

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός και Ανάπτυξη Εκπαιδευτικής Εφαρμογής  
Επαυξημένης Πραγματικότητας (AR) για την Ιστορία Ε' Δημοτικού



**Του φοιτητή**  
**Χρήστος Καριπιάν**  
**Αρ. Μητρώου: 154592**

**Επιβλέπων**  
**Ονοματεπώνυμο Ευκλείδης**  
**Κεραμόπουλος**  
**Βαθμίδα Καθηγητής**

Ημερομηνία 31/05/2026

Τίτλος Δ. Ε.: *Σχεδιασμός και Ανάπτυξη Εκπαιδευτικής Εφαρμογής Επαυξημένης Πραγματικότητας (AR)  
για την Ιστορία Ε' Δημοτικού*

Κωδικός Δ.Ε.: 23184

Όνοματεπώνυμο φοιτητή: Χρήστος Καριπιάν

Όνοματεπώνυμο εισηγητή: Κεραμόπουλος Ευκλείδης

Ημερομηνία ανάληψης Δ.Ε.: 01/04/2022

Ημερομηνία περάτωσης Δ.Ε.: 31/05/2026

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 Χρήστου Καριπιάν 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

## Πρόλογος

Η παρούσα πτυχιακή εργασία πραγματεύεται τη δημιουργία ενός εκπαιδευτικού παιχνιδιού Επαυξημένης Πραγματικότητας (AR) για το μάθημα της Ιστορίας της Ε' Δημοτικού. Η ενασχόληση με το συγκεκριμένο αντικείμενο αποτέλεσε άμεση επιλογή μου, καθώς συνδύαζε το προσωπικό μου ενδιαφέρον για την ανάπτυξη βιντεοπαιχνιδιών (Game Development) με την επιθυμία μου να μελετήσω και να εφαρμόσω στην πράξη την τεχνολογία του AR.

Στόχος της εφαρμογής είναι να λειτουργήσει συμπληρωματικά στο παραδοσιακό σχολικό βιβλίο. Εμπλουτίζοντάς το με 3D γραφικά και μηχανισμούς παιχνιδιού (gamification), η διαδικασία της μάθησης γίνεται πιο διαδραστική και ελκυστική για μαθητές που είναι εξοικειωμένοι με τα ψηφιακά περιβάλλοντα.

Η συνολική διαδικασία ανάπτυξης του παιχνιδιού στη μηχανή γραφικών Unity αποτέλεσε μια απαιτητική αλλά εξαιρετικά δημιουργική προγραμματιστική πρόκληση. Στις σελίδες που ακολουθούν, καταγράφεται αναλυτικά η τεχνική υλοποίηση της εφαρμογής, ο σχεδιασμός της διεπαφής (UI/UX) και η αρχιτεκτονική του κώδικα.

## Περίληψη

Η παρούσα πτυχιακή εργασία παρουσιάζει τον σχεδιασμό και την ανάπτυξη μιας εκπαιδευτικής εφαρμογής Επαυξημένης Πραγματικότητας (Augmented Reality - AR) με τη μορφή διαδραστικού παιχνιδιού. Απευθύνεται σε μαθητές της Ε' Δημοτικού, με γνωστικό αντικείμενο την ιστορία και τον πολιτισμό των Ρωμαίων και των Ελλήνων. Στόχος της εφαρμογής είναι να εμπλουτίσει την παραδοσιακή διδασκαλία με μια βιωματική διάσταση (active learning) μέσω της παιχνιδιοποίησης (gamification), λειτουργώντας συμπληρωματικά στο σχολικό βιβλίο. Η εφαρμογή αναπτύχθηκε στη μηχανή γραφικών Unity 3D με τη χρήση της γλώσσας C#, ενώ για τις λειτουργίες AR αξιοποιήθηκε το Vuforia Engine SDK. Ως πλατφόρμα στόχευσης επιλέχθηκε το λειτουργικό σύστημα Android. Η διεπαφή χρήστη περιλαμβάνει θεωρητικά μαθήματα με διαδραστικούς συνδέσμους (clickable words) και αξιολογήσεις πολλαπλών επιπέδων, οι οποίες ενσωματώνουν μηχανισμούς χρόνου και ζώων. Οι ερωτήσεις ποικίλλουν από δύο επιλογών (κείμενο) και επιλογής σωστής εικόνας, μέχρι προηγμένες δοκιμασίες AR όπου ο μαθητής σκανάρει το σχολικό βιβλίο, περιστρέφει τρισδιάστατα (3D) μοντέλα και αλληλεπιδρά με αυτά.

# **Design and Development of an Educational Augmented Reality (AR) Game for the History of Romans and Greeks 5th Grade Primary School**

Christos Karipian

## **Abstract**

This thesis presents the design and development of an educational Augmented Reality (AR) application in the form of an interactive game. It is aimed at 5th-grade students, focusing on the history and culture of the Romans and Greeks. The main objective of the application is to enrich traditional teaching with an experiential dimension (active learning) through gamification, complementing — rather than replacing — the school textbook. The application was developed in the Unity 3D game engine using the C# programming language, while the Vuforia Engine SDK was utilized for the AR functionalities. The Android operating system was chosen as the target platform. The user interface includes theoretical lessons with interactive links (clickable words) and multi-level assessments, which incorporate a timer and a lives system. The quizzes range from two-choice text questions and image selection to advanced AR tasks where the student scans the school textbook, rotates three-dimensional (3D) models, and interacts with them.

## Ευχαριστίες

Η ολοκλήρωση της παρούσας πτυχιακής εργασίας δεν θα ήταν εφικτή χωρίς την υποστήριξη και την καθοδήγηση των ανθρώπων που στάθηκαν δίπλα μου καθ' όλη τη διάρκεια της εκπόνησής της.

Αρχικά, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή μου, κ. Κεραμόπουλο Ευκλείδη, για την πολύτιμη καθοδήγηση, τον χρόνο που μου αφιέρωσε και τις καίριες υποδείξεις του σε κάθε στάδιο της ανάπτυξης της εφαρμογής και της συγγραφής του κειμένου. Η συμβολή του ήταν καθοριστική για την επίλυση προκλήσεων και τη διαμόρφωση του τελικού αποτελέσματος.

Επίσης, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για την αμέριστη συμπαράσταση, την υπομονή και την ψυχολογική υποστήριξη που μου παρείχαν καθ' όλη τη διάρκεια των σπουδών μου.

Τέλος, ευχαριστώ θερμά τους φίλους και συμφοιτητές μου που δοκίμασαν τα πρώτα πρωτότυπα της εφαρμογής, παρέχοντάς μου εποικοδομητικά σχόλια για τη βελτίωση του UI και του UX.

# Περιεχόμενα

Πρόλογος.....	iii
Περίληψη.....	iv
Abstract .....	v
Ευχαριστίες.....	vi
Περιεχόμενα .....	vii
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ (ΕΙΚΟΝΩΝ).....	x
Συντομογραφίες.....	xii
Κεφάλαιο 1ο: Εισαγωγή .....	1
1.1 Περιγραφή και Αντικείμενο της Εργασίας.....	1
1.2 Σκοπός και Στόχοι.....	1
1.3 Συνοπτική Αναφορά στις Τεχνολογίες .....	2
1.4 Δομή της Πτυχιακής Εργασίας .....	2
1.5 Μεθοδολογία Ανάπτυξης .....	3
1.6 Επίλογος Κεφαλαίου.....	5
Κεφάλαιο 2ο: Θεωρητικό Υπόβαθρο και Τεχνολογίες Ανάπτυξης .....	7
2.1 Εισαγωγή Κεφαλαίου .....	7
2.2 Ο Διαχωρισμός των Τεχνολογιών: AR, VR, MR και XR .....	7
2.3 Επαυξημένη Πραγματικότητα (Augmented Reality - AR) .....	8
2.3.1 Ιστορική Εξέλιξη της Τεχνολογίας .....	10
2.3.2 Η Επαυξημένη Πραγματικότητα στην Εκπαίδευση .....	10
2.4 Παιχνιδοποίηση (Gamification) στην Εκπαίδευση.....	11
2.5 Μηχανή Γραφικών Unity 3D .....	13
2.6 Αναγνώριση Εικόνας (Image Tracking).....	14
2.7 Ο Μηχανισμός Extended Tracking και Σταθερότητας.....	15
2.8 Πλεονεκτήματα της Vuforia Engine .....	16
2.9 Λειτουργικό Σύστημα Android.....	17
2.10 Γλώσσα Προγραμματισμού C#.....	17
2.11 Επίλογος Κεφαλαίου.....	18
Κεφάλαιο 3ο: Η Παρουσίαση της Εφαρμογής .....	19
3.1 Εισαγωγή Κεφαλαίου .....	19
3.2 Κεντρικό Μενού και Βασικές Επιλογές.....	19
3.3 Το Μάθημα, Ιστορική Αναδρομή και Πηγές.....	20

3.4	Έναρξη Παιχνιδιού και Σύστημα Επιπέδων .....	22
3.5	Δομή και Κλιμάκωση των Επιπέδων .....	23
3.6	Τύποι Ερωτήσεων και Δοκιμασίες AR .....	25
3.7	Οθόνη Αποτελεσμάτων και Ανατροφοδότησης .....	28
3.8	Επίλογος Κεφαλαίου.....	29
Κεφάλαιο 4ο:	Ανάπτυξη και Προγραμματιστική Υλοποίηση.....	31
4.1	Εισαγωγή Κεφαλαίου .....	31
4.2	Διαχείριση Σκηνών και Διεπαφής (ManagerPanelController).....	31
4.3	Ιεραρχία Αντικειμένων στο Unity Editor (Scene Hierarchy).....	31
4.4	Αρχιτεκτονική Δεδομένων: JSON .....	32
4.5	Δόμηση Θεωρίας και Υπερσύνδεσμοι στο JSON.....	34
4.6	Σύστημα Προόδου και Αποθήκευσης (LevelSelectionController) .....	35
4.7	Δημιουργία Γραφικού Περιβάλλοντος με χρήση Prefabs.....	35
4.8	Δυναμική Φόρτωση Θεωρίας και Σελιδοποίηση (LessonManager).....	36
4.9	Σύστημα Ιστορικής Αναδρομής (LectureManager) .....	38
4.10	Σύστημα Διαδραστικών Συνδέσμων (LessonLinkHandler).....	38
4.11	Διαχείριση Τύπων Αλληλεπίδρασης στις Δοκιμασίες AR (ARPromptController).....	38
4.12	Δυναμική Φόρτωση Δεδομένων από JSON.....	39
4.13	Λογική Παιχνιδιού και Κουίζ (QuizManager).....	40
4.14	Χρονοκαθυστέρηση και Ασύγχρονη Εκτέλεση (Invoke Method) .....	41
4.15	Σύστημα Επιβράβευσης και Σερι Απαντήσεων (Streak & Time Bonus) .....	42
4.16	Αναλυτική Ανατροφοδότηση (Detailed Feedback).....	43
4.17	Ερωτήσεις Βασισμένες σε Εικόνες (Image-Based Questions) .....	43
4.18	Δυναμική Φόρτωση Πόρων (Resources.Load).....	44
4.19	Ενσωμάτωση Vuforia Engine, AR Camera και Διαδικασία Αναγνώρισης .....	46
4.20	AR Δοκιμασίες (ARPromptController) .....	48
4.21	Μηχανισμός Ομαλής Περιστροφής Τρισδιάστατων Μοντέλων (Smooth Rotation) .....	51
4.22	Οργάνωση 3D Μοντέλων στον Unity Editor.....	52
4.23	Εργαλείο Οπτικοποίησης Colliders (ColliderVisualizer).....	53
4.24	Επίλογος Κεφαλαίου.....	55
Κεφάλαιο 5ο:	Συμπεράσματα και Μελλοντικές Επεκτάσεις.....	57
5.1	Εισαγωγή Κεφαλαίου .....	57
5.2	Σύνοψη του Έργου.....	57
5.3	Προκλήσεις κατά την Ανάπτυξη .....	57
5.4	Μελλοντικές Επεκτάσεις .....	58

5.5	Επίλογος Κεφαλαίου.....	58
	ΒΙΒΛΙΟΓΡΑΦΙΑ .....	59
	ΠΑΡΑΡΤΗΜΑ Α: ΕΙΚΟΝΕΣ-ΣΤΟΧΟΙ (IMAGE TARGETS).....	61

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ (ΕΙΚΟΝΩΝ)

Σχήμα 1.1: Η εφαρμογή απευθύνεται σε μαθητές Δημοτικού, που εξοικειώνονται με την εκπαιδευτική ύλη μέσω διαδραστικών τεχνολογιών .....	2
Σχήμα 1.2: Ο γενικός κύκλος ανάπτυξης λογισμικού της μεθοδολογίας Agile .....	4
Σχήμα 2.1: Το συνεχές φάσμα Εικονικότητας-Πραγματικότητας (Reality-Virtuality Continuum) κατά τον Paul Milgram [1].....	8
Σχήμα 2.2: Συγκριτική σχηματική απεικόνιση των βασικών χαρακτηριστικών της Επαυξημένης (AR) και της Εικονικής Πραγματικότητας (VR).....	9
Σχήμα 2.3: Η τομή της παραδοσιακής μάθησης με τα στοιχεία σχεδιασμού παιχνιδιών που δημιουργεί την Παιχνιδοποίηση (Gamification).....	11
Σχήμα 2.4 Τα βασικά δομικά στοιχεία της Παιχνιδοποίησης (game design elements) που ενσωματώθηκαν στην εκπαιδευτική εφαρμογή .....	13
Σχήμα 2.5 Το λογότυπο της μηχανής γραφικών Unity .....	14
Σχήμα 2.6: Το λογότυπο του λογισμικού Επαυξημένης Πραγματικότητας Vuforia Engine .....	14
Σχήμα 2.7 Οπτικοποίηση των Feature Points (κίτρινοι σταυροί) στο περιβάλλον του Vuforia Target Manager.....	15
Σχήμα 2.8: Τα βασικά στάδια επεξεργασίας εικόνας και απόδοσης της μηχανής Vuforia Engine .....	16
Σχήμα 2.9 Η βάση δεδομένων (Target Database) της εφαρμογής στο περιβάλλον του Vuforia Target Manager, όπου παρουσιάζεται η λίστα των εικόνων-στόχων και η αξιολόγηση της καταλληλότητάς τους για ανίχνευση (Star Rating).....	17
Σχήμα 3.1 Η οθόνη του Κεντρικού Μενού με τις τέσσερις βασικές επιλογές .....	20
Σχήμα 3.2 Οθόνη "Το Μάθημα" με χρονολογίες, εικόνα και διαδραστικές λέξεις-κλειδιά .....	21
Σχήμα 3.3 Η ενότητα "Πηγές" με τη λίστα των εξωτερικών συνδέσμων (YouTube / Sites) .....	22
Σχήμα 3.4 Η οθόνη επιλογής επιπέδων (Level Selection).....	23
Σχήμα 3.5 Το περιβάλλον (UI) κατά τη διάρκεια του παιχνιδιού, όπου διακρίνονται οι διαθέσιμες ζωές και το χρονόμετρο .....	25
Σχήμα 3.6 Παράδειγμα ερώτησης με επιλογή σωστής απάντησης (Κείμενο) και επιλογή σωστής Εικόνας .....	26
Σχήμα 3.7 Διαδικασία AR: Σκανάρισμα εικόνας βιβλίου και εμφάνιση 3D μοντέλου .....	27
Σχήμα 3.8 Διαδικασία AR: Ο χρήστης περιστρέφει το 3D μοντέλο και κάνει κλικ στο ζητούμενο σημείο .....	28
Σχήμα 3.9 Η οθόνη τελικών αποτελεσμάτων (Feedback), όπου διακρίνεται η αναλυτική λίστα των σωστών/λανθασμένων απαντήσεων και η τελική καταμέτρηση των πόντων του μαθητή.....	29
Σχήμα 4.1 Απόσπασμα κώδικα από το script διαχείρισης παραθύρων (ManagerPanelController), που δείχνει τις μεθόδους ενεργοποίησης/απενεργοποίησης των UI Panels.....	31
Σχήμα 4.2 Η ιεραρχία (Hierarchy) της κύριας σκηνής στον Unity Editor, όπου διακρίνεται η οργάνωση των UI Panels κάτω από το κεντρικό Canvas .....	32
Σχήμα 4.3 Απόσπασμα του αρχείου LessonQuestions.json που αναπαριστά μια δοκιμασία Επαυξημένης Πραγματικότητας (questionType: "ARPrompt"), με τα πεδία modelName και interactionType.....	33
Σχήμα 4.4 Απόσπασμα του αρχείου LessonQuestions.json, όπου διακρίνεται η δομή μιας ερώτησης δύο επιλογών (TwoChoice) και μιας δοκιμασίας AR .....	33
Σχήμα 4.5 Απόσπασμα κώδικα JSON που αναπαριστά μια ερώτηση επιλογής τετραγώνου πάνω σε εικόνα-collage (ImageChoice).....	33
Σχήμα 4.6 Απόσπασμα του αρχείου lesson.json, όπου φαίνεται η χρήση της ετικέτας <link> για τη δημιουργία διαδραστικών λέξεων .....	34

Σχήμα 4.7 Απόσπασμα κώδικα από τον LevelSelectionController που δείχνει τη μέθοδο ελέγχου ξεκλειδώματος επιπέδων (IsLevelUnlocked).....	35
Σχήμα 4.8: Επεξεργασία του προκατασκευασμένου αντικειμένου (Prefab) "LevelButtonPrototype" μέσα από τον Unity Editor .....	36
Σχήμα 4.9 Η μέθοδος LoadContentParts από το LessonManager, όπου υλοποιείται η λογική τεμαχισμού του κειμένου με όριο χαρακτήρων .....	37
Σχήμα 4.10 Απόσπασμα κώδικα από το LessonManager.cs που αναλαμβάνει τον τεμαχισμό του κειμένου και τη σελιδοποίηση.....	37
Σχήμα 4.11 Απόσπασμα από το LessonLinkHandler που δείχνει τη μέθοδο OnPointerClick και τη διαχείριση του υπερσυνδέσμου .....	38
Σχήμα 4.12: Ο έλεγχος των τύπων αλληλεπίδρασης (interactionType) στη μέθοδο Update() του ARPromptController.cs .....	39
Σχήμα 4.13: Απόσπασμα κώδικα όπου φαίνεται η τυπική διαδικασία φόρτωσης και αποσειριοποίησης (deserialization) ενός αρχείου JSON μέσω του JsonUtility .....	39
Σχήμα 4.14: Απόσπασμα κώδικα από το QuizManager, με έμφαση στις μεθόδους ελέγχου απαντήσεων (OnAnswerSelected) και διαχείρισης ζώων/σκορ .....	41
Σχήμα 4.15: Χρήση της μεθόδου Invoke() στο QuizManager.cs για την εισαγωγή χρονοκαθυστερήσης μεταξύ των ερωτήσεων .....	42
Σχήμα 4.16: Απόσπασμα κώδικα από το QuizManager που δείχνει τον υπολογισμό του Time Bonus και τη διαχείριση του Streak System.....	42
Σχήμα 4.17: Η μέθοδος ShowDetailedFeedback, όπου διακρίνεται η χρήση του StringBuilder για τη δημιουργία της αναλυτικής αναφοράς .....	43
Σχήμα 4.18: Απόσπασμα κώδικα από το QuizManager, εστιάζοντας στη μέθοδο SetupImageQuestion, η οποία διαχειρίζεται τη φόρτωση και προβολή των ερωτήσεων εικόνας.....	44
Σχήμα 4.19: Δυναμική φόρτωση εικόνων μέσω της Resources.Load<Sprite>() στον κώδικα της εφαρμογής .....	45
Σχήμα 4.20: Το περιβάλλον του Vuforia Target Manager, όπου διακρίνονται τα "Feature Points" (κίτρινοι σταυροί) πάνω σε μια εικόνα του σχολικού βιβλίου .....	46
Σχήμα 4.21: Το παράθυρο Vuforia Configuration μέσα στον Unity Editor, όπου εισάγεται το App License Key .....	47
Σχήμα 4.22: Η ιεραρχία ενός Image Target (π.χ. ImageTarget_Colosseum) στον Unity Editor, όπου το 3D μοντέλο λειτουργεί ως εξαρτώμενο αντικείμενο (Child GameObject) .....	48
Σχήμα 4.23 Διάγραμμα ροής της δοκιμασίας Επαυξημένης Πραγματικότητας, με τα τρία στάδια αλληλεπίδρασης (Σάρωση, Περιστροφή, Επιλογή) .....	50
Σχήμα 4.24: Απόσπασμα κώδικα από το ARPromptController, που δείχνει τη μέθοδο HandleClickInteraction και τη χρήση Raycast για την αναγνώριση του αγγίγματος στο 3D μοντέλο. ....	51
Σχήμα 4.25: Απόσπασμα κώδικα που διαχειρίζεται την ομαλή περιστροφή των 3D μοντέλων μέσω της Mathf.SmoothDampAngle .....	52
Σχήμα 4.26: Προβολή των συστατικών (Inspector) του Prefab "Pantheon". Διακρίνονται η ιεραρχία του κάτω από το ImageTarget, το script εξομάλυνσης περιστροφής και το Box Collider για την ανίχνευση του Raycast.....	53
Σχήμα 4.27: Ο κώδικας του βοηθητικού εργαλείου ColliderVisualizer.cs, το οποίο σχεδιάζει τα αόρατα όρια σύγκρουσης στο περιβάλλον ανάπτυξης.....	54

## Συντομογραφίες

2D	Two-Dimensional
3D	Three-Dimensional
AR	Augmented Reality
Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΙΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
MR	Mixed Reality
OS	Operating System
SDK	Software Development Kit
UI	User Interface
UX	User Experience
VR	Virtual Reality
XR	Extended Reality

## Κεφάλαιο 1ο: Εισαγωγή

### 1.1 Περιγραφή και Αντικείμενο της Εργασίας

Η εργασία αυτή έχει ως αντικείμενο τον σχεδιασμό και την υλοποίηση μιας εκπαιδευτικής εφαρμογής Επαυξημένης Πραγματικότητας (Augmented Reality - AR) σε μορφή διαδραστικού παιχνιδιού. Το παιχνίδι απευθύνεται σε μαθητές της Ε΄ Δημοτικού και βασίζεται στην ενότητα «Οι Έλληνες και οι Ρωμαίοι» του σχολικού βιβλίου, η οποία καλύπτει τη ρωμαϊκή κατάκτηση της Ελλάδας, τον ελληνορωμαϊκό πολιτισμό και τη δομή της Ρωμαϊκής Αυτοκρατορίας.

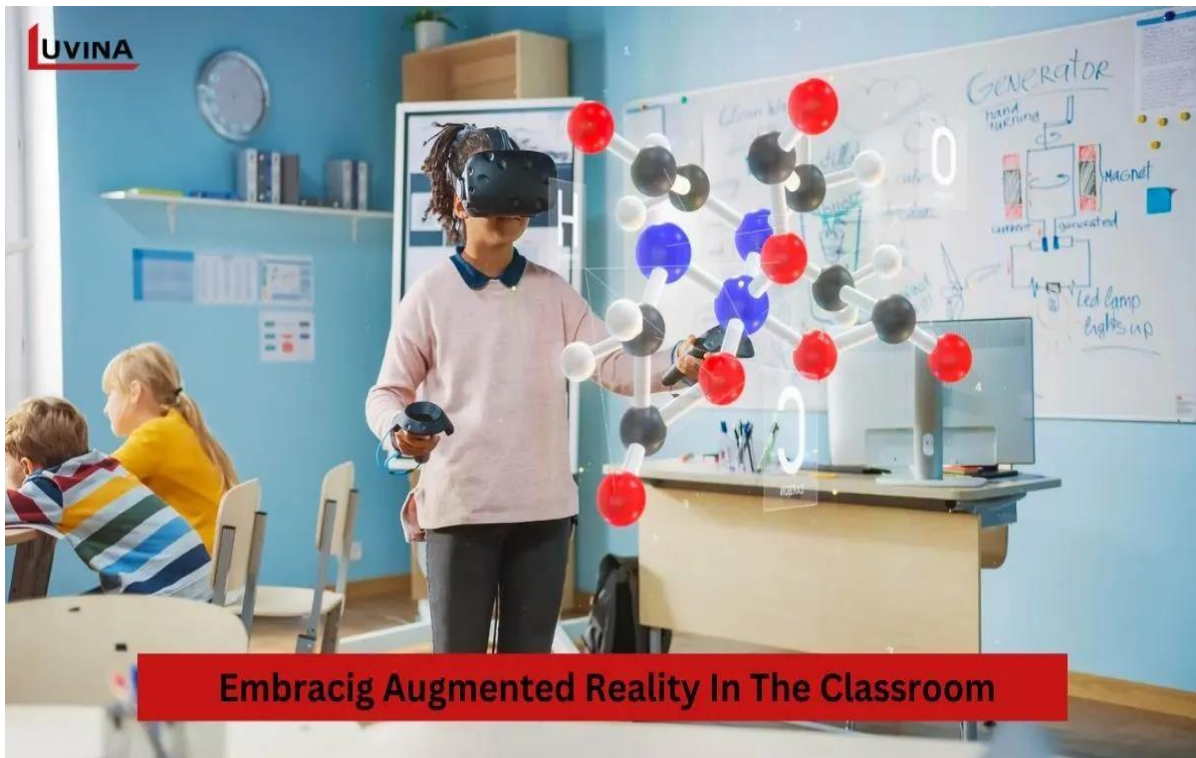
Η παραδοσιακή διδασκαλία της ιστορίας, συχνά, βασίζεται στην αποστήθιση γεγονότων και ημερομηνιών, κάτι που μπορεί να μειώσει το ενδιαφέρον των μαθητών. Η ενσωμάτωση σύγχρονων τεχνολογιών, όπως η Επαυξημένη Πραγματικότητα, προσφέρει μια εναλλακτική προσέγγιση. Μέσω της εφαρμογής, τα στατικά στοιχεία του σχολικού βιβλίου (όπως εικόνες και χάρτες) μπορούν να ζωντανέψουν, προσφέροντας στους μαθητές ψηφιακές πληροφορίες και διαδραστικά κουίζ (όπως ερωτήσεις δύο επιλογών και επιλογή σωστής εικόνας) σε πραγματικό χρόνο.

