



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ: Ηλεκτρονικής και Υπολογιστών
ΕΡΓΑΣΤΗΡΙΟ Διαδραστικών Τεχνολογιών

Επέκταση και αξιολόγηση χειριστηρίων
αλληλεπίδρασης της Μεικτής Πραγματικότητας με
χρήση κινητού τηλεφώνου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΣΤΕΦΑΝΙΩΡΟΣ ΜΙΧΑΗΛ

ΕΠΙΒΛΕΠΩΝ: ΦΕΙΔΑΣ ΧΡΗΣΤΟΣ

ΠΑΤΡΑ – ΦΛΕΒΑΡΗΣ 2025

Πανεπιστήμιο Πατρών, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών.

Στεφανιώρος Μιχαήλ

© 2025 – Με την επιφύλαξη παντός δικαιώματος

Το σύνολο της εργασίας αποτελεί πρωτότυπο έργο, παραχθέν από τον Στεφανιώρο Μιχαήλ, και δεν παραβιάζει δικαιώματα τρίτων καθ' οιονδήποτε τρόπο. Αν η εργασία περιέχει υλικό, το οποίο δεν έχει παραχθεί από τον ίδιο, αυτό είναι ευδιάκριτο και αναφέρεται ρητώς εντός του κειμένου της εργασίας ως προϊόν εργασίας τρίτου, σημειώνοντας με παρομοίως σαφή τρόπο τα στοιχεία ταυτοποίησής του, ενώ παράλληλα βεβαιώνει πως στην περίπτωση χρήσης αυτούσιων γραφικών αναπαραστάσεων, εικόνων, γραφημάτων κ.λπ., έχει λάβει τη χωρίς περιορισμούς άδεια του κατόχου των πνευματικών δικαιωμάτων για την συμπερίληψη και επακόλουθη δημοσίευση του υλικού αυτού.

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η Διπλωματική Εργασία με τίτλο

**Επέκταση και αξιολόγηση χειριστηρίων αλληλεπίδρασης της
Μεικτής Πραγματικότητας με χρήση κινητού τηλεφώνου**

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας
Υπολογιστών

ΣΤΕΦΑΝΙΩΡΟΥ ΜΙΧΑΗΛ ΤΟΥ ΝΙΚΟΛΑΟΥ

Αριθμός Μητρώου: 1072774

Παρουσιάστηκε δημόσια στο Τμήμα Ηλεκτρολόγων Μηχανικών και
Τεχνολογίας Υπολογιστών στις

25/02/2025

και εξετάστηκε από την ακόλουθη εξεταστική επιτροπή:

Χρήστος Φείδας, Αναπληρωτής Καθηγητής, Τομέας Ηλεκτρονικής και
Υπολογιστών (επιβλέπων)

Χρήστος Σιντόρης, Ε.ΔΙ.Π, Τομέας Ηλεκτρονικής και Υπολογιστών (μέλος
επιτροπής)

Κωνσταντίνος Μουστάκας, Καθηγητής, Τομέας Τηλεπικοινωνιών και
Τεχνολογίας Πληροφορίας (μέλος επιτροπής)

Ο Επιβλέπων

Φείδας Χρήστος
Αναπληρωτής Καθηγητής

Ο Διευθυντής του Τομέα

Θεοδωρίδης Γεώργιος
Αναπληρωτής Καθηγητής

ΠΡΟΛΟΓΟΣ

Αρχικά, θα ήθελα να ευχαριστήσω την οικογένειά μου, που με στήριζε καθ' όλη την διάρκεια των σπουδών μου και ήταν μαζί μου στις επιτυχίες και αποτυχίες μου. Μοιράστηκαν τις ανησυχίες μου και με βοήθησαν να γίνω καλύτερος.

Στην συνέχεια θα ήθελα να ευχαριστήσω τον καθηγητή μου, κ. Φείδα, ο οποίος δέχτηκε να είναι ο επιβλέπων καθηγητής μου σε αυτήν την εργασία. Με βοήθησε και με συμβούλεψε εξαιρετικά αυτό το διάστημα, σε θέματα και εκτός διπλωματικής εργασίας.

Φυσικά, ευχαριστώ από τα βάθη της καρδιάς μου τον συνάδελφο κ. Παπαδούλη Γεώργιο, ο οποίος με ενέπνευσε για το θέμα της διπλωματικής εργασίας, έθεσε τις βάσεις αυτής και με τις απίστευτες γνώσεις του, με βοήθησε να ξεπεράσω κάθε δυσκολία. Ήταν στην διάθεσή μου κάθε στιγμή και χωρίς εκείνον, δεν θα ήμουν εδώ σήμερα με το συγκεκριμένο αποτέλεσμα.

Επιπλέον, θα ήθελα να ευχαριστήσω τα παιδιά του εργαστηρίου που με φιλοξένησαν, με βοήθησαν σε αρκετά σημεία και μου επέτρεψαν να χρησιμοποιήσω τον εξοπλισμό τους για την υλοποίηση της εφαρμογής.

Τέλος, θα ήθελα να ευχαριστήσω όλα τα άτομα που συμμετείχαν στο πείραμα για τον χρόνο και την υπομονή τους.

ΠΕΡΙΛΗΨΗ

Επέκταση και αξιολόγηση χειριστηρίων αλληλεπίδρασης της Μεικτής Πραγματικότητας με χρήση κινητού τηλεφώνου

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ:
ΣΤΕΦΑΝΙΩΡΟΣ ΜΙΧΑΗΛ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΕΠΙΒΛΕΠΟΝΤΟΣ:
ΦΕΙΔΑΣ ΧΡΗΣΤΟΣ

Η παρούσα διπλωματική εργασία είναι εμπνευσμένη από την εργασία του συναδέλφου, κ. Παπαδούλη Γεωργίου, με τίτλο: «Σχεδίαση εξωτερικού ελεγκτή για τον έλεγχο περιβάλλοντος μικτής πραγματικότητας» [47]. Στην προκειμένη περίπτωση, χρησιμοποιήθηκε το ίδιο πλαίσιο επικοινωνίας, το MasterServer Framework, για την επικοινωνία μεταξύ ενός καθημερινού κινητού τηλεφώνου με το Microsoft HoloLens 2. Το κινητό τηλέφωνο λειτουργεί ως ελεγκτής και το HoloLens είναι η ελεγχόμενη συσκευή. Πριν ξεκινήσει η υλοποίηση, έγινε αναζήτηση της έννοιας «μεικτή πραγματικότητα» και ξεκίνησε να γίνεται μια εξοικείωση με την Unity, την γλώσσα προγραμματισμού C# και των τεχνολογιών που χρησιμοποιήθηκαν, όπως το MRTK. Η υλοποίηση ξεκίνησε με την δημιουργία του παιχνιδιού, με θέμα το ηλιακό σύστημα, και με την σχεδίαση του σκοπού του παιχνιδιού, δηλαδή τι θα πρέπει να επιτύχει ένας παίκτης. Στην συνέχεια μελετήθηκε το MasterServer Framework και προσαρμόστηκε στα πλαίσια της διπλωματικής εργασίας. Έπειτα, σχεδιάστηκαν τα χειριστήρια παιχνιδιού με χρήση των χειρονομιών που παρέχει το HoloLens 2 και μετά δημιουργήθηκαν τα χειριστήρια του κινητού τηλεφώνου. Τέλος, πραγματοποιήθηκαν πειράματα για να ερευνηθεί αν ο χειρισμός ενός παιχνιδιού με τις εισόδους που προσφέρει ένα κινητό τηλέφωνο, είναι προτιμότερος σε σχέση με τις χειρονομίες που αναγνωρίζει το HoloLens 2. Η έρευνα εξετάζει και την απόδοση του χρήστη αλλά και την συνολική εμπειρία του.

Λέξεις-Κλειδιά: Μεικτή Πραγματικότητα, Microsoft HoloLens 2, Μεταφορά, Περιστροφή, Αλλαγή μεγέθους, Τετραδόνιο, Οπτική Αδρανειακή Οδομετρία, Ελεγκτής, Unity, MasterServer Framework, MRTK

EXTENSIVE ENGLISH SUMMARY

Expansion and evaluation of Mixed Reality interaction controllers using a mobile phone

STUDENT NAME, SURNAME:
MICHAIL STEFANIOROS

SUPERVISOR NAME, SURNAME:
CHRISTOS FEIDAS

This thesis is inspired by the work of colleague Mr. Papadoulis Georgios, titled: "Designing an External Controller for interaction with Mixed Reality Environments" [47]. In this case, the same communication framework, the MasterServer Framework, was used to facilitate communication between an everyday mobile phone and the Microsoft HoloLens 2. The mobile phone functions as the controller, while the HoloLens serves as the controlled device. Before the implementation began, research was conducted on the concept of "mixed reality", and familiarization with Unity, the C# programming language, and the technologies used, such as MRTK, was initiated. The implementation started with the creation of the game, themed around the solar system, and the design of its objective, defining what a player must achieve. Subsequently, the MasterServer Framework was studied and adapted to fit the scope of the thesis. Next, the game controls were designed using the hand gestures provided by the HoloLens 2, followed by the development of mobile phone controls. Finally, experiments were conducted to investigate whether controlling a game using a mobile phone's input is preferable compared to the hand gestures recognized by the HoloLens 2. The research examines both user performance and overall user experience.

Keywords: Mixed Reality, Microsoft HoloLens 2, Translation, Rotation, Scaling, Quaternion, Visual-Inertial Odometry, Controller, Unity, MasterServer Framework, MRTK.

Περιεχόμενα

1 Θεωρητικό Υπόβαθρο.....	12
1.1 Εκτεταμένη Πραγματικότητα (Extended Reality)	12
1.1.1 Εικονική Πραγματικότητα (Virtual Reality)	12
1.1.2 Επαυξημένη Πραγματικότητα (Augmented Reality).....	13
1.1.3 Μικτή Πραγματικότητα (Mixed Reality)	13
1.2 Γραφικά και αναπαράσταση στο χώρο	14
1.2.1 Μεταφορά (Translation).....	14
1.2.2 Περιστροφή (Rotation)	15
1.2.3 Αλλαγή Μεγέθους (Scale)	16
1.2.4 Τετραδόνιο (Quaternion)	17
1.3 Οπτική Αδρανειακή Οδομετρία – Ταυτόχρονος Εντοπισμός και Χαρτογράφηση	18
1.3.1 Οδομετρία (Odometry)	18
1.3.2 Οπτική Οδομετρία (Visual Odometry - VO) [7] [8]	18
1.3.3 Αδρανειακή Οδομετρία (Inertial Odometry)	19
1.3.4 Οπτική Αδρανειακή Οδομετρία (Visual Inertial Odometry - VIO)	20
1.3.5 Ταυτόχρονος Εντοπισμός και Χαρτογράφηση (Simultaneous Localization And Mapping – SLAM)	21
2 Τεχνολογίες	22
2.1 Microsoft Hololens 2.....	23
2.1.1 Βασικά Χαρακτηριστικά	23

2.1.2 Αισθητήρες (Sensors) της συσκευής.....	24
2.1.3 Τρόποι χειρισμού (ελέγχου) του HoloLens 2	25
2.2 Unity.....	32
2.2.1 Βασικά δομικά στοιχεία της Unity	34
2.2.2 Κύκλος ζωής μιας εφαρμογής - Βασικές μέθοδοι.....	35
2.2.3 Unity Hub.....	39
2.2.4 Περιβάλλον Ανάπτυξης της Unity (Unity Editor)	40
2.2.5 Mixed Reality Toolkit (MRTK).....	41
2.2.6 Σύστημα Εισόδου της Unity (Unity's Input System)	47
2.2.7 Microsoft Visual Studio Community	48
3 Σχεδιασμός και Υλοποίηση	53
3.1 Σκοπός	53
3.2 Περιγραφή Παιχνιδιού	54
3.3 Κωδικοποίηση (Serialization).....	58
3.3.1 Class EndianBitConverter.....	59
3.3.2 Class EndianBinaryWriter	60
3.3.3 Class EndianBinaryReader	60
3.3.4 Interface ISerializablePacket.....	60
3.4 Αρχιτεκτονική Επικοινωνίας	61
3.4.1 Interface IMessage.....	61
3.4.2 Interface IMessageDispatcher.....	62
3.4.3 Interface IPeer	62
3.4.4 Interface IIncomingMessage.....	63

3.4.5 Interface IClientSocket	63
3.4.6 Interface IServerSocket	63
3.4.7 Mirror Networking	63
3.5 Υλοποίηση Πλαισίου Επικοινωνίας (Communication Framework)	65
3.5.1 Είσοδοι του κινητού τηλεφώνου	65
3.5.2 Υλοποίηση του πελάτη και του διακομιστή	67
3.6 Σχεδιασμός Παιχνιδιού.....	70
3.6.1 Ηλιακό Σύστημα και πλανήτες.....	71
3.6.2 Τροχιά γύρω από τον ήλιο και ιδιοπεριστροφές.....	72
3.6.3 Πρώτος τρόπος παιχνιδιού – Χειριστήρια Hololens 2	72
3.6.4 Δεύτερος τρόπος παιχνιδιού – Χειριστήρια κινητού τηλεφώνου	74
3.6.5 Μηχανισμός στόχευσης.....	75
3.6.6 Αντικείμενο εκτόξευσης	76
3.6.7 Μηχανισμός ανίχνευσης goals.....	77
4 Αξιολόγηση χειριστηρίων του κινητού.....	77
4.1 Σκοπός του πειράματος.....	78
4.2 Το πείραμα	78
4.2.1 Περιγραφή του πειράματος.....	78
4.2.2 Οργάνωση και διεξαγωγή του πειράματος.....	78
4.3 Επίτευξη στόχου	79
4.3 Ερωτηματολόγιο.....	81
5 Επίλογος	82

1 Θεωρητικό Υπόβαθρο

1.1 Εκτεταμένη Πραγματικότητα (Extended Reality)

Ο όρος Εκτεταμένη Πραγματικότητα (Extended Reality ή XR) περιλαμβάνει την Επαυξημένη Πραγματικότητα (Augmented Reality ή AR), την Εικονική Πραγματικότητα (Virtual Reality ή VR), την Μικτή Πραγματικότητα (Mixed Reality ή MR), και ό,τι βρίσκεται ανάμεσά τους.

1.1.1 Εικονική Πραγματικότητα (Virtual Reality)

Η εικονική πραγματικότητα (VR) είναι μια προηγμένη διεπαφή ανθρώπου-υπολογιστή που προσομοιώνει ένα ρεαλιστικό περιβάλλον. Οι χρήστες μπορούν να κινηθούν μέσα στον εικονικό κόσμο. Μπορούν να τον δουν από διαφορετικές γωνίες, να τον αγγίξουν, να τον πιάσουν και να τον αναδιαμορφώσουν [1]. Τα **γυαλιά εικονικής πραγματικότητας** (virtual reality headset) είναι μια συσκευή που υλοποιεί την εικονική πραγματικότητα για τον χρήστη. Τα γυαλιά εικονικής πραγματικότητας χρησιμοποιούνται ευρέως με τα βιντεοπαιχνίδια, αλλά έχουν και άλλες εφαρμογές, όπως στην υγεία και την εκπαίδευση. Περιλαμβάνουν μια στερεοσκοπική οθόνη τοποθετημένη στο κεφάλι (παρέχοντας ξεχωριστές εικόνες για το κάθε μάτι), στερεοφωνικό ήχο και αισθητήρες παρακολούθησης κινήσεως της κεφαλής. Ορισμένα γυαλιά VR διαθέτουν επίσης αισθητήρες παρακολούθησης των ματιών και χειριστήρια παιχνιδιών. Οι πιο διαδεδομένες συσκευές εικονικής πραγματικότητας είναι τα Oculus Rift, Quest 2 και HTC Vive.



Εικόνα 1: Oculus Rift



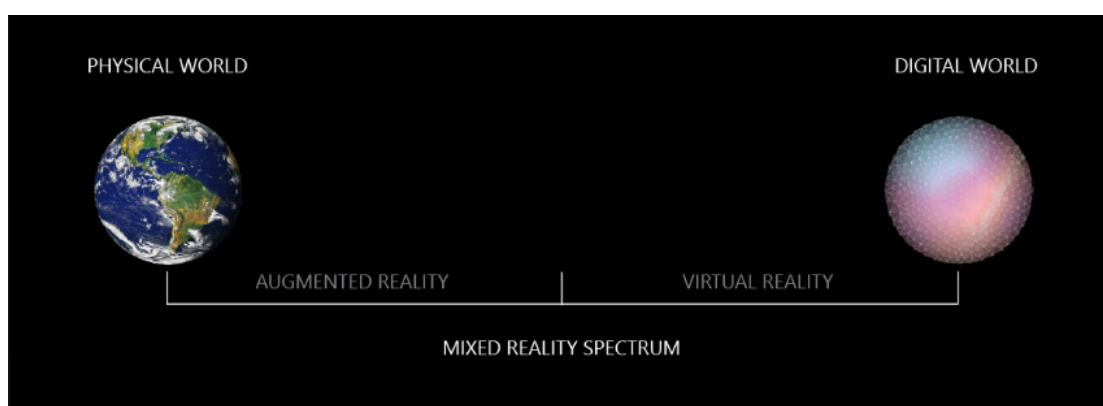
Εικόνα 2: HTC Vive

1.1.2 Επαυξημένη Πραγματικότητα (Augmented Reality)

Η επαυξημένη πραγματικότητα (AR) είναι η ενσωμάτωση ψηφιακών πληροφοριών στο περιβάλλον του χρήστη σε πραγματικό χρόνο. Σε αντίθεση με την εικονική πραγματικότητα (VR), η οποία δημιουργεί ένα εντελώς τεχνητό περιβάλλον, οι χρήστες της AR βιώνουν ένα περιβάλλον του πραγματικού κόσμου με πρόσθετες ψηφιακές πληροφορίες [2]. Για την επαυξημένη πραγματικότητα δεν απαιτείται κάποιος ιδιαίτερος εξοπλισμός, αφού το ίδιο το κινητό τηλέφωνο μπορεί να αποτελέσει συσκευή AR. Η επαυξημένη πραγματικότητα έχει πολλές εφαρμογές σε διάφορους τομείς. Στην εκπαίδευση, επιτρέπει στους μαθητές να αλληλεπιδρούν με ψηφιακά αντικείμενα, διευκολύνοντας την κατανόηση πολύπλοκων εννοιών. Στην ιατρική, οι επαγγελματίες υγείας χρησιμοποιούν AR για να δουν ανατομικές δομές σε πραγματικό χρόνο, βελτιώνοντας την ακρίβεια στις χειρουργικές επεμβάσεις. Επίσης, στην ψυχαγωγία, η AR προσφέρει εμπλουτισμένες εμπειρίες παιχνιδιών και διαδραστικών ιστοριών, ενισχύοντας την αίσθηση της πραγματικότητας.

1.1.3 Μικτή Πραγματικότητα (Mixed Reality)

Η μικτή πραγματικότητα είναι ένας συνδυασμός φυσικών και ψηφιακών κόσμων, που επιτρέπει φυσικές και διαισθητικές 3D αλληλεπιδράσεις μεταξύ ανθρώπου, υπολογιστή και περιβάλλοντος. Αυτή η νέα πραγματικότητα βασίζεται σε προόδους στην υπολογιστική όραση, στην επεξεργασία γραφικών, στις τεχνολογίες απεικόνισης, στα συστήματα εισόδου και στο cloud computing. Ο όρος "μικτή πραγματικότητα" εισήχθη σε μια εργασία του 1994 από τους Paul Milgram και Fumio Kishino. Η μικτή πραγματικότητα συνδυάζει τόσο τον φυσικό όσο και τον ψηφιακό κόσμο. Αυτές οι δύο πραγματικότητες αποτελούν τα άκρα ενός φάσματος γνωστού ως "Εικονικό συνεχές". Αναφερόμαστε σε αυτό το φάσμα πραγματικοτήτων ως το φάσμα της μεικτής πραγματικότητας. Στο ένα άκρο του φάσματος, έχουμε τη φυσική πραγματικότητα στην οποία υπάρχουμε ως άνθρωποι. Στο άλλο άκρο του φάσματος, έχουμε την αντίστοιχη ψηφιακή πραγματικότητα [3].



Εικόνα 3: The Mixed Reality Spectrum

1.2 Γραφικά και αναπαράσταση στο χώρο

Η θεωρία γύρω από τα γραφικά και την αναπαράσταση στον χώρο επικεντρώνεται στην αντίληψη του χρήστη και τη σύνδεση του πραγματικού με τον ψηφιακό κόσμο. Στη βάση αυτής της θεωρίας βρίσκονται η γεωμετρία, η φυσική του φωτός και οι αλγόριθμοι υπολογιστικής όρασης, που επιτρέπουν την ακριβή δημιουργία και τοποθέτηση εικονικών αντικειμένων σε ένα τρισδιάστατο περιβάλλον. Στο πλαίσιο των τεχνολογιών εκτεταμένης πραγματικότητας (XR), τα γραφικά δεν εξυπηρετούν μόνο την απεικόνιση, αλλά και τη διαδραστικότητα με το περιβάλλον. Για την απεικόνιση ενός τρισδιάστατου αντικειμένου στον χώρο, απαιτείται η μεταφορά (translation), η περιστροφή (rotation) και η αλλαγή του μεγέθους του (scale) στον χώρο αυτό.

Για την επεξήγηση αυτών των ενεργειών, ορίζουμε έναν τρισδιάστατο χώρο και ένα σημείο σε αυτόν με συντεταγμένες (x, y, z) . Η αρχή του χώρου είναι το $(0, 0, 0)$.

1.2.1 Μεταφορά (Translation)

Η μεταφορά είναι ο πιο απλός μετασχηματισμός στις τρεις διαστάσεις. Περιλαμβάνει τη μετακίνηση ενός αντικειμένου (ή σημείου) κατά μήκος του άξονα x , y ή z από μία θέση σε άλλη χωρίς να αλλάζει το μέγεθος ή ο προσανατολισμός του. Αυτή η λειτουργία αναπαρίσταται μαθηματικά με πράξεις πρόσθεσης. Στην ουσία, προσθέτουμε στις αρχικές συντεταγμένες του σημείου το ποσό κατά το οποίο θέλουμε να μετακινηθεί.

Ορίζουμε έναν πίνακα μεταφοράς T , ώστε:

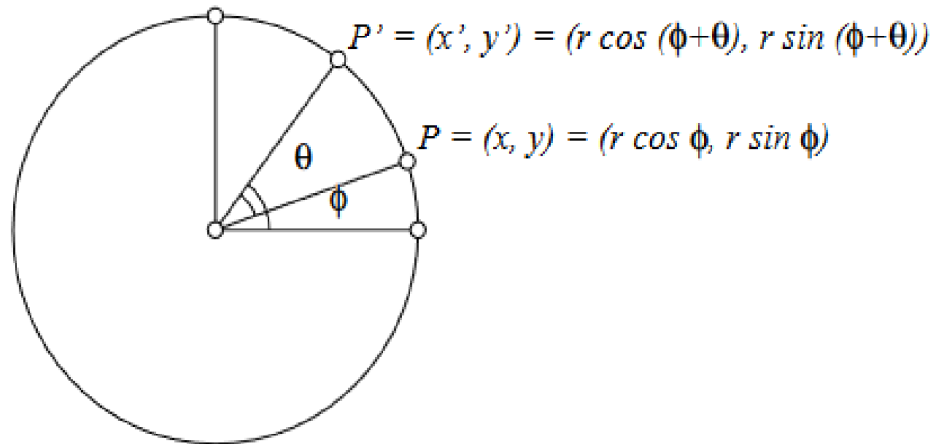
$$T = \begin{pmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + tx \\ y + ty \\ z + tz \\ 1 \end{pmatrix},$$

όπου t_x , t_y , t_z το ποσό κατά το οποίο μετακινείται το σημείο στον χώρο.

Η τέταρτη συντεταγμένη ονομάζεται ομογενής. Οι ομογενείς συντεταγμένες προσθέτουν μία διάσταση (w) στα σημεία του 3D χώρου, μετατρέποντάς τα από (x,y,z) σε (x,y,z,w) . Αυτό επιτρέπει την εύκολη εφαρμογή μετασχηματισμών (μεταφορές, περιστροφές, κλιμακώσεις) μέσω πινάκων 4×4 , κάνοντας τις πράξεις πιο απλές και συνεπείς.

1.2.2 Περιστροφή (Rotation)

Ας ξεκινήσουμε περιγράφοντας την περιστροφή ενός σημείου στο επίπεδο x-y κατά γωνία θ , γύρω από το σημείο αναφοράς μας. Το σημείο $P = (x, y)$ μπορεί να εκφραστεί σε πολικές συντεταγμένες ως $P = (x, y) = (r \cos \phi, r \sin \phi)$. Μια περιστροφή κατά γωνία θ δείχνεται στο παρακάτω σχήμα:



Εικόνα 4: Περιστροφή γύρω από τον άξονα z

Ας το αναπτύξουμε, χρησιμοποιώντας τις τριγωνομετρικές ταυτότητες:

$$\begin{aligned} (x', y') &= (r \cos(\varphi + \theta), r \sin(\varphi + \theta)) = \\ &= (r \cos \varphi \cos \theta - r \sin \varphi \sin \theta, r \cos \varphi \sin \theta + r \sin \varphi \cos \theta) = \\ &= (x \cos \theta - y \sin \theta, x \sin \theta + y \cos \theta) \end{aligned}$$

Έτσι, ορίζουμε τους τελεστές περιστροφής γύρω από κάθε άξονα:

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Για να γενικεύσουμε την περιστροφή γύρω από έναν αυθαίρετο άξονα, θα ακολουθήσουμε μια απλή διαδικασία. Αρχικά, επιλέγουμε έναν από τους κύριους άξονες, έστω τον άξονα x , και μετασχηματίζουμε το σημείο ώστε ο επιθυμητός άξονας περιστροφής - που περνάει από την αρχή των αξόνων - να γίνει παράλληλος με τον άξονα x . Για να το πετύχουμε αυτό, εφαρμόζουμε το πολύ δύο περιστροφές, πρώτα κατά γωνία φ στον άξονα x και στη συνέχεια κατά γωνία ψ στον άξονα y . Στη συνέχεια, περιστρέφουμε το σημείο κατά την επιθυμητή γωνία θ γύρω από τον άξονα x . Τέλος, αντιστρέφουμε τις αρχικές περιστροφές για να επαναφέρουμε το σημείο στο αρχικό σύστημα συντεταγμένων. Έτσι, προκύπτει η παρακάτω εξίσωση:

$$P' = Rx, -\varphi (Ry, -\psi (Rx, -\theta (Ry, \psi (Rx, \varphi (P))))))$$

Αφού έχουμε ορίσει τον τελεστή R ως έναν γενικευμένο τελεστή περιστροφής γύρω από οποιονδήποτε άξονα που περνάει από την αρχή των αξόνων, μπορούμε να χρησιμοποιήσουμε έναν τελεστή μεταφοράς για να περιγράψουμε την περιστροφή γύρω από έναν αυθαίρετο άξονα που δεν περνά απαραίτητα από την αρχή. Συνεπώς, θα προκύψει μια γενικευμένη εξίσωση που περιγράφει την περιστροφή γύρω από έναν οποιονδήποτε άξονα:

$$P' = Tx, y, z (R (T - x, -y, -z (P)))$$

1.2.3 Αλλαγή Μεγέθους (Scale)

Για να αλλάξουμε το μέγεθος ενός αντικειμένου αρκεί, πολύ απλά, να πολλαπλασιάσουμε τις συντεταγμένες του με έναν βαθμωτό αριθμό. Προκύπτει, όμως, το εξής πρόβλημα. Μαζί με το μέγεθος, θα αλλάξει επίσης και η θέση του από την αρχή των αξόνων. Η λύση είναι παρόμοια με την μέθοδο της γενικευμένης περιστροφής. Μετακινούμε το αντικείμενο στην αρχή των αξόνων, εκτελούμε την αλλαγή μεγέθους και το επαναφέρουμε στην αρχική του θέση. Έτσι προκύπτει ο παρακάτω πίνακας:

$$S = \begin{pmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\begin{pmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} sx * x \\ sy * y \\ sz * z \\ 1 \end{pmatrix},$$

όπου κάθε συντελεστής αφορά τον εκάστοτε άξονα.

1.2.4 Τετραδόνιο (Quaternion)

Μια άλλη σημαντική έννοια που βοηθά στην κατανόηση του τρόπου λειτουργίας των περιβαλλόντων ανάπτυξης τρισδιάστατων εφαρμογών είναι τα τετραδόνια. Πρόκειται για μια σχετικά σύνθετη και δύσκολη έννοια, η οποία όμως έχει ευρεία χρήση στα γραφικά. Στο πλαίσιο της διπλωματικής εργασίας, θα τα προσεγγίσουμε πολύ επιφανειακά, εστιάζοντας σε ορισμένες βασικές ιδιότητές τους και τη σημασία τους.

Στα μαθηματικά, τα τετραδόνια (quaternions) είναι μια μη αντιμεταθετική γενίκευση της θεωρίας των μιγαδικών αριθμών. Εισήχθησαν το 1843 από τον Ιρλανδό μαθηματικό Γουίλιαμ Ρόουαν Χάμιλτον και βρήκαν εφαρμογή στη μηχανική και την περιγραφή περιστροφών στον τρισδιάστατο χώρο [4].

Βασικά στοιχεία είναι τα i, j, k , για τα οποία ισχύει: $i^2 = j^2 = k^2 = ijk = -1$. Ο πολλαπλασιασμός θεωρείται προσεταιριστικός, δηλαδή κάθε τετραδόνιο αποτελεί γραμμικό συνδυασμό των βασικών τετραδονίων $1, i, j, k$. Έτσι μπορεί να εκφρασθεί με μοναδικό τρόπο ως:

$q = a + bi + cj + dk$, όπου a, b, c, d είναι πραγματικοί αριθμοί.

Παρακάτω αναφέρονται μερικές πολύ βασικές έννοιες του τετραδονίου [5]:

- 1) Έστω τετραδόνιο q . Το αντίστροφό του ορίζεται ως:

$$q^{-1} = \frac{q^*}{|q|^2}, \text{ έτσι ώστε: } q^{-1}q = qq^{-1} = 1, q^*: \text{ μιγαδικός συζυγής του } q.$$

- 2) Ορίζουμε τον τελεστή περιστροφών $R_q(q_1, q_2)$ σύμφωνα με την περιγραφή που δόθηκε για τους διαδοχικούς πολλαπλασιασμούς σε προηγούμενη ενότητα. Για λόγους συντομίας και ευκολίας, ταυτίζουμε τον τελεστή R_q με τον πολλαπλασιασμό $*$, ώστε να περιγράψουμε τις διαδοχικές περιστροφές ως:

$$q_{final} = q_1 * q_2 * \dots * q_n$$

- 3) Ορίζουμε τον τελεστή $T_q(q, v)$, όπου q είναι ένα τετραδόνιο και v ένα διάνυσμα τριών στοιχείων, έτσι ώστε το αποτέλεσμα να είναι ένα νέο διάνυσμα που έχει περιστραφεί σύμφωνα με την περιστροφή που περιγράφει το τετραδόνιο. Για λόγους συντομίας και ευκολίας, ταυτίζουμε τον τελεστή T_q με τον συμβολισμό $**$, ώστε το: $v_{final} = q ** v$ να αναπαριστά το διάνυσμα θέσης που προκύπτει από την περιστροφή του αρχικού διανύσματος v σύμφωνα με το τετραδόνιο q .

- 4) Ορίζουμε τον τελεστή $E(q) = \theta$, όπου θ είναι ένα διάνυσμα-στήλη που περιγράφει μια περιστροφή με γωνίες Euler. Επιπλέον, ορίζεται και ο αντίστροφος τελεστής, έτσι ώστε: $E^{-1}(\theta) = q$, q : είναι το τετραδόνιο που αντιστοιχεί στις γωνίες Euler θ .

