

Ατομική Διπλωματική Εργασία

**ΜΕΛΕΤΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΟΥ ΣΤΟΧΕΥΟΥΝ ΣΤΗΝ ΣΑΡΩΣΗ
ΚΑΙ ΑΝΑΓΝΩΡΙΣΗ ΕΙΚΟΝΑΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΧΕΤΙΚΗΣ
ΕΦΑΡΜΟΓΗΣ**

Γεωργία Φουκαρίδου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ιανουάριος 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Μελέτη τεχνολογιών που στοχεύουν στην σάρωση και αναγνώριση εικόνας και
υλοποίηση σχετικής εφαρμογής**

Γεωργία Φουκαρίδου

Επιβλέπων Καθηγητής

Παπαδόπουλος Γιώργος

Συνεπιβλέποντες

Χρίστος Μεττούρης

Ευαγγελία Βανέζη

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

Ιανουάριος 2021

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή του Τμήματος Πληροφορικής Δρ. Γιώργο Παπαδόπουλο που μου έδωσε την ευκαιρία να εμπλουτίσω τις γνώσεις μου σε καινούριους τομείς, όπως αυτόν της Επαυξημένης Πραγματικότητας και της ανάπτυξης συστημάτων μέσω του συστήματος Unity.

Ακόμη, θα ήθελα να ευχαριστήσω θερμά τους κ. Χρίστο Μεττούρη και κα. Ευαγγελία Βανέζη, οι οποίοι προσέφεραν τις συμβουλές και την καθοδήγηση τους καθ' όλη τη διάρκεια της ανάπτυξης της εργασίας. Ο χρόνος και η προσοχή που αφιέρωσαν αποτέλεσαν πολύτιμη βοήθεια για την ολοκλήρωση της εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την υποστήριξη τους κατά τη διάρκεια των σπουδών μου.

Περίληψη

Η διπλωματική έχει σκοπό την μελέτη τεχνολογιών που αφορούν σάρωση και ανίχνευση αντικειμένων μέσα σε μια εικόνα μέσω κινητής συσκευής, καθώς και την υλοποίηση εφαρμογής η οποία θα έχει τις εξής χρήσεις:

1. Εντοπισμός αντικειμένων σε μία εικόνα
2. Εμφάνιση πληροφοριών στην οθόνη του χρήστη κατά τη διάρκεια της σάρωσης αντικειμένων
3. Κωδικοποίηση μηνυμάτων με αλφάβητο που αποτελείται από σύμβολα

Η εφαρμογή χρησιμοποιεί *Επαυξημένη Πραγματικότητα* για να αποκωδικοποιεί σύμβολα που βρίσκονται σε κάποια εικόνα σε γράμματα, εμφανίζοντας έτσι κάποια ιστορία. Επιπλέον, κωδικοποιεί γράμματα στα αντίστοιχα σύμβολα ώστε να κωδικοποιείται μια ιστορία/φράση, καθώς και υπάρχει η δυνατότητα αποθήκευσης και κοινοποίησης της κωδικοποιημένης φράσης.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Εισαγωγή	1
	1.2 Στόχος	2
	1.3 Μεθοδολογία	2
	1.4 Δομή	3
Κεφάλαιο 2	Υπόβαθρο Θεωρίας και Ανασκόπηση Βιβλιογραφίας.....	5
	2.1 Εισαγωγή	5
	2.2 Κινητή Επαυξημένη Πραγματικότητα (Mobile Augmented Reality)	6
	2.2.1 Επαυξημένη Πραγματικότητα βάσει Δείκτη (Marker-Based Augmented Reality Systems)	7
	2.2.2 Επαυξημένη Πραγματικότητα χωρίς Δείκτη (Markerless / Location-Based / Geo Based Augmented Reality Systems)	8
	2.3 Κινητή Οπτική Αναζήτηση (Mobile Visual Search)	8
	2.4 Αναγνώριση Αντικειμένου (Object Recognition)	9
	2.5 Προσαρμογή Προτύπου (Template Matching)	11
Κεφάλαιο 3	Σχεδίαση Εφαρμογής.....	12
	3.1 Εισαγωγή	12
	3.2 Σχεδίαση Εφαρμογής	12
	3.2.1 Υλοποίηση με Κινητή Επαυξημένη Πραγματικότητα	13
	3.2.2 Υλοποίηση με Κινητή Οπτική Αναζήτηση	14
	3.2.3 Υλοποίηση με Αναγνώριση Αντικειμένου	15
	3.2.4 Υλοποίηση με Προσαρμογή Προτύπου	16
	3.2.5 Τελική Σχεδίαση Εφαρμογής	16
	3.3 Αρχιτεκτονική Εφαρμογής	17

Κεφάλαιο 4	Τελική Υλοποίηση Εφαρμογής.....	20
4.1	Εισαγωγή	20
4.2	Λειτουργίες Εφαρμογής	21
4.2.1	Αποκωδικοποίηση Ιστορίας	21
4.2.2	Κωδικοποίηση Ιστορίας	22
4.2.3	Οδηγίες Χρήσης	24
4.2.4	Προβολή Ιστορίας	24
4.2.5	Γκαλερί	27
4.2.6	Άλλες Λειτουργίες	29
4.3	Φραγμοί Εφαρμογής	29
4.4	Υλοποίηση Λειτουργιών	30
4.4.1	Vuforia AR Camera	30
4.4.2	Αποθήκευση τελευταίας ιστορίας	32
4.4.3	Βάση Δεδομένων	33
Κεφάλαιο 5	Αξιολόγηση Εφαρμογής.....	37
5.1	Εισαγωγή	37
5.2	Διαδικασία Αξιολόγησης	37
5.3	Ερωτήσεις Ερωτηματολογίου	38
5.4	Αποτελέσματα Ερωτηματολογίου	39
Κεφάλαιο 6	Συμπεράσματα	43
6.1	Εισαγωγή	43
6.2	Τελικά Συμπεράσματα	43
6.3	Μελλοντική Εργασία	44
Βιβλιογραφία		46
Παράρτημα Α.....		A-1
Παράρτημα Β.....		B-1

Π α ρ ά ρ τ η μ α Γ.....Γ-1

Π α ρ ά ρ τ η μ α Δ..... Δ-1

Π α ρ ά ρ τ η μ α Ε..... Ε-1

Π α ρ ά ρ τ η μ α ΣΤ..... ΣΤ-1

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή	1
1.2 Στόχος	2
1.3 Μεθοδολογία	2
1.4 Δομή	3

1.1 Εισαγωγή

Στα πλαίσια μιας διεπιστημονικής συνεργασίας μεταξύ του τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου και του τμήματος Αρχιτεκτονικής (υπεύθυνος καθηγητής κ. Σωκράτης Στρατής), υλοποιήθηκε μια εφαρμογή κινητής συσκευής λειτουργικού συστήματος Android, με θέμα την αναγνώριση εικόνων μέσω κινητής συσκευής. Κίνητρο αποτέλεσε το έργο του κ. Στρατή «The “Recollecting Collective Memory” Project» όπου, με βάση αυτό, μελετήθηκαν οι διάφοροι πιθανοί τρόποι που θα μπορούσαν να αποτελέσουν την υλοποίηση της σχετικής εφαρμογής.

