

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
«ΧΡΗΣΗ ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΚΑΙ
ΑΝΑΓΝΩΡΙΣΗΣ ΧΕΙΡΩΝΟΜΙΩΝ ΣΕ ΠΕΙΡΑΜΑΤΑ
ΧΗΜΕΙΑΣ ΔΕΥΤΕΡΗΣ ΤΑΞΗΣ ΤΟΥ ΓΥΜΝΑΣΙΟΥ»



Του φοιτητή
Μάρκου Καλέμη
Αρ. Μητρώου: 2019055

Επιβλέπων Καθηγητής
Ευκλείδης Κεραμόπουλος

Ημερομηνία 31/05/2026

Τίτλος Δ.Ε. Χρήση επαυξημένης πραγματικότητας και αναγνώρισης χειρονομιών σε πειράματα χημείας
δεύτερης τάξης του γυμνασίου.

Κωδικός Δ.Ε. 25231

Ονοματεπώνυμο φοιτητή Μάρκος Καλέμης

Ονοματεπώνυμο εισηγητή Ευκλείδης Κεραμόπουλος

Ημερομηνία ανάληψης Δ.Ε. 30/3/2025

Ημερομηνία περάτωσης Δ.Ε. 31/5/2026

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

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

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

«Στους Αγαπημένους μου»

Πρόλογος

Με την ραγδαία ανάπτυξη που έχει παρατηρηθεί τις τελευταίες δεκαετίες στον τομέα της πληροφορικής, οι μέθοδοι με τις οποίες πραγματοποιούμε τις εργασίες της καθημερινότητας αλλάζουν ριζικά. Ένας από τους τομείς είναι η εκπαίδευση, καθώς οι σύγχρονοι μαθητές μεγαλώνουν σε ένα περιβάλλον όπου η χρήση έξυπνων συσκευών και άλλων διαδραστικών συστημάτων αποτελεί αδιάσπαστο κομμάτι της καθημερινότητάς τους. Με αυτά τα δεδομένα στον νου, η ενσωμάτωση νέων εργαλείων μάθησης, δεν αποτελεί πλέον μια πολυτέλεια, αλλά μια βασική ανάγκη για την ανάπτυξη του σημερινού εκπαιδευτικού συστήματος. Η βασική αυτή ανάγκη με ώθησε στην εκπόνηση της διπλωματικής μου εργασίας, ώστε με την περάτωση της να συνεισφέρω στον εκσυγχρονισμό των εργαλείων μάθησης του σημερινού εκπαιδευτικού συστήματος. Στο πλαίσιο αυτό, η επαυξημένη πραγματικότητα (Augmented Reality-AR) και τα συστήματα αναγνώρισης χειρονομιών αποδεικνύονται δύο πολλά υποσχόμενες τεχνολογίες για την σύγχρονη εκπαίδευση και έρευνα.

Περίληψη

Η παρούσα διατριβή αφορά τον σχεδιασμό και τη δημιουργία μιας εκπαιδευτικής εφαρμογής επαυξημένης πραγματικότητας που χρησιμοποιεί αναγνώριση χειρονομιών για την παράδοση των μαθημάτων χημείας του δεύτερου έτους. Ο στόχος της εφαρμογής είναι να δημιουργήσει ένα διαδραστικό και ασφαλές εκπαιδευτικό μοντέλο που θα επιτρέπει στους μαθητές να έρχονται σε επαφή με πειραματικές διαδικασίες που είναι δύσκολες ή επικίνδυνες να πραγματοποιηθούν σε συμβατικά εκπαιδευτικά ιδρύματα.

Το πρόγραμμα δημιουργήθηκε στην πλατφόρμα Android και χρησιμοποιεί την τεχνολογία της επαυξημένης πραγματικότητας για την οπτικοποίηση τρισδιάστατων αντικειμένων σε φυσικούς εκπαιδευτικούς πόρους, καθώς και μεθόδους αναγνώρισης χειρονομιών για την αλληλεπίδραση με τα αντικείμενα με φυσικό και διαισθητικό τρόπο, εξαλείφοντας έτσι την ανάγκη για την κλασική εισαγωγή δεδομένων. Το πείραμα που υιοθετήθηκε αφορά την καύση χημικών στοιχείων και την παρακολούθηση της τυπικής χρωματικής διακύμανσης της φλόγας, όπου ο χρήστης μπορεί να λάβει άμεση οπτική ανατροφοδότηση.

Κατά τη διάρκεια του έργου, πραγματοποιήθηκαν η ανάλυση των απαιτήσεων, ο σχεδιασμός της αρχιτεκτονικής του συστήματος και η υλοποίηση της εφαρμογής, ενώ εξετάστηκαν η λειτουργικότητα και η εμπειρία του χρήστη. Τα ευρήματα έδειξαν ότι η επαυξημένη πραγματικότητα και η αλληλεπίδραση με χειρονομίες μπορούν να χρησιμοποιηθούν για τη διευκόλυνση της εμπειρικής μάθησης και μπορούν να χρησιμοποιηθούν ως συμπληρωματικό εργαλείο στη διδασκαλία της χημείας.

USE OF AUGMENTED REALITY AND HAND GESTURE RECOGNITION IN SECONDARY SCHOOL CHEMISTRY EXPERIMENTS

Markos Kalemis

Abstract

This thesis concerns the design and creation of an augmented reality educational application that uses gesture recognition to teach second-year chemistry courses. The goal of the application is to create an interactive and safe educational model that will allow students to engage with experimental procedures that are difficult or dangerous to perform in conventional educational institutions.

The program was created on the Android platform and uses augmented reality technology to visualize three-dimensional objects in physical educational resources, as well as gesture recognition methods to interact with objects in a natural and intuitive way, thus eliminating the need for traditional data entry. The experiment adopted concerns the combustion of chemical elements and the monitoring of the typical color variation of the flame, where the user can receive immediate visual feedback.

During the project, requirements analysis, system architecture design, and application implementation were carried out, while functionality and user experience were examined. The findings showed that augmented reality and gesture interaction can be used to facilitate experiential learning and can be used as a complementary tool in chemistry teaching.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τους φίλους και την οικογένεια μου για την συμπαράσταση τους κατά την διάρκεια εκπόνησης της διπλωματικής εργασίας, καθώς και τον καθηγητή μου κ. Κεραμόπουλο Ευκλείδη για την ευκαιρία και την βοήθεια του στην εργασία αυτή.

Περιεχόμενα

Πρόλογος.....	iv
Περίληψη.....	v
Abstract.....	vi
Ευχαριστίες.....	vii
Περιεχόμενα.....	viii
Κατάλογος Σχημάτων.....	xi
Κατάλογος Πινάκων.....	xiii
Συντομογραφίες.....	xiv
1 Εισαγωγή.....	1
1.1 Τεχνολογικές εξελίξεις και σύγχρονα διαδραστικά συστήματα.....	1
1.2 Εκπαιδευτικά πειράματα και τεχνολογικοί περιορισμοί.....	1
1.3 Ο ρόλος των νέων τεχνολογιών στην διδασκαλία πειραμάτων.....	2
1.4 Επαυξημένη Πραγματικότητα και αναγνώριση χειρονομιών.....	2
1.5 Σκοπός και αντικείμενο της διπλωματικής εργασίας.....	3
1.6 Στόχοι της εφαρμογής.....	3
1.7 Δομή της διπλωματικής εργασίας.....	3
2 Θεωρητικό Πλαίσιο.....	5
2.1 Επαυξημένη Πραγματικότητα.....	5
2.1.1 Ορισμός και βασικές έννοιες.....	5
2.1.2 Ιστορική εξέλιξη της Επαυξημένης Πραγματικότητας.....	5
2.1.3 Βασικά χαρακτηριστικά και λειτουργικά στοιχεία της Ε.Π.....	6
2.1.4 Τύποι Επαυξημένης Πραγματικότητας.....	7
2.1.5 Σύγκριση Επαυξημένης Πραγματικότητας με VR και MR.....	9
2.1.6 Ρόλος της Επαυξημένης Πραγματικότητας στα διαδραστικά συστήματα.....	13
2.2 Τεχνολογίες Υλοποίησης Συστημάτων Επαυξημένης Πραγματικότητας.....	15
2.2.1 Εισαγωγή στις τεχνολογίες υλοποίησης AR.....	15
2.2.2 Υλικό (Hardware) συστημάτων Επαυξημένης Πραγματικότητας.....	15
2.2.3 Λογισμικό και Πλατφόρμες Ανάπτυξης Συστημάτων Επαυξημένης Πραγματικότητας.....	17
2.2.4 Υπολογιστική Όραση (Computer Vision) στα συστήματα AR.....	20
2.2.5 Παρακολούθηση και Χωρική Εγγραφή.....	21
2.2.6 Απόδοση γραφικών και απόδοση σε πραγματικό χρόνο.....	21

2.2.7 Τεχνικές προκλήσεις και περιορισμοί.....	21
2.3 Αναγνώριση Χειρονομιών.....	22
2.3.1 Ορισμός και σημασία της αναγνώρισης χειρονομιών.....	22
2.3.2 Κατηγορίες χειρονομιών.....	22
2.3.3 Προσεγγίσεις αναγνώρισης χειρονομιών.....	22
2.3.4 Αλγοριθμικές τεχνικές αναγνώρισης χειρονομιών.....	25
2.3.5 Αναγνώριση χειρονομιών σε περιβάλλοντα Επαυξημένης Πραγματικότητας.....	25
2.3.6 Προκλήσεις και περιορισμοί.....	25
2.3.7 Σχέση με την παρούσα διπλωματική εργασία.....	25
2.4 Ανθρώπινη–Υπολογιστική Αλληλεπίδραση.....	26
2.4.1 Ορισμός και αντικείμενο της Ανθρώπινης–Υπολογιστικής Αλληλεπίδρασης.....	26
2.4.2 Εξέλιξη διεπαφών χρήστη.....	26
2.4.3 Φυσικές Διεπαφές Χρήστη.....	26
2.4.4 Ενσώματη Αλληλεπίδραση (Embodied Interaction).....	27
2.4.5 Εμπειρία Χρήστη.....	27
2.4.6 HCI και Εκπαιδευτικά Διαδραστικά Συστήματα.....	27
2.4.7 Σχέση της HCI με την παρούσα διπλωματική εργασία.....	27
2.5 Επαυξημένη Πραγματικότητα στην Εκπαίδευση.....	28
2.5.1 Εισαγωγή στην αξιοποίηση της Επαυξημένης Πραγματικότητας στην εκπαίδευση.....	28
2.5.2 Παιδαγωγικές θεωρίες και Επαυξημένη Πραγματικότητα.....	28
2.5.3 Πλεονεκτήματα της Επαυξημένης Πραγματικότητας στην εκπαίδευση.....	28
2.5.4 Περιορισμοί και προκλήσεις της Ε.Π. στην εκπαίδευση.....	29
2.5.5 Επαυξημένη Πραγματικότητα στη διδασκαλία των Φυσικών Επιστημών.....	29
2.5.6 Επαυξημένη Πραγματικότητα και πειράματα Χημείας στη δευτεροβάθμια εκπαίδευση.....	29
2.6 Επίλογος.....	30
3 Σχεδίαση και Αρχιτεκτονική Συστήματος.....	31
3.1 Εισαγωγή.....	31
3.2 Περιγραφή συνολικού συστήματος.....	31
3.3 Σχεδιασμός Αλληλεπίδρασης Χρήστη – Συστήματος.....	31
3.3.1 Τυπική σειρά αλληλεπίδρασης.....	32
3.3.2 Εμπειρία χρήστη (User Experience).....	32
3.4 Λειτουργικές και μη Λειτουργικές Προδιαγραφές.....	32
3.4.1 Λειτουργικές Προδιαγραφές.....	32
3.4.2 Μη Λειτουργικές Προδιαγραφές.....	33

3.5 Τεχνικές προδιαγραφές.....	33
3.5.1 Βασικές Προδιαγραφές.....	33
3.5.2 Λογισμικά και βιβλιοθήκες.....	33
3.5.3 Περιορισμοί.....	33
3.6 Επίλογος.....	34
4 Ανάλυση και Επεξήγηση της Εφαρμογής.....	35
4.1 Εισαγωγή.....	35
4.2 Περιβάλλον Ανάπτυξης.....	35
4.2.1 Χρήση και ρύθμιση του Unity για την εφαρμογή.....	35
4.3 Διεπαφή Χρήστη.....	43
4.3.1 Splash Screen Διεπαφή.....	43
4.3.2 Κεντρική Διεπαφή Επαυξημένης Πραγματικότητας.....	49
4.4 Επεξήγηση Κώδικα.....	53
4.4.1 Αρχείο ElementData.cs.....	53
4.4.2 Αρχείο ImageTargetManager.cs.....	54
4.4.3 Αρχείο HandTrackingManager.cs.....	62
4.4.4 Αρχείο HandInteractionController.cs.....	69
4.4.5 Αρχείο ElementObject.cs.....	74
4.4.6 Αρχείο FlameController.cs.....	78
4.4.7 Αρχείο GameManager.cs.....	82
4.4.8 Αρχείο UIManager.cs.....	86
4.4.9 Αρχείο SplashScreenManager.cs.....	92
5 Συμπεράσματα και Μελλοντικές Βελτιώσεις.....	95
5.1 Συμπεράσματα.....	95
5.2 Μελλοντικές Βελτιώσεις και Επεκτάσεις.....	95
BIBΛΙΟΓΡΑΦΙΑ.....	97

Κατάλογος Σχημάτων

Εικόνα 2.1 “Sword of Damocles”.....	6
Εικόνα 2.2 Marker-Based AR.....	7
Εικόνα 2.3 Markerless AR.....	8
Εικόνα 2.4 Spacial AR.....	8
Εικόνα 2.5 Wearable AR – Microsoft HoloLens.....	9
Εικόνα 2.6 Εικονική Πραγματικότητα.....	10
Εικόνα 2.7 Επαυξημένη Πραγματικότητα στην μάθηση.....	11
Εικόνα 2.8 Apple Vision Pro MR Device.....	12
Εικόνα 4.1 Εγκατάσταση Editor.....	36
Εικόνα 4.2 Προσθήκη Προεκτάσεων.....	36
Εικόνα 4.3 Εγκατάσταση Προεκτάσεων.....	37
Εικόνα 4.4 Δημιουργία Εφαρμογής.....	38
Εικόνα 4.5 Επιλογή Template και Αρχικοποίηση Εφαρμογής.....	38
Εικόνα 4.6 File > Build Settings.....	39
Εικόνα 4.7 Επιλογή Android ως Build Setting.....	39
Εικόνα 4.8 Επιλογή ελαχίστου API, στόχου API και Scripting Backend IL2CPP.....	40
Εικόνα 4.9 Επιλογή active input handling.....	40
Εικόνα 4.10 Window > Package Manager.....	41
Εικόνα 4.11 AR Foundation, Google ARCore XR Plugin, TextMeshPro.....	42
Εικόνα 4.12 XR Plugin Management και XR Interaction Toolkit.....	42
Εικόνα 4.13 ManoMotion SDK.....	43
Εικόνα 4.14 Δημιουργία Σκηνής.....	45
Εικόνα 4.15 Ιεραρχία Σκηνής SplashScene.....	46
Εικόνα 4.16 Ιεραρχία Σκηνής MainARScene.....	49
Εικόνα 4.17 Κώδικας ElementData.cs.....	53
Εικόνα 4.18 Παράδειγμα στοιχείου.....	54
Εικόνα 4.19 ImageTargetManager.cs.....	55
Εικόνα 4.20 Μέθοδοι Awake(), OnEnable()/OnDisable(), Start().....	56
Εικόνα 4.21 Μέθοδος OnTrackedImagesChanged(ARTrackedImagesChangedEventArgs args).....	57
Εικόνα 4.22 Μέθοδος OnImageFound(ARTrackedImage img).....	58

Εικόνα 4.23 Μέθοδος OnImageUpdated(ARTrackedImage img).....	58
Εικόνα 4.24 Μέθοδος OnImageLost(ARTrackedImage img).....	59
Εικόνα 4.25 Μέθοδος SpawnBunsenBurner(ARTrackedImage img).....	60
Εικόνα 4.26 Μέθοδος SpawnElements(ARTrackedImage img).....	61
Εικόνα 4.27 Μέθοδος RespawnElements().....	61
Εικόνα 4.28 HandTrackingManager.cs.....	62
Εικόνα 4.29 Μέθοδοι Awake(), Start(), Update().....	63
Εικόνα 4.30 Μέθοδος ReadManoMotionData().....	64
Εικόνα 4.31 Μέθοδος HandleNoHand().....	65
Εικόνα 4.32 Μέθοδος UpdateHandPosition(...).....	66
Εικόνα 4.33 Μέθοδοι NormalizedToScreen(Vector2 normalized), CalculateWorldPosition(Vector2 screenPos, float depth).....	67
Εικόνα 4.34 Μέθοδος ProcessGestures(...).....	68
Εικόνα 4.35 Μέθοδοι IsPinching(), GetGrabPointWorldSpace(...), GetGrabPointScreenSpace().....	69
Εικόνα 4.36 HandInteractionController.cs.....	70
Εικόνα 4.37 Μέθοδος Start().....	71
Εικόνα 4.38 Μέθοδοι RefreshElements(), Update().....	72
Εικόνα 4.39 Μέθοδοι FindNearestElement(...), HandleGrabLogic(), TryGrab().....	73
Εικόνα 4.40 Μέθοδοι Release(), CalculateThrowVelocity(), ForceRelease().....	74
Εικόνα 4.41 ElementObject.cs.....	75
Εικόνα 4.42 Μέθοδοι Awake(), Start(), Update().....	76
Εικόνα 4.43 Μέθοδοι OnGrab(), UpdateGrabbedPosition(), OnRelease(), OnTriggerEnter().....	77
Εικόνα 4.44 Μέθοδοι OnCollisionEnter(), StartRespawn(), DoRespawn(), ForeRespawn().....	78
Εικόνα 4.45 FlameController.cs.....	79
Εικόνα 4.46 Μέθοδοι Awake()/Start().....	79
Εικόνα 4.47 Μέθοδοι Update(), UpdateColors(), UpdateFlicker().....	80
Εικόνα 4.48 Μέθοδοι ChangeFlameColor(), ChangeFlameColor().....	81
Εικόνα 4.49 Μέθοδοι BoostLight(), RevertToDefaultColor(), ResetFlame().....	82
Εικόνα 4.50 GameManager.cs.....	83
Εικόνα 4.51 Μέθοδοι Awake(), Start().....	84
Εικόνα 4.52 Μέθοδοι FindReferences(), OnBookDetected(), OnBookLost().....	85
Εικόνα 4.53 Μέθοδοι OnElementThrown(), ResetGameState(), RestartExperiment().....	86
Εικόνα 4.54 UIManager.cs.....	87

Εικόνα 4.55 Μέθοδοι ShowScanPrompt()/HideScanPrompt().....	88
Εικόνα 4.56 Μέθοδοι ShowElementInfo(ElementDatadata)/HideElementInfo().....	89
Εικόνα 4.57 Μέθοδοι UpdateHandTrackingStatus()/SetHandStatus(...).....	90
Εικόνα 4.58 Μέθοδοι StartTutorial()/NextTutorialStep()/SkipTutorial()/CompleteTutorial().....	91
Εικόνα 4.59 Μέθοδος ShowMessage(stringmsg,floatduration).....	92
Εικόνα 4.60 SplashScreenManager.cs.....	93
Εικόνα 4.61 Μέθοδοι SplashSequence(), FadeIn()/FadeOut(), SimulateLoading().....	94

Κατάλογος Πινάκων

Πίνακας 2.1 Σύγκριση των AR, VR, MR.....	13
Πίνακας 2.2 Σύγκριση των SDK επαυξημένης πραγματικότητας.....	19
Πίνακας 2.3 Συγκριτική αξιολόγηση frameworks αναγνώρισης χειρονομιών.....	24

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
Ε.Π.	Επαυξημένη Πραγματικότητα
HDM	Head Mounted Display
VR	Εικονική Πραγματικότητα
MR	Μεικτή Πραγματικότητα
AR	Επαυξημένη Πραγματικότητα

1 Εισαγωγή

1.1 Τεχνολογικές εξελίξεις και σύγχρονα διαδραστικά συστήματα

Με την μεγάλη ανάπτυξη των τελευταίων ετών, στους τομείς της πληροφορικής και της ηλεκτρονικής, έχει φτιαχτεί ένα περιβάλλον που μας δίνει την δυνατότητα να αναπτύξουμε νέους τρόπους επικοινωνίας μεταξύ ανθρώπου και μηχανής εκτός από τις παλιές μεθόδους. Με τεχνολογίες όπως αυτή της επαυξημένης πραγματικότητας (AR) και της αναγνώρισης χειρονομιών μπορούμε να δημιουργήσουμε εφαρμογές και συστήματα που αλληλεπιδρούν με τον χρήστη μ' έναν πιο φυσικό, άμεσο, διαισθητικό και αρμονικό τρόπο.

Τέτοια συστήματα μπορούν να εφαρμοστούν για μια πληθώρα από χρήσεις, όπως η ιατρική, η βιομηχανία, και η εκπαίδευση. Πιο συγκεκριμένα στον τομέα της εκπαίδευσης, οι νέες αυτές τεχνολογίες, μας δίνουν την δυνατότητα να δημιουργήσουμε για τους μαθητές πιο διαδραστικά περιβάλλοντα μάθησης, όπου ο μαθητής δεν θα παρακολουθεί μόνο αλλά θα συμμετέχει ενεργά στην μαθησιακή διαδικασία.

Για να αξιοποιηθούν αποτελεσματικά αυτές οι τεχνολογίες χρειάζεται μια σύνθεση γνώσεων από πολλαπλά επιστημονικά πεδία, με κύρια να είναι αυτά της πληροφορικής και της ηλεκτρονικής. Επίσης χρειάζεται μεγάλη έμφαση στην σχεδίαση διεπαφών για τους χρήστες και πάνω από όλα η κατανόηση της παιδαγωγικής επιστήμης.

1.2 Εκπαιδευτικά πειράματα και τεχνολογικοί περιορισμοί

Η χρήση πειραμάτων για την εκμάθηση των φυσικών επιστημών στους μαθητές είναι εξαιρετικά σημαντική, διότι τους βοηθά να ερευνούν φαινόμενα και να κατανοούν πως τα θεωρητικά μοντέλα ανταποκρίνονται στην πράξη με αποτέλεσμα να καλλιεργούν την επιστημονική τους σκέψη. Μέσα από πειράματα, οι μαθητές βλέπουν τις φυσικές διεργασίες σε πραγματικό χρόνο και αποκτούν εμπειρία η οποία προστίθεται συμπληρωματικά στην θεωρητική διδασκαλία.

Παρόλα αυτά, η διεξαγωγή πειραμάτων στα σχολεία αντιμετωπίζει διάφορα εμπόδια, τα οποία είναι σε τομείς όπως οι ελλείψεις υποδομών που στα περισσότερα σχολεία αποτελεί πραγματικότητα. Δεν διατίθεται ο απαραίτητος εξοπλισμός και υλικό στα σχολεία για να μπορέσουν να πραγματοποιήσουν οι μαθητές πειράματα, και συχνά το οικονομικό κόστος για την παροχή και συντήρηση εργαστηρίων είναι ο κύριος παράγοντας για το φαινόμενο αυτό.

Ακόμα πιο πολύ στην χημεία, οι περιορισμοί εντείνονται εξαιτίας των μεγαλύτερων απαιτήσεων που υπάρχουν για την ασφάλεια των μαθητών. Επειδή τα πειράματα αυτά συχνά χρησιμοποιούν επικίνδυνες χημικές ουσίες, υψηλές θερμοκρασίες και συχνά παράγουν αέρια και άλλα βλαβερά στοιχεία, πολλές πειραματικές διαδικασίες είτε απλοποιούνται σε μεγάλο βαθμό ή δεν πραγματοποιούνται καθόλου.

Παράλληλα, άλλος ένας λόγος που περιορίζονται οι διεξαγωγές των πειραμάτων είναι η ίδια η φύση των χημικών φαινομένων. Η πλειοψηφία των χημικών διεργασιών και αντιδράσεων γίνονται σε μικροσκοπικό επίπεδο που χωρίς τον απαραίτητο εξοπλισμό δεν μπορεί να παρατηρηθεί από τους μαθητές, με αποτέλεσμα

να μην μπορούν να καταλάβουν τις συμπεριφορές των ατόμων, των μορίων και των ιόντων και έτσι να δημιουργούνται στους μαθητές απορίες και λανθασμένες αντιλήψεις για τον κόσμο της χημείας.

Όλα τα παραπάνω δείχνουν την άμεση ανάγκη που υπάρχει για τεχνολογικές λύσεις ικανές να συμπληρώσουν ή να αντικαταστήσουν την κλασσική πειραματική διδασκαλία, με έναν τρόπο ασφαλή, που να μπορεί να επαναλαμβάνεται, ενώ προσφέρει οπτικοποίηση των αφηρημένων εννοιών της χημείας.

1.3 Ο ρόλος των νέων τεχνολογιών στην διδασκαλία πειραμάτων

Με τις τελευταίες εξελίξεις στην πληροφορική, μπορούμε να αλλάξουμε την διδασκαλία των πειραμάτων με τέτοιο τρόπο ώστε να οι μαθητές να συμμετέχουν ενεργά στο μάθημα, παρά να βιώνουν μια παθητική διαδικασία επίδειξης. Με την χρήση της επαυξημένης πραγματικότητας και των ψηφιακών προσομοιώσεων δίνεται στον μαθητή η δυνατότητα να αλληλεπιδρά με πειραματικά σενάρια, να αλλάζει κάθε πιθανή παράμετρο του πειράματος και να παρατηρεί άμεσα τα αποτελέσματα από το πείραμα.

Με την αξιοποίηση των καινοτόμων τεχνολογιών όπως την αναγνώριση χειρονομιών, μπορούμε να ξεπεράσουμε τα εμπόδια της παραδοσιακής πειραματικής διδασκαλίας. Με την προσομοίωση των πειραμάτων, επιτυγχάνετε η ασφαλή εκτέλεση τους καθώς δεν υπάρχουν φυσικοί κίνδυνοι και η επανάληψη τους ώστε οι μαθητές να εξερευνήσουν κάθε πιθανό σενάριο του πειράματος, κάτι που δύσκολα συμβαίνει σε φυσικό εργαστήριο.

Επιπρόσθετα, τα πειράματα μπορούν να παρουσιαστούν σε πολλαπλά επίπεδα αναπαράστασης κάτι που γίνεται εφικτό μόνο με ψηφιακές τεχνολογίες. Δίνεται, δηλαδή στον μαθητή η δυνατότητα να παρατηρεί στον ίδιο χρόνο και το μακροσκοπικό αποτέλεσμα του πειράματος, αλλά και τις μικροσκοπικές διεργασίες που λαμβάνουν μέρος για να επιτευχθεί το μακροσκοπικό κομμάτι. Η παροχή τέτοιας πληροφορίας είναι ιδιαίτερα σημαντική στην χημεία, διότι βοηθά τους μαθητές να κατανοήσουν την σύνδεση μεταξύ των διαφορετικών επιπέδων πληροφορίας του μαθήματος.

Στο σχολικό περιβάλλον, η χρήση των διαδραστικών τεχνολογιών μπορεί να λειτουργήσει ως ένα επιπλέον και συμπληρωματικό εργαλείο για τους εκπαιδευτικούς, ώστε να ενισχύσει την μαθητική διαδικασία, χωρίς να αντικαταστήσει τις κλασσικές μεθόδους μάθησης.

1.4 Επαυξημένη Πραγματικότητα και αναγνώριση χειρονομιών

Με την επαυξημένη πραγματικότητα ο πραγματικός κόσμος μπορεί να ενισχυθεί με την πρόσθεση ψηφιακών αντικειμένων, την επίδειξη πληροφορίας και μοντέλων σε πραγματικό χρόνο. Συνδυάζοντας την αναγνώριση χειρονομιών, ο μαθητής μπορεί να αλληλεπιδράσει με το ψηφιακό περιεχόμενο, μόνο με τις φυσικές κινήσεις των χεριών του, αγνοώντας την ανάγκη για παραδοσιακές συσκευές εισόδου.

Η χρήση τεχνικών υπολογιστικής όρασης και επεξεργασίας εικόνας απαρτίζουν την βάση της αναγνώρισης χειρονομιών, η οποία επιτρέπει την εύρεση και ανάλυση της κίνησης των χεριών. Χρησιμοποιώντας αυτή την μορφή για αλληλεπίδραση, δίνει στον μαθητή μια πιο πλούσια εμπειρία χρήσης και κάνει το σύστημα γενικότερα πιο προσβάσιμο και διαισθητικό.

Ο συνδυασμός της επαυξημένης πραγματικότητας με την αναγνώριση χειρονομιών παράγει διαδραστικές εφαρμογές, με μεγάλο βάθος, που μπορούν να λειτουργήσουν για την προσομοίωση πειραμάτων και την

Κεφάλαιο 1

χειραγώγηση τρισδιάστατων μοντέλων στην προσπάθεια να κατανοήσουν οι μαθητές τις διεργασίες που λαμβάνουν χώρα στα εκάστοτε πειράματα.

1.5 Σκοπός και αντικείμενο της διπλωματικής εργασίας

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η επεξήγηση των βασικών εννοιών όλων των τεχνολογιών που χρησιμοποιήθηκαν στον σχεδιασμό, την ανάπτυξη και την αξιολόγηση της εφαρμογής με το πείραμα καύσης διαφόρων στοιχείων για την παρατήρηση αλλαγής του χρώματος της κεντρικής φλόγας.

Η διπλωματική εργασία έχει αναπτυξιακό και εφαρμοσμένο χαρακτήρα, καθώς περιλαμβάνει:

- ανάλυση απαιτήσεων,
- σχεδιασμό αρχιτεκτονικής συστήματος,
- υλοποίηση λογισμικού,
- πειραματική αξιολόγηση της λειτουργικότητας και της χρησιμότητας της εφαρμογής.

Το πείραμα χημείας λειτουργεί ως το πεδίο εφαρμογής των τεχνολογιών, με το κύριο αντικείμενο της διπλωματικής εργασίας να εντοπίζεται στις τεχνολογίες από τα πεδία της πληροφορικής και της ηλεκτρονικής που χρησιμοποιούνται στην ανάπτυξη διαδραστικών συστημάτων.

1.6 Στόχοι της εφαρμογής

Η ανάπτυξη της εφαρμογής για την προσομοίωση καύσεων διαφόρων στοιχείων για την παρατήρηση του χρώματος που παράγουν συνοδεύεται από την διατύπωση ερωτημάτων τα οποία, καθοδηγούν τον σχεδιασμό αλλά και την αξιολόγηση της εφαρμογής.

Τα βασικά ερωτήματα αφορούν:

- τη δυνατότητα υλοποίησης ενός λειτουργικού συστήματος AR με χειρονομιακή αλληλεπίδραση,
- τον βαθμό ευχρηστίας και απόκρισης του συστήματος,
- την εμπειρία χρήστη που προκύπτει από τη χρήση φυσικών τρόπων αλληλεπίδρασης,
- τη χρησιμότητα της εφαρμογής ως συμπληρωματικού εργαλείου πειραματικής παρουσίασης.

Οι στόχοι της εργασίας είναι τόσο τεχνικοί όσο και εφαρμοσμένοι και εστιάζουν στη δημιουργία ενός συστήματος που είναι τεχνολογικά ορθό, λειτουργικό και επεκτάσιμο.

1.7 Δομή της διπλωματικής εργασίας

Η παρούσα διατριβή αποτελείται από πέντε κεφάλαια. Το πρώτο κεφάλαιο εξηγεί το εισαγωγικό πλαίσιο, το σκοπό και τους στόχους της μελέτης και υπογραμμίζει τη σημασία της χρήσης σύγχρονων τεχνολογιών στη διδασκαλία της χημείας. Το δεύτερο κεφάλαιο καθιερώνει το θεωρητικό πλαίσιο, περιγράφοντας τις βασικές έννοιες της επαυξημένης πραγματικότητας, τις τεχνολογίες που χρησιμοποιούνται στην εφαρμογή της και τις μεθόδους αναγνώρισης χειρονομιών που μπορούν να είναι σχετικές με την παρούσα εργασία. Το τρίτο κεφάλαιο είναι αφιερωμένο στο σχεδιασμό και την αρχιτεκτονική του συστήματος, περιγράφοντας τη δομική οργάνωση της εφαρμογής και τα υποσυστήματα από τα οποία αποτελείται, καθώς και τις λειτουργικές και μη λειτουργικές προδιαγραφές. Το τέταρτο κεφάλαιο παρέχει μια βήμα προς βήμα περιγραφή της υλοποίησης της εφαρμογής με βάση τη λογική του πειράματος, την αλληλεπίδραση μεταξύ

του χρήστη και του συστήματος και τις λεπτομέρειες του πηγαίου κώδικα. Τέλος, το πέμπτο κεφάλαιο παρουσιάζει τα συμπεράσματα της μελέτης και τις προοπτικές για τη βελτίωση και επέκταση της εφαρμογής.

2 Θεωρητικό Πλαίσιο

2.1 Επαυξημένη Πραγματικότητα

2.1.1 Ορισμός και βασικές έννοιες

Με τον όρο επαυξημένη πραγματικότητα (AR) αναφερόμαστε στην τεχνολογία που γεφυρώνει τον πραγματικό κόσμο με τα ψηφιακά αντικείμενα, τις πληροφορίες και τις αναπαραστάσεις που προβάλλονται πάνω στο φυσικό περιβάλλον του χρήστη σε ζωντανό χρόνο. Αυτή είναι η κύρια διαφορά ενός συστήματος επαυξημένης πραγματικότητας με ένα εικονικό σύστημα που μεταφέρει τον χρήστη σε ένα πλήρως εικονικό περιβάλλον. Η Ε.Π. έχει τον αληθινό κόσμο ως βάση και ουσιαστικά τον γεμίζει με ψηφιακά αντικείμενα[1].

Σύμφωνα με τον ορισμό του Azuma[1], ένα σύστημα επαυξημένης πραγματικότητας πρέπει οπωσδήποτε να πληροί τρεις βασικές προϋποθέσεις:

- να συνδυάζει πραγματικά και εικονικά αντικείμενα,
- να λειτουργεί σε πραγματικό χρόνο,
- να είναι εγγεγραμμένο στον τρισδιάστατο χώρο.

Αυτές οι συνθήκες κάνουν την επαυξημένη πραγματικότητα μια ιδιαίτερα απαιτητική τεχνολογία, η οποία απαιτεί τον συγχρονισμό αισθητήρων, την επεξεργασία πληροφοριών σε πραγματικό χρόνο και την ευθυγράμμιση ψηφιακού και φυσικού περιεχομένου.

Η γενική αρχή της επαυξημένης πραγματικότητας (AR) δεν είναι μόνο η παρουσίαση πληροφοριών, αλλά η προσπάθεια ανάπτυξης ενός ενιαίου αντιληπτικού χώρου όπου οι χρήστες ενσωματώνουν τα εικονικά στοιχεία ως αναπόσπαστο μέρος του πραγματικού. Βάσει αυτής της αρχής υποστηρίζονται διάφορες εφαρμογές που βασίζονται στην ανάγκη φυσικής αλληλεπίδρασης, όπως εκπαιδευτικές προσομοιώσεις, βιομηχανική υποστήριξη και συστήματα διεπαφής ανθρώπου-υπολογιστή[2].

2.1.2 Ιστορική εξέλιξη της Επαυξημένης Πραγματικότητας

Οι πρώτες γραπτές αναφορές για τεχνολογικές καινοτομίες που θα μπορούσαν να θεωρηθούν ως πρόδρομοι της έννοιας της επαυξημένης πραγματικότητας έχουν τις ρίζες τους στη δεκαετία του 1960, όταν εξερευνήθηκε η τεχνολογία των οθονών που τοποθετούνται στο κεφάλι (HMD). Μία από τις πρώτες εφαρμογές ήταν το Sword of Damocles (1968) του Sutherland, το οποίο επέτρεπε την επικάλυψη πρωτόγονων τρισδιάστατων εικόνων στο οπτικό πεδίο του χρήστη[19].



Εικόνα 2.1 “Sword of Damocles”

Ο όρος «Επαυξημένη Πραγματικότητα» επινοήθηκε το 1992 από τους Caudell και Mizell, κατά τη συζήτηση για τη χρήση συστημάτων υποστήριξης συναρμολόγησης στην αεροδιαστημική βιομηχανία. Μετά από αυτή την πρώτη εφαρμογή, ο κλάδος της Επαυξημένης Πραγματικότητας έχει εξελιχθεί σε έναν όλο και πιο εξειδικευμένο τομέα αυτόνομης ακαδημαϊκής έρευνας, κυρίως χάρη στη σημαντική ανάπτυξη των δυνατοτήτων επεξεργασίας των υπολογιστών, των τεχνολογιών αισθητήρων και καμερών, των μοντέλων υπολογιστικής όρασης[2].

Πριν από τη δεκαετία του 2000, οι μόνες περιοχές όπου μπορούσε να χρησιμοποιηθεί η επαυξημένη πραγματικότητα ήταν τα ερευνητικά εργαστήρια και εξαιρετικά εξειδικευμένες βιομηχανικές χρήσεις. Αυτό οφειλόταν κυρίως στο υψηλό κόστος του απαραίτητου υλικού και στην έλλειψη φορητότητας των σύγχρονων συστημάτων. Η μαζική παραγωγή έξυπνων κινητών συσκευών και gadget μετά το 2010 λειτούργησε ως καθοριστική ώθηση για τη δημοκρατικοποίηση της επαυξημένης πραγματικότητας, καθώς τα smartphone και τα tablet διαθέτουν ενσωματωμένη κάμερα, αισθητήρες κίνησης και επαρκείς μονάδες επεξεργασίας.

Σήμερα, η Επαυξημένη Πραγματικότητα θεωρείται ώριμη τεχνολογία, με εφαρμογές στην εκπαίδευση, τη βιομηχανία, την ιατρική, τη συντήρηση εξοπλισμού, τη ναυσιπλοΐα και την ψυχαγωγία[3].

2.1.3 Βασικά χαρακτηριστικά και λειτουργικά στοιχεία της Ε.Π.

Τα συστήματα επαυξημένης πραγματικότητας απαρτίζονται από κάποια βασικά κομμάτια, τα οποία όταν συνδέονται μεταξύ τους μπορούν και παράγουν μια πλήρη εμπειρία επαυξημένης πραγματικότητας. Τα πιο σημαντικά από αυτά είναι:

- Συστήματα αισθητήρων, όπως κάμερες, επιταχυνσιόμετρα, γυροσκόπια και αισθητήρες βάθους, που συλλέγουν δεδομένα από το περιβάλλον,

Κεφάλαιο 2

- Μηχανισμοί ανίχνευσης και παρακολούθησης που επιτρέπουν την παρακολούθηση της θέσης και του προσανατολισμού συσκευών ή αντικειμένων σε έναν χώρο,
- Μηχανισμοί επεξεργασίας που εξετάζουν τις πληροφορίες που λαμβάνονται από αισθητήρες και προσδιορίζουν την κατάλληλη θέση των εικονικών αντικειμένων,
- Συστήματα που εμφανίζουν στον χρήστη μια συνδυασμένη μορφή αληθινού και ψηφιακού περιεχομένου.