1.3 Οπτική Αδρανειακή Οδομετρία – Ταυτόχρονος Εντοπισμός και Χαρτογράφηση

Στην ενότητα αυτή θα αναφερθούμε στις έννοιες της Οπτικής Αδρανειακής Οδομετρίας και του Ταυτόχρονου Εντοπισμού και Χαρτογράφησης. Αποτελούν θεμελιώδεις τεχνολογίες για την ανάπτυξη εφαρμογών Επαυξημένης (AR) και Μεικτής Πραγματικότητας (MR). Χρησιμοποιούνται για την ακριβή παρακολούθηση της θέσης και του προσανατολισμού των συσκευών στον χώρο, επιτρέποντας την τοποθέτηση ψηφιακών αντικειμένων στο φυσικό περιβάλλον με ρεαλιστικό τρόπο. Οι τεχνολογίες αυτές ενισχύουν την εμπειρία του χρήστη, διευκολύνοντας την αλληλεπίδραση με ψηφιακά και φυσικά στοιχεία σε πραγματικό χρόνο.

Παρακάτω, θα εξεταστούν οι βασικές έννοιες της Οδομετρίας, εστιάζοντας στις διαφορετικές προσεγγίσεις της, όπως η Οπτική Οδομετρία και η Αδρανειακή Οδομετρία. Στη συνέχεια, θα αναλυθεί η σύνθεση αυτών των δύο μεθόδων, που οδηγεί στην Οπτική Αδρανειακή Οδομετρία (VIO), αναδεικνύοντας τη σημασία της σε σύγχρονες εφαρμογές.

1.3.1 Οδομετρία (Odometry)

Η οδομετρία είναι η χρήση δεδομένων από αισθητήρες κίνησης για την εκτίμηση της αλλαγής θέσης με την πάροδο του χρόνου. Χρησιμοποιείται στη ρομποτική από ορισμένα ρομπότ με πόδια ή ρόδες για να εκτιμήσουν τη θέση τους σε σχέση με μια αρχική τοποθεσία. Αυτή η μέθοδος είναι ευαίσθητη σε σφάλματα λόγω της ολοκλήρωσης των μετρήσεων ταχύτητας με την πάροδο του χρόνου για την εξαγωγή εκτιμήσεων θέσης. Η ταχεία και ακριβής συλλογή δεδομένων, η βαθμονόμηση των οργάνων και η επεξεργασία είναι απαραίτητες στις περισσότερες περιπτώσεις για την αποτελεσματική χρήση της οδομετρίας [6].

1.3.2 Οπτική Οδομετρία (Visual Odometry - VO) [7] [8]

Η οπτική οδομετρία είναι η διαδικασία προσδιορισμού της κίνησης ενός ρομπότ (μετατόπιση και περιστροφή σε σχέση με ένα σύστημα αναφοράς) μέσω της ανάλυσης μιας διαδοχικής σειράς εικόνων από το περιβάλλον του. Αποτελεί μια ειδική περίπτωση της μεθόδου Δομή Από Κίνηση (Structure From Motion - SFM), η οποία επικεντρώνεται στην τρισδιάστατη ανακατασκευή της δομής του περιβάλλοντος και των θέσεων της κάμερας, είτε από διατεταγμένα είτε από μη διατεταγμένα σύνολα εικόνων. Η τεχνολογία VO χαρακτηρίζεται από μια καλή ισορροπία μεταξύ της πολυπλοκότητας υλοποίησης, της αξιοπιστίας και του κόστους.

Αν και η τεχνολογία VO έχει αποδειχθεί εξαιρετικά αποτελεσματική για τον εντοπισμό και την πλοήγηση ρομπότ σε εσωτερικούς χώρους, η χρήση της σε πολύπλοκα και δυναμικά αγροτικά περιβάλλοντα παραμένει περιορισμένη. Οι βασικές προκλήσεις που αντιμετωπίζουν τα συστήματα VO σχετίζονται κυρίως με τις συνθήκες φωτισμού και απεικόνισης, καθώς και με το υψηλό υπολογιστικό κόστος. Επιπλέον, οι σκιές που

δημιουργούνται από ρομπότ ή οχήματα μπορούν να διαταράξουν τον υπολογισμό της μετατόπισης pixel, οδηγώντας σε εσφαλμένες εκτιμήσεις.

Η τεχνολογία VO μπορεί να ταξινομηθεί ανάλογα με τον τύπο κάμερας ή αισθητήρα δεδομένων που χρησιμοποιείται για την εκτίμηση της τροχιάς του ρομπότ. Διάφοροι τύποι καμερών, όπως στερεοσκοπικές (stereo), μονοφθαλμικές (monocular), στερεοσκοπικές ή μονοφθαλμικές πανοραμικές, καθώς και κάμερες RGB-D, μπορούν να χρησιμοποιηθούν για τους σκοπούς της VO.

Η εκτίμηση της θέσης ενός κινητού ρομπότ με VO μπορεί γενικά να προσεγγιστεί με τρεις τρόπους: μέσω μιας προσέγγισης βασισμένης σε χαρακτηριστικά (feature based approach), μιας προσέγγισης βασισμένης στην εμφάνιση (appearance based approach) ή μιας υβριδικής προσέγγισης (hybrid approach) που συνδυάζει τα προηγούμενα δύο.

- **Feature based approach:** Αφορά την εξαγωγή χαρακτηριστικών εικόνας, όπως γωνίες, γραμμές και καμπύλες, από διαδοχικά καρέ, την αντιστοίχιση ή παρακολούθηση των πιο διακριτών από αυτά και, στη συνέχεια, την εκτίμηση της κίνησης.
- **Appearance based:** Εστιάζει στις αλλαγές στην εμφάνιση των ληφθέντων εικόνων και στην ένταση των πληροφοριών των pixels, χωρίς να βασίζεται στην εξαγωγή και παρακολούθηση χαρακτηριστικών.
- **Hybrid:** Συνδυάζει την παρακολούθηση διακριτών χαρακτηριστικών μεταξύ των καρέ με τη χρήση των πληροφοριών έντασης των pixels είτε ολόκληρης είτε μέρους της εικόνας.

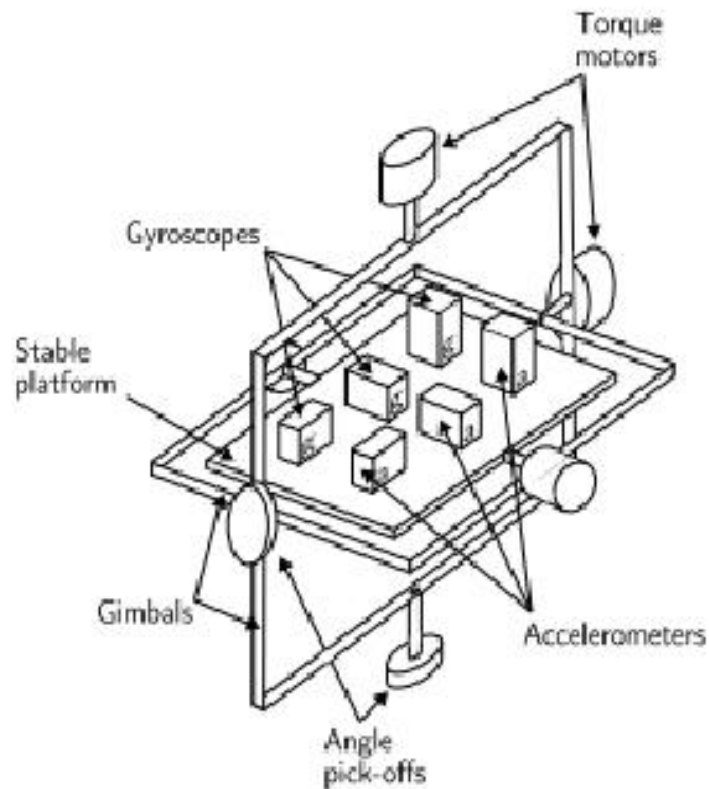
1.3.3 Αδρανειακή Οδομετρία (Inertial Odometry)

Η Αδρανειακή Οδομετρία προέκυψε από την χρήση μίας ή περισσότερων Μονάδων Αδρανειακής Μέτρησης (IMU) για τον υπολογισμό της. Μια IMU χρησιμοποιεί γυροσκόπια (gyroscopes), μαγνητόμετρα (magnetometers - που μετρούν το μαγνητικό πεδίο της Γης) και επιταχυνσιόμετρα (accelerometers) για να υπολογίσει την μετατόπιση, με την ταχύτητα και την περιστροφή [9]. Μία άμεση λύση για τον υπολογισμό της θέσης είναι η αριθμητική ολοκλήρωση της επιτάχυνσης για την εύρεση της ταχύτητας και, στη συνέχεια, η ολοκλήρωση της ταχύτητας για την εύρεση της θέσης. Αν και αυτή η προσέγγιση είναι θεωρητικά σωστή, δεν λαμβάνει υπόψη τον θόρυβο που εισάγουν οι αισθητήρες, ειδικά τα επιταχυνσιόμετρα.

Ο θόρυβος αυτός αποτελεί εγγενές χαρακτηριστικό των αισθητήρων αδράνειας, ακόμη και των πιο ακριβών, και οδηγεί στη συσσώρευση σφαλμάτων κατά την ολοκλήρωση. Το πρόβλημα γίνεται εντονότερο καθώς ο θόρυβος ολοκληρώνεται και μάλιστα στο τετράγωνο, προκαλώντας το φαινόμενο της ολίσθησης (drift). Αυτό σημαίνει ότι ενώ στην αρχή η θέση φαίνεται να υπολογίζεται σωστά, με την πάροδο του χρόνου εμφανίζονται αποκλίσεις, οι οποίες αυξάνονται συνεχώς, οδηγώντας σε λάθος αποτελέσματα.

Παρά τα μειονεκτήματα αυτά, η αδρανειακή οδομετρία παραμένει εξαιρετικά χρήσιμη, ειδικά σε περιβάλλοντα όπου το GPS ή άλλες εξωτερικές πηγές σήματος δεν είναι διαθέσιμες, όπως σε υπόγεια, υποθαλάσσια ή εσωτερικούς χώρους. Χρησιμοποιείται σε

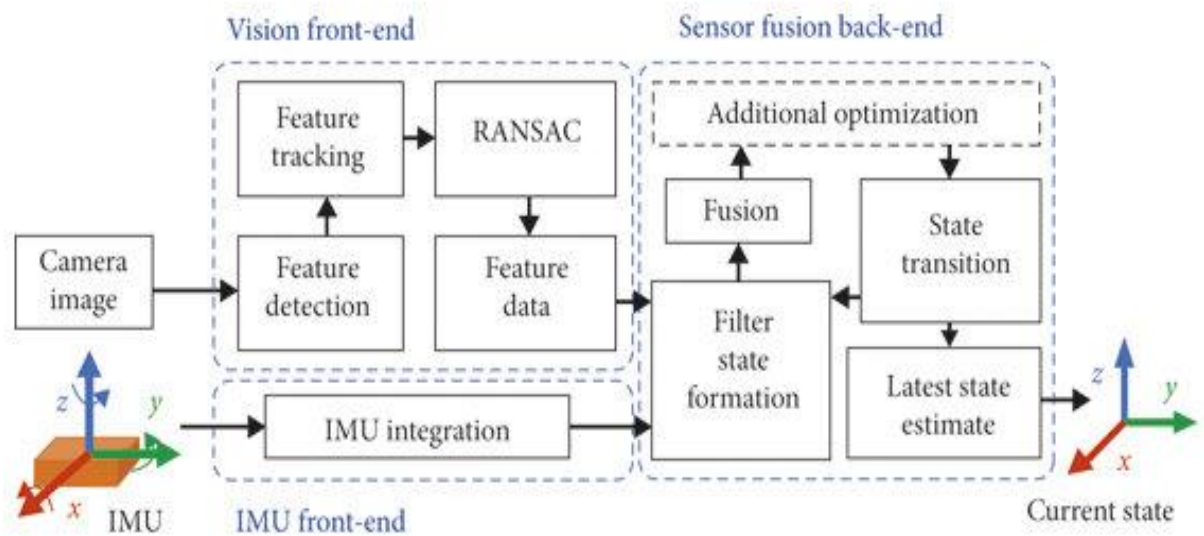
πολλούς τομείς, όπως η ρομποτική, η επαυξημένη και εικονική πραγματικότητα, τα αυτόνομα οχήματα και οι φορητές συσκευές. Για τη μείωση του φαινομένου της ολίσθησης, συνήθως συνδυάζεται με άλλες τεχνολογίες ή αλγορίθμους διόρθωσης σφαλμάτων, βελτιώνοντας την ακρίβεια της θέσης και του προσανατολισμού.



Εικόνα 5: Inertial Measurement Unit - IMU

1.3.4 Οπτική Αδρανειακή Οδομετρία (Visual Inertial Odometry - VIO)

Η Οπτική Αδρανειακή Οδομετρία, όπως υποδηλώνει και το όνομά της, συνδυάζει την Οπτική Οδομετρία με την Αδρανειακή Πλοήγηση, με στόχο την αντιμετώπιση των αδυναμιών που παρουσιάζουν οι δύο αυτές τεχνικές όταν χρησιμοποιούνται ξεχωριστά. Το VIO (Visual Inertial Odometry) βασίζεται στην Οπτική Οδομετρία για την εκτίμηση της θέσης του οχήματος μέσω εικόνων κάμερας, ενώ ταυτόχρονα αξιοποιεί τις αδρανειακές μετρήσεις από το IMU του οχήματος για τη διόρθωση σφαλμάτων που προκαλούνται από την ταχεία κίνηση, η οποία μπορεί να επηρεάσει την ποιότητα των λήψεων [10].



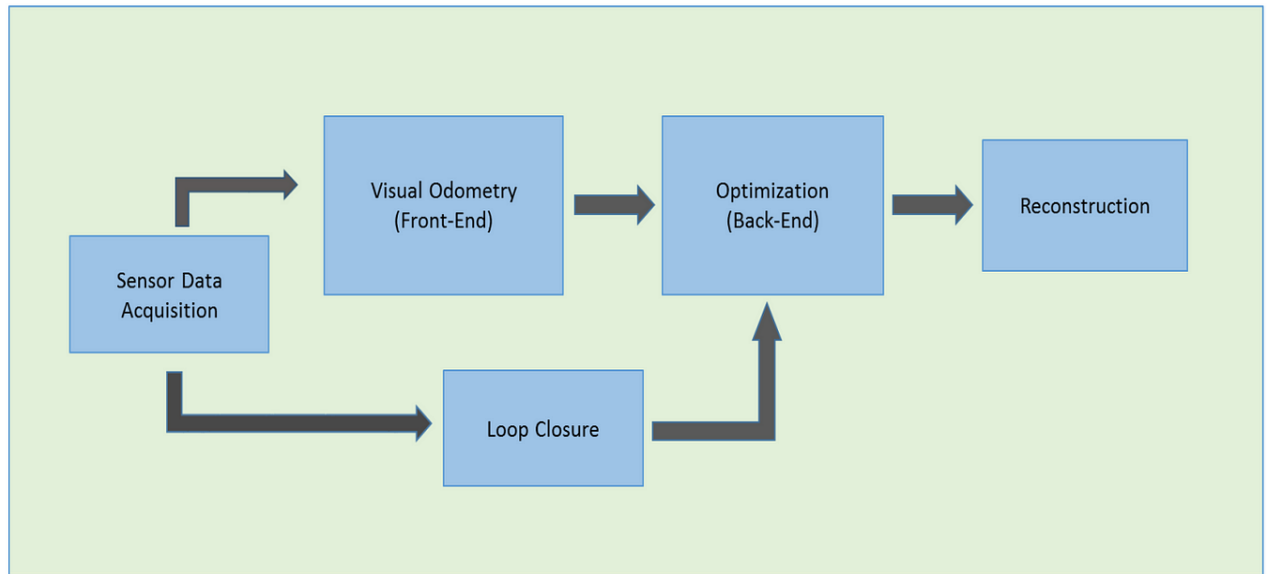
Εικόνα 6: Overview of VIO architecture

1.3.5 Ταυτόχρονος Εντοπισμός και Χαρτογράφηση (Simultaneous Localization And Mapping – SLAM)

Το SLAM (Simultaneous Localization and Mapping) είναι μια διαδικασία κατά την οποία ένα ρομπότ καλείται να εντοπίσει τη θέση του σε ένα άγνωστο περιβάλλον και ταυτόχρονα να δημιουργήσει έναν χάρτη αυτού του περιβάλλοντος, χωρίς να διαθέτει καμία προηγούμενη πληροφορία, με τη βοήθεια εξωτερικών αισθητήρων (ή ενός μόνο αισθητήρα).

Το V-SLAM (Visual Simultaneous Localization and Mapping) επεκτείνει την έννοια της Οπτικής Οδομετρίας (VO) κατασκευάζοντας ταυτόχρονα έναν χάρτη του περιβάλλοντος και εντοπίζοντας τη θέση του ρομπότ μέσα σε αυτόν τον χάρτη. Αντιμετωπίζει το πρόβλημα της απόκλισης (drift) μέσω της ενσωμάτωσης τεχνικών loop closure, οι οποίες ανιχνεύουν πότε το ρομπότ επισκέπτεται ξανά μια ήδη χαρτογραφημένη περιοχή και διορθώνουν τα συσσωρευμένα σφάλματα στις εκτιμήσεις του χάρτη και της πορείας. Μερικά βασικά χαρακτηριστικά του v-SLAM είναι:

- **Σύντηξη Αισθητήρων (Sensor Fusion):** Συνδυασμός δεδομένων από πολλούς αισθητήρες (π.χ. κάμερες, IMUs) για βελτίωση της ακρίβειας.
- **Εξαγωγή και Αντιστοίχιση Χαρακτηριστικών (Feature Extraction and Matching):** Εξαγωγή ανθεκτικών χαρακτηριστικών και αντιστοίχισή τους μεταξύ καρέ για τη διατήρηση της συνοχής του χάρτη.
- **Βελτιστοποίηση (Optimization):** Χρήση τεχνικών βελτιστοποίησης, όπως η προσαρμογή δέσμης (bundle adjustment) και η βελτιστοποίηση γραφήματος θέσης (pose graph optimization), για τη βελτίωση των εκτιμήσεων του χάρτη και της θέσης.

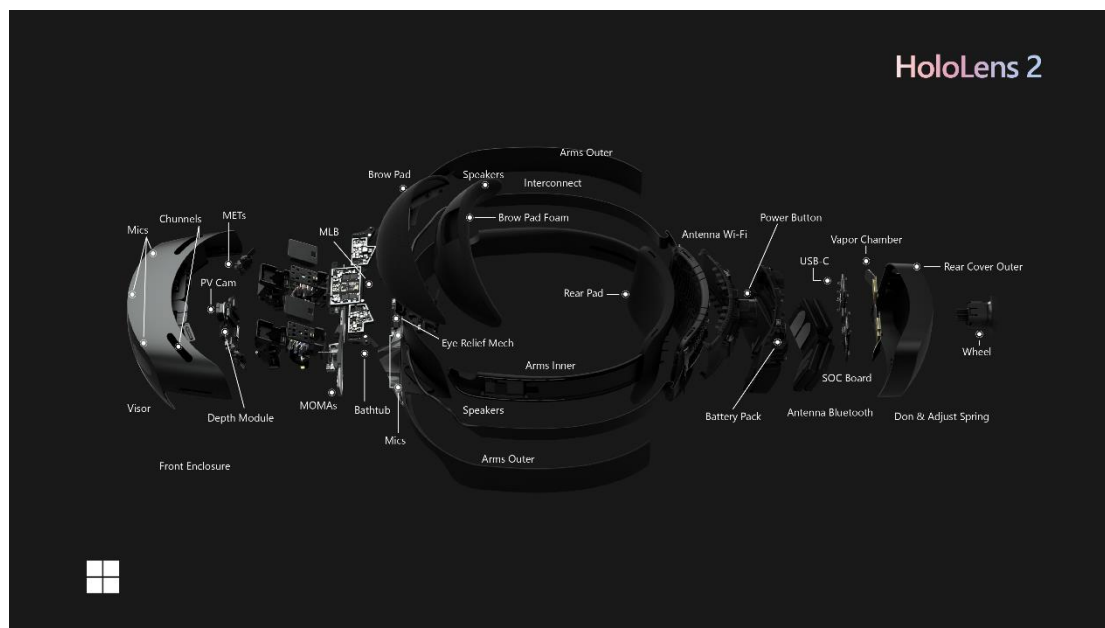


Εικόνα 7: Block διάγραμμα του v-SLAM

2 Τεχνολογίες

Η υλοποίηση της διπλωματικής έγινε πάνω στα γυαλιά μικτής πραγματικότητας της Microsoft, τα Microsoft HoloLens 2. Η δημιουργία της εφαρμογής έγινε στο περιβάλλον της Unity 2022 και η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η C#, στο Visual Studio 2022.

2.1 Microsoft HoloLens 2



Εικόνα 8: Εξαρτήματα του HoloLens 2

Το Microsoft HoloLens 2 είναι μια προηγμένη συσκευή επαυξημένης πραγματικότητας, η οποία προφανώς παράγεται από την Microsoft. Συνδυάζει εξελεγχμένες δυνατότητες ολογραφικής απεικόνισης με βελτιωμένη εργονομία και άνεση. Βασισμένο στο αρχικό HoloLens, το HoloLens 2 επιτρέπει στους χρήστες να αλληλεπιδρούν με τρισδιάστατα ολογράμματα σε πραγματικό χρόνο, προσφέροντας ανεπανάληπτες δυνατότητες για εφαρμογές στην εκπαίδευση, την υγειονομική περίθαλψη, τη βιομηχανία και άλλους τομείς. Η συσκευή διαθέτει επίσης βελτιωμένο οπτικό πεδίο και προηγμένη τεχνολογία ανίχνευσης κίνησης χεριών, κάνοντας την εμπειρία πιο φυσική και ρεαλιστική για τους χρήστες [11].

2.1.1 Βασικά Χαρακτηριστικά

Το Microsoft HoloLens 2 λειτουργεί με το **Windows** Holographic OS, μια ειδική έκδοση των Windows 10 που είναι σχεδιασμένη για επαυξημένη και μεικτή πραγματικότητα. Το Windows Holographic υποστηρίζει την αλληλεπίδραση με 3D ολογράμματα, την ενσωμάτωση εφαρμογών επαυξημένης πραγματικότητας και τη χρήση του Microsoft Mixed Reality Toolkit, επιτρέποντας στους προγραμματιστές να δημιουργούν εφαρμογές που εκμεταλλεύονται τις δυνατότητες της πλατφόρμας HoloLens 2.

Το HoloLens 2 διαθέτει τα εξής κύρια μέρη, που βελτιώνουν την άνεση και τη λειτουργικότητα της συσκευής:

- **Visor (Προσωπίδα):** Η προσωπίδα περιλαμβάνει τους αισθητήρες και τις οθόνες, ενώ μπορεί να περιστραφεί ακόμη και όταν ο χρήστης φοράει τη συσκευή, παρέχοντας μεγαλύτερη ευελιξία.
- **Headband (Κεφαλόδεσμος):** Ένας ιμάντας που περιβάλλει το κεφάλι και στηρίζει τη συσκευή. Ο χρήστης μπορεί να ρυθμίσει την έντασή του με μια ροδέλα στο πίσω μέρος, ώστε να προσαρμόζεται άνετα.
- **Brightness buttons (Κουμπιά φωτεινότητας):** Τα κουμπιά ρύθμισης φωτεινότητας βρίσκονται στην αριστερή πλευρά του κεφαλόδεσμου, κοντά στον κρόταφο του χρήστη για εύκολη πρόσβαση.
- **Volume buttons (Κουμπιά έντασης):** Τα κουμπιά ελέγχου έντασης του ήχου είναι τοποθετημένα στη δεξιά πλευρά του κεφαλόδεσμου, επίσης κοντά στον κρόταφο του χρήστη.
- **Power button (Κουμπί ενεργοποίησης):** Το κουμπί που ενεργοποιεί ή απενεργοποιεί τη συσκευή βρίσκεται στην πίσω δεξιά πλευρά του εξωτερικού καλύμματος.
- **USB-C port (Θύρα USB-C):** Κάτω από το κουμπί ενεργοποίησης υπάρχει μια θύρα USB-C, μέσω της οποίας ο χρήστης μπορεί να συνδέσει τη συσκευή στον υπολογιστή του για να μεταφέρει εφαρμογές ή δεδομένα.

2.1.2 Αισθητήρες (Sensors) της συσκευής

Το **HoloLens 2** διαθέτει μια σειρά από αισθητήρες που συνεργάζονται για να παρέχουν ανώτερες δυνατότητες ανίχνευσης και αλληλεπίδρασης στον χώρο, υλοποιώντας τις τεχνικές υπολογιστικής όρασης και οδομετρίας.

1. Ανίχνευση Κεφαλής (Head Tracking)

- Αισθητήρες:** 4 κάμερες ορατού φωτός
- Λειτουργία:** Οι κάμερες αυτές παρακολουθούν τη θέση και την κίνηση της κεφαλής του χρήστη, επιτρέποντας την ακριβή ανίχνευση προσανατολισμού και την τοποθέτηση εικονικών αντικειμένων στον χώρο.

2. Ανίχνευση Ματιών (Eye Tracking)

- Αισθητήρες:** 2 υπέρυθρες (IR) κάμερες
- Λειτουργία:** Οι IR κάμερες ανιχνεύουν την κίνηση των ματιών, υποστηρίζοντας αλληλεπιδράσεις με βάση το βλέμμα, όπως επιλογή αντικειμένων ή προσαρμογή της εστίασης.

3. Αισθητήρας Βάθους (Depth Sensor)

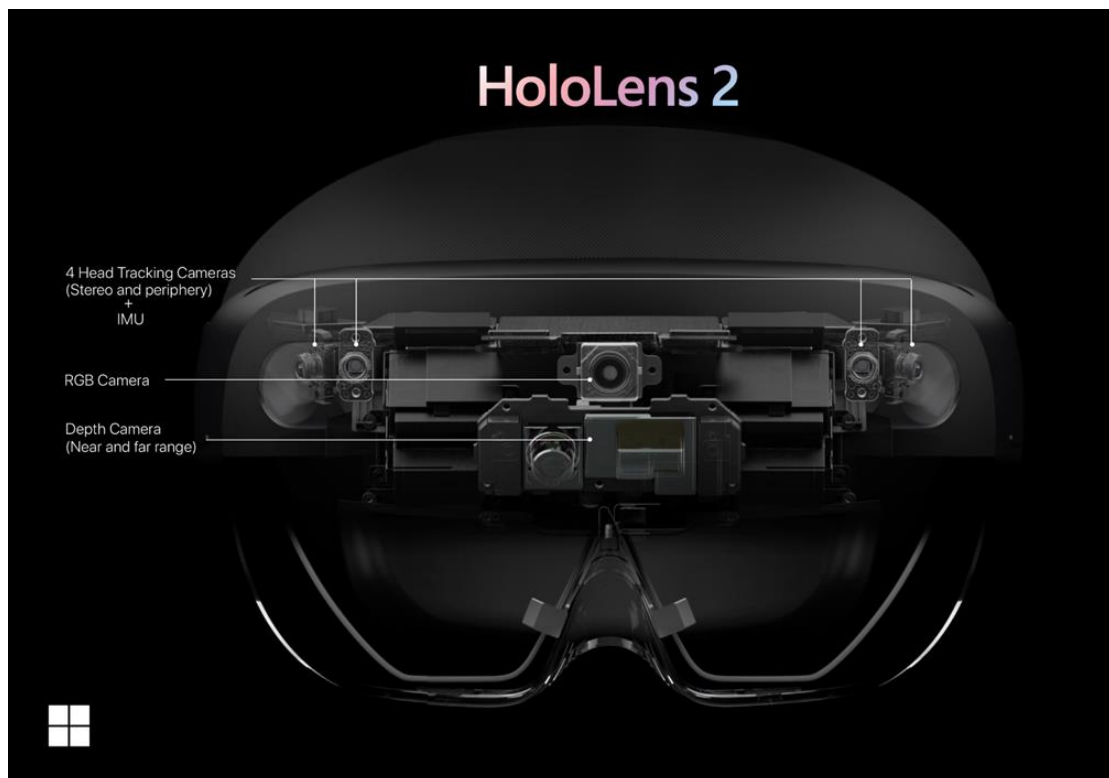
- Αισθητήρας:** 1-MP Time-of-Flight (ToF) αισθητήρας
- Λειτουργία:** Μετρά την απόσταση μεταξύ του αισθητήρα και των αντικειμένων στον χώρο σε πραγματικό χρόνο. Είναι κρίσιμος για την

ανίχνευση χεριών, την ανάλυση του περιβάλλοντος και τη δημιουργία 3D μοντέλων του χώρου.

4. Μονάδα Αδρανειακής Μέτρησης (IMU)

- a. **Αισθητήρες:** Επιταχυνσιόμετρο, γυροσκόπιο και μαγνητόμετρο
- b. **Λειτουργία:** Ανιχνεύει επιταχύνσεις, περιστροφικές κινήσεις και την κατεύθυνση του HoloLens, παρέχοντας δεδομένα για ακριβή προσανατολισμό και σταθερότητα της προβολής.