### 1.2 Σκοπός και Στόχοι

Ο κύριος στόχος αυτής της πτυχιακής είναι η δημιουργία ενός λειτουργικού, διαδραστικού παιχνιδιού που θα κάνει το μάθημα της ιστορίας πιο ενδιαφέρον, ευχάριστο και ελκυστικό για τους μαθητές (Σχήμα 1.1).

Οι επιμέρους στόχοι της εργασίας περιλαμβάνουν:

- Τον σχεδιασμό ενός φιλικού και εύχρηστου περιβάλλοντος χρήστη (UI), κατάλληλου για μαθητές δημοτικού.
- Την υλοποίηση διαφορετικών τύπων διαδραστικών ερωτήσεων για την αξιολόγηση των γνώσεων. Για παράδειγμα, ο μαθητής καλείται να απαντήσει σε ερωτήσεις δύο επιλογών (Σωστό/Λάθος ή δύο εναλλακτικές απαντήσεις κειμένου), καθώς και να επιλέξει τη σωστή απάντηση μέσα από μια σειρά εικόνων.
- Την ενσωμάτωση τεχνολογίας AR για την αναγνώριση εικόνων (Image Targets) και την εμφάνιση τρισδιάστατων (3D) μοντέλων και πληροφοριών πάνω σε αυτές.
- Τη δημιουργία ενός ευέλικτου συστήματος διαχείρισης δεδομένων, όπου το περιεχόμενο των μαθημάτων και των αξιολογήσεων φορτώνεται δυναμικά από εξωτερικά αρχεία JSON, διευκολύνοντας την αναβάθμιση του υλικού.



Σχήμα 1.1: Η εφαρμογή απευθύνεται σε μαθητές Δημοτικού, που εξοικειώνονται με την εκπαιδευτική ύλη μέσω διαδραστικών τεχνολογιών

### 1.3 Συνοπτική Αναφορά στις Τεχνολογίες

Για την υλοποίηση της εφαρμογής επιλέχθηκαν σύγχρονα, ώριμα και ευρέως τεκμηριωμένα εργαλεία ανάπτυξης λογισμικού:

**Unity:** Αποτελέσει το βασικό περιβάλλον ανάπτυξης (Game Engine) της εφαρμογής. Η Unity προσφέρει ισχυρά εργαλεία για τη δημιουργία 2D και 3D γραφικών, τη διαχείριση του User Interface (UI) και τον προγραμματισμό της λογικής του παιχνιδιού μέσω της γλώσσας C#.

**Vuforia Engine:** Χρησιμοποιήθηκε ως το βασικό SDK (Software Development Kit) για την υλοποίηση της Επαυξημένης Πραγματικότητας. Το Vuforia αναλαμβάνει την αναγνώριση των εικόνων (Image Tracking) μέσω της κάμερας της συσκευής, επιτρέποντας την προβολή του ψηφιακού περιεχομένου.

**Android:** Η εφαρμογή αναπτύχθηκε και εξήχθη (build) για λειτουργικό σύστημα Android, καθώς τα smartphones και τα tablets αποτελούν ευρέως διαδεδομένες συσκευές που μπορούν να αξιοποιηθούν σε εκπαιδευτικά περιβάλλοντα.

### 1.4 Δομή της Πτυχιακής Εργασίας

Η παρούσα πτυχιακή εργασία οργανώνεται σε πέντε κύρια κεφάλαια, ακολουθούμενα από τη Βιβλιογραφία και παράρτημα.

- Στο Κεφάλαιο 1 παρουσιάζεται το αντικείμενο της εργασίας, καθορίζεται ο σκοπός και οι στόχοι, περιγράφονται συνοπτικά οι τεχνολογίες που αξιοποιούνται και αναλύεται η μεθοδολογία ανάπτυξης.

- Στο Κεφάλαιο 2 αναλύεται το θεωρητικό υπόβαθρο και οι τεχνολογίες ανάπτυξης, με έμφαση στην Επαυξημένη Πραγματικότητα, την Παιχνιδοποίηση, καθώς και στα εργαλεία Unity, Vuforia, Android και C#.
- Στο Κεφάλαιο 3 φαίνεται η εφαρμογή όπως την βλέπει ο χρήστης (User Manual). Περιγράφεται πώς εμφανίζεται το κεντρικό μενού, πώς οργανώνεται το εκπαιδευτικό υλικό και ποιοι τύποι ερωτήσεων εμφανίζονται κατά τη διάρκεια του παιχνιδιού, μεταξύ των οποίων και οι δοκιμασίες AR.
- Στο Κεφάλαιο 4 περιγράφεται η αρχιτεκτονική του συστήματος και η προγραμματιστική υλοποίηση, μέσω των βασικών C# scripts που διαχειρίζονται τα UI panels, τα δεδομένα σε μορφή JSON, τη λογική των κουίζ και τις λειτουργίες Επαυξημένης Πραγματικότητας.
- Στο Κεφάλαιο 5 συνοψίζονται τα συμπεράσματα της εργασίας, συζητούνται οι προκλήσεις που προέκυψαν κατά την ανάπτυξη και προτείνονται μελλοντικές επεκτάσεις της εφαρμογής. Τέλος, παρατίθεται η Βιβλιογραφία.

## 1.5 Μεθοδολογία Ανάπτυξης

Για την υλοποίηση της παρούσας εκπαιδευτικής εφαρμογής υιοθετήθηκαν οι αρχές της προσέγγισης Design-Based Research (DBR) [2], σε συνδυασμό με τις αρχές της Ευέλικτης Ανάπτυξης (Agile). Η DBR ακολουθεί επαναλαμβανόμενους κύκλους όπου ένα εκπαιδευτικό εργαλείο σχεδιάζεται, δοκιμάζεται και βελτιώνεται σταδιακά. Στην παρούσα εργασία ακολουθήθηκε αυτή η λογική κατά τη φάση ανάπτυξης, δηλαδή μετά από κάθε νέα έκδοση της εφαρμογής γίνονταν εσωτερικές δοκιμές και βάσει αυτών αλλάζονταν στοιχεία πριν προχωρήσει η υλοποίηση. Η αξιολόγηση όμως σε πραγματική σχολική τάξη με μαθητές δεν εντάχθηκε στο πλαίσιο της εργασίας και αναφέρεται στις μελλοντικές επεκτάσεις.

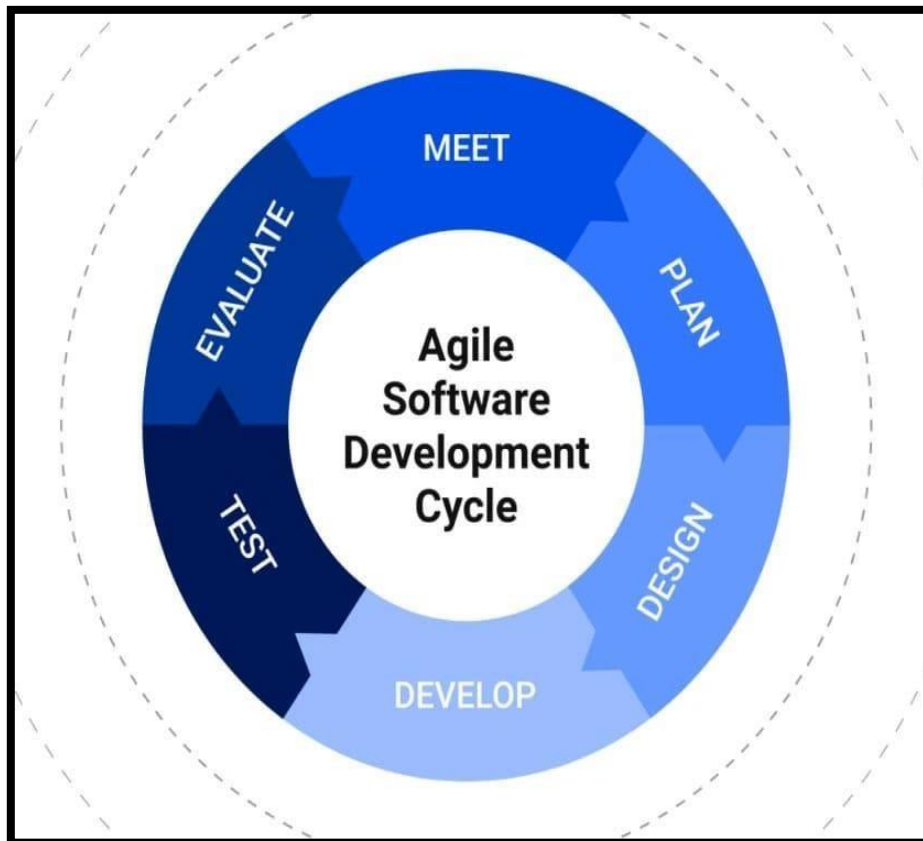
Όπως φαίνεται στον γενικό κύκλο Agile (Σχήμα 1.2), η ανάπτυξη λογισμικού ακολουθεί επαναλαμβανόμενες φάσεις σχεδιασμού, υλοποίησης και αξιολόγησης. Στην παρούσα εργασία, η διαδικασία αυτή προσαρμόστηκε σε τέσσερα (4) διακριτά και επαναληπτικά στάδια:

1. **Ανάλυση Απαιτήσεων:** Στο πρώτο βήμα καθορίστηκαν τόσο το περιεχόμενο όσο και το gameplay του παιχνιδιού. Αντλήθηκε το εκπαιδευτικό υλικό από την ύλη Ιστορίας της Ε' Δημοτικού (από την ενότητα «Οι Έλληνες και οι Ρωμαίοι») και αποφασίστηκε με ποιους τρόπους θα αλληλεπιδρά ο μαθητής (ερωτήσεις κουίζ, κλικ σε εικόνες και δραστηριότητες Επαυξημένης Πραγματικότητας).
2. **Δημιουργία Πρωτοτύπου (Prototyping):** Σχεδιάστηκε η ροή πλοήγησης (user flow) της εφαρμογής δίνοντας έμφαση στην ελεύθερη επιλογή. Έτσι, ο μαθητής ξεκινά από το Κεντρικό Μενού (Main Menu) και μπορεί να μεταβεί αυτόνομα σε όποια ενότητα επιθυμεί: στη Θεωρία, στις ιστορικές Πηγές ή απευθείας στην επιλογή επιπέδου για τα Quiz και το AR. Έπειτα, στήθηκαν τα πρώτα οπτικά περιβάλλοντα (UI Panels) μέσα στη μηχανή γραφικών Unity.
3. **Ανάπτυξη Λογισμικού:** Προγραμματίστηκε η εφαρμογή σε γλώσσα C#, ενσωματώνοντας όλους τους μηχανισμούς του παιχνιδιού (βαθμολογία, ζώες, χρονόμετρο). Παράλληλα, ρυθμίστηκε το εργαλείο Vuforia προσθέτοντας τις εικόνες-στόχους, δίνοντας έτσι τη

δυνατότητα στην κάμερα να αναγνωρίζει σελίδες του βιβλίου και να εμφανίζει πάνω τους διαδραστικά 3D μοντέλα.

4. **Έλεγχος και Επανασχεδιασμός (Debugging):** Εγκαταστάθηκε η πρώτη έκδοση της εφαρμογής (.apk) σε πραγματικές συσκευές Android για δοκιμές. Μέσα από εκτεταμένο εσωτερικό testing της εφαρμογής εντοπίστηκαν τεχνικά και σχεδιαστικά ζητήματα που έπρεπε να βελτιωθούν. Για παράδειγμα, αυξήθηκε ο όγκος στα hitboxes (Box Colliders) των 3D μοντέλων ώστε η επιλογή από τον χρήστη (tap) να αναγνωρίζεται ευκολότερα σε μικρές οθόνες. Επίσης, βελτιώθηκε η ευχρηστία του UI, με διορθώσεις όπως η αλλαγή στο μέγεθος των γραμματοσειρών για καλύτερη αναγνωσιμότητα και η οπτική ανατροφοδότηση (Detailed Feedback) με τη χρήση κατάλληλων χρωμάτων στις απαντήσεις.

Αυτή η διαδικασία δοκιμών και διορθώσεων ήταν καθοριστική, καθώς διασφάλισε ότι η εφαρμογή δεν είναι απλώς λειτουργική σε επίπεδο κώδικα, αλλά σχεδιάστηκε με γνώμονα τη χρηστικότητα για παιδιά 10-11 ετών, με γραφικά και διαδραστικά στοιχεία προσαρμοσμένα στις ανάγκες της ηλικιακής αυτής ομάδας.



Σχήμα 1.2: Ο γενικός κύκλος ανάπτυξης λογισμικού της μεθοδολογίας Agile

## 1.6 Επίλογος Κεφαλαίου

Στο κεφάλαιο αυτό προσδιορίστηκε το αντικείμενο της εργασίας και ο λόγος για τον οποίο επιλέχθηκε η Επαυξημένη Πραγματικότητα ως εργαλείο για τη διδασκαλία της Ιστορίας στην Ε΄ Δημοτικού. Παρουσιάστηκαν ο σκοπός και οι επιμέρους στόχοι, μαζί με τα βασικά εργαλεία (Unity, Vuforia, Android, C#) που αξιοποιήθηκαν κατά την ανάπτυξη.

Περιγράφηκε επίσης η μεθοδολογία που ακολουθήθηκε, η οποία στηρίχθηκε στον επαναληπτικό χαρακτήρα της προσέγγισης Design-Based Research και χωρίστηκε σε τέσσερα στάδια, από την ανάλυση των απαιτήσεων μέχρι τον έλεγχο και τον επανασχεδιασμό.



## Κεφάλαιο 2ο: Θεωρητικό Υπόβαθρο και Τεχνολογίες Ανάπτυξης

### 2.1 Εισαγωγή Κεφαλαίου

Το κεφάλαιο αυτό καλύπτει το θεωρητικό και τεχνολογικό υπόβαθρο της εργασίας. Αρχικά, αποσαφηνίζεται η έννοια της Επαυξημένης Πραγματικότητας (AR) και αναδεικνύονται τα παιδαγωγικά της οφέλη, σε συνδυασμό με τις αρχές της Παιχνιδοποίησης (Gamification) στην εκπαίδευση. Στη συνέχεια, παρουσιάζονται τα σύγχρονα εργαλεία ανάπτυξης λογισμικού που επιλέχθηκαν για την υλοποίηση της εφαρμογής. Αναλύεται η αρχιτεκτονική της μηχανής γραφικών Unity 3D, η λειτουργία του συστήματος μηχανικής όρασης Vuforia Engine, καθώς και ο ρόλος της γλώσσας προγραμματισμού C# και του λειτουργικού συστήματος Android.

### 2.2 Ο Διαχωρισμός των Τεχνολογιών: AR, VR, MR και XR

Για την πλήρη κατανόηση της τεχνολογίας που χρησιμοποιείται στην παρούσα εφαρμογή, είναι απαραίτητος ο σαφής διαχωρισμός της Επαυξημένης Πραγματικότητας (AR) από τις συγγενικές της τεχνολογίες. Συχνά στη βιβλιογραφία οι όροι αυτοί συγχέονται, ωστόσο εντάσσονται όλοι στο ευρύτερο θεωρητικό μοντέλο του **Reality-Virtuality Continuum**. Οι βασικές κατηγορίες αναλύονται παρακάτω:

#### 1. Εικονική Πραγματικότητα (Virtual Reality - VR)

Η Εικονική Πραγματικότητα προσφέρει την πλήρη βύθιση (immersion) του χρήστη σε ένα εξ ολοκλήρου ψηφιακό περιβάλλον. Ο χρήστης αποκόπτεται πλήρως από τον φυσικό, πραγματικό κόσμο, φορώντας ειδικά κράνη (Head-Mounted Displays - HMDs) όπως το Meta Quest, το PlayStation VR2 ή το Valve Index. Κάθε τι που βλέπει ή ακούει παράγεται από τον υπολογιστή.

#### 2. Επαυξημένη Πραγματικότητα (Augmented Reality - AR)

Αποτελεί την τεχνολογία στην οποία βασίζεται η παρούσα εργασία. Δεν αντικαθιστά τον πραγματικό κόσμο, αλλά προσθέτει ψηφιακά στοιχεία πάνω στο φυσικό περιβάλλον του χρήστη σε πραγματικό χρόνο (συνήθως μέσω της κάμερας ενός smartphone ή tablet).

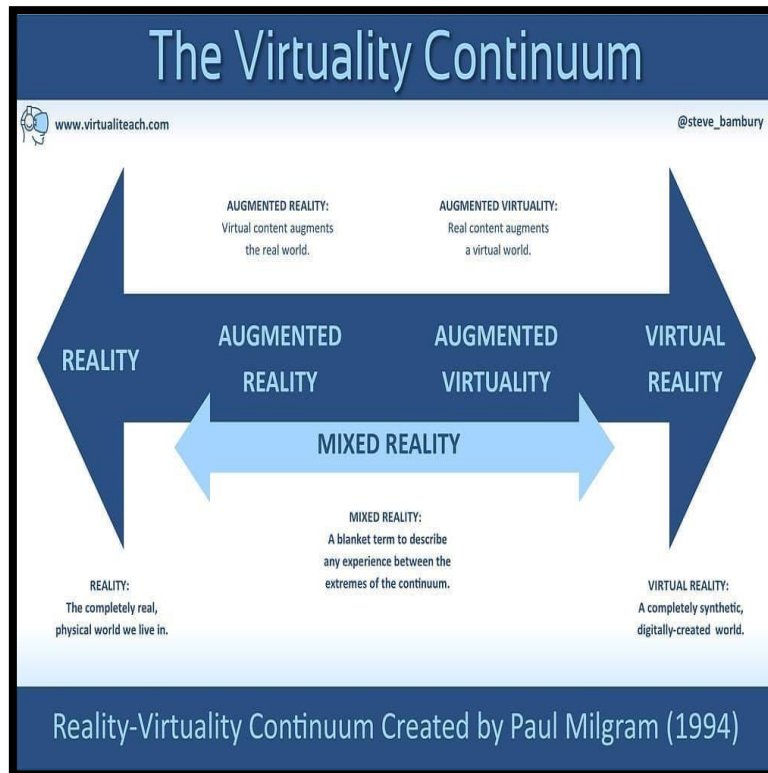
#### 3. Μεικτή Πραγματικότητα (Mixed Reality - MR)

Η Μεικτή Πραγματικότητα επεκτείνει τις δυνατότητες του AR. Σε αυτήν, τα ψηφιακά αντικείμενα όχι μόνο προβάλλονται στον πραγματικό κόσμο, αλλά μπορούν και να αλληλεπιδρούν με τα φυσικά αντικείμενα. Για παράδειγμα, ένα ψηφιακό μπαλάκι σε ένα περιβάλλον MR θα μπορούσε να χτυπήσει πάνω σε ένα πραγματικό, φυσικό τραπέζι και να αναπηδήσει, ή να κρυφτεί πίσω από έναν πραγματικό καναπέ. Η τεχνολογία αυτή απαιτεί εξαιρετικά προηγμένους αισθητήρες χαρτογράφησης χώρου και εξειδικευμένο εξοπλισμό, όπως το Microsoft HoloLens.

#### 4. Εκτεταμένη Πραγματικότητα (Extended Reality - XR)

Τέλος, η Εκτεταμένη Πραγματικότητα (XR) δεν είναι μια ξεχωριστή τεχνολογία, αλλά ο "όρος ομπρέλα" (umbrella term) που χρησιμοποιείται στη σύγχρονη βιομηχανία της Πληροφορικής για να περιγράψει συνολικά όλες τις παραπάνω τεχνολογίες (VR, AR, MR). Οποιαδήποτε εφαρμογή μεταβάλλει ή συνδυάζει την πραγματικότητα με ψηφιακά γραφικά, ανήκει στο φάσμα του XR.

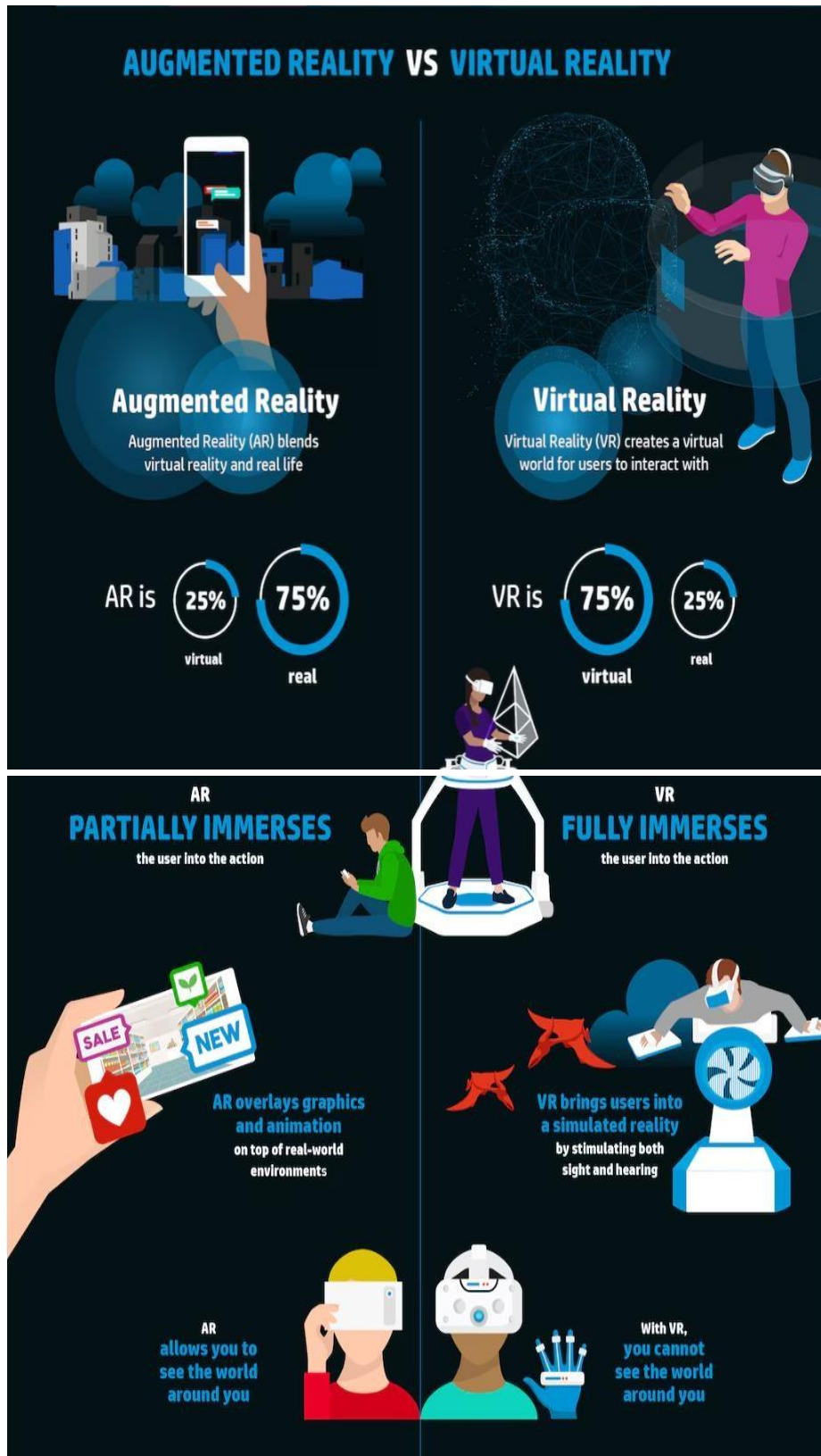
Όπως απεικονίζεται στο φάσμα Εικονικότητας-Πραγματικότητας (Σχήμα 2.1), οι τεχνολογίες VR, AR και MR αποτελούν διαβαθμίσεις του ίδιου συνεχούς. Συνοψίζοντας, η επιλογή της Επαυξημένης Πραγματικότητας (AR) για την παρούσα εκπαιδευτική εφαρμογή κρίθηκε ως η καταλληλότερη. Δεν χρειάζεται ακριβός εξοπλισμός όπως κράνη VR, παρά μόνο ένα κινητό ή tablet με Android. Επιπλέον, ο μαθητής συνεχίζει να δουλεύει με το σχολικό του βιβλίο, καθώς η AR χρησιμοποιεί τις σελίδες του ως σημεία εκκίνησης για να εμφανίσει το ψηφιακό περιεχόμενο.



Σχήμα 2.1: Το συνεχές φάσμα Εικονικότητας-Πραγματικότητας (Reality-Virtuality Continuum) κατά τον Paul Milgram [1]

### 2.3 Επαυξημένη Πραγματικότητα (Augmented Reality - AR)

Εστιάζοντας στην τεχνολογία που επιλέχθηκε για την παρούσα εφαρμογή, η Επαυξημένη Πραγματικότητα βασίζεται στον μηχανισμό της **υπέρθωσης (overlay)**. Συγκεκριμένα, επιτρέπει την προβολή ψηφιακών πληροφοριών —όπως τρισδιάστατα (3D) μοντέλα, αναδυόμενα κείμενα, ήχο και βίντεο— αρμονικά ενσωματωμένων με το φυσικό περιβάλλον του χρήστη. Σε αντίθεση με τη VR, όπου ο χρήστης βυθίζεται πλήρως σε έναν ψηφιακό κόσμο και αποκόπτεται από το περιβάλλον του, η AR διατηρεί τον πραγματικό κόσμο ως βάση και προσθέτει πάνω του τα ψηφιακά στοιχεία. Αυτή η θεμελιώδης διαφορά καθορίζει σε μεγάλο βαθμό και τον τρόπο με τον οποίο αξιοποιείται η κάθε τεχνολογία στην εκπαίδευση. Η σύγκριση των βασικών χαρακτηριστικών των δύο τεχνολογιών συνοψίζεται σχηματικά στο Σχήμα 2.2.