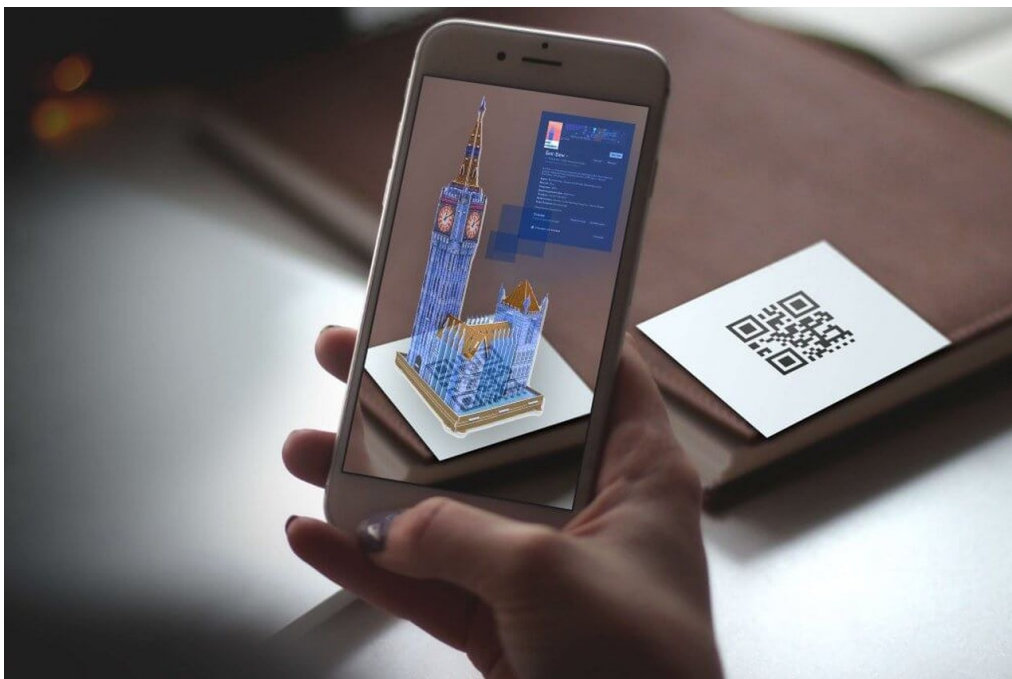
Μία από τις πιο σημαντικές πτυχές της επαυξημένης πραγματικότητας (AR) είναι η λεγόμενη χωρική καταχώριση, δηλαδή η ακριβής τοποθεσία των ψηφιακών αντικειμένων στον πραγματικό κόσμο. Ακόμη και οι παραμικρές ατέλειες στην καταχώριση μπορεί να έχουν σημαντική επίδραση στην αντίληψη του συστήματος ως ρεαλιστικού και στην αποδοχή του από τον χρήστη[1].

2.1.4 Τύποι Επαυξημένης Πραγματικότητας

Η Επαυξημένη Πραγματικότητα μπορεί να κατηγοριοποιηθεί σε διάφορους τύπους, ανάλογα με τον τρόπο ανίχνευσης και αλληλεπίδρασης:

2.1.4.1 Marker Based AR

Αυτή η κατηγορία λειτουργεί με βάση προκαθορισμένους οπτικούς δείκτες, όπως QR κωδικούς ή συγκεκριμένα μοτίβα, για την ενεργοποίηση της Ε.Π. Το σύστημα της κάμερας αναγνωρίζει τον δείκτη και στη συνέχεια τοποθετεί το ψηφιακό αντικείμενο σε ένα σταθερό σημείο. Αν και αυτή η μεθοδολογία είναι απλή και αξιόπιστη, έχει περιορισμένη ευελιξία του συστήματος.



Εικόνα 2.2 Marker-Based AR

2.1.4.2 Markerless AR

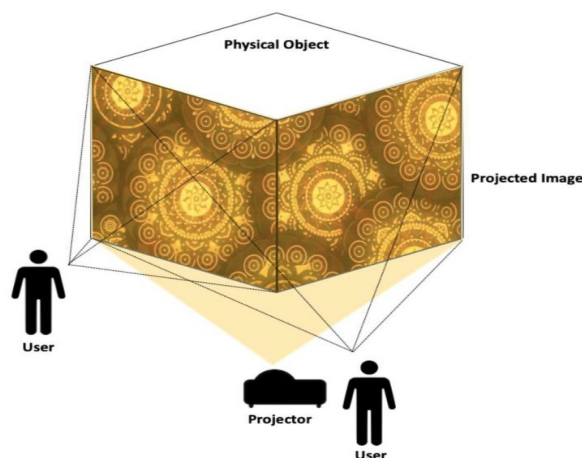
Τα συστήματα επαυξημένης πραγματικότητας χωρίς δείκτες βασίζονται στην υπολογιστική όραση καθώς και σε τεχνικές που βασίζονται σε αισθητήρες για τον προσδιορισμό επιφανειών και αντικειμένων χωρίς τη χρήση προκαθορισμένων δεικτών. Το μοντέλο είναι πιο διαισθητικά διαδραστικό και αυτό το πρότυπο έχει αποκτήσει ευρεία χρήση στις σύγχρονες πλατφόρμες κινητών συσκευών[6].



Εικόνα 2.3 Markerless AR

2.1.4.3 Spatial AR

Η Spatial AR χρησιμοποιεί προβολείς για την απευθείας απεικόνιση ψηφιακού περιεχομένου πάνω σε φυσικά αντικείμενα, χωρίς τη χρήση οθονών. Χρησιμοποιείται κυρίως σε βιομηχανικές και μουσειακές εφαρμογές.



Εικόνα 2.4 Spatial AR

2.1.4.4 Wearable AR

Περιλαμβάνει συσκευές όπως έξυπνα γυαλιά (π.χ. HoloLens), τα οποία προβάλλουν ψηφιακό περιεχόμενο στο οπτικό πεδίο του χρήστη. Παρότι προσφέρουν υψηλό επίπεδο εμπύθισης, παρουσιάζουν αυξημένο κόστος και τεχνική πολυπλοκότητα.



Εικόνα 2.5 Wearable AR – Microsoft HoloLens

2.1.5 Σύγκριση Επαυξημένης Πραγματικότητας με VR και MR

Οι τεχνολογίες εμπύθισης διαθέτουν ένα συνεχές σύστημα που χαρακτηρίζεται από διαφορετικά επίπεδα ενσωμάτωσης ψηφιακού περιεχομένου στον πραγματικό κόσμο και απομόνωση ή εμπύθιση του χρήστη από τον φυσικό κόσμο. Στο πλαίσιο αυτό, η επαυξημένη πραγματικότητα (AR), η εικονική πραγματικότητα (VR) και η μικτή πραγματικότητα (MR) είναι διαφορετικές αλλά αλληλένδετες τεχνολογίες, οι οποίες καταλαμβάνουν διαφορετικές θέσεις στο συνεχές της πραγματικότητας και της εικονικότητας, όπως περιγράφεται από τους Milgram και Kishino[7].

2.1.5.1 Το συνεχές πραγματικότητας–εικονικότητας

Το συνεχές μπορεί να κυμαίνεται μεταξύ του εντελώς πραγματικού κόσμου και του εντελώς εικονικού κόσμου, όπως συζητείται στο μοντέλο που δημιούργησαν οι Milgram και Kishino[7]. Ανάμεσά τους, υπάρχουν τεχνολογίες που είναι ένας συνδυασμός πραγματικών και εικονικών στοιχείων σε συγκεκριμένες αναλογίες. Η επαυξημένη πραγματικότητα συνδέεται περισσότερο με τον πραγματικό κόσμο, καθώς διατηρεί τον φυσικό κόσμο ως βασικό πλαίσιο αναφοράς, σε σύγκριση με την εικονική πραγματικότητα, η οποία είναι το άλλο άκρο και αντικαθιστά πλήρως τον πραγματικό κόσμο με έναν ψηφιακό. Η μικτή πραγματικότητα βρίσκεται στη μέση, επιτρέποντας πιο εξελιγμένη αλληλεπίδραση μεταξύ των πραγματικών και των ψηφιακών αντικειμένων.

Αυτή η ταξινόμηση είναι κάτι περισσότερο από εννοιολογική, έχει άμεσες επιπτώσεις στο σχεδιασμό συστημάτων, στις απαιτήσεις υλικού και λογισμικού και στην εμπειρία του χρήστη.

2.1.5.2 Εικονική Πραγματικότητα (Virtual Reality – VR)

Η εικονική πραγματικότητα δημιουργεί ένα πλήρως ψηφιακό περιβάλλον που ο χρήστης λαμβάνει με ειδικό εξοπλισμό, όπως οθόνες που τοποθετούνται στο κεφάλι (HMD). Αυτός ο προσομοιωμένος χώρος αντικαθιστά πλήρως τον φυσικό κόσμο και έτσι προσφέρει ένα ισχυρό επίπεδο εμπύθισης καθώς και μια μεγαλύτερη αίσθηση παρουσίας.



Εικόνα 2.6 Εικονική Πραγματικότητα

Η χρήση της εικονικής πραγματικότητας είναι ιδιαίτερα ωφέλιμη στον τομέα της εκπαίδευσης για επικίνδυνες καταστάσεις, της προσομοίωσης σύνθετων διαδικασιών και του ελεύθερου χρόνου για ψυχαγωγία. Ωστόσο, έχει και ορισμένους ενδιαφέροντες περιορισμούς, καθώς ο απόλυτος διαχωρισμός του χρήστη από τον φυσικό κόσμο μειώνει την εφαρμοσιμότητά της στην εκπαίδευση, όπου απαιτείται αλληλεπίδραση με συγκεκριμένα φυσικά αντικείμενα ή όπου απαιτείται η αλληλεπίδραση πολλών χρηστών σε κοινό χώρο[8].

Επιπλέον, η εικονική πραγματικότητα συχνά σχετίζεται με αυξημένο ψυχικό και φυσιολογικό φορτίο και προκαλεί συμπτώματα ζάλης και χωροταξικού αποπροσανατολισμού, ειδικά σε νεότερους χρήστες. Με τη σειρά τους, αυτές οι πτυχές καθιστούν την εικονική πραγματικότητα λιγότερο κατάλληλη για μακροχρόνια χρήση στο εκπαιδευτικό περιβάλλον.

2.1.5.3 Επαυξημένη Πραγματικότητα

Η Ε.Π. υποστηρίζει τον φυσικό κόσμο ως την πιο κοινή πλατφόρμα δράσης και τον ενισχύει προσεκτικά με ψηφιακά αντικείμενα που προσθέτουν στην αντιληπτική εμπειρία του χρήστη. Αυτό το παράδειγμα ενθαρρύνει τη συνεχή αλληλεπίδραση με το φυσικό περιβάλλον, μειώνοντας έτσι την πιθανότητα αποπροσανατολισμού ή απομόνωσης[1].

Η επαυξημένη πραγματικότητα έχει συγκεκριμένα οφέλη στην εκπαιδευτική επαφή, ειδικά στο παιδαγωγικό πείραμα. Οι μαθητές έχουν τη δυνατότητα να μελετήσουν δύσκολα αντικείμενα και ταυτόχρονα να εξετάσουν ψηφιακά αναπαραστάσεις μικροσκοπικών ή αφηρημένων φαινομένων, όπως χημικές αντιδράσεις και μοριακές δομές[6]. Με τη σειρά του, η επαυξημένη πραγματικότητα λειτουργεί καλύτερα σε περιβάλλοντα όπου η γνωστική λειτουργία βασίζεται στον συνδυασμό μακροσκοπικών και μικροσκοπικών επιπέδων.

Επιπλέον, η χρήση φυσικών μορφών αλληλεπίδρασης, όπως η αναγνώριση χειρονομιών με τη βοήθεια της επαυξημένης πραγματικότητας (AR), μπορεί να χρησιμοποιηθεί για να καταστήσει τη διαδικασία πιο διαισθητική και να ενθαρρύνει τη βιωματική μάθηση. Σε αντίθεση με την εικονική πραγματικότητα (VR), η Ε.Π. μπορεί να εφαρμοστεί σε κοινές τεχνολογίες, όπως smartphone και tablet, καθιστώντας την φθηνότερη και αυξάνοντας ενδεχομένως τις πιθανότητες διάδοσης της τεχνολογίας στο ακαδημαϊκό περιβάλλον.



Εικόνα 2.7 Επαυξημένη Πραγματικότητα στην μάθηση

2.1.5.4 Μεικτή Πραγματικότητα

Η μεικτή πραγματικότητα συνδυάζει τα στοιχεία του πλαισίου της επαυξημένης πραγματικότητας (AR) και της εικονικής πραγματικότητας (VR), διευκολύνοντας έτσι τον αμφίδρομο διάλογο μεταξύ των πραγματικών και των εικονικών αντικειμένων. Τα ψηφιακά αντικείμενα δεν προστίθενται απλώς στο φυσικό περιβάλλον

εντός του συστήματος MR, αλλά είναι σε θέση να αλληλεπιδρούν με πραγματικά αντικείμενα, με αναφορά στη γεωμετρία του περιβάλλοντος και τις φυσικές του ιδιότητες.

Παρόλο που το επίπεδο λειτουργικότητας και εμπύθισης που προσφέρει η μικτή πραγματικότητα (MR) είναι υψηλό, απαιτεί προηγμένο υλικό, όπως ειδικές οθόνες που τοποθετούνται στο κεφάλι (HMD) και αισθητήρες βάθους, γεγονός που αυξάνει περαιτέρω την επένδυση και ανεβάζει τόσο το κόστος όσο και την τεχνολογική πολυπλοκότητα. Επομένως, αυτό σημαίνει ότι η MR παραμένει σε μεγάλο βαθμό ένα φαινόμενο που αφορά τη βιομηχανία και την έρευνα, με ελάχιστη εξάπλωση στον τομέα της εκπαίδευσης.



Εικόνα 2.8 Apple Vision Pro MR Device

2.1.5.5 Συγκριτική αξιολόγηση και επιλογή τεχνολογίας

Η επιλογή μεταξύ επαυξημένης πραγματικότητας (AR), εικονικής πραγματικότητας (VR) και μικτής πραγματικότητας (MR) εξαρτάται από το πεδίο εφαρμογής, το κοινό-στόχο και τους τεχνολογικούς περιορισμούς που υπάρχουν σήμερα. Ο πίνακας 2.1 θα παρέχει μια σύντομη λίστα με τα βασικά χαρακτηριστικά των τριών μεθόδων, όπως το επίπεδο εμπύθισης, το κόστος, τα απαραίτητα έξοδα υποδομής και η καταλληλότητα για χρήση σε εκπαιδευτικό περιβάλλον.

Στην παρούσα διατριβή, η επαυξημένη πραγματικότητα έχει βρεθεί ότι είναι η πιο εφικτή τεχνολογία, καθώς προσφέρει ισορροπία μεταξύ των παραγόντων εμπύθισης, διαθεσιμότητας και ασφάλειας. Αυτός ο συνδυασμός Ε.Π. και λειτουργιών ανίχνευσης χειρονομιών διευκολύνει τη δημιουργία ενός πλαισίου φυσικής αλληλεπίδρασης, το οποίο είναι ιδιαίτερα επωφελές στο πλαίσιο των πειραμάτων και συνάδει με τις σύγχρονες αρχές εργασίας στο σχεδιασμό διαδραστικών συστημάτων[9].

Πίνακας 2.1 Σύγκριση των AR, VR, MR

Χαρακτηριστικό	Επαυξημένη Πραγματικότητα (AR)	Εικονική Πραγματικότητα (VR)	Μεικτή Πραγματικότητα (MR)
Σχέση με πραγματικό κόσμο	Ο πραγματικός κόσμος παραμένει ορατός και εμπλουτίζεται με ψηφιακά στοιχεία	Ο πραγματικός κόσμος αντικαθίσταται πλήρως από εικονικό περιβάλλον	Πραγματικά και ψηφιακά αντικείμενα συνυπάρχουν και αλληλεπιδρούν
Βαθμός εμπύθισης	Μέτριος	Πολύ υψηλός	Υψηλός
Απομόνωση χρήστη	Χαμηλή	Υψηλή	Μέτρια
Αλληλεπίδραση με φυσικά αντικείμενα	Ναι	Όχι	Ναι
Απαιτούμενος εξοπλισμός	Κινητές συσκευές, tablets, κάμερες	HMDs, χειριστήρια, αισθητήρες	Εξειδικευμένα HMDs, αισθητήρες βάθους
Κόστος υλοποίησης	Χαμηλό έως μέτριο	Μέτριο έως υψηλό	Υψηλό
Καταλληλότητα για σχολική εκπαίδευση	Υψηλή	Περιορισμένη	Περιορισμένη
Κίνδυνος αποπροσανατολισμού	Πολύ χαμηλός	Αυξημένος	Μέτριος
Υποστήριξη φυσικής αλληλεπίδρασης (gestures)	Υψηλή	Μέτρια	Υψηλή
Τυπικές εφαρμογές	Εκπαίδευση, συντήρηση, πειράματα	Προσομοιώσεις, εκπαίδευση υψηλού κινδύνου, gaming	Βιομηχανία, έρευνα, ιατρική

2.1.6 Ρόλος της Επαυξημένης Πραγματικότητας στα διαδραστικά συστήματα

Η Ε.Π. είναι μια τεχνολογία απαραίτητη για την ανάπτυξη των διαδραστικών συστημάτων επόμενης γενιάς. Σε αντίθεση με τις παραδοσιακές διεπαφές χρήστη που βασίζονται σε μια αφηρημένη διεπαφή εισόδου, όπως το πληκτρολόγιο και το ποντίκι, οι πλατφόρμες AR επιτρέπουν στους χρήστες να αλληλεπιδρούν φυσικά και άμεσα με τις ψηφιακές πληροφορίες στον φυσικό κόσμο[9].

Στα διαδραστικά συστήματα επαυξημένης πραγματικότητας, οι χωρικές παράμετροι είναι κάτι περισσότερο από φόντο οθόνης, καθώς αποτελούν μέρος της διεπαφής χρήστη. Τα τεχνητά αντικείμενα του ψηφιακού χώρου αποκτούν έτσι υλικές χωρικές δυνατότητες και αφομοιώνονται αποτελεσματικά στον πραγματικό κόσμο που περιβάλλει τον χρήστη, υποστηρίζοντας έτσι τρόπους αλληλεπίδρασης που βασίζονται στην κίνηση, στη χειρονομιακή εισαγωγή δεδομένων και στην εξερευνητική κίνηση στο χώρο. Αυτή η θεωρητική προοπτική μπορεί να αντιστοιχιστεί σαφώς σε μια μεθοδολογική θέση που βασίζεται σε ένα θεωρητικό πλαίσιο ενσώματης αλληλεπίδρασης, το οποίο υποστηρίζει ότι η γνώση και η κατανόηση στους ανθρώπους βασίζονται ουσιαστικά στην σωματική αλληλεπίδραση με το περιβάλλον[9].

Η δυνατότητα χωρικής αναπαράστασης των πληροφοριών είναι ένα από τα κύρια πλεονεκτήματα της επαυξημένης πραγματικότητας (AR) στο πλαίσιο της προσπάθειας χρήσης διαδραστικών συστημάτων. Αντί να παρουσιάζει τα δεδομένα μέσω δισδιάστατων διεπαφών που είναι αφηρημένες και δεν έχουν φυσική αναπαράσταση, η AR συνεπάγεται την εφαρμογή των δεδομένων αυτών στον πραγματικό κόσμο με τρόπο ισότιμο και κατανοητό από την ανθρώπινη αντίληψη. Η προαναφερθείσα χωρική κωδικοποίηση μειώνει το γνωστικό φορτίο και βοηθά στην κατανόηση σύνθετων δομών και διαδικασιών, ιδίως εκείνων που έχουν επιστημονικό και τεχνικό χαρακτήρα[1].

Το σύστημα αλληλεπίδρασης με την επαυξημένη πραγματικότητα βελτιώνεται επίσης μέσω της χρήσης μεθόδων φυσικής διεπαφής χρήστη (NUI), όπως η αναγνώριση χειρονομιών, η εκτίμηση στάσεων του σώματος και η φωνητική εντολή. Η δυνατότητα εισαγωγής χειρονομιών ως τύπος αλληλεπίδρασης δίνει στον χρήστη τη δυνατότητα να αλληλεπιδρά με νόμιμα εικονικά αντικείμενα με τρόπο που σχετίζεται με το ίδιο αντικείμενο στην αλληλεπίδραση του πραγματικού κόσμου, γεγονός που αυξάνει την αντίληψη του ελέγχου καθώς και την διαισθητικότητα των εικονικών αντικειμένων[10].

Η εμπειρία χρήστη (UX) είναι ένας αποφασιστικός παράγοντας για την αξιολόγηση της επιτυχίας ενός συστήματος όσον αφορά τον σχεδιασμό διαδραστικών συστημάτων. Οι ιδιαιτερότητες των συστημάτων επαυξημένης πραγματικότητας (AR) θέτουν πρωτότυπα ερωτήματα και δυνατότητες για τον σχεδιασμό της εμπειρίας χρήστη, απαιτώντας μια λογική ισορροπία μεταξύ του επιπέδου του πληροφοριακού περιεχομένου και του οπτικού φορτίου, της ακρίβειας στην χωρική τοποθέτηση των ψηφιακών αντικειμένων και της ικανότητας να ανταποκρίνεται οργανικά στις ενέργειες του χρήστη. Η κακή εφαρμογή αυτών των αρχών σχεδιασμού μπορεί να προκαλέσει αποσπάσεις της προσοχής του χρήστη ή κόπωση, αλλά όταν γίνεται με προσοχή, μπορεί να αυξήσει σημαντικά την αποδοχή και τη λειτουργικότητα του συστήματος[11].

Τα συστήματα Ε.Π. με διαδραστικές δυνατότητες είναι επίσης σημαντικά στον τομέα της εκπαίδευσης, καθώς η συμμετοχή των μαθητών αποτελεί προϋπόθεση για την επιτυχή μελέτη. Η AR συμβάλλει στη δημιουργία ενός μαθησιακού περιβάλλοντος όπου ο χρήστης δεν αρκείται στο να ακούει τις πληροφορίες, αλλά αλληλεπιδρά με αυτές, πραγματοποιεί πειράματα και προκαλεί αιτίες και αποτελέσματα. Σύμφωνα με έρευνες, αυτά τα περιβάλλοντα αυξάνουν το επίπεδο ενδιαφέροντος για τη μάθηση και την απόκτηση μακροπρόθεσμης γνώσης[12].

Τα διαδραστικά συστήματα επαυξημένης πραγματικότητας είναι ιδιαίτερα χρήσιμα στο πλαίσιο της πειραματικής χρήσης, συμπεριλαμβανομένων των παραδειγμάτων της διδασκαλίας της χημείας. Οι χρήστες μπορούν να δουν τα φυσικά αντικείμενα (π.χ. δοχεία) και ταυτόχρονα να δουν μια προσομοίωση στον υπολογιστή, των διαδικασιών που διαφορετικά δεν θα ήταν ορατές, όπως οι αλληλεπιδράσεις μεταξύ μορίων και η αντίδραση των χημικών ουσιών. Αυτή η πτυχή ενισχύει τη σύνδεση μεταξύ των θεωρητικών δομών και της συγκεκριμένης πειραματικής πρακτικής, επομένως η εμπειρική πρακτική καθιστά τη μαθησιακή εμπειρία πιο συγκεκριμένη και κατανοητή[6].

Κεφάλαιο 2

Συνολικά, η επαυξημένη πραγματικότητα δεν είναι απλώς ένα μέσο οπτικοποίησης, αλλά ένα ολοκληρωμένο παράδειγμα εννοιολόγησης διαδραστικών συστημάτων, των οποίων οι αρχές βασίζονται στην αλληλεπίδραση με τον υλικό κόσμο, τον χώρο και την ικανότητα του χρήστη να συμμετέχει άμεσα. Μια σε βάθος αντίληψη του ρόλου της επαυξημένης πραγματικότητας σε τέτοια συστήματα θα προσφέρει μια σταθερή θεωρητική βάση για το σχεδιασμό και την εκτέλεση της εφαρμογής που περιγράφεται στα επόμενα κεφάλαια αυτής της διατριβής.

2.2 Τεχνολογίες Υλοποίησης Συστημάτων Επαυξημένης Πραγματικότητας

2.2.1 Εισαγωγή στις τεχνολογίες υλοποίησης AR

Η χρήση ενός συστήματος επαυξημένης πραγματικότητας (AR) προϋποθέτει τη συνδυασμένη χρήση τεχνολογιών υλικού και λογισμικού, οι οποίες στο σύνολό τους συλλέγουν δεδομένα για τον πραγματικό κόσμο, τα επεξεργάζονται και εμφανίζουν ψηφιακό περιεχόμενο σε πραγματικό χρόνο. Σε αντίθεση με τους παραδοσιακούς τύπους εφαρμογών λογισμικού, τα συστήματα AR χρησιμοποιούνται σε συνεχώς μεταβαλλόμενα και ανεξέλεγκτα περιβάλλοντα, γεγονός που αυξάνει σημαντικά την πολυπλοκότητα του σχεδιασμού τους[1].

Η ακρίβεια της ανίχνευσης, η χαμηλή καθυστέρηση, η ακριβής χωρική καταγραφή των ψηφιακών αντικειμένων και η φυσική αλληλεπίδραση με τον χρήστη συνδέονται άμεσα με την ποιότητα της εμπειρίας της επαυξημένης πραγματικότητας. Με τη σειρά τους, οι τεχνολογίες υλοποίησης της επαυξημένης πραγματικότητας αποτελούν ένα πεδίο έντονης έρευνας στον τομέα της υπολογιστικής όρασης, των υπολογιστικών γραφικών και της αλληλεπίδρασης ανθρώπου-υπολογιστή.

2.2.2 Υλικό (Hardware) συστημάτων Επαυξημένης Πραγματικότητας

Ο σχεδιασμός του υλικού ενός συστήματος επαυξημένης πραγματικότητας αποτελεί τη βάση υλικού στην οποία βασίζεται η συλλογή δεδομένων από το περιβάλλον, η δευτερεύουσα επεξεργασία των δεδομένων και η παράδοση του επαυξημένου περιεχομένου στον τελικό χρήστη. Η επιλογή των κατάλληλων εξαρτημάτων υλικού έχει άμεσο αντίκτυπο στην ακρίβεια, τη σταθερότητα, τη χρηστικότητα και την ικανότητα του συστήματος να υποστηρίξει τρόπους φυσικής αλληλεπίδρασης, π.χ. αναγνώριση χειρονομιών.

Στα σύγχρονα συστήματα AR, το hardware αποτελείται από τρεις βασικούς άξονες: **συσκευές απεικόνισης, αισθητήρες και υπολογιστικές μονάδες επεξεργασίας**.

2.2.2.1 Συσκευές απεικόνισης

Οι συσκευές απεικόνισης καθορίζουν τον τρόπο με τον οποίο ο χρήστης αντιλαμβάνεται το επαυξημένο περιβάλλον. Οι βασικές κατηγορίες περιλαμβάνουν:

I. Κινητές συσκευές (Smartphones και Tablets)

Οι κινητές συσκευές αποτελούν την πιο κοινή πλατφόρμα μέσω της οποίας υλοποιείται η επαυξημένη πραγματικότητα και αυτό οφείλεται κυρίως στην ευρεία αποδοχή και την εύκολη προσβασιμότητα μιας μυριάδας αισθητήρων. Αυτά τα συστήματα διαθέτουν συστήματα απεικόνισης υψηλής ανάλυσης, μονάδες αδρανειακής μέτρησης (IMU) και έχουν επαρκείς δυνατότητες επεξεργασίας για την αποτελεσματική εκτέλεση κωδικών επεξεργασίας εικόνας σε πραγματικό χρόνο.

Ωστόσο, παρά τα πλεονεκτήματα αυτά, οι κινητές συσκευές υπόκεινται σε εργονομικούς περιορισμούς, καθώς απαιτούν συνεχή χειροκίνητο έλεγχο και, ως εκ τούτου, επιβάλλουν περιορισμούς στο οπτικό πεδίο. Ωστόσο, όταν εφαρμόζονται στον τομέα της εκπαίδευσης, προσφέρουν μια αποδεκτή ισορροπία μεταξύ τιμής, λειτουργικότητας και χρηστικότητας.

II. Head-Mounted Displays (HMDs)

Μια πιο εμβυθιστική τεχνολογία, όπως τα έξυπνα γυαλιά, ονομάζεται οθόνη κεφαλής (head-mounted display, HMD) και περιλαμβάνει την παρουσίαση ψηφιακού περιεχομένου σε προοπτική προβολή στον χρήστη. Ένα εξαιρετικό πλεονέκτημα των HMD είναι ότι μπορούν να αντικαταστήσουν τις χειροκίνητες λειτουργίες, κάτι που είναι ιδιαίτερα σημαντικό σε εφαρμογές που βασίζονται στις αρχές των αλληλεπιδράσεων με χειρονομίες.

Ωστόσο, τα HMD απαιτούν μεγαλύτερη υπολογιστική ισχύ, είναι πιο ακριβά και συχνά άβολα όταν χρησιμοποιούνται για μεγάλο χρονικό διάστημα, περιορίζοντας έτσι την επεκτασιμότητά τους σε εκπαιδευτικές εφαρμογές.

2.2.2.2 Αισθητήρες

Οι αισθητήρες αποτελούν τον πυρήνα της συλλογής δεδομένων σε συστήματα AR. Επιτρέπουν την ανίχνευση του περιβάλλοντος, της κίνησης και της θέσης του χρήστη. Οι βασικοί αισθητήρες που χρησιμοποιούνται σε AR συστήματα είναι οι εξής:

I. Κάμερες

Οι κάμερες RGB μπορούν να χρησιμοποιηθούν για την ανίχνευση των χαρακτηριστικών του περιβάλλοντος, την αναγνώριση αντικειμένων και την παρακολούθηση της κίνησης. Αποτελούν κεντρικό μέρος τόσο των συστημάτων επαυξημένης πραγματικότητας με δείκτες όσο και των συστημάτων χωρίς δείκτες.

Η ποιότητα της κάμερας έχει άμεση επίδραση στην ακρίβεια της παρακολούθησης, ειδικά όταν υπάρχει λιγότερος φωτισμός ή μεγάλη αντανάκλαση.

II. Αισθητήρες βάθους

Οι αισθητήρες βάθους χρησιμοποιούνται για την εκτίμηση της απόστασης μεταξύ αντικειμένων και συσκευών, καθώς και για τη δημιουργία τρισδιάστατων χωρικών χαρτών. Χρησιμοποιούνται επομένως για τη βελτίωση της χωρικής αντίληψης και μπορούν να βοηθήσουν σε εφαρμογές που απαιτούν φυσική αλληλεπίδραση των ψηφιακών αντικειμένων.

Αν και η προσθήκη αισθητήρων βάθους βελτιώνει σημαντικά την ακρίβεια της χωρικής καταγραφής, αυξάνει επίσης το κόστος ολόκληρου του συστήματος και το καθιστά πιο περίπλοκο.

III. Αισθητήρες αδράνειας (IMU)

Τα IMU συνδυάζουν επιταχυνσιόμετρα και γυροσκόπια, παρέχοντας έτσι πλήρη δεδομένα τόσο για τη μεταφορική όσο και για την περιστροφική ταχύτητα του οργάνου. Η χρήση μετρήσεων που προέρχονται από IMU θα αυξήσει την ακρίβεια της παρακολούθησης θέσης και θα μειώσει επίσης την καθυστέρηση χρόνου σε δυναμικά συστήματα.

2.2.2.3 Συνδυασμός αισθητήρων

Η χρήση ενός μόνο αισθητήρα δεν είναι στις περισσότερες περιπτώσεις αρκετή για να εξασφαλίσει την αποτελεσματική λειτουργία ενός συστήματος AR. Ως αποτέλεσμα, χρησιμοποιούνται μέθοδοι συγχώνευσης αισθητήρων, οι οποίες χρησιμοποιούνται για την ενσωμάτωση των σημάτων των καμερών, των αισθητήρων βάθους και των συσκευών αδρανούς μέτρησης και για την αύξηση της ακρίβειας και της σταθερότητας. Αυτές οι μέθοδοι δεν μπορούν να αγνοηθούν σε εφαρμογές όπου είναι απαραίτητη η πολύ ακριβής χωρική καταγραφή και η φυσική αλληλεπίδραση, π.χ. αναγνώριση χειρονομιών σε πραγματικό χρόνο.

2.2.2.4 Υπολογιστικές μονάδες επεξεργασίας

Η επεξεργασία δεδομένων αισθητήρων σε πραγματικό χρόνο και η απόδοση γραφικών απαιτούν την εφαρμογή υπολογιστικών μηχανών υψηλής απόδοσης. Στα συστήματα επαυξημένης πραγματικότητας (AR), ο υπολογισμός πραγματοποιείται συγχρονισμένα με την οθόνη του υπολογιστή ή ενός προσωπικού συστήματος επεξεργασίας.

Οι περιορισμοί σε συγκεκριμένη υπολογιστική ισχύ και κατανάλωση ενέργειας αποτελούν επίσης ένα κεντρικό ζήτημα σχεδιασμού, ειδικά στις συνθήκες των κινητών και εκπαιδευτικών εφαρμογών.

2.2.3 Λογισμικό και Πλατφόρμες Ανάπτυξης Συστημάτων Επαυξημένης Πραγματικότητας

Η δημιουργία συστημάτων επαυξημένης πραγματικότητας (AR) βασίζεται σε ένα προηγμένο οικοσύστημα λογισμικού που περιλαμβάνει μηχανές γραφικών, βιβλιοθήκες υπολογιστικής όρασης, βοηθητικά προγράμματα διαχείρισης αισθητήρων και ιδιότητα kit ανάπτυξης λογισμικού (SDK) της AR. Σε σύγκριση με τις παραδοσιακές εφαρμογές, τα συστήματα AR απαιτούν συνεχή ροή δεδομένων σε πραγματικό χρόνο και τακτικό συγχρονισμό με διάφορα υποσυστήματα, αυξάνοντας έτσι σημαντικά την πολυπλοκότητα της αρχιτεκτονικής του λογισμικού τους[2].

2.2.3.1 Αρχιτεκτονική λογισμικού συστημάτων AR

Η αρχιτεκτονική ενός συστήματος AR μπορεί να περιγραφεί ως μια πολυεπίπεδη δομή, η οποία περιλαμβάνει τα ακόλουθα βασικά επίπεδα:

1. Επίπεδο συλλογής δεδομένων

Περιλαμβάνει τη συλλογή δεδομένων από διάφορους αισθητήρες, όπως κάμερες, αισθητήρες βάθους, συσκευές αδρανειακής μέτρησης (IMU). Σε αυτό το επίπεδο πραγματοποιείται ο συγχρονισμός και η προεπεξεργασία των σημάτων που λαμβάνονται.

2. Επίπεδο αντίληψης

Η αρχιτεκτονική υλοποιείται περαιτέρω με τη χρήση αλγορίθμων υπολογιστικής όρασης για την ανίχνευση χαρακτηριστικών, την εκτίμηση της θέσης της κάμερας και την αντίληψη του περιβάλλοντος. Μερικές από τις λειτουργίες που εκτελούνται σε αυτήν περιλαμβάνουν την παρακολούθηση και τη χαρτογράφηση.

3. Επίπεδο λογικής εφαρμογής

Αυτό το επίπεδο ελέγχει τη συμπεριφορά της εφαρμογής, τη διεπαφή χρήστη και την εφαρμογή των κανόνων του συστήματος. Λειτουργεί ως μεσολαβητής μεταξύ των δεδομένων αντίληψης σε σύγκριση με τις δυνατότητες της εφαρμογής.

4. Επίπεδο απεικόνισης

Το συγκεκριμένο στοιχείο έχει ως αποστολή να παρέχει τα ψηφιακά αντικείμενα και να τους δώσει μια θέση στο πραγματικό περιβάλλον, λαμβάνοντας υπόψη τις συνθήκες φωτισμού, την ακρίβεια της προοπτικής και την απόδοση σε πραγματικό χρόνο.

Η διαχωρισμένη αρχιτεκτονική βελτιώνει τη συντηρησιμότητα και την επεκτασιμότητα των συστημάτων επαυξημένης πραγματικότητας και, ταυτόχρονα, επιτρέπει τη βελτιστοποίηση κάθε επιμέρους επιπέδου.

2.2.3.2 Μηχανές γραφικών

Οι σύγχρονες εφαρμογές της επαυξημένης πραγματικότητας (AR) βασίζονται κυρίως σε προγράμματα γραφικών γενικής χρήσης, σε ένα σύστημα φυσικής προσομοίωσης και στη διαχείριση τρισδιάστατων σκηνών, σε συνδυασμό με καταναεμημένα εργαλεία όλων των renderers. Αυτοί οι γραφικοί κινητήρες λειτουργούν ως ένα χαμηλότερο επίπεδο αφαίρεσης, εξαλείφοντας την πολυπλοκότητα του προγραμματισμού γραφικών χαμηλού επιπέδου, επιτρέποντας έτσι στον προγραμματιστή να επικεντρωθεί στον κώδικα της εφαρμογής και στη διεπαφή χρήστη. Ταυτόχρονα, υποστηρίζουν shaders, pipelines απόδοσης και βελτιστοποίηση σε πραγματικό χρόνο. Η χρήση μηχανών παιχνιδιών σε εφαρμογές AR είναι ιδιαίτερα σημαντική για την υλοποίηση σύνθετων διαδραστικών καταστάσεων, συμπεριλαμβανομένης της προσομοίωσης ενός πειράματος όπου απαιτείται συντονισμός μεταξύ οπτικοποίησης, αλληλεπίδρασης και υπολογισμών υπολογιστή.

2.2.3.3 AR Software Development Kits (SDKs)

Τα κιτ ανάπτυξης λογισμικού AR αποτελούν βασικό συστατικό στην ανάπτυξη εφαρμογών επαυξημένης πραγματικότητας, καθώς παρέχουν έτοιμες υλοποιήσεις που σχετίζονται με βασικές λειτουργίες, όπως:

- ανίχνευση επιφανειών και επιπέδων,
- εκτίμηση θέσης και προσανατολισμού συσκευής,
- αναγνώριση εικόνων και αντικειμένων,
- χωρική χαρτογράφηση (spatial mapping).

Η ενσωμάτωση ενός κιτ ανάπτυξης λογισμικού επαυξημένης πραγματικότητας θα μειώσει σημαντικά τον χρόνο που αφιερώνεται στην ανάπτυξη και θα βελτιώσει την ανθεκτικότητα του συστήματος, καθώς οι αλγόριθμοι στους οποίους βασίζεται έχουν ήδη δοκιμαστεί σε πραγματικές συνθήκες. Ταυτόχρονα, αυτά τα SDK προσφέρουν διεπαφές προγραμματισμού εφαρμογών που λειτουργούν ως μέσο πρόσβασης σε δεδομένα αισθητήρων, καθώς και σε δυνατότητες υψηλού επιπέδου.

