεσμα τα αντικείμενα να συμπεριφέρονται ως πραγματικά αντικείμενα στον χώρο.

Η έρευνα που έγινε απέδειξε ότι η επαύξηση εικονικών αντικειμένων με βάση την γεωμετρία του χώρου, είναι εφικτή σε μεγάλο βαθμό. Λαμβάνοντας υπόψη τους πιο κάτω περιορισμούς και τα προβλήματα που αναφέρονται στο μέρος 5.2, εντούτοις η έρευνα αυτή αποτελεί τη βάση για την μελλοντική επέκταση της εφαρμογής κατά την οποία η αλληλεπίδραση αναμένεται να βελτιωθεί ώστε να προσδώσει τα αναμενόμενα αποτελέσματα στον χρήστη. Με την επίλυση των προβλημάτων αυτών αναμένεται αποτελεσματική επέκταση της εφαρμογής χρησιμοποιώντας διάφορες τεχνικές και μεθόδους. Παράλληλα νέες συσκευές θα επικεντρώνονται στην αλληλεπίδραση χρήστη και εικονικού κόσμου, προσφέροντας καινοτόμες δραστηριότητες. Οι νέες λειτουργίες που θα είναι διαθέσιμες, θα βελτιώσουν την αλληλεπίδραση χρήστη και εικονικού κόσμου ενώ θα ανοίξουν και το δρόμο για νέες δυνατότητες στον κόσμο της εικονικής και επαυξημένης πραγματικότητας, ώστε να καλύψουν αδυναμίες που παρατηρούνται στον φυσικό κόσμο.

5.3 Προβλήματα που προέκυψαν και περιορισμοί

Ένα από τα προβλήματα που προέκυψαν ήταν η περιορισμένη απόσταση ανίχνευσης βάθους από τον αισθητήρα του Kinect, όπως και η περιορισμένη απόσταση ανίχνευσης θέσεως από την κάμερα θέσεως του Oculus Rift. Όπως είχαμε προαναφέρει στα μέρη 3.2.1 και 3.3.1 του τρίτου κεφαλαίου οι αποστάσεις ανίχνευσης των δύο συσκευών είχαν ως ακολούθως. Ο αισθητήρας βάθους του Kinect, έχει εμβέλεια από 40cm μέχρι 4.5m, και η κάμερα θέσεως του Oculus Rift έχει εμβέλεια από 50cm μέχρι 2.5m. Επομένως γίνεται κατανοητό ότι σε αποστάσεις μικρότερες από 50cm όπως και πάνω από 2.5m, προκύπτουν προβλήματα ανίχνευσης και από τις δύο συσκευές. Επομένως η αλληλεπίδραση στον χώρο περιορίζεται στα 2.5m. Επίσης το field of view της κάμερας ανίχνευσης θέσεως του Oculus ξεκινά στα 75°x52°, το οποίο και αυξάνεται καθώς ο χρήστης απομακρύνεται στην μέγιστη απόσταση από την κάμερα δηλαδή στα 2.5m. Επομένως εάν ο χρήστης αλληλεπιδράσει εκτός των ορίων του field of view της κάμερας, προκύπτουν εξίσου προβλήματα ανίχνευσης.

Λόγω της πολυπλοκότητας της διασύνδεσης, κατά την οποία τουλάχιστον 5 θύρες USB έπρεπε να δουλεύουν ταυτόχρονα, προέκυψαν προβλήματα αποσύνδεσης των συσκευών ή μη αναγνώρισή τους κατά διαστήματα από τον ηλεκτρονικό υπολογιστή. Η διασύνδεση έγινε σε φορητό υπολογιστή, στον οποίο οι θύρες USB δεν είναι ανεξάρτητες μεταξύ τους, οπόταν μερικές φορές η ταυτόχρονη αλληλεπίδραση μεταξύ των συσκευών δεν ήταν κατορθωτή.