Το έργο «The “Recollecting Collective Memory” Project» στοχεύει στην αποκωδικοποίηση χρωματιστών γεωμετρικών σχημάτων που θα βρίσκονται στο πάνω μέρος κυβόλιθων από μπετόν, τα οποία στο κάτω μέρος θα έχουν κρυμμένα στο χρώμα τα αντίστοιχα γράμματα, καθώς και την κωδικοποίηση φράσεων που θέλει ο χρήστης να στέλνει μέσω του τηλεφώνου του σε φίλους του σε εικόνες με γεωμετρικά σχήματα. Στόχο του έργου αποτελεί η ενθάρρυνση χρηστών στο δημόσιο χώρο να εμπλακούν σε μια διαδικασία αποκωδικοποίησης «θαμμένων» ιστοριών σχετικών με τη συλλογική μνήμη μιας κοινότητας.

Η παρούσα εργασία, βασισμένη στο έργο [12], στοχεύει στην υλοποίηση εφαρμογής η οποία, όπως στο έργο [12], αναγνωρίζει εικόνες που υπάρχουν στο φυσικό περιβάλλον και τις κωδικοποιεί σε γράμματα. Η εφαρμογή αυτή μπορεί να χρησιμοποιηθεί για διασκέδαση μεταξύ φίλων, ή ακόμα και την ασφαλή ανταλλαγή μηνυμάτων, αφού είναι κωδικοποιημένα. Οι χρήστες της μπορούν να στείλουν κωδικοποιημένα μηνύματα σε άλλους, οι οποίοι, έχοντας κατεβασμένη την εφαρμογή, θα είναι σε θέση να τα αποκωδικοποιήσουν και να διαβάσουν το μήνυμα που τους εστάλη.

1.2 Στόχος

Στόχο για αυτή τη διπλωματική αποτελεί η μελέτη των τεχνολογιών που μπορούν να χρησιμοποιηθούν για την σάρωση και αναγνώριση εικόνας από κινητή συσκευή. Αφού γίνει η μελέτη των τεχνολογιών, η διπλωματική εργασία στοχεύει στην υλοποίηση σχετικής εφαρμογής κινητής συσκευής. Η εφαρμογή αυτή αναμένεται να έχει τις λειτουργίες της αποκωδικοποίησης κάποιου κωδικοποιημένου μηνύματος, το οποίο θα αναγνωριστεί από μια εικόνα που θα σαρωθεί, της κωδικοποίησης κάποιου καινούργιου μηνύματος δημιουργημένο από τον χρήστη, της μετατροπής του κωδικοποιημένου μηνύματος σε εικόνα, καθώς και της κοινοποίησης κάποιας εικόνας με κωδικοποιημένη ιστορία. Χρησιμοποιώντας την εφαρμογή, οι χρήστες θα είναι σε θέση να επικοινωνήσουν με άλλους μέσω κωδικοποιημένων μηνυμάτων, γεγονός το οποίο συμβάλει στην διασκέδαση των χρηστών.

1.3 Μεθοδολογία

Η μεθοδολογία που ακολουθήθηκε αποτελείται από έξι βήματα:

A. Έρευνα και μελέτη διαφόρων τεχνικών αναγνώρισης εικόνας/αντικειμένου σε εικόνα (Mobile Augmented Reality, Mobile Visual Search, Object Recognition, Template Matching). Σε αυτό το στάδιο έγινε μελέτη επιστημονικών άρθρων για την κατανόηση της λειτουργίας της κάθε τεχνικής αναγνώρισης εικόνας-αντικειμένου.

Β. Μελέτη διαφόρων μεθόδων υλοποίησης για την κάθε τεχνική αναγνώρισης εικόνας-αντικειμένου σε εικόνα. Σε αυτό το στάδιο έγινε έρευνα σχετικά με τον τρόπο που θα μπορούσε να χρησιμοποιηθεί η κάθε τεχνική που μελετήθηκε για να εξυπηρετήσει τις ανάγκες της παρούσας εργασίας.

Γ. Μελέτη διαθέσιμων εργαλείων και βιβλιοθηκών λογισμικού για την κάθε τεχνική. Σε αυτό το στάδιο έγινε μελέτη για εύρεση εργαλείων τα οποία θα καθιστούσαν την υλοποίηση μιας σχετικής εφαρμογής δυνατή. Η διαδικασία έγινε για κάθε τεχνολογία που μελετήθηκε στο αρχικό βήμα της μεθοδολογίας.

Δ. Απόφαση σχετικά με το ποια τεχνική είναι η βέλτιστη για τους σκοπούς της εργασίας και απόφαση σχετικά με το ποια τεχνική θα ακολουθηθεί.

Ε. Σχεδίαση και Υλοποίηση της εφαρμογής

ΣΤ. Αξιολόγηση της εφαρμογής μέσω online ερωτηματολογίου

1.4 Δομή

Η εργασία αυτή αποτελείται από έξι κεφάλαια. Το Κεφάλαιο 1 αποτελεί την εισαγωγή, όπου περιγράφεται το κίνητρο της εργασίας, καθώς και ο στόχος και η μεθοδολογία που ακολουθήθηκε γι' αυτήν την εργασία. Στο Κεφάλαιο 2 παρουσιάζεται η απαιτούμενη γνώση που χρειάζεται για την βέλτιστη κατανόηση της εργασίας, καθώς και η έρευνα που έγινε σχετικά με τις τεχνολογίες που ασχολούνται με την αναγνώριση εικόνας-αντικειμένου σε εικόνα. Στο Κεφάλαιο 3 φαίνονται η αρχιτεκτονική και η σχεδίαση του συστήματος, καθώς και παρουσιάζονται οι τρόποι με τους οποίους μπορεί να υλοποιηθεί η εφαρμογή με βάση τους τρόπους αναγνώρισης εικόνας που παρουσιάζονται στο Κεφάλαιο 2. Επίσης, παρουσιάζονται τα διάφορα εργαλεία και βιβλιοθήκες που είναι διαθέσιμα για την υλοποίηση της εφαρμογής με την κάθε τεχνική. Στο Κεφάλαιο 4 περιγράφεται η εφαρμογή, οι διάφορες λειτουργίες που είναι διαθέσιμες μέσω αυτής, καθώς και τα προβλήματα και σφάλματα που συναντήθηκαν κατά την υλοποίηση της. Ακολουθεί το Κεφάλαιο 5 όπου παρουσιάζεται η φάση της αξιολόγησης της εφαρμογής που αναπτύχθηκε, που έγινε μέσω διαδικτυακών

ερωτηματολογίων από χρήστες. Τέλος στο Κεφάλαιο 6 καταγράφονται τα τελικά συμπεράσματα αυτής της εργασίας, μελλοντική εργασία και περιορισμοί που έχουμε. Ακολουθούν η Βιβλιογραφία και τα Παραρτήματα.