Πίνακας 2.2 Σύγκριση των SDK επαυξημένης πραγματικότητας

Χαρακτηριστικό	ARCore	ARKit	Vuforia	OpenXR
Πάροχος	Google	Apple	PTC	Khronos Group
Υποστηριζόμενες πλατφόρμες	Android	iOS / iPadOS	Android, iOS, Windows	Πολλαπλές (vendor-neutral)
Τύπος AR	Markerless	Markerless	Marker-based & Markerless	AR / VR / MR
Ανίχνευση επιφανειών	Ναι	Ναι	Ναι	Εξαρτάται από runtime
Spatial tracking	Υψηλή ακρίβεια	Πολύ υψηλή ακρίβεια	Καλή	Εξαρτάται από συσκευή
Υποστήριξη αισθητήρων	RGB, IMU	RGB, IMU, LiDAR (σε υποστηριζόμενες συσκευές)	RGB, IMU	Πολλαπλοί αισθητήρες
Υποστήριξη gesture interaction	Έμμεση (μέσω frameworks)	Έμμεση	Περιορισμένη	Υψηλή (ανάλογα runtime)
Ευκολία ανάπτυξης	Υψηλή	Υψηλή	Μέτρια	Χαμηλή–Μέτρια
Απόδοση σε πραγματικό χρόνο	Πολύ καλή	Εξαιρετική	Καλή	Εξαρτάται από υλοποίηση
Κόστος	Δωρεάν	Δωρεάν	Δωρεάν / Εμπορικό	Δωρεάν
Καταλληλότητα για εκπαίδευση	Υψηλή	Υψηλή	Υψηλή	Περιορισμένη
Καταλληλότητα για τη παρούσα εργασία	✓ Πολύ καλή	✓ Πολύ καλή	✓ Καλή	✗ Πολύ σύνθετη

Ο πίνακας 2.2 παρουσιάζει μια συγκριτική αξιολόγηση των σημαντικότερων πλατφορμών και SDK επαυξημένης πραγματικότητας. Η ανάλυση δείχνει ότι τα ARCore και ARKit προσφέρουν καλύτερη ακρίβεια χωρικής παρακολούθησης, προωθούν αποτελεσματικές διαδικασίες ανάπτυξης και έχουν πολύ χαμηλό κόστος, γεγονός που τα καθιστά ιδιαίτερα κατάλληλα για εκπαιδευτικές εφαρμογές και για πρακτικές εφαρμογές, όπως αυτές που αναφέρονται στην παρούσα διατριβή.

2.2.3.4 Βιβλιοθήκες Υπολογιστικής Όρασης

Η υπολογιστική όραση είναι αναπόσπαστο μέρος ενός λογισμικού επαυξημένης πραγματικότητας. Οι βιβλιοθήκες υπολογιστικής όρασης παρέχουν επίσης εφαρμογές επεξεργασίας εικόνας, ανίχνευσης χαρακτηριστικών και αλγορίθμων παρακολούθησης κίνησης.

Στα συστήματα AR, οι βιβλιοθήκες αυτές χρησιμοποιούνται για:

- αναγνώριση οπτικών δεικτών,
- markerless tracking,
- ανάλυση σκηνής και κατανόηση περιβάλλοντος.

Η επιλογή μιας κατάλληλης βιβλιοθήκης έχει τεράστια επίδραση στην ακρίβεια και την απόκριση του συστήματος, ειδικά σε εφαρμογές που απαιτούν αναγνώριση χειρονομιών σε πραγματικό χρόνο.

2.2.3.5 Διαχείριση δεδομένων αισθητήρων και συγχρονισμός

Ο χειρισμός και ο συντονισμός των πληροφοριών που εκπέμπονται από πολλαπλούς αισθητήρες είναι ένα από τα πιο επίπονα τεχνικά προβλήματα που αφορούν ένα σύστημα επαυξημένης πραγματικότητας. Οι ροές δεδομένων των καμερών και των μονάδων αδρανειακής μέτρησης (IMU) είναι διαφορετικές και συγχρονίζονται με μεγάλη προσοχή λόγω των διαφορετικών ρυθμών ανανέωσης, κάτι που θα εξασφάλιζε την κατάσταση του συστήματος.

Ο κακός έλεγχος συγχρονισμού μπορεί να προκαλέσει σφάλματα παρακολούθησης και υψηλή καθυστέρηση, τα οποία με τη σειρά τους επηρεάζουν αρνητικά την εμπειρία του χρήστη. Ως εκ τούτου, τα σύγχρονα κιτ ανάπτυξης λογισμικού (SDK) παρέχουν ισχυρό συγχρονισμό παράλληλα με φίλτρα εξομάλυνσης δεδομένων.

2.2.3.6 Βελτιστοποίηση απόδοσης και περιορισμοί λογισμικού

Για τα συστήματα επαυξημένης πραγματικότητας (AR), η απόδοση σε πραγματικό χρόνο είναι ένας κρίσιμος παράγοντας. Οι εφαρμογές πρέπει να επιτυγχάνουν υψηλούς ρυθμούς καρέ, ανεξάρτητα από τους περιορισμούς που προκαλούνται από συσκευές με περιορισμένες υπολογιστικές δυνατότητες.

Τα σχέδια βελτιστοποίησης περιλαμβάνουν:

- αποδοτική διαχείριση μνήμης,
- μείωση υπολογιστικού φόρτου,
- χρήση επιπέδων λεπτομέρειας (LOD),
- βελτιστοποίηση shaders και pipelines απόδοσης.

Η ανάγκη για βελτιστοποίηση έχει καθοριστική επίδραση στο σχεδιασμό της αρχιτεκτονικής του λογισμικού, η οποία, ωστόσο, ξεκινά από τα πρώτα στάδια της ανάπτυξης.

2.2.4 Υπολογιστική Όραση (Computer Vision) στα συστήματα AR

Τα περισσότερα συστήματα επαυξημένης πραγματικότητας (AR) βασίζονται στην υπολογιστική όραση. Μέσω της χρήσης συστημάτων επεξεργασίας εικόνας και ανάλυσης χαρακτηριστικών, τα συστήματα αυτά μπορούν να αναγνωρίσουν τη θέση των αντικειμένων, να εκτιμήσουν την κίνηση της κάμερας και να προσδιορίσουν τη δομή του περιβάλλοντος.

Κεφάλαιο 2

Μερικές από τις σημαντικότερες έννοιες της υπολογιστικής όρασης που σχετίζονται με την AR είναι:

- η ανίχνευση χαρακτηριστικών (feature detection),
- η αντιστοίχιση χαρακτηριστικών (feature matching),
- η εκτίμηση κίνησης (motion estimation).

Οι τεχνικές αυτές επιτρέπουν την ακριβή (χωρικά) τοποθέτηση αντικειμένων υπολογιστή, συμπεριλαμβανομένων των δυναμικών περιβαλλόντων.

2.2.5 Παρακολούθηση και Χωρική Εγγραφή

Είναι γενικά αποδεκτό ότι η σωστή εκτίμηση των χωρικών συντεταγμένων και του προσανατολισμού μιας συσκευής αποτελεί σημαντικό πρόβλημα στα συστήματα επαυξημένης πραγματικότητας (AR).

Οι προσεγγίσεις για τον εντοπισμό αυτό μπορούν να ταξινομηθούν ως εξής:

- σε οπτικούς δείκτες (marker-based),
- σε φυσικά χαρακτηριστικά του περιβάλλοντος (markerless),
- σε συνδυασμό αισθητήρων.

Η χωρική καταχώριση αναφέρεται στη σωστή συσχέτιση των ψηφιακών αντικειμένων με την πραγματική ζωή. Ακόμη και ένα μικρό σφάλμα καταχώρισης μπορεί να οδηγήσει σε έλλειψη ρεαλισμού και, κατά συνέπεια, σε απώλεια της εμπειρίας του χρήστη[1].

2.2.6 Απόδοση γραφικών και απόδοση σε πραγματικό χρόνο

Η ζωντανή δημιουργία γραφικών είναι ένα ουσιαστικό στοιχείο της λειτουργίας των συστημάτων επαυξημένης πραγματικότητας (AR). Αυτά τα συστήματα πρέπει να επιτυγχάνουν πολύ υψηλούς ρυθμούς ανανέωσης, προκειμένου να αποκλείεται η καθυστέρηση και να μειώνονται οι διαφορές μεταξύ της πραγματικής ζωής και της ψηφιακής απόδοσης.

Η βελτιστοποίηση της απόδοσης είναι επίσης ένα από τα βασικά ζητήματα, ειδικά σε κινητές πλατφόρμες με περιορισμένη υπολογιστική ικανότητα. Τεχνικές όπως η μείωση του αριθμού των πολυγώνων, η χρήση επιπέδων λεπτομέρειας (LOD) και η λογική χρήση της μνήμης είναι απαραίτητες για την ομαλή εκτέλεση των εφαρμογών.

2.2.7 Τεχνικές προκλήσεις και περιορισμοί

Παρά τη σημαντική πρόοδο που έχει επιτευχθεί τα τελευταία χρόνια, τα συστήματα AR εξακολουθούν να αντιμετωπίζουν μια σειρά από διάφορες προκλήσεις, όπως:

- μειωμένη ακρίβεια tracking σε κακές συνθήκες φωτισμού,
- αυξημένη κατανάλωση ενέργειας,
- περιορισμούς υλικού σε φορητές συσκευές,
- καθυστέρηση στην απόκριση.

Αυτά τα ζητήματα έχουν άμεση επίδραση στο σχεδιασμό και την υλοποίηση εφαρμογών επαυξημένης πραγματικότητας και πρέπει να λαμβάνονται υπόψη στα αρχικά στάδια της ανάπτυξης.

2.3 Αναγνώριση Χειρονομιών

2.3.1 Ορισμός και σημασία της αναγνώρισης χειρονομιών

Η αναγνώριση χειρονομιών είναι μια έννοια της αλληλεπίδρασης ανθρώπου-υπολογιστή και περιλαμβάνει την αυτόματη αναγνώριση χειρονομιών. Κληρονομιά χειρονομιών Ένας φυσικός τρόπος επικοινωνίας που μπορεί να χρησιμοποιηθεί για να επιτρέψει την επικοινωνία χωρίς την ανάγκη φυσικών συσκευών εισόδου όπως πληκτρολόγια ή χειριστήρια[13].

Στο διαδραστικό περιβάλλον της επαυξημένης πραγματικότητας, η αναγνώριση χειρονομιών διαδραματίζει καθοριστικό ρόλο, καθώς επιτρέπει στον χρήστη να χειρίζεται εικονικά αντικείμενα με τρόπο παρόμοιο με αυτόν που χειρίζεται τα πραγματικά αντικείμενα. Η αληθοφάνεια αυτής της επικοινωνίας ενισχύει την εμπειρία εμπύθισης και μειώνει τη γνωστική καθυστέρηση μεταξύ του χρήστη και του συστήματος[10].

2.3.2 Κατηγορίες χειρονομιών

Οι χειρονομίες μπορούν να κατηγοριοποιηθούν με βάση διάφορα κριτήρια, τα πιο σημαντικά από τα οποία είναι τα εξής:

- **Στατικές χειρονομίες:** Αυτές οι χειρονομίες αποτελούνται από μια συγκεκριμένη διάταξη των χεριών που καθορίζεται σε συγκεκριμένες χρονικές στιγμές. Παραδείγματα αυτού περιλαμβάνουν τη θέση της ανοιχτής παλάμης και την προφορά των περιγραμμένων σχημάτων των δακτύλων. Αν και η αναγνώριση με τη βοήθεια των στατικών χειρονομιών είναι αποτελεσματική, περιορίζει τη φυσικότητα της αλληλεπίδρασης μεταξύ ανθρώπου και υπολογιστή.
- **Δυναμικές χειρονομίες:** Παραδείγματα τέτοιων φαινομένων περιλαμβάνουν κινήσεις που αλλάζουν με την πάροδο του χρόνου, όπως η κατευθυντική κίνηση ενός χεριού. Παρόλο που είναι πιο εκφραστικές, απαιτούν την υιοθέτηση πιο σύνθετων αλγοριθμικών μεθόδων για την ανάλυση δεδομένων που έχουν χρονική διάσταση.
- **Συνεχείς χειρονομίες:** Χρησιμοποιούνται για συνεχή έλεγχο παραμέτρων, όπως περιστροφή ή μετακίνηση αντικειμένων. Είναι ιδιαίτερα χρήσιμες σε περιβάλλοντα AR, όπου απαιτείται συνεχής χειρισμός ψηφιακών αντικειμένων.

2.3.3 Προσεγγίσεις αναγνώρισης χειρονομιών

Η εφαρμογή της αναγνώρισης χειρονομιών μπορεί να γίνει με συγκεκριμένες τεχνολογικές επιλογές που διαφοροποιούνται ανάλογα με τον απαιτούμενο εξοπλισμό, την ακρίβεια, το κόστος και την ικανότητα εκτέλεσης συγκεκριμένων υπηρεσιών. Οι κυριότερες περιλαμβάνουν τις προσεγγίσεις που βασίζονται στην όραση και τους αισθητήρες, ενώ ορισμένα συστήματα χρησιμοποιούν υβριδικές προσεγγίσεις που συνδυάζουν στοιχεία και από τις δύο.

2.3.3.1 Vision-based αναγνώριση χειρονομιών

Η τεχνική που βασίζεται στην όραση βασίζεται στη χρήση καμερών και μεθόδων υπολογιστικής όρασης για τη διευκόλυνση της αναγνώρισης, της παρακολούθησης και της εξέτασης των κινήσεων των χεριών. Είναι η πιο κοινή μορφή διαδραστικού συστήματος στα σύγχρονα διαδραστικά συστήματα, καθώς δεν απαιτεί επαγγελματικό εξοπλισμό και μπορεί να υποστηριχθεί από τα πιο κοινά εργαλεία φιλικά προς τον καταναλωτή, όπως ταμπλέτες και κάμερες κινητών τηλεφώνων[14]. Η vision-based αναγνώριση χειρονομιών ακολουθεί συνήθως μια πολυεπίπεδη διαδικασία επεξεργασίας εικόνας:

1. **Απόκτηση εικόνας:** Η κάμερα καταγράφει συνεχώς τον χρήστη, ο οποίος συμβάλλει στο σύστημα. Η ακρίβεια της αναγνώρισης εξαρτάται άμεσα από την ποιότητα των εικόνων.
2. **Προεπεξεργασία:** Οι τεχνικές που χρησιμοποιούνται σε αυτό το μεθοδολογικό πλαίσιο περιλαμβάνουν την εξομάλυνση του θορύβου, την ομαλοποίηση του φωτισμού και τη μετατροπή του χρωματικού χώρου, με σκοπό τη βελτίωση της ποιότητας των δεδομένων.
3. **Τμηματοποίηση χεριού:** Ο στόχος είναι να κάνει το χέρι να ξεχωρίζει από το φόντο. Οι μέθοδοι που χρησιμοποιούνται είναι η χρήση χρωματικής τμηματοποίησης του δέρματος, η χρήση βάθους και η μηχανική μάθηση.
4. **Εξαγωγή χαρακτηριστικών:** Αυτό θα περιλαμβάνει την ανίχνευση σημαντικών ανατομικών χαρακτηριστικών στα σημεία αναφοράς των αρθρώσεων και των άκρων των δακτύλων, τη συστηματική περιγραφή των χωρικών και χρονικών κινηματικών ιδιοτήτων της κίνησης.
5. **Αναγνώριση και ταξινόμηση:** Τα χαρακτηριστικά που εξάγονται στη συνέχεια εισάγονται σε αλγόριθμους ταξινόμησης, όπως μοντέλα μηχανικής μάθησης και νευρωνικά δίκτυα, για την αναγνώριση χειρονομιών.

Αυτή η ακολουθία επιτρέπει την αναγνώριση τόσο στατικών όσο και δυναμικών χειρονομιών σε πραγματικό χρόνο.

Η προσέγγιση που βασίζεται στην όραση έχει το πλεονέκτημα της οικονομικής αποδοτικότητας, του ευρέος φάσματος εφαρμογών και της εύκολης ενσωμάτωσης σε εφαρμογές επαυξημένης πραγματικότητας (AR). Ωστόσο, η απόδοση εξαρτάται από περιβαλλοντικούς παράγοντες, όπως ο φωτισμός και οι αλλαγές στο φόντο, καθώς και από την κάλυψη αντικειμένων, γεγονός που απαιτεί προσεκτική ανάπτυξη του συστήματος και εφαρμογή προηγμένων αλγοριθμικών συστημάτων[13].

2.3.3.2 Sensor-based αναγνώριση χειρονομιών

Η μέθοδος που βασίζεται σε αισθητήρες βασίζεται στην ιδέα της χρήσης εξειδικευμένων οργάνων για την καταγραφή της ουσίας των κινήσεων των χεριών, συμπεριλαμβανομένων γαντιών με αισθητήρες κάμψης, κάμερες ανίχνευσης βάθους και κάμερες υπερύθρων.

Η κύρια διαφορά με το μοντέλο που βασίζεται στην όραση είναι ότι τα κινηματικά δεδομένα λαμβάνονται απευθείας από το υλικό του αισθητήρα και, ως εκ τούτου, δεν απαιτείται επεξεργασία εικόνας. Αυτή η άμεση λήψη καθιστά τη μέθοδο ακριβή και σταθερή και δεν επηρεάζεται από το ζήτημα του φωτισμού του περιβάλλοντος.

Ωστόσο, η ανάγκη για ειδικό υλικό αυξάνει το κόστος και μειώνει τη χρησιμότητα, ειδικά στον τομέα της εκπαίδευσης, όπου η απλότητα και η προσβασιμότητα είναι οι κύριοι σημαντικοί παράμετροι.

2.3.3.3 Υβριδικές προσεγγίσεις

Πιο σύνθετα συστήματα συνδυάζουν πληροφορίες από τις κάμερες και τους αισθητήρες βάθους ή τις μονάδες αδρανειακών μετρήσεων και τις συγχωνεύουν χρησιμοποιώντας υβριδικές μεθοδολογίες. Αυτές οι μεθοδολογίες χρησιμοποιούν τεχνική συγχώνευσης αισθητήρων για να αξιοποιήσουν τα συμπληρωματικά πλεονεκτήματα των διαφορετικών πηγών δεδομένων.

Παρά τα πλεονεκτήματα των υβριδικών λύσεων, όπως η αυξημένη ακρίβεια και αξιοπιστία, η εφαρμογή τους είναι πιο περίπλοκη και απαιτεί πιο ισχυρό υλικό, γεγονός που τις καθιστά λιγότερο κατάλληλες για το χαμηλού κόστους εκπαιδευτικό περιβάλλον.

2.3.3.4 Κριτική σύγκριση και επιλογή προσέγγισης

Η επιλογή της κατάλληλης μεθοδολογικής προσέγγισης εξαρτάται από τις συγκεκριμένες ανάγκες του τομέα εφαρμογής. Μια προσέγγιση βασισμένη στην όραση θεωρείται η καταλληλότερη για ένα σύστημα επαυξημένης πραγματικότητας που προορίζεται για χρήση σε εκπαιδευτικά περιβάλλοντα, καθώς προσφέρει επαρκή ακρίβεια, χαμηλό κόστος και δυνατότητα προσαρμογής σε κινητές συσκευές.

Στο πλαίσιο της παρούσας διατριβής, επιλέγεται μια προσέγγιση αναγνώρισης χειρονομιών βασισμένη στην όραση, η οποία είναι σύμφωνη με τον παιδαγωγικό σκοπό του συστήματος και επιτρέπει τη δημιουργία μιας φυσικής διεπαφής αλληλεπίδρασης χωρίς τη βοήθεια εξειδικευμένου υλικού.

Πίνακας 2.3 Συγκριτική αξιολόγηση frameworks αναγνώρισης χειρονομιών

Framework	Τύπος προσέγγισης	Απαιτούμενος εξοπλισμός	Ακρίβεια	Ευκολία ενσωμάτωσης	Καταλληλότητα για AR	Καταλληλότητα για εκπαίδευση
MediaPipe Hands	Vision-based	RGB κάμερα	Υψηλή	Πολύ υψηλή	✓ Πολύ καλή	✓ Πολύ καλή
OpenPose	Vision-based	RGB κάμερα	Πολύ υψηλή	Μέτρια	✓ Καλή	Μέτρια
Leap Motion SDK	Sensor-based	Ειδικός αισθητήρας	Πολύ υψηλή	Υψηλή	✓ Πολύ καλή	Περιορισμένη
Kinect SDK	Sensor-based	Αισθητήρας βάθους	Υψηλή	Μέτρια	✓ Καλή	Περιορισμένη
ManoMotion	Vision-based (Mobile AR)	RGB κάμερα κινητής συσκευής	Υψηλή	Πολύ υψηλή	✓ Εξαιρετική	✓ Πολύ καλή
OpenCV (custom)	Vision-based	RGB κάμερα	Εξαρτάται υλοποίησης	Χαμηλή	Μέτρια	Χαμηλή

Ο πίνακας 2.3 παρουσιάζει μια σχετική αξιολόγηση των τρεχόντων μοντέλων αναγνώρισης χειρονομιών. Είναι σαφές ότι οι προσεγγίσεις που βασίζονται στην όραση θα προσφέρουν μεγαλύτερη οικονομική προστιτότητα με μειωμένο κόστος και, ως εκ τούτου, θα είναι ιδιαίτερα κατάλληλες για εκπαιδευτικά συστήματα AR, όπως αυτό που περιγράφεται στην παρούσα διατριβή. Το πλαίσιο ManoMotion, το οποίο παρέχει δυνατότητες αναγνώρισης χειρονομιών σε κινητές πλατφόρμες χωρίς την ανάγκη χρήσης ειδικού υλικού, παρουσιάζει ιδιαίτερο ενδιαφέρον. Το ManoMotion σχεδιάστηκε για να εφαρμόζεται σε σενάρια AR

και παρέχει υψηλή ακρίβεια, εύκολη ενσωμάτωση και υποστήριξη αλληλεπίδρασης με φυσικές διεπαφές χρήστη, καθιστώντας το ιδιαίτερα κατάλληλο για εκπαιδευτικές εφαρμογές στο περιβάλλον των κινητών συσκευών.

2.3.4 Αλγοριθμικές τεχνικές αναγνώρισης χειρονομιών

Οι αλγόριθμοι αναγνώρισης χειρονομιών περιλαμβάνουν τεχνικές όπως:

- ανίχνευση και παρακολούθηση χαρακτηριστικών χεριού,
- ταξινόμηση μέσω μηχανικής μάθησης,
- νευρωνικά δίκτυα και βαθιά μάθηση για αναγνώριση προτύπων κίνησης.

Η εφαρμογή σύγχρονων μεθόδων βαθιάς μάθησης έχει βελτιώσει σημαντικά την ακρίβεια και την ισχύ των συστημάτων αναγνώρισης χειρονομιών, ειδικά σε πολυσύχναστες περιοχές.

2.3.5 Αναγνώριση χειρονομιών σε περιβάλλοντα Επαυξημένης Πραγματικότητας

Η αναγνώριση χειρονομιών χρησιμοποιείται επίσης σε περιβάλλοντα επαυξημένης πραγματικότητας (AR), όπου χρησιμοποιείται για την άμεση αλληλεπίδραση και τον χειρισμό ψηφιακών οντοτήτων, συμπεριλαμβανομένης της περιστροφής, της μεγέθυνσης και της μετακίνησης γύρω από αυτές. Αυτός ο τύπος φυσικής αλληλεπίδρασης είναι ιδιαίτερα χρήσιμος όταν εφαρμόζεται στην εκπαίδευση, όπου συμπληρώνει τη μάθηση μέσω εμπειριών και υποστηρίζει επίσης την ενεργό συμμετοχή των μαθητών[12].

Η εφαρμογή του συστήματος αναγνώρισης χειρονομιών σε εφαρμογές επαυξημένης πραγματικότητας απαιτεί στενό συντονισμό της δραστηριότητας παρακολούθησης και απόδοσης, που εγγυάται την άμεση αντανάκλαση της κίνησης του χρήστη στο αντίστοιχο ψηφιακό περιεχόμενο.

2.3.6 Προκλήσεις και περιορισμοί

Παρά τις δραστικές αυτές βελτιώσεις, η αναγνώριση χειρονομιών συνεχίζει να αντιμετωπίζει μια σειρά προκλήσεων, μεταξύ των οποίων:

- ασάφεια μεταξύ παρόμοιων χειρονομιών,
- κόπωση χρήστη,
- ανάγκη για χαμηλή καθυστέρηση,
- περιορισμοί ακρίβειας σε πραγματικά περιβάλλοντα.

Αυτοί οι περιορισμοί δεν πρέπει να υποτιμηθούν όταν πρόκειται για την πλήρη γνώση του τι είναι δυνατό και τι αδύνατο να επιτευχθεί ο καλύτερος σχεδιασμός συστημάτων επαυξημένης πραγματικότητας με αλληλεπίδραση βασισμένη σε χειρονομίες.

2.3.7 Σχέση με την παρούσα διπλωματική εργασία

Η διατριβή χρησιμοποιεί αναγνώριση χειρονομιών με βάση την όραση, η οποία αποτελεί έναν λογικό συμβιβασμό μεταξύ ακρίβειας, κόστους και προσβασιμότητας. Μια τέτοια στρατηγική ανταποκρίνεται στις ανάγκες των εκπαιδευτικών εφαρμογών και βοηθά στην αλληλεπίδραση με χημικά πειράματα που υλοποιούνται σε ένα περιβάλλον επαυξημένης πραγματικότητας με φυσικό τρόπο.

2.4 Ανθρώπινη-Υπολογιστική Αλληλεπίδραση

2.4.1 Ορισμός και αντικείμενο της Ανθρώπινης-Υπολογιστικής Αλληλεπίδρασης

Η Αλληλεπίδραση Ανθρώπου-Υπολογιστή (HCI) αποτελεί ένα διεπιστημονικό πεδίο, το οποίο ασχολείται με την επιστήμη του συστηματικού σχεδιασμού, της κριτικής αξιολόγησης και της προσεκτικής εφαρμογής διαδραστικών υπολογιστικών συστημάτων, με γνώμονα τους ανθρώπινους παράγοντες. Ο στόχος της έρευνας HCI είναι να δημιουργήσει συστήματα που να παρουσιάζουν αποτελεσματικότητα, αποδοτικότητα, χρηστικότητα και ευχαρίστηση για τους χρήστες, και να αξιοποιούν τις γνωστικές, φυσικές και κοινωνικές δυνατότητες των τελικών χρηστών[11].

Η Αλληλεπίδραση Ανθρώπου-Υπολογιστή (HCI) είναι ένας τομέας που ενσωματώνει την επιστήμη των υπολογιστών, την ψυχολογία, την εργονομία και τις γνωστικές θεωρίες και πρακτικές. Έχει ιδιαίτερη σημασία στα σύγχρονα διαδραστικά συστήματα, ιδίως σε εκείνα που χρησιμοποιούνται στην Επαυξημένη Πραγματικότητα, όταν η διεπαφή δεν είναι πλέον μόνο μια οθόνη και απλές συσκευές εισόδου, αλλά επεκτείνεται στον ευρύτερο φυσικό χώρο και στο ανθρώπινο σώμα.

2.4.2 Εξέλιξη διεπαφών χρήστη

Οι διεπαφές χρήστη έχουν εξελιχθεί σημαντικά τις τελευταίες δεκαετίες. Ξεκινώντας από τη διεπαφή γραμμής εντολών και προχωρώντας στις γραφικές διεπαφές χρήστη (GUI), τα πρότυπα αλληλεπίδρασης έχουν μετατοπιστεί σταδιακά προς πιο φυσικές μορφές αλληλεπίδρασης ανθρώπου-υπολογιστή.

Η εισαγωγή των κινητών συσκευών, των τεχνολογιών αισθητήρων και της υπολογιστικής όρασης έχει προκαλέσει την ανάπτυξη διεπαφών που βασίζονται στις λειτουργίες αφής, κίνησης και φωνής. Μια τέτοια αλλαγή στο πρότυπο απαιτούσε έναν νέο ορισμό των αρχών σχεδιασμού διεπαφών, όπου οι χρήστες αλληλεπιδρούν περισσότερο με ένα υπολογιστικό σύστημα σε πιο μη συμβατικές περιβαλλοντικές ρυθμίσεις[9].

2.4.3 Φυσικές Διεπαφές Χρήστη

Η έννοια των Φυσικών Διεπαφών Χρήστη (NUI) επιχειρεί να φέρει πιο κοντά την ανθρώπινη βούληση και τη δύναμη των υπολογιστών, ώστε να παρέχει στον χρήστη τη δυνατότητα αλληλεπίδρασης με τα συστήματα μέσω κινήσεων και χειρονομιών.

Η αναγνώριση χειρονομιών είναι επίσης ένα σημαντικό συστατικό των NUI, καθώς οι χειρονομίες επιτρέπουν τη μετάφραση των κινητικών εντολών του ανθρώπου απευθείας σε εντολές του συστήματος. Στο περιβάλλον της επαυξημένης πραγματικότητας (AR), ο ρόλος των NUI γίνεται ακόμη πιο σημαντικός, καθώς ο χρήστης αλληλεπιδρά όχι μόνο με μια συμβατική οθόνη, αλλά και με την παρουσία ψηφιακών αντικειμένων που αποκαλύπτονται στον πραγματικό κόσμο.

Πειραματική έρευνα δείχνει ότι η χρήση φυσικών διεπαφών μειώνει τη χρηστικότητα και την αλληλεπίδραση του χρήστη, ειδικά με εφαρμογές μάθησης και περιβάλλοντα μάθησης[10].

2.4.4 Ενσώματη Αλληλεπίδραση (Embodied Interaction)

Η ενσώματη αλληλεπίδραση έχει περιγραφεί ως ένα εννοιολογικό μοντέλο, το οποίο εξηγεί την αλληλεπίδραση ανθρώπου-υπολογιστή στο πλαίσιο της φυσικής παρουσίας και της σωματικής κίνησης των χρηστών στο χώρο. Ο Dourish (2001) υποστηρίζει ότι η κατανόηση και η γνώση δεν προκύπτουν ως αφηρημένη δράση, αλλά προκύπτουν μέσω της φυσικής αλληλεπίδρασης με το περιβάλλον[9].

Η ενσώματη αλληλεπίδραση είναι ένας παράγοντας σχεδιασμού στα συστήματα επαυξημένης πραγματικότητας. Ο χρήστης έχει τον έλεγχο του χώρου και χρησιμοποιεί τα χέρια του για να αλληλεπιδράσει με κάθε είδους ερεθίσματα και αλληλεπιδρά με εικονικά αντικείμενα με τρόπο που μοιάζει με την αλληλεπίδραση με πραγματικά αντικείμενα. Με αυτόν τον τρόπο, η αλληλεπίδραση γίνεται πιο εντυπωσιακή και παρούσα, καθιστώντας την πιο διαισθητική.

2.4.5 Εμπειρία Χρήστη

Η εμπειρία χρήστη (UX) περιγράφεται ως το άθροισμα των συναισθημάτων, των αντιλήψεων και των αντιδράσεων που εμφανίζονται μετά την αλληλεπίδραση ενός χρήστη με ένα σύστημα. Στα συστήματα επαυξημένης πραγματικότητας (AR), η UX καθορίζεται από:

- η ακρίβεια της χωρικής εγγραφής,
- η καθυστέρηση απόκρισης,
- ο οπτικός φόρτος,
- η φυσικότητα της αλληλεπίδρασης.

Η έλλειψη σχεδιασμού της εμπειρίας χρήστη σχετίζεται με υψηλότερα επίπεδα κόπωσης, σύγχυσης και περισπασμών. Αντίθετα, ο καλά σχεδιασμένος σχεδιασμός θα διευκολύνει την αλληλεπίδραση και θα ενισχύσει την αποτελεσματικότητα του συστήματος, ειδικά στον εκπαιδευτικό τομέα, όπου η συγκέντρωση και η κατανόηση είναι καθοριστικής σημασίας[11].

2.4.6 HCI και Εκπαιδευτικά Διαδραστικά Συστήματα

Η χρήση των αρχών της Αλληλεπίδρασης Ανθρώπου-Υπολογιστή (HCI) στο εκπαιδευτικό περιβάλλον έχει αποδειχθεί ότι συμβάλλει στη βελτίωση της εκπαιδευτικής εμπειρίας των μαθητών και στην κατανόηση περίπλοκων ιδεών. Τα διαδραστικά συστήματα παρέχουν επίσης στους μαθητές τη δυνατότητα να πειραματίζονται και να λαμβάνουν άμεση ανατροφοδότηση, καθώς και να συμμετέχουν στη μαθησιακή διαδικασία.

Η HCI αποτελεί κεντρικό στοιχείο στην υποστήριξη της θεωρίας και της πρακτικής στο πλαίσιο των περιβαλλόντων επαυξημένης πραγματικότητας (AR). Οι χειρονομιακές αλληλεπιδράσεις που χρησιμοποιούνται στη διεξαγωγή χημικών πειραμάτων βοηθούν τους μαθητές να αποκτήσουν γνώσεις μέσω της εμπειρίας και να συνδέσουν τις αφηρημένες ιδέες με την πραγματική ζωή[12].

2.4.7 Σχέση της HCI με την παρούσα διπλωματική εργασία

Στο πλαίσιο αυτής της διατριβής, οι αρχές της αλληλεπίδρασης ανθρώπου-υπολογιστή αποτελούν το θεωρητικό υπόβαθρο για την ανάπτυξη μιας εφαρμογής επαυξημένης πραγματικότητας με αναγνώριση χειρονομιών. Η χρήση φυσικών διεπαφών και η έμφαση στην ενσωματωμένη αλληλεπίδραση στοχεύουν

στην ανάπτυξη ενός συστήματος φιλικού προς τον χρήστη, παιδαγωγικά αποτελεσματικού και τεχνολογικά σταθερού.

Η βαθιάς κατανόηση των εννοιών της HCI διευκολύνει τον σχεδιασμό διεπαφών που υπερβαίνουν τη λειτουργικότητα και περιλαμβάνουν τις εμπειρίες και τις ανάγκες των χρηστών, κάτι που είναι υψίστης σημασίας για την επιτυχία της εφαρμογής.

2.5 Επαυξημένη Πραγματικότητα στην Εκπαίδευση

2.5.1 Εισαγωγή στην αξιοποίηση της Επαυξημένης Πραγματικότητας στην εκπαίδευση

Η ενσωμάτωση των ψηφιακών τεχνολογιών στη διδακτική διαδικασία έχει συμβάλει στην εμφάνιση νέων μεθόδων διδασκαλίας που δίνουν μεγαλύτερη έμφαση στην ενεργή αλληλεπίδραση των μαθητών και στην τελική μάθηση. Σε αυτό το περιβάλλον, η επαυξημένη πραγματικότητα (AR) έχει αναγνωριστεί ως μια αρκετά υποσχόμενη τεχνολογία στον τομέα της διδασκαλίας, καθώς επιτρέπει την ενίσχυση του σωματικού μαθησιακού περιβάλλοντος με ψηφιακές πληροφορίες σε πραγματικό χρόνο[3].

Η Ε.Π. είναι κάτι που προστίθεται στο φυσικό περιβάλλον και δεν το αντικαθιστά, αλλά επιτρέπει στους μαθητές να αλληλεπιδρούν με εικονικά αντικείμενα και να παραμένουν στο φυσικό χώρο. Αυτή η πτυχή καθιστά την AR ιδιαίτερα κατάλληλη για εκπαιδευτικά ιδρύματα, όπου ο συνδυασμός θεωρίας και πρακτικής πρέπει να είναι μία από τις βασικές απαιτήσεις.

2.5.2 Παιδαγωγικές θεωρίες και Επαυξημένη Πραγματικότητα

Επιβεβαιώνεται η εμπειρική συσχέτιση της ενσωμάτωσης της επαυξημένης πραγματικότητας (AR) στην εκπαίδευση με τα σύγχρονα παιδαγωγικά παραδείγματα, δηλαδή τον κονστрукτιβισμό και τις ιδεολογίες της εμπειρικής μάθησης. Σε αυτές τις θεωρητικές προοπτικές, η απόκτηση γνώσης θεωρείται ως μια διαδραστική διαδικασία και όχι ως παθητική πρόσληψη εκ μέρους ενός μαθητή που αλληλεπιδρά με το περιβάλλον του.

Η Ε.Π. υποστηρίζει τη δομητική διδασκαλία, καθώς επιτρέπει στους μαθητές να αναζητούν, να εξερευνούν και να βλέπουν τις συνέπειες των παρεμβάσεών τους σε ένα επαυξημένο περιβάλλον. Επιπλέον, υποστηρίζει τις έννοιες της εμπειρικής μάθησης, καθώς οι μαθητές χρησιμοποιούν τις φυσικές και αισθητηριακές τους ικανότητες στην προσπάθειά τους να κατανοήσουν τις αφηρημένες ιδέες[6].

2.5.3 Πλεονεκτήματα της Επαυξημένης Πραγματικότητας στην εκπαίδευση

Η χρήση της επαυξημένης πραγματικότητας στον εκπαιδευτικό τομέα προσφέρει πληθώρα πλεονεκτημάτων, τα σημαντικότερα από τα οποία παρατίθενται παρακάτω:

- **Ενίσχυση της μαθησιακής εμπλοκής:** Η επαυξημένη πραγματικότητα (AR) ενθαρρύνει τους μαθητές να δείχνουν μεγαλύτερο ενδιαφέρον και κίνητρο, αυξάνοντας έτσι το επίπεδο συμμετοχής τους χάρη στη διαδραστική συμπεριφορά.
- **Οπτικοποίηση αφηρημένων εννοιών:** Με τη βοήθεια της επαυξημένης πραγματικότητας, οι αφηρημένες (αόρατες) διαδικασίες, συμπεριλαμβανομένης της μοριακής αλληλεπίδρασης, της χημικής αντίδρασης ή οποιουδήποτε τύπου διαδικασίας, μπορούν να απεικονιστούν με τέτοιο τρόπο ώστε να βελτιώνεται η κατανόησή τους.

- **Ασφαλές περιβάλλον πειραματισμού:** Οι μαθητές μπορούν να εκτελούν πειραματικές διαδικασίες χωρίς κίνδυνο, κάτι που αποτελεί σημαντικό παράγοντα σε μαθήματα όπως η χημεία.
- **Υποστήριξη εξατομικευμένης μάθησης:** Η επαυξημένη πραγματικότητα (AR) επιτρέπει επίσης τη ρύθμιση του ρυθμού των οδηγιών και του περιεχομένου της μάθησης, ώστε να ανταποκρίνεται στις μοναδικές ανάγκες κάθε μαθητή.

Ο ισχυρισμός ότι η χρήση της επαυξημένης πραγματικότητας βελτιώνει την κατανόηση και τη μακροπρόθεσμη μνήμη των πληροφοριών αποδεικνύεται από τις εμπειρικές μελέτες[12].

2.5.4 Περιορισμοί και προκλήσεις της Ε.Π. στην εκπαίδευση

Παρά τα πλεονεκτήματα της χρήσης της επαυξημένης πραγματικότητας στην εκπαίδευση, υπάρχουν αρκετοί περιορισμοί που μπορούν να συσχετιστούν με την εφαρμογή της. Αυτοί περιλαμβάνουν:

- τεχνικοί περιορισμοί συσκευών και υποδομών,
- ανάγκη επιμόρφωσης εκπαιδευτικών,
- πιθανή αύξηση γνωστικού φορτίου λόγω υπερβολικής πληροφορίας,
- ζητήματα εργονομίας και κόπωσης.

Η επιτυχής εφαρμογή της επαυξημένης πραγματικότητας απαιτεί έναν λεπτομερή σχεδιασμό που θα εγγυάται ότι τα ψηφιακά στοιχεία θα χρησιμοποιηθούν στη μαθησιακή διαδικασία, αποφεύγοντας την απόσπαση της προσοχής των μαθητών.

2.5.5 Επαυξημένη Πραγματικότητα στη διδασκαλία των Φυσικών Επιστημών

Η χρήση της επαυξημένης πραγματικότητας στις φυσικές επιστήμες είναι ιδιαίτερα ενδιαφέρουσα, καθώς συχνά τα θέματα που μελετώνται και ερευνώνται είναι αφηρημένα φαινόμενα και διαδικασίες που δεν μπορούν να παρατηρηθούν άμεσα. Σε άλλους τομείς, όπως η χημεία, η κατανόηση των φαινομένων απαιτεί συνδυασμό μακροσκοπικής και μικροσκοπικής καθώς και συμβολικής αναπαράστασης.