Κατά την έρευνα όπως επισημάνθηκε, χρησιμοποιήθηκαν οι κάμερες Logitech C310, για την μεταφορά εικόνας στο Oculus, σε πραγματικό χρόνο. Εντούτοις, η ασυμβατότητα του field of view των καμερών που βρίσκεται στις 60°, σε σχέση με το field of view του Oculus Rift που βρίσκεται στις 110°, είχε ως επακόλουθο το εικονικό πλέγμα, δημιουργημένο από τα δεδομένα βάθους του Kinect, να μην ταυτοποιείται με την εικόνα που προβάλλεται στο Oculus μέσω των καμερών. Επομένως, οι αποστάσεις των εικονικών αντικειμένων αν και ήταν σωστές με βάση την πραγματικότητα, εμφανίζονταν σε λάθος θέση σε σχέση με την εικόνα από τις δύο κάμερες. Αυτό μπορεί να διορθωθεί με την εγκατάσταση φακών (lenses) στις δύο κάμερες, όπως αναφέρθηκε και στο μέρος 2.5 του κεφαλαίου, ώστε να αυξηθεί το field of view των καμερών. Το αποτέλεσμα της διεργασίας αυτής θα είναι η ταυτοποίηση των αποστάσεων των εικονικών αντικειμένων με την εικόνα των δύο καμερών σε πραγματικό χρόνο.

Τέλος, ένα ακόμη πρόβλημα το οποίο προέκυψε ήταν η αστάθεια της εικόνας που μεταφέρεται από τις κάμερες στο Oculus σε πραγματικό χρόνο. Αυτό προέκυψε κυρίως λόγω της ασυμβατότητας που αναφέρθηκε προηγουμένως στο field of view. Η εικόνα ήταν θολή ενώ κατά την κίνηση του χρήστη σε οποιαδήποτε κατεύθυνση, υπήρχαν προβλήματα motion blurring και αστάθειας εικόνας. Αυτό μπορεί να βελτιωθεί εάν οι κάμερες ρυθμιστούν από τον υπολογιστή μέσω προγραμμάτων που βελτιώνουν την λειτουργία τους, όπως και μέσω κωδικοποίησης από το Unity. Πιστεύεται ότι με την εγκατάσταση των φακών που σημειώθηκε στην προηγούμενη παράγραφο, τα πλείστα ζητήματα των καμερών θα επιλυθούν.

5.4 Μελλοντική εργασία και επέκταση συστήματος

Στο σύστημα που αναπτύχθηκε αν και υπήρξαν αρκετά θετικά αποτελέσματα, προέκυψαν παράλληλα και κάποια προβλήματα αλλά και σφάλματα τα οποία αναφέρθηκαν προηγουμένως. Εντούτοις, στο παρόν σημείο επισημαίνονται οι διάφορες τεχνικές και συσκευές που μπορούν να επεκτείνουν/αναβαθμίσουν το σύστημα ώστε να υπάρχει πιο αποδοτικότερη αλληλεπίδραση μεταξύ χρήστη και διασύνδεσης υλικού, με σκοπό τη βελτίωση της εμπειρίας του χρήστη. Παράλληλα με την χρησιμοποίηση νέων συσκευών που θα ενσωματώνουν νέες λειτουργικές δυνατότητες αναμένεται να συμβάλουν στη βελτίωση της υφιστάμενης μεθόδου προβολής και αλληλεπίδρασης ώστε να προκύψουν τα αναμενόμενα αποτελέσματα.

Η διασύνδεση που δημιουργήθηκε για τον σκοπό της έρευνας δεν είναι και τόσο εργονομική. Αυτό έγκειται κυρίως στο ότι απαιτείται η ταυτόχρονη λειτουργία και ενσωμάτωση των συσκευών Kinect, Oculus, και καμερών στο κεφάλι του χρήστη δεν είναι αρκετά πρακτική. είναι. Ειδικότερα επισημαίνεται ότι το πρόβλημα εστιάζεται κυρίως στη συσκευή του Kinect, το οποίο έχει το μεγαλύτερο βάρος από τις τρεις βασικές συσκευές. Όπως σημειώθηκε στο τρίτο κεφάλαιο το Kinect μπορεί να αντικατασταθεί με εναλλακτικές συσκευές ανίχνευσης βάθους. Η απόδοση του αισθητήρα είναι αρκετά ικανοποιητική, ωστόσο μια τέτοια αντικατάσταση ενδέχεται να καταστήσει τη διασύνδεση πιο εργονομική προς τον χρήστη.