Κεφάλαιο 2

Υπόβαθρο Θεωρίας και Ανασκόπηση Βιβλιογραφίας

2.1 Εισαγωγή	5
2.2 Κινητή Επαυξημένη Πραγματικότητα (Mobile Augmented Reality)	6
2.2.1 Επαυξημένη Πραγματικότητα βάσει Δείκτη (Marker-Based Augmented Reality Systems)	7
2.2.2 Επαυξημένη Πραγματικότητα χωρίς Δείκτη (Markerless / Location-Based / Geo Based Augmented Reality Systems)	8
2.3 Κινητή Οπτική Αναζήτηση (Mobile Visual Search)	8
2.4 Αναγνώριση Αντικειμένου (Object Recognition)	9
2.5 Προσαρμογή Προτύπου (Template Matching)	11

2.1 Εισαγωγή

Μια εφαρμογή που έχει ως λειτουργία την αποκωδικοποίηση συμβόλων που έχουν τοποθετηθεί στον πραγματικό κόσμο, προϋποθέτει κάποιου είδους σύνδεση του φυσικού και του ψηφιακού κόσμου. Δηλαδή, τα σύμβολα τα οποία αποκωδικοποιούνται δεν προβάλλονται από την ίδια συσκευή, αλλά είτε βρίσκονται στο φυσικό περιβάλλον (εκτυπωμένα σε χαρτί) είτε προβάλλονται μέσω κάποιας δεύτερης ηλεκτρονικής συσκευής (κινητή συσκευή, υπολογιστής) που βρίσκεται στο περιβάλλον του χρήστη. Επομένως, η αναγκαία αναγνώριση στοιχείων από την εικόνα που «σάρωσε» ο χρήστης με την συσκευή του, καθιστά την ανίχνευση και επεξεργασία της εικόνας το σημαντικότερο μέρος της διπλωματικής εργασίας αυτής.

Η ανίχνευση εικόνας μπορεί να γίνει με πολλούς τρόπους.

2.2 Κινητή Επαυξημένη Πραγματικότητα (Mobile Augmented Reality)

Η κινητή Επαυξημένη Πραγματικότητα (Mobile Augmented Reality) παράγεται μέσω των κινητών συσκευών τις οποίες μπορεί να μεταφέρει ο χρήστης. Ο συνδυασμός της κάμερας με/ή το σύστημα GPS ενός κινητού τηλεφώνου επιτρέπουν την προβολή επιπλέον πληροφοριών για ένα γεωγραφικό σημείο, διαμορφώνοντας ένα επαυξημένο πληροφοριακό τελικό αποτέλεσμα [19]. Ο εξοπλισμός που χρειάζεται για την χρήση μιας εφαρμογής για επαυξημένη πραγματικότητα, μεταφέρονται οπουδήποτε μαζί με το άτομο [9].

Η Επαυξημένη Πραγματικότητα (Augmented Reality) συνδέεται με την έννοια της Εικονικής Πραγματικότητας (Virtual Reality). Ενώ η εικονική πραγματικότητα στοχεύει στο να δημιουργήσει έναν τεχνητό κόσμο ο οποίος μπορεί να ανακαλυφθεί μέσω όρασης αλλά και ακοής, η επαυξημένη πραγματικότητα στοχεύει στην προβολή πληροφοριών αλλά και εικονικών προσώπων ή χώρων σχεδιασμένων μέσα έναν ηλεκτρονικό υπολογιστή, παρά στη δημιουργία ενός εντελώς καινούριου κόσμου [14].

Η εικονική, δηλαδή, πραγματικότητα είναι η προσομοίωση ενός περιβάλλοντος – πραγματικού ή φανταστικού– από έναν υπολογιστή. Αντίθετα, η επαυξημένη πραγματικότητα είναι η -σε πραγματικό χρόνο- θέαση ενός πραγματικού περιβάλλοντος, του οποίου στοιχεία (π.χ. ήχος, βίντεο ή γραφικά) επαυξάνονται από στοιχεία που παρήχθησαν από συσκευές υπολογιστών.

Ένα σύστημα επαυξημένης πραγματικότητας αποτελεί έναν συνδυασμό πραγματικών και παραγόμενων από υπολογιστή πληροφοριών, ενώνοντας τεχνητά και φυσικά αντικείμενα. Η κινητή επαυξημένη πραγματικότητα απαιτεί τον συνδυασμό αρκετών τεχνολογιών για να γίνει εφικτή η έννοια. Ένα σύστημα κινητής επαυξημένης πραγματικότητας είναι σχεδιασμένο αποκλειστικά για κινητές πλατφόρμες (π.χ. smartphone ή tablet) οι οποίες ανιχνεύουν αντικείμενα που αναγνωρίζονται από τη βάση δεδομένων [1].

Υπάρχουν 2 κατηγορίες Συστημάτων Επαυξημένης Πραγματικότητας, ως εξής:

2.2.1 Επαυξημένη Πραγματικότητα βάσει Δείκτη (Marker-Based Augmented Reality Systems)

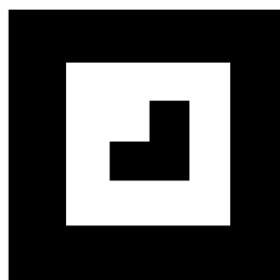
Στόχος της marker-based επαυξημένης πραγματικότητας είναι ο εντοπισμός δεικτών (markers) στον φυσικό κόσμο. Δηλαδή, χρησιμοποιώντας μία απλή κάμερα, μπορούμε να εντοπίσουμε ένα αντικείμενο -τον marker- , και με τη βοήθεια του κατάλληλου λογισμικού να εντοπίσουμε την θέση του marker στον πραγματικό κόσμο, σε σχέση με την θέση της κάμερας, έτσι ώστε να τοποθετηθεί εκεί κάποιο ψηφιακό στοιχείο.

Για παράδειγμα, στο Σχήμα 2.1 φαίνεται η σάρωση ενός marker με ένα κινητό τηλέφωνο, και στην οθόνη του εμφανίζεται ένα 3D αντικείμενο, το σπίτι.



Σχήμα 2.1: Παράδειγμα Marker Based AR

Τα περισσότερα markers είναι ασπρόμαυρα, παρόλο που η αντίθεση μεταξύ των χρωμάτων μπορεί να αναγνωριστεί από μία φωτογραφική μηχανή. Οι απλοί markers επαυξημένης πραγματικότητας συνήθως είναι ένα ή περισσότερα βασικά σχήματα, τοποθετημένα σε ένα μαύρο τετράγωνο με άσπρο φόντο, όπως φαίνεται και στο Σχήμα 2.2. Πιο περίπλοκοι markers, μπορεί να είναι μία κανονική εικόνα, αρκεί να διαβάζονται σωστά από την κάμερα. [2]



Σχήμα 2.2: Ένας απλός marker

Όταν η εικόνα εντοπιστεί από την κάμερα, μετατρέπεται σε μία γκρίζα εικόνα, έτσι ώστε ο αλγόριθμος επεξεργασίας εικόνας να επιταχυνθεί [8]. Αυτός ο τρόπος χρησιμοποιείται όταν είναι αναγκαίο να ενωθεί μία ψηφιακή εικόνα/στοιχείο με τον πραγματικό κόσμο. Εάν ο marker μετακινηθεί ή περιστραφεί, το ίδιο θα γίνει και με την ψηφιακή εικόνα-στοιχείο. Παραδειγματικά, σύμφωνα με το άρθρο [5], αυτός ο τρόπος είναι εξαιρετικός σε μουσεία, όπου τα εκθέματα σημαδεύονται με markers.