Η επαυξημένη πραγματικότητα λειτουργεί ως μεσολαβητής μεταξύ των δύο επιπέδων, επιτρέποντας την προβολή σε πραγματικό χρόνο των πειραματικών διαδικασιών και την οπτικοποίηση της διαδικασίας της μοριακής δυναμικής. Αυτό σημαίνει ότι η γνωστική απόσταση μειώνεται, δημιουργώντας έτσι καλύτερη εννοιολογική κατανόηση.

2.5.6 Επαυξημένη Πραγματικότητα και πειράματα Χημείας στη δευτεροβάθμια εκπαίδευση

Στο γυμνάσιο, ειδικά στη δεύτερη τάξη, τα εργαστηριακά πειράματα στη χημεία αποτελούν ένα σημαντικό εργαλείο για την κατανόηση των εννοιών. Ωστόσο, η εφαρμογή τους συχνά περιορίζεται λόγω ανεπαρκούς εξοπλισμού, έλλειψης χρόνου ή ακόμη και λόγω ανησυχιών για την ασφάλεια.

Οι τεχνολογίες AR μπορούν να χρησιμοποιηθούν για να συμπληρώσουν το παραδοσιακό εργαστηριακό περιβάλλον, επιτρέποντας στους μαθητές να πραγματοποιούν πειράματα σε ένα εικονικό περιβάλλον και, ταυτόχρονα, να παρακολουθούν τις εννοιολογικές επιπτώσεις της παρέμβασής τους. Επιπλέον, η χρήση τεχνολογιών αναγνώρισης χειρονομιών θα συμβάλει στην αύξηση της φυσικότητας της αλληλεπίδρασης με τον χρήστη, γεγονός που θα κάνει την εκπαιδευτική διαδικασία πιο ενδιαφέρουσα και αποτελεσματική.

2.6 Επίλογος

Η διατριβή χρησιμοποιεί την επαυξημένη πραγματικότητα ως παιδαγωγικό εργαλείο για τη διδασκαλία εργαστηριακών πειραμάτων χημείας σε μαθητές της δευτέρας γυμνασίου. Μπορεί να υποτεθεί ότι με την ενσωμάτωση της επαυξημένης πραγματικότητας και άλλων προσεγγίσεων που βασίζονται στην αναγνώριση χειρονομιών, σε συνδυασμό με τις αρχές της αλληλεπίδρασης ανθρώπου-υπολογιστή, θα είναι δυνατή η δημιουργία μιας εκπαιδευτικής εφαρμογής που θα εφαρμόζει το μοντέλο της βιοματικής μάθησης και θα συμβάλλει στην ενεργό συμμετοχή και τη θεωρητική κατανόηση.

3 Σχεδίαση και Αρχιτεκτονική Συστήματος

3.1 Εισαγωγή

Πριν ξεκινήσει η ανάπτυξη της εφαρμογής, θα οριοθετηθεί η δομική διάταξη και η λειτουργική ιεραρχία του συστήματος. Αυτός ο σχεδιασμός βοηθά στην αποσαφήνιση των βασικών στοιχείων της εφαρμογής, καθορίζει τα απαραίτητα χαρακτηριστικά και την ανάλυση των αλληλεπιδράσεων μεταξύ χρηστών, μονάδων λογισμικού και υλικού. Η συστηματική προσέγγιση εγγυάται κατά συνέπεια μια διαδικασία ανάπτυξης βασισμένη σε ακολουθίες, μετριάζει πιθανά λάθη και ενισχύει τη μακροπρόθεσμη βιωσιμότητα του έργου.

Το κεφάλαιο αυτό παρουσιάζει τον συνολικό σχεδιασμό μιας εφαρμογής επαυξημένης πραγματικότητας με λειτουργία αναγνώρισης χειρονομιών, η οποία αναπτύσσεται ως μέρος της παρούσας διατριβής. Ο σχεδιασμός αυτός βασίζεται στο θεωρητικό υπόβαθρο που παρουσιάζεται στο Κεφάλαιο 2 και εφαρμόζει τις βασικές αρχές που χρησιμοποιούνται στην αλληλεπίδραση ανθρώπου-υπολογιστή, στην επαυξημένη πραγματικότητα και στην εκπαιδευτική τεχνολογία.

3.2 Περιγραφή συνολικού συστήματος

Η εφαρμογή σχεδιάστηκε ως εκπαιδευτικό εργαλείο σε κινητές πλατφόρμες Android ως επαυξημένη πραγματικότητα (AR) και είχε ως στόχο την οπτικοποίηση ενός χημικού πειράματος χρησιμοποιώντας διαδραστικά στοιχεία. Το σύστημα χρησιμοποιεί την κάμερα της συσκευής για να πραγματοποιήσει τη διαδικασία αναγνώρισης του περιβάλλοντος και να προβάλει τρισδιάστατα αντικείμενα, επιτρέποντας έτσι στους χρήστες να αλληλεπιδρούν με εικονικά αντικείμενα μέσω χειρονομιών χωρίς τη χρήση πρόσθετων συσκευών.

Η εφαρμογή απαρτίζεται από τα παρακάτω υποσυστήματα:

- **Υποσύστημα Επαυξημένης Πραγματικότητας (AR Engine):** Με την χρήση του AR Foundation και του ARCore, μπορεί να αναγνωρίζει το περιβάλλον και τα στοιχεία από τα οποία απαρτίζεται, και να προσθέτει τα ψηφιακά αντικείμενα στον χώρο.
- **Υποσύστημα Αναγνώρισης Χειρονομιών (Gesture Recognition):** Χρησιμοποιεί το Manomotion SDK για να αναγνωρίζει σε πραγματικό χρόνο τις κινήσεις του χεριού.
- **Υποσύστημα Εκπαιδευτικού Περιεχομένου:** Σε αυτό το μέρος βρίσκονται τα τρισδιάστατα αντικείμενα που χρησιμοποιούνται στο πείραμα, όπως το φλόγιστρο, και τα διάφορα σενάρια διάδρασης που έχουν τις διαφορετικές συμπεριφορές κατά την χρήση.
- **Υποσύστημα Χρήστη (User Interaction Layer):** Αυτό το υποσύστημα γεφυρώνει τις κινήσεις των χεριών με τα αποτελέσματα που δίνει η εφαρμογή για τις εκάστοτε κινήσεις.

3.3 Σχεδιασμός Αλληλεπίδρασης Χρήστη – Συστήματος

Η εμπειρία αλληλεπίδρασης και η διεπαφή χρήστη θεωρούνται βασικοί παράγοντες για την επιτυχία των εφαρμογών μάθησης AR. Στην συγκεκριμένη εφαρμογή, ο σχεδιασμός της αλληλεπίδρασης ακολουθεί ένα φυσικό και διαισθητικό μοντέλο, όπου οι χρήστες μπορούν να χειρίζονται το περιεχόμενο μέσω χειρονομιών και να αποφεύγουν τη χρήση φυσικών κουμπιών ή βοηθητικών συσκευών.

Ο χρήστης κρατά τη φορητή συσκευή android με το ένα χέρι, ενώ με το άλλο πραγματοποιεί κινήσεις στον χώρο, μπροστά από την κάμερα. Η εφαρμογή λειτουργεί με το Manomotion SDK ώστε να μπορεί να αναγνωρίζει χειρονομίες και να τις μετατρέπει σε ενέργειες στο εικονικό περιβάλλον.

3.3.1 Τυπική σειρά αλληλεπίδρασης

Η συνολική ροή λειτουργίας περιλαμβάνει την εξής σειρά:

1. Ο χρήστης κρατάει τη συσκευή με το ένα χέρι, εκκινεί την εφαρμογή και εμφανίζεται ένα splashscreen.
2. Ενεργοποιείται η κάμερα.
3. Στη συνέχεια εμφανίζεται ένα παράθυρο με οδηγίες για τον χρήστη.
4. Ο χρήστης σημαδεύει με την κάμερα το εξώφυλλο του βιβλίου χημείας της 'β γυμνασίου.
5. Στη συνέχεια εμφανίζεται η AR σκηνή με το φλόγιστρο και τα αντικείμενα.
6. Το Manomotion SDK αναγνωρίζει το χέρι και μεταβιβάζει πληροφορίες στο σύστημα.
7. Η εφαρμογή προσομοιώνει τις αντίστοιχες χημικές καύσεις.

Η εφαρμογή έχει δημιουργηθεί με τέτοιο τρόπο ώστε να επιτρέπει την επέκταση της και με άλλα στοιχεία και λειτουργίες.

3.3.2 Εμπειρία χρήστη (User Experience)

Η επικοινωνία βασίζεται σε ένα πρότυπο παρατήρησης και δράσης. Ορισμένα αντικείμενα εμφανίζονται πάνω από το βιβλίο χημείας και δίνουν στον χρήστη την αίσθηση πως είναι πραγματικά και μπορεί να τα πιάσει. Τα βήματα της διαδικασίας σχεδιάστηκαν με τέτοιο τρόπο ώστε να παρέχουν οπτική και λογική ανατροφοδότηση, η οποία θα ενίσχυε την κατανόηση και τη μνήμη. Αντί για τη συμβατική διεπαφή με κουμπιά ή γραφική απεικόνιση, επιλέχθηκαν χειρονομίες, καθώς αποτελούν έναν τρόπο ενίσχυσης της εμπειρικής μάθησης και επιτρέπουν στον μαθητή να προσεγγίσει το φαινόμενο με έναν ενεργό ερευνητικό τρόπο.

3.4 Λειτουργικές και μη Λειτουργικές Προδιαγραφές

Ο σχεδιασμός του συστήματος περιλαμβάνει την διατύπωση των απαιτήσεων, η οποία αποτελεί ένα ζωτικό βήμα στη διαδικασία σχεδιασμού του συστήματος, καθώς καθορίζει τα όρια λειτουργίας και τα επιθυμητά χαρακτηριστικά του τελικού προϊόντος. Στο πλαίσιο αυτής της εφαρμογής, οι απαιτήσεις χωρίζονται σε λειτουργικές προδιαγραφές, οι οποίες αναφέρονται στις ενέργειες που πραγματοποιούνται στο σύστημα, και σε μη λειτουργικές προδιαγραφές, οι οποίες αναφέρονται στον τρόπο με τον οποίο πραγματοποιούνται αυτές οι ενέργειες.

3.4.1 Λειτουργικές Προδιαγραφές

Οι βασικές λειτουργικές προδιαγραφές της εφαρμογής περιλαμβάνουν:

- Χρήση του ARCore για να εκκινήσει το περιβάλλον Ε.Π. όταν ενεργοποιείται η κάμερα,
- Εμφάνιση των στοιχείων του εικονικού πειράματος σε επαυξημένη πραγματικότητα,
- Αναγνώριση και εκτέλεση των χειρονομιών του χρήστη με την χρήση κάμερας,
- Παροχή οπτικής ανατροφοδότησης,

- Βοήθεια προς τον χρήστη με χρήση γραπτών οδηγιών.

3.4.2 Μη Λειτουργικές Προδιαγραφές

Οι μη λειτουργικές προδιαγραφές διαμορφώνονται βάσει τεχνικών, παιδαγωγικών και εμπειρικών παραμέτρων και αποτελούν τα παρακάτω:

- Ένα περιβάλλον φιλικό προς τον χρήστη, με απλά μηνύματα και οδηγίες ώστε να γίνεται εύκολα κατανοητό,
- Εύκολη εγκατάσταση χωρίς να χρειάζεται περαιτέρω εξοπλισμός,
- Όσο το δυνατόν πιο ομαλή απόδοση στις λειτουργίες,
- Το πείραμα να αντιστοιχεί στο παιδαγωγικό προφίλ μαθητών γυμνασίου,
- Να είναι συμβατό με συσκευές android που υποστηρίζουν ARCore και έχουν κάμερα RGB,
- Να είναι επεκτάσιμο το πείραμα, ώστε να μπορεί να εμπλουτίζεται με περαιτέρω λειτουργίες.

3.5 Τεχνικές προδιαγραφές

Ο τρόπος λειτουργίας της εφαρμογής βασίζεται σε ένα σύνολο τεχνολογιών που απαιτούν τη δημιουργία συγκεκριμένων απαιτήσεων για τις συσκευές και το περιβάλλον, προκειμένου να παρέχεται στον χρήστη μια εμπειρία χωρίς διακοπές. Ταυτόχρονα, λαμβάνονται υπόψη οι περιορισμοί που προκύπτουν από τις αποφάσεις σχετικά με την τεχνολογία, τον τρόπο χρήσης της και το συγκεκριμένο δημογραφικό προφίλ.

3.5.1 Βασικές Προδιαγραφές

Για τη σωστή εκτέλεση της εφαρμογής απαιτείται η τήρηση των παρακάτω βασικών προδιαγραφών:

- Λειτουργικό σύστημα Android 7 (API Level 24) ή νεότερο,
- Κάμερα RGB με τουλάχιστον 720p για την ανίχνευση χειρονομιών,
- Συμβατή συσκευή με ARCore,
- Χώρο αποθήκευσης το ελάχιστο 300MB,
- Επεξεργαστή τετραπύρηνο με αρχιτεκτονική ARM64,
- Διαθέσιμη RAM το ελάχιστο 4GB.

3.5.2 Λογισμικά και βιβλιοθήκες

Η εφαρμογή έχει αναπτυχθεί με την χρήση:

- **Unity 2022.3.62f3** με εγκατεστημένα τα πακέτα AR Foundation, ARCore και XR Plugin
- **Manomotion SDK(Unity Asset Store)** μαζί με το απαραίτητο πακέτο **TextMeshPro**

3.5.3 Περιορισμοί

Η εφαρμογή αναπτύχθηκε με βασική προτεραιότητα στη σχεδίαση την ευχρηστία και την αποτελεσματικότητα. Υπάρχουν οι εξής περιορισμοί που πρέπει να ληφθούν υπόψη κατά την χρήση:

- **Φωτισμός:** Το ιδανικό σενάριο είναι να υπάρχει ομοιόμορφος φωτισμός κατά την διάρκεια της χρήσης της εφαρμογής, καθώς το φως επηρεάζει την ποιότητα της αναγνώρισης χειρονομιών σε μεγάλο βαθμό.
- **Ορατότητα χειρονομιών:** Οι χειρονομίες πρέπει να γίνονται μέσα στο οπτικό πεδίο της κάμερας με σαφήνεια για να μπορούν να εντοπιστούν σωστά κατά την διάρκεια του πειράματος.
- **Θέση κάμερας:** Η κάμερα πρέπει να βρίσκεται σε σωστή απόσταση από το βιβλίο και από το χέρι του χρήστη ώστε τα ψηφιακά αντικείμενα και οι χειρονομίες να λειτουργούν αποτελεσματικά.

- **Ασταθής κινήσεις:** Εάν ο χρήστης μετακινεί με αστάθεια το τηλέφωνο η εφαρμογή δεν μπορεί να λάβει αξιόπιστα δεδομένα για να εκτελέσει την αναγνώριση χειρονομιών σωστά.

3.6 Επίλογος

Αυτό το κεφάλαιο παρέχει μια αναλυτική παρουσίαση του σχεδιασμού και της αρχιτεκτονικής του συστήματος επαυξημένης πραγματικότητας και της αναγνώρισης χειρονομιών που αναπτύχθηκε στο πλαίσιο αυτής της διατριβής. Μέσω μιας συστηματικής διαδικασίας, εξηγούνται η αρχιτεκτονική διάταξη, η λειτουργικότητα και τα διακριτά υποσυστήματα της εφαρμογής με σκοπό τη δημιουργία ενός λογικού, επεκτάσιμου και παιδαγωγικά κατάλληλου εκπαιδευτικού εργαλείου. Στο πρώτο στάδιο, περιγράφηκε η γενική δομή του συστήματος και αναλύθηκαν αναλυτικά τα υποσυστήματα στα οποία βασίζεται, δηλαδή το υποσύστημα επαυξημένης πραγματικότητας, το υποσύστημα αναγνώρισης χειρονομιών που υποστηρίχθηκε από το ManoMotion SDK, το υποσύστημα εκπαιδευτικού περιεχομένου και το επίπεδο αλληλεπίδρασης χρήστη. Αυτή η ανάλυση έδωσε μια σαφή κατανόηση των συγκεκριμένων ρόλων κάθε συστατικής μονάδας και των αμοιβαίων λειτουργιών τους.

Στη συνέχεια, πραγματοποιήθηκε μια πιο προσεκτική εξέταση της διαμόρφωσης της αλληλεπίδρασης χρήστη-συστήματος, και η φυσικότητα και η διαισθητικότητα της εμπειρίας του χρήστη έγιναν το επίκεντρο. Η μεροληπτική επιλογή της αλληλεπίδρασης με βάση τις χειρονομίες, καθώς και η κατάργηση των παραδοσιακών μέσων εισαγωγής δεδομένων, αντιστοιχεί στις παραδοχές της Αλληλεπίδρασης Ανθρώπου-Υπολογιστή και της ενσωματωμένης γνώσης, που υποστηρίζει τη μεθοδολογία της εμπειρίας που σχετίζεται με τη διδασκαλία των χημικών πειραμάτων.

Επιπλέον, οι λειτουργικές και μη λειτουργικές απαιτήσεις της εφαρμογής περιγράφηκαν λεπτομερώς με σαφή όρια και στόχους υλοποίησης του συστήματος. Οι προδιαγραφές καθορίστηκαν με βάση τις τεχνικές απαιτήσεις των εν λόγω τεχνολογιών (ARCore, Unity, ManoMotion) καθώς και τις εκπαιδευτικές ανάγκες των μαθητών της δευτεροβάθμιας εκπαίδευσης.

Τέλος, παρουσιάστηκαν οι τεχνικές προδιαγραφές, τα πακέτα λογισμικού και οι βιβλιοθήκες που χρησιμοποιήθηκαν, καθώς και οι βασικοί περιορισμοί του συστήματος. Η αναγνώριση των περιορισμών, συμπεριλαμβανομένης της κατάστασης του φωτισμού και των πιο αποτελεσματικών θέσεων γωνίας λήψης, αποτελεί κρίσιμο στοιχείο ενός υπεύθυνου σχεδιασμού και παρέχει το πλαίσιο για μελλοντικές βελτιώσεις και εξελίξεις της εφαρμογής.

Σε γενικές γραμμές, το κεφάλαιο αυτό παρέχει ένα σαφές και καλά τεκμηριωμένο πλαίσιο σχεδιασμού που γεφυρώνει το χάσμα μεταξύ θεωρίας και πράξης. Το επόμενο κεφάλαιο θα παρέχει μια ολοκληρωμένη ανασκόπηση της πρακτικής χρήσης της εφαρμογής, εξηγώντας την τεχνολογική επιλογή, τη διαδικασία λειτουργίας και τα αποτελέσματα της εργασίας του συστήματος.

4 Ανάλυση και Επεξήγηση της Εφαρμογής

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιαστεί αναλυτικά ο τρόπος με τον οποίο αναπτύχθηκε η εφαρμογή, αλλά και ο κώδικας που χρησιμοποιήθηκε ώστε η εφαρμογή να γίνει λειτουργική. Παρακάτω αναφέρονται τα εργαλεία και οι βιβλιοθήκες που χρησιμοποιήθηκαν καθώς και τα τρισδιάστατα μοντέλα που απαρτίζουν το ψηφιακό κομμάτι της επαυξημένης πραγματικότητας.

Η εφαρμογή υλοποιήθηκε με την μηχανή και το περιβάλλον του Unity, ένα εύχρηστο εργαλείο για την ανάπτυξη εφαρμογών επαυξημένης πραγματικότητας όλων των τύπων. Στην συγκεκριμένη εφαρμογή το Unity χρησιμοποιήθηκε για android συσκευές.

Επιπρόσθετα, οι βιβλιοθήκες AR Foundation που επιτρέπει μια πληθώρα από λειτουργίες AR, και η Manomotion SDK που προσθέτει εξαιρετική ευχρηστία στην υλοποίηση της αναγνώρισης χειρονομιών αποτέλεσαν κύριοι πυλώνες στην ανάπτυξη της εφαρμογής.

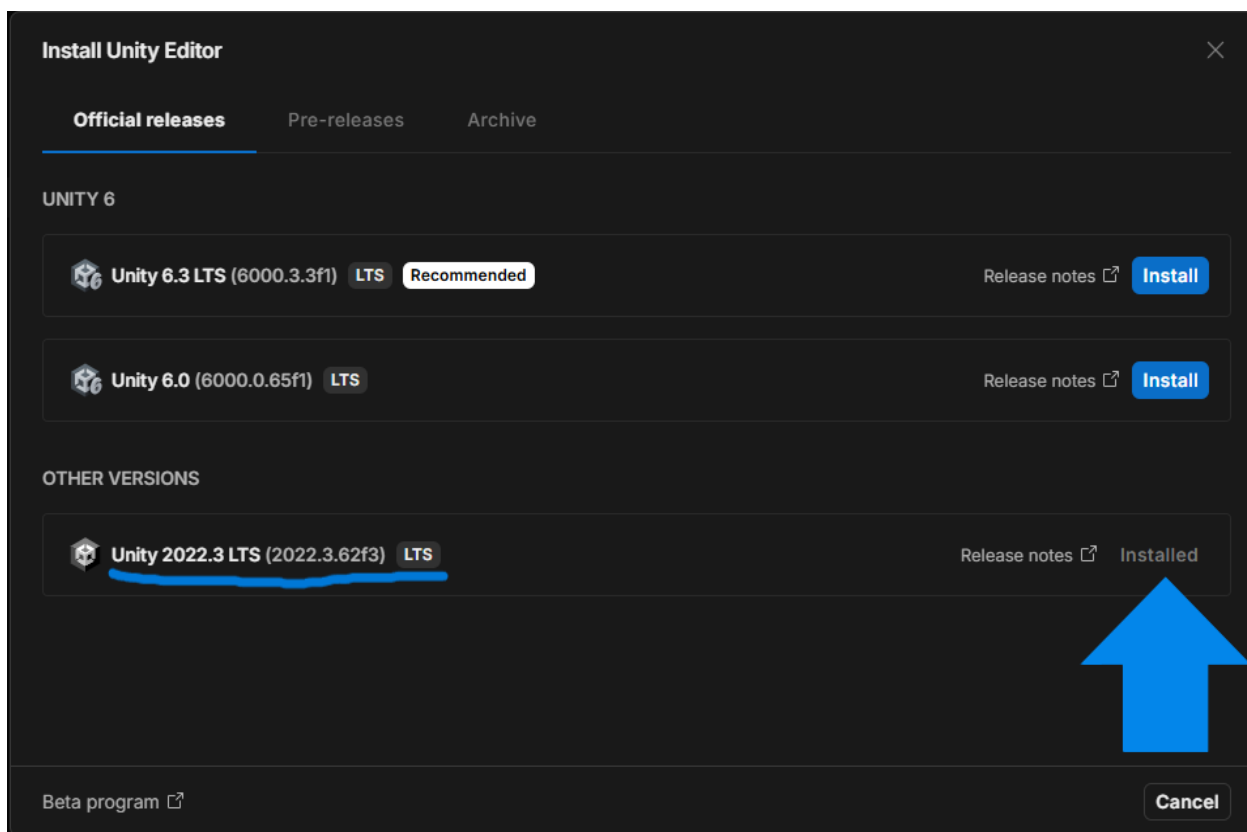
Η εφαρμογή ξεκίνησε να υλοποιείται σε στάδια, με το πρώτο να είναι η δημιουργία μιας κενής σκηνής και στην ενσωμάτωση των απαραίτητων βιβλιοθηκών, μετά ακολούθησε η δημιουργία της διεπαφής του χρήστη και της εικονικής σκηνής, με τελευταίο βήμα να είναι η δημιουργία του απαραίτητου κώδικα ώστε κάθε πτυχή της εφαρμογής, από την διεπαφή του χρήστη έως την αναγνώριση χειρονομιών να είναι λειτουργική.

4.2 Περιβάλλον Ανάπτυξης

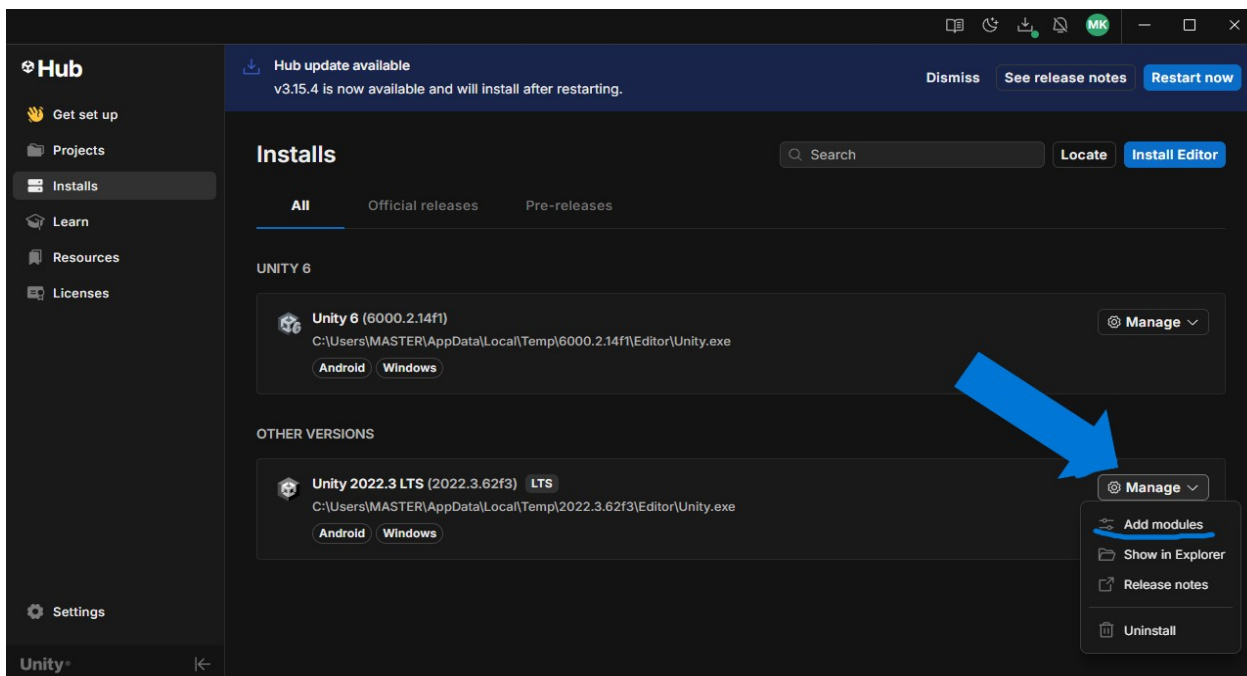
Η εφαρμογή δημιουργήθηκε στο περιβάλλον Unity 2022.3.62F3, το οποίο δίνει την δυνατότητα της ευρείας χρήσης του AR Foundation για εφαρμογές επαυξημένης πραγματικότητας. Το editor 2022.3.62f3 επιτρέπει την εύκολη προσθήκη διαφόρων βιβλιοθηκών χρησιμοποιώντας το package manager και παραλλήλα δίνει μεγάλη ευκολία και ευελιξία για την υλοποίηση εικονικών σκηνών σε συσκευές android.

4.2.1 Χρήση και ρύθμιση του Unity για την εφαρμογή

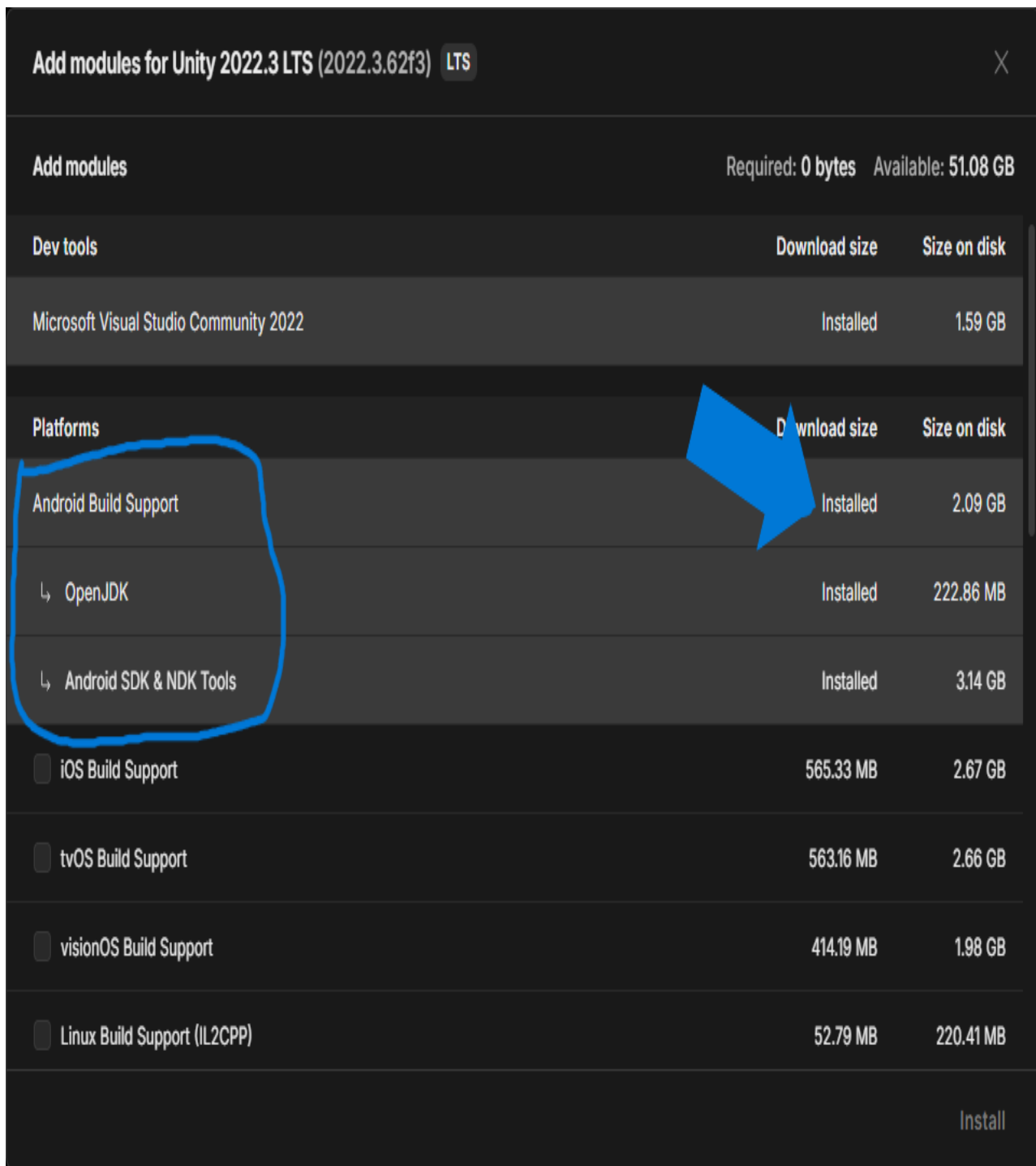
Το πρώτο βήμα στην υλοποίηση της εφαρμογής ήταν η επιλογή editor και συγκεκριμένα το Unity 2022.3.62f3 LTS, όπως φαίνεται στην εικόνα 4.1, μαζί με τις απαραίτητες προεκτάσεις του Android build support όπως είναι το OpenJDK και τα Android SDK & NDK Tools που παρατηρούνται στις εικόνες 4.2 και 4.3.



Εικόνα 4.1 Εγκατάσταση Editor

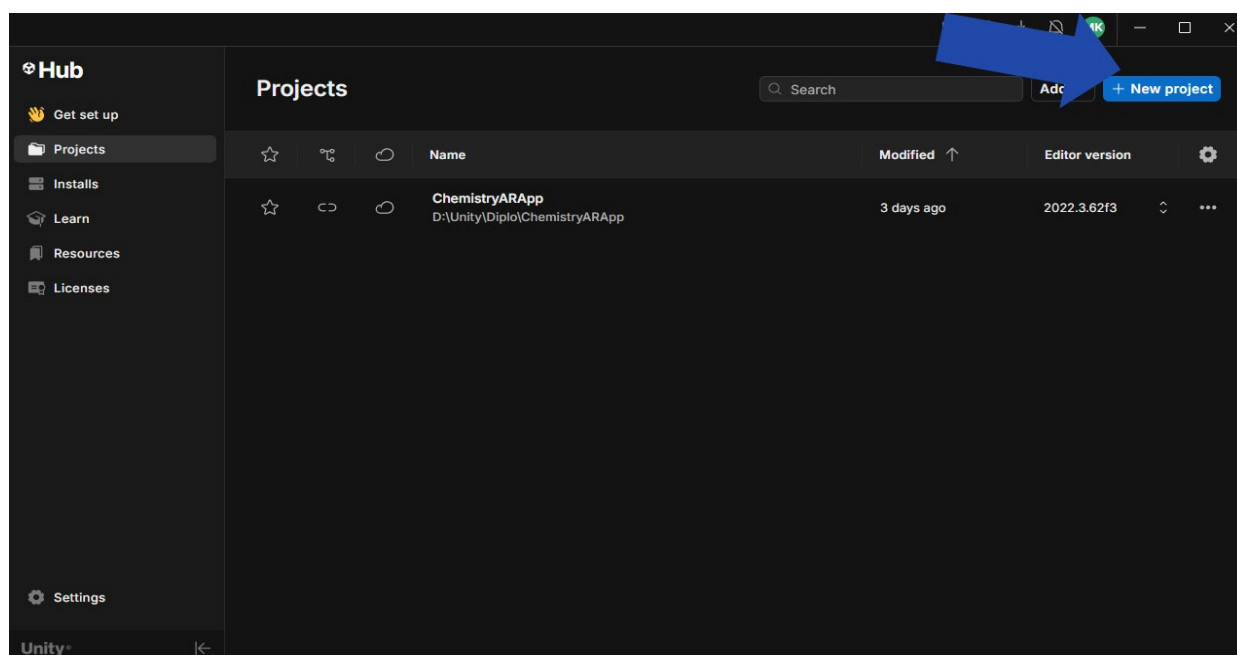


Εικόνα 4.2 Προσθήκη Προεκτάσεων

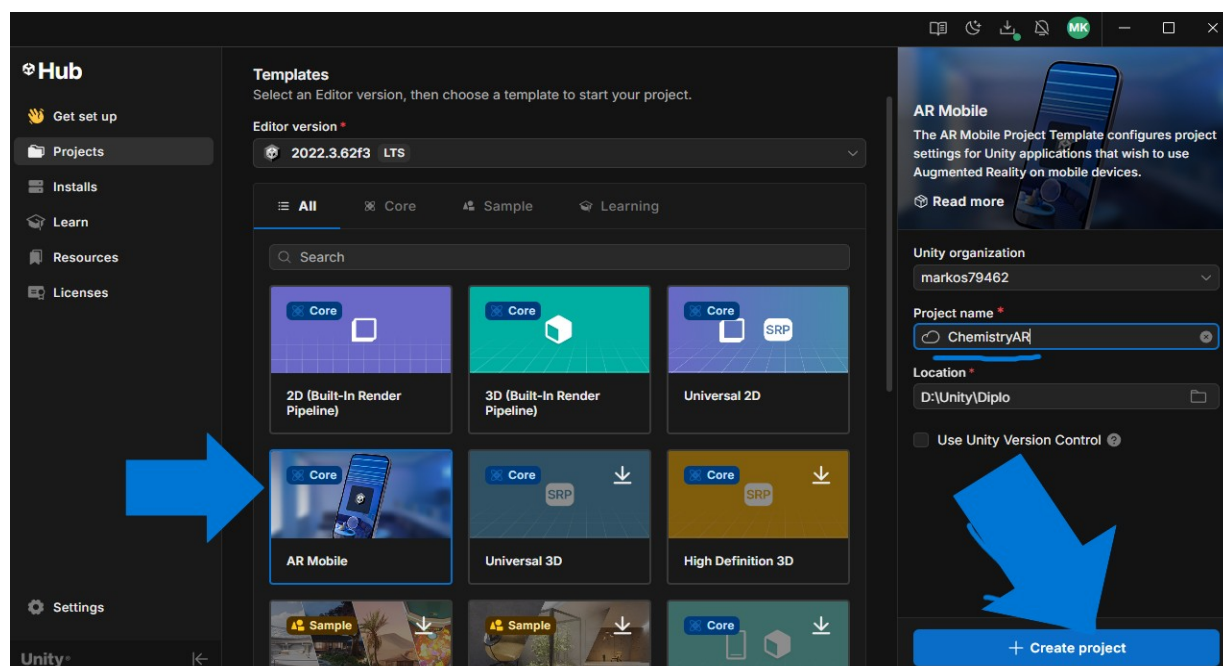


Εικόνα 4.3 Εγκατάσταση Προεκτάσεων

Στην συνέχεια, ξεκινά το έργο από το Unity Hub πατώντας το κουμπί New Project όπως στην εικόνα 4.4 και επιλέγοντας το AR Mobile template, το οποίο έχει ήδη προεγκατεστημένα ορισμένα πακέτα επαυξημένης πραγματικότητας τα οποία παρατηρούνται στην εικόνα 4.5.