Το Oculus Rift DK2 είναι μια επαναστατική συσκευή με αμέτρητες δυνατότητες και λειτουργίες, ενώ δίνει στον χρήστη μια καινοτόμα εμπειρία στον κόσμο της εικονικής πραγματικότητας. Η χρησιμοποίησή του στην συγκεκριμένη έρευνα κρίνεται θετική σε ικανοποιητικό βαθμό. Ωστόσο, νέες συσκευές προβολής εικονικής πραγματικότητας, οι

οποίες αναμένεται να αναπτυχθούν σε μελλοντικό στάδιο αναμένεται να βελτιώσουν την εμπειρία μεταξύ χρήστη και εφαρμογής, παρέχοντας νέες δυνατότητες και αλληλεπιδράσεις με διαφορετικές και καινοτόμες τεχνολογίες. Στο τρίτο κεφάλαιο της διπλωματικής έγινε αναφορά στις συσκευές αυτές όπως και στα πιο σημαντικά τους χαρακτηριστικά. Μια εξ αυτών είναι και η συσκευή SteamVR, η οποία περιλαμβάνει ανίχνευση κινήσεων σε ολόκληρο δωμάτιο, κάτι που αυτομάτως εξαλείφει τον περιορισμό της ανίχνευσης από την κάμερα θέσεως του Oculus, η οποία είχε περιορισμένη έκταση ανίχνευσης.

Ένα από τα κύρια προβλήματα που προέκυψαν ήταν η λιγότερο εργονομική διασύνδεση χρήστη και υλικού. Το κύριο πρόβλημα ήταν η περιορισμένη μετακίνηση του χρήστη στον χώρο αλληλεπίδρασης. Κι' αυτό λόγω της καλωδίωσης των συσκευών κατά την σύνδεση τους με τον ηλεκτρονικό υπολογιστή. Λύση στο πρόβλημα αυτό μπορεί να υπάρξει καθώς με την ανάπτυξη της τεχνολογίας ολοένα και περισσότερες συσκευές με καλωδίωση μπορούν να χρησιμοποιηθούν και ασύρματα. Ένα από τα παραδείγματα αυτά είναι και το Oculus Power Hack, που αναφέρθηκε στο δεύτερο κεφάλαιο και το οποίο επιτρέπει την ασύρματη μεταφορά εικόνας υψηλής ευκρίνειας από τον υπολογιστή στο Oculus [16].

Η τοποθέτηση των εικονικών αντικειμένων στον εικονικό χώρο έγινε όπως αναφέρθηκε προηγουμένως στο 4.4.1 με εντολή από τα κουμπιά του ποντικιού του υπολογιστή, ή με το πάτημα ενός πλήκτρου από το πληκτρολόγιο. Αυτό όμως δεν είναι και πολύ εύχρηστο, καθώς όπως σημειώθηκε πιο πάνω, υπάρχει η μελλοντική προσδοκία για ολοκληρωμένη ασύρματη αλληλεπίδραση χρήστη και εφαρμογής. Επομένως, η μελλοντική έρευνα αναμένεται να συμβάλει στην χρήση της αναγνώρισης φωνής από τον αισθητήρα του Kinect, ώστε ο χρήστης να μπορεί να δίνει φωνητικές εντολές για την τοποθέτηση των εικονικών αντικειμένων. Δηλαδή σε τέτοιο ενδεχόμενο αναμένεται ο χρήστης να μπορεί να δώσει εντολή για τοποθέτηση στατικού αντικειμένου κάνοντας χρήση φωνητικής εντολής "Static" ή αντίστοιχα δυναμικού αντικειμένου με την εντολή "Dynamic".

Τέλος, μια ακόμη συσκευή η οποία μελετήθηκε κατά τη διάρκεια της έρευνας της παρούσας διπλωματικής εργασίας είναι και η συσκευή ανίχνευσης κινήσεων χεριού Leap Motion. Η συσκευή αυτή δεν τοποθετήθηκε στη διασύνδεση λόγω προβλημάτων που αφορούσαν την πολυπλοκότητα της υφιστάμενης διασύνδεσης. Παρ' όλα αυτά

όμως η ενσωμάτωση μιας τέτοιας συσκευής παρέχει στον χρήστη νέες δυνατότητες καθώς θα μπορεί να αλληλεπιδρά με το εικονικό περιβάλλον και με τα χέρια του, όπως για παράδειγμα να μετακινήσει επαυξημένα αντικείμενα κατά μήκος του χώρου αλληλεπίδρασης.

Βιβλιογραφία

[1] Gobbetti, E. & Scateni, R. 1998, "Virtual reality: Past, present, and future", *Virtual environments in clinical psychology and neuroscience: Methods and techniques in advanced patient-therapist interaction*.

[2] "Oculus Rift" [Online].
Available: https://www.oculus.com/

[3] "Kinect for Windows" [Online].
Available: https://www.microsoft.com/en-us/kinectforwindows/

[4] "Logitech Cameras C310" [Online].
Available: http://www.logitech.com/en-us/product/hd-webcam-c310

[5] Newcombe, R.A. & Davison, A.J. 2010, "Live dense reconstruction with a single moving camera", *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* IEEE, pp. 1498.