2.2.2 Επαυξημένη Πραγματικότητα χωρίς Δείκτη (Markerless / Location-Based / Geo Based Augmented Reality Systems)

Αυτό το είδος επαυξημένης πραγματικότητας παίρνει στοιχεία από το φυσικό περιβάλλον, έτσι ώστε να ανιχνεύσει τη θέση της συσκευής στον πραγματικό κόσμο και να βρει πού πρέπει να μπει το ψηφιακό αντικείμενο. Η markerless επαυξημένη πραγματικότητα δεν χρειάζεται markers. Χρησιμοποιεί GPS, επιταχυνσιόμετρο (accelerometer), ψηφιακή πυξίδα (digital compass) και άλλες τεχνολογίες που βοηθούν έτσι ώστε να βρεθεί η θέση της συσκευής και η θέση του ψηφιακού στοιχείου [5]. Η τεχνολογία SLAM (Simultaneous Localization And Mapping), συνέβαλε στην βελτίωση της ακρίβειας της markerless επαυξημένης πραγματικότητας [13]. Η τεχνολογία αυτή, σαρώνει το φυσικό περιβάλλον και δημιουργεί χάρτες οι οποίοι βοηθούν στην εύρεση της τοποθεσίας που πρέπει να μπει το ψηφιακό αντικείμενο. Ακόμα και αν τα ψηφιακά αντικείμενα δεν είναι στο οπτικό πεδίο του χρήστη, δεν κινούνται όταν ο χρήστης κινείται και ο χρήστης δεν χρειάζεται να σαρώσει ξανά το περιβάλλον [13].

2.3 Κινητή Οπτική Αναζήτηση (Mobile Visual Search)

Αυτός ο τύπος αναγνώρισης εικόνας είναι επίσης σχεδιασμένος για κινητές συσκευές. Μέσα από μία εικόνα-ερώτηση που έχει τραβηχτεί με την συσκευή, μπορεί να βρεθεί η επιθυμητή πληροφορία στο Διαδίκτυο. Το περιεχόμενο, το σχήμα και το χρώμα της εικόνας συγκρίνονται με τα δεδομένα στην βάση δεδομένων και επιστρέφεται ένα κατά προσέγγιση αποτέλεσμα από το Διαδίκτυο [1].

Η διαδικασία έχει ως εξής:

1. Η εικόνα στέλνεται στον application server (ένας διακομιστής ο οποίος κάνει host εφαρμογές) και εκεί η εικόνα αναλύεται με βάση διάφορες ομάδες στοιχείων. Η κάθε ομάδα στοιχείων εξειδικεύεται σε ένα διαφορετικό μέρος της δημιουργίας μιας εικόνας.
2. Αποφασίζεται εάν η εικόνα που παραλήφθηκε περιέχει πεδίο της ειδικότητας του κάθε στοιχείου ή όχι.
3. Ένας κεντρικός υπολογιστής αναλύει τα δεδομένα και δημιουργεί μία σελίδα με τα ταξινομημένα αποτελέσματα της κάθε ομάδας, και η σελίδα αυτή αποστέλλεται τελικά στη συσκευή του χρήστη [16].

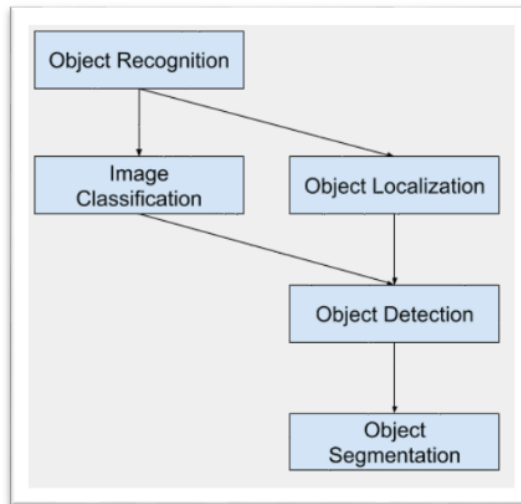
2.4 Αναγνώριση Αντικειμένου (Object Recognition)

Η αναγνώριση αντικειμένου (Object Recognition) είναι μία τεχνική για ταυτοποίηση αντικειμένων σε εικόνες ή βίντεο με την «όραση» ενός υπολογιστή. Η αναγνώριση του αντικειμένου γίνεται εφικτή με μία συλλογή από συνδεδεμένα βήματα :

1. Ταξινόμηση Εικόνας (Image Classification): Η πρόβλεψη του τύπου (ή της κατηγορίας) του αντικειμένου σε μία εικόνα.
2. Εντοπισμός Αντικειμένων (Object Localization): Εντοπισμός του αντικειμένου στην εικόνα και υπόδειξη της θέσης του με ένα bounding box (Δηλαδή ένα πλαίσιο που περικλείει το αντικείμενο).
3. Ανίχνευση Αντικειμένων (Object Detection): Εντοπισμός της παρουσίας του αντικειμένου με bounding box και τύπων ή κατηγοριών αντικειμένων που βρίσκονται μέσα στην εικόνα.

Ακόμη, μία περαιτέρω επέκταση σε αυτά τα βήματα είναι η Κατάτμηση Αντικειμένων (Object Segmentation), όπου αναγνωρισμένα αντικείμενα υποδεικνύονται από τα συγκεκριμένα εικονοκύτταρα (pixels) αντί με bounding box [1].

Στο Σχήμα 2.3 [1] φαίνεται η διαδικασία που ακολουθείται για την αναγνώριση αντικειμένου.



Σχήμα 2.3: Object Recognition: Βήματα

Δύο τεχνικές που είναι σήμερα δημοφιλείς για επίλυση προβλημάτων αναγνώρισης αντικειμένων, είναι η μηχανική μάθηση (machine learning) και η βαθιά μάθηση (deep learning) [10].

Η βαθιά μάθηση λειτουργεί αρχικά με την εκπαίδευση ενός μοντέλου από το μηδέν και έπειτα χρησιμοποιώντας ένα προηγούμενο μοντέλο βαθιάς μάθησης, κάτι το οποίο εξοικονομεί χρόνο. Η βαθιά μάθηση χρειάζεται μεγάλο όγκο πληροφοριών έτσι ώστε να προβλέψει αποτελέσματα μεγάλης ακρίβειας.

Ένας από τους πιο δημοφιλείς αλγόριθμους για βαθιά μάθηση είναι ένα convolutional neural network (CNN or ConvNet). Τα CNNs μαθαίνουν απευθείας από δεδομένα της εικόνας και χρησιμοποιούνται κυρίως για να βρίσκουν μοτίβα σε εικόνες για να αναγνωρίζουν αντικείμενα, πρόσωπα ή σκηνές [3].

Κάποιες δημοφιλείς προσεγγίσεις βαθιάς μάθησης χρησιμοποιώντας CNNs είναι τα R-CNN and YOLO v2, τα οποία αυτόματα μαθαίνουν να ανιχνεύουν αντικείμενα σε εικόνες.