5. **Κάμερα (Camera):** 8 MP για φωτογραφίες, εγγραφή βίντεο 1080p στα 30 fps



Εικόνα 9: Αισθητήρες του HoloLens 2

2.1.3 Τρόποι χειρισμού (ελέγχου) του HoloLens 2

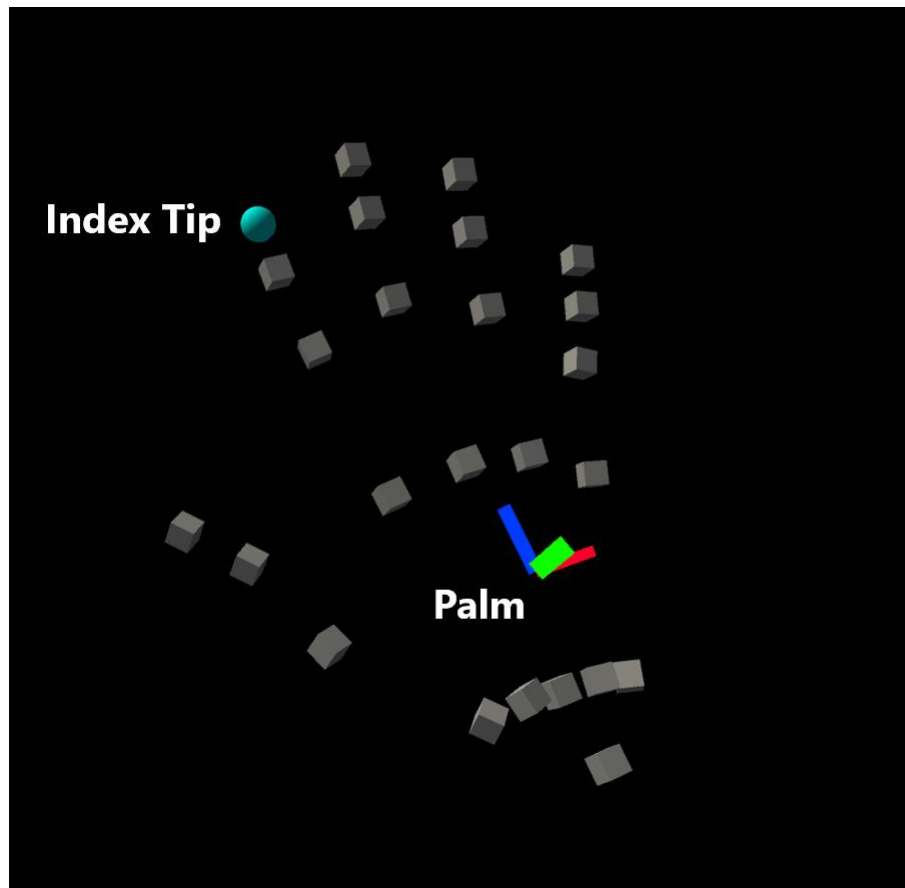
Το HoloLens 2 προσφέρει πολλαπλούς τρόπους αλληλεπίδρασης που επιτρέπουν στον χρήστη να χειρίζεται το εικονικό περιβάλλον με φυσικό και διαισθητικό τρόπο. Η συσκευή επικεντρώνεται στη χρήση των χεριών, της φωνής και της όρασης, μειώνοντας την ανάγκη για παραδοσιακά εργαλεία εισόδου, όπως το πληκτρολόγιο ή το ποντίκι. Ακολουθούν οι βασικοί τρόποι αλληλεπίδρασης [12] [13]:

1) **Ανίχνευση βλέμματος (Gaze):** Το «βλέμμα» του χρήστη αναπαρίσταται από μια εικονική κουκίδα που εμφανίζεται μέσα στο οπτικό του πεδίο. Η κουκίδα αυτή, η οποία μπορεί να θεωρηθεί ανάλογη με τον κέρσορα του υπολογιστή, ελέγχεται μέσω δύο διαφορετικών μηχανισμών:

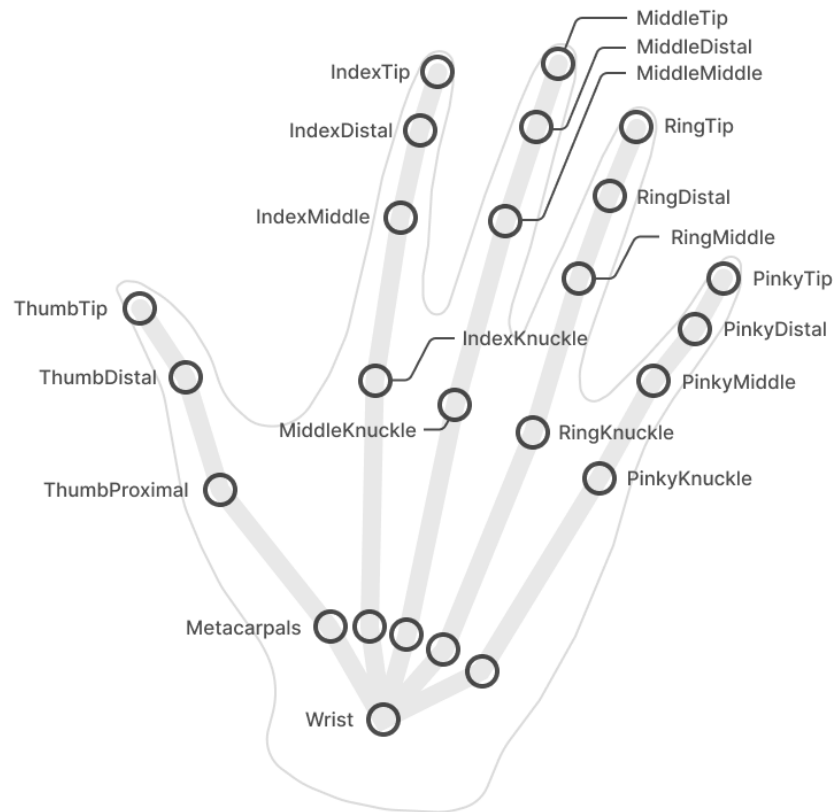
- **Βλέμμα μέσω κεφαλιού (Head tracking):** Ο δείκτης στο οπτικό πεδίο του χρήστη κινείται καθώς ο χρήστης γυρνά το κεφάλι του, προσφέροντας μια προσέγγιση παρόμοια με τον έλεγχο του ποντικιού.
- **Βλέμμα μέσω ματιών (Eye tracking):** Η συσκευή παρακολουθεί την κίνηση των ματιών, επιτρέποντας μεγαλύτερη ακρίβεια και ταχύτερη αλληλεπίδραση. Για να αποφεύγεται η κόπωση, η Microsoft συστήνει να μην εμφανίζεται συνεχώς ο δείκτης όταν βασίζεται στα μάτια. Αυτή η μέθοδος προσφέρει υψηλή ταχύτητα, χαμηλή προσπάθεια, υποστήριξη οπτικής προσοχής και καλύτερη συνεργασία με άλλα μέσα εισόδου. Ωστόσο, έχοντας πάντα ενεργό βλέμμα, μπορεί να πραγματοποιηθούν λανθασμένες ενέργειες, υπάρχει δυσκολία στον ακριβή έλεγχο, προβλήματα με μικρούς στόχους και πιθανή αναξιοπιστία παρακολούθησης σε μεταβαλλόμενο φωτισμό.

2) **Ανίχνευση χειρών (Hand Tracking):** Η ανίχνευση χειρών είναι μία από τις πιο πρωτοποριακές δυνατότητες του HoloLens 2, επιτρέποντας στον χρήστη να αλληλεπιδρά με φυσικό και διαισθητικό τρόπο με τα εικονικά αντικείμενα.

Το HoloLens 2 χρησιμοποιεί κάμερες υψηλής ακρίβειας και τεχνολογίες υπολογιστικής όρασης σε συνδυασμό με αλγορίθμους μηχανικής εκμάθησης. Με αυτές τις τεχνολογίες, η συσκευή χαρτογραφεί τη θέση και την κίνηση των χειρών του χρήστη σε πραγματικό χρόνο. Όταν ανιχνεύονται χέρια, το HoloLens δημιουργεί έναν εικονικό σκελετό, δηλαδή ένα σύνολο σημείων που αναπαριστά τη δομή και τις κινήσεις του χεριού στον τρισδιάστατο χώρο [14] [15].



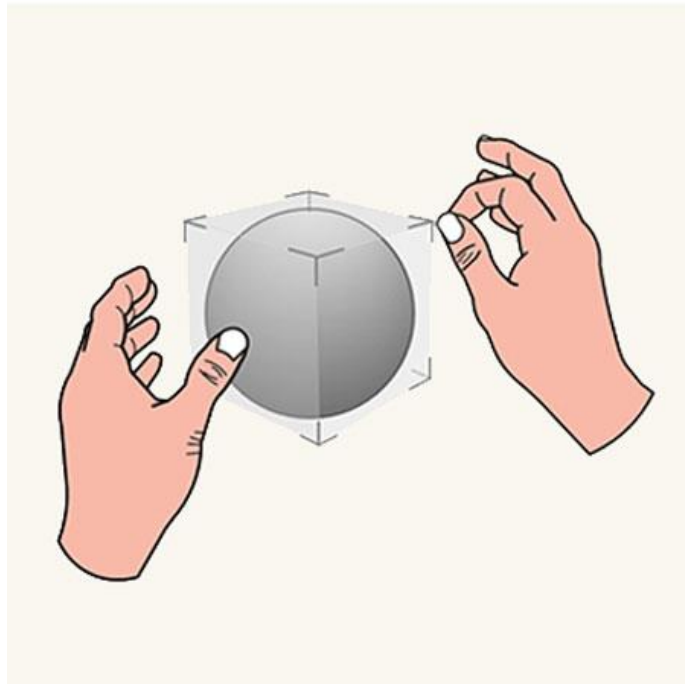
Εικόνα 10: Εικονική αναπαράσταση χεριού



Εικόνα 11: Ονομαστικές ετικέτες

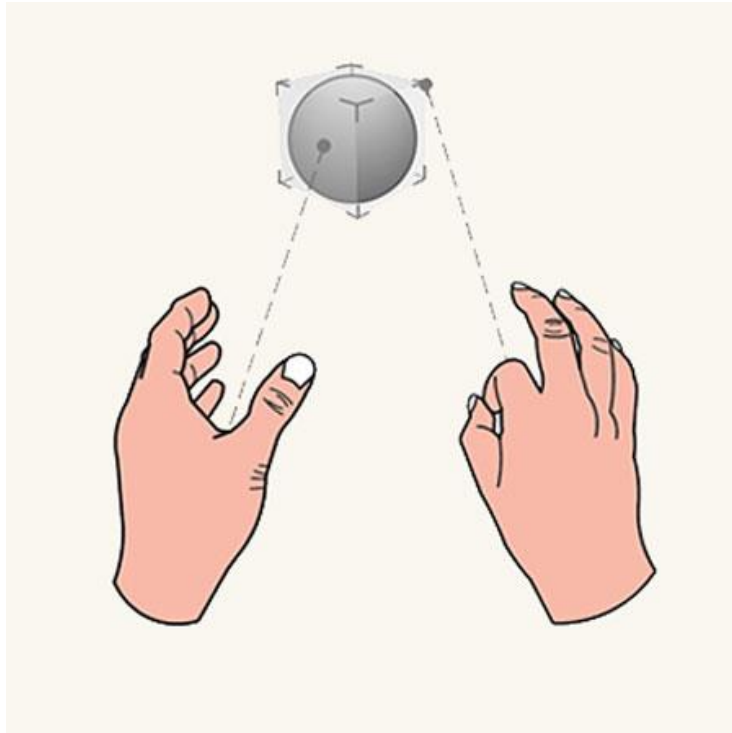
Υπάρχουν οι εξής τύποι αλληλεπίδρασης με τα χέρια:

- **Κοντινή αλληλεπίδραση (Near Interaction):** Ο χειρισμός από κοντά απαιτεί το εικονικό αντικείμενο να βρίσκεται σε απόσταση χεριού από τον χρήστη, περίπου 50 cm. Τότε, οι ακτίνες απενεργοποιούνται αυτόματα και ο χρήστης μπορεί να πιάσει το αντικείμενο, να το μετακινήσει, να το περιστρέψει και να αλλάξει το μέγεθός του.



Εικόνα 12: Near interaction

- **Μακρινή αλληλεπίδραση (Far interaction):** Ο χειρισμός από μακριά, προφανώς, λειτουργεί αντίθετα από τον χειρισμό από κοντά. Όταν το αντικείμενο βρίσκεται σε απόσταση 50 cm από τον χρήστη, οι ακτίνες ενεργοποιούνται και επιτρέπεται η αλληλεπίδραση με το εικονικό αυτό αντικείμενο.

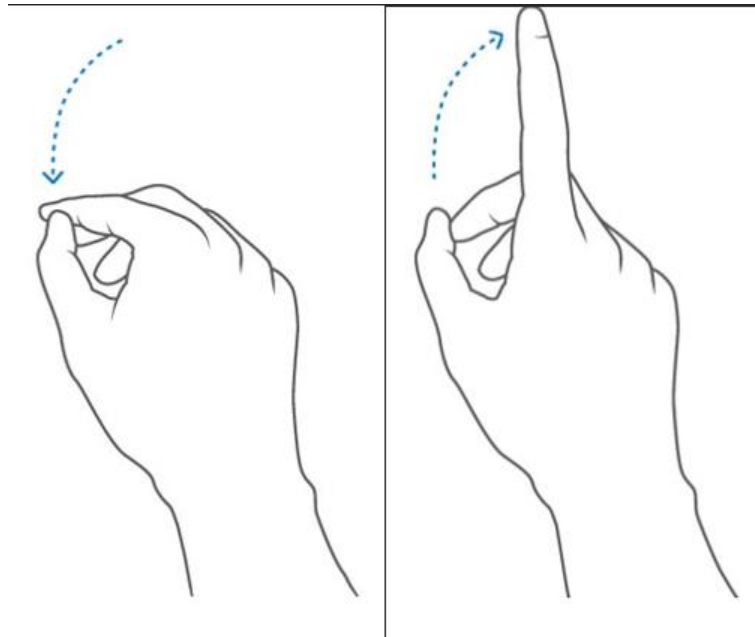


Εικόνα 13: Far interaction

Η εναλλαγή μεταξύ των δύο αλληλεπιδράσεων είναι ομαλή και γίνεται είτε πλησιάζοντας το εικονικό αντικείμενο, είτε αλλάζοντας την θέση του στο χώρο μέσω χειρισμών με τα χέρια. Επιπλέον, οι δύο τύποι χειρισμών περιλαμβάνουν τα ίδια gestures που θα αναφερθούν παρακάτω.

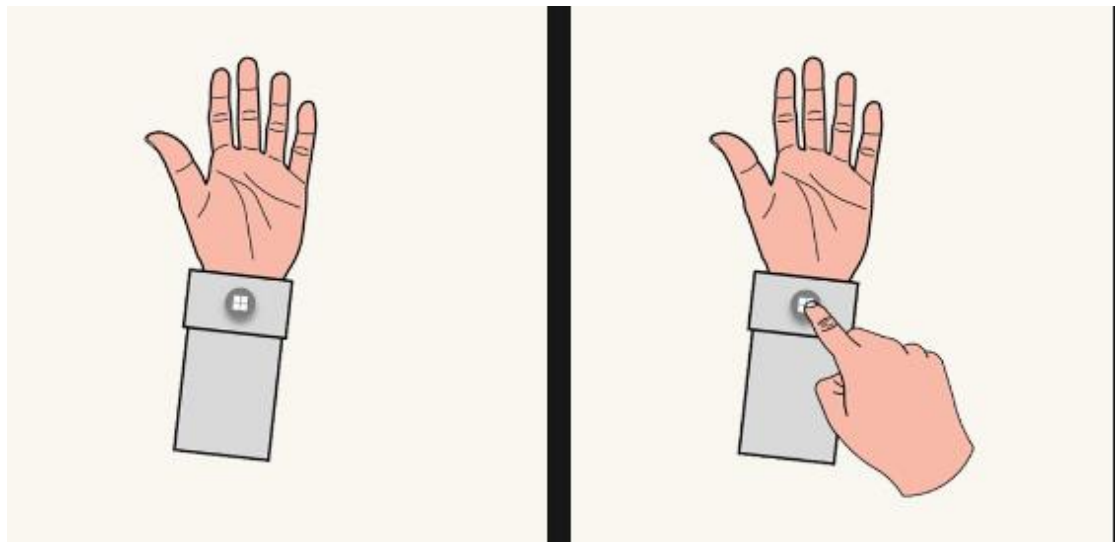
3) **Χειρονομίες (Gestures):** Μέσω του hand tracking, το hololens 2 μπορεί και ανιχνεύει χειρονομίες. Υπάρχουν δύο βασικές χειρονομίες που ανιχνεύει από μόνη της η συσκευή, το **Air-Tap** και το **Start Gesture**. Ωστόσο, είναι εφικτό, ο χρήστης να επεκτείνει την αναγνώριση με επιπλέον χειρονομίες [16] [17] [18].

- Η χειρονομία **Air-Tap** επιτρέπει την επιλογή αντικειμένων και αντιστοιχεί στο αριστερό κλικ ενός ποντικιού σε έναν υπολογιστή. Για να πραγματοποιήσετε το Air-Tap, κρατήστε το χέρι σας μπροστά σας σε χαλαρή γροθιά, δείχνοντας τον δείκτη σας προς το ταβάνι, λυγίστε γρήγορα τον δείκτη προς τα κάτω και επαναφέρετέ τον στην αρχική θέση. Επιπλέον, μπορείτε να κάνετε και το λεγόμενο **Tap and hold**, πραγματοποιώντας την κίνηση του tap αλλά κρατώντας το δάχτυλό σας κάτω, αντί να το σηκώνετε, όπως πριν.

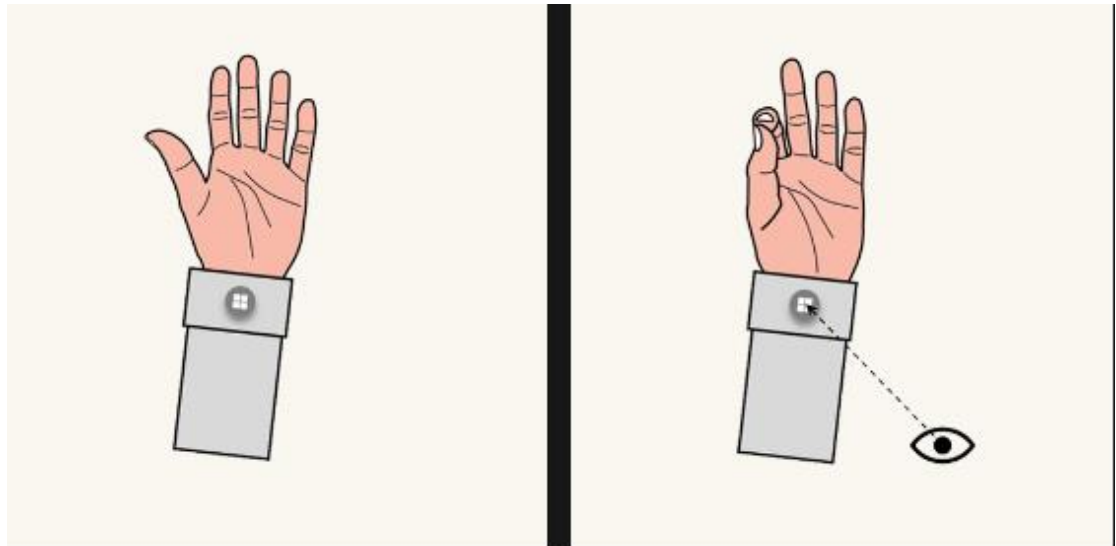


Εικόνα 14: Air Tap Gesture

- Το **Start Gesture** εμφανίζει το μενού έναρξης. Για να ανοίξετε το κύριο μενού, σηκώστε τον καρπό σας προς τις κάμερες. Θα εμφανιστεί το εικονίδιο του μενού έναρξης της Microsoft. Χρησιμοποιήστε το άλλο σας χέρι για να το επιλέξετε. Μάλιστα, υπάρχει και η δυνατότητα επιλογής με το ένα χέρι όταν ο χρήστης κοιτάξει το εικονίδιο στον καρπό του και ακουμπήσει τον δείκτη με τον αντίχειρά του (παρόμοια κίνηση με το air tap).



Εικόνα 15: Two handed Start Gesture



Εικόνα 16: One handed Start Gesture

- 4) Φωνητικές εντολές (Voice commands) [19]:** Τα Hololens 2 παρέχουν την δυνατότητα αναγνώρισης φωνής για την χρήση φωνητικών εντολών σε πληθώρα διαφορετικών γλωσσών. Οι ηχητικές εντολές μπορούν να αντικαταστήσουν όλες τις ενέργειες που θα πραγματοποιούσε ο χρήστης με τα χέρια του. Μερικές βασικές εντολές δίνονται στον παρακάτω πίνακα:

Φωνητική εντολή	Ενέργεια
Select	Say "select" to bring up the gaze cursor. Then, turn your head to position the cursor on the thing you want to select, and say "select" again.
Go to start	Open the Start menu
Close	Open the Start menu
Hide/Show hand ray	Hide and show hand ray
What can I say?	See available speech commands

2.2 Unity



Εικόνα 17: Unity logo

Η Unity είναι μια game engine που επιτρέπει την ανάπτυξη δισδιάστατων (2D) και τρισδιάστατων (3D) εφαρμογών. Κυκλοφόρησε το 2005 από τη Unity Technologies και γρήγορα έγινε ιδιαίτερα δημοφιλής. Η επιτυχία της οφείλεται κυρίως στην ικανότητά της να υποστηρίζει ανάπτυξη λογισμικού για πολλές διαφορετικές πλατφόρμες, όπως υπολογιστές με λειτουργικό σύστημα Windows, MacOS και Linux, καθώς και κινητές συσκευές με Android και iOS. Επίσης, υποστηρίζει την ανάπτυξη εφαρμογών για το οικοσύστημα UWP (Universal Windows Platform), στο οποίο περιλαμβάνονται και τα HoloLens της Microsoft [20].

Ένα από τα μεγαλύτερα πλεονεκτήματα της Unity είναι η δυνατότητα μεταφοράς του ίδιου κώδικα σε διαφορετικές πλατφόρμες με ελάχιστες ή καθόλου αλλαγές. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν εφαρμογές που μπορούν να λειτουργούν σε πολλαπλές συσκευές χωρίς την ανάγκη επανασχεδιασμού ή προσαρμογής του κώδικα για κάθε λειτουργικό σύστημα ξεχωριστά.

Η μηχανή είναι γραμμένη σε C++ για το backend-runtime, ενώ οι προγραμματιστές χρησιμοποιούν κυρίως τη γλώσσα C# για την ανάπτυξη των scripts. Η C# είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού, η οποία είναι ιδιαίτερα οικεία στους περισσότερους προγραμματιστές λόγω της συγγένειάς της με τη C.

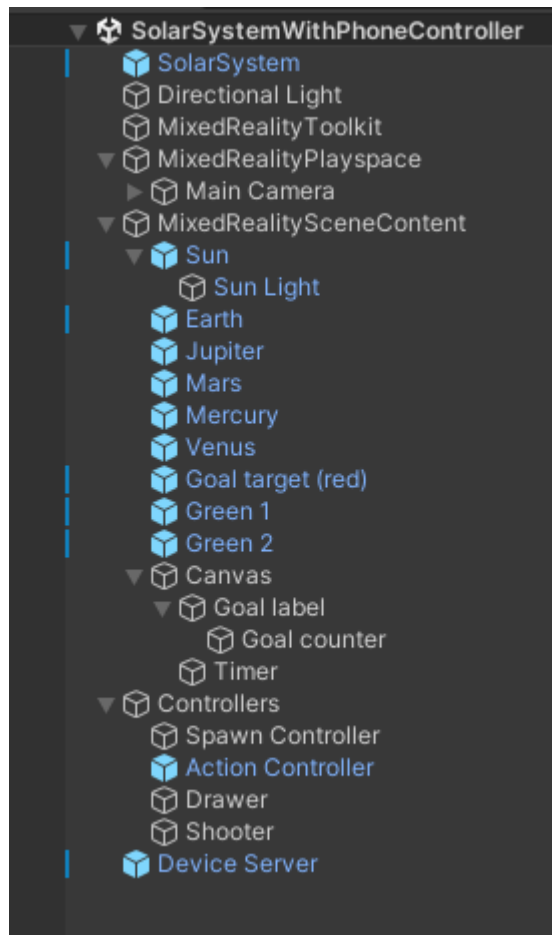
Επιπλέον, η Unity διαθέτει εκτενές documentation και ένα πολύ ισχυρό community, που επιτρέπει στους προγραμματιστές να ανταλλάσσουν γνώσεις και να βρίσκουν λύσεις σε προβλήματα που αντιμετωπίζουν. Λόγω αυτών των χαρακτηριστικών, έχει χρησιμοποιηθεί για την ανάπτυξη πολλών δημοφιλών εφαρμογών και παιχνιδιών, τόσο σε υπολογιστές όσο και σε κινητές συσκευές. Η Microsoft έχει δείξει ξεκάθαρα την προτίμησή της στη Unity για την ανάπτυξη εφαρμογών Mixed Reality, κάτι που αποδεικνύεται και από την κυκλοφορία του Mixed Reality Toolkit (MRTK), το οποίο παρέχει εργαλεία ειδικά προσαρμοσμένα για εφαρμογές επαυξημένης πραγματικότητας στα HoloLens [21] [22].

2.2.1 Βασικά δομικά στοιχεία της Unity

Η Unity είναι μια ισχυρή πλατφόρμα ανάπτυξης που επιτρέπει τη δημιουργία βιντεοπαιχνιδιών και γενικότερων εφαρμογών, προσφέροντας ένα περιβάλλον υψηλού επιπέδου με πολλές έτοιμες λειτουργίες. Οι προγραμματιστές δεν χρειάζεται να υλοποιήσουν από την αρχή μηχανισμούς γραφικών ή μαθηματικών τρισδιάστατου χώρου, καθώς η Unity παρέχει έτοιμα αντικείμενα (objects) και components, όπως φυσικές ιδιότητες και συμπεριφορές, που μπορούν να προστεθούν εύκολα. Αυτό επιτρέπει ακόμα και σε αρχάριους με βασικές γνώσεις προγραμματισμού να δημιουργήσουν απλές εφαρμογές, ενώ για πιο σύνθετα έργα απαιτείται η ανάπτυξη κώδικα, κυρίως σε C#, καθιστώντας την πλατφόρμα προσιτή τόσο σε αρχάριους όσο και σε έμπειρους προγραμματιστές.

Στην Unity, τα **GameObjects** είναι οι θεμελιώδεις μονάδες κάθε σκηνής και αντιπροσωπεύουν όλα τα αντικείμενα μέσα στο παιχνίδι ή την εφαρμογή. Ένα GameObject μπορεί να είναι οτιδήποτε, όπως ένας χαρακτήρας, ένα φως, μια κάμερα ή ένα απλό κενό αντικείμενο που λειτουργεί ως κοντέινερ για άλλα στοιχεία. Τα GameObjects από μόνα τους δεν έχουν κάποια ιδιαίτερη συμπεριφορά, αλλά αποκτούν λειτουργικότητα μέσω των **C# scripts** που γράφει ο προγραμματιστής. Ένα απλό C# script που τοποθετείται πάνω σε ένα gameobject κληρονομεί την κλάση **MonoBehavior** η οποία παρέχει μερικές πολύ βασικές λειτουργίες, όπως η διαχείριση συμβάντων του κύκλου ζωής (life cycle) ενός αντικειμένου. Όταν ένα script τοποθετείται πάνω σε ένα αντικείμενο, τότε αυτό αποκτάει ένα **Component**. Υπάρχουν πολλών ειδών components τα οποία περιγράφουν φυσική συμπεριφορά, το υλικό του αντικειμένου, απόδοση γραφικών, αλληλεπίδραση με το περιβάλλον και άλλα. Κάθε gameobject περιέχει πάντα τουλάχιστον ένα βασικό component, το **Transform**, το οποίο καθορίζει τη θέση, την περιστροφή και την κλίμακά του στη σκηνή [23] [24] [25] [26].

Επιπλέον, τα GameObjects μπορούν να οργανωθούν σε **ιεραρχίες (hierarchies)**, όπου ένα GameObject μπορεί να γίνει **γονέας (parent)** άλλων αντικειμένων, που ονομάζονται **παιδιά (children)**. Αυτό επιτρέπει τη δημιουργία πολύπλοκων δομών, όπως χαρακτήρες με μέρη του σώματος που κινούνται συνδεδεμένα ή UI στοιχεία που μετακινούνται μαζί.



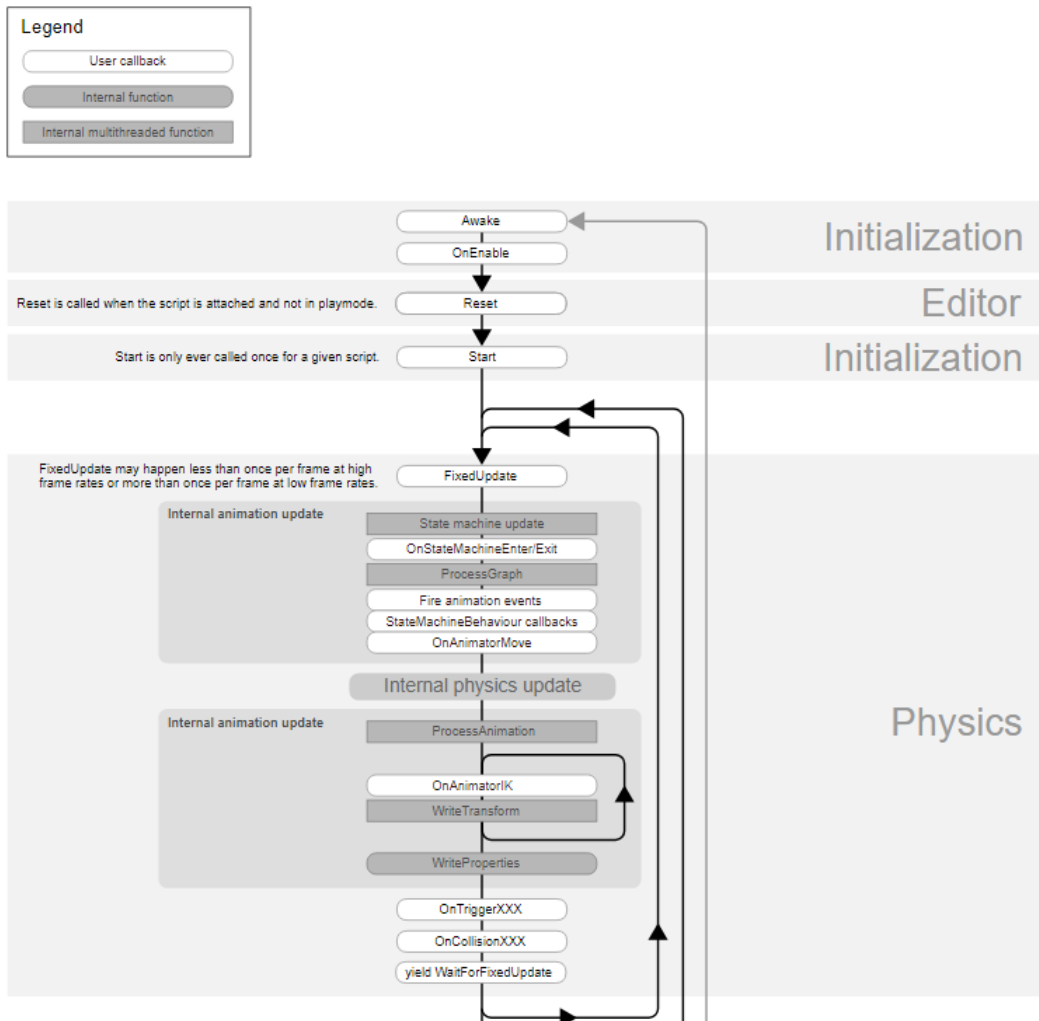
Εικόνα 18: GameObject hierarchy

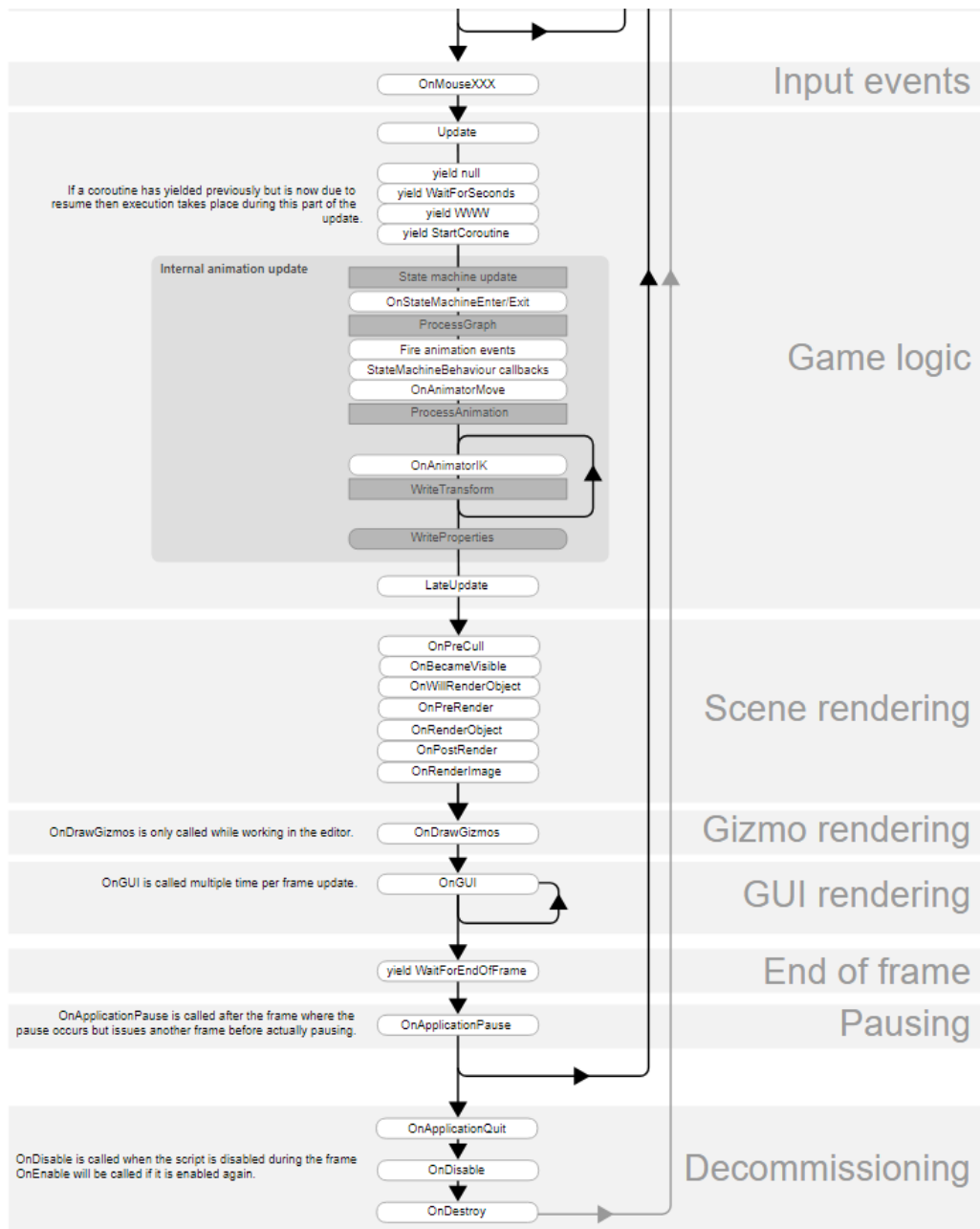
2.2.2 Κύκλος ζωής μιας εφαρμογής – Βασικές μέθοδοι

Ο **κύκλος ζωής (Life Cycle)** ενός **MonoBehaviour** στην Unity περιγράφει τη σειρά με την οποία εκτελούνται οι μέθοδοι ενός script από τη στιγμή που δημιουργείται ένα **GameObject** μέχρι να καταστραφεί. Οι πιο βασικές μέθοδοι του κύκλου ζωής είναι [27]:

1. **Awake()** – Εκτελείται μία φορά μόλις δημιουργηθεί το αντικείμενο, πριν ακόμα ενεργοποιηθεί. Χρησιμοποιείται για αρχικοποιήσεις που δεν εξαρτώνται από άλλα αντικείμενα.
2. **OnEnable()** – Εκτελείται κάθε φορά που το GameObject ή το script ενεργοποιείται.
3. **Start()** – Εκτελείται μία φορά **στο πρώτο frame** μετά την ενεργοποίηση του αντικειμένου. Χρησιμοποιείται για αρχικοποιήσεις που απαιτούν άλλα αντικείμενα να είναι έτοιμα.
4. **Update()** – Εκτελείται σε κάθε frame και χρησιμοποιείται για συνεχή ενημέρωση, όπως κινήσεις και έλεγχοι εισόδου (input).
5. **FixedUpdate()** – Εκτελείται σε σταθερά χρονικά διαστήματα και χρησιμοποιείται για φυσική (physics), διασφαλίζοντας σταθερή συμπεριφορά ανεξάρτητα από το FPS.