[6] "Point Cloud" [Online].
Available: http://en.wikipedia.org/wiki/Point_cloud

[7] "SFM/SLAM" [Online].
Available: http://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping

[8] "KinectFusion" [Online].
Available: https://msdn.microsoft.com/en-us/library/dn188670.aspx

[9] Khattak, S., Cowan, B., Chepurna, I. & Hogue, A. "A real-time reconstructed 3D environment augmented with virtual objects rendered with correct occlusion"

[10] "Creative Senz3D Camera" [Online].
Available: http://us.creative.com/p/web-cameras/creative-senz3d

[11] "Leap Motion" [Online].
Available: https://www.leapmotion.com/

[12] "Six Degrees of Freedom – 6Dof" [Online].
Available: http://en.wikipedia.org/wiki/Six_degrees_of_freedom

[13] Piumsomboon, T., Clark, A. & Billinghurst, M. 2011, "Physically-based interaction for tabletop augmented reality using a depth-sensing camera for environment mapping".

[14] Benko, H., Jota, R. & Wilson, A. 2012, "MirageTable: freehand interaction on a projected augmented reality tabletop", *Proceedings of the SIGCHI conference on human factors in computing systems*ACM, pp. 199.

[15] "AR-Rift" [Online].
Available: http://willsteptoe.com/post/66968953089/ar-rift-part-1

[16] "Oculus Rift USB Power Hack" [Online].
Available: http://mikejmoffitt.com/wp/?p=156

[17] "Oculus Rift" [Online – Wikipedia].
Available: http://en.wikipedia.org/wiki/Oculus_Rift

[18] "OLED Display" [Online].
Available: http://en.wikipedia.org/wiki/OLED

[19] "Comparison between Oculus Rift DK1/DK2 1" [Online].
Available: http://riftinfo.com/oculus-rift-specs-dk1-vs-dk2-comparison

[20] "Comparison between Oculus Rift DK1/DK2 2" [Online].
Available: http://in2gpu.com/2014/08/10/oculus-rift-dk1-vs-dk2/

[21] "Project Morpheus" [Online].
Available:
http://en.wikipedia.org/wiki/Project_Morpheus_%28virtual_reality%29

[22] "Samsung Gear VR" [Online].
Available:
http://www.samsung.com/global/microsite/gearvr/gearvr_features.html

[23] "Google Cardboard" [Online].
Available: https://www.google.com/get/cardboard/

[24] "Microsoft HoloLens" [Online].
Available: https://www.microsoft.com/microsoft-hololens/en-us

[25] "Steam VR" [Online].
Available: http://store.steampowered.com/universe/vr/

[26] "Oculus Rift Alternatives" [Online].
Available:
http://riftinfo.com/top-oculus-rift-alternatives-virtual-reality-is-getting-huge

[27] "Kinect for Windows" [Online – Wikipedia].
Available: http://en.wikipedia.org/wiki/Kinect

[28] "Comparison between Kinect v1 and v2" [Online].
Available: http://zugara.com/how-does-the-kinect-2-compare-to-the-kinect-1

[29] "SENTIS TOF – P510 from BlueTechnix" [Online].
Available:
http://www.bluetechnix.com/en/products/depthsensing/product/sentis-tof-p510/

[30] "PhotonlCs from PMD Intelligence" [Online].
Available: http://pmdtec.com/solutions/camera_developer.php

[31] "SoftKinetic DepthSensor" [Online].
Available: http://www.softkinetic.com/Products/DepthSenseSensors

[32] "Matrix Vision mvBlueFOX depth sensr" [Online].
Available: http://www.matrix-vision.com/home-de.html

[33] "Unity 3D" [Online].
Available: https://unity3d.com/

[34] "Unity 3D" [Online – Wikipedia].
Available: http://en.wikipedia.org/wiki/Unity_%28game_engine%29

[35] "Stereoscopy" [Online – Wikipedia].

Available: http://en.wikipedia.org/wiki/Stereoscopy

[36] "FOVE" [Online].