Κατά την φάση της ανίχνευσης αντικειμένων (Object Detection), καλούμαστε να αποφασίσουμε τον τύπο του δικτύου ανίχνευσης αντικειμένων που θέλουμε να χρησιμοποιήσουμε. Για παράδειγμα, το R-CNN είναι ένα δίκτυο δύο φάσεων, ενώ το YOLO v2 είναι δίκτυο μίας φάσης. [18]

Η μηχανική μάθηση λειτουργεί χρησιμοποιώντας αρχικά με μία συλλογή από εικόνες ή βίντεο, από τις οποίες επιλέγονται τα σχετικά χαρακτηριστικά της κάθε μίας. Τα χαρακτηριστικά αυτά προστίθενται σε ένα μοντέλο μηχανικής μάθησης, το οποίο τα ξεχωρίζει σε διαφορετικές κατηγορίες, έτσι ώστε να χρησιμοποιηθούν αυτές οι πληροφορίες για ανάλυση και ταξινόμηση καινούργιων αντικειμένων. Μερικές τεχνικές μηχανικής μάθησης είναι τα συγκεντρωτικά χαρακτηριστικά καναλιού (Aggregate channel features - ACF), η ταξινόμηση SVM (Support Vector Machine) χρησιμοποιώντας ιστογράμματα προσανατολισμένης κλίσης (Histogram of oriented gradients -HOG), και ο αλγόριθμος Viola-Jones για αναγνώριση ανθρωπίνου προσώπου ή άνω σώματος [10].

2.5 Προσαρμογή Προτύπου (Template Matching)

Η προσαρμογή προτύπου είναι επίσης μία τεχνική στο πεδίο της επεξεργασίας ψηφιακής εικόνας που χρησιμοποιείται για εύρεση κομματιών σε μία εικόνα που ταιριάζουν με μία εικόνα-πρότυπο. Μπορεί να χρησιμοποιηθεί ως μέρος ποιοτικού ελέγχου, ως τρόπος πλοήγησης σε ένα κινητό ρομπότ ή ως τρόπος ανίχνευσης των άκρων σε εικόνες.

Το template matching μπορεί να γίνει με δύο τρόπους, χρησιμοποιώντας τα χαρακτηριστικά ή χρησιμοποιώντας την εικόνα-πρότυπο.

Η προσέγγιση βάση χαρακτηριστικών, βασίζεται στην απόσπαση χαρακτηριστικών της εικόνας (π.χ χρώματα, σχήμα), για να την ταιριάξει με την εικόνα-στόχο. Η προσέγγιση αυτή επιτυγχάνεται με τη βοήθεια νευρωνικών δικτύων και ταξινομητών βαθιάς μάθησης (VGG, AlexNet, ResNet).

Η προσέγγιση βάση προτύπου χρησιμοποιείται σε εικόνες που δεν περιέχουν κάποιο δυνατό χαρακτηριστικό, ή για εικόνες-πρότυπα που το κύριο μέρος τους αποτελεί την εικόνα στόχο [20].

Κεφάλαιο 3

Σχεδίαση Εφαρμογής

3.1 Εισαγωγή	12
3.2 Σχεδίαση Εφαρμογής	12
3.2.1 Υλοποίηση με Κινητή Επαυξημένη Πραγματικότητα	13
3.2.2 Υλοποίηση με Κινητή Οπτική Αναζήτηση	14
3.2.3 Υλοποίηση με Αναγνώριση Αντικειμένου	15
3.2.4 Υλοποίηση με Προσαρμογή Προτύπου	16
3.2.5 Τελική Σχεδίαση Εφαρμογής	16
3.3 Αρχιτεκτονική Εφαρμογής	17

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο, παρουσιάζονται η αρχιτεκτονική και η σχεδίαση του συστήματος. Στα πλαίσια της ανάλυσης και έρευνας για την υλοποίηση της εφαρμογής, στο Κεφάλαιο 2, έχουν περιγραφεί 4 μεθοδολογίες που μπορούν να χρησιμοποιηθούν. Στο υποκεφάλαιο 3.2 «Σχεδίαση Συστήματος» παρουσιάζονται οι τρόποι οι οποίοι θα μπορούν να εφαρμοστούν για την υλοποίηση της εφαρμογής και στο υποκεφάλαιο 3.4 «Υλοποίηση Εφαρμογής», παρουσιάζεται ο τρόπος που τελικά επιλέχθηκε για την υλοποίηση της εφαρμογής.

3.2 Σχεδίαση Εφαρμογής

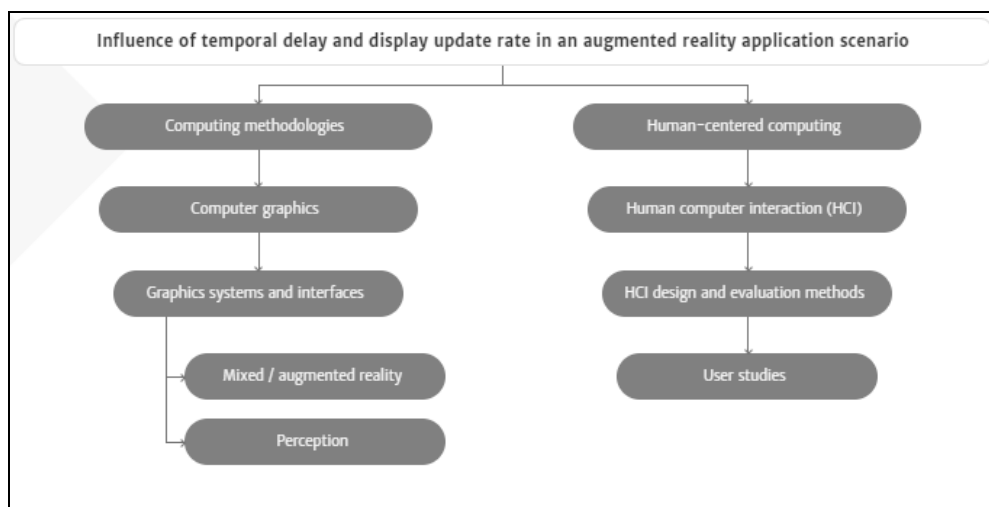
Στο υποκεφάλαιο αυτό παρουσιάζονται οι τρόποι με τους οποίους είναι δυνατό να υλοποιηθεί η εφαρμογή.

3.2.1 Υλοποίηση με Κινητή Επαυξημένη Πραγματικότητα

Η επαυξημένη πραγματικότητα είναι ένας μοντέρνος και σύνθετος τρόπος να υλοποιήσουμε το έργο μας.

Στην περίπτωση αυτή, ο χρήστης σαρώνει με την κινητή συσκευή του τη σειρά από σχήματα, και – πιθανόν με μία καθυστέρηση [7] - τα κατάλληλα γράμματα εμφανίζονται στην οθόνη του πάνω από τα αντίστοιχα σχήματα.

Στο Σχήμα 3.1 [7] φαίνονται κάποιοι παράγοντες που μπορεί να επηρεάσουν την καθυστέρηση της φόρτωσης μιας εφαρμογής επαυξημένης πραγματικότητας.



Σχήμα 3.1: Παράγοντες για καθυστέρηση σε AR εφαρμογή

Η υλοποίηση μπορεί να γίνει με marker-based Augmented Reality, χρησιμοποιώντας ως marker το κάθε σχήμα ξεχωριστά στο πλαίσιο της κάμερας. Για παράδειγμα, σε κάθε σχήμα μπορεί να υπάρχει ο αντίστοιχος ασπρόμαυρος marker που θα αναγνωρίζεται για το κάθε γράμμα, και έπειτα τα γράμματα εμφανίζονται στη θέση του αντίστοιχου σχήματος. Αυτή θα αποτελούσε μια πιο περίπλοκη υλοποίηση, όπως αναφέρθηκε στο υποκεφάλαιο Mobile Augmented Reality.