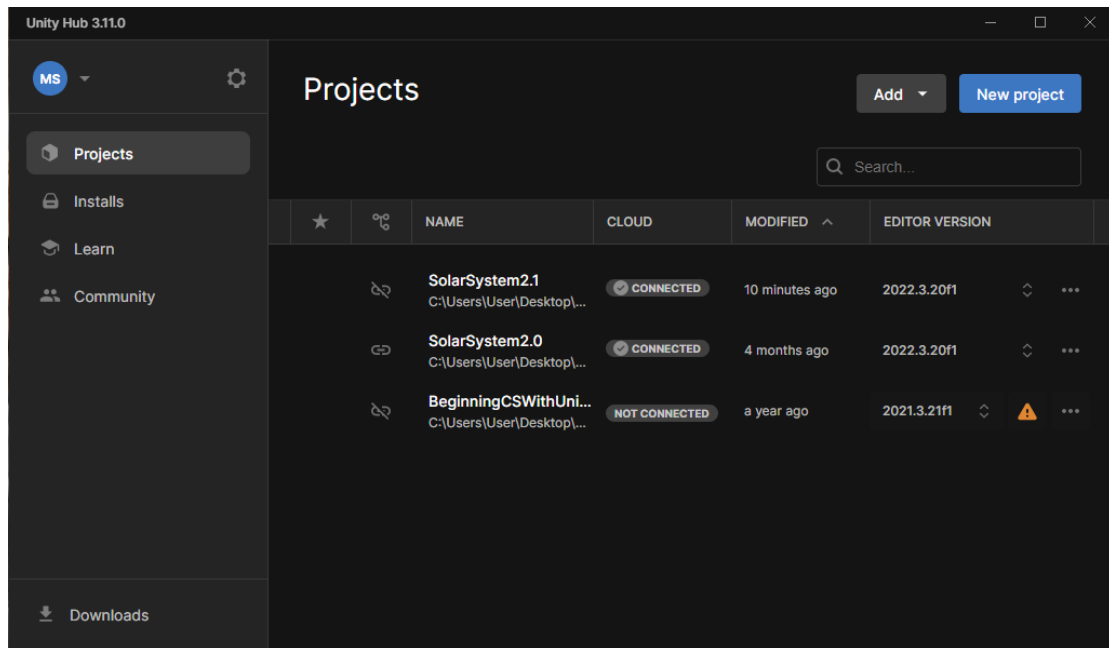
6. **LateUpdate()** – Εκτελείται μετά το Update(), χρήσιμο για λειτουργίες που πρέπει να εκτελεστούν αφού όλα τα άλλα αντικείμενα έχουν ενημερωθεί (π.χ. ακολουθία κάμερας).
7. **OnDisable()** – Εκτελείται όταν το script ή το GameObject απενεργοποιηθεί.
8. **OnDestroy()** – Εκτελείται όταν το αντικείμενο διαγραφεί από τη σκηνή. Χρησιμοποιείται για καθαρισμό πόρων (cleanup).





Εικόνα 19: Script Life Cycle Flowchart

2.2.3 Unity Hub



Εικόνα 20: Unity Hub window

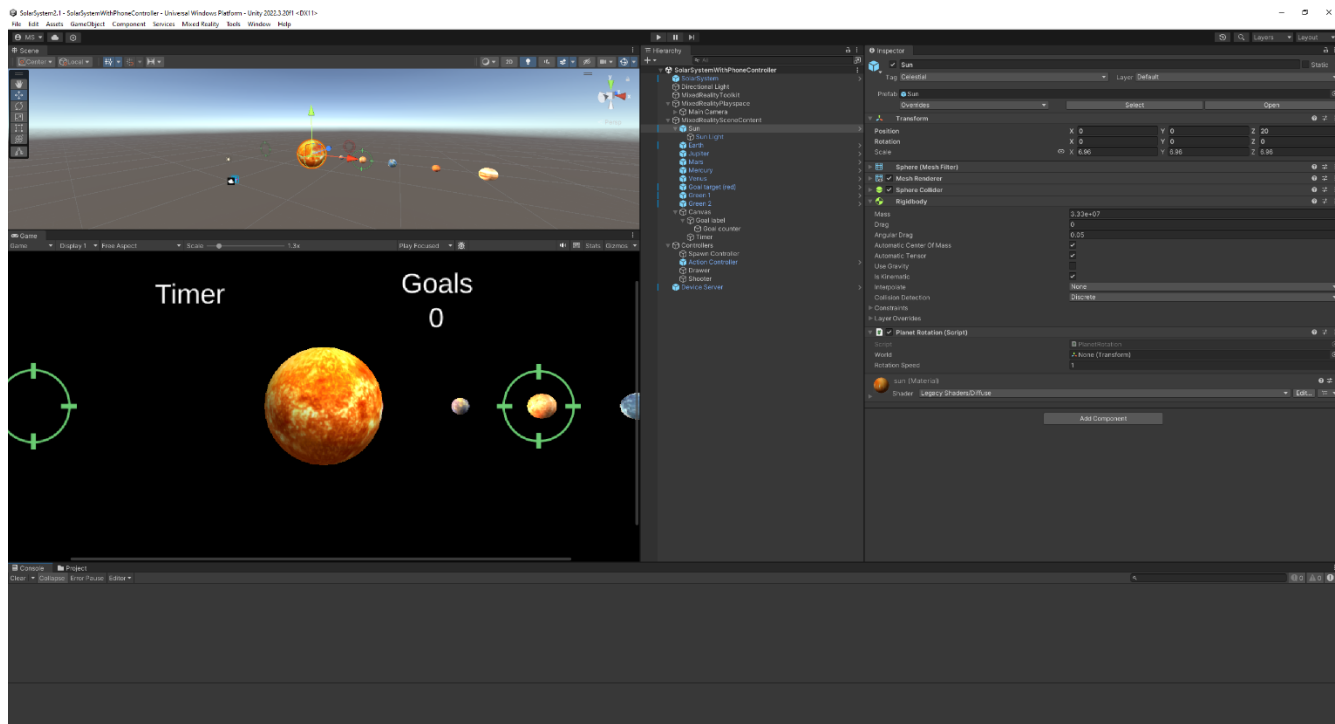
Το **Unity Hub** είναι μια εφαρμογή διαχείρισης της Unity, που επιτρέπει στους χρήστες να εγκαθιστούν, να οργανώνουν και να διαχειρίζονται τις διάφορες εκδόσεις της Unity και τα project τους. Μερικά βασικά χαρακτηριστικά του Unity Hub είναι τα εξής [28]:

- **Διαχείριση εκδόσεων της Unity** – Επιτρέπει την εγκατάσταση και διαχείριση πολλαπλών εκδόσεων της Unity, ώστε να μπορεί ο χρήστης να χρησιμοποιεί διαφορετικές εκδόσεις για διαφορετικά project.
- **Δημιουργία και άνοιγμα project** – Ο χρήστης μπορεί να δημιουργήσει νέα Unity projects ή να ανοίξει υπάρχοντα με τις κατάλληλες ρυθμίσεις.
- **Προσαρμογή ρυθμίσεων των project** – Παρέχει επιλογές για τον καθορισμό της πλατφόρμας-στόχου (PC, κινητά, VR/AR κ.λπ.).
- **Διαχείριση λογαριασμού και αδειών χρήσης** – Επιτρέπει στους χρήστες να συνδεθούν στον Unity λογαριασμό τους, να διαχειριστούν άδειες (Personal, Plus, Pro) και να αποκτήσουν πρόσβαση σε online υπηρεσίες της Unity.
- **Πρόσβαση σε μαθήματα και πόρους** – Προσφέρει γρήγορη πρόσβαση σε tutorials, documentation και το Unity Learn.
- **Community** – Παρέχει πρόσβαση σε πόρους και υποστήριξη από την κοινότητα της Unity.

Στην παρούσα διπλωματική εργασία, χρησιμοποιήθηκε η Unity 2022.3.20f1 λόγω συμβατότητας με το εργαλείο Μεικτής Πραγματικότητας MRTK 2.8.

2.2.4 Περιβάλλον Ανάπτυξης της Unity (Unity Editor)

Στην παρακάτω εικόνα φαίνεται ο editor της unity.



Εικόνα 21: Unity Editor

Τα παράθυρα μπορεί να τα τοποθετήσει ο προγραμματιστής όπως επιθυμεί. Στην συνέχεια, αναλύονται τα βασικά παράθυρα του editor:

Το **Project Window** στην Unity είναι το παράθυρο όπου βρίσκονται και οργανώνονται όλα τα αρχεία που χρησιμοποιούνται στο project. Κάθε εφαρμογή στην Unity έχει έναν φάκελο, τα **Assets**, ο οποίος βρίσκεται στο Project Window και περιέχει αυτά τα αρχεία.

Το **Console Window** εμφανίζει μηνύματα, προειδοποιήσεις και σφάλματα (logs, warnings, errors) που προκύπτουν κατά την εκτέλεση του παιχνιδιού, βοηθώντας στη διόρθωση προβλημάτων (**debugging**).

Το **Hierarchy Window** στην Unity εμφανίζει όλα τα **GameObjects** που υπάρχουν στη σκηνή σε μια **ιεραρχική δομή (tree structure)**, επιτρέποντας τη διαχείρισή τους μέσω σχέσεων **γονέα-παιδιού (parent-child)**. Αυτή η οργάνωση διευκολύνει τη δημιουργία πολύπλοκων αντικειμένων, όπως χαρακτήρες με μέρη σώματος που κινούνται μαζί ή UI στοιχεία που μετακινούνται ως ομάδα. Από το Hierarchy Window, οι χρήστες μπορούν να προσθέσουν, διαγράψουν, μετακινήσουν και οργανώσουν αντικείμενα.

Το **Inspector Window** εμφανίζει τις **ιδιότητες** του επιλεγμένου GameObject. Από εδώ, ο χρήστης μπορεί να αλλάξει **components, scripts, μεταβλητές και ρυθμίσεις** που σχετίζονται με το αντικείμενο.

Το **Scene View** είναι το παράθυρο όπου οι χρήστες μπορούν να βλέπουν και να επεξεργάζονται τα αντικείμενα μέσα στη σκηνή τους. Παρέχει εργαλεία πλοήγησης και επιλογής αντικειμένων, επιτρέποντας την τοποθέτηση, περιστροφή και κλιμάκωση τους.

Το **Game View** δείχνει πώς θα φαίνεται το παιχνίδι όταν εκτελείται. Είναι το παράθυρο όπου οι προγραμματιστές μπορούν να δοκιμάσουν και να προσομοιώσουν την εμπειρία του χρήστη, ακολουθώντας τις ρυθμίσεις της κάμερας.

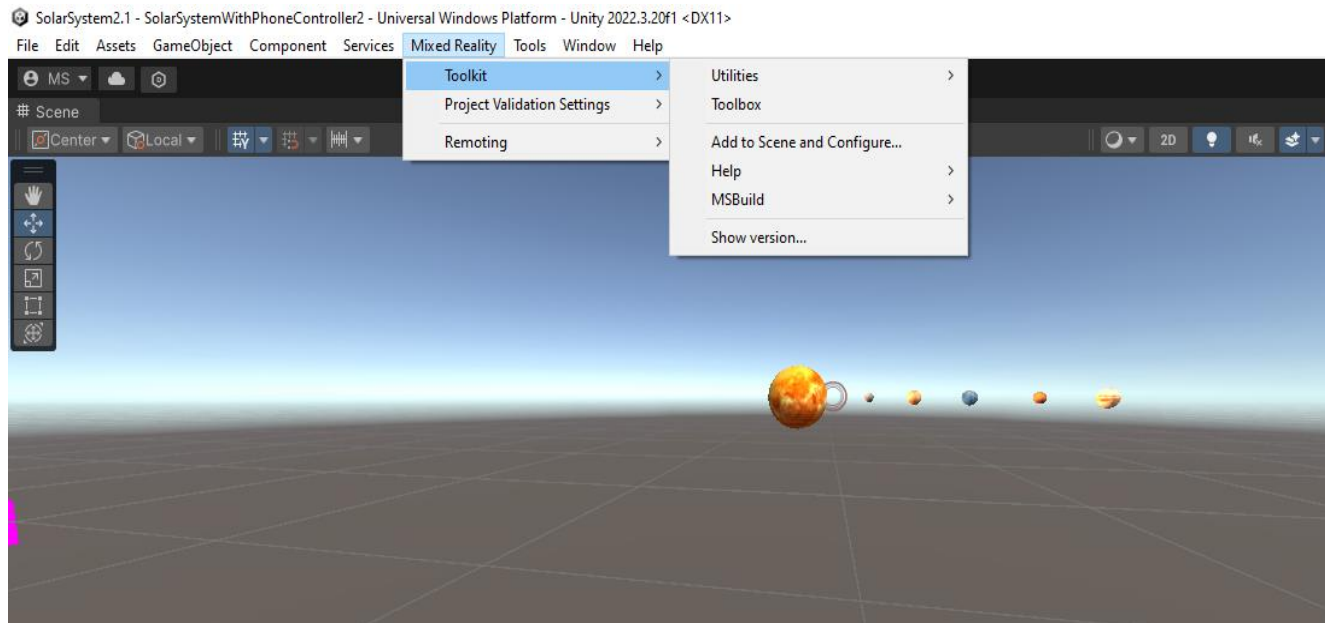
2.2.5 Mixed Reality Toolkit (MRTK)

Το **Mixed Reality Toolkit (MRTK)** είναι ένα ανοιχτού κώδικα **Software Development Kit (SDK)** που αναπτύχθηκε από τη Microsoft το 2016 για τη δημιουργία εφαρμογών **Mixed Reality (MR)** και **Augmented Reality (AR)**. Αρχικά σχεδιάστηκε για το HoloLens 1, όμως σήμερα υποστηρίζει μια ποικιλία από πλατφόρμες, επιτρέποντας στους προγραμματιστές να δημιουργούν εφαρμογές MR σε διάφορες συσκευές. Οι συσκευές που υποστηρίζονται είναι οι εξής:

XR SDK Plugin (Unity XR Plugin Management Plugin Providers)	Supported Devices
Unity OpenXR Plugin (Unity 2020 or 2021 LTS) (Mixed Reality OpenXR Plugin required for certain features on certain devices)	Microsoft HoloLens 2 Windows Mixed Reality headsets Meta Quest Device running on SteamVR via OpenXR
Windows XR Plugin	Microsoft HoloLens Microsoft HoloLens 2 Windows Mixed Reality headsets
Oculus XR Plugin (Unity 2019 or newer LTS)	Meta Quest (via Oculus Integration Package)
ARCore XR Plug-in	Android (via AR Foundation)
ARKit XR Plug-in	iOS (via AR Foundation)

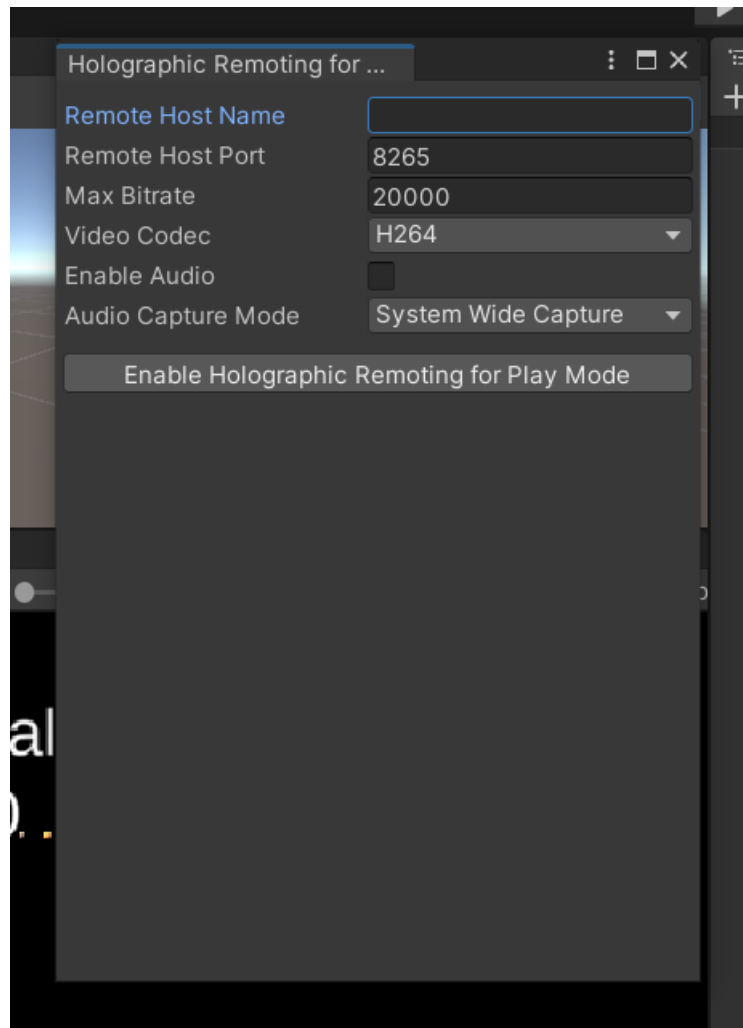
Εικόνα 22: MRTK supported devices

Μπορεί να εγκατασταθεί εύκολα μέσω του **Mixed Reality Feature Tool** για το οποίο θα μιλήσουμε παρακάτω. Η διαφορά έγκειται στο γεγονός ότι δεν γίνεται εγκατάσταση μέσω του **Package Manager** της Unity, όπως άλλα πακέτα. Παρ' όλ' αυτά, μόλις εγκατασταθεί, εμφανίζεται στον package manager και προστίθεται ένα ακόμα unity tab στο πάνω μέρος του παραθύρου [29].



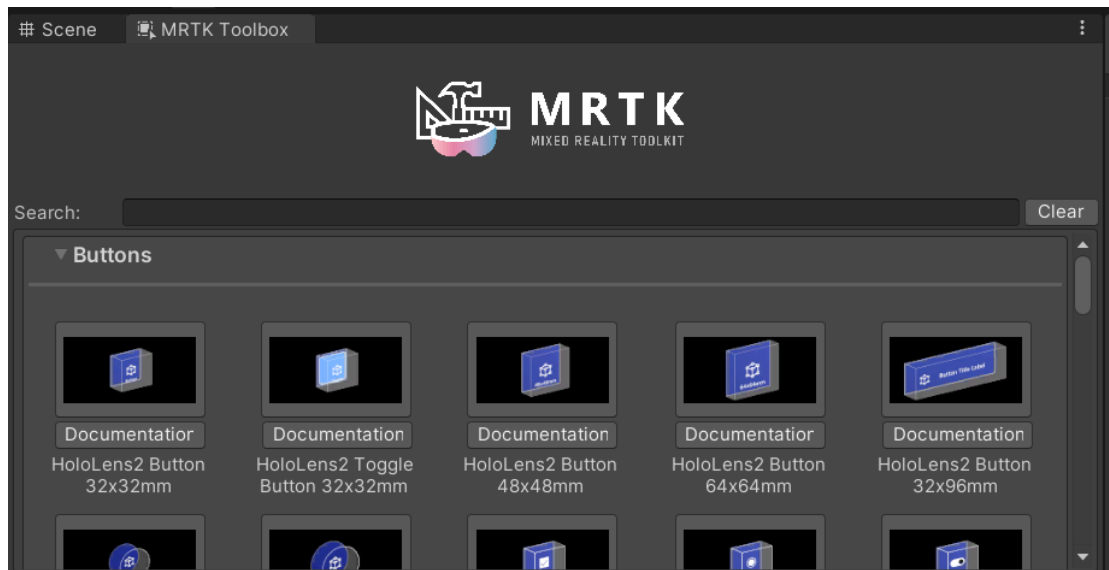
Εικόνα 23: MRTK Unity tab

Το πακέτο MRTK διευκολύνει στο setup της σκηνής του project για χρήση Μεικτής Πραγματικότητας και υλοποίηση εφαρμογών που προορίζονται για συσκευές MR, όπως τα HoloLens 2. Επιπλέον, ένα εξαιρετικά χρήσιμο εργαλείο που προσφέρει το MRTK είναι το **Holographic Remoting for Play Mode**. Πληκτρολογώντας μόνο την IP διεύθυνση της συσκευής holoLens στο παράθυρο του remoting, γίνεται δυνατό να τρέξει η εφαρμογή κατευθείαν στα γυαλιά μεικτής πραγματικότητας, χωρίς να απαιτείται να γίνει build πάνω στη συσκευή. Αυτό εξοικονομεί πολύ χρόνο και διευκολύνει εξαιρετικά το έργο του προγραμματιστή στο debugging.



Εικόνα 24: Holographic Remoting

Ένα από τα βασικά χαρακτηριστικά του MRTK είναι το **cross-platform input system**, το οποίο παρέχει τα απαραίτητα εργαλεία για αλληλεπιδράσεις στον τρισδιάστατο χώρο, συμπεριλαμβανομένων χειρονομιών, ανίχνευσης χειρών και φωνητικών εντολών. Επιπλέον, διαθέτει έτοιμα δομικά στοιχεία, όπως sliders, και εικονικά κουμπιά, για τη δημιουργία χωρικών αλληλεπιδράσεων και UI, διευκολύνοντας έτσι την ανάπτυξη εφαρμογών MR.

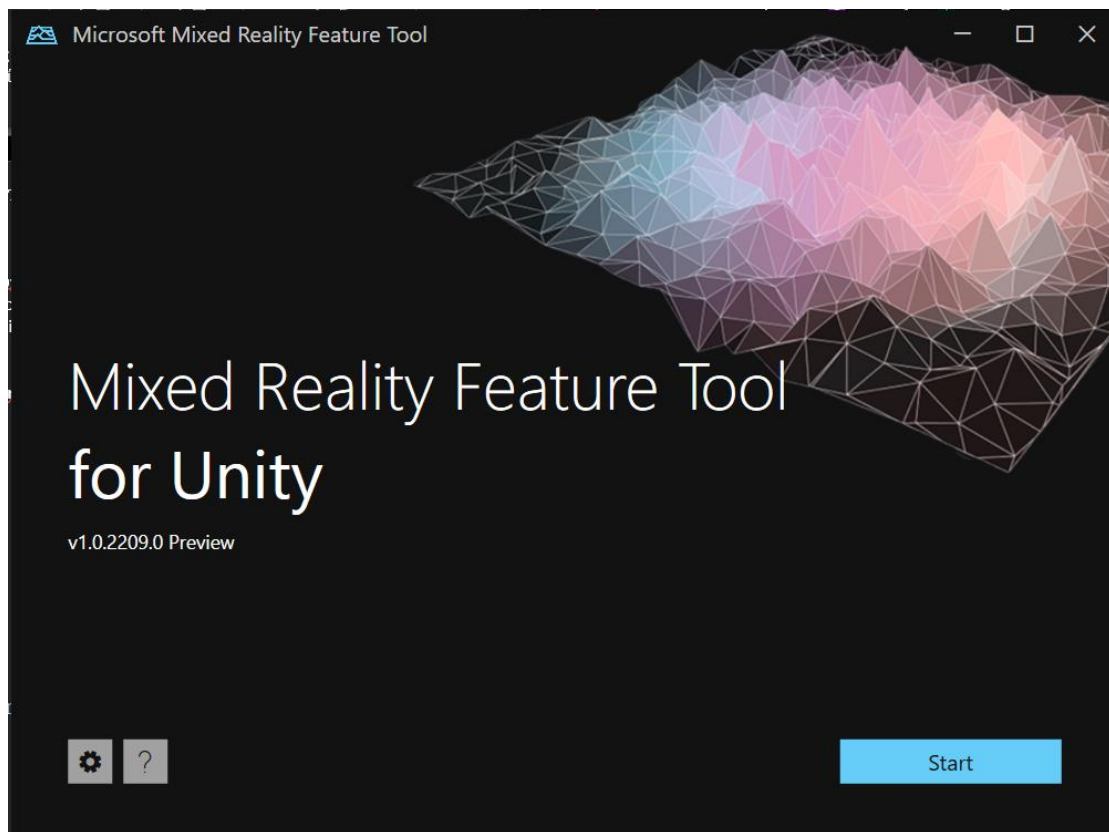


Εικόνα 25: MRTK Toolbox

Το MRTK υποστηρίζει επίσης **γρήγορη πρωτοτυποποίηση (rapid prototyping)** μέσω της **in-editor προσομοίωσης**, που επιτρέπει στους προγραμματιστές να βλέπουν άμεσα τις αλλαγές τους χωρίς να χρειάζεται να κάνουν build την εφαρμογή σε συσκευή. Αυτό μειώνει σημαντικά τον χρόνο ανάπτυξης και δοκιμών.

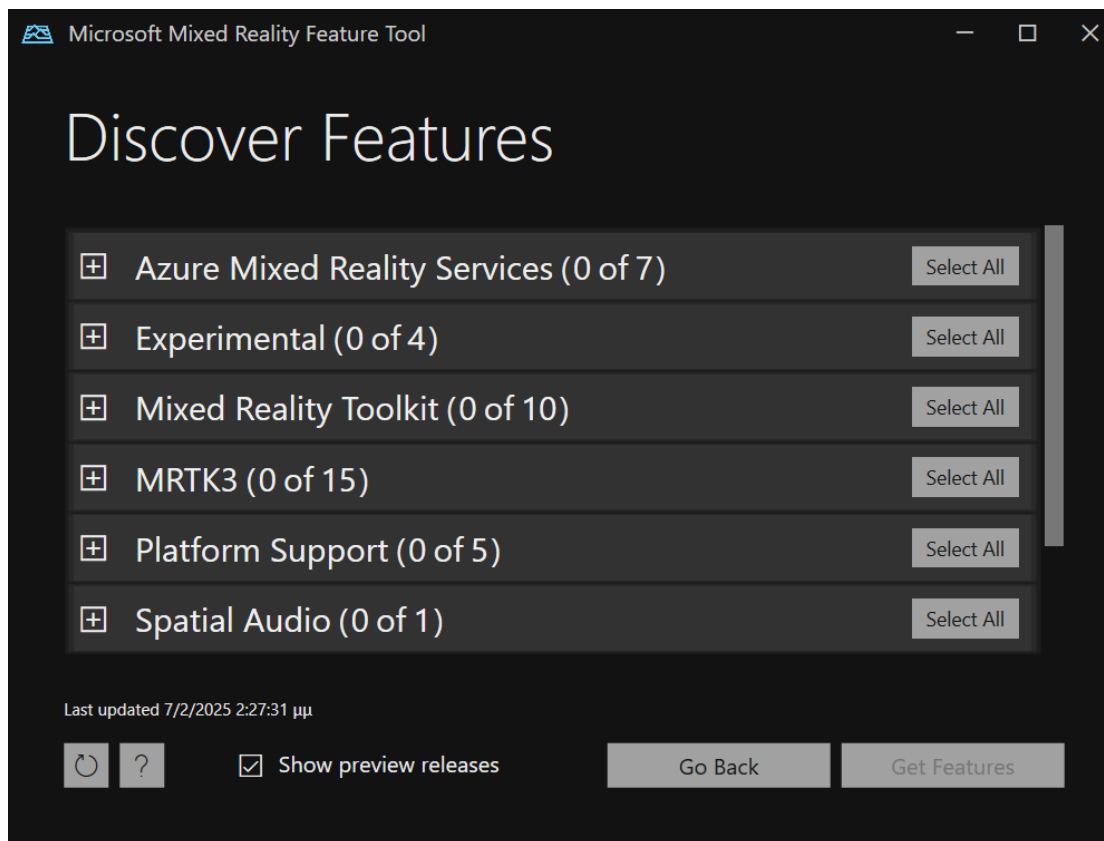
Τέλος, το MRTK είναι ένα **επεκτάσιμο framework**, προσφέροντας τη δυνατότητα στους προγραμματιστές να αντικαθιστούν βασικά του στοιχεία με δικές τους υλοποιήσεις [30] [31].

2.2.5.1 Mixed Reality Feature Tool



Εικόνα 26: Mixed Reality Feature Tool

Το **Mixed Reality Feature Tool** παρέχει στους προγραμματιστές έναν εύκολο τρόπο για να ανακαλύπτουν, να ενημερώνουν και να προσθέτουν πακέτα Mixed Reality στα Unity projects τους. Μέσω του εργαλείου, οι χρήστες μπορούν να αναζητούν πακέτα βάσει ονόματος ή κατηγορίας, να βλέπουν τις εξαρτήσεις τους και να ελέγχουν τις αλλαγές που θα γίνουν στο αρχείο manifest του έργου τους πριν από την εισαγωγή. Το αρχείο manifest είναι ένα JSON αρχείο που περιέχει όλα τα πακέτα που χρησιμοποιεί το project.



Εικόνα 27: Mixed Reality Feature Tool Packages

Αφού επιλέξουν και επικυρώσουν τα επιθυμητά πακέτα, το **Mixed Reality Feature Tool** τα κατεβάζει και τα ενσωματώνει αυτόματα στο έργο τους [32].