Available: http://www.getfove.com/

Παράρτημα Α

1. Ενδεικτικός Κώδικας (Unity3D – C#)

1.1 Raycast.cs

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class Raycast : MonoBehaviour {
5
6   public ObjectSpawner _ObjectSpawner;
7   public Transform target;
8   public int counter;
9
10  // Use this for initialization
11  void Start ()
12  {
13      _ObjectSpawner = new ObjectSpawner();
14  }
15
16  // Update is called once per frame
17  void Update ()
18  {
19
20      OVRCameraRig rig =
(OVRCameraRig)GameObject.FindObjectOfType(typeof(OVRCameraRig));
21
22      Transform eye = rig.centerEyeAnchor;
23
24      // raycast from cursor click
25      // OVRCameraRig rig =
(OVRCameraRig)GameObject.FindObjectOfType(typeof(OVRCameraRig));
26      // Camera eye = rig.rightEyeAnchor.GetComponent<Camera>();
27      // Vector3 textureSize = new Vector3(eye.targetTexture.width,
eye.targetTexture.height, 1f);
28      // Vector3 screenSize = new Vector3(Screen.width,
Screen.height, 1f);
29      // Vector3 vrMousePosition = new Vector3();
30      // vrMousePosition.x = Input.mousePosition.x * textureSize.x /
screenSize.x;
31      // vrMousePosition.y = Input.mousePosition.y * textureSize.y /
screenSize.y;
32      // vrMousePosition.z = Input.mousePosition.z * textureSize.z /
screenSize.z;
33      // Ray ray = eye.ScreenPointToRay(vrMousePosition);
34
35      // set collision layerMask => RaycastLayer Index
36      // raycast only on RaycastLayer - ignore other layers
37      int layerMask = 1 << 8;
38
39      // focus with Oculus Rift to get ray
40      RaycastHit hit;
41
42      if (Input.GetMouseButtonDown (0))
43      {
44          //eye.position, eye.forward
```

```
45          if (Physics.Raycast(eye.position, eye.forward , out hit,
Mathf.Infinity, layerMask) == true)
46          {
47              try
48              {
49                  // test collision object
50                  // Debug.Log ("collision at " +
hit.collider.gameObject.name);
51
52                  // test raycast hit distance
53                  Debug.Log (hit.distance);
54
55                  Vector3 positionHit = checkCollision(hit);
56                  counter++;
57                  // Spawn sphere at position pos
58                  _ObjectSpawner.spawn(positionHit, 0, counter);
59                  GameObject obj =
GameObject.FindGameObjectWithTag("SpawnedObject");
60                  obj.transform.rotation =
rig.centerEyeAnchor.rotation;
61
62              }
63              catch (System.NullReferenceException e)
64              {
65                  Debug.Log ("No hit detected");
66                  Debug.Log (e.ToString());
67              }
68
69          }
70          else
71          {
72              Debug.Log ("Did not hit anything");
73          }
74      }
75      else if (Input.GetMouseButtonDown (1))
76      {
77          if (Physics.Raycast(eye.position, eye.forward , out hit,
Mathf.Infinity, layerMask) == true)
78          {
79              try
80              {
81                  // test collision object
82                  // Debug.Log ("collision at " +
hit.collider.gameObject.name);
83
84                  Debug.Log (hit.distance);
85
86                  Vector3 positionHit = checkCollision(hit);
87                  counter++;
88                  // Spawn sphere at position pos
89                  _ObjectSpawner.spawn(positionHit, 1, counter);
90                  GameObject obj =
GameObject.FindGameObjectWithTag("SpawnedObject");
91                  // transform object rotation as the rig transform
rotation
92                  obj.transform.rotation =
rig.centerEyeAnchor.rotation;
93              }
94              catch (System.NullReferenceException e)
95              {
```

```
96                  Debug.Log ("No hit detected");
97                  Debug.Log (e.ToString());
98              }
99
100             }
101             else
102             {
103                 Debug.Log ("Did not hit anything");
104             }
105         }
106
107         if (Input.GetKeyDown(KeyCode.R))
108         {
109             foreach (GameObject obj in
GameObject.FindGameObjectsWithTag("SpawnedObject"))
110             {
111                 if (obj.GetComponent<SpawnedObject>().age ==
counter)
112                 {
113                     Debug.Log
(obj.GetComponent<SpawnedObject>().age + " " + counter);
114                     Destroy (obj);
115                     counter--;
116                 }
117             }
118         }
119     }
120
121     Vector3 findAveragePosition (Vector3 p1, Vector3 p2, Vector3
p3)
122     {
123         // find average triangle position
124
125         Vector3 avg = new Vector3();
126
127         avg.x = (p1.x + p2.x + p3.x) / 3;
128         avg.y = (p1.y + p2.y + p3.y) / 3;
129         avg.z = (p1.z + p2.z + p3.z) / 3;
130
131         return avg;
132     }
133
134     Vector3 checkCollision (RaycastHit hit)
135     {
136         Mesh m = (hit.collider as MeshCollider).sharedMesh;
137
138         Vector3[] mVertices = m.vertices;
139         int[] mTriangles = m.triangles;
140
141         // Return triangles of raycast hit - Way 1
142         var p0 = mVertices[mTriangles[hit.triangleIndex * 3 + 0]];
143         var p1 = mVertices[mTriangles[hit.triangleIndex * 3 + 1]];
144         var p2 = mVertices[mTriangles[hit.triangleIndex * 3 + 2]];
145
146         Transform hitTransform = hit.collider.transform;
147
148         //Debug.Log ("Distance from camera: " + hit.distance);
149
150         //Debug.Log( string.Format("Triangle Before Transform:
{0}; {1}; {2}", p0, p1, p2) );
```