Ακόμη, η υλοποίηση μπορεί να γίνει με markerless Augmented Reality, όπου τα γράμματα θα εμφανίζονται στην οθόνη του χρήστη χωρίς η κάμερα να διαβάσει κάποιο marker, αλλά αναγνωρίζοντας την τοποθεσία της κινητής συσκευής, και να εμφανίζει στον χρήστη την ιστορία που αντιστοιχεί σε αυτή την θέση. Έτσι, ο χρήστης χρησιμοποιεί την κάμερα της κινητής συσκευής του, σαρώνει το πεδίο που χρειάζεται,

και με τη βοήθεια της τεχνολογίας SLAM (υποκεφάλαιο 2.2.2 Markerless / Location-Based / Geo Based Augmented Reality Systems), τα γράμματα ή η ιστορία εμφανίζονται στην οθόνη του.

Επίσης, η υλοποίηση θα μπορούσε να γίνει αναγνωρίζοντας όχι μόνο ένα-ένα τα σχήματα, αλλά αναγνωρίζοντας ολόκληρη την εικόνα που σαρώνει ο χρήστης. Η διαδικασία παραμένει η ίδια, με τη διαφορά ότι αντί να εμφανίζονται γράμματα πάνω από το κάθε σχήμα, θα εμφανίζεται ολόκληρη η ιστορία πάνω από την εικόνα που σαρώνει ο χρήστης.

Η υλοποίηση μπορεί να γίνει με τη βοήθεια του Android Studio, είτε του Unity, σε συνδυασμό με τα κατάλληλα SDKs (Software Developer Kits), όπως:

- Google's ARCore (Android Studio)
- DroidAR (Android Studio), για αναγνώριση τοποθεσίας
- Vuforia (Unity)
- ARKit (Unity)

Η υλοποίηση με επαυξημένη πραγματικότητα δεν θα μπορούσε να γίνει σε μορφή κρεμάλας, αφού, κατά την μελέτη που έγινε για το κεφάλαιο 2, δεν βρέθηκε τρόπος με τον οποίο μπορεί να γίνει αποθήκευση της εικόνας που αναγνωρίζεται από την AR κάμερα

3.2.2 Υλοποίηση με Κινητή Οπτική Αναζήτηση

Η εφαρμογή θα μπορούσε να υλοποιηθεί με Mobile Visual Search, σαρώνοντας ολόκληρη την εικόνα μέσω της κινητής συσκευής του χρήστη και συγκρίνοντας με τις εικόνες που υπάρχουν στην βάση δεδομένων μας. Βάσει της ομάδας σχημάτων που αναγνωρίζονται από τη βάση (της εικόνας που ταιριάζει καλύτερα με αυτή που σάρωσε ο χρήστης), η εφαρμογή παραπέμπει τον χρήστη σε μία αντίστοιχη ιστοσελίδα που περιέχει την κατάλληλη ιστορία.

Ακόμη, η υλοποίηση θα μπορούσε να γίνει και σε μορφή κρεμάλας. Σαρώνοντας ολόκληρη την εικόνα μέσω της κινητής συσκευής του χρήστη, η εφαρμογή παραπέμπει τον χρήστη σε μια αντίστοιχη ιστοσελίδα που περιέχει το παιχνίδι κρεμάλα. Στην

ιστοσελίδα εμφανίζονται στον χρήστη οι εικόνες - σύμβολα που αντιστοιχούν σε κάθε χαρακτήρα της ιστορίας, και τα γράμματα του αλφάβητου. Ο χρήστης επιλέγει το γράμμα που επιθυμεί και, αν αυτό αντιστοιχεί σε κάποια εικόνα-σύμβολο τότε την θέση της παίρνει το συγκεκριμένο γράμμα, αλλιώς ο χρήστης προσπαθεί ξανά.

3.2.3 Υλοποίηση με Αναγνώριση Αντικειμένου

Ένας σχετικά πιο απλός τρόπος υλοποίησης της εφαρμογής είναι το Object Recognition. Σε αυτή την περίπτωση, το κάθε σχήμα που υπάρχει στην σαρωμένη εικόνα, συγκρίνεται με τα περιεχόμενα της βάσης δεδομένων, και, όπως και στην υλοποίηση του Augmented Reality, αντιστοιχεί σε ένα συγκεκριμένο γράμμα. Χρησιμοποιώντας Object Recognition, η εφαρμογή μπορεί να εκτελεστεί με περισσότερους από έναν τρόπους.

Παραδειγματικά, μπορεί να έχει τη μορφή ενός παιχνιδιού-κρεμάλας, όπου ο χρήστης είναι αυτός ο οποίος αποκωδικοποιεί το μήνυμα. Δηλαδή, έστω ότι ο χρήστης προσπαθεί να μαντέψει σε ποιο γράμμα αντιστοιχεί ένα συγκεκριμένο σχήμα. Τότε, ο χρήστης επιλέγει το σχήμα που επιθυμεί, και του εμφανίζεται κάποιο πλαίσιο για να καταθέσει το γράμμα που πιστεύει πως του αντιστοιχεί. Η σύγκριση ελέγχεται μέσω των στοιχείων/εικόνων που βρίσκονται στη βάση δεδομένων και ανάλογα του εμφανίζεται κατάλληλο μήνυμα (να προσπαθήσει ξανά ή ότι η αντιστοιχία του ήταν σωστή). Έχοντας ένα όριο ευκαιριών για προσπάθειες, εάν ο χρήστης δεν μαντέψει σωστά κανένα γράμμα, τότε είτε μπορεί να δοκιμάσει αργότερα, είτε η ιστορία του εμφανίζεται χωρίς να έχει ολοκληρώσει το παιχνίδι.

Ένας ακόμη τρόπος για την υλοποίηση της εφαρμογής με Object Recognition, είναι μία απλή εμφάνιση της ιστορίας σε μία καινούρια οθόνη όταν τα γράμματα σαρωθούν από την συσκευή, χωρίς δηλαδή την υλοποίηση του παιχνιδιού-κρεμάλας.

Η υλοποίηση επίσης μπορεί να γίνει αναγνωρίζοντας ολόκληρη την εικόνα και συγκρίνοντας την με στοιχεία στη βάση δεδομένων (αντίστοιχο κείμενο-ιστορία ή εικόνα), εμφανίζει την κατάλληλη πληροφορία.

Η υλοποίηση μπορεί να γίνει με τη βοήθεια του Android Studio, στο οποίο μπορούν να εγκατασταθούν διάφορες βιβλιοθήκες όπως:

- Tensorflow (Object Detection API)
- Firebase ML Kit Object Detection API
- OpenCV Image Recognition

Μία ακόμη επιλογή θα ήταν και το Unity, με συνδυασμό το Vuforia, όμως η βιβλιοθήκη η οποία χρησιμοποιείται από το Vuforia για Object Recognition, δεν υποστηρίζεται σε αρκετές συσκευές.

3.2.4 Υλοποίηση με Προσαρμογή Προτύπου

Σε αυτή την περίπτωση, η υλοποίηση είναι περίπου η ίδια με αυτή του Object Recognition. Η υλοποίηση θα πρέπει να γίνει με βάση τα χαρακτηριστικά, αφού η εικόνα θα πρέπει να έχει ένα αριθμό από χαρακτηριστικά σχήματα, τα οποία πρέπει να αναγνωριστούν.