Εικόνα 4.4 Δημιουργία Εφαρμογής

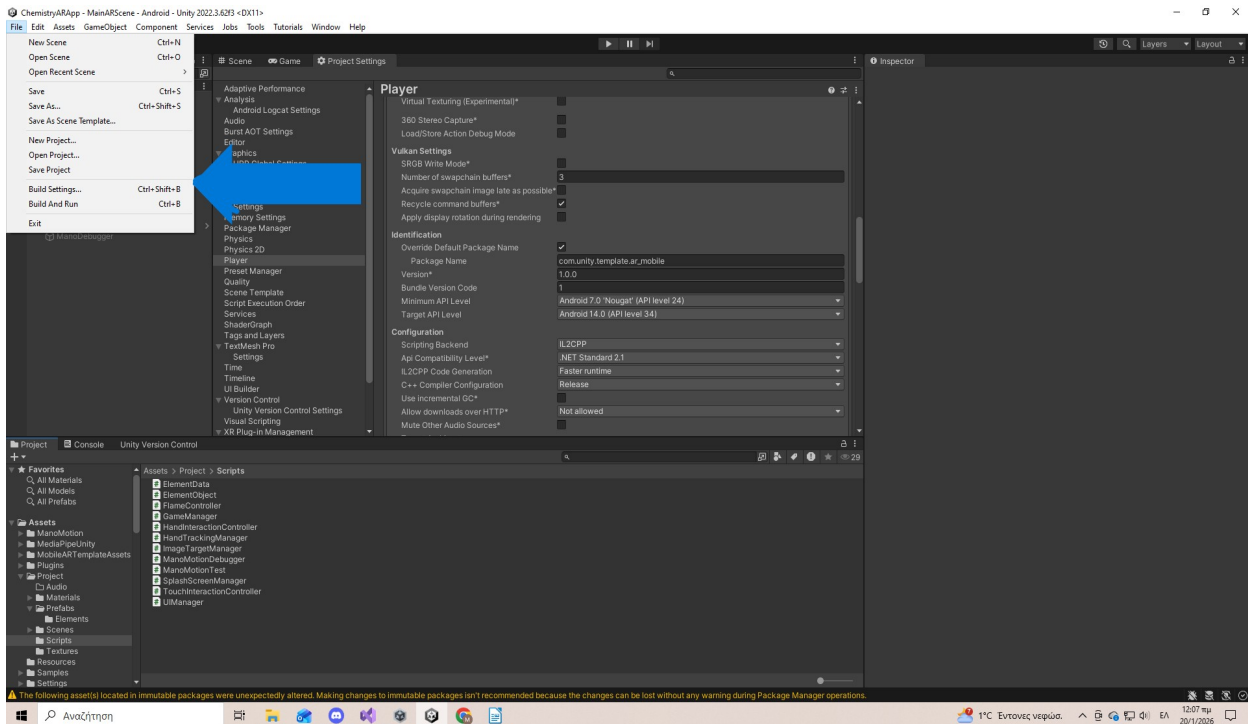


Εικόνα 4.5 Επιλογή Template και Αρχικοποίηση Εφαρμογής

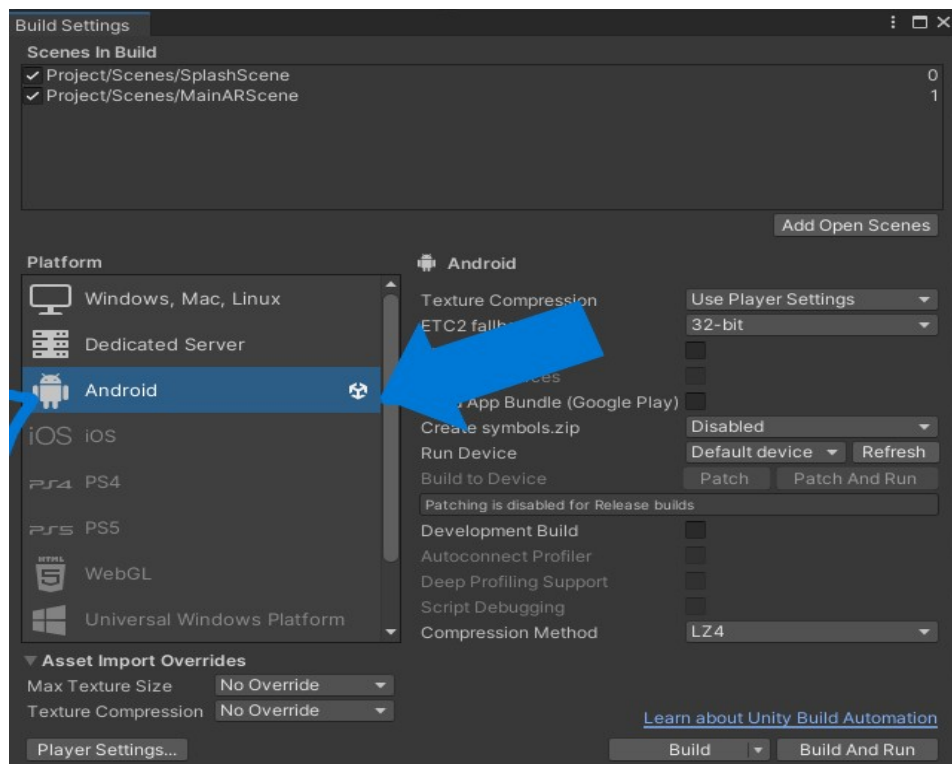
Με την πρώτη έναρξη του περιβάλλοντος του Unity, έπρεπε να γίνουν κάποιες ρυθμίσεις ώστε η εφαρμογή να παραχθεί και να λειτουργεί ομαλά. Οι εικόνες 4.6 και 4.7 δείχνουν με ακρίβεια τα βήματα που πρέπει να

Κεφάλαιο 4

γίνονται. Στις ρυθμίσεις μέσω του File > Build Settings γίνονται οι πρώτες αλλαγές όπου το Build Target αλλάζει και επιλέγεται η πλατφόρμα android.

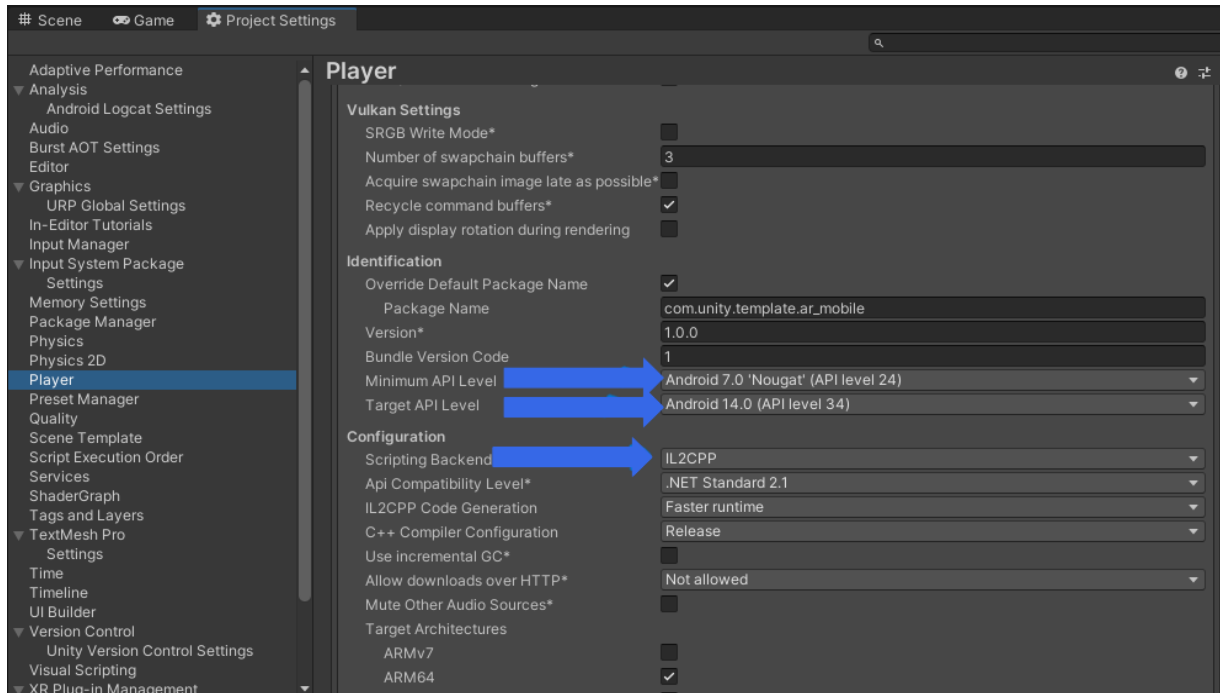


Εικόνα 4.6 File > Build Settings

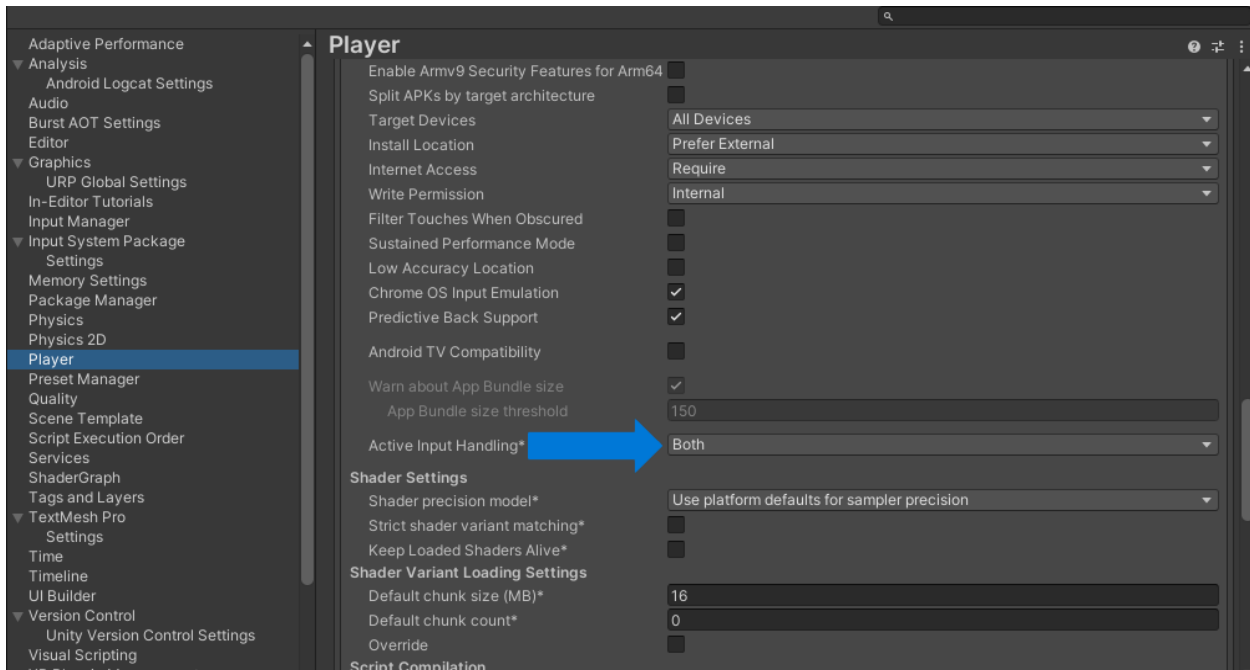


Εικόνα 4.7 Επιλογή Android ως Build Setting

Το επόμενο βήμα ήταν η αλλαγή ορισμένων ρυθμίσεων στην κατηγορία Player Settings, με τις αλλαγές να είναι το ελάχιστο επίπεδο android να είναι 7.0 με τον στόχο να είναι το 14.0 επίπεδο. Μια ακόμη αλλαγή ήταν η επιλογή Scripting Backend IL2CPP και η ενεργοποίηση της αρχιτεκτονικής ARM64, η οποία απαιτείται από το Google Play. Τελευταία αλλαγή ήταν η επιλογή του active input handling από new σε both ώστε να είναι συμβατό με τον κώδικα της εφαρμογής. Στις παρακάτω εικόνες 4.8 και 4.9 παρατηρούνται αναλυτικά τα παραπάνω βήματα.



Εικόνα 4.8 Επιλογή ελαχίστου API, στόχου API και Scripting Backend IL2CPP

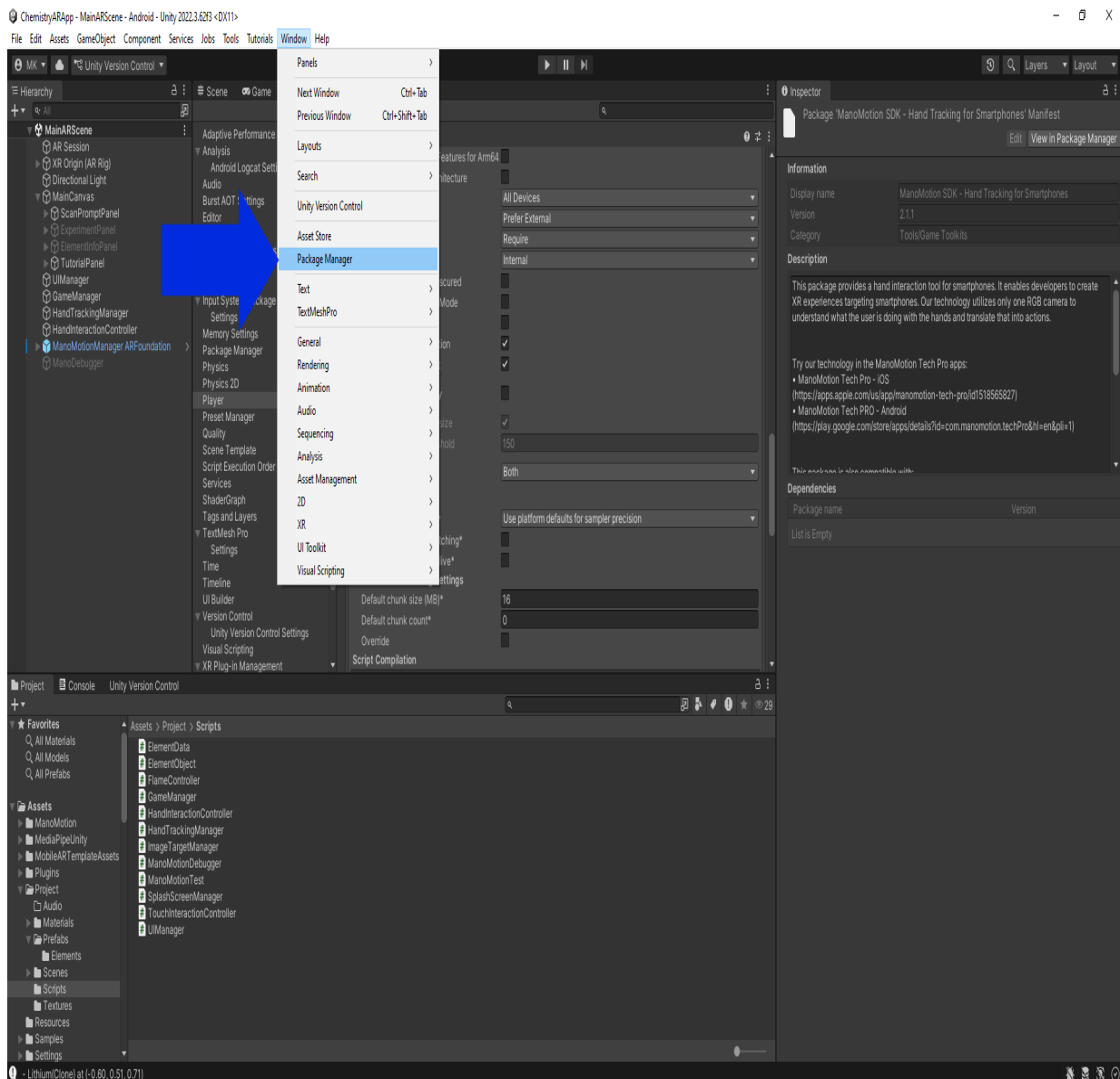


Εικόνα 4.9 Επιλογή active input handling

Κεφάλαιο 4

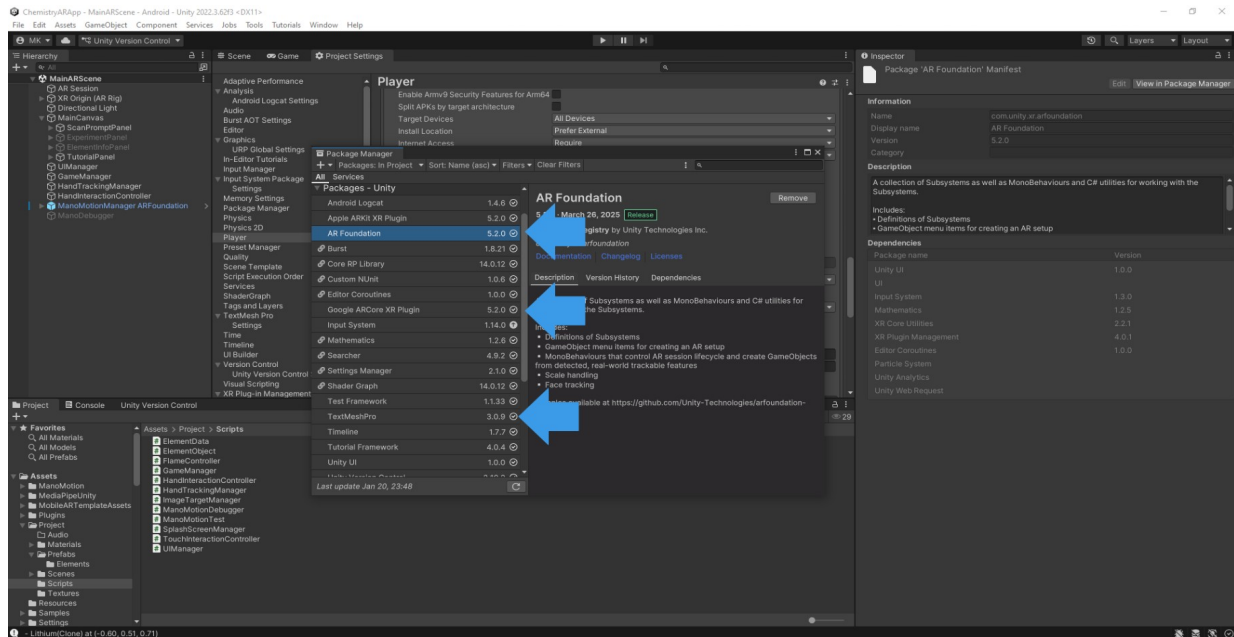
Επόμενο βήμα ήταν η εγκατάσταση των απαραίτητων πακέτων από το Package Manager. Στις εικόνες 4.10, 4.11, 4.12 και 4.13 εμφανίζονται τα βήματα εγκατάστασης με ακρίβεια. Παρακάτω ακολουθούν τα πακέτα που εγκαταστάθηκαν για την εφαρμογή, τα οποία είναι:

- AR Foundation,
- Google ARCore XR Plugin,
- XR Plugin Management,
- XR Interaction Toolkit,
- TextMeshPro,
- ManoMotion SDK (Unity Asset Store).

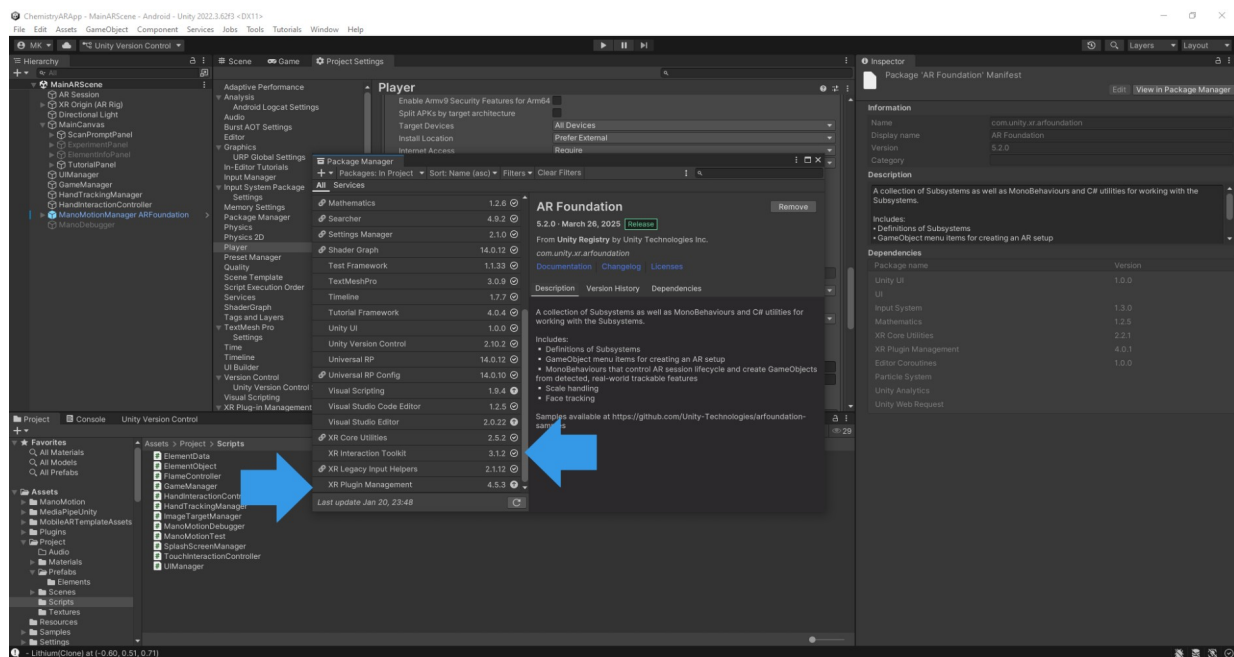


Εικόνα 4.10 Window > Package Manager

ΑΝΑΛΥΣΗ ΚΑΙ ΕΠΕΞΗΓΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

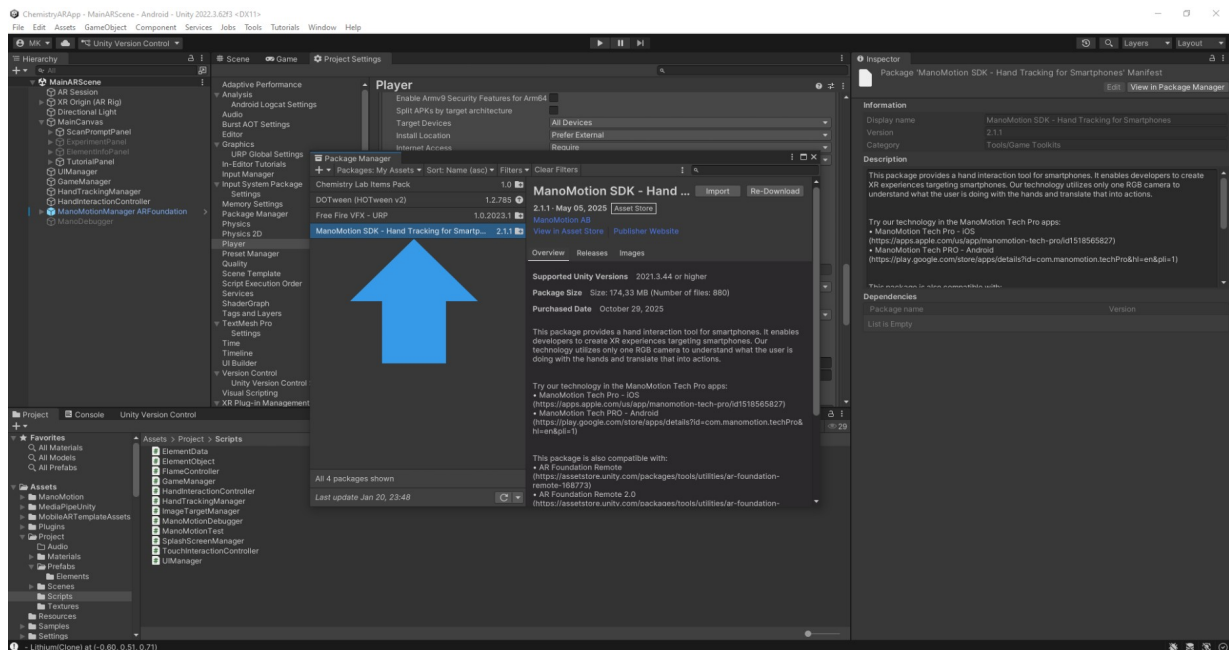


Εικόνα 4.11 AR Foundation, Google ARCore XR Plugin, TextMeshPro



Εικόνα 4.12 XR Plugin Management και XR Interaction Toolkit

Κεφάλαιο 4



Εικόνα 4.13 ManoMotion SDK

4.3 Διεπαφή Χρήστη

4.3.1 Splash Screen Διεπαφή

Η οθόνη εκκίνησης είναι η πρώτη εντύπωση που έχει ο χρήστης για την εφαρμογή. Εκπληρώνει μια σειρά από κρίσιμους στόχους στο σύντομο χρονικό διάστημα της ύπαρξής της:

1. Ενίσχυση της επωνυμίας: Παρουσίαση του λογότυπου και του ονόματος της επωνυμίας, που δημιουργεί οπτική αναγνωρισιμότητα και τυποποίηση της επωνυμίας.
2. Ένδειξη φόρτωσης: Η ένδειξη φόρτωσης χρησιμοποιείται για να δείξει ότι υπάρχει πρόοδος καθώς η εφαρμογή φορτώνει υποσυστήματα AR.
3. Ρύθμιση προσδοκιών χρήστη: Προετοιμάζει τους χρήστες για το τι αναμένεται να συμβεί μέσω της οπτικής γλώσσας του σχεδιασμού.
4. Τεχνική αναγκαιότητα: Δίνει χρόνο για την αρχικοποίηση των στοιχείων του AR Foundation και του ManoMotion SDK.

Η διεπαφή χρήστη αποτελείται από την native πλατφόρμα Unity UI (Unity UI / uGUI) με τη χρήση του TextMeshPro για την απόκτηση υψηλής ποιότητας διεπαφή χρήστη. Τα οφέλη αυτής της ενσωμάτωσης είναι τα εξής:

- Σχέδιο βασισμένο σε καμβά που είναι ενσωματωμένο με υλικό AR.
- Κλιμάκωση UI ανεξάρτητα από την ανάλυση των οθονών των συσκευών.
- Πλούσια μορφοποίηση κειμένου, η οποία είναι δυνατή χάρη στις ειδικές λειτουργίες τυπογραφίας που προσφέρει το TextMeshPro.
- Διείκδυση των κλήσεων σχεδίασης UI σε παρτίδες για μεγιστοποίηση της χρήσης του.
- Ενσωματωμένη υποστήριξη των κινούμενων εικόνων και των μεταβάσεων UI.

Η οθόνη εκκίνησης παρουσιάζει μια προσεκτικά σχεδιασμένη ακολουθία κινούμενων εικόνων που δημιουργεί μια αρκετά επαγγελματική πρώτη εντύπωση, η οποία αποτελείται από τρεις φάσεις:

1. Φάση 1: Εμφάνιση (0,0 s – 0,5 s)

Όλες οι ομάδες καμβά μετακινούνται σε πλήρη διαφάνεια (0) για να γίνουν πλήρως αδιαφανείς (1). Το αποτέλεσμα αυτής της σταδιακής εισαγωγής είναι η μείωση των ξαφνικών εισόδων στο σύστημα, η ορατότητα των στοιχείων της διεπαφής χρήστη και η δημιουργία μιας ομαλής μετάβασης της κατάστασης φόρτωσης του συστήματος.

2. Φάση 2: Πρόοδος φόρτωσης (0,5 δευτερόλεπτα – 3,5 δευτερόλεπτα)

Σε αυτό το χρονικό διάστημα, η γραμμή φόρτωσης μετακινείται από το 0 στο 100 τοις εκατό και το συνοδευτικό κείμενο αυξάνεται μέσω διαφόρων διαγνωστικών μηνυμάτων. Δεδομένου ότι η προσομοιωμένη πρόοδος παρέχει ομοιόμορφο χρονισμό, σε ένα περιβάλλον παραγωγής θα μπορούσε να συσχετιστεί με τους πραγματικούς δείκτες φόρτωσης στοιχείων.

Φόρτωση μηνυμάτων που εμφανίζονται σε σειρά:

1. "Initializing AR..." (0% - 20%)
2. "Loading Chemistry Data..." (20% - 40%)
3. "Preparing Elements..." (40% - 60%)
4. "Setting Up Experiments..." (60% - 80%)
5. "Almost Ready..." (80% - 99%)
6. "Ready!" (100%)

3. Φάση 3: Fade Out (3,5 δευτερόλεπτα – 4,0 δευτερόλεπτα)

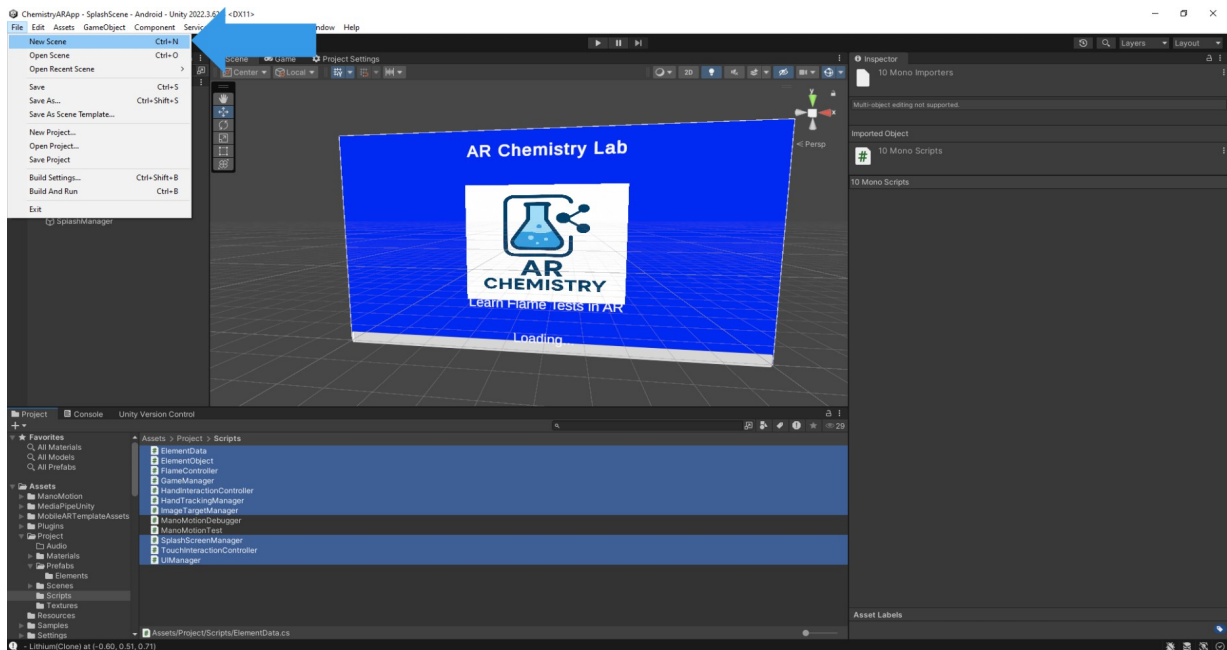
Αφού ολοκληρωθεί η φόρτωση και εμφανιστεί για λίγο το μήνυμα Ready! (Ετοιμο!), ο καμβάς γίνεται μαύρος μέσα σε 0,5 δευτερόλεπτα και στη συνέχεια ξεκινά η μετάβαση στην κύρια σκηνή επαυξημένης πραγματικότητας.

4.3.1.1 Δημιουργία Splash Screen

Ακολουθούν τα βήματα που απαιτούνται για τη ρύθμιση της σκηνής Splash Screen:

1. Πρέπει να ξεκινήσει η πρώτη σκηνή. Αυτό γίνεται ανοίγοντας το File-> New Scene, όπως στην εικόνα 4.14, και αποθηκευόντάς το με το όνομα SplashScreen.
2. Αφαιρούμε την προεπιλεγμένη κύρια κάμερα, καθώς η τελική διαμόρφωση βασίζεται αποκλειστικά στο UI.
3. Δημιουργούμε ένα Canvas όπου θα βρίσκονται όλα τα στοιχεία της διεπαφής χρήστη.
4. Το σενάριο SplashScreenManager πρέπει να συνδεθεί με ένα GameObject που δεν έχει δημιουργηθεί.

Κεφαλαίο 4



Εικόνα 4.14 Δημιουργία Σκηνής

4.3.1.2 Ρυθμίσεις στοιχείου Canvas

Δημιουργούμε και διαμορφώνουμε το κύριο Canvas με τις ακόλουθες ρυθμίσεις:

- Render Mode: Screen Space – Overlay
- Pixel Perfect: Enabled (for crisp UI on all devices)
- Sort Order: 0

Ρυθμίσεις στοιχείου Canvas Scaler:

- UI Scale Mode: Scale With Screen Size
- Reference Resolution: 1080 x 1920 (portrait mobile)
- Screen Match Mode: Match Width Or Height
- Match: 0.5 (balanced scaling)

Στοιχείο γραφικού Raycaster:

- Ignore Reversed Graphics: Enabled
- Blocking Objects: None

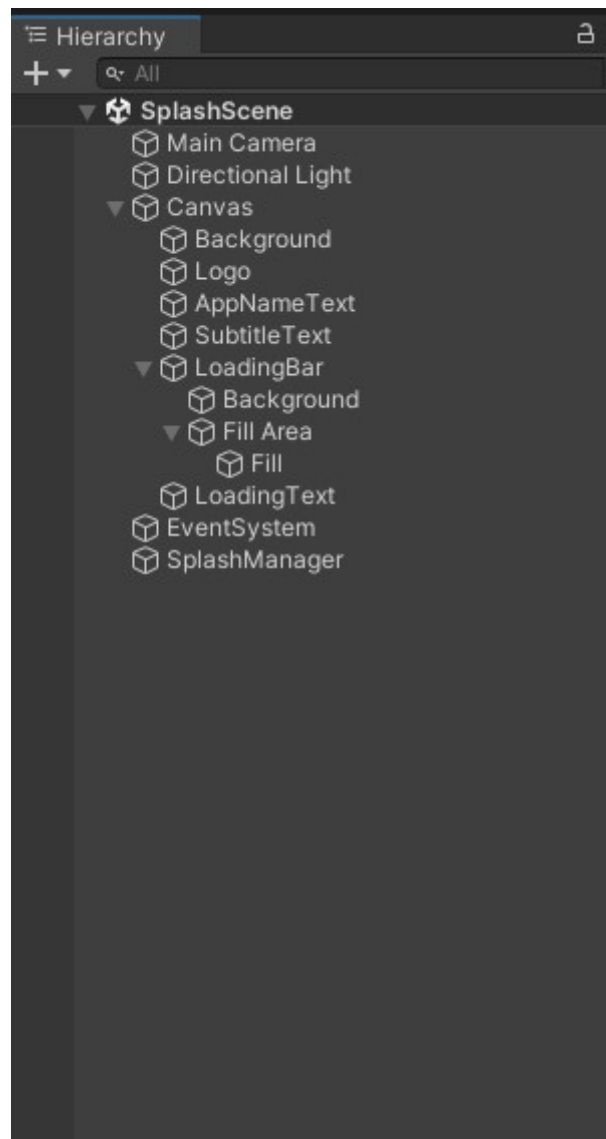
Προσθέτουμε ένα **στοιχείο CanvasGroup** στο **Canvas** για εφέ εξασθένισης:

- Alpha: 1 (will be controlled by script)
- Interactable: Enabled
- Blocks Raycasts: Enabled
- Ignore Parent Groups: Disabled

Το CanvasGroup διευκολύνει τη δυνατότητα του διαχειριστή της οθόνης εκκίνησης να μεταβάλλει τη διαφάνεια ολόκληρου του περιβάλλοντος εργασίας χρήστη, αλλάζοντας συλλογικά την παράμετρο άλφα, αντί να απαιτεί την ταυτόχρονη κίνηση των διαφόρων στοιχείων του περιβάλλοντος εργασίας.

4.3.1.3 Ιεραρχία Σκηνής SplashScreen

Παρακάτω στην εικόνα 4.15 εμφανίζεται η πλήρης λίστα και ιεραρχία των στοιχείων που απαρτίζουν το σκηνικό του splash screen.



Εικόνα 4.15 Ιεραρχία Σκηνής SplashScreen

Στην συνέχεια ακολουθούν οι ρυθμίσεις κάθε στοιχείου της σκηνής του splash screen. Οι ρυθμίσεις αυτές είναι:

1. Ρυθμίσεις Background

RectTransform:

- Anchor: Stretch-Stretch
- Left, Right, Top, Bottom: 0

Image Component:

- Color: Dark blue gradient (#002BF1)
- Image Type: Sliced or Simple
- Raycast Target: Disabled

2. Ρυθμίσεις Logo

RectTransform:

- Anchor: Top-Center
- Pivot: (0.5, 0.5)
- Pos Y: -300 (from top)
- Width: 250, Height: 250

Image Component:

- Source Image: Logo
- Preserve Aspect: Enabled
- Raycast Target: Disabled

3. Ρυθμίσεις AppNameText

RectTransform:

- Anchor: Top-Center
- Pos Y: -580 (below logo)
- Width: 800, Height: 100

TextMeshPro:

- Text: "AR Chemistry Lab"
- Font Size: 56
- Font Style: Bold
- Alignment: Center, Middle
- Color: White (#FFFFFF)
- Enable Auto Size: Optional (Min: 40, Max: 56)

4. Ρυθμίσεις LoadingBar

RectTransform:

- Anchor: Bottom-Center
- Pos Y: 300 (from bottom)

- Width: 600, Height: 20

Slider Component:

- Interactable: Disabled (controlled by script only)
- Transition: None
- Direction: Left To Right
- Min Value: 0
- Max Value: 1
- Value: 0 (starting point)

Child Objects Configuration:

- Background: Light gray color (#4A5568), rounded sprite
- Fill Area → Fill: Accent color (#4299E1), rounded sprite

5. Ρυθμίσεις LoadingText

RectTransform:

- Anchor: Bottom-Center
- Pos Y: 220 (κάτω από το loading bar)
- Width: 600, Height: 50

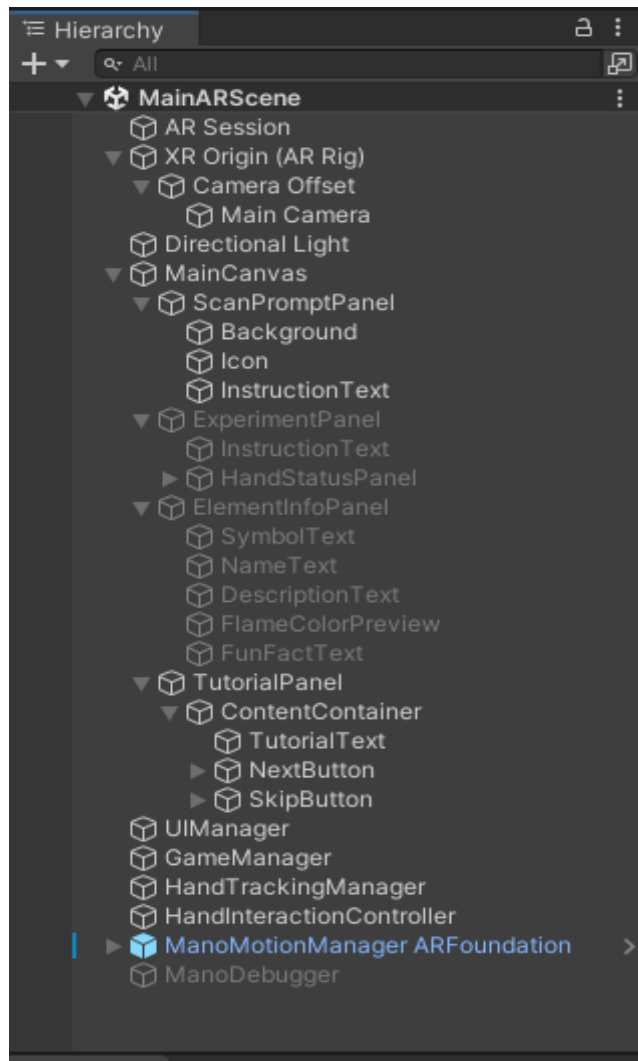
TextMeshPro:

- Text: "Initializing..." (αλλάζει με script)
- Font Size: 24
- Alignment: Center, Middle
- Color: Light gray (#A0AEC0)

4.3.2 Κεντρική Διεπαφή Επαυξημένης Πραγματικότητας

4.3.2.1 Ιεραρχία Σκηνής MainARScene

Παρακάτω στην εικόνα 4.16 εμφανίζεται η πλήρης λίστα και ιεραρχία των στοιχείων που απαρτίζουν το σκηνικό του main screen.