```
151
152         p0 = hitTransform.TransformPoint(p0);
153         p1 = hitTransform.TransformPoint(p1);
154         p2 = hitTransform.TransformPoint(p2);
155
156         //Debug.Log( string.Format("Triangle After Transform: {0};
{1}; {2}", p0, p1, p2) );
157
158         Vector3 pos = findAveragePosition(p0,p1,p2);
159
160         return pos;
161     }
162
163 }
```

1.2 ObjectSpawner.cs

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class ObjectSpawner : MonoBehaviour {
5
6   public GameObject obj;
7
8   public void spawn (Vector3 pos, int behaviour, int counter)
9   {
10      // check if the central depth value of kinect is valid
11      if (DepthSourceView.centralDepth !=0 &&
DepthSourceView.centralDepth < 250000)
12      {
13          // create cube
14          obj = GameObject.CreatePrimitive(PrimitiveType.Cube);
15          GameObject objectGroup =
GameObject.FindGameObjectWithTag("ObjectGroup");
16          obj.transform.parent = objectGroup.transform;
17          obj.tag = "SpawnedObject";
18
19          // if object is dynamic add rigidbody
20          if (behaviour == 0)
21          {
22              obj.AddComponent<BoxCollider>();
23              obj.AddComponent<Rigidbody>();
24              //obj.AddComponent<DownForce>();
25          }
26
27          obj.AddComponent<SpawnedObject>();
28          obj.GetComponent<SpawnedObject>().age = counter;
29          obj.transform.localScale = new Vector3(0.25f,0.25f,0.25f);
30          obj.transform.localPosition = adjustPosition(pos,
behaviour);
31          //obj.AddComponent<TransformCube>();
32      }
33      else
34      {
35          print ("Position Out of Bounds");
36      }
37
38  }
39
```

```
40  public Vector3 adjustPosition (Vector3 pos, int behaviour)
41  {
42      // place object based on the users input
43      // dynamic or static
44      Vector3 newPos = new Vector3();
45
46      if (behaviour == 0)
47      {
48          newPos.x = pos.x;
49          newPos.y = pos.y + obj.transform.localScale.y/2;
50          newPos.z = pos.z - obj.transform.localScale.z/2;
51      }
52      else
53      {
54          newPos.x = pos.x;
55          newPos.y = pos.y;
56          newPos.z = DepthSourceView.centralDepth;
57      }
58
59      return newPos;
60  }
61
62 }
```