Σχήμα 2.2: Συγκριτική σχηματική απεικόνιση των βασικών χαρακτηριστικών της Επαυξημένης (AR) και της Εικονικής Πραγματικότητας (VR).

### 2.3.1 Ιστορική Εξέλιξη της Τεχνολογίας

Η AR έχει μακρά ερευνητική ιστορία που ξεκινά από τη δεκαετία του 1960, με τα πρώτα συστήματα προβολής (Head-Mounted Displays) από ερευνητές όπως ο Ivan Sutherland [3]. Ωστόσο, ο όρος "Augmented Reality" επινοήθηκε επίσημα το 1990 από τον Thomas Caudell [4], ερευνητή της Boeing, ο οποίος σχεδίασε ένα σύστημα που πρόβαλλε ψηφιακά διαγράμματα καλωδιώσεων στο οπτικό πεδίο των εργατών συναρμολόγησης αεροσκαφών.

Στα τέλη της δεκαετίας του 1990, η έρευνα επικεντρώθηκε στην ανάπτυξη βιβλιοθηκών λογισμικού, με χαρακτηριστικό παράδειγμα το ARToolKit, που αναπτύχθηκε το 1999 από τον Hirokazu Kato [5]. Το ARToolKit επέτρεπε εντοπισμό θέσης κάμερας σε πραγματικό χρόνο μέσω ασπρόμαυρων δεικτών (markers) τυπωμένων σε χαρτί, και υπήρξε από τα πρώτα λογισμικά ανοιχτού κώδικα στον τομέα. Η πραγματική όμως μαζική εξάπλωση της τεχνολογίας συνέβη μετά το 2010, με την ανάπτυξη των smartphones και tablets — συσκευών εξοπλισμένων με κάμερες υψηλής ανάλυσης, αισθητήρες γυροσκοπίου και ισχυρούς επεξεργαστές γραφικών.

### 2.3.2 Η Επαυξημένη Πραγματικότητα στην Εκπαίδευση

Η ενσωμάτωση της AR στην εκπαιδευτική διαδικασία έχει συγκεντρώσει τεράστιο ερευνητικό ενδιαφέρον την τελευταία δεκαετία. Σε αντίθεση με την παραδοσιακή διδασκαλία, η οποία συχνά βασίζεται σε δισδιάστατα μέσα (όπως πίνακες και βιβλία), το AR προσφέρει τη δυνατότητα οπτικοποίησης αφηρημένων ή πολύπλοκων εννοιών σε τρεις διαστάσεις. Για παράδειγμα, στο μάθημα της ιστορίας, οι μαθητές δεν χρειάζεται να φανταστούν πώς έμοιαζε ένα αρχαίο μνημείο, αλλά μπορούν να το δουν να προβάλλεται τρισδιάστατα (3D) πάνω στο θρανίο τους και να το εξερευνήσουν από διαφορετικές οπτικές γωνίες.

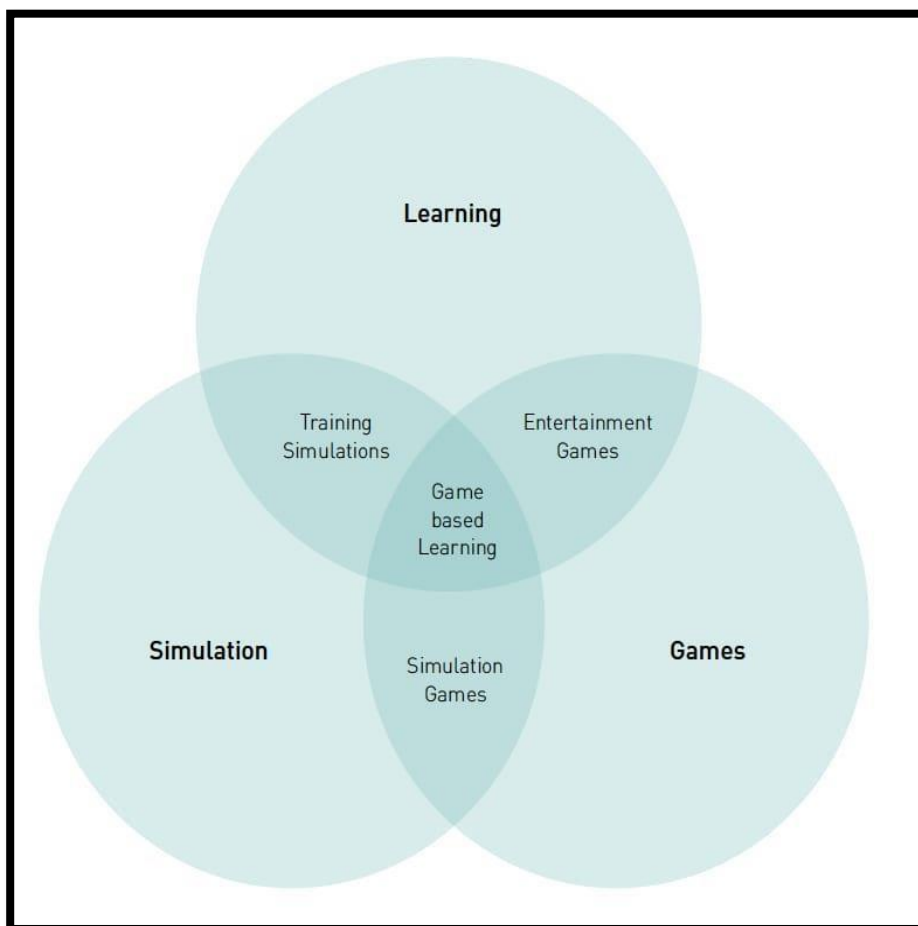
Σύμφωνα με πλήθος ακαδημαϊκών μελετών [6], [7], τα κύρια πλεονεκτήματα από τη χρήση AR συστημάτων στη σχολική τάξη περιλαμβάνουν:

- **Αυξημένη Δέσμευση (Engagement):** Οι μαθητές δείχνουν μεγαλύτερο ενδιαφέρον και προσοχή, καθώς η διαδικασία μάθησης μετατρέπεται σε εμπειρία ανακάλυψης.
- **Βελτίωση Χωρικής Αντίληψης:** Η διαδραστικότητα με τρισδιάστατα ψηφιακά αντικείμενα βοηθά στην καλύτερη κατανόηση του χώρου, της γεωμετρίας και της αρχιτεκτονικής.
- **Ενεργητική Μάθηση (Active Learning):** Αντί να παρακολουθεί παθητικά, ο μαθητής αναζητά και ανακαλύπτει την πληροφορία μόνος του, μέσα από τον εντοπισμό image targets και τον χειρισμό 3D μοντέλων.
- **Κίνητρο για Μάθηση (Motivation):** Η χρήση σύγχρονων τεχνολογιών οι οποίες είναι οικείες στους μαθητές της σημερινής γενιάς (όπως τα tablets) ενισχύει το κίνητρο και την ενεργό συμμετοχή των μαθητών στο μάθημα [8].

## 2.4 Παιγνιδοποίηση (Gamification) στην Εκπαίδευση

Ένας βασικός πυλώνας στον σχεδιασμό της παρούσας εφαρμογής είναι η ενσωμάτωση στοιχείων παιγνιδοποίησης (gamification). Ως παιγνιδοποίηση ορίζεται η χρήση μηχανισμών, αισθητικής και τρόπου σκέψης που προέρχονται από τα παιχνίδια (game design elements) σε περιβάλλοντα που δεν αποτελούν αμιγώς παιχνίδια, όπως είναι η εκπαίδευση [9]. Η συνένωση αυτή των δύο κόσμων της παραδοσιακής μάθησης και του σχεδιασμού παιχνιδιών αποτυπώνεται στο Σχήμα 2.3.

Όπως φαίνεται στο Σχήμα 2.3, η Παιγνιδοκεντρική Μάθηση (Game-Based Learning) βρίσκεται στο κέντρο, στην τομή τριών πεδίων: της Μάθησης (Learning), της Προσομοίωσης (Simulation) και των Παιχνιδιών (Games). Οι επιμέρους τομές δημιουργούν ενδιάμεσες κατηγορίες, όπως τις Εκπαιδευτικές Προσομοιώσεις (Training Simulations), τα Ψυχαγωγικά Παιχνίδια (Entertainment Games) και τα Παιχνίδια Προσομοίωσης (Simulation Games). Η παρούσα εφαρμογή τοποθετείται ακριβώς στο κεντρικό σημείο αυτού του διαγράμματος, καθώς συνδυάζει εκπαιδευτικό περιεχόμενο (μάθημα Ιστορίας), στοιχεία παιχνιδιού (πόντοι, ζωές, χρονόμετρο) και προσομοίωση μέσω της τεχνολογίας AR που αναπαριστά ιστορικά γεγονότα σε τρισδιάστατο περιβάλλον.



Σχήμα 2.3: Η τομή της παραδοσιακής μάθησης με τα στοιχεία σχεδιασμού παιχνιδιών που δημιουργεί την Παιγνιδοποίηση (Gamification)

Στην παραδοσιακή διδασκαλία, η αξιολόγηση των μαθητών συχνά προκαλεί άγχος, καθώς τα λάθη έχουν άμεσο αντίκτυπο στη βαθμολογία. Αντίθετα, στον σχεδιασμό παιχνιδιών, η λανθασμένη επιλογή δεν τιμωρεί αλλά καθοδηγεί: ο παίκτης προχωρά μέσα από δοκιμή και σφάλμα (trial and error) σε ένα ασφαλές περιβάλλον, χωρίς τον φόβο μόνιμης αποτυχίας. Η προσέγγιση αυτή είναι γνωστή ως Παιχνιδοκεντρική Μάθηση (Game-Based Learning) και έχει συσχετιστεί με βελτίωση της συγκράτησης πληροφοριών [10].

Για την επιτυχή εφαρμογή της **Παιχνιδοποίησης (Gamification)** σε εκπαιδευτικά εργαλεία, αξιοποιούνται συγκεκριμένοι μηχανισμοί, που στηρίζονται στη Θεωρία του Αυτοκαθορισμού (Self-Determination Theory) των Deci και Ryan [11]. Σύμφωνα με τη θεωρία αυτή, οι μαθητές εμπλέκονται ενεργά όταν ικανοποιούνται οι βασικές τους ανάγκες για αυτονομία, ικανότητα και κοινωνική σύνδεση. Στα πλαίσια της παρούσας εφαρμογής, αυτό επιτυγχάνεται μέσω των παρακάτω μηχανισμών:

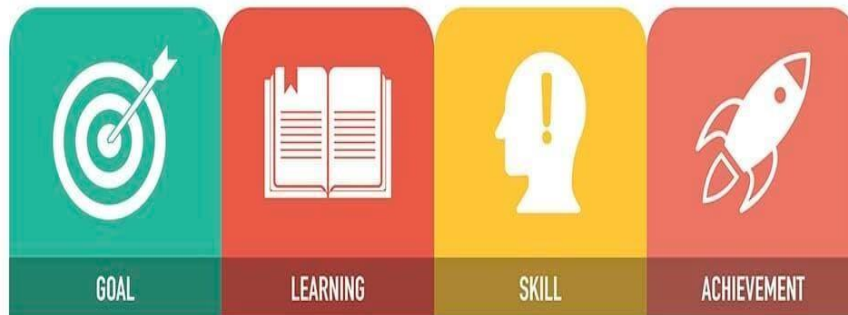
- **Σύστημα Πόντων (Scoring System):** Η επιβράβευση των σωστών απαντήσεων με πόντους προσφέρει άμεση ανατροφοδότηση (immediate feedback) στον μαθητή, ενισχύοντας το αίσθημα προόδου και επιτυχίας.
- **Χρονικό Όριο Απάντησης (Timer):** Η χρήση χρονομέτρου προσθέτει έναν παράγοντα έντασης και εγρήγορσης, κάνοντας τη διαδικασία πιο δυναμική και λιγότερο μονότονη.
- **Σύστημα Ζωών (Lives/Hearts System):** Ο μαθητής δεν αποκλείεται άμεσα με μια λάθος απάντηση, αλλά διαθέτει ένα περιθώριο σφαλμάτων (π.χ. 3 ζωές), το οποίο μειώνει τον φόβο της αποτυχίας και ενθαρρύνει τον πειραματισμό.
- **Σερί Σωστών Απαντήσεων (Streak Multipliers):** Η επιβράβευση διαδοχικών σωστών απαντήσεων διατηρεί την προσοχή του μαθητή σε υψηλά επίπεδα, δημιουργώντας μια αίσθηση "ροής" (flow state).

Με αυτόν τον τρόπο, το μάθημα της ιστορίας μπορεί να μετατραπεί από παθητική αποστήθιση σε διαδραστική εμπειρία για μαθητές Δημοτικού.

Το Σχήμα 2.4 παρουσιάζει τις βασικές αρχές της παιχνιδοποίησης, όπως ο Στόχος (Goal), η Πρόκληση (Challenge), η Ανταμοιβή (Reward), η Δεξιότητα (Skill), η Εμπλοκή Χρήστη (User Engagement) και το Επίτευγμα (Achievement). Από αυτές τις γενικές αρχές, η παρούσα εφαρμογή αξιοποιεί συγκεκριμένα τις εξής: η αρχή της Ανταμοιβής υλοποιείται μέσω του Συστήματος Πόντων, η Πρόκληση μέσω του Χρονικού Ορίου Απάντησης, η Εμπλοκή Χρήστη μέσω του Συστήματος Ζωών που δίνει στον μαθητή περιθώριο λάθους χωρίς να τον αποκλείει, και το Επίτευγμα μέσω του μηχανισμού Σερί Σωστών Απαντήσεων που επιβραβεύει τη συνέπεια και τη συγκέντρωση.

Πιο αναλυτικά, το **Σύστημα Πόντων** λειτουργεί ως άμεση ανατροφοδότηση (immediate feedback): κάθε φορά που ο μαθητής απαντά σωστά σε μια ερώτηση, κερδίζει πόντους που εμφανίζονται στην οθόνη, ενισχύοντας το αίσθημα προόδου. Το **Χρονικό Όριο Απάντησης** προσθέτει έναν παράγοντα εγρήγορσης, ο μαθητής δεν μπορεί να αφήσει μια ερώτηση ανοιχτή επ' αόριστον, κάτι που κάνει τη διαδικασία πιο δυναμική και αποτρέπει τη μονοτονία. Το **Σύστημα Ζωών** διαχειρίζεται τη σχέση του μαθητή με το λάθος, δηλαδή αντί μια λανθασμένη απάντηση να τερματίζει αμέσως το παιχνίδι, ο μαθητής διαθέτει τρεις ζωές, γεγονός που μειώνει το άγχος και ενθαρρύνει τη δοκιμή και το σφάλμα (trial and error) ως μέθοδο μάθησης. Τέλος, ο μηχανισμός **Σερί Σωστών Απαντήσεων** επιβραβεύει τη

συνέχεια, όταν ο μαθητής δίνει διαδοχικές σωστές απαντήσεις, κερδίζει πόντους με πολλαπλασιαστεί, δημιουργώντας μια αίσθηση «ροής» (flow state) που τον κρατά συγκεντρωμένο στο μάθημα.



## GAMIFICATION CONCEPT



Σχήμα 2.4 Τα βασικά δομικά στοιχεία της Παιγνιοποίησης (game design elements) που ενσωματώθηκαν στην εκπαιδευτική εφαρμογή

### 2.5 Μηχανή Γραφικών Unity 3D

Για τον σχεδιασμό και την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε η Unity 3D [12], μία από τις πιο δημοφιλείς μηχανές ανάπτυξης (game engines) παγκοσμίως. Η Unity παρέχει έναν ολοκληρωμένο Editor για τη δημιουργία 2D και 3D εφαρμογών, με δυνατότητα εξαγωγής (build) του ίδιου κώδικα σε πολλαπλές πλατφόρμες (Android, iOS, Windows). Η επιλογή της έναντι άλλων εργαλείων (π.χ. Unreal Engine) κρίθηκε καταλληλότερη λόγω της εύκολης ενσωμάτωσης με το Vuforia Engine SDK, της εκτεταμένης τεκμηρίωσής της και της ισχυρής υποστήριξης της κοινότητας προγραμματιστών.

Ένα από τα βασικά πλεονεκτήματα της Unity είναι η αρχιτεκτονική της που στηρίζεται σε αντικείμενα (GameObjects) και στοιχεία (Components). Κάθε οπτικό ή λειτουργικό στοιχείο της εφαρμογής, από ένα κουμπί του μενού μέχρι ένα τρισδιάστατο μοντέλο μνημείου, αποτελεί ένα GameObject στο οποίο προσαρτώνται διάφορα Components που καθορίζουν τη συμπεριφορά του. Η λογική του παιχνιδιού υλοποιείται μέσω scripts σε γλώσσα C#, τα οποία επίσης προσαρτώνται ως Components. Η προσέγγιση αυτή διευκολύνει τον modular σχεδιασμό που ακολουθήθηκε στην παρούσα εφαρμογή, καθώς κάθε λειτουργία μπορεί να αναπτυχθεί και να ελεγχθεί ανεξάρτητα. Το λογότυπο της μηχανής γραφικών Unity παρουσιάζεται στο Σχήμα 2.5.



Σχήμα 2.5 Το λογότυπο της μηχανής γραφικών Unity

## 2.6 Αναγνώριση Εικόνας (Image Tracking)

Η κύρια λειτουργία του Vuforia Engine [13] (Σχήμα 2.6) που αξιοποιήθηκε στην παρούσα εργασία είναι η Αναγνώριση Εικόνας (Image Tracking). Το σύστημα δεν αναγνωρίζει εικόνες ως απλά σύνολα pixels, αλλά χρησιμοποιεί μια πιο αποδοτική τεχνική βασισμένη στα "Σημεία Ενδιαφέροντος" (Feature Points).



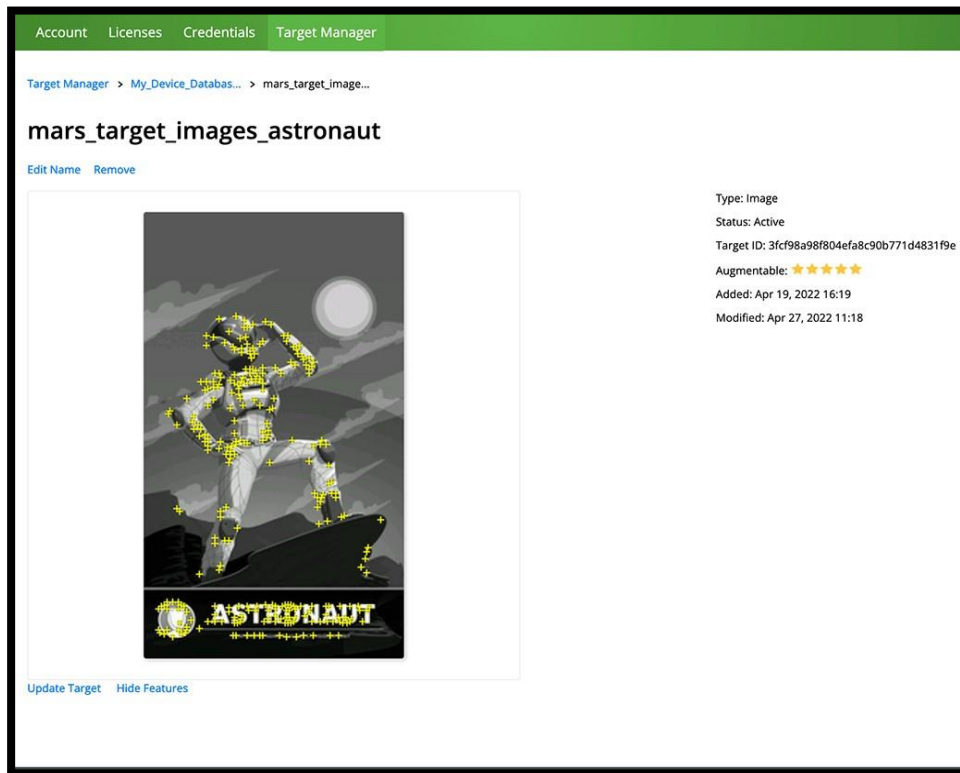
Σχήμα 2.6: Το λογότυπο του λογισμικού Επαυξημένης Πραγματικότητας Vuforia Engine

Όταν μια εικόνα (π.χ. μια σελίδα από το σχολικό βιβλίο Ιστορίας) ανεβαίνει στο Vuforia Target Manager, ο αλγόριθμος τη μετατρέπει σε ασπρόμαυρη (grayscale) και εντοπίζει περιοχές υψηλής αντίθεσης (high contrast), όπως έντονες γωνίες και ακμές. Με βάση αυτά τα χαρακτηριστικά, το σύστημα δημιουργεί ένα προφίλ για κάθε εικόνα-στόχο, το οποίο χρησιμοποιείται για την ταχύτερη και πιο αξιόπιστη αναγνώρισή της από την κάμερα (Σχήμα 2.7).

Κατά την εκτέλεση της εφαρμογής, όταν η κάμερα εντοπίσει μια διάταξη σημείων που ταυτίζεται με αυτό το αποτύπωμα, η Vuforia αναγνωρίζει και σταθεροποιεί τον στόχο (Image Target), επιτρέποντας την προβολή του αντίστοιχου 3D μοντέλου.

Οι εικόνες-στόχοι αξιολογούνται επίσης και ως προς την ποιότητά τους μέσα από το Star Rating του Vuforia Target Manager, ώστε να επιλέγονται όσες προσφέρουν πιο σταθερή αναγνώριση κατά την εκτέλεση της εφαρμογής. Η διαδικασία αυτή είναι ιδιαίτερα σημαντική σε εκπαιδευτικό περιβάλλον, επειδή οι σελίδες ενός σχολικού βιβλίου δεν έχουν πάντα την ίδια οπτική πολυπλοκότητα· ορισμένες εικόνες περιέχουν περισσότερα χαρακτηριστικά σημεία, ενώ άλλες είναι πιο "φτωχές" και επομένως λιγότερο αξιόπιστες ως Image Targets. Για τον λόγο αυτό, κατά τον σχεδιασμό της εφαρμογής επιλέχθηκαν εικόνες με υψηλή αντίθεση και σαφή γεωμετρικά στοιχεία, ώστε η αναγνώριση να παραμένει γρήγορη και σταθερή ακόμη και σε διαφορετικές συνθήκες φωτισμού ή γωνίας λήψης.

Με αυτόν τον τρόπο, μειώνονται τα σφάλματα αναγνώρισης και βελτιώνεται η συνολική εμπειρία χρήσης, κάτι που είναι κρίσιμο σε μια διαδραστική εφαρμογή AR για μαθητές Δημοτικού.



Σχήμα 2.7 Οπτικοποίηση των Feature Points (κίτρινοι σταυροί) στο περιβάλλον του Vuforia Target Manager

## 2.7 Ο Μηχανισμός Extended Tracking και Σταθερότητας

Ένα από τα σημαντικότερα χαρακτηριστικά που καθιστούν τη Vuforia κατάλληλη για εκπαιδευτικές εφαρμογές είναι το Extended Tracking (Εκτεταμένη Ιχνηλάτηση). Σε συστήματα AR χωρίς υποστήριξη Extended Tracking, αν το Image Target (το σχολικό βιβλίο) έβγαινε έστω και λίγο εκτός του οπτικού πεδίου της κάμερας, το 3D μοντέλο εξαφανιζόταν απότομα.

Η Vuforia Engine λύνει αυτό το πρόβλημα χαρτογραφώντας (mapping) τον περιβάλλοντα χώρο τη στιγμή της επιτυχούς αναγνώρισης. Χρησιμοποιώντας τεχνικές SLAM (Simultaneous Localization and Mapping), το σύστημα "θυμάται" πού βρισκόταν η εικόνα σε σχέση με τον υπόλοιπο χώρο. Έτσι, ο μαθητής μπορεί να απομακρύνει το κινητό του ή να κοιτάξει το τρισδιάστατο μοντέλο από άλλη γωνία, και το μοντέλο θα παραμείνει "καρφωμένο" στη θέση του (spatially anchored), προσφέροντας μια πιο ομαλή εμπειρία χρήστη και μειώνοντας τις απότομες απενεργοποιήσεις του 3D μοντέλου όταν αυτό βγαίνει εκτός κάδρου.