Εικόνα 4.16 Ιεραρχία Σκηνής MainARScene

4.3.2.2 Ρυθμίσεις στοιχείου MainCanvas

Το κύριο UI Canvas πρέπει να διαμορφωθεί για Screen Space - Overlay, ώστε να διασφαλίζεται ότι θα εμφανίζεται πάντα πάνω από το περιεχόμενο AR. Παρακάτω ακολουθούν οι ρυθμίσεις του MainCanvas, οι οποίες είναι:

Ρυθμίσεις Canvas:

- Render Mode: Screen Space - Overlay
- Pixel Perfect: Enabled
- Sort Order: 10 (above AR content)

Ρυθμίσεις Canvas Scaler:

- UI Scale Mode: Scale With Screen Size
- Reference Resolution: 1080 x 1920
- Match: 0.5

4.3.2.3 Ρυθμίσεις TutorialPanel

Το TutorialPanel είναι υπεύθυνο για τις οδηγίες χρήσης που εμφανίζονται στην αρχή για τον χρήστη, ώστε να μάθει ο χρήστης πως να χρησιμοποιήσει την εφαρμογή. Οι ρυθμίσεις για το TutorialPanel είναι:

Panel RectTransform:

- Anchor: Stretch-Stretch
- All offsets: 0 (fullscreen overlay)

Panel Image Component:

- Color: Black with 85% opacity (0, 0, 0, 0.85)
- Raycast Target: Enabled (blocks touches behind)

Tutorial Text (TextMeshPro):

- Anchor: Center
- Position: (0, 50, 0)
- Size: 800 x 400
- Font Size: 28
- Alignment: Center, Middle
- Color: White

Next Button:

- Anchor: Bottom-Center
- Position: (80, 150, 0)
- Size: 200 x 60
- Background Color: #3182CE (blue)
- Text: "Next", White, Bold, 24pt

Skip Button:

- Anchor: Bottom-Center
- Position: (-80, 150, 0)
- Size: 120 x 60
- Background: Transparent with border
- Text: "Skip", Gray (#718096), 20pt

4.3.2.4 Ρυθμίσεις ScanPromptPanel

Panel Configuration:

- Anchor: Bottom-Stretch
- Height: 200
- Background: Semi-transparent gradient

Scan Icon (Image):

- Anchor: Center-Top of panel
- Size: 80 x 80
- Source: Camera or scan icon sprite
- Note: Animated by script (rotation + pulse)

Scan Instruction Text:

- Text: "Point your camera at the Chemistry book cover"
- Font Size: 22
- Color: White
- Alignment: Center

4.3.2.5 Ρυθμίσεις ExperimentPanel

Αυτό το Panel εμφανίζεται μετά την ανίχνευση του δείκτη AR και περιέχει το περιβάλλον εργασίας χρήστη του τρέχοντος πειράματος. Οι ρυθμίσεις του ακολουθούν παρακάτω και είναι:

Instruction Text:

- Anchor: Bottom-Center
- Position: Near bottom of screen
- Text: "Pinch to grab an element and throw it into the flame!"
- Font Size: 20
- Background: Ημιδιαφανές Μαύρο

4.3.2.6 Ρυθμίσεις HandTrackingIndicator

Container RectTransform:

- Anchor: Top-Left
- Position: (20, -20)
- Size: 150 x 50

Hand Icon (Image):

- Size: 32 x 32
- Anchor: Left

- Color: Αλλάζει ανάλογα την κατάσταση μέσω script

Hand Status Text:

- Anchor: Δεξιά από την εικόνα
- Font Size: 18
- Text: "Show Hand" / "Hand Ready" / "Grabbing"

4.3.2.7 Ρυθμίσεις ElementInfoPanel

Αυτό το Panel εμφανίζεται όταν ένα στοιχείο έρχεται σε επαφή με τη φλόγα, εμφανίζοντας εκπαιδευτικό περιεχόμενο. Παρακάτω ακολουθούν οι ρυθμίσεις για τα στοιχεία του Panel:

Panel Configuration:

- Anchor: Bottom-Center
- Position: 100 units από bottom
- Size: 90% screen width, auto height
- Background: Λευκό με στρογγυλεμένες γωνίες, ελαφριά σκιά
- Initially: SetActive(false)

Element Name Text:

- Font Size: 32, Bold
- Color: Dark gray (#2D3748)
- Παράδειγμα: "Lithium"

Element Symbol Text:

- Font Size: 48, Bold
- Color: Ταιριάζει με το χρώμα της φλόγας του στοιχείου
- Παράδειγμα: "Li"

Flame Color Preview (Image):

- Size: 50 x 50
- Shape: Circle
- Color: Ορισμός του στοιχείου flameOuterColor

Description Text:

- Font Size: 18
- Color: Gray (#4A5568)
- Max Lines: 4

Fun Fact Text:

- Font Size: 16, Italic

- Background: Light highlight
- Prefix:"FunFact:"

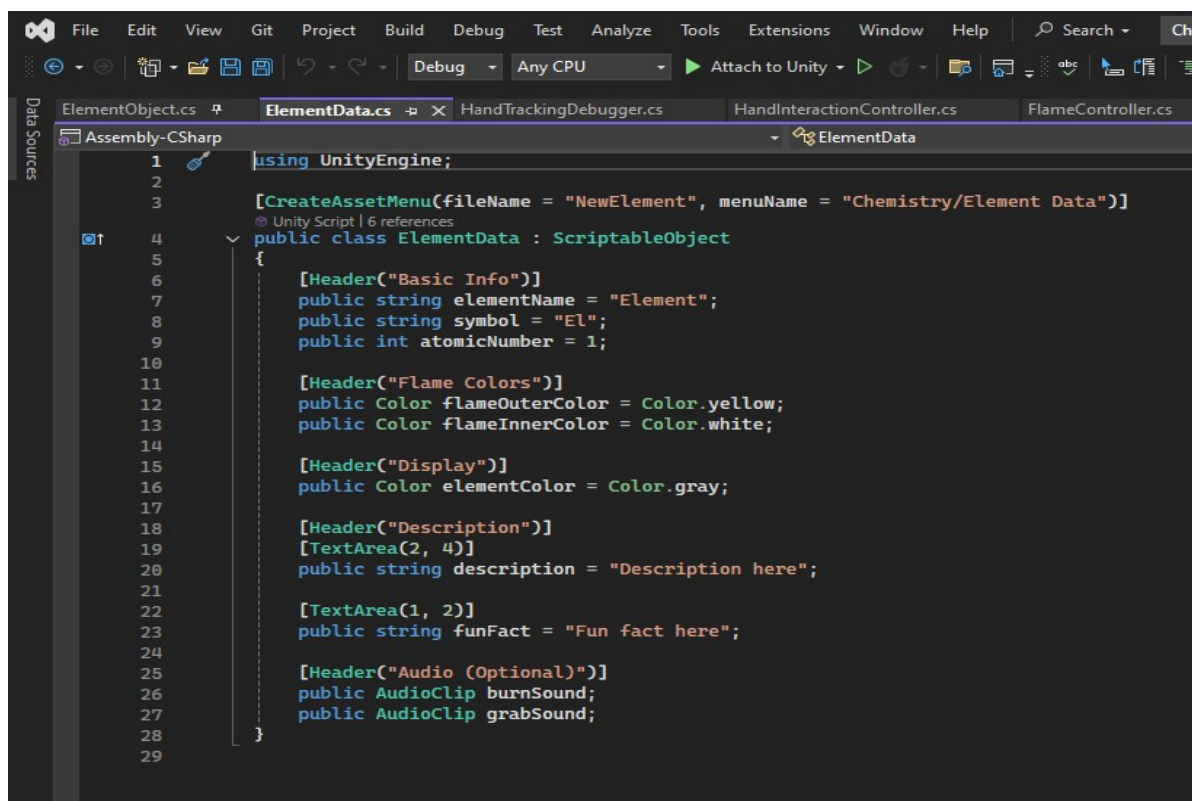
4.4 Επεξήγηση Κώδικα

4.4.1 Αρχείο ElementData.cs

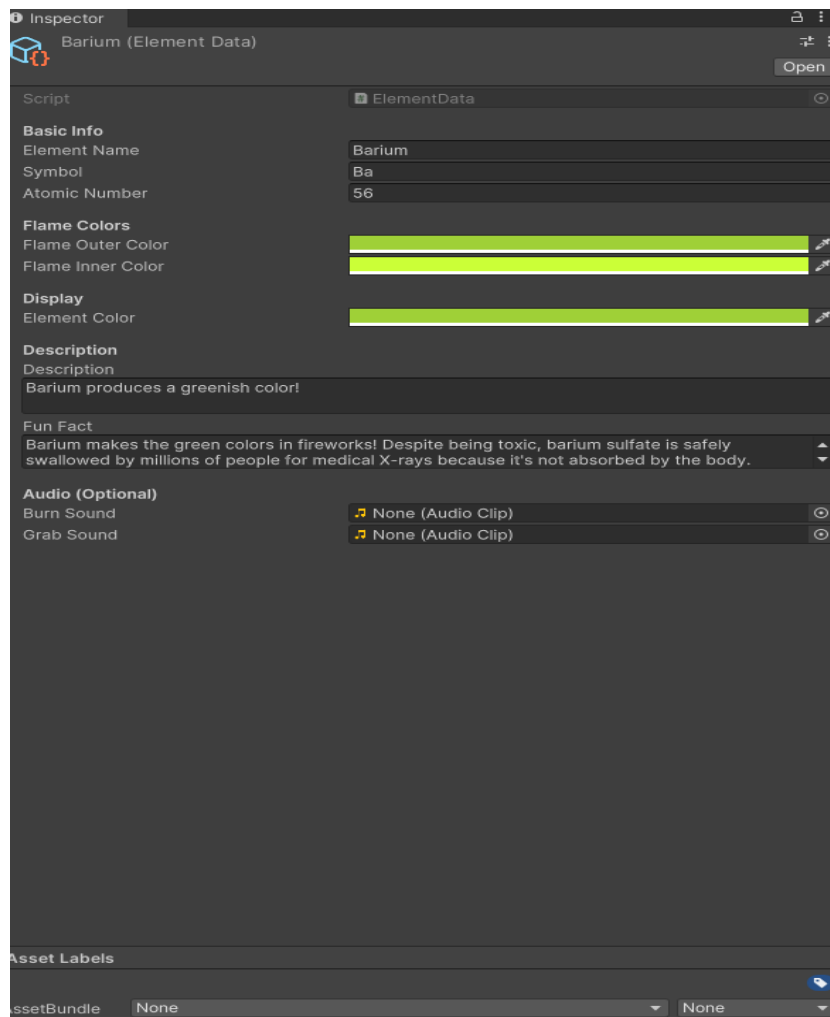
Το σύνολο δεδομένων κάθε χημικού στοιχείου βρίσκεται στην κλάση ElementData, η οποία είναι ένα ScriptableObject όπως φαίνεται στην εικόνα 4.17. Λειτουργεί επίσης ως ένας έγκυρος φορέας από τον οποίο άλλα υποσυστήματα, όπως η φλόγα, η διεπαφή χρήστη, τα υποσυστήματα ήχου, μεταξύ άλλων, λαμβάνουν τις απαραίτητες πληροφορίες οι οποίες φαίνονται στην εικόνα 4.18. Μέσω του ScriptableObject, η δομή επιτρέπει μια προσέγγιση που μπορεί να περιγραφεί ως μοντέλο βασισμένο σε δεδομένα, επιτρέποντας έτσι την υλοποίηση νέων στοιχείων χωρίς να γίνονται αλλαγές στον κώδικα. Τα πεδία που περιλαμβάνει είναι:

Πεδία (Fields):

- elementName, symbol, atomicNumber: βασική ταυτότητα στοιχείου.
- flameOuterColor, flameInnerColor: χρωματισμός φλόγας κατά το “flame test”.
- elementColor: χρώμα εμφάνισης του στοιχείου (προαιρετικό).
- description, funFact: εκπαιδευτικό κείμενο προς εμφάνιση στο UI.
- burnSound, grabSound: προαιρετικοί ήχοι ανατροφοδότησης.



Εικόνα 4.17 Κώδικας ElementData.cs



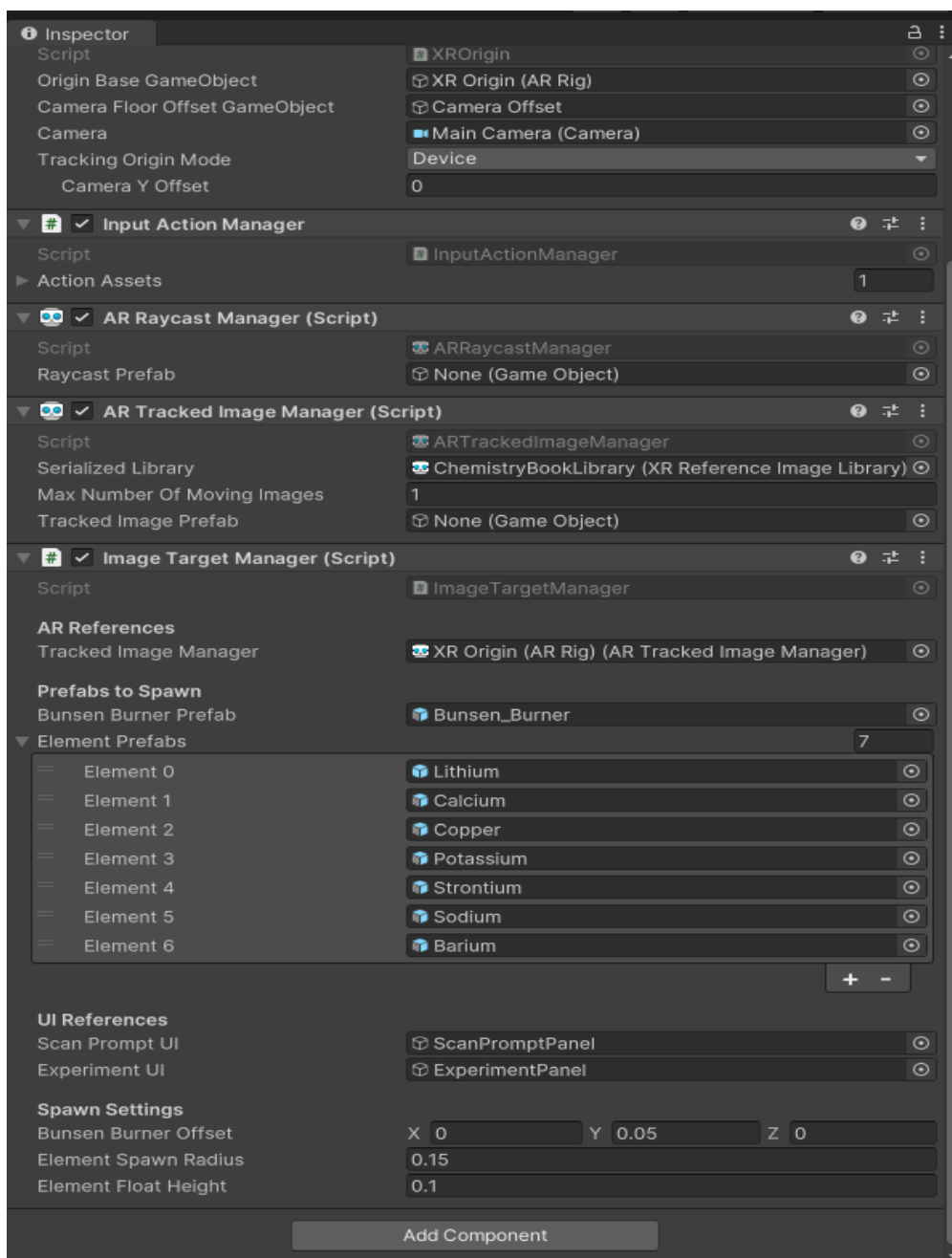
Εικόνα 4.18 Παράδειγμα στοιχείου

4.4.2 Αρχείο ImageTargetManager.cs

Το ImageTargetManager διαχειρίζεται την παρατήρηση και τη δημιουργία του βιβλίου Χημείας ως ελεγχόμενη εικόνα και τη δυναμική δημιουργία της σκηνής επανξημένης πραγματικότητας κατά την χρήση του. Όταν εντοπιστεί ο στόχος, το σύστημα θα τοποθετήσει τον καυστήρα και τα στοιχεία στη σωστή χωρική διάταξη, και όταν χαθεί ο στόχος, το σύστημα θα επανέλθει στην καθοδηγούμενη σάρωση. Τα βασικά του πεδία ακολουθούν, τα οποία βρίσκονται και στην εικόνα 4.19, και είναι τα εξής:

Βασικά πεδία/αντικείμενα:

- Αναφορά σε ARTrackedImageManager (AR Foundation).
- Prefabs: φλόγιστρο (bunsen burner), στοιχεία.
- UI αντικείμενα (scan prompt/experiment UI).
- Κατάσταση: isBookDetected, currentTrackedImage.
- “Root” αντικείμενο αλληλεπίδρασης: interactionRoot ώστε όλα τα spawned αντικείμενα να ακολουθούν τον στόχο.



Εικόνα 4.19 ImageTargetManager.cs

Μέθοδοι κύκλου ζωής Unity

Οι μέθοδοι του κύκλου ζωής του unity που φαίνονται και στην εικόνα 4.20 είναι οι παρακάτω:

Awake()

Αρχικοποιεί μεταβλητές/αναφορές και προετοιμάζει το υποσύστημα για τη λήψη συμβάντων.

OnEnable()/OnDisable()

Καταχωρεί/αποκαταχωρεί την κλήση επιστροφής στο συμβάν `trackedImagesChanged`. Αυτό είναι κρίσιμο για τη σωστή διαχείριση των πόρων και την αποφυγή «διπλών» κλήσεων επιστροφής.

Start()

Ορίζει την αρχική κατάσταση του περιβάλλοντος εργασίας χρήστη (π.χ. εμφάνιση προτροπής σάρωσης) και προετοιμάζει την εφαρμογή για ανίχνευση.

```

56 public Transform BunsenBurnerTransform => currentBunsenBurner?.transform;
57
58 @ Unity Message | 0 references
59 private void Awake()
60 {
61     if (Instance == null)
62     {
63         Instance = this;
64     }
65     else
66     {
67         Destroy(gameObject);
68         return;
69     }
70     if (trackedImageManager == null)
71     {
72         trackedImageManager = GetComponent<ARTrackedImageManager>();
73     }
74 }
75
76 @ Unity Message | 0 references
77 private void OnEnable()
78 {
79     if (trackedImageManager != null)
80     {
81         trackedImageManager.trackedImagesChanged += OnTrackedImagesChanged;
82     }
83 }
84
85 @ Unity Message | 0 references
86 private void OnDisable()
87 {
88     if (trackedImageManager != null)
89     {
90         trackedImageManager.trackedImagesChanged -= OnTrackedImagesChanged;
91     }
92 }
93
94 @ Unity Message | 0 references
95 private void Start()
96 {
97     if (scanPromptUI != null)
98         scanPromptUI.SetActive(true);
99
100     if (experimentUI != null)
101         experimentUI.SetActive(false);
102 }

```

Εικόνα 4.20 Μέθοδοι Awake(), OnEnable()/OnDisable(), Start()

Κύρια μέθοδος χειρισμού tracking**OnTrackedImagesChanged(ARTrackedImagesChangedEventArgs args)**

Διαχειρίζεται τρεις λίστες, οι οποίες είναι οι παρακάτω με τον κώδικα να φαίνεται αναλυτικά στην εικόνα 4.21:

- 1) added: νέες εικόνες που εντοπίστηκαν → καλείται OnImageFound(...)
- 2) updated: εικόνες που συνεχίζουν να παρακολουθούνται → καλείται OnImageUpdated(...)
- 3) removed: εικόνες που χάθηκαν → καλείται OnImageLost(...)

```

2 references
private void OnTrackedImagesChanged(ARTrackedImagesChangedEventArgs args)
{
    foreach (ARTrackedImage trackedImage in args.added)
    {
        OnImageFound(trackedImage);
    }

    foreach (ARTrackedImage trackedImage in args.updated)
    {
        OnImageUpdated(trackedImage);
    }

    foreach (ARTrackedImage trackedImage in args.removed)
    {
        OnImageLost(trackedImage);
    }
}

```

Εικόνα 4.21 Μέθοδος OnTrackedImagesChanged(ARTrackedImagesChangedEventArgs args)

Λειτουργικές μέθοδοι

OnImageFound(ARTrackedImage img)

Ο κώδικας της μεθόδου απεικονίζεται στην εικόνα 4.22. Όταν εντοπιστεί το βιβλίο κάνει τις παρακάτω ενέργειες:

1. Ενεργοποιείται κατάσταση πειράματος (isBookDetected = true).
2. Δημιουργείται interactionRoot και “δένει” στο transform της εικόνας.
3. Γίνεται instantiate του φλόγιστρου: SpawnBunsenBurner(img).
4. Γίνεται instantiate των στοιχείων σε διάταξη γύρω από το φλόγιστρο: SpawnElements(img).
5. Εναλλαγή UI από scan → experiment.
6. Ενημέρωση κεντρικού διαχειριστή με GameManager.OnBookDetected() (ώστε να γνωρίζει ότι το πείραμα ξεκίνησε).

```

1 reference
private void OnImageFound(ARTrackedImage trackedImage)
{
    string imageName = trackedImage.referenceImage.name;

    if (imageName == "ChemistryBook" || imageName.ToLower().Contains("chemistry"))
    {
        currentTrackedImage = trackedImage;
        isBookDetected = true;
        interactionRoot = new GameObject("InteractionRoot").transform;
        interactionRoot.position = trackedImage.transform.position;
        interactionRoot.rotation = trackedImage.transform.rotation;

        if (scanPromptUI != null)
            scanPromptUI.SetActive(false);
        if (experimentUI != null)
            experimentUI.SetActive(true);

        SpawnBunsenBurner(trackedImage);
        SpawnElements(trackedImage);

        GameManager.Instance?.OnBookDetected();
    }
}

```

Εικόνα 4.22 Μέθοδος OnImageFound(ARTrackedImage img)

OnImageUpdated(ARTrackedImage img)

Όσο το tracking είναι ενεργό, ενημερώνει/συντηρεί τη σωστή τοποθέτηση των αντικειμένων. Αν υποβαθμιστεί το tracking, προετοιμάζει μετάβαση σε scan mode. Παρακάτω στην εικόνα 4.23 φαίνεται πλήρως ο κώδικας της μεθόδου.

```

1 reference
private void OnImageUpdated(ARTrackedImage trackedImage)
{
    string imageName = trackedImage.referenceImage.name;

    if (trackedImage.trackingState == TrackingState.Tracking)
    {
        if (spawnedObjects.TryGetValue(imageName, out GameObject spawnedObject))
        {
            spawnedObject.transform.position = trackedImage.transform.position + bunsenBurnerOffset;
            spawnedObject.transform.rotation = trackedImage.transform.rotation;
            spawnedObject.SetActive(true);
        }
        isBookDetected = true;
    }
    else if (trackedImage.trackingState == TrackingState.Limited)
    {
        if (spawnedObjects.TryGetValue(imageName, out GameObject spawnedObject))
        {
            spawnedObject.SetActive(false);
        }
        isBookDetected = false;
    }
}

```

Εικόνα 4.23 Μέθοδος OnImageUpdated(ARTrackedImage img)

OnImageLost(ARTrackedImage img)

Η μέθοδος αυτή χειρίζεται το σενάριο όπου το βιβλίο βγαίνει από το πλάνο με τον κώδικα της εικόνας 4.24. Με την απώλεια του στόχου:

- κρύβονται ή απενεργοποιούνται τα AR αντικείμενα
- εμφανίζεται ξανά το UI σάρωσης
- ειδοποιείται ο GameManager.OnBookLost()

```
1 reference
private void OnImageLost(ARTrackedImage trackedImage)
{
    string imageName = trackedImage.referenceImage.name;

    if (spawnedObjects.TryGetValue(imageName, out GameObject spawnedObject))
    {
        spawnedObject.SetActive(false);
    }

    isBookDetected = false;
    currentTrackedImage = null;

    if (scanPromptUI != null)
        scanPromptUI.SetActive(true);
    if (experimentUI != null)
        experimentUI.SetActive(false);

    GameManager.Instance?.OnBookLost();
    ...
}
```

Εικόνα 4.24 Μέθοδος OnImageLost(ARTrackedImage img)

Δημιουργία αντικειμένων**SpawnBunsenBurner(ARTrackedImage img)**

Κάνει instantiate του prefab του φλόγιστρου στο pose του στόχου (με offset/rotation όπου απαιτείται). Τα συμβολικά αντικείμενα που χρησιμοποιούνται στην πειραματική διαδικασία δημιουργούνται δυναμικά κατά τη διάρκεια της διαδικασίας εκτέλεσης μέσω της πολυμορφικής συνάρτησης instantiate() του Unity που βρίσκονται στην εικόνα 4.25. Επομένως, όταν το βιβλίο ανιχνεύεται ως event βιβλίου με τη μορφή target image, αναπαράγεται, σταθεροποιείται ως υπο-αντικείμενο στο interactionRoot, διατηρώντας έτσι μια συνεπή χωρική σχέση του στόχου παρακολούθησης κατά τη διάρκεια του πειράματος.

```

1 reference
private void SpawnBunsenBurner(ARTrackedImage trackedImage)
{
    if (bunsenBurnerPrefab == null) return;

    Vector3 spawnPosition = trackedImage.transform.position + bunsenBurnerOffset;
    Quaternion spawnRotation = trackedImage.transform.rotation;

    currentBunsenBurner = Instantiate(bunsenBurnerPrefab, spawnPosition, spawnRotation);
    currentBunsenBurner.transform.SetParent(trackedImage.transform);

    spawnedObjects[trackedImage.referenceImage.name] = currentBunsenBurner;
}

```

Εικόνα 4.25 Μέθοδος SpawnBunsenBurner(ARTrackedImage img)

SpawnElements(ARTrackedImage img)

Δημιουργεί τα στοιχεία γύρω από το φλόγιστρο, συνήθως σε κυκλικό σχηματισμό (μέσω sin/cos) ώστε να είναι οπτικά προσβάσιμα και ισοκαταναμημένα στο χώρο. Στην παρακάτω εικόνα 4.26 ακολουθεί ο κώδικας.

```

2 references
private void SpawnElements(ARTrackedImage trackedImage)
{
    if (elementPrefabs == null || elementPrefabs.Length == 0) return;

    foreach (GameObject element in spawnedElements)
    {
        if (element != null) Destroy(element);
    }
    spawnedElements.Clear();

    for (int i = 0; i < elementPrefabs.Length; i++)
    {
        if (elementPrefabs[i] == null) continue;

        float angle = (360f / elementPrefabs.Length) * i * Mathf.Deg2Rad;
        Vector3 offset = new Vector3(
            Mathf.Cos(angle) * elementSpawnRadius,
            elementFloatHeight,
            Mathf.Sin(angle) * elementSpawnRadius
        );

        Vector3 spawnPosition = trackedImage.transform.position +
            trackedImage.transform.TransformDirection(offset);

        GameObject element = Instantiate(elementPrefabs[i], spawnPosition, Quaternion.identity);
        element.transform.SetParent(interactionRoot);
        spawnedElements.Add(element);
    }
}

```

Εικόνα 4.26 Μέθοδος SpawnElements(ARTrackedImage img)

RespawnElements()

Χρησιμοποιείται σε restart: καθαρίζει και επαναδημιουργεί τα στοιχεία, ώστε να επανέλθει το πείραμα σε αρχική κατάσταση. Η εικόνα 4.27 παρουσιάζει τον κώδικα της μεθόδου RespawnElements αναλυτικά.

```

1 reference
public void RespawnElements()
{
    if (currentTrackedImage != null)
    {
        SpawnElements(currentTrackedImage);
    }
}

```

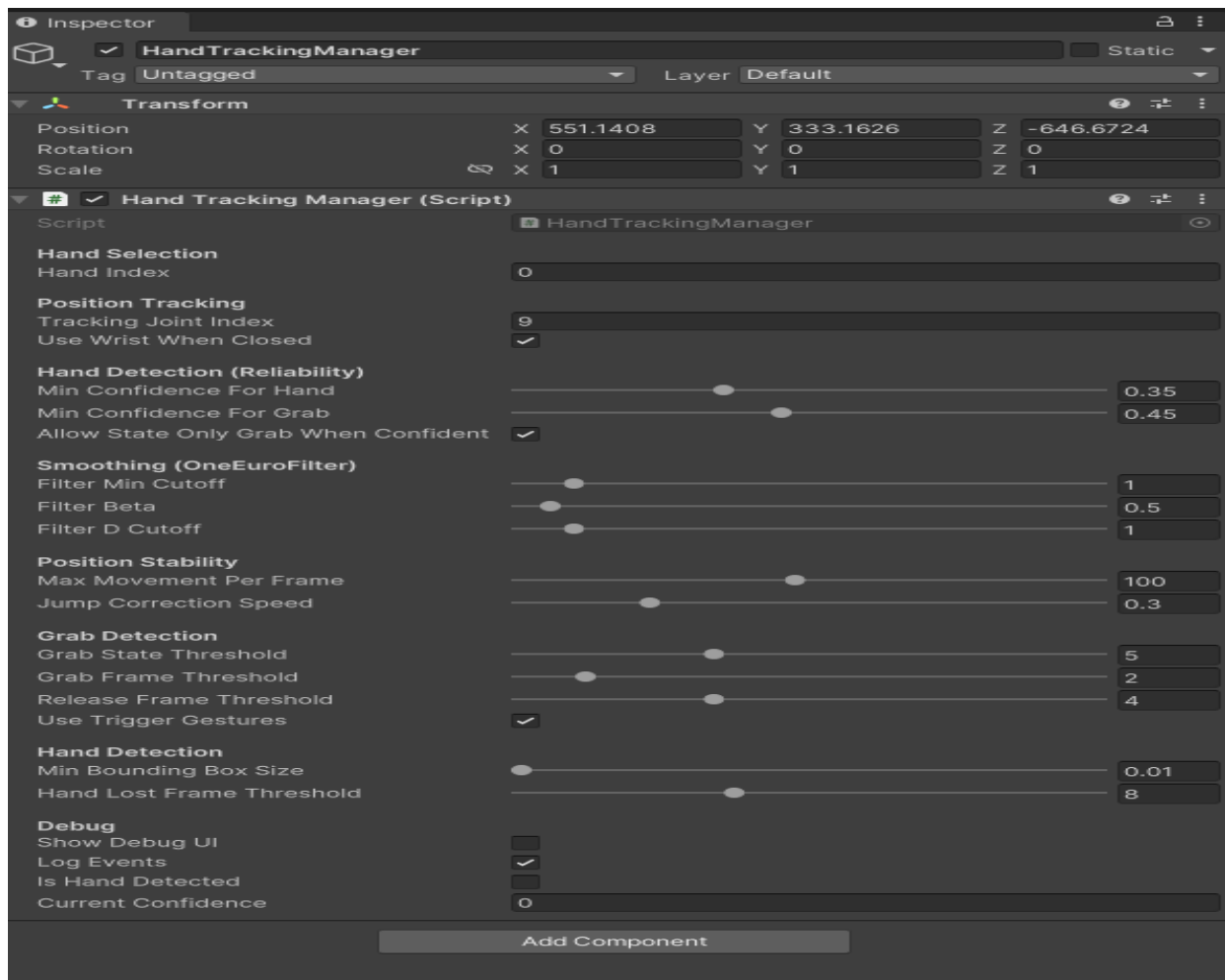
Εικόνα 4.27 Μέθοδος RespawnElements()

4.4.3 Αρχείο HandTrackingManager.cs

Η ροή χειρονομιών είναι το κεντρικό στοιχείο που ελέγχει το script της εικόνας 4.28. Τροφοδοτείται από το ManoMotion και σταθεροποιείται όταν ένα χέρι είναι διαθέσιμο σε οποιοδήποτε δεδομένο καρέ, μαζί με τον υπολογισμό της σταθερής θέσης του χεριού (οθόνη/κόσμος) και μιας αποτελεσματικής κατάστασης pitch/grab, η οποία περιλαμβάνει αντιθορυβικές ιδιότητες όπως φίλτρα και υστέρηση. Οι κύριες παράμετροι αποτελούνται από τις εξής:

Κύρια πεδία/παράμετροι:

- Κατάσταση: isHandDetected, isPinching (ή/και isGrabbing).
- Thresholds: confidence/bounding box thresholds.
- Counters: frames για “hand lost”, frames για “pinch start/end”.
- Smoothing: φίλτρα τύπου OneEuroFilter ανά άξονα.
- Jump prevention: μέγιστη μετακίνηση ανά frame και ομαλή διόρθωση.
- Depth mapping: παράμετροι που μετατρέπουν εκτίμηση βάθους σε world distance.



Εικόνα 4.28 HandTrackingManager.cs

Μέθοδοι κύκλου ζωής

Όπως φαίνεται και στον κώδικα της εικόνας 4.29, οι κύριες μέθοδοι κύκλου ζωής αποτελούνται από τις εξής:

- **Awake()**
Θέτει αρχικές τιμές και προετοιμάζει μεταβλητές κατάστασης.
- **Start()**
Ενεργοποιεί/ρυθμίζει το ManoMotion (hand skeleton/gesture features όπου απαιτείται) και αρχικοποιεί φίλτρα με InitializeFilters().
- **Update()**
Εκτελεί ανά frame την αναγνώριση:
 - σε συσκευή: ReadManoMotionData(),
 - σε editor: SimulateHandTracking() (δοκιμές με mouse).

```
#region Unity Lifecycle

// Unity Message | 0 references
private void Awake()
{
    if (Instance == null)
        Instance = this;
    else
    {
        Destroy(gameObject);
        return;
    }

    mainCamera = Camera.main;
    InitializeFilters();
}

// Unity Message | 0 references
private void Start()
{
    if (ManoMotionManager.Instance != null)
    {
        manoMotionFound = true;
        statusMessage = "ManoMotion OK";

        try
        {
            ManoMotionManager.Instance.ShouldCalculateSkeleton3D(true);
            ManoMotionManager.Instance.ShouldCalculateGestures(true);
            Debug.Log("[HandTracking] ManoMotion found and configured");
        }
        catch (Exception e)
        {
            Debug.LogWarning($"[HandTracking] Could not configure features: {e.Message}");
        }
    }
    else
    {
        manoMotionFound = false;
        statusMessage = "ManoMotion NOT FOUND";
        Debug.LogError($"[HandTracking] ManoMotionManager.Instance is null!*");
    }
}

// Unity Message | 0 references
private void Update()
{
    UpdateFilterParams();

    #if UNITY_EDITOR
        SimulateHandTracking();
        return;
    #endif

    #pragma warning disable CS0162
    ReadManoMotionData();
    #pragma warning restore CS0162
}
```

Εικόνα 4.29 Μέθοδοι Awake(), Start(), Update()

Κύρια επεξεργασία ManoMotion**ReadManoMotionData()**

Η μέθοδος αυτή, με τον κώδικα της στην εικόνα 4.30, είναι υπεύθυνη για τις παρακάτω ενέργειες:

1. Ανακτά πληροφορίες χεριού από το ManoMotion.
2. Επιλέγει “καλύτερο χέρι” όταν υπάρχουν πολλαπλά detections.
3. Αν δεν υπάρχει έγκυρο χέρι → HandleNoHand().
4. Αν υπάρχει:
 - ενημερώνει θέση: UpdateHandPosition(...)
 - υπολογίζει χειρονομίες: ProcessGestures(...)

```

370 private void ReadManoMotionData()
371 {
372     if (!manoMotionFound || ManoMotionManager.Instance == null)
373     {
374         isHandDetected = false;
375         return;
376     }
377     try
378     {
379         HandInfo[] handInfos = ManoMotionManager.Instance.HandInfos;
380
381         if (handInfos == null || handInfos.Length == 0)
382         {
383             HandleNoHand();
384             return;
385         }
386
387         int bestIdx = 0;
388         float bestScore = float.NegativeInfinity;
389
390         for (int i = 0; i < handInfos.Length; i++)
391         {
392             var ti = handInfos[i].trackingInfo;
393             float conf = ti.skeleton.confidence;
394             float area = ti.boundingBox.width * ti.boundingBox.height;
395             float score = conf + area;
396
397             if (score > bestScore)
398             {
399                 bestScore = score;
400                 bestIdx = i;
401             }
402         }
403
404         HandInfo handInfo = handInfos[bestIdx];
405
406         TrackingInfo trackingInfo = handInfo.trackingInfo;
407         BoundingBox bbox = trackingInfo.boundingBox;
408         boundingBoxWidth = bbox.width;
409         boundingBoxHeight = bbox.height;
410
411         depthEstimation = trackingInfo.depthEstimation;
412
413         SkeletonInfo skeleton = trackingInfo.skeleton;
414         currentConfidence = skeleton.confidence;
415
416         bool bboxOk = (boundingBoxWidth > minBoundingBoxSize &&
417             boundingBoxHeight > minBoundingBoxSize);
418
419         bool confOk = (currentConfidence > 0.15f);
420         bool handDetectedThisFrame = bboxOk && confOk;
421
422         if (handDetectedThisFrame)
423         {
424             noHandFrameCount = 0;
425             if (!isHandDetected)
426             {
427                 isHandDetected = true;
428                 ResetFilter();
429                 OnHandFound?.Invoke();
430                 if (logEvents) Debug.Log("[HandTracking] Hand FOUND");
431             }
432
433             UpdateHandPosition(trackingInfo, handInfo.gestureInfo);
434
435             ProcessGestures(handInfo.gestureInfo);
436
437             statusMessage = $"Tracking";
438         }
439         else
440         {
441             HandleNoHand();
442         }
443     }
444     catch (Exception e)
445     {
446         statusMessage = $"Error: {e.Message}";
447         Debug.LogError($"[HandTracking] {e.Message}");
448     }
449 }

```

Εικόνα 4.30 Μέθοδος ReadManoMotionData()

HandleNoHand()

Υλοποιεί σταθεροποίηση απώλειας χεριού, δεν αλλάζει άμεσα σε “hand lost” στο πρώτο χαμένο frame, αλλά μετά από συγκεκριμένο πλήθος frames. Αν το χέρι χαθεί:

- μηδενίζει counters
- παράγει κατάσταση “hand not detected”
- τερματίζει τυχόν ενεργό pinch

Στην παρακάτω εικόνα 4.31 εμφανίζεται ο κώδικας της μεθόδου.

```

2 references
private void HandleNoHand()
{
    noHandFrameCount++;

    if (noHandFrameCount >= handLostFrameThreshold && isHandDetected)
    {
        isHandDetected = false;
        OnHandLost?.Invoke();
        if (logEvents) Debug.Log("[HandTracking] Hand LOST");

        if (isPinching)
        {
            isPinching = false;
            OnPinchEnd?.Invoke();
        }

        grabFrameCount = 0;
        releaseFrameCount = 0;
        positionSource = "None";
    }
    statusMessage = "No hand";
}

```

Εικόνα 4.31 Μέθοδος HandleNoHand()

Υπολογισμός θέσης χεριού**UpdateHandPosition(...)**

Με τον κώδικα της εικόνας 4.32, υπολογίζει θέση χεριού με λογική fallback:

1. αν υπάρχει αξιόπιστη θέση joint (π.χ. palm) → χρήση joint
2. αλλιώς χρήση κέντρου bounding box

Στη συνέχεια:

- μετατρέπει normalized → screen (NormalizedToScreen(...))
- εφαρμόζει φίλτρα για ομαλοποίηση
- υπολογίζει world position (CalculateWorldPosition(...))
- εφαρμόζει jump prevention για αποφυγή απότομων “αλμάτων”.

```

1 reference
private void UpdateHandPosition(TrackingInfo trackingInfo, GestureInfo gestureInfo)
{
    Vector3 newScreenPos = Vector3.zero;
    bool positionFound = false;
    float timestamp = Time.time;

    SkeletonInfo skeleton = trackingInfo.skeleton;

    int jointToUse = trackingJointIndex;

    if (useWristWhenClosed && gestureInfo.state >= grabStateThreshold)
    {
        jointToUse = 0;
    }

    if (skeleton.jointPositions != null && skeleton.jointPositions.Length > jointToUse)
    {
        Vector3 jointPos = skeleton.jointPositions[jointToUse];

        if (IsValidJointPosition(jointPos))
        {
            newScreenPos = NormalizedToScreen(jointPos);
            positionFound = true;
            positionSource = jointToUse == 0 ? "Wrist" : $"Joint{jointToUse}";
        }
    }

    if (!positionFound)
    {
        BoundingBox bbox = trackingInfo.boundingBox;
        Vector3 bboxCenter = new Vector3(
            bbox.topLeft.x + bbox.width / 2f,
            bbox.topLeft.y + bbox.height / 2f,
            0f
        );
        newScreenPos = NormalizedToScreen(bboxCenter);
        positionFound = true;
        positionSource = "BBBox";
    }

    if (positionFound && IsValidScreenPosition(newScreenPos))
    {
        rawScreenPosition = newScreenPos;

        float filteredX = filterX.Filter(newScreenPos.x, timestamp);
        float filteredY = filterY.Filter(newScreenPos.y, timestamp);
        float filteredDepth = filterDepth.Filter(depthEstimation, timestamp);

        Vector3 filteredPos = new Vector3(filteredX, filteredY, filteredDepth);

        if (lastValidScreenPosition != Vector3.zero)
        {
            float distance = Vector2.Distance(
                new Vector2(filteredPos.x, filteredPos.y),
                new Vector2(lastValidScreenPosition.x, lastValidScreenPosition.y)
            );

            if (distance > maxMovementPerFrame)
            {
                filteredPos = Vector3.Lerp(lastValidScreenPosition, filteredPos, jumpCorrectionSpeed);
            }
        }

        smoothedScreenPosition = filteredPos;
        lastValidScreenPosition = smoothedScreenPosition;

        CalculateWorldPosition();
    }
}

```

Εικόνα 4.32 Μέθοδος UpdateHandPosition(...)