Η υλοποίηση μπορεί να γίνει με τη βοήθεια του Android Studio, στο οποίο μπορούν να εγκατασταθούν διάφορες βιβλιοθήκες όπως OpenCV.

3.2.5 Τελική Σχεδίαση Εφαρμογής

Ο τρόπος με τον οποίο αποφασίσαμε να υλοποιήσουμε την εφαρμογή είναι με Markerless επαυξημένη πραγματικότητα, χρησιμοποιώντας το εργαλείο Unity. Η μέθοδος της επαυξημένης πραγματικότητας επιλέχθηκε λόγω της ευχάριστης και ενδιαφέρουσας αλληλεπίδρασης μεταξύ χρήστη και εφαρμογής και η markerless επαυξημένη πραγματικότητα επιλέχθηκε λόγω της απλούστερης χρήσης της εφαρμογής. Δηλαδή, τα σύμβολα που χρησιμοποιήθηκαν δεν είναι markers, αλλά απλές εικόνες.

Το σύστημα αποτελείται από την χρήστη, την κινητή συσκευή που χρησιμοποιεί για την σάρωση της κωδικοποιημένης ιστορίας και ακόμη ένα μέσο το οποίο προβάλλει την

κωδικοποιημένη ιστορία. Αυτό μπορεί να είναι μία δεύτερη κινητή συσκευή, ένας υπολογιστής, ένα tablet ή ένα χαρτί με την εκτυπωμένη ιστορία.

Ο χρήστης χρησιμοποιεί την κινητή συσκευή του για να χρησιμοποιήσει τις λειτουργίες Decode και Encode της εφαρμογής. Στην λειτουργία Decode, ο χρήστης καλείται να σαρώσει την κωδικοποιημένη ιστορία με την AR κάμερα της μηχανής Vuforia, ενώ στην λειτουργία Encode ο χρήστης καλείται να εισάγει την δική του ιστορία μέσω πληκτρολογίου, με στόχο την κωδικοποίησή της. Συγκεκριμένα, ο χρήστης αρχικά πρέπει να πληκτρολογήσει την ιστορία του και να την αποθηκεύσει. Ο χρήστης έπειτα έχει τις επιλογές της εμφάνισης της τελευταίας του αποθηκευμένης ιστορίας, ή της κωδικοποίησης της ιστορίας του.

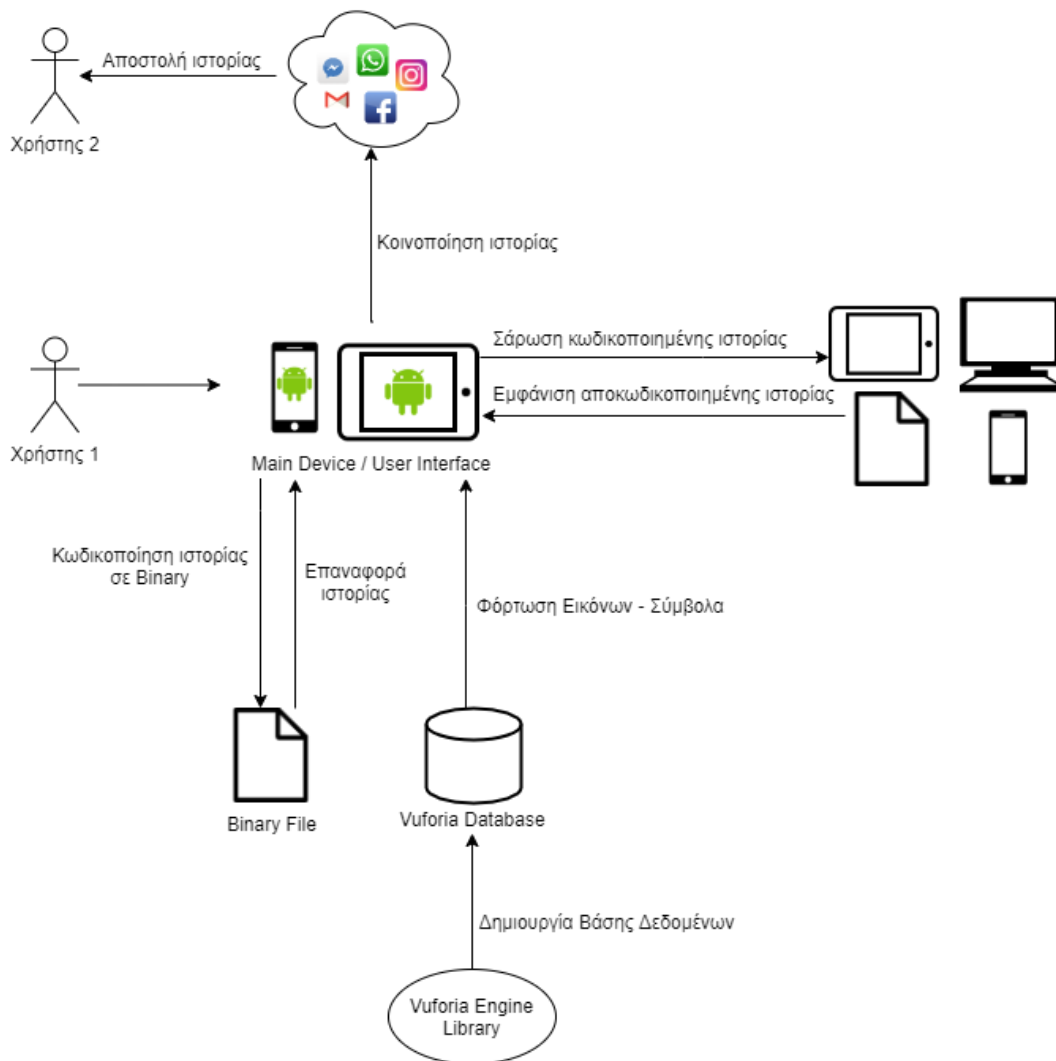
Όταν ο χρήστης επιλέξει να κωδικοποιήσει την ιστορία του, τότε του παρουσιάζεται η επιλογή της λήψης ενός στιγμιότυπου οθόνης, έτσι ώστε να αποθηκεύσει την ολοκληρωμένη κωδικοποιημένη ιστορία του σαν εικόνα στην γκαλερί της εφαρμογής.

3.3 Αρχιτεκτονική Συστήματος

Η αρχιτεκτονική συστημάτων αποτελεί ένα εννοιολογικό μοντέλο που καθορίζει τη συμπεριφορά και τη δομή ενός συστήματος. Το Σχήμα 3.2 περιγράφει την αρχιτεκτονική του συστήματος. Το σύστημα αποτελείται από τον χρήστη της εφαρμογής, ο οποίος, χρησιμοποιώντας την κυρίως Android συσκευή του, σαρώνει μια κωδικοποιημένη ιστορία από ένα άλλο μέσο το οποίο μπορεί να είναι υπολογιστής, κινητή συσκευή, tablet ή χαρτί με την εκτυπωμένη κωδικοποιημένη ιστορία. Τότε, στην κυρίως συσκευή του χρήστη εμφανίζεται η αποκωδικοποιημένη ιστορία. Οι εικόνες – σύμβολα που η AR κάμερα της κυρίως συσκευής του χρήστη αναγνωρίζει, φορτώθηκαν από την βάση δεδομένων που δημιουργήθηκε μέσω της μηχανής Vuforia.