1.3 SpawnedObject.cs

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class SpawnedObject : MonoBehaviour {
5
6   // age value used for unique object indentification
7   // used to destroy cubes spawned incorrectly
8   public int age;
9
10 }
```

1.4 WebcamTextureScript.cs

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class WebcamTextureScript : MonoBehaviour {
5   public Material mat;
6   public GameObject quad;
7   public GameObject cam;
8   public float imageOffsetX;
9   public float imageOffsetY;//
10
11  private WebCamDevice[] devices;
12  private WebCamTexture texture;
13  private int deviceHeight;
14  private int deviceWidth;
15  private int scaleFactor;//
16
17
18  // Use this for initialization
```

```
19  void Start () {
20      devices = WebCamTexture.devices;
21      deviceHeight = 480;
22      deviceWidth = 640;
23      scaleFactor = 30;
24
25      if (devices.Length > 0)   //Here we look at our devices, and
pick one based on our camera.
26      {
27          if (cam.name == "RightEyeAnchor")
28          {
29              texture = new WebCamTexture( devices[ 1 ].name,
deviceWidth, deviceHeight, 60 );
30          }
31
32          if (cam.name == "LeftEyeAnchor")
33          {
34              texture = new WebCamTexture( devices[ 0 ].name,
deviceWidth, deviceHeight, 60 );
35          }
36
37
38          texture.requestedFPS = 100.0f;
39          texture.Play();    //This starts the webcam stream
40
41          quad.transform.localScale = new Vector3 ( scaleFactor * (
(float)texture.width / (float)texture.height ), scaleFactor, 1.0f );
//This scales our quad (The thing our texture is displayed on) to be
the same size as the webcam stream.
42          quad.transform.Translate(imageOffsetX , imageOffsetY, 0 );
//Sometimes our alignment isn't quite right. By adjusting the offset,
you get the two pictures to line up perfectly.
43
44          mat.mainTexture = texture; //Set the material to the
webcam stream.
45      }
46
47  }
48
49  // Update is called once per frame
50  void Update () {
51      if ( texture != null && texture.didUpdateThisFrame )    //Make
sure we're passing something that exists
52      {
53          quad.transform.rotation = cam.transform.rotation;
//adjust the quad to follow the user's head.
54          quad.transform.Rotate( 0, 0, -90);               // Rotate
the quad so the quad is in portrait mode (Just like our webcam
orientation.
55          print(1/Time.deltaTime); //Print Framerate to console
56      }
57  }
58}
```

Παράρτημα Β

1. Απαιτούμενο Υλικό και Λογισμικό

1.1 Απαιτούμενο Υλικό (Hardware)

Το υλικό που απαιτείται για την λειτουργία του συστήματος είναι:

- Microsoft Kinect Sensor for Windows v2 (Xbox One)
- Kinect Adapter for Windows
- Oculus Rift Development Kit 2
- Logitech Cameras C310
- Έξοδος HDMI/DVI για το Oculus Rift
- Τουλάχιστον 5 διαθέσιμες θύρες USB (1 USB 3.0 για το Kinect, 2 για τις κάμερες, 1 για το Oculus, 1 για την Positional Camera)
- Τουλάχιστον 2-3 διακόπτες/πρίζες (ένα για το Kinect, ένα για τον υπολογιστή και ένα προαιρετικό για το Oculus Rift DK2)

1.2 Απαιτούμενο Λογισμικό (Software)

Το λογισμικό που απαιτείται για την λειτουργία του συστήματος είναι:

- Microsoft Windows 8/8.1 ή νεότερο (το Kinect v2 δεν λειτουργεί σε παλαιότερες εκδόσεις Windows)
- Unity3D 5.0.0f4
- Oculus Configuration Utility v1.6 (SDK 0.6.0.0)
- Kinect for Windows SDK v2.0_1409

2. Εγχειρίδιο Χρήστη

2.1 Οδηγίες σύνδεσης συσκευών

Οι συσκευές συνδέονται μέσω διασύνδεσης USB στον υπολογιστή. Εάν δεν υπάρχουν διαθέσιμες θύρες προτείνεται η χρήση USB Hub 3.0 για αποφυγή αποσύνδεσης κατά διαστήματα των συσκευών. Επίσης όπως αναφέρθηκε στο μέρος 1.1 δύο από τις συσκευές απαιτούν υποστήριξη τροφοδοσίας (Oculus – προαιρετικά, Kinect), ενώ συνιστάται η χρήση τροφοδοσίας και για τον υπολογιστή για την καλύτερη εκτέλεση της εφαρμογής. Τέλος όσον αφορά τις κάμερες Logitech, πρέπει να είναι τοποθετημένες

σε απόσταση ~ 6.5cm, με την απόσταση να είναι μεταβλητή από άνθρωπο σε άνθρωπο. Η απόσταση μπορεί να τύχει αλλαγής χάρη στο custom-made camera rig.

2.2 Οδηγίες εγκατάστασης απαιτούμενου λογισμικού

Το λογισμικό που απαιτείται για την λειτουργία του συστήματος πλην των Windows 8/8.1, είναι διαθέσιμο δωρεάν από τις αντίστοιχες ιστοσελίδες που αναφέρονται και στην βιβλιογραφία [2,3,33]. Κατά την εγκατάσταση των εφαρμογών χρειάζεται επανεκκίνηση του υπολογιστή για ομαλή λειτουργία.