NormalizedToScreen(Vector2 normalized)

Μετατρέπει normalized συντεταγμένες σε pixels οθόνης και διορθώνει τον άξονα Y αν απαιτείται από το σύστημα συντεταγμένων του SDK, με τον κώδικα που βρίσκεται στην εικόνα 4.33.

CalculateWorldPosition(Vector2 screenPos, float depth)

Κάνει ray από την κάμερα προς το screen point και επιστρέφει σημείο στο επιθυμητό βάθος.

```

2 references
private Vector3 NormalizedToScreen(Vector3 normalizedPos)
{
    float screenX = normalizedPos.x * Screen.width;
    float screenY = (1f - normalizedPos.y) * Screen.height;
    return new Vector3(screenX, screenY, depthEstimation);
}

1 reference
private void CalculateWorldPosition()
{
    if (mainCamera != null)
    {
        Ray ray = mainCamera.ScreenPointToRay(smoothedScreenPosition);

        float distance = Mathf.Lerp(0.2f, 0.8f, smoothedScreenPosition.z);
        handWorldPosition = ray.GetPoint(distance);
    }
}

```

Εικόνα 4.33 Μέθοδοι NormalizedToScreen(Vector2 normalized), CalculateWorldPosition(Vector2 screenPos, float depth)

Αναγνώριση χειρονομίας (pinch/grab)

ProcessGestures(...)

Συνδυάζει περισσότερα από ένα σήματα (π.χ. state/class/triggers), όπως φαίνεται στην εικόνα 4.34 για να μειώσει τα ψευδώς θετικά. Εφαρμόζει υστέρηση:

- pinch start μόνο αν η κατάσταση “grab” κρατηθεί για X συνεχόμενα frames
 - pinch end μόνο αν η κατάσταση “no grab” κρατηθεί για Y frames
- Έτσι η αλληλεπίδραση γίνεται σταθερή σε πραγματικές συνθήκες φωτισμού/θορύβου.

```

1 reference
private void ProcessGestures(GestureInfo gestureInfo)
{
    currentManoClass = gestureInfo.manoClass;
    currentContinuousGesture = gestureInfo.manoGestureContinuous;
    currentTriggerGesture = gestureInfo.manoGestureTrigger;
    currentState = gestureInfo.state;

    bool stateIndicatesGrab = (currentState >= grabStateThreshold);

    bool continuousIndicatesGrab =
        (currentContinuousGesture == ManoGestureContinuous.CLOSED_HAND_GESTURE) ||
        (currentContinuousGesture == ManoGestureContinuous.HOLD_GESTURE);

    bool triggerIndicatesGrab = useTriggerGestures &&
        (currentTriggerGesture == ManoGestureTrigger.CLICK ||
         currentTriggerGesture == ManoGestureTrigger.GRAB_GESTURE);

    bool classIndicatesGrab =
        (currentManoClass == ManoClass.GRAB_GESTURE ||
         currentManoClass == ManoClass.PINCH_GESTURE);

    bool shouldBeGrabbing =
        triggerIndicatesGrab ||
        (continuousIndicatesGrab && stateIndicatesGrab) ||
        (classIndicatesGrab && stateIndicatesGrab);

    if (shouldBeGrabbing)
    {
        grabFrameCount++;
        releaseFrameCount = 0;

        if (!isPinching && grabFrameCount >= grabFrameThreshold)
        {
            isPinching = true;
            OnPinchStart?.Invoke();
        }
    }
    else
    {
        releaseFrameCount++;

        if (isPinching && releaseFrameCount >= releaseFrameThreshold)
        {
            isPinching = false;
            grabFrameCount = 0;
            OnPinchEnd?.Invoke();
        }
    }
}

```

Εικόνα 4.34 Μέθοδος ProcessGestures(...)

Δημόσιο API προς interaction layer

Η κλάση εκθέτει συναρτήσεις, που φαίνονται αναλυτικά στην εικόνα 4.35, που χρησιμοποιούνται από HandInteractionController, οι οποίες είναι οι παρακάτω:

- **IsPinching()**: επιστρέφει αν υπάρχει ενεργό pinch.
- **GetGrabPointWorldSpace(...)**: επιστρέφει το world-space σημείο “λαβής”.
- **GetGrabPointScreenSpace()**: επιστρέφει το screen-space σημείο “λαβής”.

```

745 #region Public API
746
747 /// <summary>Returns true if hand is detected AND grabbing/pinching</summary>
748 5 references
749 public bool IsPinching() => isHandDetected && isPinching;
750
751 /// <summary>Returns true if hand is grabbing (same as IsPinching)</summary>
752 0 references
753 public bool IsGrabbing() => IsPinching();
754
755 /// <summary>Get hand position in world space (smoothed)</summary>
756 0 references
757 public Vector3 GetHandWorldPosition() => handWorldPosition;
758
759 /// <summary>Get hand position in screen space (smoothed)</summary>
760 0 references
761 public Vector3 GetHandScreenPosition() => smoothedScreenPosition;
762
763 /// <summary>Get grab point in world space</summary>
764 1 reference
765 public Vector3 GetGrabPointWorldSpace(float depthMeters = 0.5f)
766 {
767     if (mainCamera == null) return handWorldPosition;
768     Vector3 sp = smoothedScreenPosition;
769     sp.z = 0f;
770
771     Ray ray = mainCamera.ScreenPointToRay(sp);
772     return ray.GetPoint(depthMeters);
773 }
774
775 /// <summary>Get grab point in screen space</summary>
776 1 reference
777 public Vector3 GetGrabPointScreenSpace() => smoothedScreenPosition;
778
779 /// <summary>Get palm center position</summary>
780 0 references
781 public Vector3 GetPalmCenter() => handWorldPosition;
782
783 /// <summary>Get current hand state (0=open, 13=closed)</summary>
784 0 references
785 public int GetHandState() => currentState;
786
787 /// <summary>Get current ManoClass</summary>
788 0 references
789 public ManoClass GetManoClass() => currentManoClass;
790
791 /// <summary>Get depth estimation (0=close, 1=far)</summary>
792 0 references
793 public float GetDepth() => depthEstimation;
794
795 /// <summary>Get raw (unfiltered) screen position</summary>
796 0 references
797 public Vector3 GetRawScreenPosition() => rawScreenPosition;

```

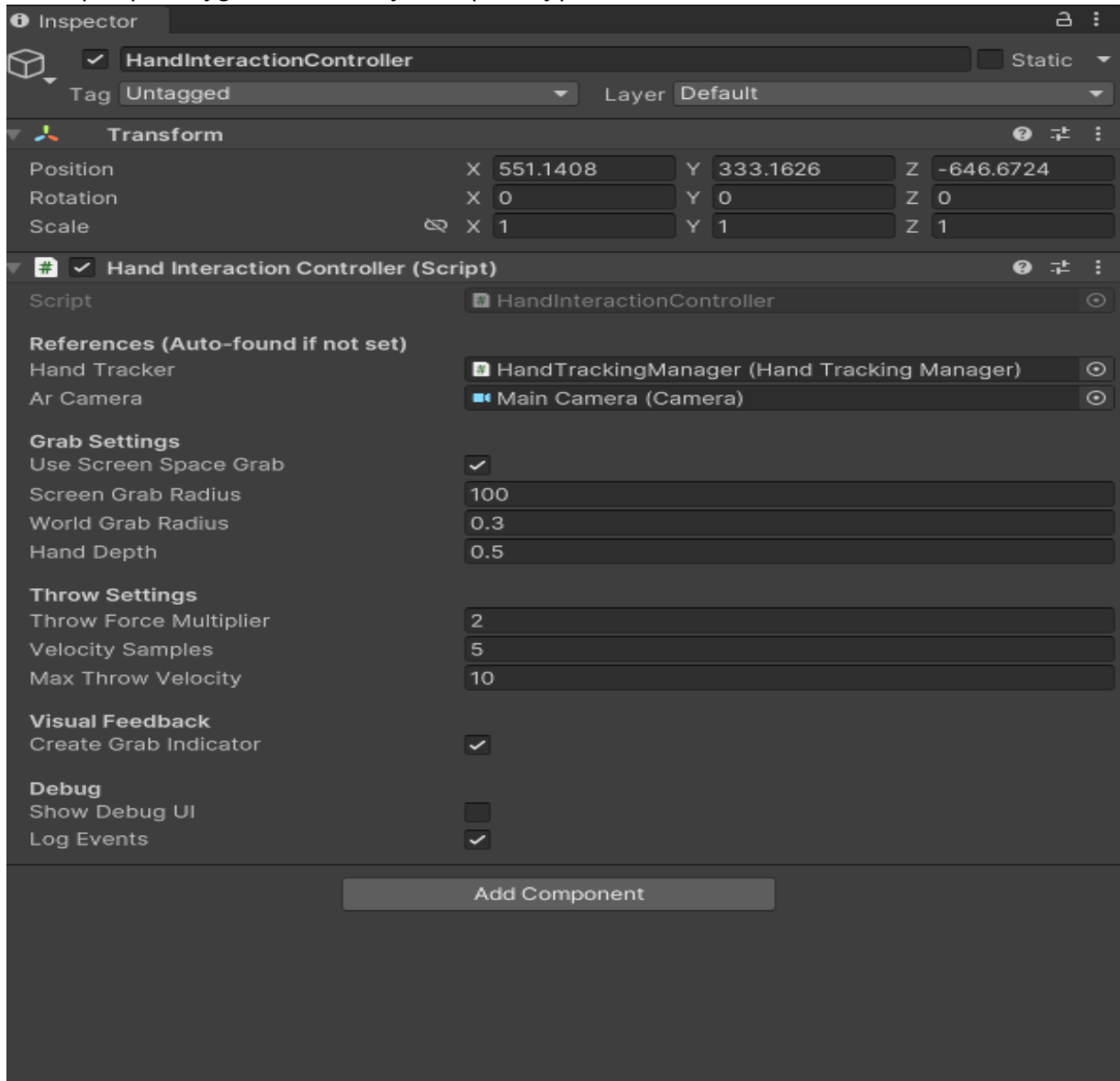
Εικόνα 4.35 Μέθοδοι IsPinching(), GetGrabPointWorldSpace(...), GetGrabPointScreenSpace()

4.4.4 Αρχείο HandInteractionController.cs

Το σύστημα μετατρέπει τη χειρονομία τσιμπήματος και τη στάση του χεριού σε ενέργειες στο ElementObject, οι οποίες περιλαμβάνουν την επιλογή αντικειμένου, τη σύλληψη, τη μετακίνηση και την απελευθέρωση ή το πέταγμα. Τα κύρια του πεδία είναι τα παρακάτω:

Κύρια πεδία:

- Αναφορά σε HandTrackingManager και Camera.main.
- Λίστα ElementObject που είναι διαθέσιμα στη σκηνή.
- Παράμετροι grab (ακτίνα σε screen/world).
- Παράμετροι throw: throwForceMultiplier, maxThrowVelocity.
- Buffers ιστορικού θέσεων/χρόνων για υπολογισμό ταχύτητας.
- Προαιρετικός grabIndicator ως οπτική ένδειξη.



Εικόνα 4.36 HandInteractionController.cs

Μέθοδοι

Οι μέθοδοι του HandInteractionController απεικονίζονται στις εικόνες 4.37 έως 4.40 και είναι οι εξής:

- **Start()**
Εντοπίζει dependencies (tracker/camera), δημιουργεί indicator και συλλέγει στοιχεία (RefreshElements()).

```

Unity Message | 0 references
private void Start()
{
    // Auto-find references
    if (handTracker == null)
        handTracker = HandTrackingManager.Instance;

    if (handTracker == null)
        handTracker = FindObjectOfType<HandTrackingManager>();

    if (arCamera == null)
        arCamera = Camera.main;

    // Create visual indicator
    if (createGrabIndicator)
        CreateGrabIndicator();

    // Initial element scan
    RefreshElements();

    if (handTracker != null)
        Debug.Log("[HandInteraction] Initialized successfully!");
    else
        Debug.LogError("[HandInteraction] HandTrackingManager NOT FOUND!");

    if (allElements.Count == 0)
        Debug.LogWarning("[HandInteraction] No ElementObjects found in scene!");
}

```

Εικόνα 4.37 Μέθοδος Start()

- **RefreshElements()**
Σαρώνει τη σκηνή και αποθηκεύει όλα τα ElementObject σε λίστα.
- **Update()**
 1. Αν δεν υπάρχει χέρι: κρύβει indicator, εκτελεί ForceRelease() αν χρειάζεται.
 2. Αν υπάρχει χέρι: ενημερώνει indicator, βρίσκει πλησιέστερο στοιχείο (FindNearestElement()), και εκτελεί λογική grab/release (HandleGrabLogic()).

```

2 references
public void RefreshElements()
{
    allElements.Clear();

    ElementObject[] found = FindObjectsOfType<ElementObject>();
    allElements.AddRange(found);

    lastElementRefresh = Time.time;

    if (LogEvents)
    {
        Debug.Log($"[HandInteraction] Found {allElements.Count} elements:");
        foreach (var el in allElements)
        {
            Debug.Log($" - {el.name} at {el.transform.position}");
        }
    }
}

0 references
public void RegisterElement(ElementObject element)
{
    if (element != null && !allElements.Contains(element))
    {
        allElements.Add(element);
        Debug.Log($"[HandInteraction] Registered: {element.name}");
    }
}

0 Unity Message | 0 references
private void Update()
{
    if (Time.time - lastElementRefresh > 2f)
    {
        RefreshElements();
    }

    if (handTracker == null || !handTracker.isHandDetected)
    {
        if (grabIndicator != null)
            grabIndicator.SetActive(false);

        if (grabbedElement != null)
        {
            ForceRelease();
        }
        return;
    }

    currentHandWorldPos = handTracker.GetGrabPointWorldSpace(handDepth);
    currentHandScreenPos = handTracker.GetGrabPointScreenSpace();

    UpdateGrabIndicator();

    FindNearestElement();

    HandleGrabLogic();

    if (grabbedElement != null)
    {
        grabbedElement.UpdateGrabbedPosition(currentHandWorldPos);
        RecordPosition(currentHandWorldPos);
    }
}

```

Εικόνα 4.38 Μέθοδοι RefreshElements(), Update()

- **FindNearestElement(...)**
Υπολογίζει το πλησιέστερο στοιχείο συνήθως με screen-space απόσταση για σταθερότητα σε mobile.
- **HandleGrabLogic()**
 1. pinch start → TryGrab()
 2. pinch hold → UpdateGrabbedPosition() + καταγραφή ιστορικού (throw velocity)

3. pinch end → Release()

• TryGrab()

Αν το πιο κοντινό στοιχείο είναι εντός ακτίνας, καλεί ElementObject.OnGrab() και ξεκινά ιστορικό.

```
1 reference
private void FindNearestElement()
{
    nearestElement = null;
    nearestDistance = float.MaxValue;
    nearestScreenDistance = float.MaxValue;

    if (grabbedElement != null) return;
    if (arCamera == null) return;

    foreach (var element in allElements)
    {
        if (element == null) continue;
        if (element.IsGrabbed || element.IsThrown) continue;

        float worldDist = Vector3.Distance(currentHandWorldPos, element.transform.position);

        Vector3 elementScreenPos = arCamera.WorldToScreenPoint(element.transform.position);
        float screenDist = Vector2.Distance(
            new Vector2(currentHandScreenPos.x, currentHandScreenPos.y),
            new Vector2(elementScreenPos.x, elementScreenPos.y)
        );

        float compareDist = useScreenSpaceGrab ? screenDist : worldDist;
        float currentBest = useScreenSpaceGrab ? nearestScreenDistance : nearestDistance;

        if (compareDist < currentBest)
        {
            nearestDistance = worldDist;
            nearestScreenDistance = screenDist;
            nearestElement = element;
        }
    }
}

1 reference
private void HandleGrabLogic()
{
    bool isGrabbing = handTracker.IsPinching();

    // Started pinching
    if (isGrabbing && !wasGrabbing)
    {
        TryGrab();
    }
    // Released pinch
    else if (!isGrabbing && wasGrabbing)
    {
        Release();
    }

    wasGrabbing = isGrabbing;
}

1 reference
private void TryGrab()
{
    if (nearestElement != null && CanGrabNearest())
    {
        grabbedElement = nearestElement;
        grabbedElement.OnGrab();

        // Clear velocity history
        positionHistory.Clear();
        timeHistory.Clear();
        RecordPosition(currentHandWorldPos);

        if (LogEvents)
            Debug.Log($"[HandInteraction] *** GRABBED: {grabbedElement.name} ***");
    }
    else
    {
        if (LogEvents)
        {
            if (nearestElement != null)
            {
                if (useScreenSpaceGrab)
                    Debug.Log($"[HandInteraction] Too far to grab {nearestElement.name} (screen dist: {nearestScreenDistance:F0}px, need: {screenGrabRadius:F0}px)");
                else
                    Debug.Log($"[HandInteraction] Too far to grab {nearestElement.name} (world dist: {nearestDistance:F3}m, need: {worldGrabRadius:F3}m)");
            }
            else
            {
                Debug.Log($"[HandInteraction] No element found to grab!");
            }
        }
    }
}
}
```

Εικόνα 4.39 Μέθοδοι FindNearestElement(...), HandleGrabLogic(), TryGrab()

• Release()

Υπολογίζει ταχύτητα από ιστορικό (CalculateThrowVelocity()), και καλεί ElementObject.OnRelease(velocity).

• CalculateThrowVelocity()

Υπολογίζει μέση/τελική ταχύτητα χεριού από τα τελευταία δείγματα θέσης και χρόνου.

• ForceRelease()

“Ασφαλής” απελευθέρωση όταν χαθεί το χέρι ή διακοπεί η διαδικασία.

```

1 reference
private void Release()
{
    if (grabbedElement == null) return;

    Vector3 velocity = CalculateThrowVelocity() * throwForceMultiplier;

    // Clamp velocity
    if (velocity.magnitude > maxThrowVelocity)
        velocity = velocity.normalized * maxThrowVelocity;

    if (LogEvents)
        Debug.Log($"[HandInteraction] *** RELEASED: {grabbedElement.name} velocity: {velocity.magnitude:F2} ***");

    grabbedElement.OnRelease(velocity);
    grabbedElement = null;
}

1 reference
public void ForceRelease()
{
    if (grabbedElement != null)
    {
        if (LogEvents)
            Debug.Log($"[HandInteraction] Force released: {grabbedElement.name}");
        grabbedElement.OnRelease(Vector3.down * 0.5f);
        grabbedElement = null;
    }

    positionHistory.Clear();
    timeHistory.Clear();
}

2 references
private void RecordPosition(Vector3 pos)
{
    positionHistory.Add(pos);
    timeHistory.Add(Time.time);

    while (positionHistory.Count > velocitySamples)
    {
        positionHistory.RemoveAt(0);
        timeHistory.RemoveAt(0);
    }
}

1 reference
private Vector3 CalculateThrowVelocity()
{
    if (positionHistory.Count < 2) return Vector3.zero;

    Vector3 totalVel = Vector3.zero;
    int samples = 0;

    for (int i = 1; i < positionHistory.Count; i++)
    {
        float dt = timeHistory[i] - timeHistory[i - 1];
        if (dt > 0.001f)
        {
            totalVel += (positionHistory[i] - positionHistory[i - 1]) / dt;
            samples++;
        }
    }

    return samples > 0 ? totalVel / samples : Vector3.zero;
}

```

Εικόνα 4.40 Μέθοδοι Release(), CalculateThrowVelocity(), ForceRelease()

4.4.5 Αρχείο ElementObject.cs

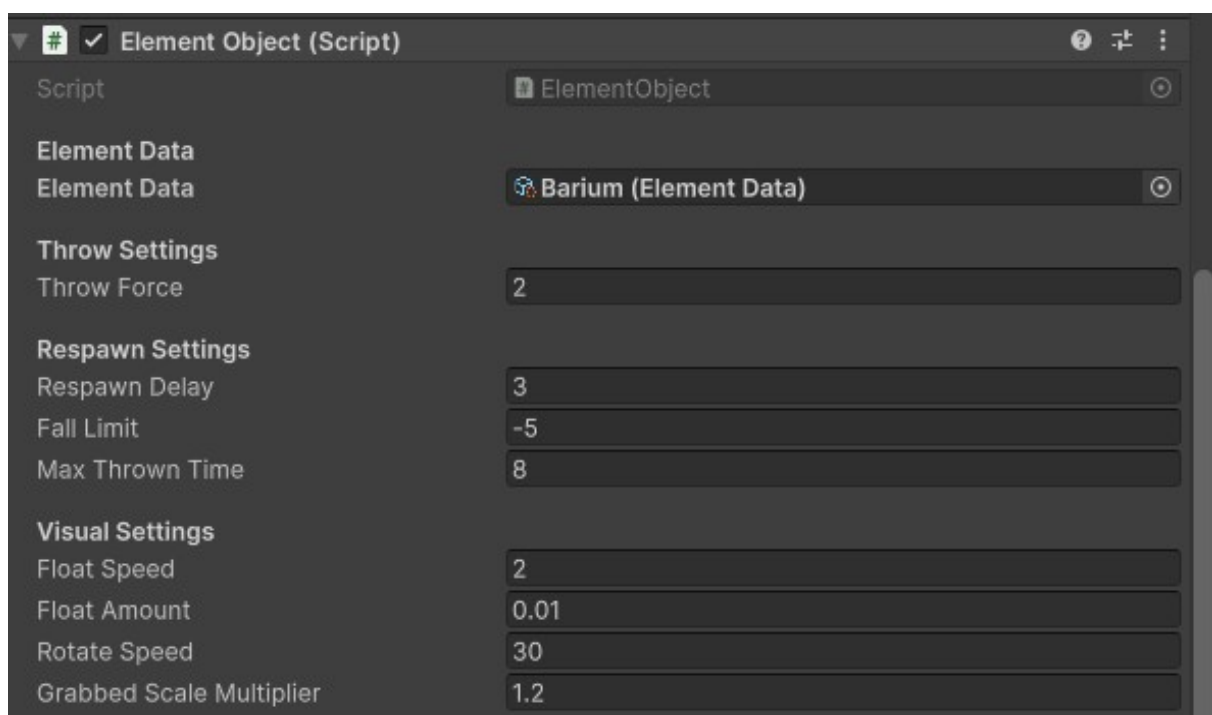
Το στοιχείο ελέγχει τη δυναμική της αναπαράστασης του χημικού στοιχείου κατά τη διάρκεια της εκτέλεσης και λειτουργεί ως ενεργοποιητής της πειραματικής λογικής. Εκτός από την επεξεργασία κινηματικών αλληλεπιδράσεων (άρπαγμα και ρίψη), επεξεργάζεται συμβάντα σύγκρουσης και ενεργοποίησης που σχετίζονται με τη φλόγα και, στη συνέχεια, στέλνει εκπαιδευτική ανατροφοδότηση με τη μορφή οπτικών

Κεφαλαίο 4

δεικτών φλόγας και υλικών διεπαφής χρήστη. Επιπλέον, ενισχύει έναν μηχανισμό αναγέννησης που καθιστά το πείραμα επαναλαμβανόμενο. Τα κύρια πεδία του είναι:

Κύρια πεδία:

- `elementData`: δεδομένα στοιχείου (όνομα/χρώματα φλόγας/περιγραφή).
- `rb`, `col`, `meshRenderer`: components φυσικής και ορατότητας.
- Παράμετροι `idle animation`: `floatSpeed`, `floatAmount`, `rotateSpeed`.
- Παράμετροι `grab`: `grabbedScaleMultiplier` (οπτική ένδειξη “πιασίματος”).
- Παράμετροι `throw`: `throwForce`.
- Παράμετροι ασφαλείας/`respawn`: `respawnDelay`, `fallLimit`, `maxThrownTime`.
- Flags κατάστασης: `isGrabbed`, `isThrown`, `isRespawning`.
- Αρχικές τιμές επαναφοράς: `startPosition`, `startRotation`, `startScale`.



Εικόνα 4.41 ElementObject.cs

Μέθοδοι

Οι μέθοδοι του `ElementObject` απεικονίζονται στις εικόνες 4.42 έως 4.44 και είναι οι εξής:

- **Awake()**
Εξασφαλίζει ότι υπάρχουν τα βασικά components (`Rigidbody/Collider`) και θέτει αρχική κατάσταση “idle” χωρίς gravity/physics.
- **Start()**
Αποθηκεύει αρχική θέση/περιστροφή/κλίμακα για χρήση στο `respawn`.
- **Update()**
 - Αν είναι `idle`: εκτελεί αιώρηση/περιστροφή (οπτική ένδειξη διαδραστικότητας).
 - Αν είναι `thrown`: ελέγχει αν έπεσε κάτω από `fallLimit` ή αν πέρασε `maxThrownTime` → `respawn`.

```

@ Unity Message | 0 references
private void Awake()
{
    rb = GetComponent<Rigidbody>();
    if (rb == null)
    {
        rb = gameObject.AddComponent<Rigidbody>();
    }

    col = GetComponent<Collider>();
    if (col == null)
    {
        col = gameObject.AddComponent<BoxCollider>();
    }

    meshRenderer = GetComponent<MeshRenderer>();

    rb.isKinematic = true;
    rb.useGravity = false;
}

@ Unity Message | 0 references
private void Start()
{
    startPosition = transform.position;
    startRotation = transform.rotation;
    startScale = transform.localScale;

    Debug.Log($"[Element] {gameObject.name} ready at {startPosition}");
}

@ Unity Message | 0 references
private void Update()
{
    if (!isGrabbed && !isThrown && !isRespawning)
    {
        float yOffset = Mathf.Sin(Time.time * floatSpeed) * floatAmount;
        transform.position = startPosition + new Vector3(0, yOffset, 0);
        transform.Rotate(Vector3.up, rotateSpeed * Time.deltaTime);
    }

    if (isThrown && !isRespawning)
    {
        thrownTime += Time.deltaTime;

        if (transform.position.y < fallLimit)
        {
            Debug.Log($"[Element] {gameObject.name} fell out of bounds, respawning...");
            StartRespawn();
        }

        else if (thrownTime > maxThrownTime)
        {
            Debug.Log($"[Element] {gameObject.name} timed out, respawning...");
            StartRespawn();
        }
    }
}

```

Εικόνα 4.42 Μέθοδοι Awake(), Start(), Update()

- **OnGrab()**
Θέτει κατάσταση grabbed, μηδενίζει ταχύτητες, απενεργοποιεί φυσική (kinematic) και εφαρμόζει scale feedback.
- **UpdateGrabbedPosition(Vector3position)**
Μετακινεί ομαλά το στοιχείο προς τη θέση χειριού/στόχου (σταθερή και φυσική κίνηση).
- **OnRelease(Vector3velocity)**
Θέτει thrown state, ενεργοποιεί gravity/physics και εφαρμόζει αρχική ταχύτητα ρίψης.
- **OnTriggerEnter(Colliderother)**
Αν ανιχνευτεί είσοδος στη φλόγα και το στοιχείο είναι thrown:
 - καλεί αλλαγή χρώματος φλόγας (FlameController.ChangeFlameColor με βάση elementData),
 - προβάλλει πληροφορίες στοιχείου (UIManager.ShowElementInfo(elementData)),

- ξεκινά διαδικασία respawn.

```

2 references
public void OnGrab()
{
    if (isThrown || isRespawning) return;

    isGrabbed = true;
    rb.isKinematic = true;
    rb.velocity = Vector3.zero;
    rb.angularVelocity = Vector3.zero;

    transform.localScale = startScale * grabbedScaleMultiplier;

    Debug.Log($"[Element] GRABBED: {gameObject.name}");
}

5 references
public void OnRelease(Vector3 velocity)
{
    isGrabbed = false;
    isThrown = true;
    thrownTime = 0f;

    rb.isKinematic = false;
    rb.useGravity = true;
    rb.velocity = velocity * throwForce;

    transform.localScale = startScale;

    Debug.Log($"[Element] THROWN: {gameObject.name} velocity: {velocity.magnitude:F2}");
}

2 references
public void UpdateGrabbedPosition(Vector3 position)
{
    if (!isGrabbed) return;
    transform.position = Vector3.Lerp(transform.position, position, Time.deltaTime * 15f);
}

@ Unity Message | 0 references
private void OnTriggerEnter(Collider other)
{
    if (!isThrown || isRespawning) return;

    FlameController flame = other.GetComponent<FlameController>();
    if (flame == null)
        flame = other.GetComponentInParent<FlameController>();

    if (flame != null)
    {
        Debug.Log($"[Element] {gameObject.name} HIT THE FLAME!");

        if (elementData != null)
        {
            flame.ChangeFlameColor(elementData.flameOuterColor, elementData.flameInnerColor);
        }

        if (UIManager.Instance != null)
        {
            UIManager.Instance.ShowElementInfo(elementData);
        }
    }

    StartRespawn();
}

```

Εικόνα 4.43 Μέθοδοι OnGrab(), UpdateGrabbedPosition(), OnRelease(), OnTriggerEnter()

- **OnCollisionEnter(Collision collision)**
Αν χτυπήσει στο περιβάλλον ενώ είναι thrown, ξεκινά respawn ώστε να μην “χαθεί” το στοιχείο.
- **StartRespawn()**
Θέτει isRespawning, κρύβει οπτικά το στοιχείο, απενεργοποιεί collider, μηδενίζει velocities και προγραμματίζει επαναφορά.
- **DoRespawn()**
Επαναφέρει θέση/περιστροφή/κλίμακα στις αρχικές, επαναφέρει idle physics και ενεργοποιεί ξανά mesh/collider.
- **ForceRespawn()**
Εξαναγκασμένο reset (π.χ. από restart του πειράματος).

```

0 Unity Message | 0 references
private void OnCollisionEnter(Collision collision)
{
    if (isThrown && !isRespawning)
    {
        Debug.Log($"[Element] {gameObject.name} hit: {collision.gameObject.name}");

        StartRespawn();
    }
}

4 references
public void StartRespawn()
{
    if (isRespawning) return;

    isRespawning = true;
    isThrown = false;
    isGrabbed = false;

    if (meshRenderer != null)
        meshRenderer.enabled = false;

    rb.isKinematic = true;
    rb.velocity = Vector3.zero;
    rb.angularVelocity = Vector3.zero;

    if (col != null)
        col.enabled = false;

    Debug.Log($"[Element] {gameObject.name} respawning in {respawnDelay}s...");

    Invoke(nameof(DoRespawn), respawnDelay);
}

3 references
private void DoRespawn()
{
    transform.position = startPosition;
    transform.rotation = startRotation;
    transform.localScale = startScale;

    rb.isKinematic = true;
    rb.useGravity = false;
    rb.velocity = Vector3.zero;
    rb.angularVelocity = Vector3.zero;

    if (meshRenderer != null)
        meshRenderer.enabled = true;

    if (col != null)
        col.enabled = true;

    isRespawning = false;
    isThrown = false;
    isGrabbed = false;
    throwTime = 0f;

    Debug.Log($"[Element] {gameObject.name} RESPAWNED!");
}

0 references
public void ForceRespawn()
{
    CancelInvoke(nameof(DoRespawn));
    isRespawning = false;
    DoRespawn();
}

```

Εικόνα 4.44 Μέθοδοι OnCollisionEnter(), StartRespawn(), DoRespawn(), ForeRespawn()

4.4.6 Αρχείο FlameController.cs

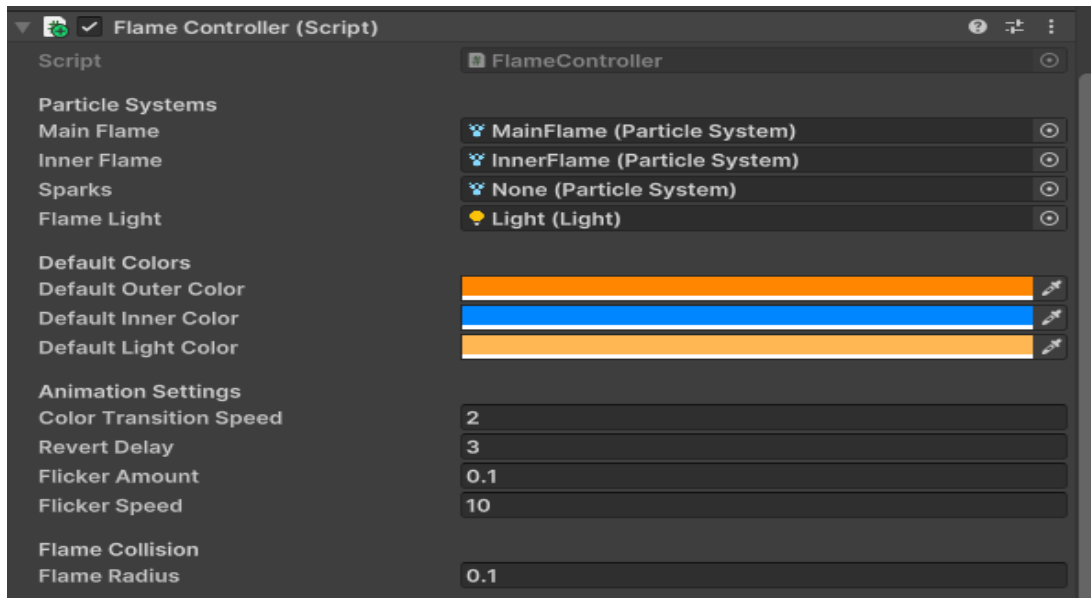
Παρέχει οπτική ενίσχυση στις αντιδράσεις καύσης κατά την εφαρμογή σωματιδίων και φωτισμού και επιτυγχάνει αλλαγή στο χρώμα των φλογών ανάλογα με τα συστατικά στοιχεία. Αυτή η διαδικασία βελτιώνει την ψευδαίσθηση της επιτυχίας μέσω της διαδικασίας της δημιουργίας σπινθήρων και της προσωρινής αύξησης της έντασης του φωτός και, ταυτόχρονα, επαναφέρει τη φλόγα στην αρχική της κατάσταση, ώστε να είναι δυνατή η διαδοχική δοκιμή της πειραματικής διαδικασίας. Τα βασικά πεδία του αποτελούνται από τα εξής:

Κύρια πεδία:

- outerFlameParticles, innerFlameParticles: κύρια particle systems φλόγας.
- sparkParticles: εφέ επιτυχίας (σπινθήρες).

Κεφαλαίο 4

- `flameLight`: φωτισμός φλόγας (χρώμα/ένταση).
- Default χρώματα: `defaultOuterColor`, `defaultInnerColor`, `defaultLightColor`.
- Target χρώματα: `targetOuterColor`, `targetInnerColor`, `targetLightColor` (για ομαλή μετάβαση).
- Παράμετροι μετάβασης: `colorLerpSpeed`.
- Παράμετροι flicker: `flickerIntensity`, `flickerSpeed`.
- Χρονισμός επαναφοράς: `revertDelay`.
- Trigger περιοχή (προαιρετικά): `collider/radius` για ανίχνευση.



Εικόνα 4.45 FlameController.cs

Μέθοδοι

Οι μέθοδοι του `FlameController` απεικονίζονται στις εικόνες 4.46 έως 4.49 και είναι οι εξής:

- **Awake()/Start()**
Αρχικοποιεί default χρώματα, ρυθμίζει particle systems, φωτισμό και trigger collider.

```
Unity Message | 0 references
private void Awake()
{
    Instance = this;
}

Unity Message | 0 references
private void Start()
{
    InitializeColors();
    SetupParticleSystems();
    SetupCollider();
}
```

Εικόνα 4.46 Μέθοδοι Awake()/Start()

- **Update()**
 - **UpdateColors()** για gradual μετάβαση στα target colors.