Στην παρούσα εφαρμογή, η λειτουργία Extended Tracking ενεργοποιήθηκε για όλα τα Image Targets, ώστε η εμπειρία του μαθητή να παραμένει αδιάκοπη ακόμα και όταν το βιβλίο βγαίνει εν μέρει εκτός κάδρου. Αυτό αποδείχθηκε ιδιαίτερα χρήσιμο κατά τη διάρκεια των δοκιμών, καθώς οι μαθητές τείνουν να κινούν το κινητό τους ελεύθερα γύρω από το βιβλίο για να εξερευνήσουν το 3D μοντέλο από διαφορετικές οπτικές γωνίες. Χωρίς το Extended Tracking, κάθε τέτοια κίνηση θα προκαλούσε

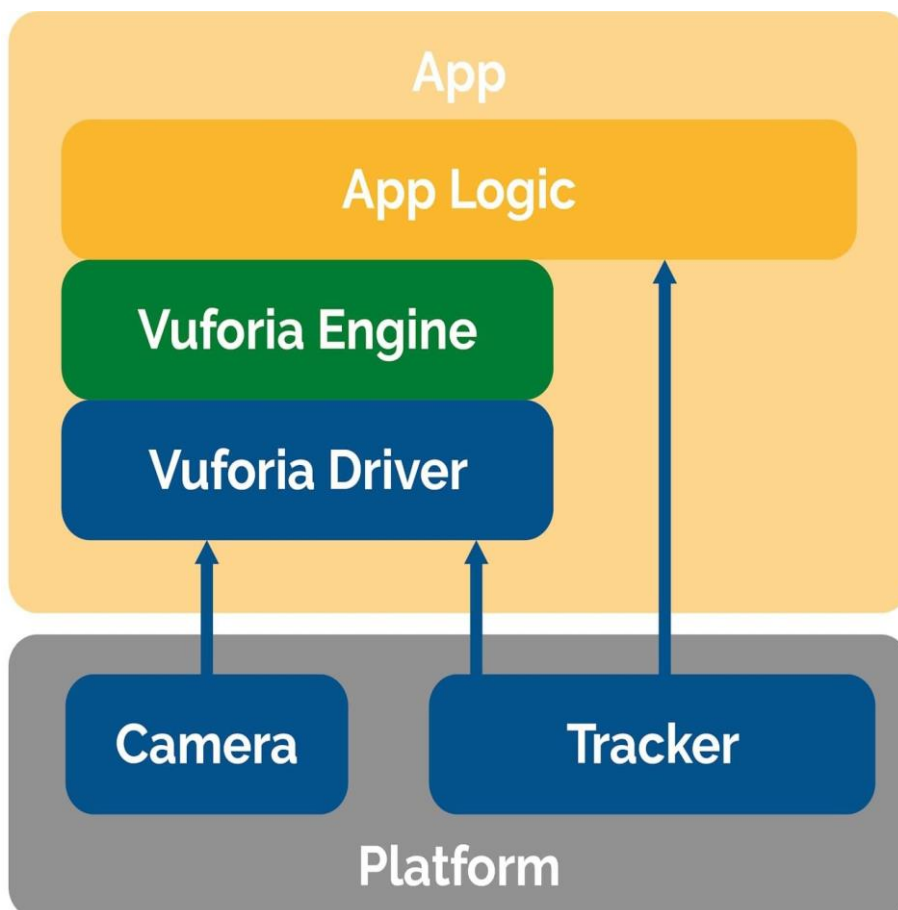
απώλεια της αναγνώρισης και απότομη εξαφάνιση του μοντέλου, υποβαθμίζοντας σημαντικά την εκπαιδευτική εμπειρία.

## 2.8 Πλεονεκτήματα της Vuforia Engine

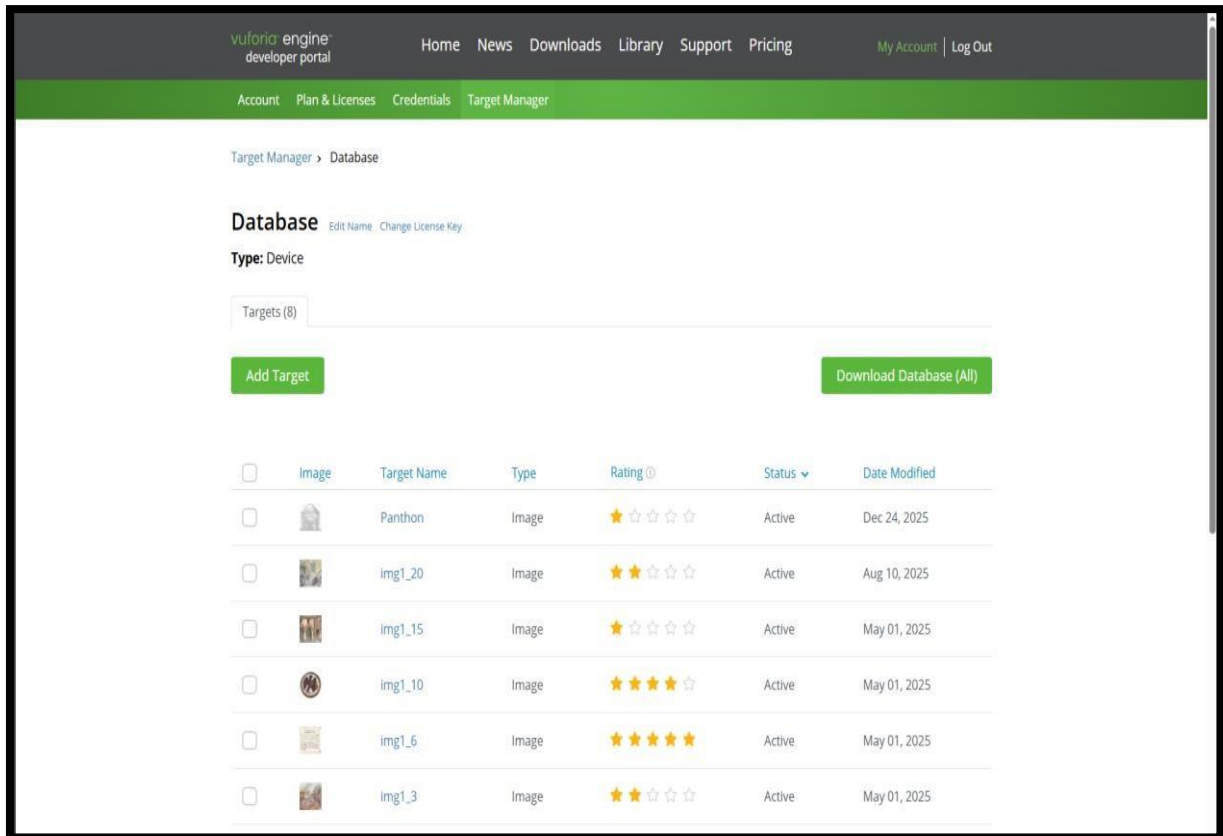
Η Vuforia Engine διαθέτει ορισμένα χαρακτηριστικά που την καθιστούν ιδιαίτερα κατάλληλη για εκπαιδευτικές εφαρμογές AR (Σχήμα 2.8):

1. **Αξιόπιστη αναγνώριση τυπωμένων εικόνων:** Η Vuforia έχει μακρά ιστορία και πολύ ώριμη τεχνολογία στον τομέα της αναγνώρισης επίπεδων εικόνων (Image Tracking), κατάλληλη για εκπαιδευτικό υλικό όπως οι σελίδες σχολικών βιβλίων.
2. **Ευρεία συμβατότητα συσκευών:** Λειτουργεί σε ένα μεγάλο εύρος Android συσκευών, παλαιότερων και νεότερων, χωρίς να απαιτεί εξειδικευμένο υλικό (hardware).
3. **Άμεση ενσωμάτωση με Unity:** Το SDK της λειτουργεί ως plugin μέσα στο Unity Editor, επιτρέποντας τη ρύθμιση Image Targets και 3D μοντέλων απευθείας από το περιβάλλον ανάπτυξης.
4. **Δυνατότητα εξαγωγής σε πολλαπλές πλατφόρμες:** Ο ίδιος κώδικας μπορεί να εξαχθεί για Android και iOS, κάτι χρήσιμο για μελλοντική επέκταση της εφαρμογής.

Συνολικά, στην εφαρμογή ενσωματώθηκαν αρκετές εικόνες-στόχοι από το σχολικό βιβλίο, οργανωμένες σε ενιαία βάση δεδομένων (Σχήμα 2.9).



Σχήμα 2.8: Τα βασικά στάδια επεξεργασίας εικόνας και απόδοσης της μηχανής Vuforia Engine



Σχήμα 2.9 Η βάση δεδομένων (Target Database) της εφαρμογής στο περιβάλλον του Vuforia Target Manager, όπου παρουσιάζεται η λίστα των εικόνων-στόχων και η αξιολόγηση της καταλληλότητάς τους για ανίχνευση (Star Rating).

## 2.9 Λειτουργικό Σύστημα Android

Ως πλατφόρμα στόχευσης (target platform) για την ανάπτυξη και τις δοκιμές της εφαρμογής επιλέχθηκε το λειτουργικό σύστημα Android [14]. Η επιλογή αυτή συνδέεται άμεσα με τον εκπαιδευτικό χαρακτήρα του έργου. Το Android είναι το πιο διαδεδομένο λειτουργικό σύστημα για κινητές συσκευές παγκοσμίως, γεγονός που εξασφαλίζει ευρεία προσβασιμότητα.

Επιπλέον, οι Android tablets είναι γενικά πιο οικονομικές από τις αντίστοιχες συσκευές iOS, γεγονός που τις καθιστά μια πιο προσιτή επιλογή για εκπαιδευτικά περιβάλλοντα. Η εγκατάσταση της εφαρμογής μέσω αρχείων (.apk) γίνεται γρήγορα και χωρίς πολύπλοκες διαδικασίες, κάτι που μελλοντικά θα διευκολύνει τη δοκιμή και αξιολόγηση (testing) σε πραγματικό εκπαιδευτικό περιβάλλον.

## 2.10 Γλώσσα Προγραμματισμού C#

Η λογική της εφαρμογής υλοποιήθηκε εξ ολοκλήρου σε C# [15], τη γλώσσα προγραμματισμού που υποστηρίζεται άμεσα από τη Unity. Πρόκειται για μια σύγχρονη, αντικειμενοστραφή (object-oriented) γλώσσα.

Στα πλαίσια αυτού του έργου, αναπτύχθηκαν εξειδικευμένα scripts σε C#, τα οποία αναλαμβάνουν τον πλήρη έλεγχο της εφαρμογής. Μέσα από αυτά καλύπτονται λειτουργίες όπως η πλοήγηση μεταξύ οθονών, η ροή των μαθημάτων, η φόρτωση ερωτήσεων, η καταγραφή απαντήσεων και σκορ, και ο

χειρισμός των AR αντικειμένων. Η αρχιτεκτονική και η λειτουργία των σημαντικότερων από αυτά αναλύονται εκτενώς στο **Κεφάλαιο 4**.

### **2.11 Επίλογος Κεφαλαίου**

Βάσει των παραπάνω, διαμορφώθηκε το τεχνολογικό πλαίσιο πάνω στο οποίο στηρίχθηκε η υλοποίηση. Αναλύθηκαν οι έννοιες της Επαυξημένης Πραγματικότητας και της Παιχνιδοποίησης, καθώς και τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίησή της: Unity 3D, Vuforia Engine, C# και Android. Το επόμενο κεφάλαιο παρουσιάζει το Εγχειρίδιο Χρήσης, με αναλυτική περιγραφή της εμπειρίας του χρήστη.

## Κεφάλαιο 3ο: Η Παρουσίαση της Εφαρμογής

### 3.1 Εισαγωγή Κεφαλαίου

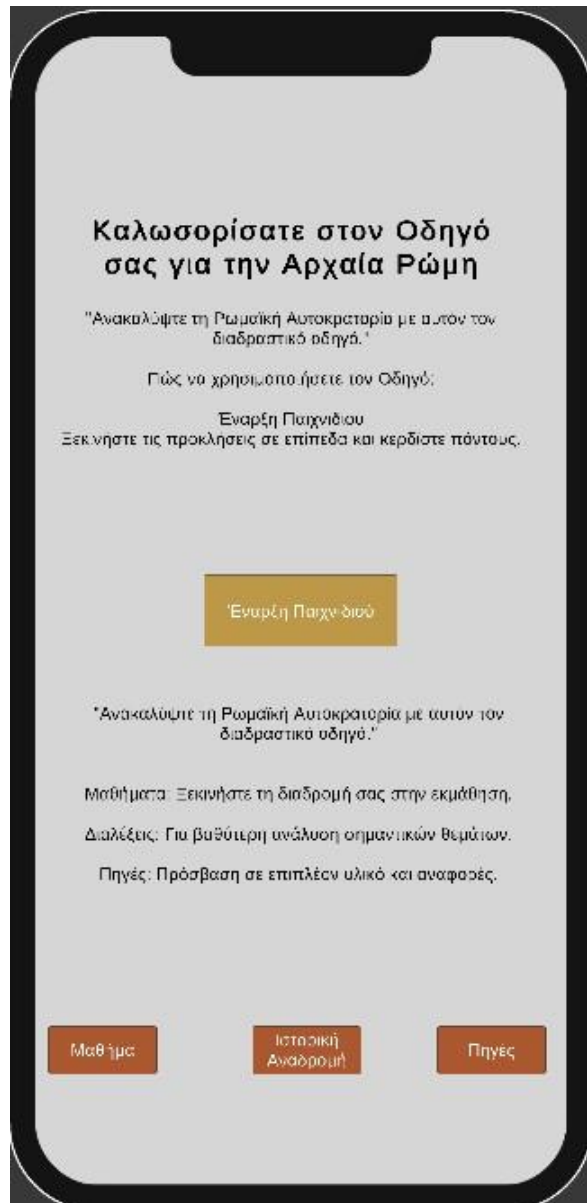
Στο παρόν κεφάλαιο παρουσιάζεται αναλυτικά η τελική μορφή της εκπαιδευτικής εφαρμογής, όπως αυτή σχεδιάστηκε για τον τελικό χρήστη (μαθητή). Η παρουσίαση γίνεται με τη βοήθεια screenshots από την ίδια την εφαρμογή, ώστε να αποτυπωθεί με ακρίβεια η εμπειρία του χρήστη. Καλύπτονται η διεπαφή, το κεντρικό μενού, το εκπαιδευτικό υλικό και οι δοκιμασίες AR. Στόχος είναι να γίνει πλήρως κατανοητή η πορεία του χρήστη (user flow) μέσα στο παιχνίδι.

### 3.2 Κεντρικό Μενού και Βασικές Επιλογές

Κατά την εκκίνηση της εφαρμογής, ο χρήστης μεταφέρεται στην οθόνη του Κεντρικού Μενού. Η σχεδίαση είναι φιλική και προσιτή για μαθητές Δημοτικού, παρέχοντας τέσσερις (4) ξεκάθαρες επιλογές (Σχήμα 3.1):

1. **Έναρξη Παιχνιδιού:** Μεταφέρει τον χρήστη στην επιλογή επιπέδων (Levels), όπου μπορεί να παίξει τα κουίζ της εφαρμογής. Η συγκεκριμένη επιλογή αποτελεί το βασικό σημείο εισόδου στο διαδραστικό μέρος του παιχνιδιού και συνδέεται άμεσα με τη διαδικασία αξιολόγησης του μαθητή.
2. **Το Μάθημα:** Ανοίγει την ενότητα με το θεωρητικό υλικό. Μέσα από αυτήν, ο μαθητής μπορεί να μελετήσει οργανωμένα το εκπαιδευτικό περιεχόμενο πριν προχωρήσει στις ερωτήσεις και στις δοκιμασίες της εφαρμογής.
3. **Ιστορική Αναδρομή:** Παρουσιάζει σημαντικά γεγονότα και μάχες μεταξύ Ελλήνων και Ρωμαίων. Η ενότητα αυτή λειτουργεί υποστηρικτικά προς το μάθημα, προσφέροντας στον χρήστη ένα συνοπτικό ιστορικό πλαίσιο για καλύτερη κατανόηση του περιεχομένου.
4. **Πηγές:** Περιλαμβάνει εξωτερικό πολυμεσικό υλικό (βίντεο και ιστοσελίδες). Με τον τρόπο αυτό, η εφαρμογή επεκτείνει τη μαθησιακή εμπειρία πέρα από το βασικό περιεχόμενό της, δίνοντας πρόσβαση σε πρόσθετο εκπαιδευτικό υλικό.

Η διάταξη και η ονομασία των επιλογών σχεδιάστηκαν με γνώμονα τη διαισθητική χρήση από μαθητές Δημοτικού. Ο χρήστης μπορεί να πλοηγηθεί σε οποιαδήποτε ενότητα με ένα μόνο άγγιγμα, ενώ η επιστροφή στο Κεντρικό Μενού είναι πάντα διαθέσιμη από κάθε οθόνη της εφαρμογής, εξασφαλίζοντας ότι ο μαθητής δεν χάνεται στη ροή του παιχνιδιού. Επιπλέον, η σειρά με την οποία εμφανίζονται οι επιλογές δεν είναι τυχαία — ακολουθεί τη φυσική ροή χρήσης: ο μαθητής ξεκινά από την Έναρξη Παιχνιδιού, μελετά τη θεωρία μέσω του Μαθήματος, αποκτά ιστορικό πλαίσιο από την Ιστορική Αναδρομή και τέλος εξερευνά επιπλέον υλικό μέσω των Πηγών. Αυτή η διάταξη καθοδηγεί υπόρρητα τον χρήστη προς μια ολοκληρωμένη εκπαιδευτική εμπειρία, χωρίς να απαιτεί ρητές οδηγίες.



Σχήμα 3.1 Η οθόνη του Κεντρικού Μενού με τις τέσσερις βασικές επιλογές

### 3.3 Το Μάθημα, Ιστορική Αναδρομή και Πηγές

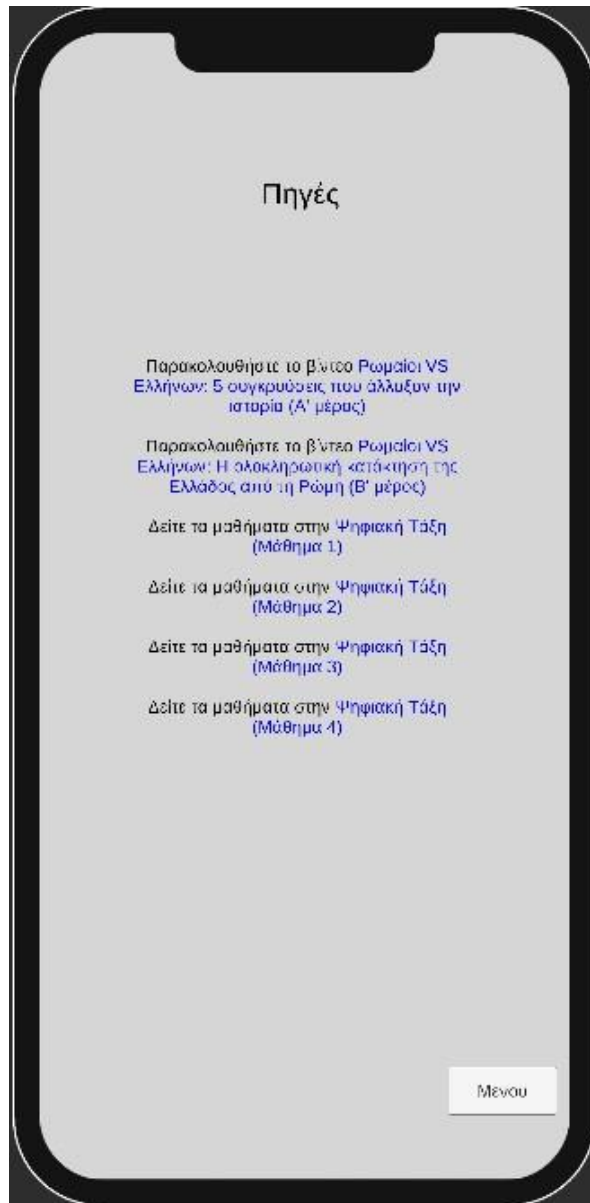
Επιλέγοντας "Το Μάθημα" ή "Ιστορική Αναδρομή", ο μαθητής έχει πρόσβαση στη θεωρία. Η πληροφορία δεν παρουσιάζεται με τη μορφή εκτενών, συμπαγών κειμένων, αλλά είναι οργανωμένη σε μικρές ενότητες με συνοδευτικές εικόνες και ξεκάθαρες χρονολογίες (Σχήμα 3.2).

Ένα ιδιαίτερο χαρακτηριστικό αυτής της ενότητας είναι οι διαδραστικές λέξεις. Ορισμένες λέξεις κλειδιά μέσα στο κείμενο είναι διαδραστικές (clickable). Όταν ο μαθητής τις πατήσει, η εφαρμογή ανοίγει τον περιηγητή (browser) της συσκευής και τον μεταφέρει στο Βικιλεξικό (Wiktionary) για να διαβάσει τον ορισμό της λέξης.

Στην ενότητα "Πηγές", ο χρήστης μπορεί να βρει μια συγκεντρωμένη λίστα με εξωτερικούς συνδέσμους (Σχήμα 3.3). Πατώντας πάνω σε αυτούς, ανακατευθύνεται σε επιλεγμένα βίντεο στο YouTube ή σε εκπαιδευτικές ιστοσελίδες που προσφέρουν οπτικοακουστικό υλικό για τη ζωή και τον πολιτισμό των Ρωμαίων και των Ελλήνων.



Σχήμα 3.2 Οθόνη "Το Μάθημα" με χρονολογίες, εικόνα και διαδραστικές λέξεις-κλειδιά



Σχήμα 3.3 Η ενότητα "Πηγές" με τη λίστα των εξωτερικών συνδέσμων (YouTube / Sites)

### 3.4 Έναρξη Παιχνιδιού και Σύστημα Επιπέδων

Πατώντας "Έναρξη Παιχνιδιού", εμφανίζεται η οθόνη επιλογής επιπέδου (Σχήμα 3.4). Το παιχνίδι ακολουθεί μια γραμμική εξέλιξη δυσκολίας. Αρχικά, μόνο το Επίπεδο 1 είναι διαθέσιμο. Για να ξεκλειδωθεί το επόμενο επίπεδο, ο μαθητής πρέπει να απαντήσει σε όλες τις ερωτήσεις του τρέχοντος επιπέδου χωρίς να εξαντλήσει τις ζωές του.

Κάθε επίπεδο ενσωματώνει δύο μηχανισμούς πίεσης: χρονόμετρο και σύστημα ζωών. Συγκεκριμένα, για κάθε ερώτηση ο μαθητής έχει στη διάθεσή του 15 δευτερόλεπτα. Αν ο χρόνος εξαντληθεί πριν δοθεί απάντηση, χάνεται μία ζωή. Το ίδιο συμβαίνει και με κάθε λανθασμένη απάντηση. Ο μαθητής ξεκινά κάθε επίπεδο με 3 ζωές συνολικά. Αν χαθούν και οι τρεις, το επίπεδο επανεκκινεί από την αρχή. Αν ο

μαθητής ολοκληρώσει όλες τις ερωτήσεις διατηρώντας τουλάχιστον μία ζωή, το επόμενο επίπεδο ξεκλειδώνεται.



Σχήμα 3.4 Η οθόνη επιλογής επιπέδων (Level Selection)

### 3.5 Δομή και Κλιμάκωση των Επιπέδων

Συνολικά, η εφαρμογή περιλαμβάνει πέντε επίπεδα, καθένα από τα οποία αποτελείται από έξι δραστηριότητες. Η σειρά των επιπέδων ακολουθεί μια λογική κλιμάκωσης ως προς το είδος της αλληλεπίδρασης. Ο παρακάτω πίνακας (Πίνακας 3.1) συνοψίζει το περιεχόμενο και τη δομή κάθε επιπέδου.

Επίπεδο	Μνημείο / Θέμα	Δοκιμασίες AR	Ερωτήσεις Κουίζ	Σύνολο
1	Ωδείο Ηρώδου του Αττικού	3 (Σάρωση, Περιστροφή, Επιλογή)	3 (Δύο Επιλογών)	6
2	Πάνθεον της Ρώμης	3 (Σάρωση, Περιστροφή, Επιλογή)	3 (Δύο Επιλογών)	6
3	Κολοσσαίο	3 (Σάρωση, Περιστροφή, Επιλογή)	3 (Δύο Επιλογών)	6
4	Ρωμαϊκή Διοίκηση & Κοινωνία	—	6 (Δύο Επιλογών)	6
5	Ελληνορωμαϊκός Πολιτισμός	—	3 (Δύο Επιλογών) + 3 (Επιλογή Εικόνας)	6

Πίνακας 3.1: Η δομή των πέντε επιπέδων

Όπως φαίνεται στον Πίνακα 3.1, τα τρία πρώτα επίπεδα είναι δομημένα γύρω από ένα συγκεκριμένο αρχαίο μνημείο, το οποίο ο μαθητής εξερευνά μέσω της δοκιμασίας AR πριν απαντήσει στις σχετικές ερωτήσεις. Στο τέταρτο επίπεδο, η εφαρμογή στρέφεται σε ερωτήσεις κειμένου που αφορούν την οργάνωση της ρωμαϊκής κοινωνίας, ενώ στο πέμπτο επίπεδο εισάγεται ο τύπος ερώτησης με επιλογή εικόνας, ώστε ο μαθητής να αναγνωρίσει οπτικά στοιχεία του ελληνορωμαϊκού πολιτισμού. Η σταδιακή αυτή μετάβαση, από τη βιωματική εξερεύνηση μέσω AR προς την εμπέδωση και την οπτική αναγνώριση, διατηρεί το ενδιαφέρον του μαθητή σε όλη τη διάρκεια του παιχνιδιού.

Κατά τη διάρκεια του παιχνιδιού, η οθόνη εμφανίζει τα βασικά στοιχεία του interface (Σχήμα 3.5): στο επάνω μέρος διακρίνονται οι εναπομείνουσες ζωές του μαθητή και το χρονόμετρο αντίστροφης μέτρησης, ενώ στο κάτω μέρος εμφανίζεται η ερώτηση με τις διαθέσιμες επιλογές.



Σχήμα 3.5 Το περιβάλλον (UI) κατά τη διάρκεια του παιχνιδιού, όπου διακρίνονται οι διαθέσιμες ζωές και το χρονόμετρο

### 3.6 Τύποι Ερωτήσεων και Δοκιμασίες AR

Κατά τη διάρκεια του παιχνιδιού, ο μαθητής καλείται να αντιμετωπίσει τρία διαφορετικά είδη ερωτήσεων (Σχήμα 3.6):

1. Ερώτηση Δύο Επιλογών (Κείμενο): Ο μαθητής διαβάζει την εκφώνηση της ερώτησης και πρέπει να επιλέξει τη σωστή απάντηση ανάμεσα σε δύο εναλλακτικές επιλογές κειμένου (Σωστό/Λάθος ή δύο διαφορετικές απαντήσεις).
2. Επιλογή Σωστής Εικόνας: Αντί για κείμενο, παρουσιάζεται μια ενιαία εικόνα η οποία περιλαμβάνει τέσσερα διαφορετικά τμήματα (collage). Ο μαθητής πρέπει να αναγνωρίσει οπτικά και να επιλέξει το σωστό τμήμα που αντιστοιχεί στην ερώτηση.