Τα πακέτα που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία αναφέρονται παρακάτω (πρόκειται για τις νεότερες εκδόσεις):

- **Mixed Reality Toolkit**
 - Mixed Reality Toolkit Examples
 - Mixed Reality Toolkit Extensions
 - Mixed Reality Toolkit Foundation
 - Mixed Reality Toolkit Standard Assets
 - Mixed Reality Toolkit Tools
- **Platform Support**
 - Mixed Reality OpenXR Plugin

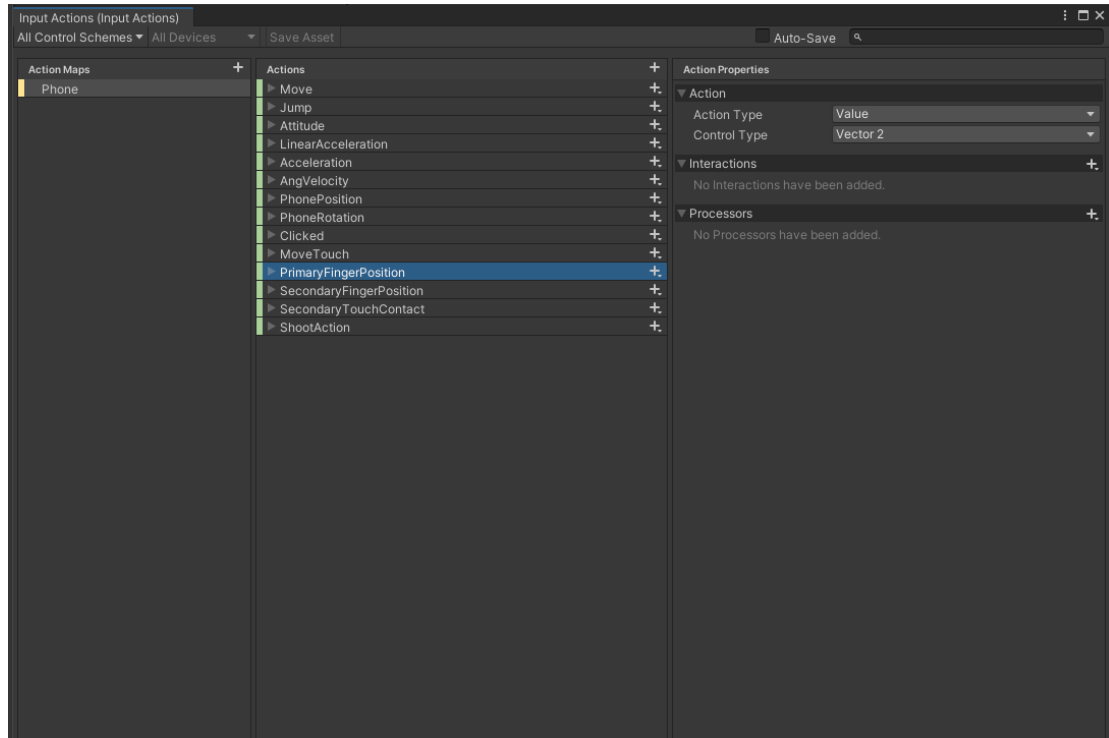
2.2.6 Σύστημα Εισόδου της Unity (Unity's Input System)

Το Input System της Unity αποτελεί ένα σύγχρονο και ευέλικτο σύστημα διαχείρισης εισόδων, το οποίο σχεδιάστηκε για να αντικαταστήσει το παλαιότερο Input Manager. Είναι ένα πακέτο που παρέχεται από την Unity μέσω του Package Manager και επιτρέπει την χρήση μιας **Συσκευής Εισόδου (Input Device)** για τον έλεγχο οποιασδήποτε λειτουργίας μέσα στο project [33].

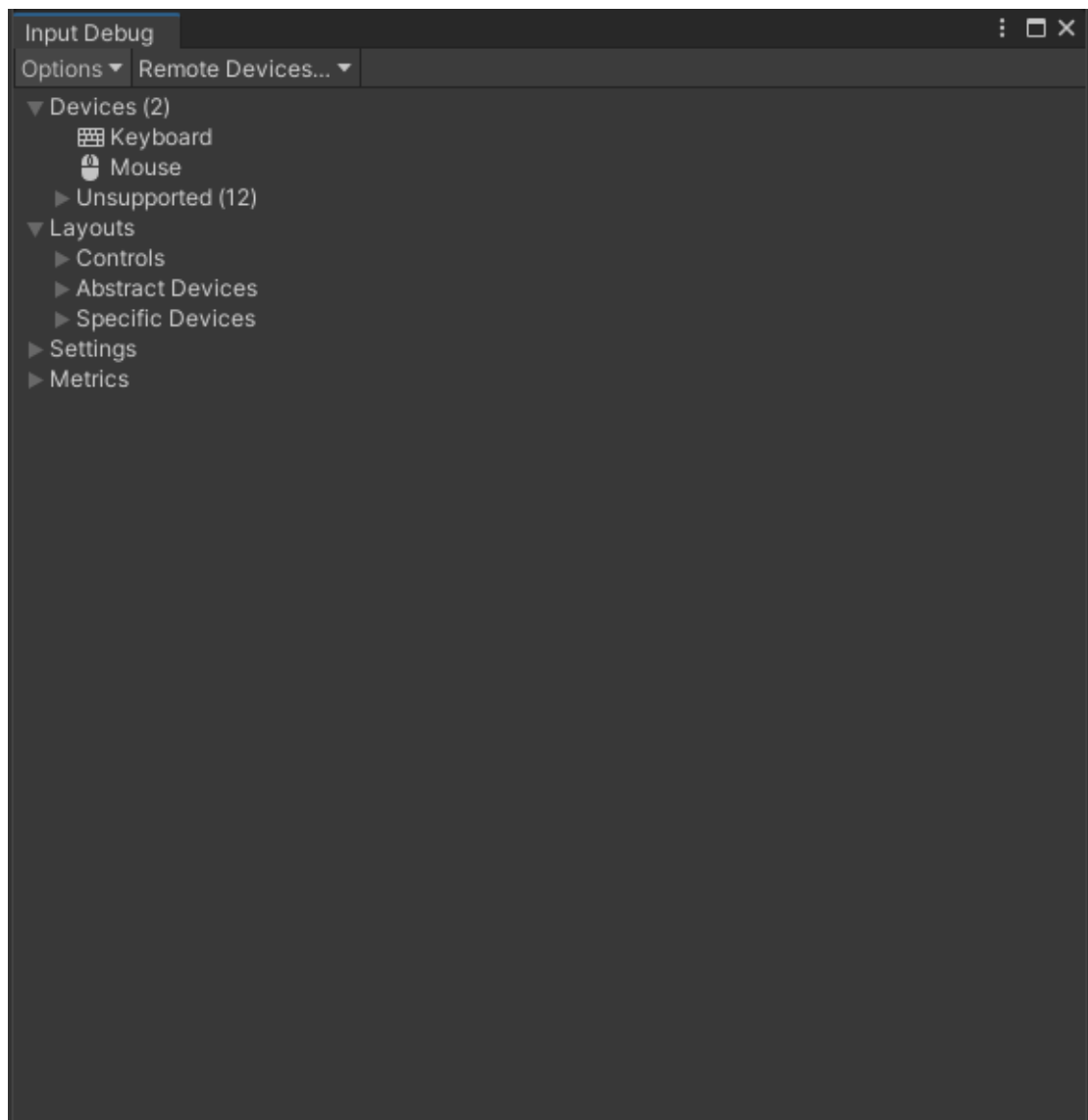
Το Input System βασίζεται σε ένα **event-driven μοντέλο**, γεγονός που επιτρέπει τη διαχείριση των εισόδων μέσω **callbacks** αντί για συνεχή έλεγχο της κατάστασης των πλήκτρων ή κουμπιών. Αυτό προσφέρει σημαντικά πλεονεκτήματα στην αποδοτικότητα του κώδικα και στη διαχείριση πολύπλοκων αλληλεπιδράσεων.

Επιπλέον, το σύστημα εισάγει τη χρήση των **Action Maps**, τα οποία επιτρέπουν την αφηρημένη διαχείριση των εισόδων μέσω του ορισμού "**ενεργειών**" (**actions**), όπως jump, move, shoot. Οι ενέργειες αυτές μπορούν να αντιστοιχιστούν σε διαφορετικές συσκευές χωρίς την ανάγκη τροποποίησης του βασικού κώδικα [34].

Ο **Input Debugger** της Unity είναι ένα εργαλείο για την ανάλυση και τον έλεγχο των συσκευών εισόδου σε πραγματικό χρόνο. Παρέχει πληροφορίες για όλες τις συνδεδεμένες συσκευές, εμφανίζει τις τρέχουσες τιμές εισόδου και επιτρέπει την καταγραφή δεδομένων για debugging. Μέσω αυτού, οι προγραμματιστές μπορούν να εντοπίσουν προβλήματα και να βελτιώσουν την απόκριση των εισόδων στις εφαρμογές τους.



Εικόνα 28: Action Map editor

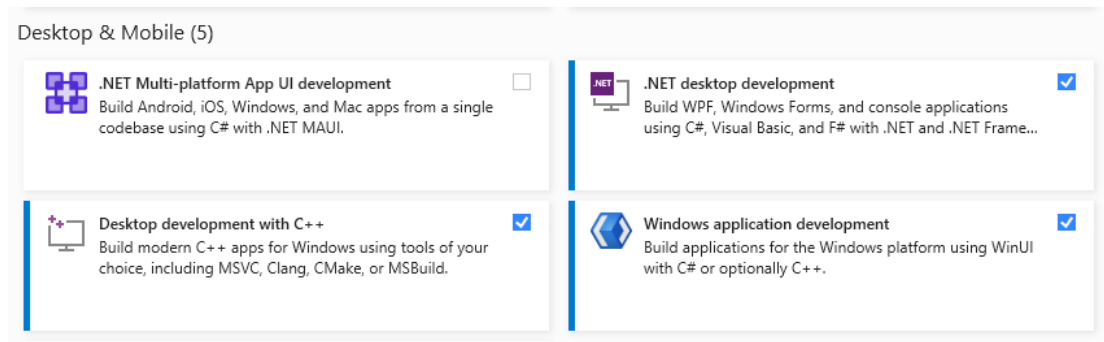


Εικόνα 29: Input Debugger

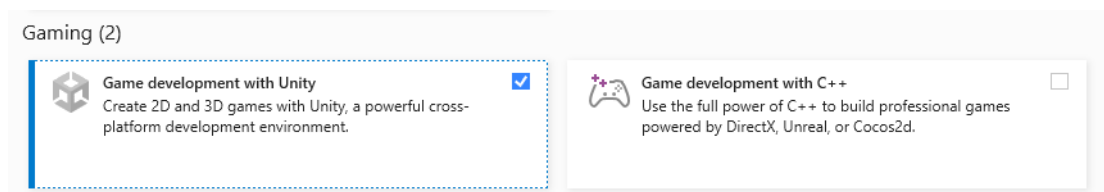
2.2.7 Microsoft Visual Studio Community

Το **Visual Studio Community** είναι ένα ισχυρό και δωρεάν IDE της Microsoft, ιδανικό για ανάπτυξη εφαρμογών και παιχνιδιών. Σε συνδυασμό με την Unity, αποτελεί μία από τις πιο δημοφιλείς επιλογές για προγραμματιστές παιχνιδιών, ειδικά για όσους χρησιμοποιούν C#. Στο visual studio υλοποιούνται όλα τα C# scripts που χρησιμοποιούνται ως components στα game objects. Για την παρούσα διπλωματική εργασία, χρησιμοποιήθηκε το Visual Studio 2022. Μαζί με την εγκατάσταση του editor, χρειάζονται και κάποιες επιπλέον ρυθμίσεις, modifications όπως ονομάζονται, για να είναι εφικτή η ενασχόληση με project μεικτής πραγματικότητας στην Unity. Μπορεί κανείς να βρει πολύ εύκολα τι χρειάζεται για κάθε

project από το διαδίκτυο. Συνοπτικά, τα modifications που εγκαταστάθηκαν για την παρούσα υλοποίηση είναι τα εξής:



Εικόνα 30: Visual Studio modifications 1



Εικόνα 31: Visual Studio modifications 2

Installation details

▸ Visual Studio core editor

▼ .NET desktop development

▼ Included

- ✓ .NET desktop development tools
- ✓ .NET Framework 4.7.2 development tools
- ✓ C# and Visual Basic

▼ Optional

- ☒ Development tools for .NET
- ☒ .NET Framework 4.8 development tools
- ☒ Entity Framework 6 tools
- ☒ .NET profiling tools
- ☒ IntelliCode
- ☒ Just-In-Time debugger
- ☒ Live Share
- ☒ ML.NET Model Builder
- ☒ GitHub Copilot
- ☒ Blend for Visual Studio
- ☐ F# desktop language support
- ☐ PreEmptive Protection - Dotfuscator
- ☐ .NET Framework 4.6.2-4.7.1 development t...
- ☐ .NET Portable Library targeting pack
- ☐ Windows Communication Foundation
- ☐ SQL Server Express 2019 LocalDB
- ☒ MSIX Packaging Tools
- ☐ JavaScript diagnostics
- ☒ Windows App SDK C# Templates
- ☐ .NET Framework 4.8.1 development tools

▼ Desktop development with C++

▼ Included

- ✓ C++ core desktop features

▼ Optional

- ☒ MSVC v143 - VS 2022 C++ x64/x86 build t...
- ☒ C++ ATL for latest v143 build tools (x86 &...
- ☐ C++ Build Insights
- ☒ Just-In-Time debugger
- ☒ C++ profiling tools
- ☒ C++ CMake tools for Windows
- ☒ Test Adapter for Boost.Test
- ☒ Test Adapter for Google Test
- ☒ Live Share
- ☒ IntelliCode
- ☒ C++ AddressSanitizer
- ☒ Windows 11 SDK (10.0.22621.0)
- ☐ vcpkg package manager
- ☒ GitHub Copilot
- ☐ C++ MFC for latest v143 build tools (x86...
- ☐ C++ Modules for v143 build tools (x64/x8...
- ☐ Incredibuild - Build Acceleration
- ☐ C++/CLI support for v143 build tools (Late...
- ☐ C++ Clang tools for Windows (18.1.8 - x64...
- ☐ JavaScript diagnostics
- ☐ Windows 11 SDK (10.0.26100.0)

Εικόνα 32: Visual Studio installation details 1

Installation details

- ☐ Incredibuild - Build Acceleration
- ☐ C++/CLI support for v143 build tools (Late...
- ☐ C++ Clang tools for Windows (18.1.8 - x64...
- ☐ JavaScript diagnostics
- ☐ Windows 11 SDK (10.0.26100.0)
- ☒ Windows 11 SDK (10.0.22000.0)
- ☒ Windows 10 SDK (10.0.20348.0)
- ☒ Windows 10 SDK (10.0.19041.0)
- ☒ Windows 10 SDK (10.0.18362.0)
- ☒ MSVC v142 - VS 2019 C++ x64/x86 build t...
- ☒ MSVC v141 - VS 2017 C++ x64/x86 build t...
- ☐ MSVC v140 - VS 2015 C++ build tools (v1...
- ☐ Windows App SDK C++ Templates

▼ Windows application development

▼ Included

- ✓ .NET WinUI app development tools
- ✓ .NET Native and .NET Standard
- ✓ NuGet package manager
- ✓ Blend for Visual Studio

▼ Optional

- ☒ IntelliCode
- ☒ GitHub Copilot
- ☐ C++ WinUI app development tools
- ☒ Universal Windows Platform tools
- ☒ C++ (v143) Universal Windows Platform to...
- ☒ C++ (v142) Universal Windows Platform to...
- ☒ C++ (v141) Universal Windows Platform to...
- ☒ Graphics debugger and GPU profiler for Di...
- ☐ Windows 11 SDK (10.0.26100.0)
- ☒ Windows 11 SDK (10.0.22621.0)
- ☒ Windows 11 SDK (10.0.22000.0)
- ☒ Windows 10 SDK (10.0.19041.0)
- ☒ Windows 10 SDK (10.0.18362.0)
- ☒ USB Device Connectivity

▼ Game development with Unity

▼ Included

- ✓ Visual Studio Tools for Unity
- ✓ C# and Visual Basic

▼ Optional

- ☒ Unity Hub
- ☒ IntelliCode
- ☒ HLSL Tools
- ☒ GitHub Copilot

▼ Individual components

- ☒ Live Share

▼ Extensions

- ☒ ML.NET Model Builder 2022
- ☒ Single-project MSIX Packaging Tools for VS 2...

Εικόνα 33: Visual Studio installation details 2

Επιπλέον, όταν το unity project προορίζεται για την συσκευή Hololens 2, πρέπει να γίνει το λεγόμενο build. Όταν γίνει το build, δημιουργείται ένα αρχείο .sln μέσα στον φάκελο του project (sln → solution αρχείο). Αφού ανοιχθεί το sln αρχείο στο visual studio γίνονται οι εξής ρυθμίσεις:

- **Solution configuration:** Release
- **Platform:** ARM64
- Επιλογή **Device** για να τρέξει απευθείας στα Hololens 2 ή **Remote Machine** για ανάπτυξη μέσω WiFi.
- **Deploy** για να εγκατασταθεί στη συσκευή

Η διαδικασία του Deployment είναι εξαιρετικά χρονοβόρα. Για τον λόγο αυτό χρησιμοποιείται η λειτουργία του Remoting, που αναφέρθηκε προηγουμένως.

3 Σχεδιασμός και Υλοποίηση

Στο παρόν κεφάλαιο, αναλύεται ο σχεδιασμός και η υλοποίηση της διπλωματικής εργασίας. Σε πρώτη φάση, αναλύεται ο σκοπός, ο λόγος για τον οποίο θα μπορούσε να είναι χρήσιμη η επικοινωνία μιας ελεγχόμενης συσκευής και ενός ελεγκτή, καθώς επίσης και μια περιγραφή του συνολικού project. Στην συνέχεια, θα αναλυθεί το project σε τέσσερα (4) κύρια τμήματα και θα γίνει λόγος για καθ' ένα από αυτά. Τα τρία από αυτά αναφέρονται στην σχεδίαση του πλαισίου επικοινωνίας μεταξύ δύο συσκευών και το τέταρτο αφορά την λειτουργία του παιχνιδιού. Με άλλα λόγια, θα γίνει αναφορά στην κωδικοποίηση των δεδομένων, στην αρχιτεκτονική της επικοινωνίας και στην υλοποίηση του πλαισίου επικοινωνίας και το τελευταίο αναφέρεται στην υλοποίηση της φυσικής και της αξιοποίησης των εισόδων από τον ελεγκτή, για την λειτουργία του παιχνιδιού.

3.1 Σκοπός

Όπως αναφέρθηκε και σε προηγούμενη ενότητα, η συσκευή Hololens 2 υποστηρίζει ποικίλους τρόπους αλληλεπίδρασης με το μεικτό περιβάλλον που δημιουργεί. Παρά το γεγονός ότι προσφέρουν μια εντυπωσιακή εμπειρία και ο χρήστης νομίζει ότι βρίσκεται σε έναν εκτεταμένο κόσμο, πέραν του πραγματικού, υπάρχουν και κάποια μειονεκτήματα.

1. Για να αναγνωριστούν οι χειρονομίες, προφανώς πρέπει τα χέρια του χρήστη να βρίσκονται μέσα στο οπτικό πεδίο της συσκευής. Αυτό προϋποθέτει, τα χέρια να είναι στο ύψος του προσώπου, δηλαδή μόνιμα σε κατάσταση κάποιας ανάτασης η οποία, μετά την πάροδο κάποιας ώρας, μπορεί να είναι κουραστική. Επιπλέον, είναι ακόμα δυσκολότερο να χρησιμοποιηθεί από άτομα με κινητικές δυσκολίες.

2. Όσον αφορά την ανίχνευση του βλέμματος, είναι μια διαδικασία κουραστική για τα μάτια του χρήστη και ίσως να είναι κάπως αργή.

Για τους λόγους αυτούς, προέκυψε η ιδέα επέκτασης της μεικτής πραγματικότητας, που προσφέρει η συσκευή HoloLens 2, με χρήση ενός κινητού. Το κινητό είναι μια συσκευή προσβάσιμη και γνώριμη σχεδόν από όλους τους ανθρώπους, η οποία μάλιστα παρέχει πολλούς αισθητήρες που μπορούν να χρησιμοποιηθούν ως είσοδοι για την δημιουργία νέων τρόπων αλληλεπίδρασης.

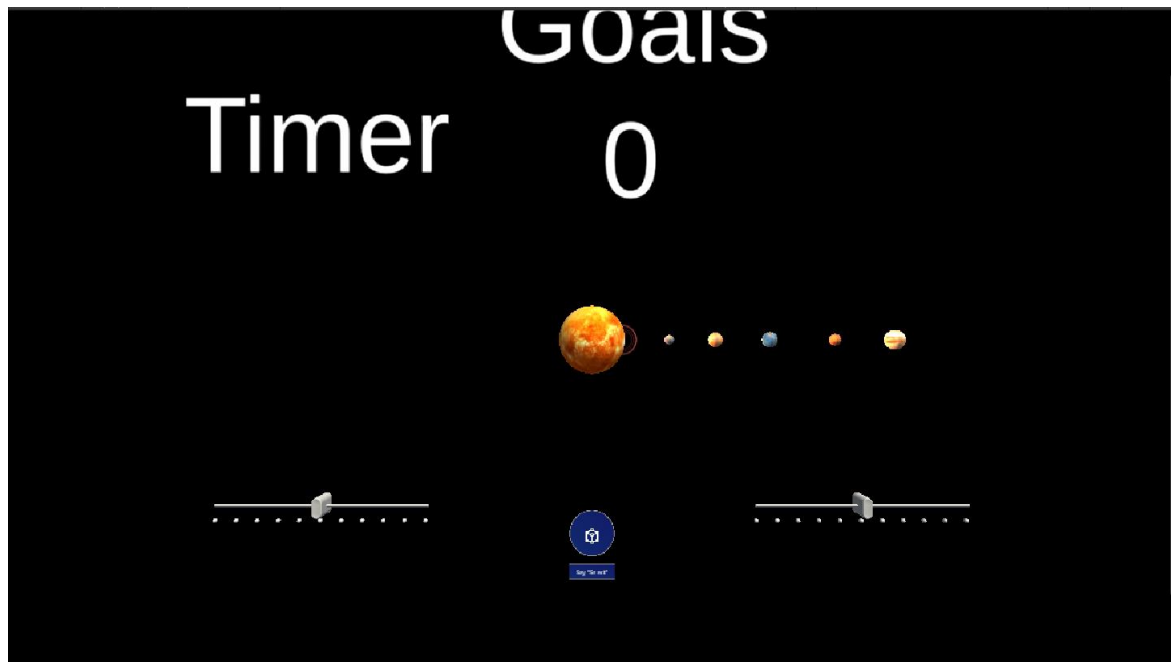
Να σημειωθεί ότι, όπως αναφέρθηκε και στον πρόλογο, η υλοποίηση της διπλωματικής είναι βασισμένη στην δουλειά του κυρίου Παπαδούλη Γεώργιου, όπου το πλαίσιο επικοινωνίας είναι βασισμένο στο **Master Server Framework**, το οποίο θα αναλυθεί στην συνέχεια[36].

3.2 Περιγραφή Παιχνιδιού

Ολόκληρο το παιχνίδι και οι λειτουργίες του, η εφαρμογή του κινητού, καθώς επίσης και το πλαίσιο επικοινωνίας μεταξύ client και server είναι γραμμένα σε C# μέσα στη Unity.

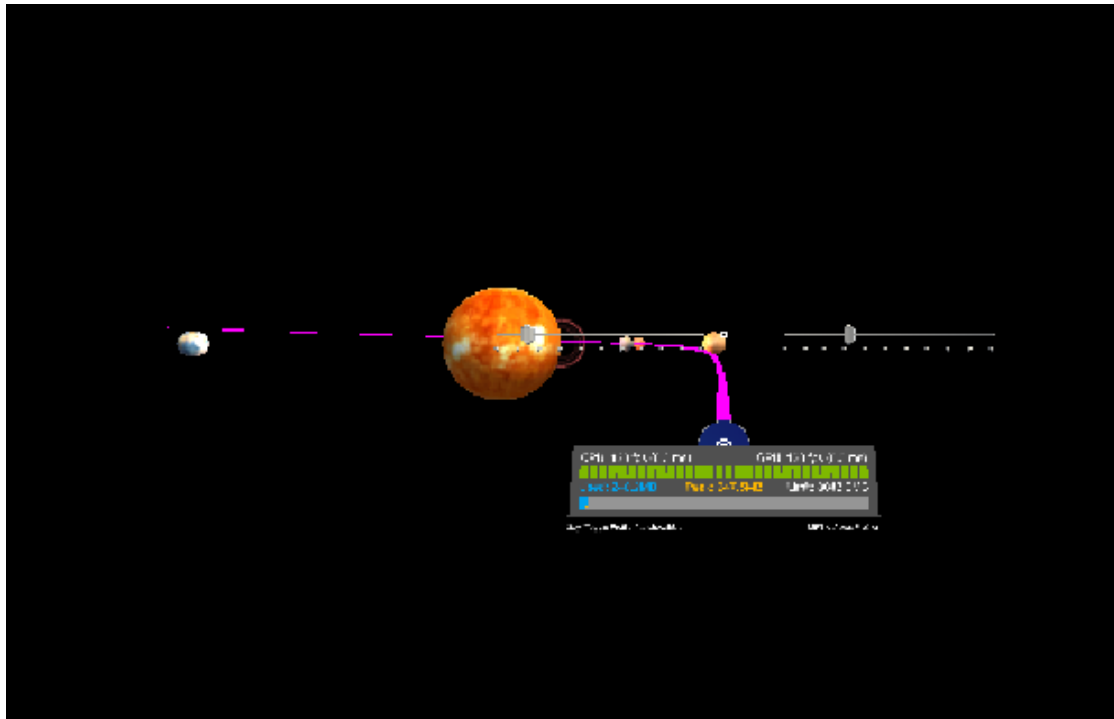
Το κεντρικό θέμα του παιχνιδιού είναι το **ηλιακό σύστημα**. Έχουν δημιουργηθεί ο **ήλιος** και **πέντε πλανήτες**, από τους 7, με scaled μάζες, μεγέθη και αποστάσεις. Όλοι οι πλανήτες εκτελούν περιστροφή γύρω από τον ήλιο, αλλά και ιδιοπεριστροφή, χρησιμοποιώντας τους νόμους της βαρύτητας. Επιπλέον, έχει τοποθετηθεί ένας **κόκκινος στόχος** σε μια συγκεκριμένη θέση πίσω από τον ήλιο. Σκοπός του παιχνιδιού είναι οι παίκτες να πετύχουν αυτόν τον στόχο, ρίχνοντας μια μπάλα-κομήτη μέσα από τα βαρυτικά πεδία των ουράνιων σωμάτων. Επίσης, υπάρχει ένας **timer** για να κρατάει χρόνο, και ένας **goal counter** για το σκορ.

Υπάρχουν δύο τρόποι χειρισμού του παιχνιδιού. Στο τέλος, εξετάζεται ποιος από τους δύο τρόπους είναι πιο αποτελεσματικός για την επίτευξη του στόχου. Η μια από τις δύο λειτουργίες του παιχνιδιού είναι με τις χειρονομίες που αναγνωρίζει το HoloLens 2. Συγκεκριμένα, έχουν τοποθετηθεί δύο sliders και ένα εικονικό κουμπί στο οπτικό πεδίο του χρήστη. Να σημειωθεί, ότι πρόκειται για **PinchSliders** και το **VisualTouchButton** από το **MRTK Toolbox**, που αναφέρθηκε προηγουμένως. Τέλος, υπάρχει και μια **ακτίνα** η οποία δείχνει την πορεία που θα ακολουθήσει η μπάλα, μόλις εκτοξευθεί. Η πορεία υπολογίζεται με βάση την μάζα του αντικειμένου, την δύναμη εκτόξευσης, την θέση του παίκτη και τα βαρυτικά πεδία των πλανητών και του ήλιου.

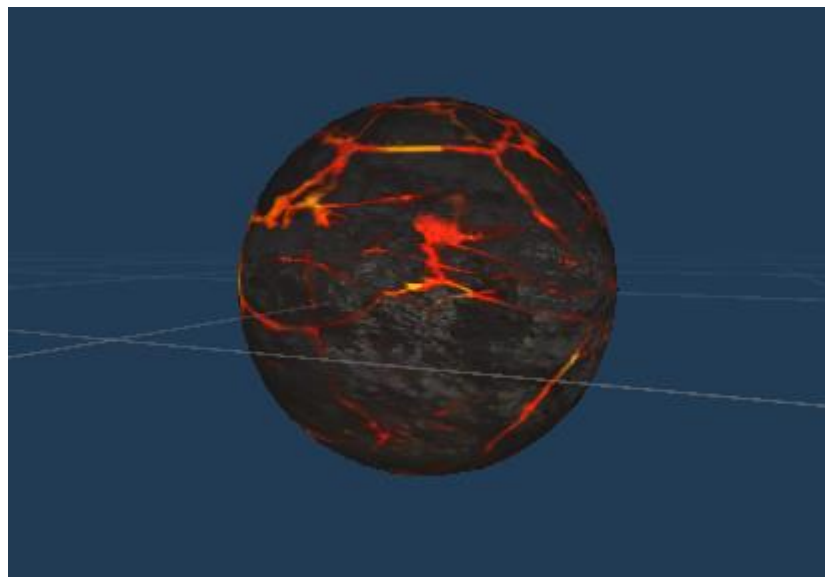


Εικόνα 34: Πρώτος τρόπος παιχνιδιού: Hololens 2

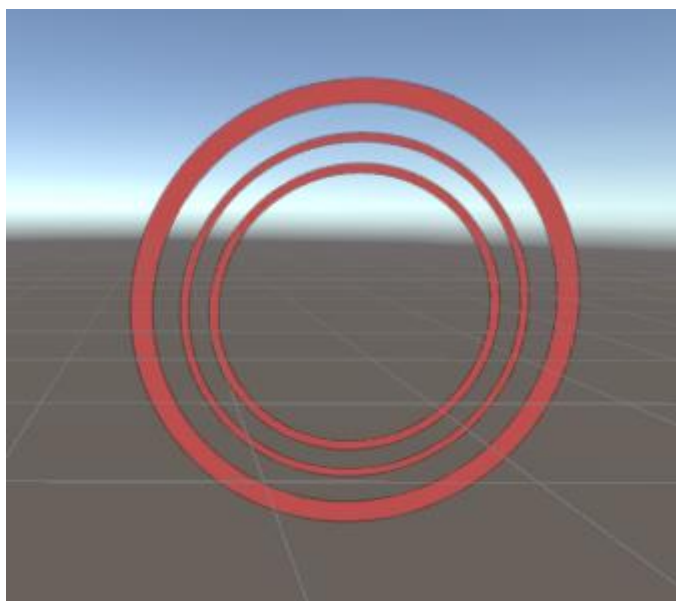
Ο αριστερός slider ρυθμίζει την μάζα του projectile και ο δεξιός slider ρυθμίζει την δύναμη εκτόξευσης. Όσο ο παίκτης «πειράζει» τους sliders, η ακτίνα ανανεώνεται σε πραγματικό χρόνο. Έτσι μπορεί να στοχεύσει, γυρνώντας το κεφάλι του κατάλληλα, για να αποφύγει την ισχυρή βαρυτική έλξη του ήλιου. Όταν θεωρήσει ότι είναι έτοιμος, πατάει το εικονικό κουμπί και σουτάρει.