- **UpdateFlicker()** για ρεαλιστικό “τρέμουλο” φωτισμού.
- **UpdateColors()**
Κάνει Lerp current→target και εφαρμόζει χρώματα σε particles/light.
- **UpdateFlicker()**
Μεταβάλλει ελαφρά την ένταση του φωτός (noise/περιοδικότητα) για φυσικό αποτέλεσμα.

```

Unity Message | 0 references
private void Update()
{
    UpdateColors();
    UpdateFlicker();
}

1 reference
private void UpdateColors()
{
    currentOuterColor = Color.Lerp(currentOuterColor, targetOuterColor, Time.deltaTime * colorTransitionSpeed);
    currentInnerColor = Color.Lerp(currentInnerColor, targetInnerColor, Time.deltaTime * colorTransitionSpeed);
    currentLightColor = Color.Lerp(currentLightColor, targetLightColor, Time.deltaTime * colorTransitionSpeed);

    if (mainFlame != null)
    {
        mainFlameMain.startColor = currentOuterColor;
    }

    if (innerFlame != null)
    {
        innerFlameMain.startColor = currentInnerColor;
    }

    if (flameLight != null)
    {
        flameLight.color = currentLightColor;
    }
}

1 reference
private void UpdateFlicker()
{
    if (flameLight != null)
    {
        float flicker = Mathf.PerlinNoise(Time.time * flickerSpeed, 0f) * flickerAmount;
        flameLight.intensity = baseLightIntensity + flicker;
    }
}

```

Εικόνα 4.47 Μέθοδοι Update(), UpdateColors(), UpdateFlicker()

- **ChangeFlameColor(ColorouterColor,ColorinnerColor)**
Θέτει τα target colors, ενεργοποιεί sparks και ξεκινά boost φωτισμού. Προγραμματίζει auto-revert.
- **ChangeFlameColor(ElementDataelement)**
Overload που αντλεί χρώματα απευθείας από elementData.

```

2 references
public void ChangeFlameColor(Color outerColor, Color innerColor)
{
    targetOuterColor = outerColor;
    targetInnerColor = innerColor;

    targetLightColor = outerColor;

    PlaySparkEffect(outerColor);

    StartCoroutine(BoostLight());

    if (revertCoroutine != null)
    {
        StopCoroutine(revertCoroutine);
    }
    revertCoroutine = StartCoroutine(RevertToDefaultColor());

    Debug.Log($"[Flame] Color changed to {outerColor}");
}

0 references
public void ChangeFlameColor(ElementData element)
{
    if (element != null)
    {
        ChangeFlameColor(element.flameOuterColor, element.flameInnerColor);
    }
}

```

Εικόνα 4.48 Μέθοδοι ChangeFlameColor(), ChangeFlameColor()

- **BoostLight()**
Coroutine που αυξάνει προσωρινά την ένταση και επιστρέφει σε κανονικά επίπεδα.
- **RevertToDefaultColor()**
Coroutine που μετά από revertDelay επαναφέρει target colors στα default.
- **ResetFlame()**
Άμεση επαναφορά φλόγας (χρώμα + ένταση) σε αρχικές τιμές, χρήσιμο σε restart.

```

1 reference
private IEnumerator BoostLight()
{
    if (FlameLight == null) yield break;

    float boostDuration = 8.5f;
    float boostMultiplier = 2f;
    float originalIntensity = baseLightIntensity;
    float elapsed = 0f;

    while (elapsed < boostDuration / 2)
    {
        elapsed += Time.deltaTime;
        float t = elapsed / (boostDuration / 2);
        baseLightIntensity = Mathf.Lerp(originalIntensity, originalIntensity * boostMultiplier, t);
        yield return null;
    }

    elapsed = 0f;
    float boostedIntensity = baseLightIntensity;

    while (elapsed < boostDuration / 2)
    {
        elapsed += Time.deltaTime;
        float t = elapsed / (boostDuration / 2);
        baseLightIntensity = Mathf.Lerp(boostedIntensity, originalIntensity, t);
        yield return null;
    }

    baseLightIntensity = originalIntensity;
}

1 reference
private IEnumerator RevertToDefaultColor()
{
    yield return new WaitForSeconds(revertDelay);

    targetOuterColor = defaultOuterColor;
    targetInnerColor = defaultInnerColor;
    targetLightColor = defaultLightColor;

    Debug.Log("[Flame] Reverting to default color");
}

0 reference
public bool IsInFlame(Vector3 position)
{
    float distance = Vector3.Distance(transform.position, position);
    return distance < flameRadius;
}

0 reference
public Vector3 GetFlameCenter()
{
    return transform.position;
}

1 reference
public void ResetFlame()
{
    if (revertCoroutine != null)
    {
        StopCoroutine(revertCoroutine);
    }

    currentOuterColor = defaultOuterColor;
    targetOuterColor = defaultOuterColor;
    currentInnerColor = defaultInnerColor;
    targetInnerColor = defaultInnerColor;
    currentLightColor = defaultLightColor;
    targetLightColor = defaultLightColor;
}

```

Εικόνα 4.49 Μέθοδοι BoostLight(), RevertToDefaultColor(), ResetFlame()

4.4.7 Αρχείο GameManager.cs

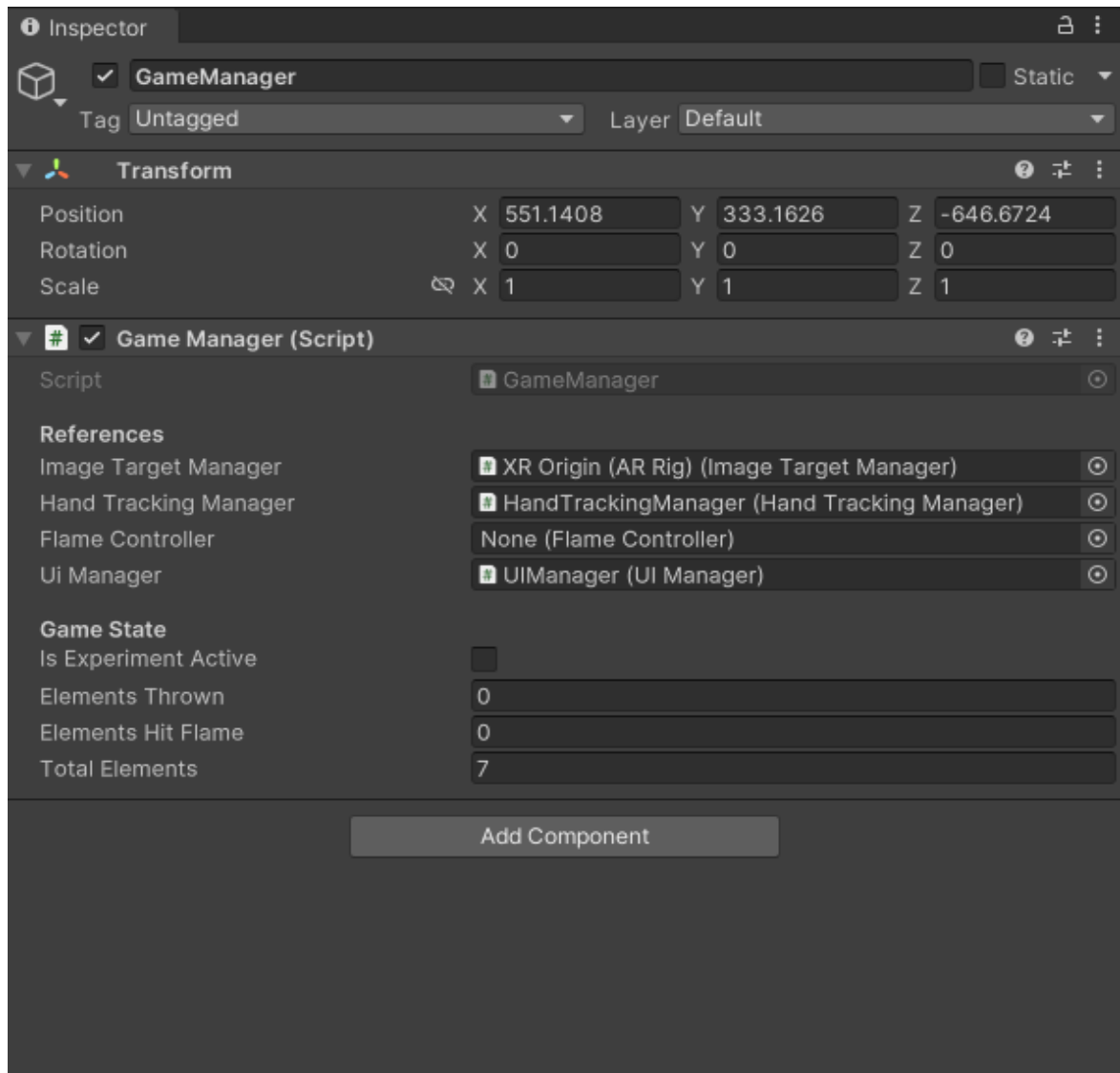
Λειτουργεί ως κεντρικός συντονιστής κατάστασης. Παρακολουθεί την ανίχνευση παρουσίας βιβλίων και την κατάσταση ενεργοποίησης πειραμάτων, επεξεργάζεται μετρήσεις όπως αποτυχίες και επιτυχίες και παρέχει εντολές επαναφοράς και επανεκκίνησης για να φέρει το σύστημα σε ένα ασφαλές σημείο εκκίνησης. Τα κύρια πεδία του GameManager είναι:

Κύρια πεδία:

- Singleton: static GameManager Instance.

Κεφαλαίο 4

- Κατάσταση: `isExperimentActive`.
- Μετρικές: `elementsThrown`, `elementsHitFlame`, `totalElements`.
- Αναφορές: `ImageTargetManager`, `HandTrackingManager`, `FlameController`, `UIManager`.
- Ρυθμίσεις σκηνών/permissions (όπου υπάρχουν).



Εικόνα 4.50 GameManager.cs

Μέθοδοι

Οι μέθοδοι του GameManager απεικονίζονται στις εικόνες 4.51 έως 4.53 και είναι οι εξής:

- **Awake()**
Ορίζει singleton και (αν χρησιμοποιείται) DontDestroyOnLoad για επιβίωση μεταξύ σκηνών.
- **Start()**
Ζητά permissions (κάμερα σε Android) και καλεί FindReferences().

```

Unity Message | 0 references
private void Awake()
{
    if (Instance == null)
    {
        Instance = this;
        DontDestroyOnLoad(gameObject);
    }
    else
    {
        Destroy(gameObject);
        return;
    }
}

Unity Message | 0 references
private void Start()
{
    RequestPermissions();

    FindReferences();

    ResetGameState();
}

```

Εικόνα 4.51 Μέθοδοι Awake(), Start()

- **FindReferences()**
Εντοπίζει managers στη σκηνή όταν δεν έχουν περαστεί από Inspector.
- **OnBookDetected()**
Θέτει isExperimentActive = true και ενημερώνει UI ώστε να ξεκινήσει το πείραμα.
- **OnBookLost()**
Θέτει isExperimentActive = false και επιστρέφει την εφαρμογή σε κατάσταση σάρωσης.

```

2 references
private void FindReferences()
{
    if (imageTargetManager == null)
        imageTargetManager = FindObjectOfType<ImageTargetManager>();

    if (handTrackingManager == null)
        handTrackingManager = FindObjectOfType<HandTrackingManager>();

    if (flameController == null)
        flameController = FindObjectOfType<FlameController>();

    if (uiManager == null)
        uiManager = FindObjectOfType<UIManager>();
}

1 reference
public void OnBookDetected()
{
    isExperimentActive = true;

    Debug.Log("[GameManager] Book detected - Experiment started!");

    uiManager?.HideScanPrompt();
}

1 reference
public void OnBookLost()
{
    isExperimentActive = false;

    Debug.Log("[GameManager] Book lost - Experiment paused");

    uiManager?.ShowScanPrompt();
}

```

Εικόνα 4.52 Μέθοδοι FindReferences(), OnBookDetected(), OnBookLost()

- **OnElementThrown(ElementDataelement,boolhitFlame)**
Ενημερώνει μετρικές και μπορεί να ενεργοποιήσει λογική ολοκλήρωσης.
- **ResetGameState()**
Μηδενίζει counters/flags για καθαρή επανεκκίνηση του σεναρίου.
- **RestartExperiment()**
Καλεί reset state και ενεργοποιεί:
 - respawn στοιχείων (ImageTargetManager.RespawnElements()),

- reset φλόγας (FlameController.ResetFlame()),
- επαναφορά UI σε σωστή κατάσταση.

```

@reference
public void OnElementThrown(ElementData element, bool hitFlame)
{
    elementsThrown++;

    if (hitFlame)
    {
        elementsHitFlame++;
        Debug.Log($"[GameManager] Element hit flame! Score: {elementsHitFlame}/{elementsThrown}");
    }
    else
    {
        Debug.Log($"[GameManager] Element missed flame. Score: {elementsHitFlame}/{elementsThrown}");
    }

    if (elementsThrown >= totalElements)
    {
        OnExperimentComplete();
    }
}

1 reference
private void OnExperimentComplete()
{
    Debug.Log($"[GameManager] Experiment complete! Final score: {elementsHitFlame}/{elementsThrown}");

    float accuracy = (float)elementsHitFlame / elementsThrown * 100f;
    string message;

    if (accuracy >= 90f)
        message = "Excellent work, chemist! 🏆";
    else if (accuracy >= 70f)
        message = "Great job! Keep practicing! 🏆";
    else if (accuracy >= 50f)
        message = "Good effort! Try again! 🏆";
    else
        message = "Keep practicing! You'll get better! 🏆";
}

4 references
public void ResetGameState()
{
    elementsThrown = 0;
    elementsHitFlame = 0;
    isExperimentActive = false;
}

8 references
public void RestartExperiment()
{
    ResetGameState();

    ImageTargetManager?.RespawnElements();

    FlameController?.ResetFlame();

    Debug.Log($"[GameManager] Experiment restarted");
}

```

Εικόνα 4.53 Μέθοδοι OnElementThrown(), ResetGameState(), RestartExperiment()

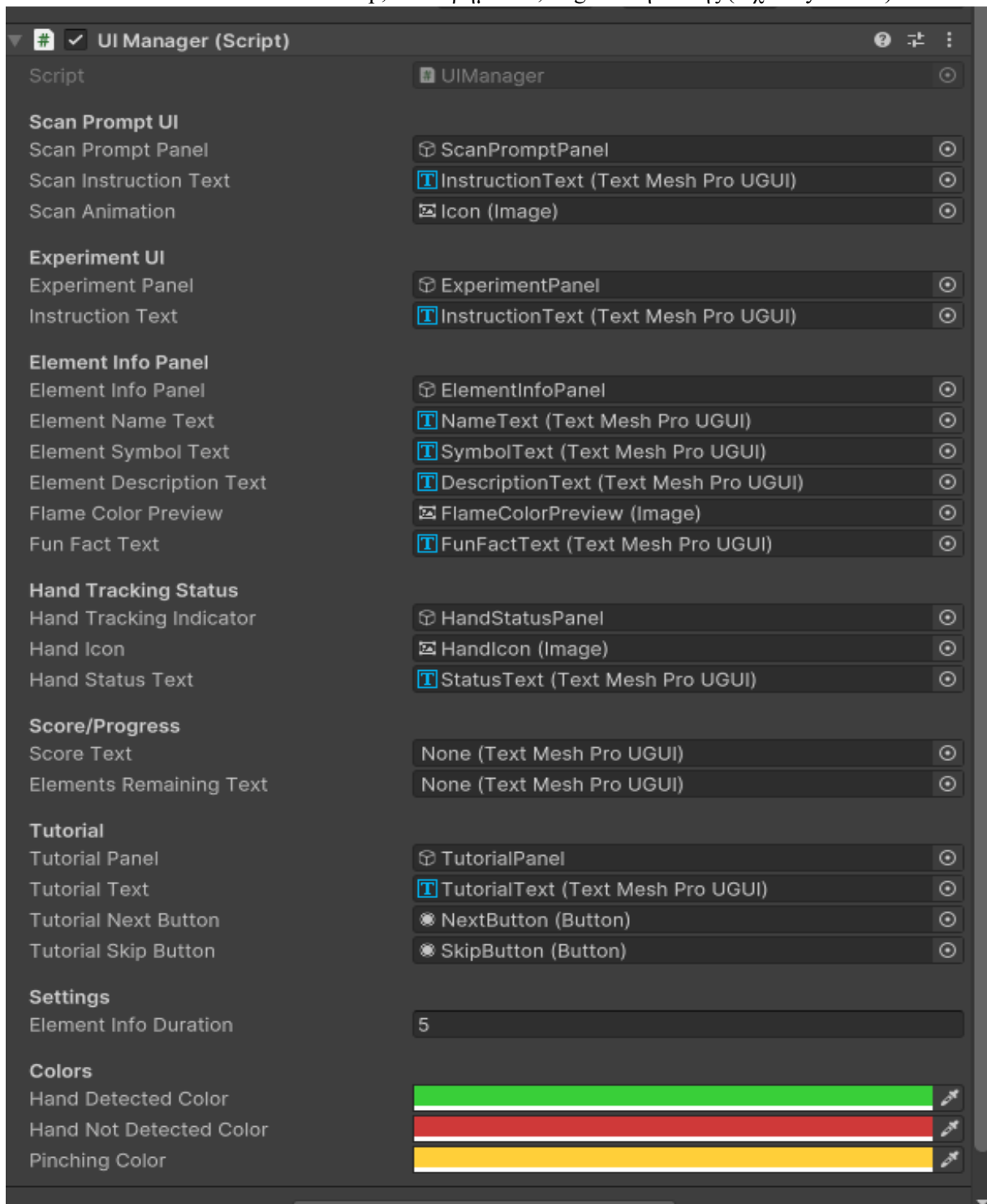
4.4.8 Αρχείο UIManager.cs

Επιβλέπει όλες τις πτυχές των διεπαφών χρήστη, συμπεριλαμβανομένων των προτροπών σάρωσης, της οθόνης πληροφοριών στοιχείων, του προσδιορισμού της κατάστασης των χεριών (συμπεριλαμβανομένων των σταδίων ανίχνευσης και πιασήματος), των σταδιακών φάσεων εκμάθησης και των προσωρινών μηνυμάτων ανατροφοδότησης. Τα πεδία του UIManager αποτελούνται από:

Κύρια πεδία:

- Panels: scanPromptPanel, experimentPanel, elementInfoPanel, tutorialPanel.

- UI στοιχεία: Text (όνομα/σύμβολο/περιγραφή), icons, progress/score.
- Hand status: indicator (text/color), colors για detected/pinch/not detected.
- Tutorial state: currentTutorialStep, λίστα βημάτων, flag αποθήκευσης (π.χ. PlayerPrefs).



Εικόνα 4.54 UIManager.cs

Μέθοδοι

Οι μέθοδοι του UIManager απεικονίζονται στις εικόνες 4.55 έως 4.59 και είναι οι εξής:

- **ShowScanPrompt()/HideScanPrompt()**

Εναλλαγή οθονών ανάλογα με το αν ανιχνεύεται το βιβλίο.

```

2 references
public void ShowScanPrompt()
{
    if (scanPromptPanel != null)
    {
        scanPromptPanel.SetActive(true);
        if (scanInstructionText != null)
        {
            scanInstructionText.text = "Point your camera at the\nChemistry book cover";
        }
    }

    if (experimentPanel != null)
        experimentPanel.SetActive(false);
}

1 reference
public void HideScanPrompt()
{
    if (scanPromptPanel != null)
        scanPromptPanel.SetActive(false);

    if (experimentPanel != null)
    {
        experimentPanel.SetActive(true);
        if (instructionText != null)
        {
            instructionText.text = "Pinch to grab an element\nand throw it into the flame!";
        }
    }
}

```

Εικόνα 4.55 Μέθοδοι ShowScanPrompt()/HideScanPrompt()

- **ShowElementInfo(ElementDatadata)/HideElementInfo()**

Γεμίζει και εμφανίζει panel με στοιχεία από ElementData (όνομα, σύμβολο, περιγραφή, fun fact).

```

1 reference
public void ShowElementInfo(ElementData element)
{
    if (element == null) return;

    if (elementInfoPanel != null)
    {
        elementInfoPanel.SetActive(true);
        StartCoroutine(AnimatePanelIn(elementInfoPanel));

        if (elementNameText != null)
            elementNameText.text = element.elementName;

        if (elementSymbolText != null)
            elementSymbolText.text = element.symbol;

        if (elementDescriptionText != null)
            elementDescriptionText.text = element.description;

        if (flameColorPreview != null)
            flameColorPreview.color = element.flameOuterColor;

        if (funFactText != null && !string.IsNullOrEmpty(element.funFact))
        {
            funFactText.gameObject.SetActive(true);
            funFactText.text = "Fun Fact: " + element.funFact;
        }
        else if (funFactText != null)
        {
            funFactText.gameObject.SetActive(false);
        }
    }

    if (hideInfoCoroutine != null)
        StopCoroutine(hideInfoCoroutine);
    hideInfoCoroutine = StartCoroutine(HideInfoAfterDelay(elementInfoDuration));
}

1 reference
private IEnumerator HideInfoAfterDelay(float delay)
{
    yield return new WaitForSeconds(delay);
    HideElementInfo();
}

2 references
public void HideElementInfo()
{
    if (elementInfoPanel != null)
    {
        StartCoroutine(AnimatePanelOut(elementInfoPanel));
    }
}

```

Εικόνα 4.56 Μέθοδοι ShowElementInfo(ElementDatadata)/HideElementInfo()

- **UpdateHandTrackingStatus()/SetHandStatus(...)**

Διαβάζει κατάσταση από HandTrackingManager και ενημερώνει indicator (π.χ. “Hand Detected”, “Pinching”, “Show Hand”).

```

1 reference
private void UpdateHandTrackingStatus()
{
    if (handTrackingIndicator == null) return;

    HandTrackingManager handTracker = HandTrackingManager.Instance;

    if (handTracker != null && handTracker.IsHandDetected)
    {
        if (handTracker.IsPinching())
        {
            SetHandStatus(pinchingColor, "Grabbing");
        }
        else
        {
            SetHandStatus(handDetectedColor, "Hand Ready");
        }
    }
    else
    {
        SetHandStatus(handNotDetectedColor, "Show Hand");
    }
}

3 references
private void SetHandStatus(Color color, string status)
{
    if (handIcon != null)
        handIcon.color = color;

    if (handStatusText != null)
        handStatusText.text = status;
}

```

Εικόνα 4.57 Μέθοδοι UpdateHandTrackingStatus()/SetHandStatus(...)

- **StartTutorial()/NextTutorialStep()/SkipTutorial()/CompleteTutorial()**

Διαχειρίζεται tutorial βημάτων και αποθηκεύει ολοκλήρωση για να μην εμφανίζεται κάθε φορά.

```

2 references
public void StartTutorial()
{
    Debug.Log("[UIManager] Starting tutorial...");

    tutorialStep = 0;

    if (tutorialPanel == null)
    {
        Debug.LogError("[UIManager] Tutorial Panel is not assigned!");
        return;
    }

    tutorialPanel.SetActive(true);
    ShowTutorialStep();
}

2 references
private void ShowTutorialStep()
{
    Debug.Log($"[UIManager] Showing tutorial step {tutorialStep + 1} of {tutorialSteps.Length}");

    if (tutorialText != null && tutorialStep < tutorialSteps.Length)
    {
        tutorialText.text = tutorialSteps[tutorialStep];
    }

    // Update Next button text for last step
    if (tutorialNextButton != null)
    {
        TextMeshProUGUI buttonText = tutorialNextButton.GetComponentInChildren<TextMeshProUGUI>();
        if (buttonText != null)
        {
            buttonText.text = (tutorialStep >= tutorialSteps.Length - 1) ? "Start!" : "Next";
        }

        // Also try regular Text component
        Text regularText = tutorialNextButton.GetComponentInChildren<Text>();
        if (regularText != null)
        {
            regularText.text = (tutorialStep >= tutorialSteps.Length - 1) ? "Start!" : "Next";
        }
    }
}

1 reference
public void NextTutorialStep()
{
    Debug.Log($"[UIManager] NextTutorialStep called. Current step: {tutorialStep}");

    tutorialStep++;

    if (tutorialStep >= tutorialSteps.Length)
    {
        CompleteTutorial();
    }
    else
    {
        ShowTutorialStep();
    }
}

1 reference
public void SkipTutorial()
{
    Debug.Log("[UIManager] SkipTutorial called");
    CompleteTutorial();
}

2 references
private void CompleteTutorial()
{
    Debug.Log("[UIManager] Tutorial completed!");

    if (tutorialPanel != null)
        tutorialPanel.SetActive(false);

    PlayerPrefs.SetInt("TutorialCompleted", 1);
    PlayerPrefs.Save();
}

0 references
public void ResetTutorial()
{
    PlayerPrefs.DeleteKey("TutorialCompleted");
    StartTutorial();
}

```

Εικόνα 4.58 Μέθοδοι StartTutorial()/NextTutorialStep()/SkipTutorial()/CompleteTutorial()

- **ShowMessage(stringmsg,floatduration)**
Εμφανίζει προσωρινό μήνυμα (π.χ. οδηγία) και μετά επαναφέρει το βασικό UI κείμενο.

```

0 references
public void ShowMessage(string message, float duration = 2f)
{
    StartCoroutine(ShowTemporaryMessage(message, duration));
}

1 reference
private IEnumerator ShowTemporaryMessage(string message, float duration)
{
    if (instructionText != null)
    {
        string originalText = instructionText.text;
        instructionText.text = message;
        yield return new WaitForSeconds(duration);
        instructionText.text = originalText;
    }
}

```

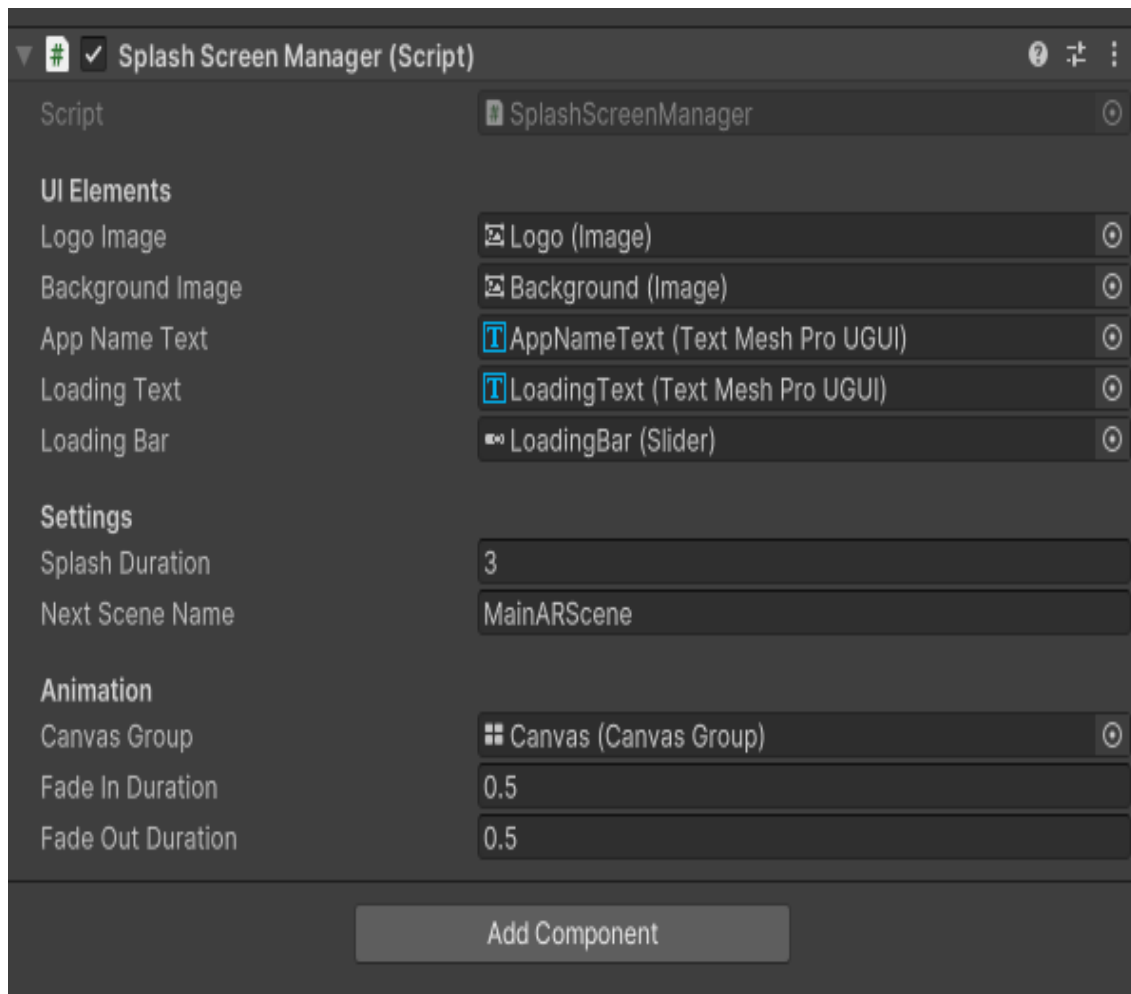
Εικόνα 4.59 Μέθοδος ShowMessage(stringmsg,floatduration)

4.4.9 Αρχείο SplashScreenManager.cs

Ομαλή εκκίνηση της εφαρμογής: εμφανίζεται η οθόνη εκκίνησης, προσομοιώνεται η διαδικασία φόρτωσης και πραγματοποιείται η μετάβαση στην κύρια σκηνή AR, βελτιώνοντας έτσι την εμπειρία του χρήστη και αποτρέποντας την ξαφνική είσοδο στην κάμερα/AR. Τα κύρια του πεδία είναι:

Κύρια πεδία:

- UI: CanvasGroup (alpha για fade), progress bar, loading text.
- Χρονισμοί: fadeInDuration, loadingDuration, fadeOutDuration.
- Σκηνή στόχος: nextSceneName.



Εικόνα 4.60 SplashScreenManager.cs

Μέθοδοι

Οι μέθοδοι του SplashScreenManager απεικονίζονται στην εικόνα 4.61 είναι οι εξής:

- **Start()**
Ξεκινά την coroutine SplashSequence().
- **SplashSequence()**
Εκτελεί τη σειρά:
 1. FadeIn()
 2. SimulateLoading()
 3. FadeOut()
 4. φόρτωση επόμενης σκηνής
- **FadeIn()/FadeOut()**
Ομαλή αλλαγή alpha στο UI.
- **SimulateLoading()**
Ενημερώνει progress bar και μηνύματα φόρτωσης ώστε ο χρήστης να έχει αίσθηση προόδου.

```

1 reference
private IEnumerator SplashSequence()
{
    if (canvasGroup != null)
        canvasGroup.alpha = 0f;

    if (loadingBar != null)
        loadingBar.value = 0f;

    yield return StartCoroutine(FadeIn());

    yield return StartCoroutine(SimulateLoading());

    yield return StartCoroutine(FadeOut());

    SceneManager.LoadScene(nextSceneName);
}

1 reference
private IEnumerator FadeIn()
{
    float elapsed = 0f;

    while (elapsed < fadeInDuration)
    {
        elapsed += Time.deltaTime;

        if (canvasGroup != null)
            canvasGroup.alpha = elapsed / fadeInDuration;

        yield return null;
    }

    if (canvasGroup != null)
        canvasGroup.alpha = 1f;
}

1 reference
private IEnumerator FadeOut()
{
    float elapsed = 0f;

    while (elapsed < fadeOutDuration)
    {
        elapsed += Time.deltaTime;

        if (canvasGroup != null)
            canvasGroup.alpha = 1f - (elapsed / fadeOutDuration);

        yield return null;
    }

    if (canvasGroup != null)
        canvasGroup.alpha = 0f;
}

1 reference
private IEnumerator SimulateLoading()
{
    string[] loadingMessages = new string[]
    {
        "Initializing AR...",
        "Loading Chemistry Data...",
        "Preparing Elements...",
        "Setting Up Experiments...",
        "Almost Ready..."
    };

    float elapsed = 0f;
    int messageIndex = 0;

    while (elapsed < splashDuration)
    {
        elapsed += Time.deltaTime;
        float progress = elapsed / splashDuration;

        if (loadingBar != null)
            loadingBar.value = progress;

        int newIndex = Mathf.FloorToInt(progress * loadingMessages.Length);
        newIndex = Mathf.Clamp(newIndex, 0, loadingMessages.Length - 1);

        if (newIndex != messageIndex)
        {
            messageIndex = newIndex;
            if (loadingText != null)
                loadingText.text = loadingMessages[messageIndex];
        }

        yield return null;
    }

    if (loadingBar != null)
        loadingBar.value = 1f;

    if (loadingText != null)
        loadingText.text = "Ready!";

    yield return new WaitForSeconds(0.3f);
}

```

Εικόνα 4.61 Μέθοδοι SplashSequence(), FadeIn()/FadeOut(), SimulateLoading()

5 Συμπεράσματα και Μελλοντικές Βελτιώσεις

5.1 Συμπεράσματα

Ο στόχος αυτής της διατριβής ήταν ο σχεδιασμός και η ανάπτυξη μιας εκπαιδευτικής εφαρμογής επαυξημένης πραγματικότητας με αναγνώριση χειρονομιών, η οποία θα προσομοίωνε χημικά πειράματα σε μαθητές της δεύτερης τάξης του γυμνασίου. Η υλοποίηση και οι επακόλουθες δοκιμές της εφαρμογής απέδειξαν ότι η επαυξημένη πραγματικότητα και οι τεχνολογίες φυσικής αλληλεπίδρασης μπορούν να χρησιμοποιηθούν αποτελεσματικά ως βοηθητικά μέσα διδασκαλίας των μαθημάτων φυσικών επιστημών στο πρόγραμμα σπουδών των φυσικών επιστημών.

Η εφαρμογή που χρησιμοποιήθηκε απεικονίζει το πείραμα καύσης των χημικών στοιχείων και, ως εκ τούτου, δημιουργείται μια άμεση συσχέτιση μεταξύ των φυσικών χειρονομιών του μαθητή και του τελικού οπτικού αποτελέσματος. Η ενσωμάτωση του κινητήρα Unity με το AR Foundation και το ARCore αύξησε την χωρική παρακολούθηση, ενώ ταυτόχρονα η ενσωμάτωση του Manomotion SDK υποστήριξε την απλή αναγνώριση χειρονομιών χωρίς να χρειάζεται να βασίζεται σε εξωτερικό υλικό. Η εφαρμογή είναι επομένως όχι μόνο χρησιμοποιήσιμη, αλλά και κατάλληλη για χρήση στο σχολείο.

Η αρχιτεκτονική του συστήματος ήταν τεχνικά κλιμακωτή και λειτουργική. Ο σαφής ορισμός των υποσυστημάτων, όπως η αναγνώριση στόχων, η αλληλεπίδραση κινήσεων, η πειραματική λογική, η οπτική ανατροφοδότηση και η διαχείριση της διεπαφής χρήστη, επέτρεψε την αποτελεσματική ανάπτυξη, συντήρηση και μελλοντική ανάπτυξη. Η χρήση μηχανισμών, π.χ. αναγέννηση στοιχείων και αυτόματη επαναφορά φλόγας, εγγυήθηκε την επαναληψιμότητα των πειραμάτων, καθώς και τη σταθερότητα της εμπειρίας του χρήστη.

Από εκπαιδευτική άποψη, η εφαρμογή υποστηρίζει τη βιωματική μάθηση, επιτρέποντας στους μαθητές να ξεπεράσουν την παθητική παρατήρηση και να συμμετέχουν ενεργά στη διαδικασία του πειράματος. Η χρήση φυσικών χειρονομιών ενισχύει την διαισθητικότητα και μειώνει τις γνωστικές απαιτήσεις. Από την άλλη πλευρά, η οπτική ανατροφοδότηση διευκολύνει την κατανόηση των σχέσεων αιτίου-αποτελέσματος. Η εφαρμογή παρέχει επομένως ένα ασφαλές πειραματικό περιβάλλον, ειδικά σε περιπτώσεις όπου η φυσική πραγματικότητα είναι πιο πρακτική ή επικίνδυνη.

Σε γενικές γραμμές, η παρούσα εργασία δείχνει ότι η επαυξημένη πραγματικότητα σε συνδυασμό με την αναγνώριση χειρονομιών αποτελεί ένα ισχυρό εργαλείο για τη διευκόλυνση της εκπαιδευτικής διαδικασίας, ιδίως σε επιστήμες που είναι γνωστικά απαιτητικές, όπως η χημεία, όπου η οπτικοποίηση και η αλληλεπίδραση διαδραματίζουν καθοριστικό ρόλο.

5.2 Μελλοντικές Βελτιώσεις και Επεκτάσεις

Αν και η εφαρμογή πληροί τους αρχικούς στόχους που τέθηκαν, υπάρχει ακόμη πολύ περιθώριο για προσθήκες και περαιτέρω ανάπτυξη τόσο σε τεχνολογικό όσο και σε παιδαγωγικό επίπεδο.

Από τεχνολογική άποψη, μια ιδιαίτερα ενδιαφέρουσα βελτίωση θα ήταν η επέκταση του εύρους της αναγνώρισης χειρονομιών ώστε να καλύπτει πιο σύνθετες χειρονομίες υψηλού επιπέδου, όπως η αμφιδέξια αλληλεπίδραση, με σκοπό να διευρυνθεί το εύρος των ενεργειών που μπορεί να εκτελέσει ο χρήστης. Επιπλέον, η ενσωμάτωση συσκευών ανίχνευσης βάθους ή άλλων δυνατοτήτων παρακολούθησης των χειρών θα συνέβαλε στην αύξηση της ακρίβειας της αναγνώρισης χειρονομιών και στη μείωση της ευαισθησίας στις αλλαγές του φωτισμού.

Επίσης, για την παροχή ποσοτικών δεδομένων για την αξιολόγηση των μοτίβων μάθησης ή για την καθοδήγηση αυτών των βελτιώσεων που γίνονται στην εφαρμογή, θα ήταν χρήσιμο ένα σύστημα μέτρησης της αλληλεπίδρασης του χρήστη: ο αριθμός των προσπαθειών, το ποσοστό επιτυχίας, ο χρόνος ολοκλήρωσης μιας εργασίας κ.λπ.

Για την επίτευξη του παιδαγωγικού σκοπού, το λογισμικό θα μπορούσε να επεκταθεί στη χρήση περισσότερων χημικών πειραμάτων, βαθμονομημένων σε διαφορετικά επίπεδα δυσκολίας ανάλογα με την ηλικία ή τις γνωστικές ικανότητες των μαθητών. Ταυτόχρονα, η αποτελεσματικότητα της πλατφόρμας όσον αφορά τη διδασκαλία, γενικά, θα αυξηθεί με την προσθήκη ορισμένων οργανωμένων μαθησιακών διαδρομών, τεστ ή σύντομων εξετάσεων μετά το πείραμα.

Τελικά, η πιθανή μελλοντική μελέτη θα ήταν η διεξαγωγή εμπειρικής αξιολόγησης της χρήσης σε πραγματικά σχολικά περιβάλλοντα και η συμμετοχή τόσο των μαθητών όσο και των εκπαιδευτικών. Η συστηματική χρήση ανατροφοδότησης από πραγματικούς χρήστες θα παρέχει πρακτική ανατροφοδότηση για τη βελτίωση της εμπειρίας του χρήστη και θα δικαιολογήσει τα παιδαγωγικά αποτελέσματα του λογισμικού.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Azuma, R. T. , *A survey of augmented reality*, *Presence: Teleoperators & Virtual Environments*, 6(4), 355–385 (1997).
- [2] Azuma, R. T., Bailiot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B. , Recent advances in augmented reality, *IEEE Computer Graphics and Applications*, 21(6), 34–47 (2001) .
- [3] Bacca, J., Baldiris, S., Fabregat, R., Graf, S., & Kinshuk, Augmented reality trends in education: A systematic review of research and applications, *Educational Technology & Society*, 17(4), 133–149 (2014).
- [4] K. Kimura and A. Lipeles, “Fuzzy controller component,” U. S. Patent 14, 860,040, 14 Dec., 2006.
- [5] Caudell, T. P., & Mizell, D. W, Augmented reality: An application of heads-up display technology to manual manufacturing processes, *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, 659–669 (1992).
- [6] Ibáñez, M. B., & Delgado-Kloos, C, Augmented reality for STEM learning: A systematic review, *Computers & Education*, 123, 109–123 (2018).
- [7] Milgram, P., & Kishino, F. , A taxonomy of mixed reality visual displays, *IEICE Transactions on Information and Systems*, E77-D(12), 1321–1329 (1994).
- [8] Radianti, J., Majchrzak, T. A., Fromm, J., & Wohlgenannt, I. , A systematic review of immersive virtual reality applications for higher education, *Computers & Education*, 147, 103778 (2020).
- [9] Dourish, P. , *Where the action is: The foundations of embodied interaction*, MIT Press (2001).
- [10] Vatavu, R.-D., Wobbrock, J. O., & Wilson, A. D. , *Gesture interaction*. Springer, (2017).
- [11] Norman, D. A. , *The design of everyday things* (Revised and expanded ed.), Basic Books, (2013).
- [12] Akçayır, M., & Akçayır, G. , Advantages and challenges associated with augmented reality for education: A systematic review of the literature, *Educational Research Review*, 20, 1–11 (2017).
- [13] «Al-Saedi, Ahmed Kadem Hamed & Al-Asadi, Abbas. Survey of Hand Gesture Recognition Systems. *Journal of Physics: Conference Series*. 1294. 042003. 10.1088/1742- 6596/1294/4/042003. (2019)».
- [14] Mitra, S., & Acharya, T. , *Gesture recognition: A survey*. **IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews**, 37(3), 311–324 (2007).
- [15] Rautaray, S. S., & Agrawal, A. , Vision-based hand gesture recognition for human–computer interaction: A survey, *Artificial Intelligence Review*, 43(1), 1–54 (2015).
- [16] «ManoMotion AB, “ManoMotion Unity SDK Documentation,” (2024).,».
- Available: <https://www.manomotion.com/products/unity/>.
- [17] «Unity Technologies, "AR Foundation documentation," Unity Manual, (2023).,».
- Available: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@6.1/manual/index.html>.

- [18] «V. Bazarevsky et al., "MediaPipe Hands: On-device Real-time Hand Tracking," arXiv preprint, arXiv:2006.10214, (2020).,». Available: <https://arxiv.org/abs/2006.10214>.
- [19] «Sutherland, I. E., A head-mounted three dimensional display. In Proceedings of the Fall Joint Computer Conference (pp. 757–764). Association for Computing Machinery (1968) .