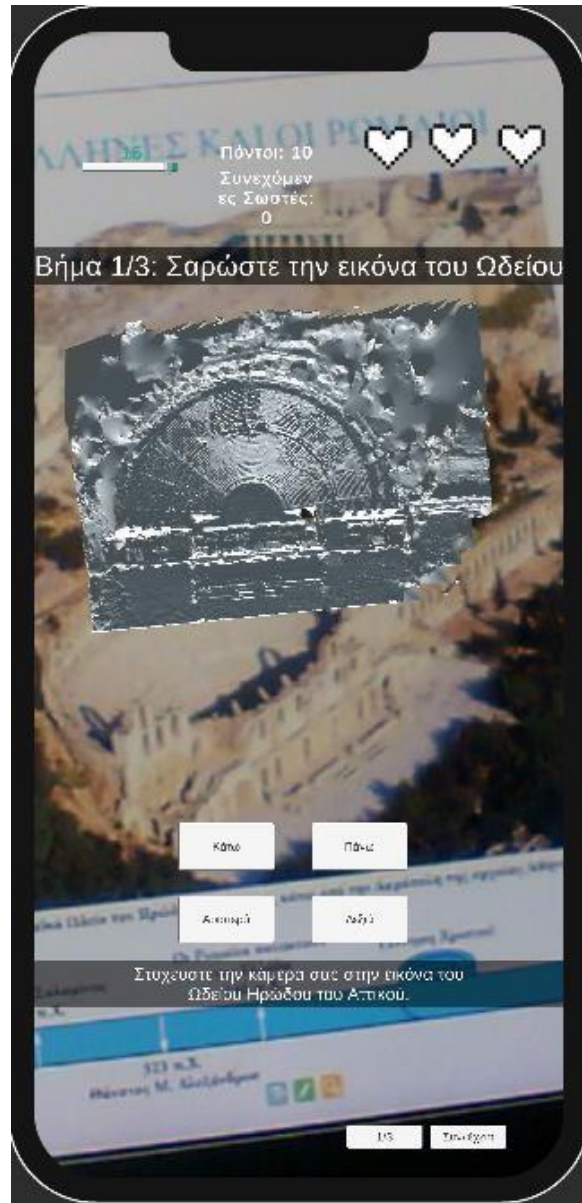
Σχήμα 3.6 Παράδειγμα ερώτησης με επιλογή σωστής απάντησης (Κείμενο) και επιλογή σωστής Εικόνας

### 3. Δοκιμασία Επαυξημένης Πραγματικότητας (AR Game):

Πρόκειται για την πιο διαδραστική λειτουργία της εφαρμογής. Όταν εμφανίζεται μια δοκιμασία AR, ανοίγει η κάμερα της συσκευής και ο μαθητής καλείται να ολοκληρώσει τρία συγκεκριμένα βήματα (Σχήμα 3.7, Σχήμα 3.8):

- **Σκανάρισμα:** Ο μαθητής στοχεύει με την κάμερα μια συγκεκριμένη εικόνα μέσα από το σχολικό του βιβλίο (Image Target).
- **Περιστροφή (Rotate):** Μόλις το σύστημα αναγνωρίσει την εικόνα, εμφανίζεται πάνω της ένα τρισδιάστατο (3D) αντικείμενο. Ο μαθητής πρέπει να χρησιμοποιήσει τα κουμπιά στην οθόνη για να περιστρέψει το 3D μοντέλο.

- **Επιλογή (Click):** Αφού επεξεργαστεί το μοντέλο, ο μαθητής πρέπει να "πατήσει" (click/tap) στο σωστό σημείο πάνω στο 3D αντικείμενο που του ζητάει η ερώτηση, προκειμένου να κερδίσει τους πόντους.



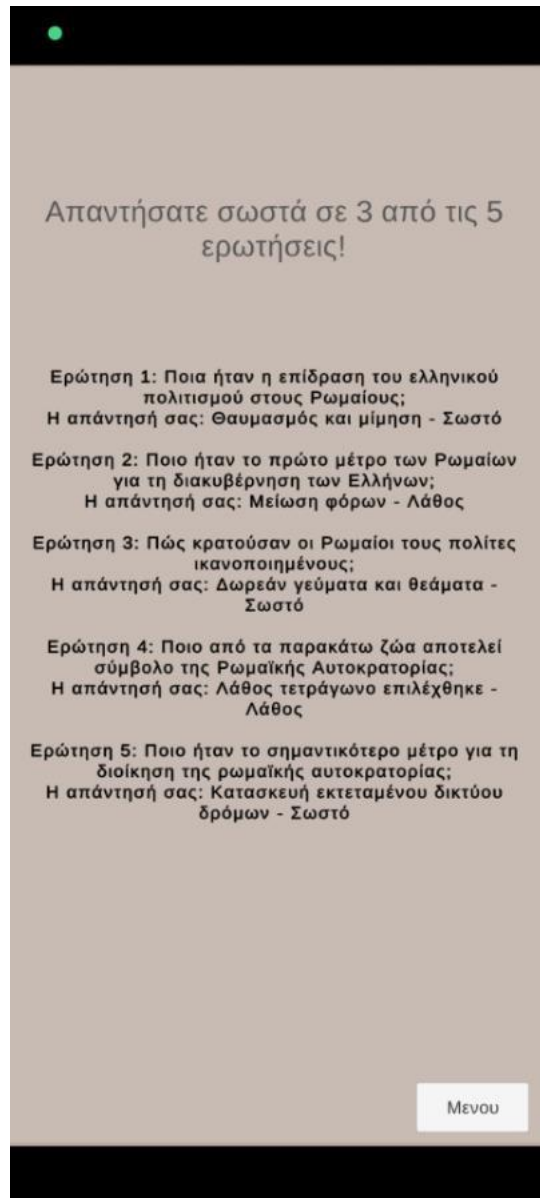
Σχήμα 3.7 Διαδικασία AR: Σκανάρισμα εικόνας βιβλίου και εμφάνιση 3D μοντέλου



Σχήμα 3.8 Διαδικασία AR: Ο χρήστης περιστρέφει το 3D μοντέλο και κάνει κλικ στο ζητούμενο σημείο

### 3.7 Οθόνη Αποτελεσμάτων και Ανατροφοδότησης

Με την ολοκλήρωση όλων των ερωτήσεων ενός επιπέδου, εμφανίζεται η οθόνη Αποτελεσμάτων (Game Stats Panel), η οποία προσφέρει άμεση ανατροφοδότηση στον μαθητή για την απόδοσή του. Όπως φαίνεται στο Σχήμα 3.9, στην κορυφή της οθόνης εμφανίζεται το συνολικό αποτέλεσμα του μαθητή — δηλαδή πόσες ερωτήσεις απάντησε σωστά από το σύνολο (π.χ. «Απαντήσατε σωστά σε 4 από τις 6 ερωτήσεις»). Στη συνέχεια, παρουσιάζεται αναλυτική λίστα με κάθε ερώτηση, την απάντηση που έδωσε ο μαθητής και την ένδειξη «Σωστό» ή «Λάθος», επιτρέποντάς του να εντοπίσει τα σημεία που χρειάζονται περαιτέρω μελέτη. Με αυτόν τον τρόπο, η οθόνη αποτελεσμάτων δεν λειτουργεί απλώς ως τελική βαθμολογία, αλλά ως εργαλείο αυτοαξιολόγησης και ανατροφοδότησης.



Σχήμα 3.9 Η οθόνη τελικών αποτελεσμάτων (Feedback), όπου διακρίνεται η αναλυτική λίστα των σωστών/λανθασμένων απαντήσεων και η τελική καταμέτρηση των πόντων του μαθητή

### 3.8 Επίλογος Κεφαλαίου

Η εφαρμογή προσφέρει μια ολοκληρωμένη εκπαιδευτική εμπειρία που συνδυάζει τη μελέτη θεωρητικού υλικού με την αξιολόγηση μέσω παιχνιδιού. Το σύστημα επιπέδων με χρονόμετρο και ζωές διατηρεί τον μαθητή συγκεντρωμένο, ενώ η οθόνη αποτελεσμάτων του επιτρέπει να εντοπίσει και να κατανοήσει τα λάθη του. Ιδιαίτερα οι δοκιμασίες AR, οι οποίες απαιτούν τον συνδυασμό φυσικού (βιβλίο) και ψηφιακού (3D μοντέλα) περιβάλλοντος, μπορούν να μετατρέψουν την παραδοσιακή μάθηση σε μια πιο διαδραστική και βιωματική εμπειρία.



## Κεφάλαιο 4ο: Ανάπτυξη και Προγραμματιστική Υλοποίηση

### 4.1 Εισαγωγή Κεφαλαίου

Στο παρόν κεφάλαιο αναλύεται η αρχιτεκτονική της εφαρμογής και η προγραμματιστική υλοποίηση της λογικής του παιχνιδιού. Ο κώδικας οργανώθηκε σε ανεξάρτητα, modular Scripts γραμμένα σε C#. Κάθε Script έχει έναν διακριτό ρόλο, εξασφαλίζοντας την ομαλή λειτουργία του συστήματος. Παρακάτω παρουσιάζονται τα σημαντικότερα τμήματα του κώδικα μέσω εικόνων από το περιβάλλον ανάπτυξης (screenshots), και επεξηγείται ο ρόλος των βασικών μεθόδων που ελέγχουν τα μενού, τη διαχείριση δεδομένων, τη ροή των κουίζ και τις λειτουργίες Επαυξημένης Πραγματικότητας (AR).

### 4.2 Διαχείριση Σκηνών και Διεπαφής (ManagerPanelController)

Η πλοήγηση μεταξύ των διαφόρων οθονών της εφαρμογής (Κεντρικό Μενού, Ιστορική Αναδρομή, Μάθημα, Πηγές, κουίζ) δεν πραγματοποιείται με τη φόρτωση νέων σκηνών, αλλά μέσω της εναλλαγής παραθύρων (Panels) εντός του ίδιου καμβά (UI Canvas). Αυτή η λογική εξοικονομεί πόρους συστήματος και κάνει την εφαρμογή πιο γρήγορη.

Ο έλεγχος αυτής της εναλλαγής γίνεται από ένα κεντρικό script (ManagerPanelController), το οποίο λειτουργεί ως διαχειριστής της διεπαφής. Οι βασικές μέθοδοι του script αναλαμβάνουν να ενεργοποιούν (activate) το παράθυρο που επέλεξε ο χρήστης και να απενεργοποιούν (deactivate) όλα τα υπόλοιπα (Σχήμα 4.1).

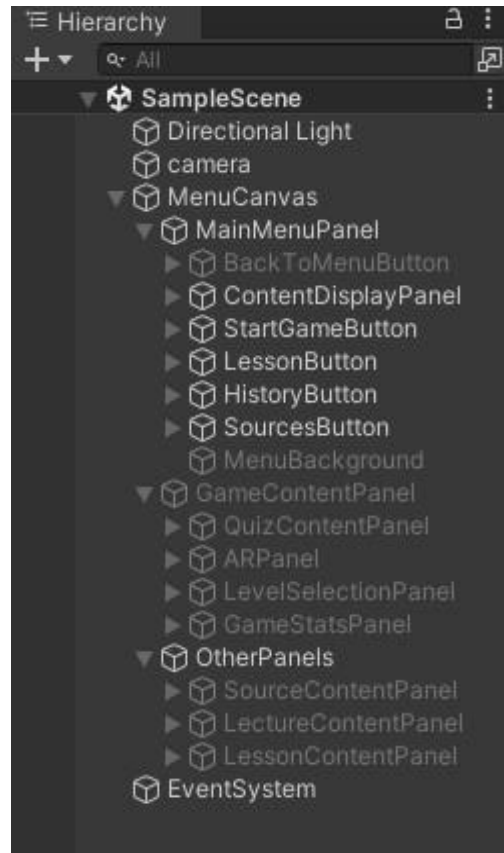
```
public void ShowLessonPanelFromManager()
{
    managerPanel.SetActive(true);
    managerDetailPanel.SetActive(true);
    lessonPanel.SetActive(true);
    lessonDetailPanel.SetActive(false);
    foreach (var button in lessonButtons)
    {
        button.SetActive(true);
    }
    ManagerTitle.text = "Επιλέξτε ένα μάθημα";
    ManagerContent.text = "Παρακαλώ επιλέξτε ένα μάθημα από τα παρακάτω για να ξεκινήσετε.";
    HideMainMenu();
    backToMenuButton.gameObject.SetActive(true);
    quizPanel.SetActive(false);
    sourcePanel.SetActive(false);
    lecturePanel.SetActive(false);
}
```

Σχήμα 4.1 Απόσπασμα κώδικα από το script διαχείρισης παραθύρων (ManagerPanelController), που δείχνει τις μεθόδους ενεργοποίησης/απενεργοποίησης των UI Panels

### 4.3 Ιεραρχία Αντικειμένων στο Unity Editor (Scene Hierarchy)

Σε επίπεδο αρχιτεκτονικής του Unity Editor, ακολουθήθηκε η λογική της ενιαίας σκηνής (Single-Scene Architecture). Αντί να δημιουργηθούν πολλαπλές ξεχωριστές σκηνές (.unity files) για το Μενού, τη Θεωρία και το Παιχνίδι (πρακτική που θα απαιτούσε συνεχείς φορτώσεις (loading screens) και σπατάλη πόρων μνήμης), επιλέχθηκε η χρήση ενός ενιαίου Canvas τύπου Screen Space - Overlay. Όπως φαίνεται στο Σχήμα 4.2, όλα τα γραφικά περιβάλλοντα είναι οργανωμένα ως Child GameObjects κάτω από ένα κεντρικό αντικείμενο MenuCanvas. Το MenuCanvas περιλαμβάνει τρεις κύριες ομάδες: το MainMenuPanel με τα κουμπιά πλοήγησης, το GameContentPanel που περιέχει τα υποπάνελ του

παιχνιδιού (QuizContentPanel, ARPanel, LevelSelectionPanel, GameStatsPanel), και το OtherPanels που φιλοξενεί τα πάνελ περιχομένου (SourceContentPanel, LectureContentPanel, LessonContentPanel).



Σχήμα 4.2 Η ιεραρχία (Hierarchy) της κύριας σκηνής στον Unity Editor, όπου διακρίνεται η οργάνωση των UI Panels κάτω από το κεντρικό Canvas

#### 4.4 Αρχιτεκτονική Δεδομένων: JSON

Μια από τις βασικές αποφάσεις κατά την ανάπτυξη της εφαρμογής ήταν ο διαχωρισμός του περιχομένου (θεωρία, ερωτήσεις) από τον βασικό κώδικα. Για τον σκοπό αυτό, επιλέχθηκε η μορφή JSON (JavaScript Object Notation), μια ελαφριά και εύκολα αναγνώσιμη μορφή ανταλλαγής δεδομένων.

Η χρήση JSON προσφέρει πολλαπλά πλεονεκτήματα: επιτρέπει την ανανέωση του περιχομένου χωρίς να απαιτείται νέο build ολόκληρης της εφαρμογής, μειώνει το μέγεθος των αρχείων C# και κάνει το σύστημα modular και εύκολα επεκτάσιμο.

Για κάθε κύρια λειτουργία της εφαρμογής δημιουργήθηκε ξεχωριστό αρχείο JSON, όπως το lesson.json για τη θεωρία των μαθημάτων και το LessonQuestions.json για τις ερωτήσεις. Η δομή του αρχείου ερωτήσεων υποστηρίζει πολλαπλούς τύπους αξιολόγησης, καθώς κάθε ερώτηση αποτελεί ένα αυτόνομο αντικείμενο (object) μέσα σε έναν πίνακα (array) questions (Σχήμα 4.3, Σχήμα 4.4).

```
"questions": [
  {
    "id": 1,
    "levelID": 1,
    "questionType": "ARPrompt",
    "promptStep": 1,
    "questionText": "Βήμα 1/3: Σαρώστε την εικόνα του Ωδείου",
    "factText": "Στοχεύστε την κάμερα σας στην εικόνα του Ωδείου Ηρώδου του Αττικού.",
    "successText": "Εντοπίστηκε το Ωδείο! Το Ωδείο χτίστηκε το 161 μ.Χ.",
    "modelName": "Odeon",
    "interactionType": "find",
    "targetAngle": 0,
    "targetObjectName": ""
  },
  {
```

Σχήμα 4.3 Απόσπασμα του αρχείου LessonQuestions.json που αναπαριστά μια δοκιμασία Επαυξημένης Πραγματικότητας (questionType: "ARPrompt"), με τα πεδία modelName και interactionType

```
{
  "id": 15,
  "levelID": 3,
  "questionType": "TwoChoice",
  "questionText": "Ποια ήταν η επίδραση του ελληνικού πολιτισμού στους Ρωμαίους;",
  "choices": ["Αδιαφορία", "Θαυμασμός και μίμηση"],
  "correctAnswerIndex": 1
},
{
```

Σχήμα 4.4 Απόσπασμα του αρχείου LessonQuestions.json, όπου διακρίνεται η δομή μιας ερώτησης δύο επιλογών (TwoChoice) και μιας δοκιμασίας AR

```
{
  "id": 25,
  "levelID": 5,
  "questionType": "ImageChoice",
  "questionText": "Κάντε κλικ στο τετράγωνο που δείχνει το Κολοσσαίο.",
  "isImageBased": true,
  "imagePath": "Col",
  "correctAnswerIndex": 3
},
{
  "id": 26,
  "levelID": 5,
  "questionType": "ImageChoice",
  "questionText": "Κάντε κλικ στο τετράγωνο που δείχνει τέχνη με ελληνική επιρροή.",
  "isImageBased": true,
  "imagePath": "art_collage",
  "correctAnswerIndex": 0
},
{
```

Σχήμα 4.5 Απόσπασμα κώδικα JSON που αναπαριστά μια ερώτηση επιλογής τετραγώνου πάνω σε εικόνα-collage (ImageChoice)

Όπως διακρίνεται στα Σχήματα 4.3 έως 4.5, η δομή χρησιμοποιεί συγκεκριμένα "κλειδιά" (keys) για να καθοδηγήσει τον κώδικα της Unity. Η μεταβλητή questionType ενημερώνει το QuizManager για το ποιο γραφικό περιβάλλον (UI) πρέπει να εμφανίσει. Αν η τιμή είναι "TwoChoice", το σύστημα διαβάζει τον πίνακα choices για να γεμίσει τα κουμπιά απαντήσεων και ελέγχει τη σωστή απάντηση μέσω του correctAnswerIndex. Αν η τιμή είναι "ARPrmpt", το σύστημα αγνοεί τα παραδοσιακά κουμπιά και ενεργοποιεί την κάμερα επαυξημένης πραγματικότητας, αναζητώντας το τρισδιάστατο μοντέλο που ορίζεται στη μεταβλητή modelName. Τέλος, στην περίπτωση που η τιμή είναι "ImageChoice", το σύστημα φορτώνει δυναμικά μια ενιαία εικόνα (collage) με βάση τη μεταβλητή imagePath, και τοποθετεί από πάνω της τέσσερα αόρατα κουμπιά (squares) τα οποία αντιστοιχούν στις πιθανές απαντήσεις. Ο μαθητής επιλέγει το σωστό τετράγωνο και η επιλογή του ελέγχεται μέσω του correctAnswerIndex. Η διαδικασία μετατροπής (deserialization) όλων αυτών των δεδομένων στη μνήμη του κινητού αναλύεται στις επόμενες ενότητες.

#### 4.5 Δόμηση Θεωρίας και Υπερσύνδεσμοι στο JSON

Πέρα από τις ερωτήσεις, εξίσου σημαντική ήταν η δυναμική φόρτωση του θεωρητικού περιεχομένου. Για τον σκοπό αυτό, δημιουργήθηκε το αρχείο lesson.json, το οποίο περιέχει τα κείμενα, τους τίτλους, τις χρονολογίες και τις διαδρομές εικόνων (image paths) που προβάλλονται κατά τη διάρκεια του μαθήματος.

Μια από τις βασικές προκλήσεις ήταν η ενσωμάτωση λέξεων-κλειδιών μέσα στο κείμενο ως διαδραστικοί υπερσύνδεσμοι. Δεδομένου ότι για την απόδοση κειμένων χρησιμοποιήθηκε το TextMeshPro της Unity (το οποίο υποστηρίζει διαδραστικούς συνδέσμους και εμπλουτισμένη μορφοποίηση), απαιτούνταν συγκεκριμένη σύνταξη ετικετών (tags) απευθείας μέσα στο JSON. Αντί να δημιουργηθούν ξεχωριστά κουμπιά ή παράλληλα κείμενα, αποφασίστηκε η απευθείας ενσωμάτωση ετικετών <link> στο πεδίο content κάθε σελίδας (Σχήμα 4.6).

```

"lessons": [
  {
    "Title": "Οι Ρωμαίοι κυβερνούν τους Έλληνες",
    "Content": "Το 146 π.Χ., οι Ρωμαίοι εισέβαλαν στην Ελλάδα και άλλαξαν οριζικά τη ζωή των ανθρώπων. Για να κρατήσουν την εξουσία επέλεξαν τη μέθοδο «διαίρει και βασίλευε»<link="define:διαίρει και βασίλευε"><color=#0000FF><u>[1]</u></color></link>:
    "Image": "1lesson",
    "Year": "146 π.Χ."
  },

```

Σχήμα 4.6 Απόσπασμα του αρχείου lesson.json, όπου φαίνεται η χρήση της ετικέτας <link> για τη δημιουργία διαδραστικών λέξεων

Όπως φαίνεται στην παραπάνω εικόνα, όταν μια λέξη πρέπει να είναι διαδραστική, περικλείεται από την ετικέτα <link="define:ΌνομαΛέξης">. Αυτή η δομή επιτρέπει στον αλγόριθμο της εφαρμογής (και συγκεκριμένα στο script LessonLinkHandler) να εντοπίσει ότι ο χρήστης έκανε κλικ στη συγκεκριμένη λέξη. Το πρόθεμα define: λειτουργεί ως αναγνωριστικό (ID) για το σύστημα. Όταν το TextMeshPro διαβάζει το κείμενο από το JSON, χρωματίζει και υπογραμμίζει αυτόματα τη λέξη, καθιστώντας σαφές στον μαθητή ότι πρόκειται για όρο του λεξιλογίου. Μόλις πατηθεί, το σύστημα αφαιρεί το πρόθεμα "define:" και χρησιμοποιεί την καθαρή λέξη για να ανοίξει την αντίστοιχη σελίδα του ελληνικού Βικιλεξικού (<https://el.wiktionary.org/wiki/{όρος}>) στον περιηγητή της συσκευής. Έτσι, για να προστεθεί ένας νέος υπερσύνδεσμος, αρκεί να επεξεργαστεί κανείς το JSON — ο κώδικας δεν χρειάζεται να αλλάξει καθόλου.

#### 4.6 Σύστημα Προόδου και Αποθήκευσης (LevelSelectionController)

Ένα βασικό χαρακτηριστικό της παιχνιδοποίησης (gamification) είναι η γραμμική πρόοδος του παίκτη. Αυτό επιτυγχάνεται μέσω ενός συστήματος όπου τα επίπεδα (levels) είναι αρχικά κλειδωμένα και ξεκλειδώνουν σταδιακά, μόνο όταν ο μαθητής ολοκληρώσει επιτυχώς το προηγούμενο μάθημα. Η διαχείριση αυτής της λογικής γίνεται από το script LevelSelectionController.