Εικόνα 35: Ray for projectile's path

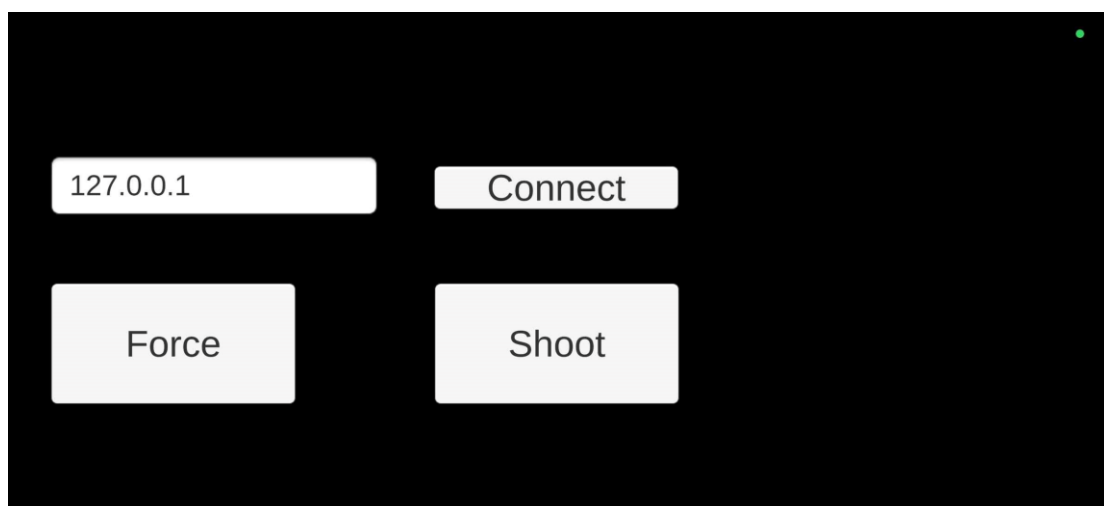


Εικόνα 36: Projectile



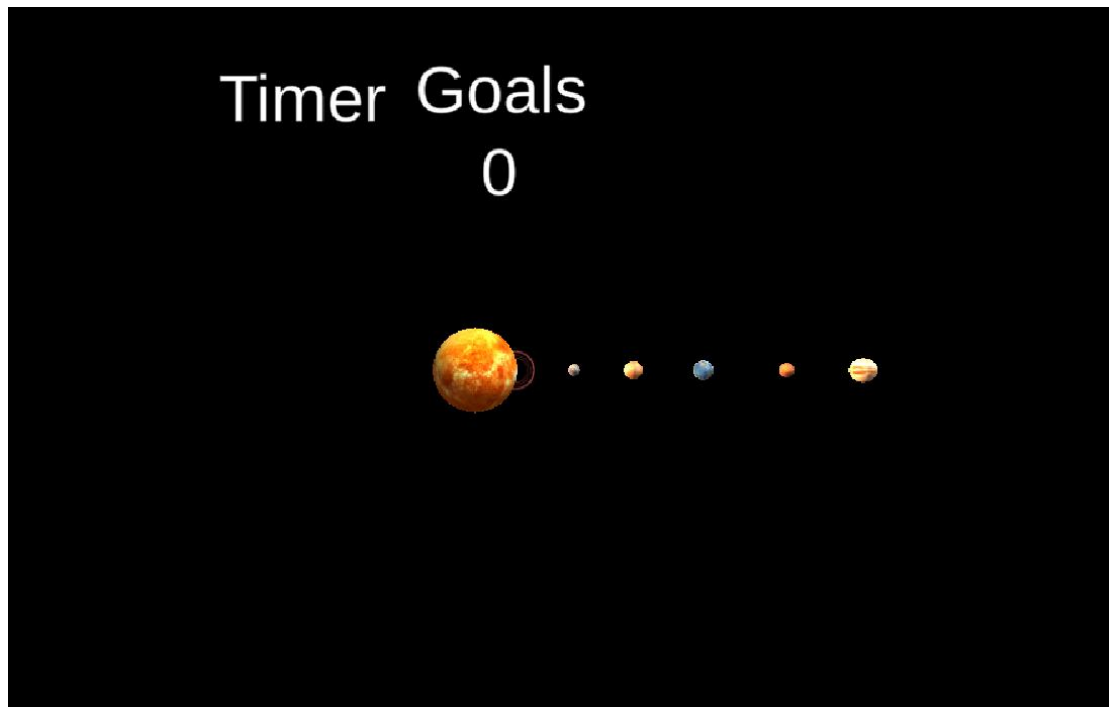
Εικόνα 37: Goal target

Ο δεύτερος τρόπος παιχνιδιού γίνεται με την χρήση του κινητού τηλεφώνου ως εξωτερικό ελεγκτή. Παρακάτω φαίνεται η εφαρμογή του κινητού.



Εικόνα 38: Εφαρμογή κινητού

Σε αυτή την λειτουργία, δεν υπάρχουν οι sliders και το κουμπί, ωστόσο ο σκοπός του παιχνιδιού είναι ο ίδιος με πριν.



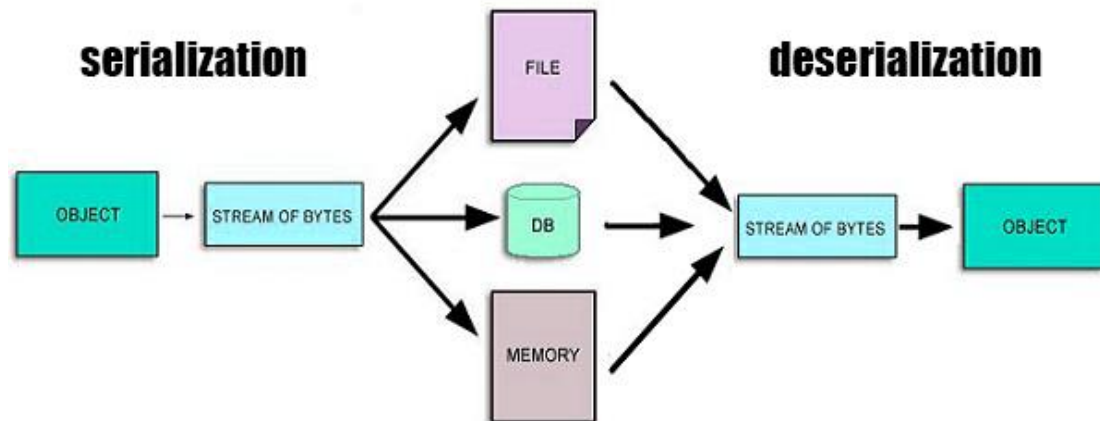
Εικόνα 39: Δεύτερος τρόπος παιχνιδιού: Κινητό τηλέφωνο

Σε αυτή την περίπτωση, το παιχνίδι παίζεται ως εξής. Μόλις ξεκινήσει η εφαρμογή, ο χρήστης πληκτρολογεί την **διεύθυνση IP** της ελεγχόμενης συσκευής στο **input text field** και πατάει **connect**. Είναι σημαντικό να επισημανθεί ότι οι δύο συσκευές, τα HoloLens και το κινητό, πρέπει να είναι συνδεδεμένα στο ίδιο δίκτυο. Έτσι πραγματοποιείται η σύνδεση μεταξύ τους. Έπειτα, ο παίκτης κάνει **tap στην μαύρη κενή οθόνη** της εφαρμογής για να εμφανίσει το αντικείμενο που θα εκτοξεύσει. Στην συνέχεια, χρησιμοποιώντας δύο δάχτυλα, κάνει zoom-in και zoom-out για να αλλάξει την μάζα και το μέγεθος της μπάλας. Μόλις ρυθμίσει την μάζα, πατάει το κουμπί **Force** για να επιλέξει την δύναμη εκτόξευσης. Όταν γίνει αυτό, ο χρήστης θα παρατηρήσει την καμπύλωση της ακτίνας να μεταβάλλεται σταδιακά. Όταν η δύναμη αυξάνεται, η ακτίνα θα απομακρύνεται από τον ήλιο, ενώ στην αντίθετη περίπτωση, θα συγκλίνει πάνω του. Έτσι μπορεί να καταλάβει και να προσαρμόσει κατάλληλα την δύναμη. Τέλος, πατάει το κουμπί Shoot για να σουτάρει. Προφανώς, μπορεί να πατήσει κατευθείαν το shoot, κρατώντας τις προηγούμενες ρυθμίσεις.

3.3 Κωδικοποίηση (Serialization)

Για να υλοποιηθεί ένα πλαίσιο επικοινωνίας μεταξύ συσκευών, πρέπει να εδραιωθεί μια διαδρομή που θα μεταφέρεται πληροφορία μεταξύ τους. Με άλλα λόγια, πρέπει να υπάρχει μια ροή από bytes η οποία θα κωδικοποιείται από τον «πελάτη», θα στέλνεται μέσω κάποιου δικτύου στον «διακομιστή» και θα αποκωδικοποιείται από αυτόν για χρήση σε κάποια εφαρμογή. Κωδικοποίηση, λοιπόν, είναι η διαδικασία κατά την οποία μια δομή

δεδομένων μεταφράζεται σε μια κατάσταση, ώστε να μπορεί να αποθηκευτεί ή να μεταφερθεί [35].



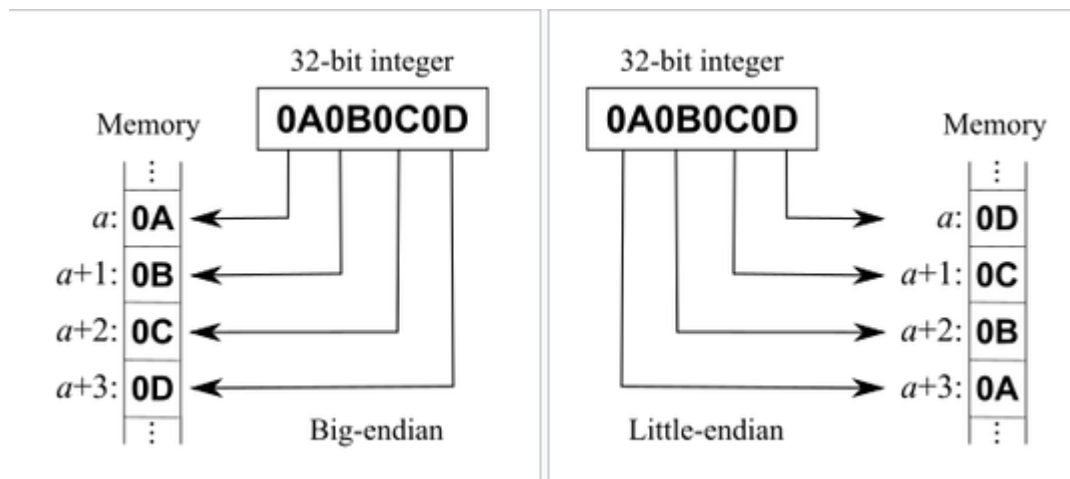
Εικόνα 40: Serialization diagram

Παρακάτω, αναλύονται οι κλάσεις που αφορούν την κωδικοποίηση μιας ροής δεδομένων.

3.3.1 Class EndianBitConverter

Είναι μια **abstract class**, δηλαδή μια αφαιρετική κλάση, η οποία υλοποιεί μεθόδους-συναρτήσεις που μετατρέπουν τους βασικούς τύπους δεδομένων σε έναν πίνακα από bytes, αλλά και το αντίστροφο.

Μια αφαιρετική κλάση υλοποιεί η ίδια τις μεθόδους της. Αντιθέτως, δημιουργούνται άλλες κλάσεις οι οποίες την «επεκτείνουν» και οι συναρτήσεις της υλοποιούνται σε αυτές. Στην προκειμένη περίπτωση, οι κλάσεις που υλοποιούν τις μεθόδους μετατροπής είναι η **LittleEndianBitConverter** και η **BigEndianBitConverter**. Κάθε μια από αυτές τις κλάσεις είναι υπεύθυνη για τον αντίστοιχο τρόπο μετατροπής των δεδομένων σε πίνακα από bytes. Πιο συγκεκριμένα, όταν μιλάμε για big endian τρόπο, η μετατροπή ξεκινάει από το **Most Significant Bit (MSB)**, ενώ στην αντίθετη περίπτωση η μετατροπή ξεκινάει από το **Least Significant Bit (LSB)**.



Εικόνα 41: Big and Little endian conversion

3.3.2 Class EndianBinaryWriter

Η κλάση αυτή επιτρέπει την εγγραφή διάφορων βασικών τύπων δεδομένων σε ένα **stream** από bytes. Είναι μια προσαρμοσμένη υλοποίηση του `BinaryWriter` και υποστηρίζει **big endian** αλλά και **little endian formats**, μέσω της κλάσης `EndianBitConverter` που αναλύθηκε προηγουμένως, η οποία είναι υπεύθυνη για την μετατροπή των δεδομένων στην σωστή σειρά από bytes πριν την εγγραφή στο stream. Να σημειωθεί ότι εδώ χρησιμοποιείται το **MemoryStream**, που παρέχεται από την βασική βιβλιοθήκη.

3.3.3 Class EndianBinaryReader

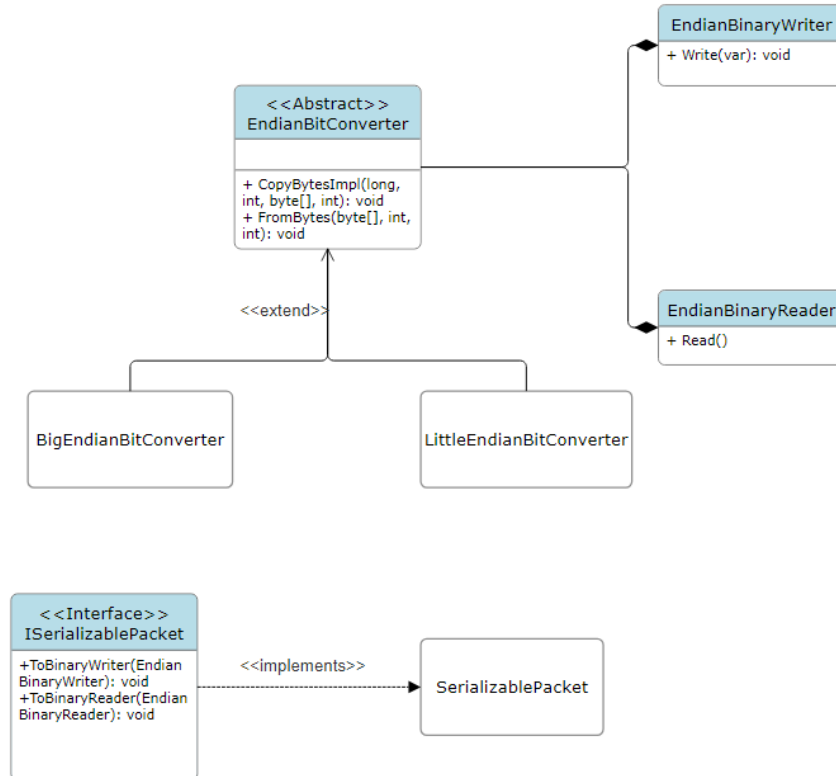
Η κλάση αυτή έχει την αντίθετη λειτουργία από την `EndianBinaryWriter`. Διαβάζει ένα **stream από bytes**, και μέσω του `EndianBitConverter` μετατρέπει αυτό το **stream** στους **βασικούς τύπους δεδομένων** και τελικά επιστρέφει το αποτέλεσμα.

3.3.4 Interface ISerializablePacket

Η διεπαφή αυτή ορίζει κάποιες μεθόδους που κωδικοποιούν και αποκωδικοποιούν πακέτα, χρησιμοποιώντας τις κλάσεις `EndianBinaryWriter` και `EndianBinaryReader`, οι οποίες την υιοθετούν. Συγκεκριμένα, οι μέθοδοι που ορίζει αυτή η διεπαφή είναι:

- **ToBinaryWriter:** Μετατρέπει ένα «αντικείμενο» σε δυαδική μορφή (κωδικοποίηση) και το γράφει σε ένα `EndianBinaryWriter`.
- **FromBinaryReader:** Διαβάζει τα δεδομένα από ένα `EndianBinaryReader` και τα αποκωδικοποιεί.

- **ToBytes:** Επιστρέφει την κλάση ως έναν πίνακα από bytes.



Εικόνα 42: UML Diagram - Serialization

3.4 Αρχιτεκτονική Επικοινωνίας

Σε αυτό το σημείο, θα αναλυθεί ο σχεδιασμός ενός γενικού συστήματος επικοινωνίας μεταξύ ενός πελάτη (client) και ενός διακομιστή (server). Η επικοινωνία είναι αμφίδρομη, δηλαδή και ο πελάτης και ο διακομιστής μπορούν να στέλνουν και να διαβάζουν μηνύματα.

3.4.1 Interface IMessage

Η διεπαφή αυτή ορίζει μια δομή, η οποία είναι υπεύθυνη να χειρίζεται εξερχόμενα μηνύματα. Κάθε μήνυμα χαρακτηρίζεται από κάποια δεδομένα που το περιγράφουν.

1. **short OpCode:** Είναι ένας αριθμός που δείχνει τον τύπο του μηνύματος και είναι μόνο για ανάγνωση (read-only).

2. **byte[] Data:** Είναι ένας πίνακας από byte και περιέχει τα δεδομένα του μηνύματος σε δυαδική μορφή.
3. **bool HasData:** Είναι true αν το μήνυμα περιέχει δεδομένα, αλλιώς false.
4. **int? AckRequestId:** Αν έχει τιμή, σημαίνει ότι το μήνυμα περιμένει απάντηση.
5. **int? AckResponseId:** Δείχνει ότι το μήνυμα αποτελεί απάντηση σε προηγούμενο μήνυμα. Πρέπει να είναι ίσο με το AckRequestId.
6. **byte Flags:** Περιέχει βοηθητικές πληροφορίες (flags) για το μήνυμα, όπως π.χ. αν είναι κρυπτογραφημένο, αν απαιτεί άμεση παράδοση κλπ.
7. **ResponseStatus Status:** Δείχνει την κατάσταση του μηνύματος (π.χ. επιτυχία, σφάλμα, χρονικό όριο κλπ.).

Στο τελευταίο, ο τύπος ResponseStatus είναι ένα **enum**, το οποίο περιγράφει την κατάσταση του μηνύματος. Οι τιμές που μπορεί να πάρει είναι:

- **Success = 0:** Επιτυχής απάντηση
- **Failed = 1:** Γενική αποτυχία
- **Unauthorized = 2:** Ο χρήστης δεν έχει άδεια (π.χ. λάθος credentials)
- **NotFound = 3:** Το ζητούμενο αντικείμενο δεν βρέθηκε
- **Timeout = 4:** Η απάντηση καθυστέρησε πάρα πολύ
- **Error = 5:** Παρουσιάστηκε σφάλμα κατά την επεξεργασία
- **Invalid = 6:** Τα δεδομένα ή η αίτηση ήταν άκυρα

3.4.2 Interface IMessageDispatcher

Η διεπαφή αυτή λειτουργεί ως **μηχανισμός αποστολής μηνυμάτων**, όπως περιγράφει και το όνομά της. Επιτρέπει την **αποστολή δεδομένων** σε έναν απομακρυσμένο **peer (IPeer)**, υποστηρίζοντας διαφορετικούς τύπους δεδομένων. Οι μέθοδοι **SendMessage** στέλνουν μηνύματα με συγκεκριμένο **OpCode**, ενώ ορισμένες επιτρέπουν **επιλογή μεθόδου παράδοσης (DeliveryMethod)**, **διαχείριση απαντήσεων (ResponseCallback)** και **χρονικό όριο (timeoutSecs)**.

3.4.3 Interface IPeer

Η διεπαφή αυτή αναπαριστά μια **σύνδεση (peer)** σε δίκτυο, διαχειρίζεται την αποστολή και λήψη μηνυμάτων μεταξύ πελάτη και διακομιστή και επιπλέον επεκτείνει την **IMessageDispatcher**. Ορίζει γεγονότα (**events**) που αφορούν την παραλαβή ενός μηνύματος και την αποσύνδεση. Τέλος, περιέχει μεθόδους για ανάκτηση και αποθήκευση δεδομένων του μηνύματος.

3.4.4 Interface IncomingMessage

Η διεπαφή αυτή, όπως δείχνει και το όνομά της, περιγράφει **ένα εισερχόμενο μήνυμα** είτε αυτό λαμβάνεται από τον πελάτη είτε από τον διακομιστή. Περιέχει όλα τα δεδομένα ενός μηνύματος, όπως και η διεπαφή IMessage, καθώς επίσης γνωρίζει και τον peer, δηλαδή τον αποστολέα. Επιπλέον, παρέχει την δυνατότητα άμεσης απάντησης στο λαμβανόμενο μήνυμα.

3.4.5 Interface IClientSocket

Η διεπαφή αυτή αφορά έναν **πελάτη (client)** σε ένα δίκτυο. Παρέχει μεθόδους για **σύνδεση και αποσύνδεση με τον διακομιστή (server)** καθώς και **events** για τις αλλαγές στην κατάσταση σύνδεσης. Συγκεκριμένα, υπάρχει μια μέθοδος **connect**, η οποία παίρνει ως ορίσματα μια **IP Address** και **ένα port**, για να γίνει η σύνδεση με τον διακομιστή. Επιπλέον, είναι υπεύθυνη για την αποστολή μηνυμάτων, εφόσον πρόκειται για πελάτη.

3.4.6 Interface IServerSocket

Σε αντίθεση με την IClientSocket, η διεπαφή αυτή αναπαριστά έναν διακομιστή (server) και διαχειρίζεται την σύνδεση με πελάτες (clients). Επιτρέπει την εκκίνηση και διακοπή «ακρόασης» σε μια θύρα (port). Επίσης, παρέχει γεγονότα (events) για την παρακολούθηση της σύνδεσης με τον πελάτη.

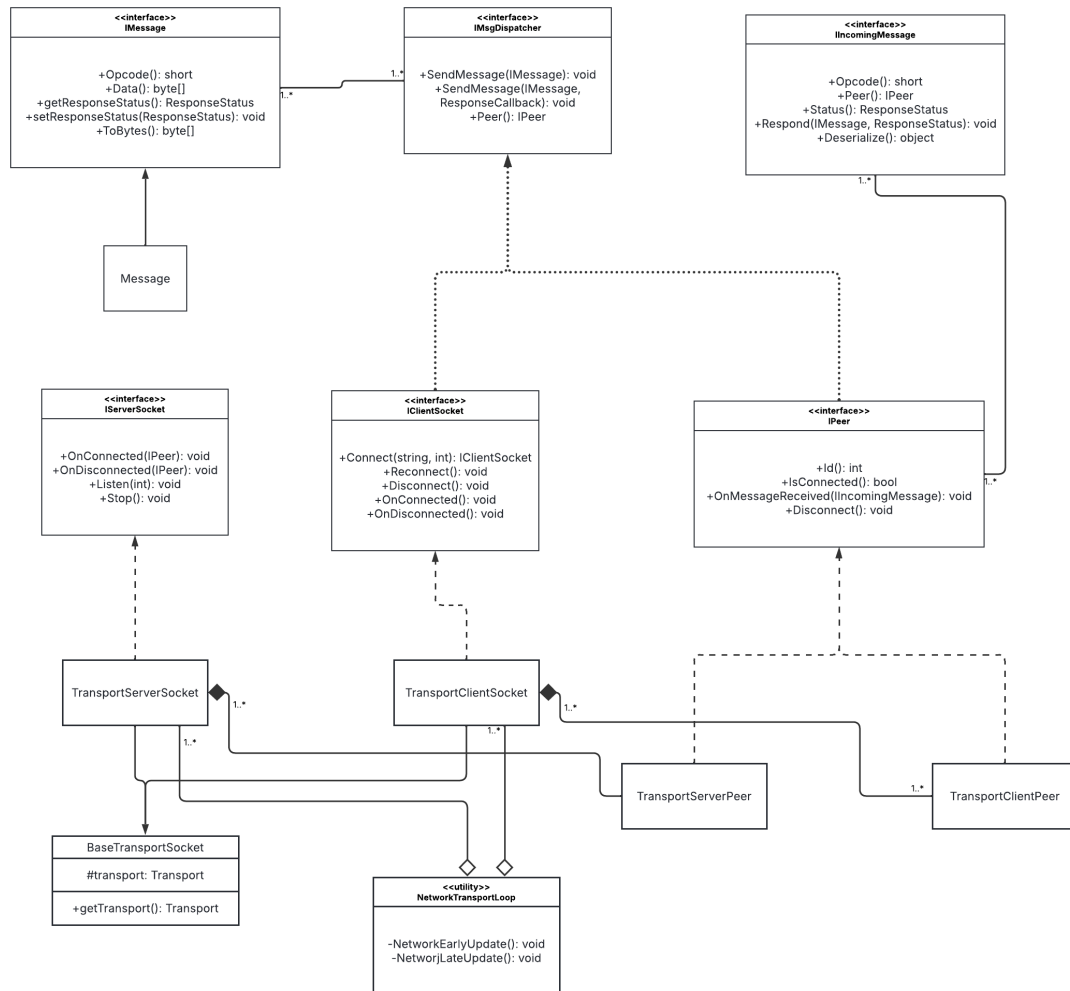
3.4.7 Mirror Networking

Το **Mirror** είναι μια υψηλού επιπέδου (**high-level**) βιβλιοθήκη δικτύωσης (**networking**) της Unity. Είναι ιδιαίτερα χρήσιμη στην ανάπτυξη multiplayer παιχνιδιών λόγω της **client-server** αρχιτεκτονικής που προσφέρει. Μπορεί εύκολα να εγκατασταθεί στο project μέσω του Package Manager της Unity [37].

Ένα χαρακτηριστικό της που την καθιστά εξαιρετικά σημαντική είναι οι υλοποιήσεις αρκετών πρωτοκόλλων επικοινωνίας που παρέχει. Ένα από αυτά τα πρωτόκολλα υλοποιεί το **Transport**. Το transport είναι μια **low-level class** η οποία χειρίζεται την αποστολή και λήψη δεδομένων μεταξύ server και clients [38].

Το Mirror ορίζει διάφορα transports ανάλογα με τις ανάγκες του προγραμματιστή. Μερικά από αυτά είναι το **Telepathy**, το οποίο βασίζεται στο πρωτόκολλο **TCP**. Έπειτα υπάρχει το **WebSockets & WebRTC**, το οποίο είναι για παιχνίδια που προορίζονται για browser. Τέλος, έχουμε το **KCP2k**, το οποίο βασίζεται στο UDP αλλά χρησιμοποιεί το KCP πρωτόκολλο για μεγαλύτερη αξιοπιστία και είναι προτιμότερο για **real-time multiplayer** παιχνίδια λόγω του χαμηλού **latency** [39].

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το **KCP Transport**. Η σύνδεση πραγματοποιείται μέσω Wi-Fi, ωστόσο μπορεί να τροποποιηθεί για να περιλαμβάνει και άλλα πρωτόκολλα επικοινωνίας.



Εικόνα 43: UML Diagram - Communication Architecture

Από το παραπάνω UML διάγραμμα γίνεται αμέσως αντιληπτό ότι για την υλοποίηση της αρχιτεκτονικής επικοινωνίας χρησιμοποιήθηκαν κυρίως διεπαφές. Η προσέγγιση αυτή κάνει τον κώδικα πιο ευανάγνωστο και πιο εύκολο για δοκιμές. Επιπλέον, ο προγραμματιστής θα μπορεί να αλλάξει κάποια υλοποίηση χωρίς να επηρεαστεί η υπόλοιπη αρχιτεκτονική.

3.5 Υλοποίηση Πλαισίου Επικοινωνίας (Communication Framework)

Μέχρι τώρα παρουσιάστηκε η κωδικοποίηση και η αποκωδικοποίηση των δεδομένων καθώς και η ανταλλαγή μηνυμάτων. Στην συνέχεια, λοιπόν, θα γίνει ανάλυση για το πώς θα χρησιμοποιηθεί η παραπάνω αρχιτεκτονική επικοινωνίας στην δημιουργία ενός πλαισίου όπου ένας ελεγκτής θα στέλνει τα δεδομένα σε μια ελεγχόμενη συσκευή. Στην παρούσα διπλωματική εργασία, ο ελεγκτής είναι ένα καθημερινό κινητό τηλέφωνο Android και η ελεγχόμενη συσκευή τα HoloLens 2. Τα δεδομένα που θα στέλνει το κινητό στα HoloLens θα λειτουργούν ως εισοδοί για τον χειρισμό τους. Για την επεξεργασία των εισόδων του κινητού, χρησιμοποιήθηκε εκτενώς το Σύστημα Εισόδου της Unity.

3.5.1 Είσοδοι του κινητού τηλεφώνου

Όπως αναφέρθηκε και προηγουμένως, χρησιμοποιήθηκε το Σύστημα Εισόδου της Unity για την κωδικοποίηση των εισόδων του ελεγκτή. Για κάθε συσκευή του κινητού που παρέχει δεδομένα, δημιουργήθηκε και ένας τύπος εισόδου. Παρακάτω παρουσιάζεται ένας πίνακας με τις δυνατές εισόδους.

Συσκευή	Είσοδος
Accelerometer	AccelerometerInput
AmbientTemperatureSensor	AmbientTemperatureSensorInput
AttitudeSensor	AttitudeSensorInput
Gamepad	GamepadInput
GravitySensor	GravitySensorInput
Gyroscope	GyroscopeInput
HumiditySensor	HumiditySensorInput
Keyboard	KeyboardInput

LightSensor	LightSensorInput
LinearAccelerationSensor	LinearAccelerationSensorInput
MagneticFieldSensor	MagneticFieldSensorInput
Mouse	MouseInput
PressureSensor	PressureSensorInput
ProximitySensor	ProximitySensorInput
StepCounter	StepCounterInput
TouchScreen	TouchScreenInput
TrackedDevice	TrackedDeviceInput

Οι βασικές κλάσεις που είναι καλό να αναλυθούν, φαίνονται παρακάτω:

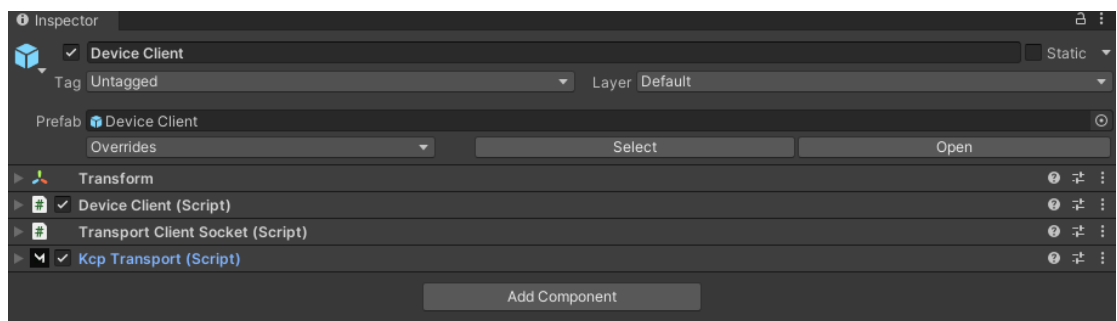
- **BaseInput:** Είναι μια αφαιρετική (abstract) κλάση η οποία κληρονομεί από την **SerializablePacket**, που αναφέρθηκε σε προηγούμενο κεφάλαιο. Περιέχει δύο μεθόδους. Την **SetUp** η οποία διαβάζει τα δεδομένα από μια συσκευή εισόδου και την **QueueInput** η οποία στέλνει τα δεδομένα μέσω του δικτύου, στις εισόδους που αναφέρονται στον πίνακα.
- **DeviceDescription:** Κληρονομεί από την κλάση **SerializablePacket**. Περιέχει τα δεδομένα που περιγράφουν μια συσκευή εισόδου. Είναι εξαιρετικά χρήσιμη για επιλογή συσκευών εισόδου στο σύστημα.
- **InputData:** Κληρονομεί από την κλάση **SerializablePacket**. Αντιπροσωπεύει τα δεδομένα εισόδου από μια συσκευή του συστήματος. Με αυτήν, ο σχεδιαστής μπορεί να προσθέτει ή να αφαιρεί συσκευές του ελεγκτή, ώστε να μην επηρεάζουν την υπόλοιπη εφαρμογή.
- **SubscriptionData:** Κληρονομεί από την κλάση **SerializablePacket**. Είναι ένα αντικείμενο που περιέχει μια λίστα με τις συσκευές εισόδου που στέλνονται στην ελεγχόμενη συσκευή.

- **DeviceData:** Κληρονομεί από την κλάση **SerializablePacket**. Αποθηκεύει δεδομένα εισόδου από συσκευές και πληροφορίες για events και καθυστέρηση δικτύου (latency).