Όσον αφορά το Oculus Configuration Utility, υπάρχει μια επιλογή για την λειτουργία προβολής του Oculus Rift. Η κύρια επιλογή έγκειται μεταξύ Direct HMD Access from Apps και Extend Desktop to the HMD. Η πρώτη μπορεί να επιλεχθεί εάν η εφαρμογή που βρίσκεται στο Unity3D γίνει build από την εφαρμογή οπόταν και μπορεί να τρέξει απευθείας στο Oculus Rift. Η δεύτερη επιλογή γίνεται όταν η εκτέλεση της εφαρμογής γίνεται μέσα από το Unity3D κατά το run. Στην περίπτωση αυτή το Oculus Rift γίνεται μέρος της οθόνης του υπολογιστή ο οποίος βρίσκεται σε λειτουργία επέκτασης – extend mode.

Η μεταφορά της εικόνας από την κύρια οθόνη του υπολογιστή στην οθόνη επέκτασης του Oculus Rift, γίνεται είτε με την χρήση του ποντικιού είτε με την χρήση της συντόμευσης συνδυασμού (Shift + Window Key + Arrow (Left or Right)).

Η επιλογή DK1 Legacy App Support, αφορά την υποστήριξη παλαιότερων εφαρμογών που δημιουργήθηκαν αποκλειστικά για DK1, ενώ η επιλογή Rift Saver μπορεί να απενεργοποιήσει αυτόματα την συσκευή μετά από μια περίοδο αδράνειας. Επίσης στην εφαρμογή υπάρχει επιλογή προβολής σκηνής επίδειξης για λόγους ελέγχου.

Το Kinect for Windows SDK v2 περιέχει επίσης σκηνές επίδειξης για υποστήριξη των λειτουργιών του Kinect, ενώ παρέχεται ο πηγαίος κώδικας σε γλώσσες προγραμματισμού C# ή C++. Για την ομαλή λειτουργία του Kinect πρέπει να μην υπάρχει προηγούμενη έκδοση του SDK προεγκατεστημένη στον υπολογιστή. Το ίδιο ισχύει και για το Unity3D. Σε περίπτωση που προτιμηθεί παλαιότερη έκδοση του Unity3D, δηλαδή πριν από την τελευταία έκδοση 5.0, τότε απαιτείται η έκδοση Pro του Unity, καθώς η δωρεάν έκδοση δεν υποστηρίζει την λειτουργία των συσκευών που χρησιμοποιούνται.

2.3 Οδηγίες χρήσης εφαρμογής

Η εφαρμογή είναι διαθέσιμη μέσω του Unity3D, στην οποία και αναπτύχθηκε. Η εκτέλεση του γίνεται είτε για λόγους εκτέλεσης, είτε για λόγους ελέγχου και ανάπτυξης. Στην πρώτη περίπτωση το πρόγραμμα πρέπει να γίνει build από το Unity3D με target platform Windows, και architecture x86_64. Έπειτα εκτελούμε το πρόγραμμα ενώ για την έξοδο από το πρόγραμμα πατούμε το κουμπί Esc. Στην δεύτερη περίπτωση, αφού ανοίξει το project στο Unity3D, επιλέγουμε το κουμπί Run . Σε περίπτωση που θέλουμε να σταματήσουμε την εφαρμογή ξαναεπιλέγουμε το ίδιο κουμπί. Για την μεταφορά εικόνας από την κύρια οθόνη του υπολογιστή στην δευτερεύουσα οθόνη επέκτασης Oculus Rift, ακολουθούμε τα βήματα που αναφέρθηκαν στο προηγούμενο μέρος 2.1.

Η κύρια σκηνή εκτέλεσης είναι η `DepthOnlyScene`. Τα scripts που χρησιμοποιήθηκαν στην εφαρμογή είναι διαθέσιμα στο φάκελο `Scripts`. Παράλληλα τα scripts που χρησιμοποιούνται τόσο για το Oculus όσο και για το Kinect βρίσκονται στον φάκελο `Scripts` στου φακέλους `OVR` και `KinectView` αντίστοιχα. Στους φακέλους αυτούς υπάρχουν όλα τα `Scripts` που διανέμονται μέσω του `OculusUnityIntegration.package` (Oculus SDK v0.6.0.0), όπως επίσης και μέσω του `Kinect.2.0.1410.19000.package` (Kinect v2 SDK), τα οποία παρέχονται στην βιβλιογραφική αναφορά. Μέρος του πηγαίου κώδικα προβάλλεται στο Παράρτημα Α της διπλωματικής εργασίας ενώ ο υπόλοιπος κώδικας που χρησιμοποιήθηκε βρίσκεται στο ενδεικτικό project στο Unity3D.