Για την υλοποίηση της αποθήκευσης της προόδου χρησιμοποιήθηκε το σύστημα PlayerPrefs της Unity, το οποίο επιτρέπει την τοπική αποθήκευση δεδομένων στη συσκευή του χρήστη. Πιο συγκεκριμένα, η μέθοδος IsLevelUnlocked θεωρεί το πρώτο επίπεδο πάντα ξεκλειδωμένο, ενώ για κάθε επόμενο επίπεδο ελέγχει αν το προηγούμενο (levelID - 1) έχει ολοκληρωθεί, συγκρίνοντάς το με την τιμή LastCompletedLevel που είναι αποθηκευμένη στη μνήμη της συσκευής. Αν το επίπεδο είναι κλειδωμένο, το κουμπί του απενεργοποιείται και εμφανίζεται γκρι (Σχήμα 4.7).

```
private bool IsLevelUnlocked(int levelID)
{
    if (levelID == 1) return true;

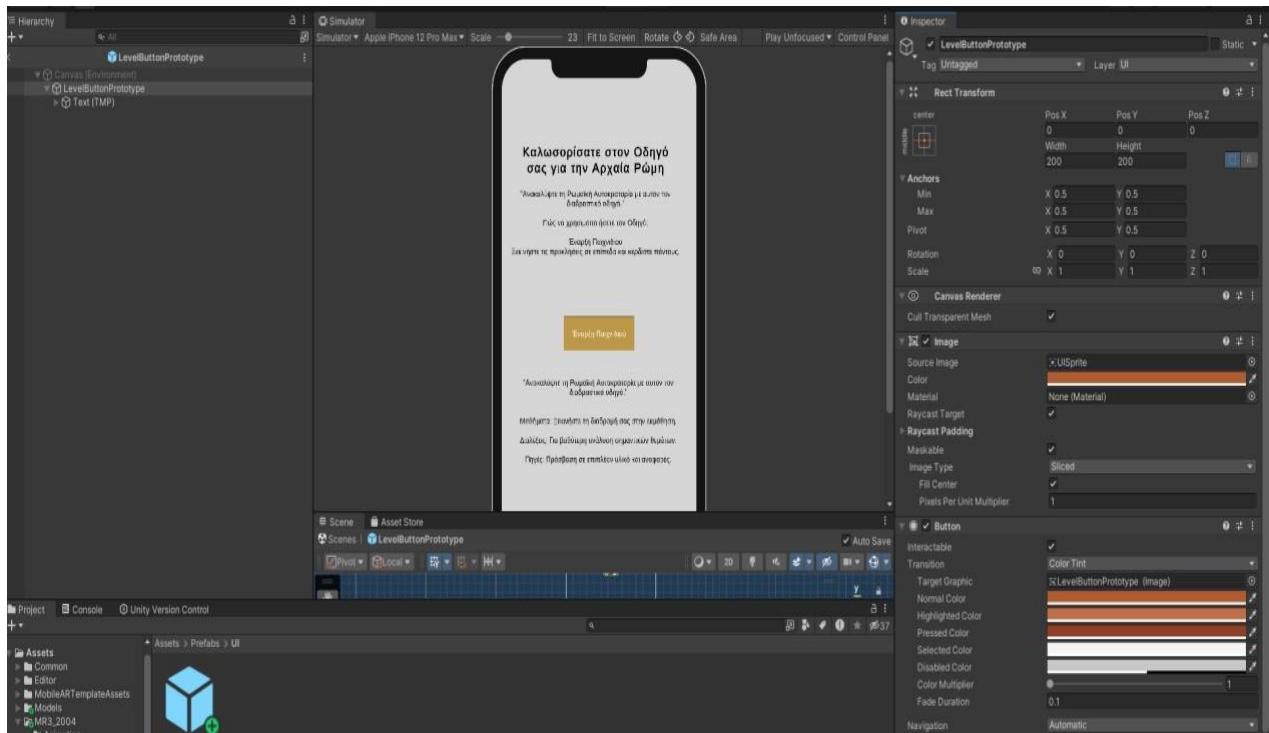
    int lastCompletedLevel = PlayerPrefs.GetInt("LastCompletedLevel", 0);
    int previousLevel = levelID - 1;
    return lastCompletedLevel >= previousLevel;
}
```

Σχήμα 4.7 Απόσπασμα κώδικα από τον LevelSelectionController που δείχνει τη μέθοδο ελέγχου ξεκλειδώματος επιπέδων (IsLevelUnlocked)

#### 4.7 Δημιουργία Γραφικού Περιβάλλοντος με χρήση Prefabs

Για την κατασκευή του μενού επιλογής επιπέδων (Level Selection), καθώς και άλλων επαναλαμβανόμενων στοιχείων της εφαρμογής, υιοθετήθηκε η αρχιτεκτονική των "Prefabs" (Προκατασκευασμένα Αντικείμενα) του Unity. Αντί να σχεδιαστεί χειροκίνητα το κάθε κουμπί επιπέδου (Level 1, Level 2 κ.ο.κ.), δημιουργήθηκε ένα ενιαίο, πρότυπο αντικείμενο με την ονομασία LevelButtonPrototype, το οποίο αποθηκεύτηκε ως Prefab.

Όπως φαίνεται στο Σχήμα 4.8, η προσέγγιση αυτή προσφέρει τεράστια ευελιξία και ομοιομορφία: οποιαδήποτε αλλαγή στα γραφικά (π.χ. χρώμα, στυλ, γραμματοσειρά) εφαρμόζεται κεντρικά στο Prefab και κληρονομείται αυτόματα σε όλα τα κουμπιά της σκηνής. Κατά την εκτέλεση της εφαρμογής, ο κώδικας διαβάζει τα δεδομένα από το αρχείο levels.json (π.χ. τίτλος επιπέδου, αν είναι ξεκλειδωμένο) και δημιουργεί δυναμικά στιγμιότυπα (instantiate) αντίγραφα αυτού του Prefab, προσαρμόζοντας μόνο το κείμενο και την κατάστασή τους.



Σχήμα 4.8: Επεξεργασία του προκατασκευασμένου αντικειμένου (Prefab) "LevelButtonPrototype" μέσα από τον Unity Editor

#### 4.8 Δυναμική Φόρτωση Θεωρίας και Σελιδοποίηση (LessonManager)

Ένα από τα βασικά προβλήματα στον σχεδιασμό εκπαιδευτικών εφαρμογών για κινητά τηλέφωνα είναι η παρουσίαση μεγάλου όγκου κειμένου σε μικρές οθόνες. Η εμφάνιση όλου του κειμένου σε μία ενιαία σελίδα με χρήση κάθετης κύλισης (scrolling) συχνά κουράζει τον μαθητή και μειώνει την αναγνωσιμότητα. Για την αντιμετώπιση αυτού του προβλήματος, αναπτύχθηκε στο script LessonManager ένας μηχανισμός δυναμικής σελιδοποίησης (pagination), ο οποίος χωρίζει το θεωρητικό κείμενο σε μικρότερα, διαχειρίσιμα τμήματα.

Η μέθοδος LoadContentParts δέχεται ως παράμετρο το συνολικό κείμενο του μαθήματος και το διαχωρίζει σε μικρότερα τμήματα. Αρχικά, αρχικοποιεί μια κενή λίστα (contentParts) και ορίζει μέγιστο όριο 500 χαρακτήρων ανά τμήμα. Στη συνέχεια, μέσω της κλάσης Regex, το κείμενο τεμαχίζεται σε λέξεις, οι οποίες επανασυντίθενται σε κομμάτια φροντίζοντας κανένα τμήμα να μην ξεπερνά το καθορισμένο όριο. Τέλος, το εναπομείναν τμήμα προστίθεται στη λίστα εφόσον δεν είναι κενό.

Η πλοήγηση μεταξύ των τμημάτων υλοποιείται μέσω δύο μεθόδων. Η GoToNextPart() ελέγχει αν ο τρέχων δείκτης (currentPartIndex) είναι μικρότερος από το τελευταίο τμήμα· αν ισχύει, αυξάνει τον δείκτη και ενημερώνει το περιβάλλον διεπαφής (UI) μέσω των DisplayCurrentPart(), UpdateNextButton() και UpdatePreviousButton(). Εάν ο χρήστης βρίσκεται ήδη στο τελευταίο τμήμα, καλείται η GoToNextLesson() για μετάβαση στο επόμενο μάθημα. Αντίστοιχα, η GoToPreviousPart() μειώνει τον δείκτη εφόσον αυτός είναι μεγαλύτερος του μηδενός, ενημερώνοντας το UI με τον ίδιο τρόπο.

Αυτή η προσέγγιση μειώνει το γνωστικό φορτίο (cognitive load) του μαθητή, παρουσιάζοντας τη γνώση σε μικρά, διαχειρίσιμα τμήματα (Σχήμα 4.9).

```
private void LoadContentParts(string content)
{
    contentParts = new List<string>();
    int maxLength = 500;

    string[] words = Regex.Split(content, @"(\s+)");
    string currentPart = "";

    foreach (string word in words)
    {
        if (currentPart.Length + word.Length <= maxLength)
        {
            currentPart += word;
        }
        else
        {
            contentParts.Add(currentPart.Trim());
            currentPart = word;
        }
    }

    if (!string.IsNullOrEmpty(currentPart))
    {
        contentParts.Add(currentPart.Trim());
    }
}
```

Σχήμα 4.9 Η μέθοδος LoadContentParts από το LessonManager, όπου υλοποιείται η λογική τεμαχισμού του κειμένου με όριο χαρακτήρων

Στο Σχήμα 4.10 παρουσιάζεται το πλήρες απόσπασμα κώδικα από το LessonManager.cs, όπου φαίνεται η υλοποίηση του τεμαχισμού κειμένου και η λογική σελιδοποίησης που περιγράφηκε παραπάνω.

```
public void GoToNextPart()
{
    if (currentPartIndex < contentParts.Count - 1)
    {
        currentPartIndex++;
        DisplayCurrentPart();
        UpdateNextButton();
        UpdatePreviousButton();
    }
    else
    {
        GoToNextLesson();
    }
}

public void GoToPreviousPart()
{
    if (currentPartIndex > 0)
    {
        currentPartIndex--;
        DisplayCurrentPart();
        UpdateNextButton();
        UpdatePreviousButton();
    }
}
```

Σχήμα 4.10 Απόσπασμα κώδικα από το LessonManager.cs που αναλαμβάνει τον τεμαχισμό του κειμένου και τη σελιδοποίηση

#### 4.9 Σύστημα Ιστορικής Αναδρομής (LectureManager)

Η ίδια αρχιτεκτονική σελιδοποίησης (LoadContentParts) και πλοήγησης (GoToNextPart / GoToPreviousPart) εφαρμόζεται και στο script LectureManager, που διαχειρίζεται την ενότητα «Ιστορική Αναδρομή». Η μόνη ουσιαστική διαφορά είναι ότι το LectureManager διαβάζει τα δεδομένα του από το αρχείο lectures.json και ενσωματώνει επιπλέον ένα slider χρονολογιών (yearSlider), που επιτρέπει στον μαθητή να μεταβεί άμεσα σε συγκεκριμένη χρονική περίοδο της ιστορικής αναδρομής.

#### 4.10 Σύστημα Διαδραστικών Συνδέσμων (LessonLinkHandler)

Για να μπορεί ο μαθητής να εξερευνήσει τις έννοιες του μαθήματος, η θεωρία ενσωματώνει λέξεις-κλειδιά οι οποίες λειτουργούν ως υπερσύνδεσμοι. Η υλοποίηση αυτού του συστήματος απαιτούσε ειδικό χειρισμό, καθώς το κείμενο προβάλλεται μέσω του εργαλείου TextMeshPro (TMP) της Unity.

Το script LessonLinkHandler υλοποιεί τη διεπαφή IPointerClickHandler, η οποία του επιτρέπει να ανιχνεύει τα κλικ του χρήστη πάνω στο κείμενο. Μέσω της ενσωματωμένης συνάρτησης TMP\_TextUtilities.FindIntersectingLink, το σύστημα υπολογίζει ποια ακριβώς λέξη-σύνδεσμο άγγιξε ο χρήστης. Στη συνέχεια, αντλεί το αναγνωριστικό (ID) του συνδέσμου, αντικαθιστά τα κενά με κάτω παύλα ( ) και κωδικοποιεί τους ειδικούς (μη ASCII) χαρακτήρες σε URL-safe μορφή μέσω της μεθόδου EscapeURL. Τέλος, καλεί την Application.OpenURL για να ανοίξει την αντίστοιχη σελίδα στο διαδικτυακό λεξικό (Βικιλεξικό) μέσω του περιηγητή της συσκευής (Σχήμα 4.11).

```

public void OnPointerClick(PointerEventData eventData)
{
    Camera cam = (LessonText.canvas.renderMode == RenderMode.ScreenSpaceOverlay) ? null : Camera.main;
    int linkIndex = TMP_TextUtilities.FindIntersectingLink(lessonText, eventData.position, cam);

    if (linkIndex != -1)
    {
        TMP_LinkInfo linkInfo = lessonText.textInfo.linkInfo[linkIndex];
        string linkID = linkInfo.GetLinkID();

        if (linkID.StartsWith("define:"))
        {
            string term = linkID.Substring("define:".Length);
            string formattedTerm = term.Replace(" ", "_");
            string encodedTerm = UnityEngine.Networking.UnityWebRequest.EscapeURL(formattedTerm);
            string url = "https://el.wiktionary.org/wiki/" + encodedTerm;
            Application.OpenURL(url);
        }
    }
}

```

Σχήμα 4.11 Απόσπασμα από το LessonLinkHandler που δείχνει τη μέθοδο OnPointerClick και τη διαχείριση του υπερσυνδέσμου

#### 4.11 Διαχείριση Τύπων Αλληλεπίδρασης στις Δοκιμασίες AR (ARPromptController)

Οι δοκιμασίες Επαυξημένης Πραγματικότητας ακολουθούν μια συγκεκριμένη αλληλουχία βημάτων: ο παίκτης πρέπει πρώτα να εντοπίσει το Image Target, στη συνέχεια να δει το μοντέλο, και έπειτα να αλληλεπιδράσει μαζί του. Για να οργανωθεί αυτή η λογική, το script ARPromptController χρησιμοποιεί ένα σύστημα διαχείρισης τύπων αλληλεπίδρασης. Στη μέθοδο Update(), το σύστημα ελέγχει το πεδίο interactionType της τρέχουσας δοκιμασίας, το οποίο ορίζεται ως συμβολοσειρά (string) στο αρχείο JSON. Ανάλογα με την τιμή του ("find", "rotation", "click"), καλείται η αντίστοιχη μέθοδος χειρισμού: HandleFindInteraction(), HandleRotationInteraction() ή HandleClickInteraction(). Αυτή η

δομή εξασφαλίζει ότι κάθε τύπος δοκιμασίας εκτελεί μόνο τη δική της λογική, ενώ παράλληλα καθιστά τον κώδικα εύκολα επεκτάσιμο για τυχόν νέες δοκιμασίες AR στο μέλλον (Σχήμα 4.12).

```
private void Update()
{
    if (arPrompts == null || arPrompts.Count == 0) return;
    if (currentPromptIndex >= arPrompts.Count || currentPromptCompleted) return;

    var prompt = arPrompts[currentPromptIndex];

    if (prompt.interactionType == "find")
    {
        HandleFindInteraction(prompt);
    }
    else if (prompt.interactionType == "rotation")
    {
        HandleRotationInteraction(prompt);
    }
    else if (prompt.interactionType == "click")
    {
        HandleClickInteraction(prompt);
    }
}
```

Σχήμα 4.12: Ο έλεγχος των τύπων αλληλεπίδρασης (*interactionType*) στη μέθοδο *Update()* του *ARPromptController.cs*

#### 4.12 Δυναμική Φόρτωση Δεδομένων από JSON

Για να μην είναι οι ερωτήσεις, τα κείμενα και οι χρονολογίες στατικά αποθηκευμένα μέσα στον κώδικα (*hardcoded*), χρησιμοποιήθηκε ένα σύστημα δυναμικής φόρτωσης δεδομένων μέσω αρχείων JSON. Αυτό επιτρέπει την εύκολη ανανέωση του εκπαιδευτικού υλικού χωρίς να απαιτείται επαναπρογραμματισμός.

Αντί για ένα ενιαίο σημείο φόρτωσης, η ευθύνη φόρτωσης JSON έχει ανατεθεί σε κάθε *manager* ξεχωριστά (*QuizManager*, *LessonManager*, *LectureManager*, *LevelSelectionController*, *ARPromptController*), διασφαλίζοντας ότι κάθε υποσύστημα φορτώνει μόνο τα δεδομένα που το αφορούν. Το κοινό μοτίβο που ακολουθούν είναι: εντοπισμός του αρχείου μέσω της *Resources.Load<TextAsset>*, ανάγνωση του περιεχομένου ως απλό κείμενο και μετατροπή του (*deserialization*) σε αντικείμενα C# μέσω του ενσωματωμένου εργαλείου *JsonUtility.FromJson*. Αυτή η προσέγγιση κρατά τη λογική φόρτωσης κοντά στη λογική χρήσης των δεδομένων και διευκολύνει τη συντήρηση (Σχήμα 4.13).

```
private void LoadLesson()
{
    TextAsset jsonTextFile = Resources.Load<TextAsset>("lesson");
    if (jsonTextFile != null)
    {
        LessonData = JsonUtility.FromJson<LessonData>(jsonTextFile.text);
    }
}
```

Σχήμα 4.13: Απόσπασμα κώδικα όπου φαίνεται η τυπική διαδικασία φόρτωσης και αποσειριοποίησης (*deserialization*) ενός αρχείου JSON μέσω του *JsonUtility*

### 4.13 Λογική Παιχνιδιού και Κουίζ (QuizManager)

Το κεντρικό σύστημα που ελέγχει τη ροή του παιχνιδιού και των αξιολογήσεων είναι το QuizManager. Το συγκεκριμένο script έχει ίσως τον πιο σύνθετο ρόλο, καθώς διαχειρίζεται τη γραμμική εξέλιξη των επιπέδων, τον χρόνο και το σκορ του παίκτη.

Οι βασικότερες μέθοδοι του QuizManager περιλαμβάνουν:

- **Μέθοδος Έναρξης Κουίζ (StartQuiz):** Αναλαμβάνει να φορτώσει τις ερωτήσεις του επιλεγμένου επιπέδου από το αρχείο JSON, να επαναφέρει τις διαθέσιμες "ζωές" του παίκτη στην αρχική τιμή και να ξεκινήσει την αντίστροφη μέτρηση του χρονομέτρου.
- **Μέθοδοι Ελέγχου Απάντησης (OnAnswerSelected / ValidateSquare):** Λαμβάνουν την απάντηση που επέλεξε ο παίκτης (είτε επιλογή κειμένου μέσω της OnAnswerSelected, είτε επιλογή τετραγώνου εικόνας μέσω της ValidateSquare) και τη συγκρίνουν με τη σωστή απάντηση που έχει φορτωθεί από το JSON. Αν η απάντηση είναι σωστή, αυξάνεται το σκορ· διαφορετικά αφαιρείται μία ζωή (Σχήμα 4.13).

**Μέθοδοι Τερματισμού (EndQuiz / RestartLevel):** Ελέγχουν τις συνθήκες τερματισμού. Αν τελειώσουν οι ζωές, καλείται η RestartLevel που επανεκκινεί το επίπεδο από την αρχή. Αν απαντηθούν όλες οι ερωτήσεις, καλείται η EndQuiz, η οποία ενημερώνει το σύστημα αποθήκευσης (PlayerPrefs) ώστε να ξεκλειδώσει το επόμενο επίπεδο. Το σύστημα των ζώων και του χρόνου αποτελεί βασικό στοιχείο της παιχνιδοποίησης. Ο παίκτης ξεκινά κάθε επίπεδο με έναν συγκεκριμένο αριθμό ζώων, ο οποίος ορίζεται ως σταθερά (MAX\_LIVES) στην αρχή του script, ώστε να μπορεί εύκολα να τροποποιηθεί. Αντίστοιχα, για κάθε ερώτηση ορίζεται ένα χρονικό όριο απάντησης (QUESTION\_TIME\_LIMIT). Όταν ο χρόνος εξαντληθεί χωρίς απάντηση, η ερώτηση θεωρείται λανθασμένη και ο παίκτης χάνει μία ζωή. Με τον τρόπο αυτό, το χρονομέτρο προσθέτει μια ελαφριά πίεση που κρατά τον μαθητή συγκεντρωμένο, χωρίς όμως να τον αγχώνει υπερβολικά, καθώς ο διαθέσιμος χρόνος είναι επαρκής για την ανάγνωση και την κατανόηση κάθε ερώτησης.

Σημαντικό ρόλο στην υλοποίηση παίζει και ο πίνακας userAnswers, στον οποίο καταγράφονται οι απαντήσεις του μαθητή καθ' όλη τη διάρκεια του επιπέδου. Τα δεδομένα αυτά δεν χρησιμοποιούνται μόνο για τον υπολογισμό του τελικού σκορ, αλλά αξιοποιούνται και στην οθόνη αναλυτικής ανατροφοδότησης που εμφανίζεται στο τέλος, όπου ο μαθητής βλέπει ποιες ερωτήσεις απάντησε σωστά και ποιες λάθος. Με αυτόν τον τρόπο, το QuizManager λειτουργεί όχι μόνο ως μηχανισμός βαθμολόγησης, αλλά και ως εργαλείο ανατροφοδότησης που βοηθά τον μαθητή να εντοπίσει τα σημεία στα οποία χρειάζεται περαιτέρω μελέτη.

```

public void OnAnswerSelected(int index)
{
    timerActive = false;

    float timeBonus = Mathf.Max(0, questionTimer / QUESTION_TIME_LIMIT);
    int bonusPoints = Mathf.RoundToInt(timeBonus * 10);

    userAnswers[currentQuestionIndex] = index;

    if (feedbackText != null)
        feedbackText.gameObject.SetActive(true);

    if (index == correctSquareIndex || index == correctAnswerIndex)
    {
        streak++;

        int points = 1 + bonusPoints;
        if (streak >= STREAK_THRESHOLD)
            points *= 2;

        score += points;
        UpdatePointsUI();

        if (feedbackText != null)
        {
            feedbackText.text = (streak >= STREAK_THRESHOLD)
                ? "Σωστό! x2 Πόντοι!"
                : "Σωστό!";
            feedbackText.color = Color.green;
        }

        UpdateStreakUI();
    }
    else
    {
        if (feedbackText != null)
        {
            feedbackText.text = "Λάθος.";
            feedbackText.color = Color.red;
        }

        lives--;
        streak = 0;
        UpdateLivesUI();
        UpdateStreakUI();

        if (lives <= 0)
        {
            RestartLevel();
            return;
        }
    }
}

```

Σχήμα 4.14: Απόσπασμα κώδικα από το QuizManager, με έμφαση στις μεθόδους ελέγχου απαντήσεων (OnAnswerSelected) και διαχείρισης ζώων/σκορ

#### 4.14 Χρονοκαθυστέρηση και Ασύγχρονη Εκτέλεση (Invoke Method)

Όταν ο μαθητής επιλέγει μια απάντηση, το σύστημα εμφανίζει οπτική ανατροφοδότηση (π.χ. πράσινο ή κόκκινο χρώμα) και περιμένει 1 δευτερόλεπτο πριν προχωρήσει στην επόμενη ερώτηση. Για να μην "παγώσει" η εφαρμογή κατά τη διάρκεια αυτής της αναμονής, χρησιμοποιήθηκε η μέθοδος Invoke() της Unity.

Όπως φαίνεται στο Σχήμα 4.15, μετά από κάθε απάντηση αυξάνεται ο δείκτης ερωτήσεων (currentQuestionIndex). Αν υπάρχουν άλλες ερωτήσεις, καλείται η Invoke(nameof(LoadQuestion), 1f), η οποία φορτώνει την επόμενη ερώτηση έπειτα από 1 δευτερόλεπτο· διαφορετικά καλείται η Invoke(nameof(EndQuiz), 1f) για τον τερματισμό του επιπέδου. Έτσι η εφαρμογή συνεχίζει να λειτουργεί κανονικά κατά τη διάρκεια της αναμονής.

```

currentQuestionIndex++;

if (currentQuestionIndex < selectedQuestions.Count)
    Invoke(nameof(LoadQuestion), 1f);
else
    Invoke(nameof(EndQuiz), 1f);
}

```

Σχήμα 4.15: Χρήση της μεθόδου *Invoke()* στο *QuizManager.cs* για την εισαγωγή χρονοκαθυστέρησης μεταξύ των ερωτήσεων

#### 4.15 Σύστημα Επιβράβευσης και Σερί Απαντήσεων (Streak & Time Bonus)

Για την ενίσχυση της παρακίνησης, το παιχνίδι δεν περιορίζεται στην απλή καταγραφή σωστών και λανθασμένων απαντήσεων, αλλά ανταμείβει επιπλέον την ταχύτητα και τη συνέπεια του παίκτη. Οι μηχανισμοί αυτοί υλοποιήθηκαν στο script *QuizManager*. Η ταχύτητα αξιολογείται μέσω της μεταβλητής *questionTimer*: όταν ο παίκτης απαντήσει σωστά, ο υπολειπόμενος χρόνος μετατρέπεται σε επιπλέον πόντους (*bonusPoints*).

Παράλληλα λειτουργεί το σύστημα Σερί (Streak): κάθε διαδοχική σωστή απάντηση αυξάνει τη μεταβλητή *streak*, και εφόσον αυτή φτάσει το όριο *STREAK\_THRESHOLD* (ορισμένο στο 3), ενεργοποιείται πολλαπλασιαστής σκορ *x2*. Στην οθόνη εμφανίζεται τότε το μήνυμα «Σωστό! *x2* Πόντοι!» με πράσινο χρώμα, ενώ σε λανθασμένη απάντηση το σερί μηδενίζεται και ο παίκτης χάνει μια ζωή (Σχήμα 4.16).

```

public void OnAnswerSelected(int index)
{
    timerActive = false;

    float timeBonus = Mathf.Max(0, questionTimer / QUESTION_TIME_LIMIT);
    int bonusPoints = Mathf.RoundToInt(timeBonus * 10);

    userAnswers[currentQuestionIndex] = index;

    if (feedbackText != null)
        feedbackText.gameObject.SetActive(true);

    if (index == correctSquareIndex || index == correctAnswerIndex)
    {
        streak++;

        int points = 1 + bonusPoints;
        if (streak >= STREAK_THRESHOLD)
            points *= 2;

        score += points;
        UpdatePointsUI();

        if (feedbackText != null)
        {
            feedbackText.text = (streak >= STREAK_THRESHOLD)
                ? "Σωστό! x2 Πόντοι!"
                : "Σωστό!";
            feedbackText.color = Color.green;
        }

        UpdateStreakUI();
    }
}

```

Σχήμα 4.16: Απόσπασμα κώδικα από το *QuizManager* που δείχνει τον υπολογισμό του Time Bonus και τη διαχείριση του Streak System

#### 4.16 Αναλυτική Ανατροφοδότηση (Detailed Feedback)

Στο τέλος κάθε επιπέδου, είναι σημαντικό ο μαθητής να βλέπει όχι μόνο το συνολικό του σκορ, αλλά και ποια λάθη έκανε. Για τον σκοπό αυτό αναπτύχθηκε η μέθοδος ShowDetailedFeedback().

Η μέθοδος συλλέγει τις απαντήσεις που αποθηκεύτηκαν στον πίνακα userAnswers και τις συγκρίνει με τη λίστα των σωστών απαντήσεων. Στη συνέχεια επεξεργάζεται τις ερωτήσεις του επιπέδου μία προς μία, καταγράφοντας την εκφώνηση, την επιλογή του χρήστη και οπτική ένδειξη ορθότητας.