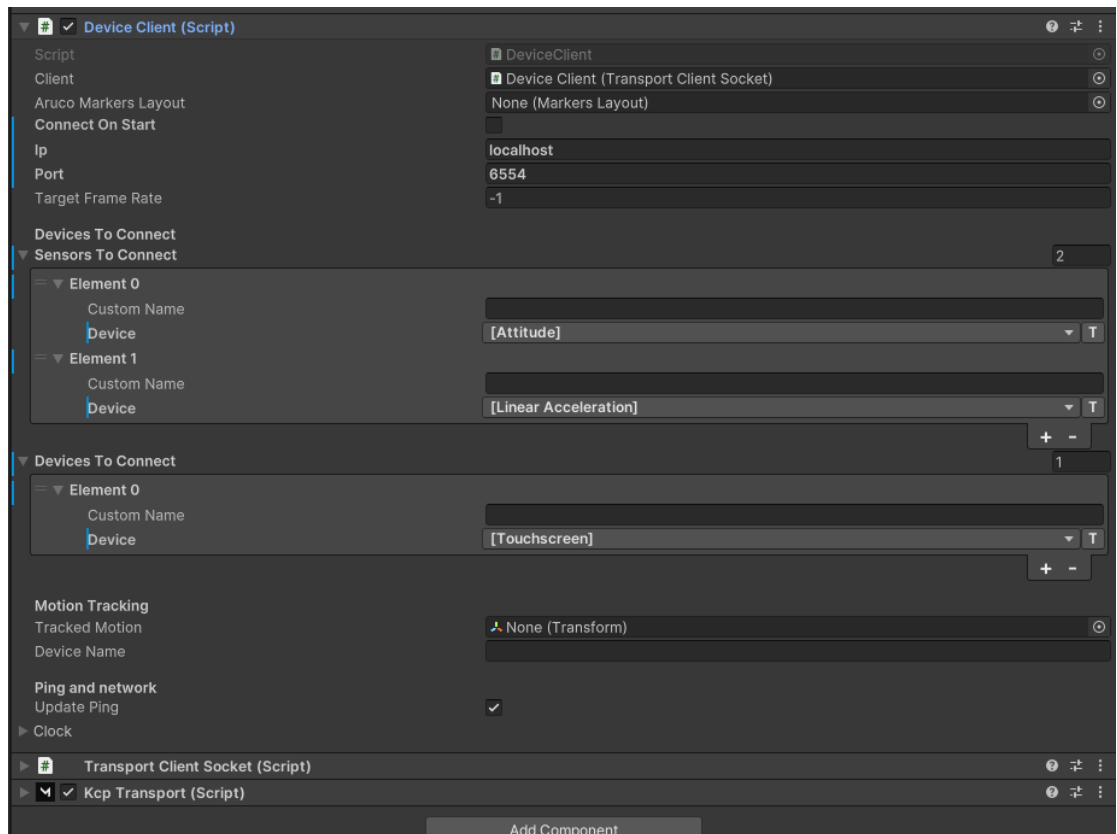
3.5.2 Υλοποίηση του πελάτη και του διακομιστή

Σε αυτό το κεφάλαιο ορίζονται δύο κλάσεις, η **DeviceClient** για τον πελάτη και η **DeviceServer** για τον διακομιστή. Κάθε μια από αυτές τις κλάσεις τοποθετούνται ως components σε κάποιο αντικείμενο, ώστε να μπορούν να ρυθμιστούν από τον editor της Unity. Να σημειωθεί ότι έχουν δημιουργηθεί δύο σκηνές στην Unity, όπου η μια αφορά τον πελάτη και η άλλη τον διακομιστή.

- **DeviceClient:** Η κλάση αυτή λειτουργεί ως πελάτης (client) και είναι υπεύθυνη για την σύνδεση με τον διακομιστή, δηλαδή τα HoloLens 2. Επιπλέον, διαχειρίζεται και αποστέλλει τα δεδομένα από τις συσκευές εισόδου του ελεγκτή μέσω του δικτύου. Τα δεδομένα που θα σταλθούν, τα διαχειρίζεται εύκολα ο προγραμματιστής μέσω του editor, όπως αναφέρθηκε προηγουμένως.



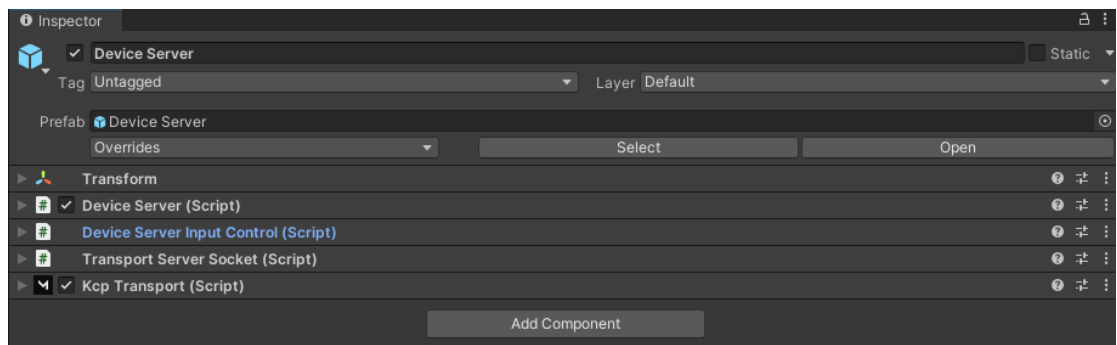
Εικόνα 44: Device Client στον Inspector



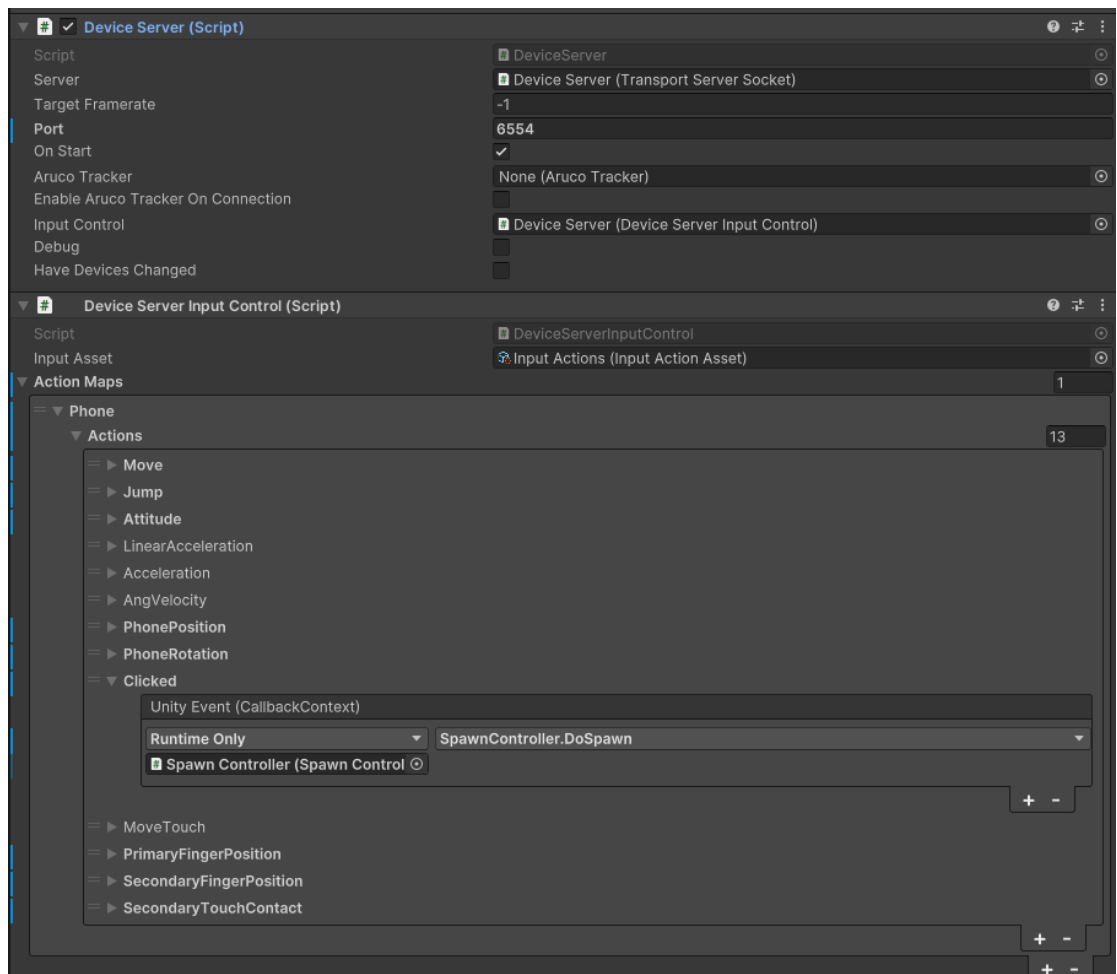
Εικόνα 45: Ρυθμίσεις της κλάσης DeviceClient

Στην εικόνα βλέπουμε τα σημεία που μπορεί να ρυθμίσει ο προγραμματιστής στην κλάση του πελάτη. Πιο συγκεκριμένα, έχει οριστεί ως default ip address ο localhost, ωστόσο αυτή την αλλάζει ο χρήστης της εφαρμογής με την ip της ελεγχόμενης συσκευής. Στέλνεται η πόρτα 6554, η οποία επιλέχθηκε με δοκιμές. Στην συνέχεια έχουμε τα βασικά σημεία της κλάσης αυτής. Ο προγραμματιστής επιλέγει εύκολα ποιες συσκευές εισόδου του κινητού θέλει να στέλνει στα HoloLens 2, ώστε να διαβάζονται και να χρησιμοποιούνται από τον διακομιστή.

- **DeviceServer:** Η κλάση αυτή λειτουργεί ως διακομιστής (server). Είναι υπεύθυνη για την λήψη και διαχείριση των δεδομένων που στέλνει ο client μέσω του δικτύου. Πιο συγκεκριμένα, μόλις συνδεθεί το κινητό τηλέφωνο, δημιουργεί εικονικές συσκευές εισόδου στο Σύστημα Εισόδου της Unity, ανάλογα με τα δεδομένα που στέλνει ο ελεγκτής. Τέλος, υποστηρίζει Aruco Tracking, λειτουργία που δεν χρησιμοποιείται στην παρούσα διπλωματική.



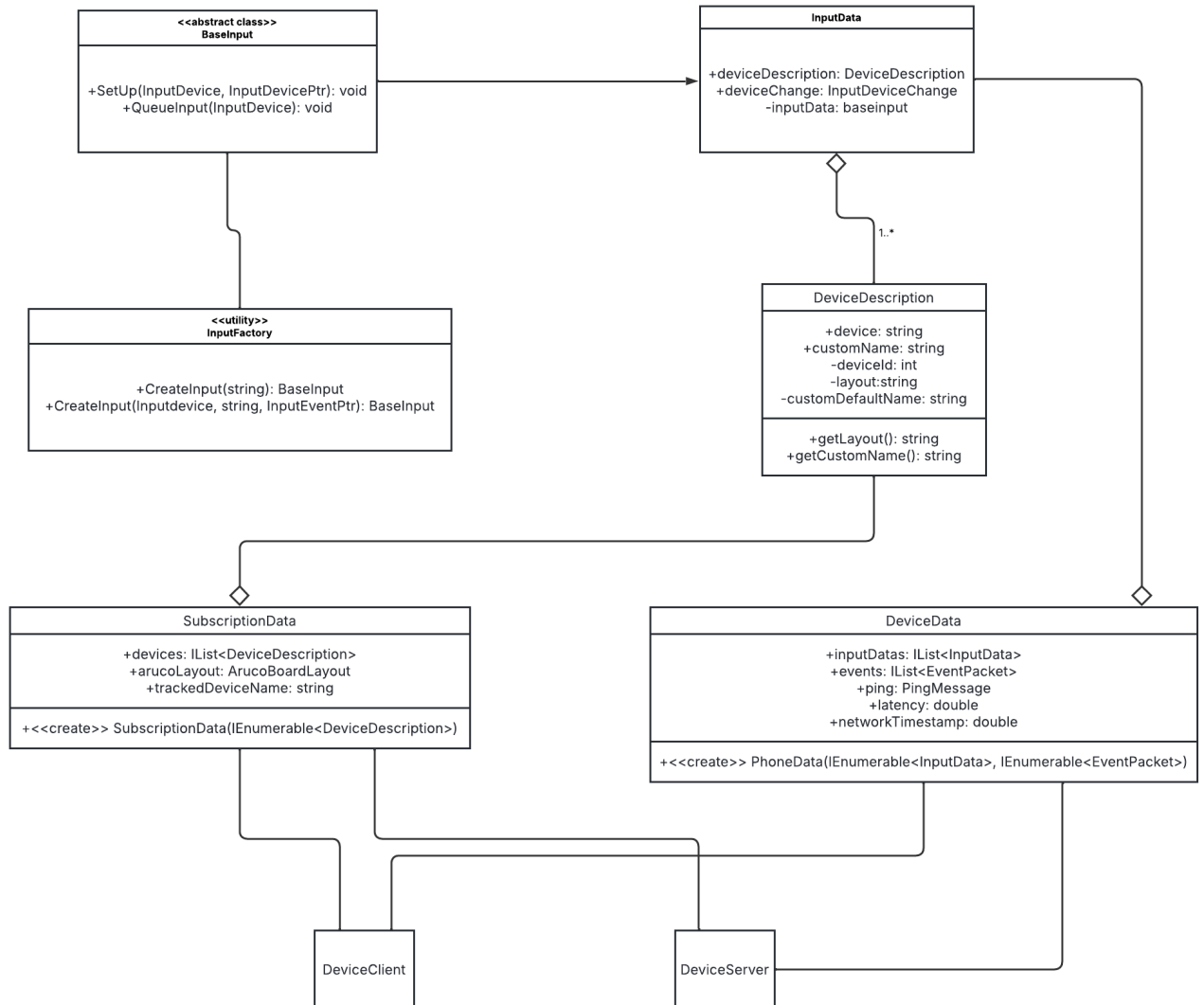
Εικόνα 46: Device Server στον Inspector



Εικόνα 47: Ρυθμίσεις της κλάσης DeviceServer

Όπως και στην κλάση του client, έτσι και εδώ στέλνουμε την πόρτα στην οποία «ακούν» οι συσκευές. Υπάρχει ιδιαίτερο ενδιαφέρον στην κλάση **DeviceServerInputControl**. Σε αυτήν, υπάρχει ένα Action Map με τις ενέργειες που

έχουν δημιουργηθεί από τον προγραμματιστή. Εδώ ο σχεδιαστής τοποθετεί ένα GameObject, το οποίο έχει ως component μια κλάση που υλοποιεί κάποια ενέργεια. Με αυτόν τον τρόπο, ο διακομιστής, αφού λάβει τα δεδομένα εισόδου από τον πελάτη, μπορεί να τα χρησιμοποιήσει για την ολοκλήρωση αυτής της ενέργειας.



Εικόνα 48: UML Diagram - Communication Framework

3.6 Σχεδιασμός Παιχνιδιού

Στο σημείο αυτό έχει ολοκληρωθεί το πλαίσιο επικοινωνίας μεταξύ του ελεγκτή και της ελεγχόμενης συσκευής. Επομένως, στην συνέχεια θα αναλυθεί ο σχεδιασμός και η υλοποίηση του παιχνιδιού, στο οποίο βρίσκουν εφαρμογή οι είσοδοι του ελεγκτή.

3.6.1 Ηλιακό Σύστημα και πλανήτες

Ο ήλιος και οι πλανήτες είναι **GameObjects** τύπου σφαίρας, στα οποία έχει τοποθετηθεί ένα εικονικό υλικό, το λεγόμενο **material**. Οι μάζες και οι αποστάσεις των πλανητών από τον ήλιο είναι σε αναλογία, ώστε να φαίνονται καλύτερα στο παιχνίδι. Να σημειωθεί ότι τα μεγέθη δεν είναι σε ακριβή αναλογία ώστε να φαίνονται καλύτερα στο μάτι του χρήστη. Ωστόσο αυτό δεν επηρεάζει τις εξισώσεις βαρύτητας που θα αναλυθούν στην συνέχεια. Τέλος, πρέπει να αναφερθεί ότι ως μάζα αναφερόμαστε στο **mass** του **Rigidbody** κάθε **GameObject**, ως μέγεθος είναι το **scale** στο **Transform** και η απόσταση από τον ήλιο είναι η **συντεταγμένη X**. Στον παρακάτω πίνακα, αναγράφονται οι τιμές για κάθε ουράνιο σώμα

Ουράνιο σώμα	Μάζα	Μέγεθος	Απόσταση από τον ήλιο
Ήλιος	3.33×10^7	6.96	0
Ερμής	0.825	1	8.14
Αφροδίτη	12.225	1.5	13
Γη	15	1.6	18.73
Άρης	1.605	1.2	25.59
Δίας	4770	2	31.89

Επίσης, έχει υλοποιηθεί το φως του ήλιου ώστε να φωτίζει τους πλανήτες στο ένα ημισφαίριο, όπως ακριβώς συμβαίνει και στην πραγματικότητα.

3.6.2 Τροχιά γύρω από τον ήλιο και ιδιοπεριστροφές

Για να έχουμε ένα ολοκληρωμένο μοντέλο του ηλιακού συστήματος, πρέπει να υλοποιηθούν και τα **βαρυτικά πεδία** των σωμάτων, ώστε να εκτελούν περιστροφή γύρω από τον ήλιο αλλά και γύρω από τον άξονά τους.

Ξεκινώντας, έχει δημιουργηθεί ένα **Empty GameObject**, το **SolarSystem**. Σε αυτό, έχει τοποθετηθεί ως component ένα script, η κλάση **PhysicsOrbit**. Αρχικά, θέτουμε την σταθερά της βαρύτητας ίση με **G = 0.0001f**, σε αντίθεση με την πραγματική της τιμή (6.67408×10^{-11}) ώστε να υπάρχει μια πιο ομοιόμορφη κίνηση των πλανητών, καθώς οι μάζες είναι πολύ μικρότερες λόγω αναλογίας. Ο προγραμματιστής μπορεί να αλλάξει το G σε περίπτωση που θέλει να επιταχύνει ή να επιβραδύνει την προσομοίωση, ώστε να μην χρειάζεται κάθε φορά να αλλάξει το timestep της Unity. Έπειτα, όλα τα ουράνια σώματα αποθηκεύονται σε μια λίστα. Σε κάθε ένα δίνουμε μια αρχική ταχύτητα, σύμφωνα με την **Νευτώνεια εξίσωση για κυκλική τροχιά**:

$$U = \sqrt{\frac{G \times M}{r}}$$

Στην συνέχεια, υπολογίζουμε την βαρυτική δύναμη που ασκεί κάθε σώμα “a”, στο “b”. Η δύναμη υπολογίζεται σύμφωνα με τον **Νόμο της παγκόσμιας έλξης του Νεύτωνα [40]**:

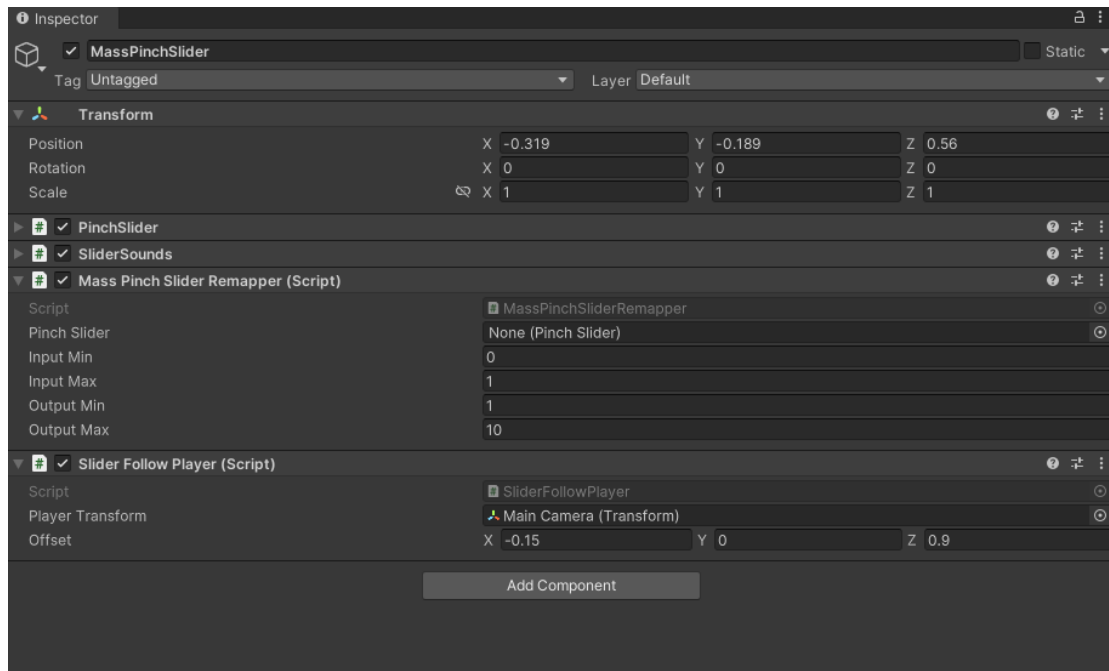
$$F = G \frac{m_1 m_2}{r^2}, \quad G = 6.67 \times 10^{-11}$$

και εφαρμόζεται σε όλα τα σώματα σε κάθε frame του παιχνιδιού.

Για την **ιδιοπεριστροφή** κάθε ουράνιου σώματος γύρω από τον άξονά του, δημιουργήθηκε η κλάση **PlanetRotation**, η οποία είναι component σε κάθε GameObject. Η υλοποίηση είναι πολύ απλή, αφού δεν χρειάζεται τίποτε άλλο παρά μια χρήση της συνάρτησης **Rotate** σε κάθε frame, στον άξονα Y. Η ταχύτητα περιστροφής ορίζεται από τον προγραμματιστή.

3.6.3 Πρώτος τρόπος παιχνιδιού – Χειριστήρια Hololens 2

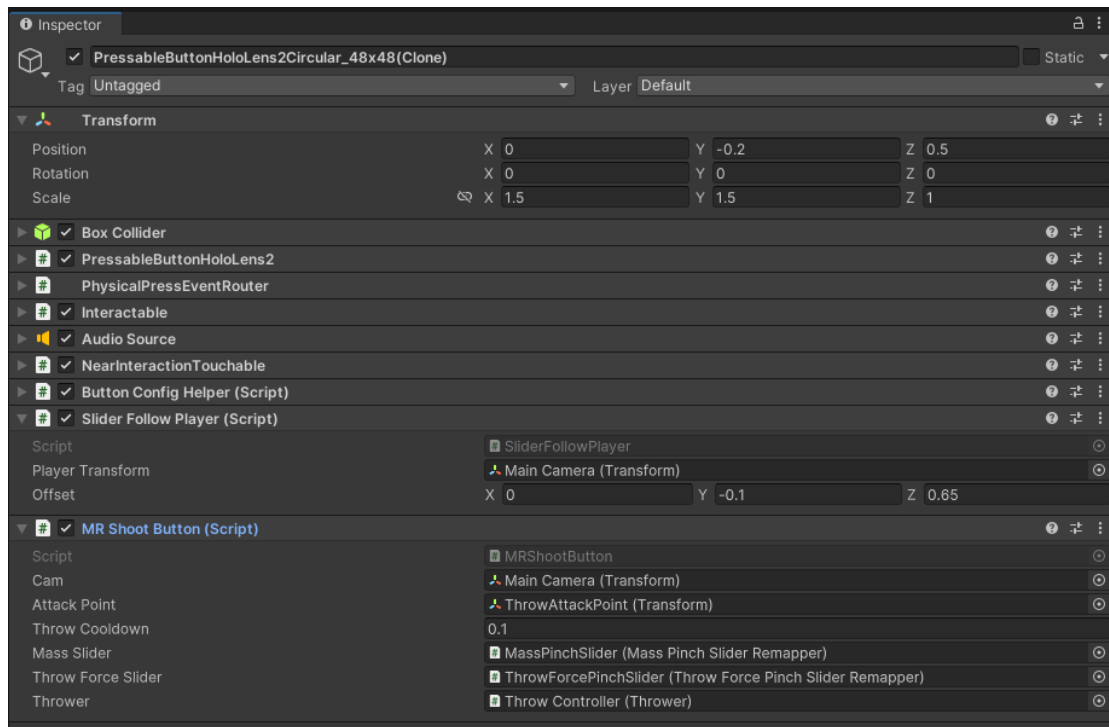
Όπως αναφέρθηκε και προηγουμένως, στην περιγραφή του παιχνιδιού, ο πρώτος τρόπος λειτουργίας γίνεται μόνο με χειρονομίες που αναγνωρίζει το Hololens. Χρησιμοποιήθηκαν δύο **MRTK PinchSliders**. Από την φύση τους, οι ακραίες τιμές τους είναι το 0 και το 1 [41]. Οπότε, δημιουργήθηκε μια κλάση, η **PinchSliderRemapper** για να αλλάξει τις ακραίες τιμές ανάλογα με τις ανάγκες του σχεδιαστή. Επιπλέον, τόσο οι sliders όσο και το κουμπί, ακολουθούν το οπτικό πεδίο του χρήστη, μέσω της κλάσης **FollowPlayer**. Ο χρήστης μπορεί να καθορίσει την θέση των «χειριστηρίων» αυτών από τον Inspector της Unity.



Εικόνα 49: Ο Inspector του MassPinchSlider

Το κουμπί είναι ένα **VisualTouchButton [42]**, του οποίου η λειτουργία είναι να εκτοξεύει ένα αντικείμενο. Οι κλάσεις που υλοποιούν αυτήν την ενέργεια είναι:

- **Thrower:** Η κλάση αυτή ορίζει μεθόδους για την ρίψη αντικειμένων με διαφορετικούς τρόπους. Αρχικά, υπάρχει η μέθοδος **ThrowWithCooldown** που χρησιμοποιεί την τεχνική των **Coroutines [43]**. Μέσα στην συνάρτηση, καλείται μια άλλη μέθοδος, η **Throw** για να δημιουργήσει και να ρίξει ένα αντικείμενο. Τότε, ξεκινά μια παράλληλη διεργασία που μετράει τον χρόνο **cooldown**. Μόλις ο χρόνος μηδενιστεί, η διεργασία τερματίζει και καλείται ξανά η **Throw**. Η **Throw** δημιουργεί ένα αντικείμενο το οποίο θα εκτοξευτεί από ένα σημείο εκτόξευσης. Χρησιμοποιεί την μάζα και την δύναμη εκτόξευσης και τα εφαρμόζει στο αντικείμενο που μόλις δημιουργήθηκε, εκτοξεύοντάς το.
- **MRShootButton:** Η κλάση αυτή χρησιμοποιείται για την ρίψη κάποιου αντικειμένου όταν ο χρήστης πατάει ένα εικονικό κουμπί. Ορίζει το σημείο εκτόξευσης, λαμβάνει τιμές από τους **sliders remappers** για την μάζα και την δύναμη ρίψης και καλεί χρησιμοποιεί ένα αντικείμενο τύπου **Thrower** για να πραγματοποιηθεί το shoot.



Εικόνα 50: Ο Inspector του shoot button

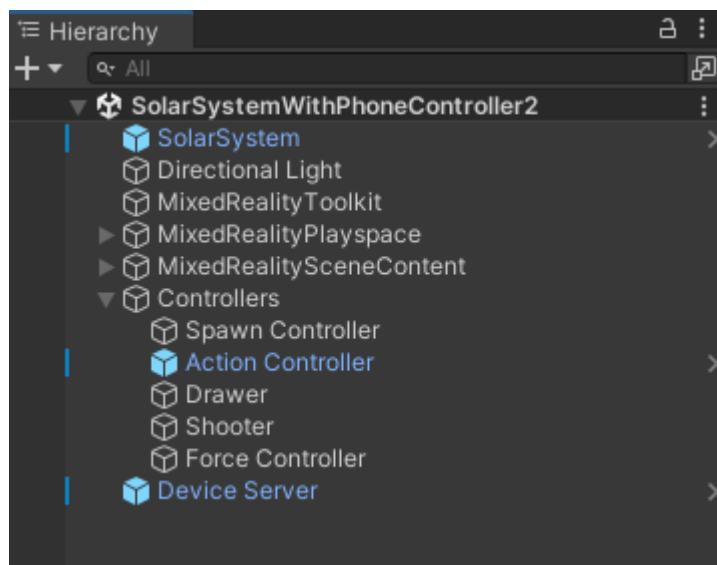
3.6.4 Δεύτερος τρόπος παιχνιδιού – Χειριστήρια κινητού τηλεφώνου

Ο δεύτερος τρόπος παιχνιδιού πραγματοποιείται με την επικοινωνία του κινητού-ελεγκτή με τα HoloLens. Η συσκευή εισόδου που στέλνει το κινητό είναι η οθόνη αφής του, το touchscreen δηλαδή. Έχουν υλοποιηθεί χειρονομίες και λειτουργίες για να αναγνωρίζει η ελεγχόμενη συσκευή. Πιο συγκεκριμένα, έχουμε τις εξής σημαντικές κλάσεις:

- **SpawnController:** Διαχειρίζεται την δημιουργία αντικειμένων σε ένα προκαθορισμένο σημείο. Ορίζει την μέθοδο **DoSpawn** η οποία χρησιμοποιεί **Callbacks** για να ελέγξει αν το input ξεκίνησε. Προφανώς το input είναι ένα απλό tap στο κινητό. Μόλις γίνει η ενέργεια, δημιουργεί το αντικείμενο στο σημείο που έχει οριστεί πάνω στην σκηνή και ενεργοποιεί κάποια scripts που θα αναλυθούν παρακάτω.
- **PinchDetector:** Ανιχνεύει χειρονομίες τύπου zoom από το κινητό τηλέφωνο και μεταβάλλει την μάζα και το μέγεθος ενός αντικειμένου αναλόγως. Σε αυτή την περίπτωση δεν χρησιμοποιούνται Callbacks. Αντιθέτως, δέχεται τις διαδρομές των ενεργειών (**Action Paths**) από τον χάρτη ενεργειών (**Action Map**) και αντιστοιχίζει τις ενέργειες. Ανιχνεύει αν και τα δύο δάχτυλα του παίκτη βρίσκονται σε επαφή με την οθόνη του κινητού και ανάλογα ξεκινάει μια Coroutine. Αποθηκεύει τις συντεταγμένες των δαχτύλων πάνω στην οθόνη και υπολογίζει την απόσταση μεταξύ τους. Αν η απόσταση αυξάνεται, τότε γίνεται zoom in και μεγαλώνει το

αντικείμενο, ενώ αντίθετα το μικραίνει. Αν ένα από τα δύο δάχτυλα χάσει την επαφή, τότε η coroutine τερματίζει.