Τέλος, το κείμενο αυτό αποδίδεται στο πεδίο ανατροφοδότησης, ενώ καλείται η εντολή LayoutRebuilder.ForceRebuildLayoutImmediate της Unity, ώστε το παράθυρο κύλισης (Scroll View) να αναπροσαρμόσει αυτόματα το μέγεθός του βάσει του περιεχομένου (Σχήμα 4.17).

```

void ShowDetailedFeedback()
{
    if (detailedFeedbackContainer == null || allFeedbackText == null)
        return;

    detailedFeedbackContainer.SetActive(true);

    System.Text.StringBuilder sb = new System.Text.StringBuilder();
    int shownNumber = 1;

    for (int i = 0; i < selectedQuestions.Count; i++)
    {
        var q = selectedQuestions[i];
        if (q.questionType == "ARPrompt") continue;

        int ua = (userAnswers != null && i < userAnswers.Length) ? userAnswers[i] : -1;
        string yourAnswerText;

        if (q.isImageBased && q.questionType == "ImageChoice")
        {
            if (ua == q.correctAnswerIndex)
                yourAnswerText = "Σωστό τετράγωνο";
            else if (ua == -1)
                yourAnswerText = "Καμία απάντηση";
            else
                yourAnswerText = "Λάθος τετράγωνο";
        }
        else
        {
            if (ua >= 0 && q.choices != null && ua < q.choices.Length)
                yourAnswerText = q.choices[ua];
            else
                yourAnswerText = "Καμία απάντηση";
        }

        bool isCorrect = (ua == q.correctAnswerIndex);

        sb.AppendLine($"{shownNumber}. {q.questionText}");
        sb.AppendLine($"{yourAnswerText} - {(isCorrect ? "Σωστό" : "Λάθος")}");
        sb.AppendLine();
        shownNumber++;
    }
}

```

Σχήμα 4.17: Η μέθοδος ShowDetailedFeedback, όπου διακρίνεται η χρήση του StringBuilder για τη δημιουργία της αναλυτικής αναφοράς

#### 4.17 Ερωτήσεις Βασισμένες σε Εικόνες (Image-Based Questions)

Για να μην περιορίζεται η αξιολόγηση αποκλειστικά σε ερωτήσεις κειμένου, αναπτύχθηκε ένα σύστημα οπτικών ερωτήσεων όπου ο μαθητής καλείται να παρατηρήσει και να αλληλεπιδράσει με εικόνες. Η λειτουργία αυτή ελέγχεται μέσω της μεθόδου SetupImageQuestion εντός του QuizManager. Κατά τη φόρτωση μιας νέας ερώτησης από το αρχείο JSON, το σύστημα ελέγχει τη σημαία (flag) isImageBased. Αν η τιμή είναι true, η εφαρμογή αποκρύπτει τα κουμπιά κειμένου (answerButtons[i].gameObject.SetActive(false)) και προετοιμάζει το γραφικό περιβάλλον για την προβολή εικόνων. Στη συνέχεια, μέσω της εντολής Resources.Load<Sprite>, ανακτάται δυναμικά η εικόνα της ερώτησης βάσει της διαδρομής imagePath που ορίζεται στο JSON, και τοποθετείται στο

κεντρικό πλαίσιο προβολής (questionImage). Ο δείκτης της σωστής απάντησης αποθηκεύεται στη μεταβλητή correctSquareIndex, ώστε κατά την επιλογή του χρήστη (μέσω της μεθόδου ValidateSquare) το σύστημα να συγκρίνει την επιλογή και να προχωρήσει στη βαθμολόγηση (Σχήμα 4.18).

```
private void SetupImageQuestion(Question question)
{
    if (questionImage != null)
        questionImage.SetActive(true);

    foreach (var button in answerButtons)
        button.gameObject.SetActive(false);

    string imagePath = !string.IsNullOrEmpty(question.imagePath)
        ? question.imagePath
        : (question.imageOptions != null && question.imageOptions.Length > 0
            ? question.imageOptions[0]
            : "");

    Sprite imageSprite = Resources.Load<Sprite>(imagePath);
    if (imageSprite != null)
    {
        questionImage.GetComponent<Image>().sprite = imageSprite;
    }

    correctSquareIndex = question.correctAnswerIndex;
    correctAnswerIndex = -1;

    for (int i = 0; i < squares.Length; i++)
    {
        squares[i].gameObject.SetActive(true);
    }
}
```

Σχήμα 4.18: Απόσπασμα κώδικα από το QuizManager, εστιάζοντας στη μέθοδο SetupImageQuestion, η οποία διαχειρίζεται τη φόρτωση και προβολή των ερωτήσεων εικόνας

#### 4.18 Δυναμική Φόρτωση Πόρων (Resources.Load)

Όπως αναφέρθηκε προηγουμένως, βασική αρχιτεκτονική επιλογή ήταν ο διαχωρισμός του περιεχομένου από τον κώδικα. Αυτή η αρχή εφαρμόστηκε και στη διαχείριση των εικόνων: αντί να συνδεθούν οι πόροι απευθείας στη σκηνή μέσω του Inspector (hard-reference), τα ονόματα των αρχείων αποθηκεύτηκαν ως απλές συμβολοσειρές (strings) στο αρχείο JSON.

Κατά την εκτέλεση της εφαρμογής, το σύστημα διαβάζει το όνομα αρχείου από το JSON και χρησιμοποιεί την εντολή Resources.Load<Sprite>() για να ανακτήσει την αντίστοιχη εικόνα από τον ειδικό φάκελο Resources της Unity. Με αυτόν τον τρόπο, οποιαδήποτε αλλαγή στο περιεχόμενο (π.χ. αντικατάσταση ή προσθήκη εικόνας) πραγματοποιείται αποκλειστικά μέσω του JSON, χωρίς να απαιτείται τροποποίηση του κώδικα ή νέο build της εφαρμογής (Σχήμα 4.19).

```
private void SetupImageQuestion(Question question)
{
    if (questionImage != null)
        questionImage.SetActive(true);

    foreach (var button in answerButtons)
        button.gameObject.SetActive(false);

    string imagePath = !string.IsNullOrEmpty(question.imagePath)
        ? question.imagePath
        : (question.imageOptions != null && question.imageOptions.Length > 0
            ? question.imageOptions[0]
            : "");

    Sprite imageSprite = Resources.Load<Sprite>(imagePath);
    if (imageSprite != null)
    {
        questionImage.GetComponent<Image>().sprite = imageSprite;
    }

    correctSquareIndex = question.correctAnswerIndex;
    correctAnswerIndex = -1;

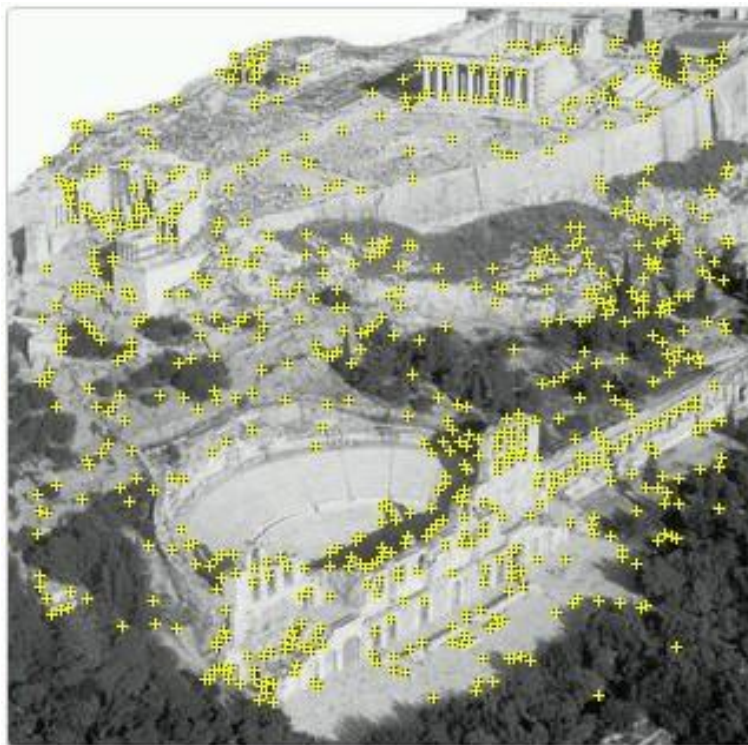
    for (int i = 0; i < squares.Length; i++)
    {
        squares[i].gameObject.SetActive(true);
    }
}
```

Σχήμα 4.19: Δυναμική φόρτωση εικόνων μέσω της `Resources.Load<Sprite>()` στον κώδικα της εφαρμογής

#### 4.19 Ενσωμάτωση Vuforia Engine, AR Camera και Διαδικασία Αναγνώρισης

Η εμπειρία Επαυξημένης Πραγματικότητας (AR) της εφαρμογής βασίζεται στο Vuforia Engine. Για την υλοποίηση αναγνώρισης εικόνων (Image Tracking) και την εμφάνιση τρισδιάστατων μοντέλων, ακολουθήθηκε συγκεκριμένη αλληλουχία τεχνικών βημάτων μεταξύ της διαδικτυακής πλατφόρμας Vuforia και του Unity Editor.

**Δημιουργία βάσης δεδομένων στο Vuforia Target Manager:** Οι εικόνες από το σχολικό βιβλίο αναρτήθηκαν στο Vuforia Developer Portal, όπου ο αλγόριθμος ανέλυσε κάθε εικόνα εντοπίζοντας περιοχές έντονης αντίθεσης (Feature Points). Η πλατφόρμα αξιολόγησε την καταλληλότητα κάθε εικόνας μέσω βαθμολογίας (star rating), επιβεβαιώνοντας ότι οι σελίδες του βιβλίου είναι κατάλληλες για ανίχνευση (Σχήμα 4.20). Οι εικόνες αυτές οργανώθηκαν σε ενιαία βάση δεδομένων (Target Database).



Σχήμα 4.20: Το περιβάλλον του Vuforia Target Manager, όπου διακρίνονται τα "Feature Points" (κίτρινοι σταυροί) πάνω σε μια εικόνα του σχολικού βιβλίου

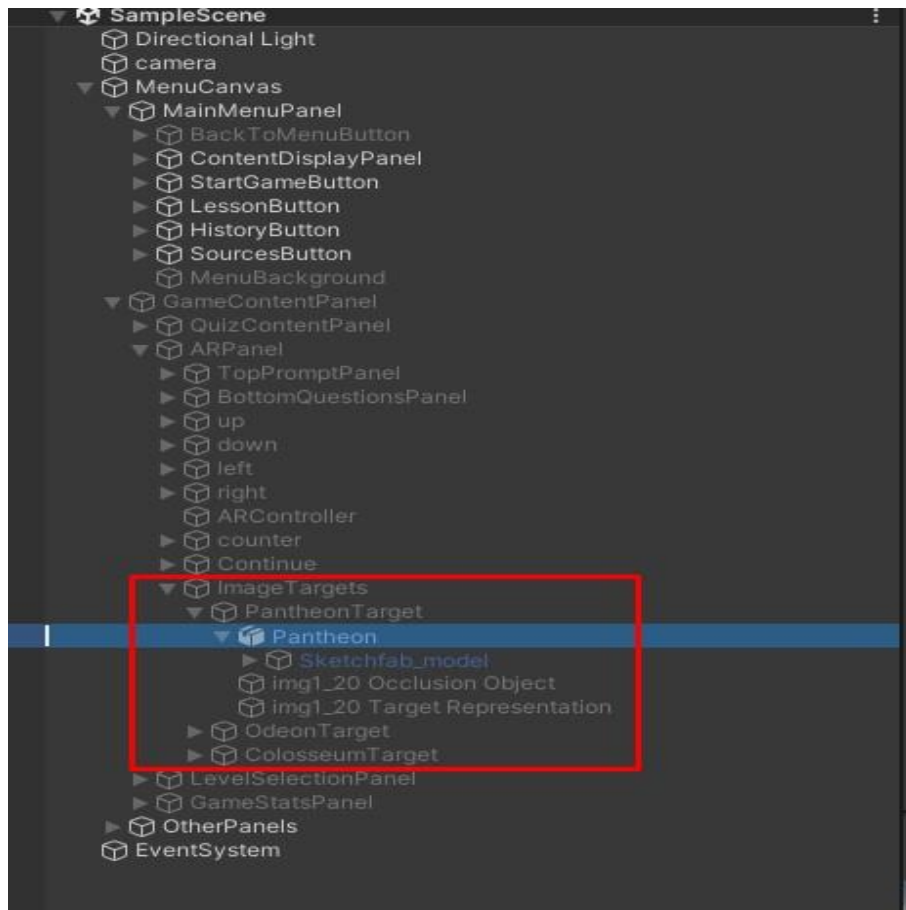
**Εισαγωγή στο Unity και διαμόρφωση License Key:** Η βάση δεδομένων εξήχθη ως πακέτο (.unitypackage) και εισήχθη στο project του Unity. Παράλληλα, μέσω των ρυθμίσεων Vuforia Configuration, καταχωρήθηκε το App License Key, το οποίο συνδέει την εφαρμογή με τον λογαριασμό ανάπτυξης και ενεργοποιεί τις AR λειτουργίες στη συσκευή (Σχήμα 4.21).



Σχήμα 4.21: Το παράθυρο Vuforia Configuration μέσα στον Unity Editor, όπου εισάγεται το App License Key

**Παραμετροποίηση σκηνής με την AR Camera:** Ένα από τα κρίσιμα βήματα ήταν η αντικατάσταση της προεπιλεγμένης κάμερας του Unity (Main Camera) με το εξειδικευμένο component της Vuforia, την AR Camera. Σε αντίθεση με μια απλή κάμερα απόδοσης γραφικών (rendering), η AR Camera ελέγχει απευθείας την κάμερα της συσκευής, ρυθμίστηκε δε ώστε να εστιάζει αυτόματα (Continuous Auto-Focus) και να τροφοδοτεί σε πραγματικό χρόνο το οπτικό σήμα στους αλγορίθμους αναγνώρισης της Vuforia.

**Σύνδεση Image Targets με το τρισδιάστατο περιεχόμενο:** Στη συνέχεια εισήχθησαν στη σκηνή αντικείμενα τύπου Image Target, τα οποία αντιστοιχίστηκαν με τις εικόνες της βάσης δεδομένων. Τα τρισδιάστατα μοντέλα τοποθετήθηκαν ιεραρχικά ως child objects κάτω από τα αντίστοιχα targets (Σχήμα 4.22).



Σχήμα 4.22: Η ιεραρχία ενός Image Target (π.χ. ImageTarget\_Colosseum) στον Unity Editor, όπου το 3D μοντέλο λειτουργεί ως εξαρτώμενο αντικείμενο (Child GameObject)

**Λειτουργία κατά την εκτέλεση:** Κατά την εκτέλεση της εφαρμογής, η AR Camera σαρώνει συνεχώς το περιβάλλον αναζητώντας τα Feature Points της βάσης δεδομένων. Όταν αυτά εντοπιστούν στον πραγματικό χώρο (status: TRACKED), το σύστημα συντεταγμένων της κάμερας ευθυγραμμίζεται με την πραγματική εικόνα, ενεργοποιώντας αυτόματα τα τρισδιάστατα μοντέλα που είναι συνδεδεμένα στο αντίστοιχο Image Target. Από εκεί, τα scripts (όπως το ARPromptController) αναλαμβάνουν τη διαδραστικότητα.

#### 4.20 AR Δοκιμασίες (ARPromptController)

Η πιο πολύπλοκη και εξελιγμένη λειτουργία της εφαρμογής είναι η ενσωμάτωση των δοκιμασιών Επαυξημένης Πραγματικότητας. Ο έλεγχος αυτού του συστήματος γίνεται μέσα από το script ARPromptController, το οποίο συνεργάζεται με το Vuforia Engine SDK. Η δοκιμασία AR χωρίζεται προγραμματιστικά σε τρία (3) διακριτά στάδια αλληλεπίδρασης, τα οποία ελέγχονται συνεχώς μέσω της μεθόδου Update().

##### Στάδιο 1: Αναγνώριση Εικόνας (Scanning & Tracking)

Στην αρχή της δοκιμασίας, το σύστημα ενεργοποιεί την κάμερα και περιμένει να εντοπιστεί η κατάλληλη εικόνα-στόχος (Image Target) από το σχολικό βιβλίο. Το script διαθέτει ένα λεξικό (Dictionary) που αποθηκεύει όλα τα διαθέσιμα τρισδιάστατα μοντέλα. Μέσω της μεθόδου SwitchToModel, η εφαρμογή ελέγχει ποιο είναι το ζητούμενο μοντέλο (βάσει των δεδομένων JSON) και το καθιστά ενεργό (SetActive(true)). Παράλληλα, το σύστημα παρακολουθεί την κατάσταση

αναγνώρισης (Tracking Status) μέσω του ObserverBehaviour. Αν το Vuforia κλειδώσει τον στόχο (status TRACKED), η διαδικασία προχωρά στο επόμενο στάδιο.

### Στάδιο 2: Περιστροφή του Μοντέλου (Manual Rotation)

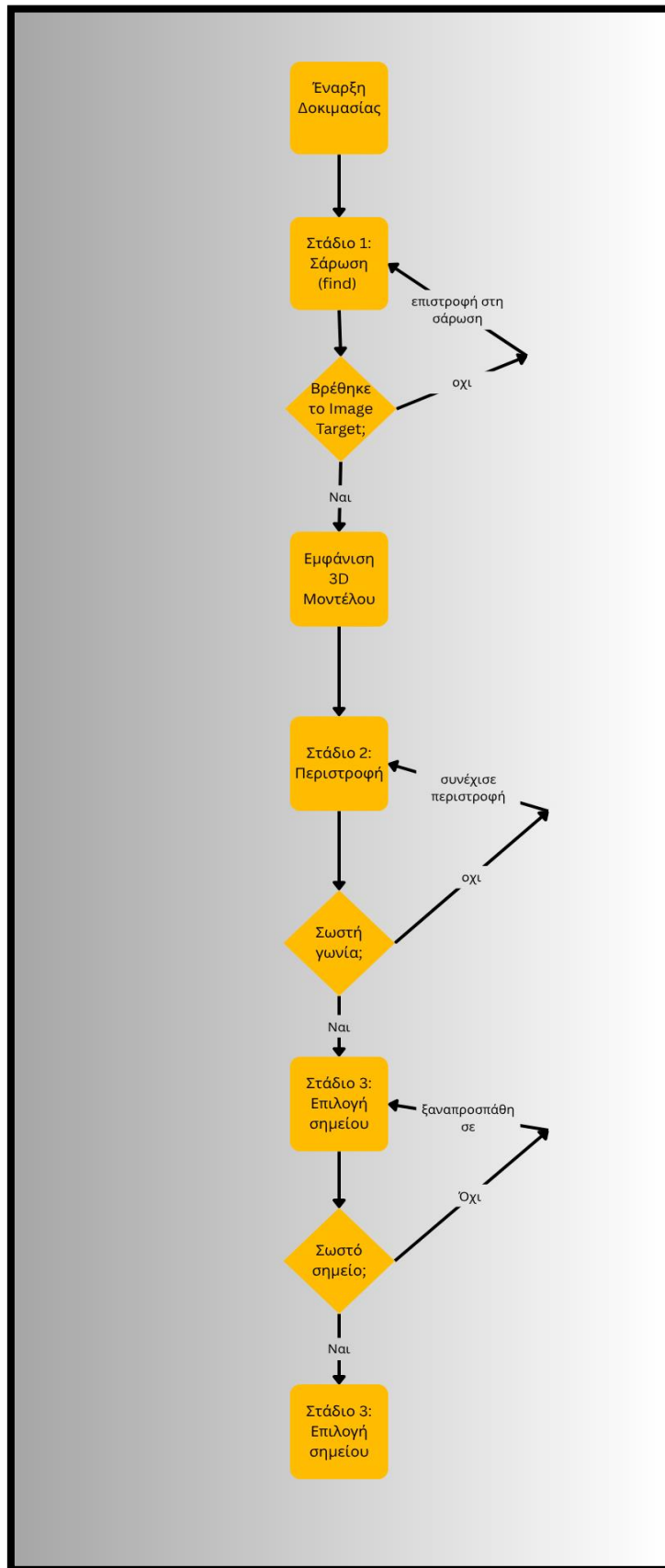
Μόλις το τρισδιάστατο μοντέλο εμφανιστεί πάνω στην εικόνα του βιβλίου, ο μαθητής καλείται να το περιστρέψει στη ζητούμενη γωνία. Ο χειρισμός της περιστροφής γίνεται μέσω κουμπιών κατεύθυνσης (πάνω, κάτω, αριστερά, δεξιά) που εμφανίζονται στην οθόνη. Σε δοκιμασίες όπου `interactionType == "rotation"`, το σύστημα απαιτεί από τον παίκτη να περιστρέψει το μοντέλο σε συγκεκριμένη γωνία. Η μέθοδος `HandleRotationInteraction` υπολογίζει συνεχώς τη διαφορά (Delta Angle) μεταξύ της τρέχουσας γωνίας του μοντέλου και της γωνίας-στόχου (`targetAngle`). Αν η διαφορά είναι μικρότερη από το επιτρεπτό όριο σφάλματος (`angleTolerance`), η δοκιμασία θεωρείται επιτυχής.

### Στάδιο 3: Επιλογή Σημείου μέσω Ακτίνας (Raycast & Click)

Το τελικό και πιο διαδραστικό στάδιο απαιτεί από τον μαθητή να πατήσει (tap) σε ένα συγκεκριμένο σημείο πάνω στο 3D μοντέλο (όπου `interactionType == "click"`). Για την υλοποίηση αυτής της λειτουργίας, γράφτηκε η μέθοδος `HandleClickInteraction`. Όταν το σύστημα καταγράψει ένα άγγιγμα στην οθόνη (`TouchPhase.Began`), δημιουργεί μια νοητή ακτίνα (Ray) που ξεκινά από την κάμερα και κατευθύνεται προς τον τρισδιάστατο χώρο (`ScreenPointToRay`).

Χρησιμοποιώντας τη συνάρτηση `Physics.RaycastAll`, ο κώδικας ελέγχει αν η ακτίνα "χτύπησε" (`intersected`) κάποιο αόρατο όριο (`Collider`) του μοντέλου. Αν το όνομα του αντικείμενου που χτυπήθηκε (`hit.collider.gameObject.name`) ταυτίζεται με το ζητούμενο αντικείμενο (`targetObjectName`), τότε η εφαρμογή επιβεβαιώνει τη σωστή επιλογή. Η κάμερα απενεργοποιείται, οι πόντοι (`bonus points`) προστίθενται στο σκορ του παίκτη και η δοκιμασία ολοκληρώνεται.

Η συνολική ροή της δοκιμασίας AR, με τους ελέγχους που πραγματοποιούνται σε κάθε στάδιο, απεικονίζεται συνοπτικά στο Σχήμα 4.23. Η διαδικασία ξεκινά με τη σάρωση της εικόνας-στόχου και, εφόσον αυτή αναγνωριστεί, εμφανίζεται το τρισδιάστατο μοντέλο. Ο μαθητής περνά διαδοχικά από τα τρία στάδια (`find`, `rotation`, `click`) και σε κάθε στάδιο η εφαρμογή ελέγχει αν πληρούνται οι προϋποθέσεις επιτυχίας πριν προχωρήσει στο επόμενο. Μόνο όταν ολοκληρωθούν και τα τρία στάδια, η δοκιμασία θεωρείται επιτυχής.



Σχήμα 4.23 Διάγραμμα ροής της δοκιμασίας Επαυξημένης Πραγματικότητας, με τα τρία στάδια αλληλεπίδρασης (Σάρωση, Περιστροφή, Επιλογή)

Στο Σχήμα 4.24 παρουσιάζεται η μέθοδος `HandleClickInteraction` του `ARPromptController`, η οποία χρησιμοποιεί τη τεχνική `Raycast` της Unity για να εντοπίσει αν ο μαθητής άγγιξε το σωστό σημείο του 3D μοντέλου.

```

8 private void HandleClickInteraction(ARPrompt prompt)
9 {
10     if (Input.GetMouseButtonDown(0) ||
11         (Input.touchCount > 0 && Input.GetTouch(0).phase == TouchPhase.Began))
12     {
13         Vector3 inputPosition = Input.GetMouseButtonDown(0)
14             ? Input.mousePosition
15             : (Vector3)Input.GetTouch(0).position;
16
17         Ray ray = arCamera.ScreenPointToRay(inputPosition);
18
19         RaycastHit[] hits = Physics.RaycastAll(ray, 3000f, ~0, QueryTriggerInteraction.Collide);
20
21         if (hits.Length == 0)
22         {
23             BoxCollider[] allColliders = FindObjectsOfType<BoxCollider>();
24             foreach (var col in allColliders)
25             {
26                 if (col.gameObject.activeInHierarchy)
27                 {
28                     continue;
29                 }
30             }
31         }
32         else
33         {
34             for (int i = 0; i < hits.Length; i++)
35             {
36                 var hit = hits[i];
37
38                 if (hit.collider.gameObject.name == prompt.targetObjectName ||
39                     (hit.collider.transform.parent != null &&
40                      hit.collider.transform.parent.name == prompt.ta
41                      (field) string ARPrompt.targetObjectName))
42                 {
43                     OnInteractionCompleted();
44                     return;
45                 }
46             }
47         }
48     }
49 }

```

Σχήμα 4.24: Απόσπασμα κώδικα από το `ARPromptController`, που δείχνει τη μέθοδο `HandleClickInteraction` και τη χρήση `Raycast` για την αναγνώριση του αγγίγματος στο 3D μοντέλο

#### 4.21 Μηχανισμός Ομαλής Περιστροφής Τρισδιάστατων Μοντέλων (Smooth Rotation)

Κατά τη διάρκεια των δοκιμασιών AR, ο μαθητής καλείται να περιστρέψει τα τρισδιάστατα μοντέλα που εμφανίζονται πάνω στο σχολικό βιβλίο. Η αρχική υλοποίηση της περιστροφής προκάλεσε απότομες και σπασμωδικές κινήσεις, γεγονός που μείωνε την ποιότητα της εμπειρίας.

Για την επίλυση αυτού του προβλήματος, αναπτύχθηκε ξεχωριστό script εξομάλυνσης (damping). Αντί το μοντέλο να ακολουθεί ακαριαία την εισερχόμενη εντολή, υπολογίζεται ένας ενδιάμεσος στόχος περιστροφής (target angle). Στη συνέχεια, η συνάρτηση `Mathf.SmoothDampAngle` της Unity, που εκτελείται εντός της `Update()`, μεταβάλλει προοδευτικά τη γωνία του μοντέλου προς τον στόχο, αποδίδοντας ομαλή και φυσική κίνηση (Σχήμα 4.25).

```

void Update()
{
    currentYaw = Mathf.SmoothDampAngle(currentYaw, targetYaw, ref yawVelocity, smoothTime);
    currentPitch = Mathf.SmoothDampAngle(currentPitch, targetPitch, ref pitchVelocity, smoothTime);
    currentPosY = Mathf.SmoothDamp(currentPosY, targetPosY, ref posYVelocity, smoothTime);

    Vector3 eulerAngles = transform.localEulerAngles;
    eulerAngles.y = currentYaw;
    eulerAngles.x = currentPitch;
    transform.localEulerAngles = eulerAngles;

    Vector3 position = transform.localPosition;
    position.y = currentPosY;
    transform.localPosition = position;
}

```

Σχήμα 4.25: Απόσπασμα κώδικα που διαχειρίζεται την ομαλή περιστροφή των 3D μοντέλων μέσω της *Mathf.SmoothDampAngle*

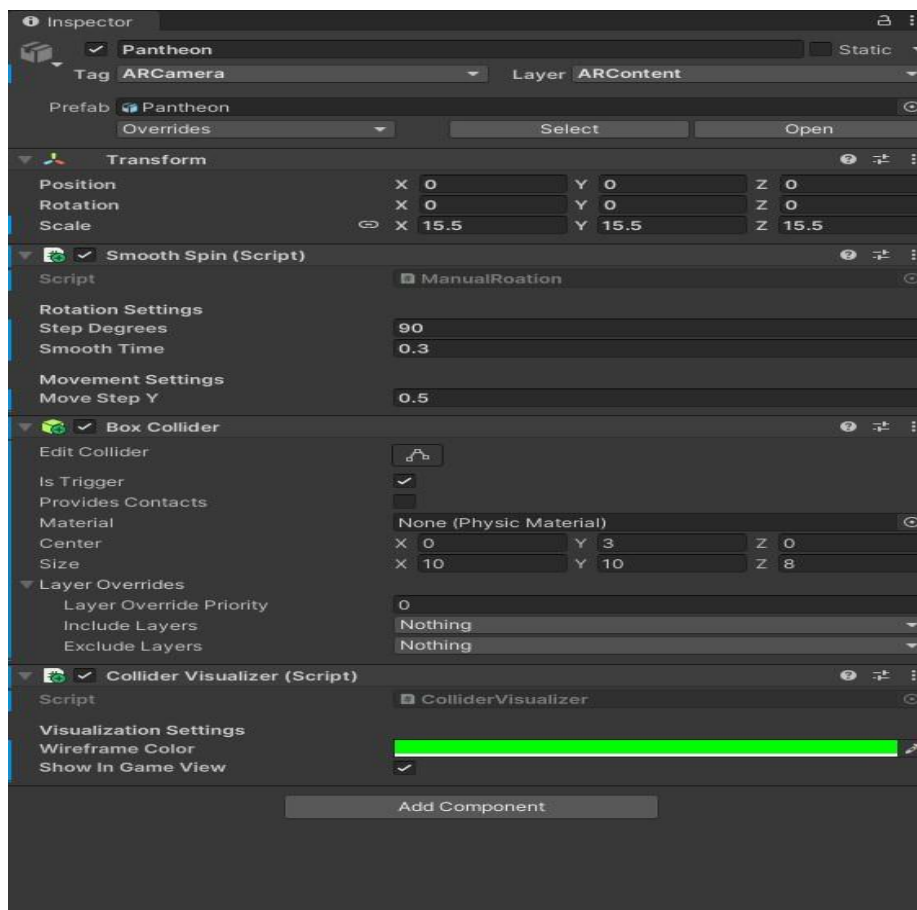
## 4.22 Οργάνωση 3D Μοντέλων στον Unity Editor

Για να λειτουργήσουν σωστά οι μηχανισμοί εμφάνισης, περιστροφής και εντοπισμού στον χώρο της AR, απαιτήθηκε συγκεκριμένη οργάνωση των τρισδιάστατων μοντέλων (π.χ. Πάνθεον, Ωδείο, Κολοσσαίο) μέσα στον Unity Editor. Κάθε μνημείο μετατράπηκε σε Prefab και τοποθετήθηκε ιεραρχικά ως child object κάτω από το αντίστοιχο Image Target της Vuuforia, εξασφαλίζοντας ότι το μοντέλο εμφανίζεται ακριβώς πάνω από την εικόνα του σχολικού βιβλίου κατά τη σάρωση.

Όπως διακρίνεται στην παρακάτω εικόνα, για να αποκτήσει το μοντέλο διαδραστικότητα, προστέθηκαν πάνω του συγκεκριμένα components:

1. Smooth Spin (Script): Το script ελέγχου της περιστροφής. Μέσω του Inspector παραμετροποιήθηκαν οι μεταβλητές *stepDegrees* (γωνία περιστροφής ανά κίνηση) και *smoothTime* (χρόνος εξομάλυνσης), επιτρέποντας τη ρύθμιση της συμπεριφοράς περιστροφής για κάθε μοντέλο.
2. Box Collider: Για να αντιλαμβάνεται το σύστημα τα αγγίγματα του χρήστη κατά τις δοκιμασίες εντοπισμού, δημιουργήθηκε ένα αόρατο φυσικό όριο γύρω από το αντικείμενο. Οι διαστάσεις και το κέντρο του Box Collider προσαρμόστηκαν ώστε να περικλείουν ακριβώς τον όγκο του μνημείου. Με αυτή τη διαμόρφωση, η συνάρτηση *Physics.RaycastAll()* ελέγχει αν η νοητή ακτίνα από την κάμερα τέμνει το collider, και μόνο σε αυτή την περίπτωση επιβεβαιώνεται ο εντοπισμός.

Στο Σχήμα 4.26 παρουσιάζεται το παράθυρο Inspector του Unity Editor για ένα τυπικό Prefab (Pantheon), όπου διακρίνονται τα βασικά components: η ιεραρχία κάτω από το ImageTarget, το script εξομάλυνσης περιστροφής και ο Box Collider που επιτρέπει την ανίχνευση αγγίγματος μέσω Raycast.



Σχήμα 4.26: Προβολή των συστατικών (Inspector) του Prefab "Pantheon". Διακρίνονται η ιεραρχία του κάτω από το ImageTarget, το script εξομάλυνσης περιστροφής και το Box Collider για την ανίχνευση του Raycast

#### 4.23 Εργαλείο Οπτικοποίησης Colliders (ColliderVisualizer)

Κατά τη φάση ανάπτυξης και ελέγχου των δοκιμασιών AR, προέκυψε η ανάγκη για ακριβή οπτικοποίηση των αόρατων ορίων σύγκρουσης (Box Colliders) των τρισδιάστατων μοντέλων. Η προεπιλεγμένη απεικόνιση του Unity Editor δεν ήταν επαρκής για την ακριβή ρύθμισή τους, ιδίως σε μοντέλα με πολύπλοκα πλέγματα (meshes).

Για τον σκοπό αυτό αναπτύχθηκε το βοηθητικό script ColliderVisualizer, το οποίο αξιοποιεί τη συνάρτηση OnDrawGizmos() της Unity — μια μέθοδο που εκτελείται αποκλειστικά εντός του Editor και προορίζεται για οπτικά βοηθήματα κατά την ανάπτυξη. Το script διαβάζει τις διαστάσεις και το κέντρο του εκάστοτε Box Collider και σχεδιάζει ένα ορατό πλαίσιο (WireCube) γύρω από το αντικείμενο (Σχήμα 4.27).

```

public class ColliderVisualizer : MonoBehaviour
{
    [Header("Visualization Settings")]
    public Color wireframeColor = Color.green;
    public bool showInGameView = true;

    private BoxCollider boxCollider;

    void Start()
    {
        boxCollider = GetComponent<BoxCollider>();
        if (boxCollider == null)
        {
            Debug.LogWarning("No BoxCollider found on " + gameObject.name);
        }
    }

    void OnDrawGizmos()
    {
        DrawColliderBox();
    }

    void OnDrawGizmosSelected()
    {
        DrawColliderBox();
    }

    void DrawColliderBox()
    {
        BoxCollider bc = GetComponent<BoxCollider>();
        if (bc == null) return;

        Gizmos.color = wireframeColor;

        Matrix4x4 oldMatrix = Gizmos.matrix;
        Gizmos.matrix = transform.localToWorldMatrix;

        Gizmos.DrawWireCube(bc.center, bc.size);
    }

    void DrawColliderBox()
    {
        BoxCollider bc = GetComponent<BoxCollider>();
        if (bc == null) return;

        Gizmos.color = wireframeColor;

        Matrix4x4 oldMatrix = Gizmos.matrix;
        Gizmos.matrix = transform.localToWorldMatrix;

        Gizmos.DrawWireCube(bc.center, bc.size);

        Gizmos.matrix = oldMatrix;
    }

    void OnRenderObject()
    {
        if (!showInGameView) return;
    }
}

```

Σχήμα 4.27: Ο κώδικας του βοηθητικού εργαλείου *ColliderVisualizer.cs*, το οποίο σχεδιάζει τα αόρατα όρια σύγκρουσης στο περιβάλλον ανάπτυξης

#### 4.24 Επίλογος Κεφαλαίου

Στο κεφάλαιο αυτό αναλύθηκε η προγραμματιστική υλοποίηση της εφαρμογής. Αρχικά παρουσιάστηκε η επιλογή να αποθηκευτούν τα δεδομένα του παιχνιδιού (ερωτήσεις, μαθήματα, χρονολογίες) σε αρχεία JSON αντί να ενσωματωθούν στον κώδικα C#. Στη συνέχεια αναπτύχθηκε η λογική του κουίζ με τους μηχανισμούς χρόνου και ζωών, η ενσωμάτωση της κάμερας Vuuforia για τις δοκιμασίες AR, καθώς και τα βοηθητικά εργαλεία ανάπτυξης, όπως ο ColliderVisualizer για την οπτικοποίηση των ορίων σύγκρουσης. Κοινός άξονας σε όλες τις αποφάσεις υπήρξε η αρχή του διαχωρισμού των αρμοδιοτήτων, δηλαδή κάθε υποσύστημα σχεδιάστηκε ώστε να λειτουργεί ανεξάρτητα, διευκολύνοντας μελλοντικές προσθήκες, όπως ένα νέο επίπεδο ή ένα νέο τρισδιάστατο μοντέλο μνημείου.

Η υιοθέτηση modular αρχιτεκτονικής με διακριτά scripts για κάθε λειτουργία δεν εξυπηρετεί μόνο την ορθή λειτουργία της εφαρμογής κατά την παρούσα φάση, αλλά διευκολύνει σημαντικά τη μελλοντική συντήρηση και επέκτασή της. Τα συμπεράσματα που προκύπτουν από την ανάπτυξη αυτή, καθώς και οι προτάσεις για μελλοντικές επεκτάσεις, παρουσιάζονται στο επόμενο κεφάλαιο.



## Κεφάλαιο 5ο: Συμπεράσματα και Μελλοντικές Επεκτάσεις

### 5.1 Εισαγωγή Κεφαλαίου

Στο παρόν, τελευταίο κεφάλαιο της πτυχιακής εργασίας, επιχειρείται η συνολική αποτίμηση του έργου που αναπτύχθηκε. Εξετάζεται κατά πόσο οι αρχικοί στόχοι του έργου επιτεύχθηκαν μέσα από την ενσωμάτωση AR και gamification. Επιπλέον, αναλύονται οι σημαντικότερες τεχνικές και σχεδιαστικές προκλήσεις που προέκυψαν κατά τη διάρκεια της υλοποίησης στο περιβάλλον της μηχανής Unity 3D, καθώς και οι λύσεις που εφαρμόστηκαν. Τέλος, προτείνονται ρεαλιστικές κατευθύνσεις για τη μελλοντική επέκταση και βελτίωση του λογισμικού, με στόχο την περαιτέρω αξιοποίησή του σε πραγματικές συνθήκες σχολικής τάξης.

### 5.2 Σύνοψη του Έργου

Η παρούσα πτυχιακή εργασία επικεντρώθηκε στον σχεδιασμό και την ανάπτυξη μιας εκπαιδευτικής εφαρμογής Επαυξημένης Πραγματικότητας (AR), με στόχο τον εκσυγχρονισμό της διδασκαλίας της Ιστορίας για μαθητές της Ε΄ Δημοτικού. Μέσα από την ενσωμάτωση μηχανισμών παιχνιδοποίησης (gamification) και τη χρήση του σχολικού βιβλίου ως εκκινήτη της AR εμπειρίας, η τεχνολογία λειτουργεί ως συμπλήρωμα της παραδοσιακής διδασκαλίας, χωρίς να την αντικαθιστά.

Η υλοποίηση κατέληξε σε ένα πλήρως λειτουργικό πρωτότυπο για συσκευές Android. Το πρωτότυπο καλύπτει θεωρία, αξιολόγηση με χρονόμετρο και σκορ, και αλληλεπίδραση με 3D μοντέλα στον χώρο του μαθητή. Το σύστημα αναλυτικής ανατροφοδότησης, που ενημερώνει τον μαθητή για τα λάθη του στο τέλος κάθε επιπέδου, ενισχύει την ενεργό συμμετοχή στη μαθησιακή διαδικασία. Σε επίπεδο τεχνικής υλοποίησης, η εφαρμογή επιτυγχάνει τους αρχικούς της στόχους· η συστηματική αξιολόγηση από πραγματικούς μαθητές αποτελεί επόμενο βήμα και αναφέρεται στις μελλοντικές επεκτάσεις.

### 5.3 Προκλήσεις κατά την Ανάπτυξη

Η ανάπτυξη της εφαρμογής στο περιβάλλον Unity σε συνδυασμό με το Vuforia Engine παρουσίασε αρκετές τεχνικές προκλήσεις, οι οποίες αντιμετωπίστηκαν μέσω επαναληπτικού σχεδιασμού. Ιδιαίτερη προσοχή απαιτήθηκε στον σχεδιασμό της διεπαφής χρήστη. Ο στόχος ήταν να δημιουργηθεί ένα περιβάλλον οπτικά ελκυστικό και εύκολο στη χρήση για μαθητές ηλικίας 10-11 ετών, διατηρώντας παράλληλα την εκπαιδευτική του αξία χωρίς να καταλήξει σε υπερβολικά απλοποιημένο ή γνωστικά επιβαρυντικό αποτέλεσμα. Η ισορροπία μεταξύ παιγνιώδους αισθητικής και παιδαγωγικής σαφήνειας αποτέλεσε μια από τις βασικές σχεδιαστικές προκλήσεις του έργου.

#### 5.4 Μελλοντικές Επεκτάσεις

Η εφαρμογή μπορεί να επεκταθεί σε διάφορες κατευθύνσεις. Οι κυριότερες προτάσεις διαρθρώνονται ως εξής:

1. **Ανάλυση Δεδομένων Μάθησης:** Ως άμεσο επόμενο βήμα, εξετάζεται η εξαγωγή των αποτελεσμάτων κάθε μαθητή σε μορφή PDF, παρέχοντας στον εκπαιδευτικό σαφή εικόνα της επίδοσης της τάξης. Μακροπρόθεσμα, η ενσωμάτωση απομακρυσμένης βάσης δεδομένων (π.χ. Firebase) θα επέτρεπε την παρακολούθηση της προόδου σε πραγματικό χρόνο, με εκτενέστερα στατιστικά όπως ποσοστά επιτυχίας ανά ερώτηση και μέσος χρόνος απόκρισης.
2. **Εμπλουτισμός Παιγνιδοποίησης:** Προσθήκη συστήματος επιτευγμάτων και πινάκων κατάταξης σε επίπεδο σχολικής τάξης, ώστε να ενισχυθεί η ευγενής άμιλλα μεταξύ των μαθητών.
3. **Υποστήριξη Πολλαπλών Πλατφορμών:** Επέκταση της εφαρμογής σε iOS μέσω του ARKit, διευρύνοντας τον αριθμό των συμβατών συσκευών.
4. **Αξιολόγηση από Πραγματικούς Χρήστες:** Ως κρίσιμο επόμενο βήμα, αξιολόγηση της εφαρμογής σε πραγματική σχολική τάξη με μαθητές της Ε΄ Δημοτικού, ώστε να μετρηθούν τα μαθησιακά αποτελέσματα και να συλλεχθεί ποιοτική ανατροφοδότηση από εκπαιδευτικούς και μαθητές.

#### 5.5 Επίλογος Κεφαλαίου

Με αυτό ολοκληρώνεται η παρουσίαση της εφαρμογής και των κατευθύνσεων που μπορεί να ακολουθήσει στο μέλλον. Οι προτεινόμενες επεκτάσεις δείχνουν ότι η εφαρμογή έχει τη δυνατότητα να εξελιχθεί περαιτέρω, προσαρμοζόμενη στις πραγματικές ανάγκες της σχολικής τάξης.

**ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [1] P. Milgram, H. Takemura, A. Utsumi, F. Kishino, «Augmented reality: A class of displays on the reality-virtuality continuum», *Telem manipulator and Telepresence Technologies*, τ. 2351, σ. 282-292, 1994. [Online]. Διαθέσιμο στο: <https://doi.org/10.1117/12.197321>.
- [2] Design-Based Research Collective, «Design-based research: An emerging paradigm for educational inquiry», *Educational Researcher*, τ. 32, τχ. 1, σ. 5-8, 2003. [Online]. Διαθέσιμο στο: <https://doi.org/10.3102/0013189X032001005>.
- [3] I. E. Sutherland, «A head-mounted three dimensional display», *Proceedings of the Fall Joint Computer Conference*, σ. 757-764, 1968.
- [4] T. P. Caudell, D. W. Mizell, «Augmented reality: An application of heads-up display technology to manual manufacturing processes», *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, σ. 659-669, 1992.
- [5] H. Kato, M. Billinghurst, «Marker tracking and HMD calibration for a video-based augmented reality conferencing system», *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, σ. 85-94, 1999.
- [6] M. Akçayır, G. Akçayır, «Advantages and challenges associated with augmented reality for education: A systematic review», *Educational Research Review*, τ. 20, σ. 1-11, 2017. [Online]. Διαθέσιμο στο: <https://doi.org/10.1016/j.edurev.2016.11.002>.
- [7] J. Bacca, S. Baldiris, R. Fabregat, S. Graf, Kinshuk, «Augmented reality trends in education: A systematic review», *Educational Technology & Society*, τ. 17, τχ. 4, σ. 133-149, 2014. [Online]. Διαθέσιμο στο: <https://www.jstor.org/stable/jeductechsoci.17.4.133>.
- [8] J. Garzón, J. Pavón, S. Baldiris, «Systematic review and meta-analysis of augmented reality in educational settings», *Virtual Reality*, τ. 23, τχ. 4, σ. 447-459, 2019. [Online]. Διαθέσιμο στο: <https://doi.org/10.1007/s10055-019-00379-9>.
- [9] K. M. Kapp, *The Gamification of Learning and Instruction*. San Francisco: John Wiley & Sons, 2012.
- [10] J. L. Plass, B. D. Homer, C. K. Kinzer, «Foundations of game-based learning», *Educational Psychologist*, τ. 50, τχ. 4, σ. 258-283, 2015. [Online]. Διαθέσιμο στο: <https://doi.org/10.1080/00461520.2015.1122533>.
- [11] E. L. Deci, R. M. Ryan, «The what and why of goal pursuits: Human needs and the self-determination of behavior», *Psychological Inquiry*, τ. 11, τχ. 4, σ. 227-268, 2000. [Online]. Διαθέσιμο στο: [https://doi.org/10.1207/S15327965PLI1104\\_01](https://doi.org/10.1207/S15327965PLI1104_01).

[12] Unity Technologies, «Unity User Manual» [Online]. Διαθέσιμο στο: <https://docs.unity3d.com/Manual/index.html>. **[Προσπελάστηκε: 01/05/2026]**.

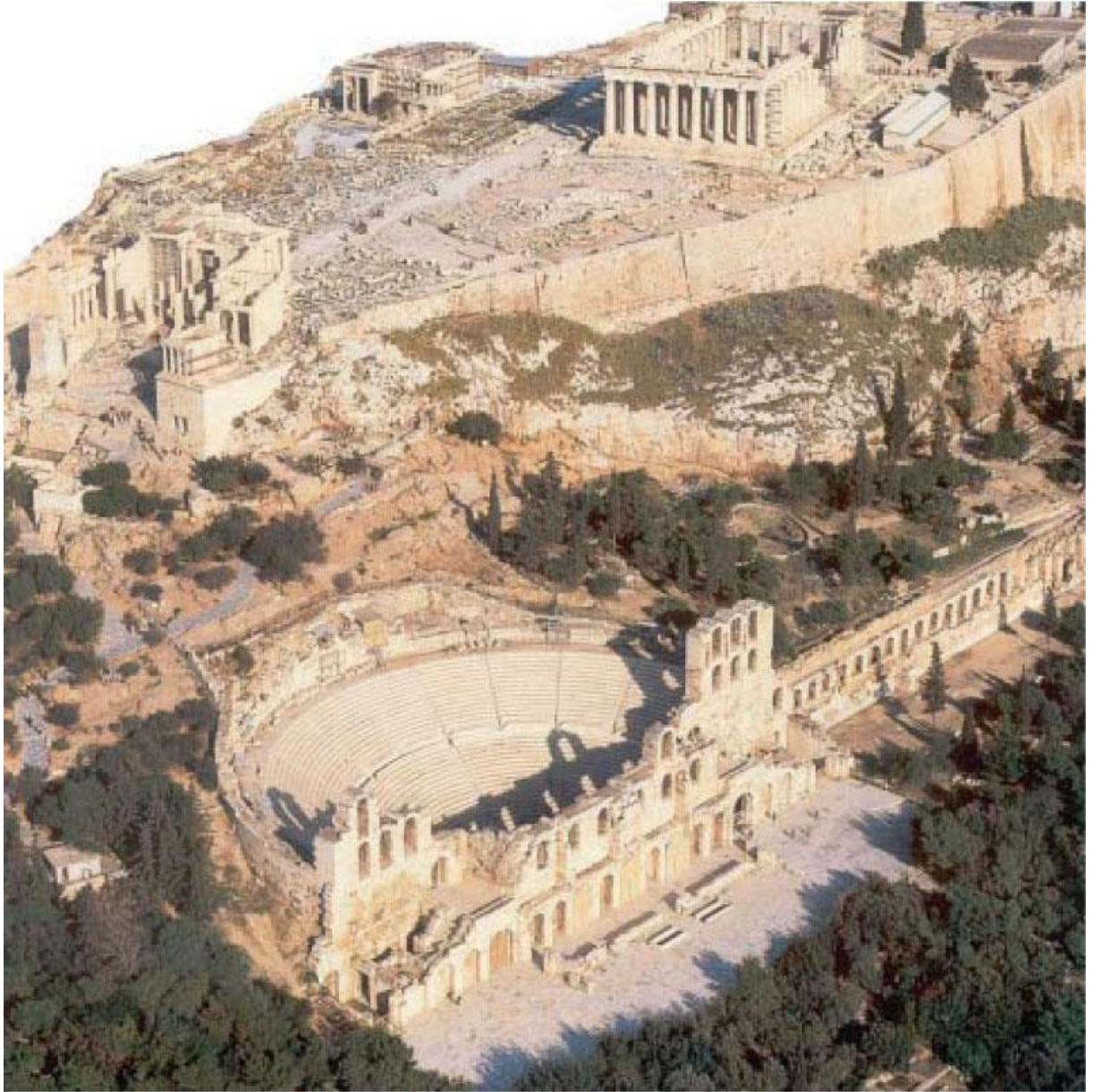
[13] PTC Inc., «Vuforia Engine Developer Portal» [Online]. Διαθέσιμο στο: <https://developer.vuforia.com>. **[Προσπελάστηκε: 01/05/2026]**.

[14] Android Developers, «Android Studio and OS Documentation» [Online]. Διαθέσιμο στο: <https://developer.android.com>. **[Προσπελάστηκε: 01/05/2026]**.

[15] Microsoft, «C# Programming Guide» [Online]. Διαθέσιμο στο: <https://learn.microsoft.com/en-us/dotnet/csharp>. **[Προσπελάστηκε: 01/05/2026]**.

## ΠΑΡΑΡΤΗΜΑ Α: ΕΙΚΟΝΕΣ-ΣΤΟΧΟΙ (IMAGE TARGETS)

Στο παρόν παράρτημα παρουσιάζονται οι εικόνες-στόχοι από το σχολικό βιβλίο Ιστορίας Ε΄ Δημοτικού που χρησιμοποιήθηκαν ως Image Targets της εφαρμογής. Κάθε εικόνα ενεργοποιεί την εμφάνιση του αντίστοιχου τρισδιάστατου μοντέλου.



*Εικόνα 1: Image Target — Ωδείο Ηρώδου του Αττικού (Επίπεδο 1)*



*Εικόνα 2 : Image Target — Πάνθεον της Ρώμης (Επίπεδο 2)*



*Εικόνα 3 : Image Target — Κολοσσαίο (Επίπεδο 3)*