- **ForceController:** Ρυθμίζει την δύναμη εκτόξευσης όταν ο παίκτης πατάει στην περιοχή του κουμπιού πάνω στην οθόνη αφής του κινητού. Και σε αυτήν την περίπτωση χρησιμοποιούνται τα action paths. Όσο το δάχτυλο του χρήστη βρίσκεται εντός της περιοχής του κουμπιού, ενεργοποιείται μια συν-ρουτίνα η οποία αυξομειώνει την δύναμη, εντός κάποιου ορίου. Μόλις ο χρήστης πατήσει εκτός αυτής της περιοχής, η δύναμη μένει σταθερή στην τιμή που είχε εκείνη την στιγμή.
- **ShootController:** Διαχειρίζεται την εκτόξευση ενός αντικειμένου μέσω ενός tap στο κινητό. Γίνεται έλεγχος αν πατήθηκε το κουμπί στην οθόνη του κινητού και στην συνέχεια παίρνει την μάζα του αντικειμένου και την δύναμη εκτόξευσης που ορίστηκαν προηγουμένως. Τελικά, σουτάρει το αντικείμενο με την ίδια μέθοδο του Thrower που αναπτύχθηκε στο προηγούμενο κεφάλαιο.
- **ActionsController:** Πρόκειται για ένα script που διαχειρίζεται την ομαλή λειτουργία των προαναφερθέντων ενεργειών. Ενεργοποιεί και απενεργοποιεί flags και scripts ώστε όλες οι ενέργειες να γίνονται με την σωστή σειρά και να μην γίνεται ανάμειξη των εισόδων



Εικόνα 51: Controllers

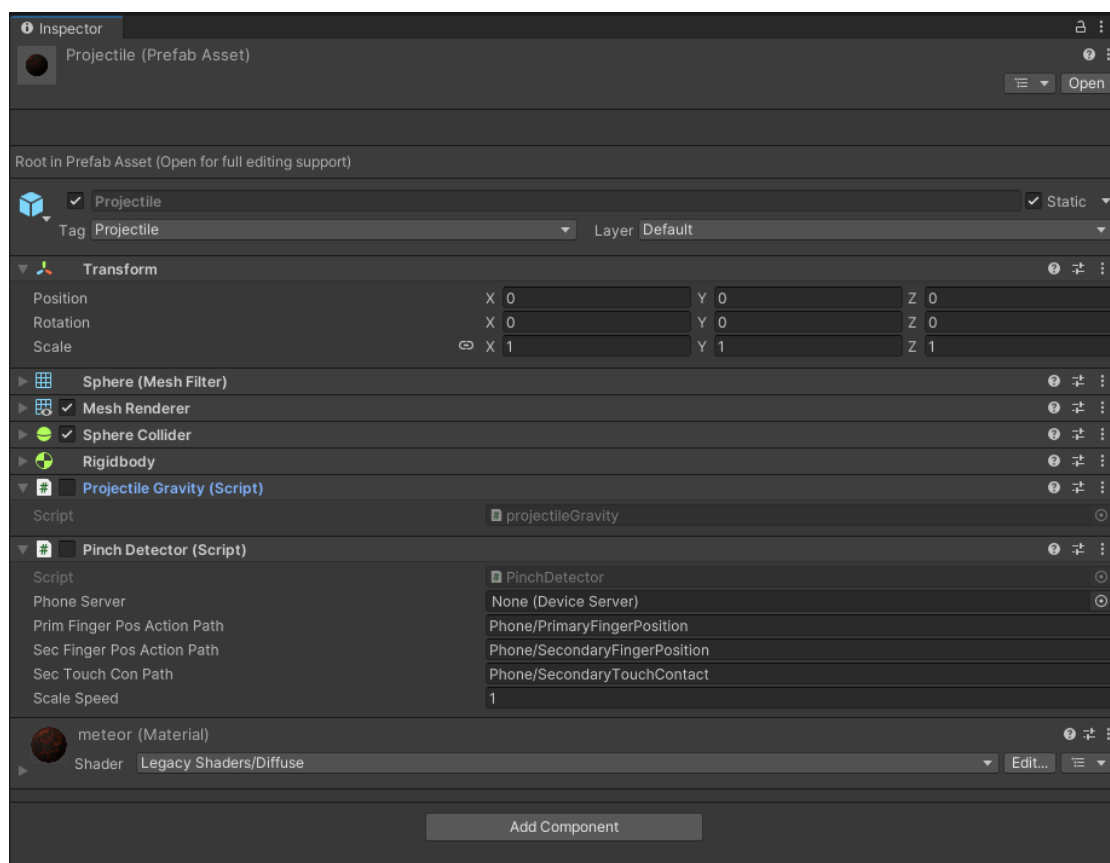
3.6.5 Μηχανισμός στόχευσης

Σε προηγούμενο κεφάλαιο, στην περιγραφή του παιχνιδιού, έγινε λόγος σε μια ακτίνα η οποία δείχνει την πορεία που θα ακολουθήσει η μπάλα. Η ακτίνα αυτή είναι ο **LineRenderer** της Unity, στον οποίο έχουν προστεθεί κλάσεις για να λειτουργεί όπως αναφέρει η περιγραφή. Συγκεκριμένα, υπάρχουν οι εξής κλάσεις:

- **BaseTrajectoryRenderer:** Είναι μια αφαιρετική κλάση που λειτουργεί ως βάση για τον σχεδιασμό της τροχιάς ενός αντικειμένου. Παρέχει μεθόδους που υλοποιούνται από την κλάση που την κληρονομεί.
- **LineTrajectoryRenderer:** Κληρονομεί την κλάση BaseTrajectoryRenderer. Χρησιμοποιεί την προσομοίωση της βαρύτητας που αναπτύχθηκε πιο πάνω, λαμβάνοντας υπόψη τόσο τα ουράνια σώματα του ηλιακού συστήματος όσο και τα αντικείμενα που έχουν εκτοξευθεί κατά την διάρκεια του παιχνιδιού. Επιπλέον, συνυπολογίζει και την αρχική ταχύτητα εκτόξευσης από την μάζα, την δύναμη και την θέση που έχει ορίσει ο χρήστης του παιχνιδιού.

3.6.6 Αντικείμενο εκτόξευσης

Το αντικείμενο εκτόξευσης είναι ένα GameObject τύπου σφαίρας, όπως οι πλανήτες και ο ήλιος. Περιέχει δύο κλάσεις ως components, όπου η μια είναι ο PinchDetector και η άλλη είναι η **ProjectileGravity**. Ουσιαστικά, η λειτουργία της είναι να εντάξει τον «κομήτη» στο βαρυτικό σύστημα των υπόλοιπων σωμάτων.

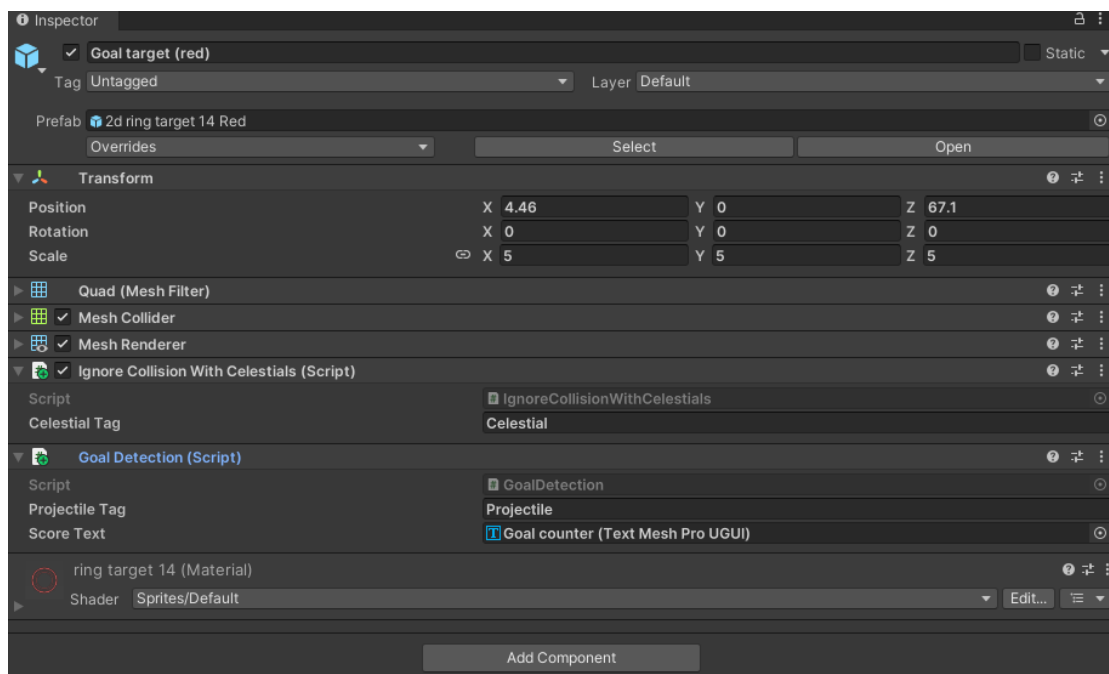


Εικόνα 52: Ο Inspector του projectile

3.6.7 Μηχανισμός ανίχνευσης goals

Στο GameObject του στόχου έχουν προσκολληθεί δύο script τα οποία συνολικά υλοποιούν την λειτουργία της καταμέτρησης γκολ.

- Η κλάση **IgnoreCollisionWithCelestials** παίρνει όλα τα αντικείμενα που έχουν δηλωθεί ως **celestials** και απενεργοποιεί τον **collider** του στόχου, με αποτέλεσμα να μην επιτρέπει την φυσική σύγκρουση αυτού με τα ουράνια σώματα.
- Η κλάση **GoalDetection** ανιχνεύει τότε ένα αντικείμενο με tag **projectile** περνά μέσα από τον δακτύλιο. Συγκεκριμένα, χρησιμοποιεί τον **trigger collider** του στόχου για να εξετάσει αν κάποιο αντικείμενο το διαπεράσει. Είναι ένας μηχανισμός που δεν επηρεάζεται από φυσικές συγκρούσεις και απλά τσεκάρει αν ένα δεύτερο αντικείμενο έρθει σε επαφή με αυτό [44].



Εικόνα 53: Ο Inspector του στόχου

4 Αξιολόγηση χειριστηρίων του κινητού

Στην ενότητα αυτή, παρουσιάζεται η οργάνωση και τα αποτελέσματα του πειράματος.

4.1 Σκοπός του πειράματος

Για να έχουμε ένα ολοκληρωμένο έργο, είναι απαραίτητο μαζί με την υλοποίηση, να διεξαχθεί και μια έρευνα ώστε να εξετάσει ο σχεδιαστής την χρησιμότητα και το πόσο φιλική προς τους χρήστες είναι η εφαρμογή του. Για τον λόγο αυτό, πραγματοποιήθηκαν πειράματα, όπου διάφορα άτομα δοκίμασαν το συγκεκριμένο παιχνίδι με τους δύο τρόπους αλληλεπίδρασης.

4.2 Το πείραμα

4.2.1 Περιγραφή του πειράματος

Το άτομο εισέρχεται στον χώρο του πειράματος. Στην συνέχεια, καλείται να βάλει όσα περισσότερα goal μπορεί στον στόχο που έχει τοποθετηθεί πίσω από τον ήλιο, μέσα σε 5 λεπτά. Ο χρήστης παίζει το παιχνίδι με 2 τρόπους.

1. Με τις χειρονομίες που προσφέρει το HoloLens 2. Συγκεκριμένα, ελέγχει τους sliders με το air-tap και πατάει το κουμπί με παρόμοιο τρόπο. Για να στοχεύσει χρησιμοποιεί τον δείκτη του κεφαλιού του και το βλέμμα του, μαζί με την ακτίνα της τροχιάς.
2. Με τα χειριστήρια του κινητού τηλεφώνου που στέλνονται στα γυαλιά, μέσω του πλαισίου επικοινωνίας που υλοποιήθηκε. Ωστόσο, ο τρόπος που το υποκείμενο στοχεύει για να βάλει goal γίνεται πάλι με το στρίψιμο της κεφαλής και με τη χρήση του βλέμματος.

Μέσα από αυτή τη διαδικασία, γίνεται σύγκριση στην επίδοση των παικτών ανάμεσα στους δύο τρόπους αλληλεπίδρασης.

4.2.2 Οργάνωση και διεξαγωγή του πειράματος

Τα πειράματα πραγματοποιήθηκαν στο τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, στον χώρο του εργαστηρίου Συστημάτων Υπολογιστών. Ο συνολικός αριθμός των συμμετεχόντων ήταν 8.

Κάθε άτομο ερχόταν μεμονωμένα στον χώρο του πειράματος. Στην αρχή, γινόταν μια σύντομη εισαγωγή του θέματος της διπλωματικής εργασίας καθώς και κάποιων βασικών εννοιών. Έπειτα, γινόταν περιγραφή του πειράματος, του παιχνιδιού και του σκοπού που τους είχε ανατεθεί. Πριν από την εκκίνηση κάθε παιχνιδιού, δινόταν η δυνατότητα στα άτομα να πειραματιστούν τόσο με τις χειρονομίες του HoloLens όσο και με τα χειριστήρια του κινητού, σε κάποιες σκηνές που είχαν δημιουργηθεί για αυτό τον λόγο. Ο χρόνος αφηνόταν στην ευχέρεια του κάθε ατόμου. Κάθε υποκείμενο ξεκινούσε με διαφορετική

λειτουργία παιχνιδιού κάθε φορά. Η σειρά γινόταν εναλλάξ, ώστε να υπάρχει κάποια τυχαιότητα στο πείραμα.

Μετά το πέρας κάθε παιχνιδιού, δινόταν ένα ερωτηματολόγιο εμπειρίας χρήστη που αφορούσε τον χειρισμό που είχε μόλις χρησιμοποιηθεί. Τα άτομα συμπλήρωναν το όνομά τους, κατέγραφαν το σκορ τους και απαντούσαν στις ερωτήσεις.

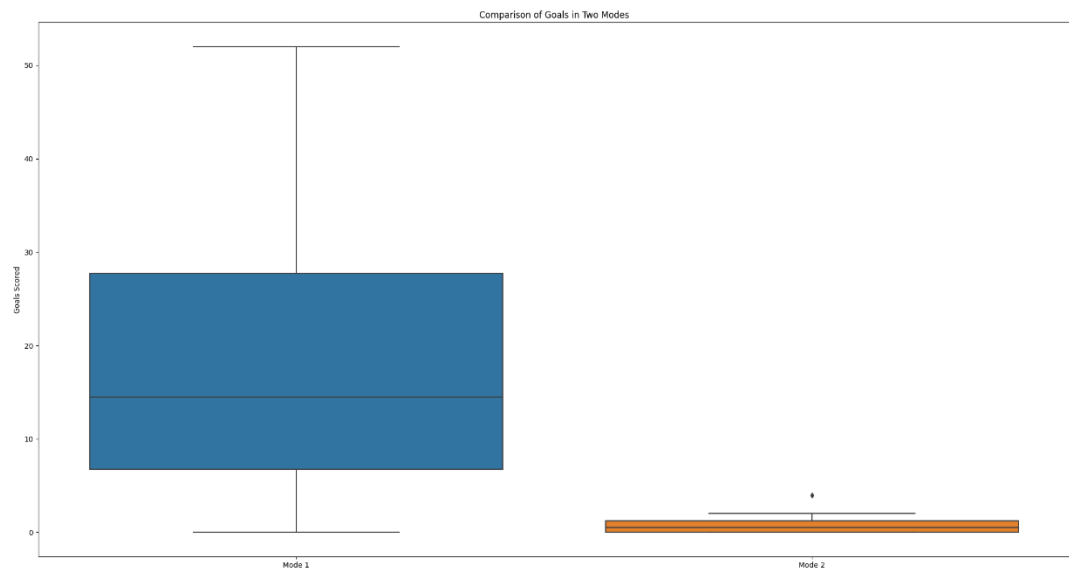
4.3 Επίτευξη στόχου

Στην συνέχεια, αναλύονται τα αποτελέσματα των πειραμάτων. Θέλουμε να εξετάσουμε σε ποια από τις δύο περιπτώσεις ο χρήστης πέτυχε περισσότερες φορές τον στόχο. Για την στατιστική ανάλυση χρησιμοποιήθηκε το **Wilcoxon signed-rank test [45]**, το οποίο εξετάζει αν οι διαφορές μεταξύ των δειγμάτων είναι συστηματικά θετικές ή αρνητικές. Για να το χρησιμοποιήσουμε πρέπει να κάνουμε κάποιους ελέγχους. Συγκεκριμένα:

1. Να εξετάσουμε αν υπάρχουν ακραίες τιμές στα δείγματα
2. Αν ακολουθούν μη κανονική κατανομή

Παρακάτω παρουσιάζονται τα διαγράμματα «κουτιού» για τα σκορ των παικτών όταν:

- Χρησιμοποιούν τις χειρονομίες του Hololens (μπλε)
- Χρησιμοποιούν το κινητό ως ελεγκτή (πορτοκαλί)



Εικόνα 54: Box Diagrams of two gameplays

Στα διαγράμματα δεν παρατηρούμε κάποιες ακραίες τιμές.

Για να εξετάσουμε την κανονικότητα, υπολογίζουμε το **p-value**. Υπολογίστηκε:

$p = 0.0234 < 0.05$ του **Shapiro-Wilk**. Επομένως, τα δείγματα δεν είναι κανονικά κατανομημένα.

Άρα μπορούμε να προχωρήσουμε με το **Wilcoxon signed – rank test**.

Υπολογίζουμε τις μέσες τιμές των σκορ στις δύο περιπτώσεις:

- Gameplay 1 (με Hololens): 19.625
- Gameplay 2 (με κινητό-ελεγκτή): 1.0

Και τις τυπικές αποκλίσεις:

- Gameplay 1 (με Hololens): 18.647
- Gameplay 2 (με κινητό-ελεγκτή): 1.41

Εξετάζουμε την εξής υπόθεση:

Ο παίκτης βάζει περισσότερα goal χρησιμοποιώντας το κινητό.

Αν **p-value < 0.05**, υπάρχει στατιστικά σημαντική διαφορά στα σκορ των δύο περιπτώσεων. Βρίσκουμε ότι: **p-value = 0.0234 < 0.05**.

Επομένως, μπορούμε να αποφανθούμε πως:

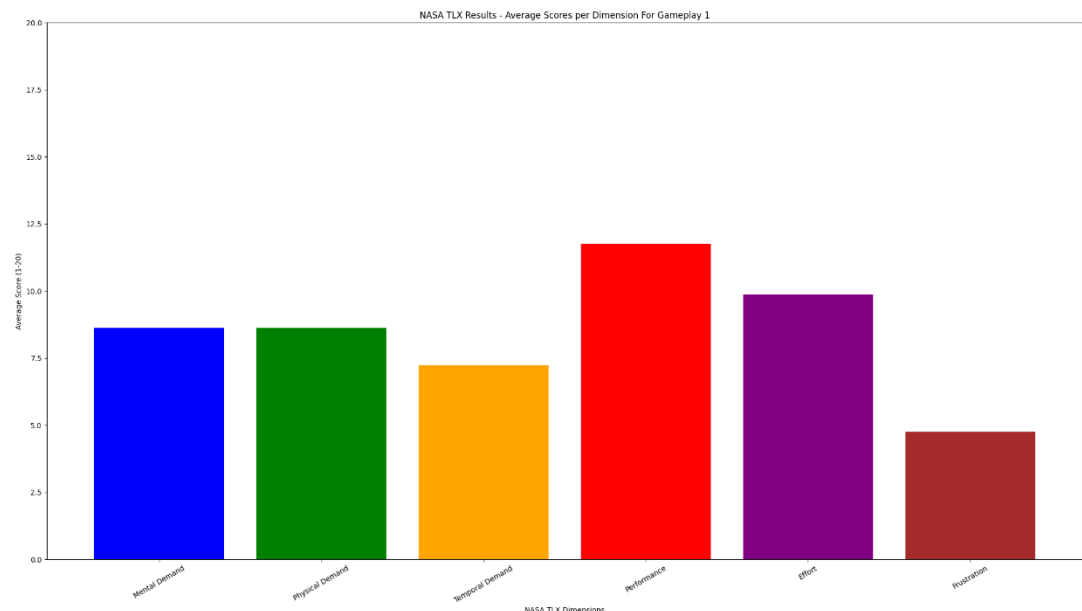
Τα χειριστήρια του κινητού δεν διευκόλυναν τους παίκτες να πετύχουν τον στόχο.

4.3 Ερωτηματολόγιο

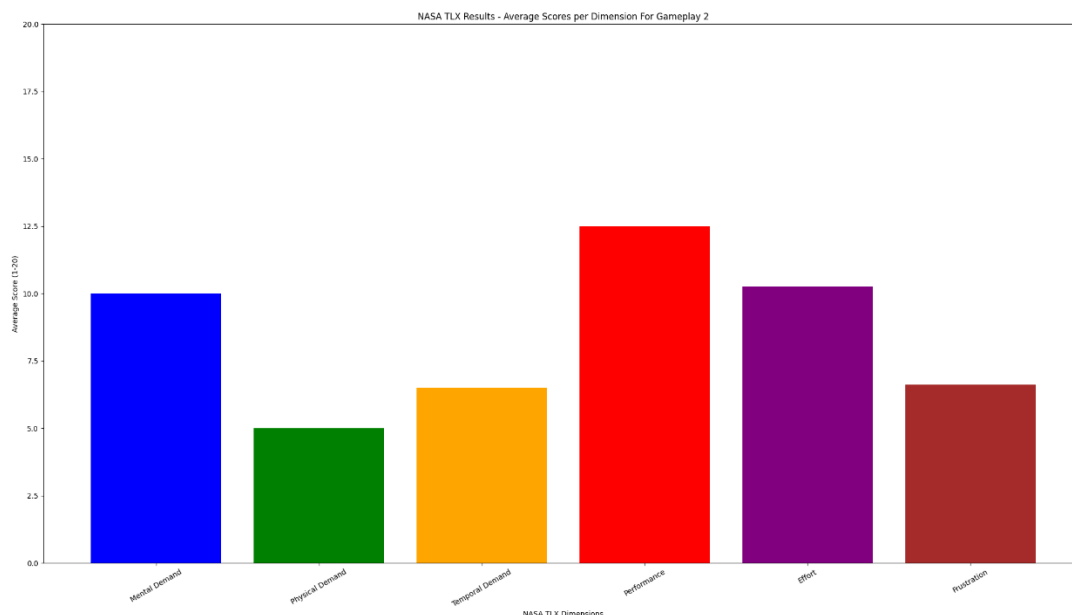
Μόλις ολοκληρωνόταν κάθε μέρος του πειράματος, δινόταν ένα ερωτηματολόγιο που αφορά το **user experience**. Το ερωτηματολόγιο που δόθηκε ήταν το **NASA – TLX [46]**. Οι ερωτήσεις είχαν βαθμολογική κλίμακα από 1-20, όπου 1 – Very Low και 20 – Very High. Οι ερωτήσεις ήταν οι εξής:

1. How mentally demanding was the task?
2. How physically demanding was the task?
3. How hurried or rushed was the pace of the task?
4. How successful were you in accomplishing what you were asked to do?
5. How hard did you have to work to accomplish your level of performance?
6. How insecure, discouraged, irritated, stressed and annoyed were you?

Παρακάτω φαίνονται τα αποτελέσματα:



Εικόνα 55: Chart απαντήσεων - Χειρισμός με Holograms



Εικόνα 56: Chart απαντήσεων - Χειρισμός με κινητό τηλέφωνο

Μπορεί να γίνει αντιληπτό από τις απαντήσεις των χρηστών, ότι δεν υπήρχε ιδιαίτερη «πνευματική» κούραση, ο ρυθμός δεν ήταν βεβιασμένος και δεν ένιωσαν κάποιον εκνευρισμό. Ωστόσο, όπως αποδείχτηκε και στην στατιστική ανάλυση, θεώρησαν ότι τα κατάφεραν πολύ καλύτερα με τις χειρονομίες του HoloLens. Παρά το γεγονός αυτό, ένιωσαν πολύ πιο άνετοι με το κινητό και κουράστηκαν αρκετά λιγότερο σωματικά.

Τέλος, να σημειωθεί ότι το feedback που ελήφθη από όλους είναι ότι βρήκαν την ρύθμιση της μάζας πιο εύκολη με τους sliders, αλλά υπήρξε μια προτίμηση στην εκτόξευση του αντικειμένου με το κινητό.

5 Επίλογος

Τα γυαλιά μεικτής πραγματικότητας HoloLens 2 υποστηρίζουν φυσικούς τρόπους αλληλεπίδρασης χρησιμοποιώντας τα χέρια, όμως το περιορισμένο πεδίο αντίληψης μπορεί να επηρεάσει την εμπειρία του χρήστη. Στο πλαίσιο της διπλωματικής εργασίας, αναπτύχθηκε ένα σύστημα επικοινωνίας που επεκτείνει τις δυνατότητες αλληλεπίδρασης, προσφέροντας εναλλακτικούς τρόπους χειρισμού, μέσω χειρονομιών και ενεργειών πάνω στην οθόνη ενός κινητού τηλεφώνου.

Δημιουργήθηκε ένα πλαίσιο στο οποίο η ελεγχόμενη συσκευή αναγνωρίζει ενέργειες που πραγματοποιούνται με ένα κινητό τηλέφωνο, παρέχοντας στον χρήστη έναν πιο άνετο και καθημερινό τρόπο αλληλεπίδρασης. Ωστόσο, αξίζει να αναφερθεί ότι η εφαρμογή δεν δουλεύει τέλεια, λόγω της τεχνολογίας του κινητού ή μικρών σφαλμάτων κατά την υλοποίηση της εργασίας.

Τέλος, μέσα από το πείραμα, αξιολογήθηκε αυτός ο νέος τρόπος χειρισμού της μεικτής πραγματικότητας, καλώντας τους χρήστες να πετύχουν έναν στόχο, πετώντας μια μπάλα μέσα από το ηλιακό σύστημα χρησιμοποιώντας το κινητό. Παρακάτω, αναφέρονται συνοπτικά τα συμπεράσματα:

1. Οι χρήστες πέτυχαν περισσότερες φορές τον στόχο χρησιμοποιώντας τον εγγενή τρόπο αλληλεπίδρασης που προσφέρει το HoloLens.
2. Οι χρήστες βρήκαν την εμπειρία πιο ξεκούραστη χρησιμοποιώντας το κινητό τηλέφωνο.

Παρά το γεγονός ότι στην παρούσα περίπτωση, το πείραμα έδειξε ότι οι χρήστες είχαν μικρότερο ποσοστό επιτυχίας χρησιμοποιώντας την εφαρμογή του κινητού τηλεφώνου, αξίζει να ερευνηθεί περαιτέρω αυτό το εγχείρημα καθώς η αλληλεπίδραση ήταν πιο ξεκούραστη και ευχάριστη.

Σημείωση: Οι κώδικες της διπλωματικής εργασίας βρίσκονται στο προσωπικό μου GitHub [48].

Αναφορές

- [1] K. C. I. G. J.M. Zheng,
«<https://ieeexplore.ieee.org/document/666641>,» 2002.
[Ηλεκτρονικό].
- [2] A. S. Gillis,
«<https://www.techtarget.com/whatis/definition/augmented-reality-AR>,» [Ηλεκτρονικό].
- [3] «<https://learn.microsoft.com/en-us/windows/mixed-reality/discover/mixed-reality>,» [Ηλεκτρονικό].

- [4] «<https://el.wikipedia.org/wiki/Τετραδόνιο>,» [Ηλεκτρονικό].
- [5] J. Wyss-Gallifent,
«https://math.umd.edu/~immortal/MATH431/book/ch_quaternions.pdf,» 5 December 2021. [Ηλεκτρονικό].
- [6] «<https://en.wikipedia.org/wiki/Odometry>,» [Ηλεκτρονικό].
- [7] A. B.-H. R. H. Khalid Yousif, «ResearchGate,» [Ηλεκτρονικό].
Available:
https://www.researchgate.net/publication/284136264_An_Overview_to_Visual_Odometry_and_Visual_SLAM_Applications_to_Mobile_Robotics.
- [8] Y. Bai, «ScienceDirect,» 2023. [Ηλεκτρονικό]. Available:
<https://www.sciencedirect.com/topics/computer-science/visual-odometry>.
- [9] [Ηλεκτρονικό]. Available:
<https://www.thinkautonomous.ai/blog/visual-inertial-odometry/>.
- [10] [Ηλεκτρονικό]. Available:
https://docs.px4.io/v1.11/en/computer_vision/visual_inertial_odometry.html.
- [11] «Microsoft,» [Ηλεκτρονικό]. Available:
<https://learn.microsoft.com/en-us/hololens/hololens2-hardware>.

- [12] «Microsoft,» [Ηλεκτρονικό]. Available:
<https://learn.microsoft.com/en-us/windows/mixed-reality/design/eye-tracking>.
- [13] «Microsoft,» [Ηλεκτρονικό]. Available:
<https://learn.microsoft.com/en-us/windows/mixed-reality/design/eye-gaze-interaction>.
- [14] «Microsoft,» [Ηλεκτρονικό]. Available:
<https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unreal/unreal-hand-tracking?tabs=426>.
- [15] «Microsoft,» [Ηλεκτρονικό]. Available:
<https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/input/hand-tracking?view=mrtkunity-2022-05>.
- [16] «Unity,» [Ηλεκτρονικό]. Available:
<https://docs.unity.com/visuallive/en/manual/holo-gettingstarted-gestures>.
- [17] «Microsoft,» [Ηλεκτρονικό]. Available:
<https://learn.microsoft.com/en-us/dynamics365/mixed-reality/guides/operator-gestures-hl2>.
- [18] «Microsoft,» [Ηλεκτρονικό]. Available:
<https://learn.microsoft.com/en-us/windows/mixed-reality/design/system-gesture>.
- [19] «Microsoft,» [Ηλεκτρονικό]. Available:
<https://learn.microsoft.com/en-us/hololens/hololens-cortana>.

- [20] «Wikipedia,» [Ηλεκτρονικό]. Available:
[https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)).
- [21] «Unity discussions,» Unity, [Ηλεκτρονικό]. Available:
<https://discussions.unity.com>.
- [22] «Unity documentation,» Unity, [Ηλεκτρονικό]. Available:
<https://docs.unity3d.com/Manual/index.html>.
- [23] «Unity Documentation,» Unity, [Ηλεκτρονικό]. Available:
<https://docs.unity3d.com/Manual/GameObjects.html>.
- [24] «Unity Documentation,» Unity, [Ηλεκτρονικό]. Available:
<https://docs.unity3d.com/6000.0/Documentation/Manual/class-GameObject.html>.
- [25] «Unity Documentation,» Unity, [Ηλεκτρονικό]. Available:
<https://docs.unity3d.com/Manual/Components.html>.
- [26] «Unity Documentation,» Unity, [Ηλεκτρονικό]. Available:
<https://docs.unity3d.com/Manual/scripting-get-started.html>.
- [27] «Unity Documentation,» Unity, [Ηλεκτρονικό]. Available:
<https://docs.unity3d.com/6000.0/Documentation/Manual/execution-order.html>.
- [28] «Unity Hub,» Unity, [Ηλεκτρονικό]. Available:
<https://unity.com/unity-hub>.
- [29] «Wikipedia,» [Ηλεκτρονικό]. Available:
https://en.wikipedia.org/wiki/Mixed_Reality_Toolkit.

- [30] «Microsoft learn,» Microsoft, 05 2022. [Ηλεκτρονικό]. Available: <https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05>.
- [31] «Microsoft learn,» Microsoft, [Ηλεκτρονικό]. Available: <https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk3-overview/>.
- [32] «Microsoft Learn,» Microsoft, 14 12 2022. [Ηλεκτρονικό]. Available: <https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/welcome-to-mr-feature-tool>.
- [33] «Unity Docs,» Unity, [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.13/manual/index.html>.
- [34] «Unity Docs,» Unity, [Ηλεκτρονικό]. Available: <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.4/manual/Actions.html#action-callbacks>.
- [35] «Wikipedia,» [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/Serialization>.
- [36] «GitHub,» [Ηλεκτρονικό]. Available: <https://github.com/alvyxaz/barebones-masterserver>.
- [37] «Mirror Networking,» Mirror, [Ηλεκτρονικό]. Available: <https://mirror-networking.gitbook.io/docs>.
- [38] «GitHub,» [Ηλεκτρονικό]. Available: <https://github.com/MirrorNetworking/Mirror>.

- [39] «Mirror Networking,» Mirror, [Ηλεκτρονικό]. Available:
<https://mirror-networking.gitbook.io/docs/manual/transport>.
- [40] «Wikipedia,» [Ηλεκτρονικό]. Available:
https://el.wikipedia.org/wiki/Σταθερά_της_βαρύτητας.
- [41] «Microsoft Learn,» Microsoft, [Ηλεκτρονικό]. Available:
<https://learn.microsoft.com/en-us/dotnet/api/microsoft.mixedreality.toolkit.ui.pinchslider?view=mixed-reality-toolkit-unity-2020-dotnet-2.8.0>.
- [42] «Microsoft Learn,» Microsoft, [Ηλεκτρονικό]. Available:
<https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/ux-building-blocks/button?view=mrtkunity-2022-05>.
- [43] «Unity Documentation,» Unity, [Ηλεκτρονικό]. Available:
<https://docs.unity3d.com/6000.0/Documentation/ScriptReference/Coroutine.html>.
- [44] «Unity Documentation,» Unity, [Ηλεκτρονικό]. Available:
<https://docs.unity3d.com/6000.0/Documentation/ScriptReference/Collider.OnTriggerEnter.html>.
- [45] «Wikipedia,» [Ηλεκτρονικό]. Available:
https://en.wikipedia.org/wiki/Wilcoxon_signed-rank_test.
- [46] «NTRS,» NASA, 1 1 1986. [Ηλεκτρονικό]. Available:
<https://ntrs.nasa.gov/citations/20000021487>. [Πρόσβαση 7 September 2013].

- [47] Π. Γεώργιος, «Nemertes,» UPatras, 11 10 2022. [Ηλεκτρονικό]. Available: <https://nemertes.library.upatras.gr/items/d14fb696-3763-4346-929e-eacda47c6bf0>.
- [48] «GitHub,» [Ηλεκτρονικό]. Available: <https://github.com/MichaelChios/Diploma-Thesis>.