Ο χρήστης, ακόμη, μπορεί να εισάγει την δική του ιστορία, και αποθηκεύοντας την, αυτή κωδικοποιείται σε Binary και αποθηκεύεται στο Binary αρχείο. Όταν ο χρήστης επιθυμεί να εμφανίζει ξανά την κωδικοποιημένη του ιστορία, γίνεται επαναφορά της ιστορίας από το Binary αρχείο και εμφανίζεται στον χρήστη.

Τέλος, ο χρήστης μπορεί να κοινοποιήσει την ιστορία του χρησιμοποιώντας διάφορα μέσα επικοινωνίας, και να την στείλει σε ένα άλλο χρήστη της εφαρμογής.

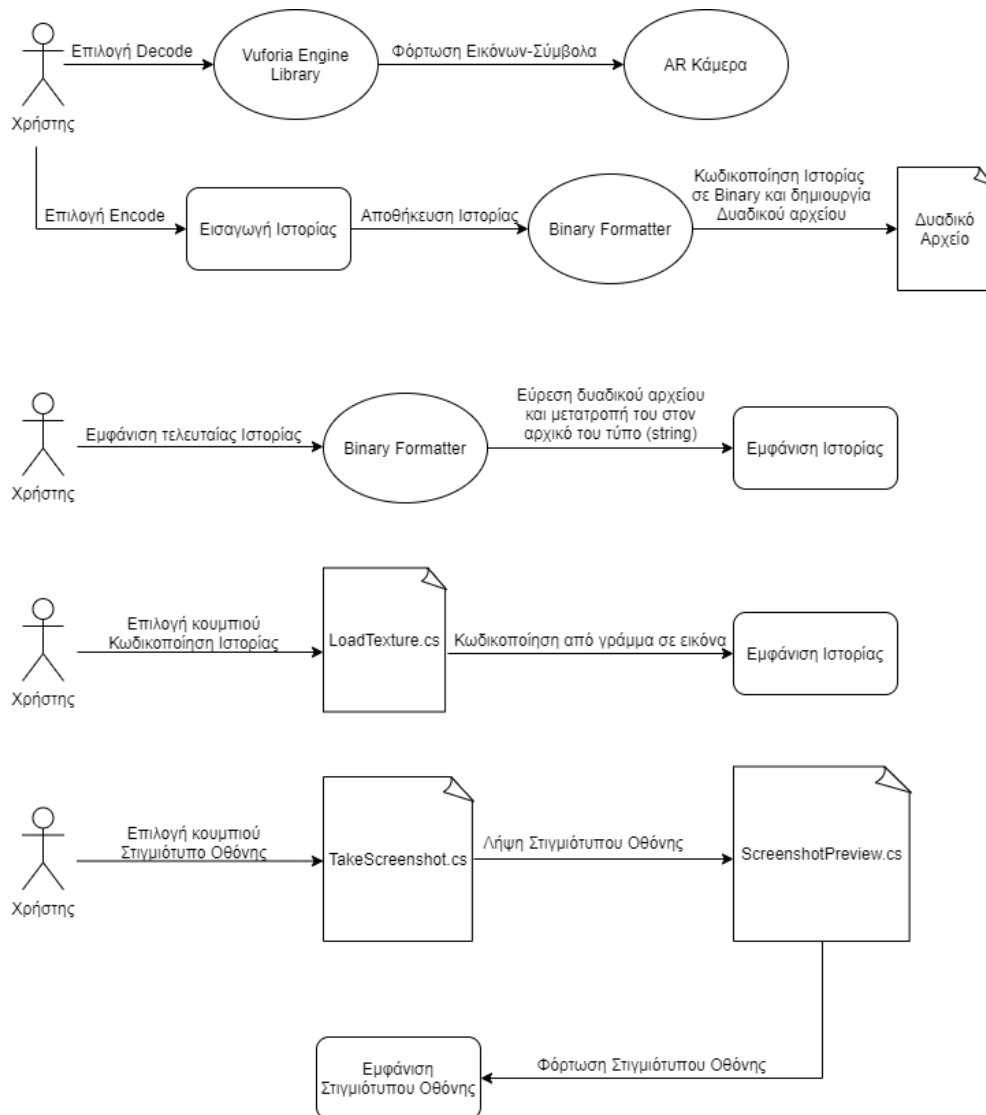


Σχήμα 3.2: Αρχιτεκτονική Συστήματος

Όπως φαίνεται στο Σχήμα 3.3, το οποίο παρουσιάζει τη ροή εργασιών του συστήματος, ο χρήστης αρχικά έχει τις επιλογές Decode και Encode. Όταν ο χρήστης επιλέξει την επιλογή Decode, τότε γίνεται χρήση της βιβλιοθήκης της μηχανής Vuforia, μέσω της οποίας γίνεται η φόρτωση των εικόνων – σύμβολα που η AR κάμερα περιμένει να αναγνωρίσει και τέλος η AR κάμερα ενεργοποιείται.

Επιλέγοντας Encode, ο χρήστης καλείται να εισάγει την ιστορία του και να την αποθηκεύσει. Έτσι, καλείται ο Binary Formatter, ο οποίος κωδικοποιεί την ιστορία σε Binary και δημιουργεί το δυαδικό αρχείο που την περιέχει. Όταν ο χρήστης επιθυμεί να

εμφανίσει την τελευταία του ιστορία, τότε πρέπει να ακουμπήσει το αντίστοιχο κουμπί, και ο Binary Formatter ξανακαλείται για να εντοπίσει το δυαδικό αρχείο και να το μετατρέψει ξανά στην αναγνώσιμη ιστορία. Έπειτα, η ιστορία εμφανίζεται στην οθόνη του χρήστη. Στη συνέχεια, όταν ο χρήστης επιλέξει να κωδικοποιήσει την ιστορία του, τότε μέσω του κώδικα στο αρχείο LoadTexture.cs γίνεται η κωδικοποίηση από γράμμα σε εικόνα και εμφανίζεται η κάθε εικόνα-σύμβολο στην οθόνη του χρήστη. Τέλος, όταν ο χρήστης επιλέξει να γίνει λήψη του στιγμιότυπου οθόνης, για να αποθηκεύσει την ολοκληρωμένη κωδικοποιημένη ιστορία του, τότε, μέσω του κώδικα στο αρχείο TakeScreenshot.cs γίνεται η λήψη του στιγμιότυπου, και μέσω του κώδικα στο αρχείο ScreenshotPreview.cs γίνεται η φόρτωση και η εμφάνιση του στιγμιότυπου στην οθόνη του χρήστη.



Σχήμα 3.3: Ροή Εργασιών Συστήματος

Κεφάλαιο 4

Υλοποίηση Εφαρμογής

4.1 Εισαγωγή	20
4.2 Λειτουργίες Εφαρμογής	21
4.2.1 Αποκωδικοποίηση Ιστορίας	21
4.2.2 Κωδικοποίηση Ιστορίας	22
4.2.3 Οδηγίες Χρήσης	24
4.2.4 Προβολή Ιστορίας	24
4.2.5 Γκαλερί	27
4.2.6 Άλλες Λειτουργίες	29
4.3 Φραγμοί Εφαρμογής	29
4.4 Υλοποίηση Λειτουργιών	30
4.4.1 Vuforia AR Camera	30
4.4.2 Αποθήκευση τελευταίας ιστορίας	32
4.4.3 Βάση Δεδομένων	33

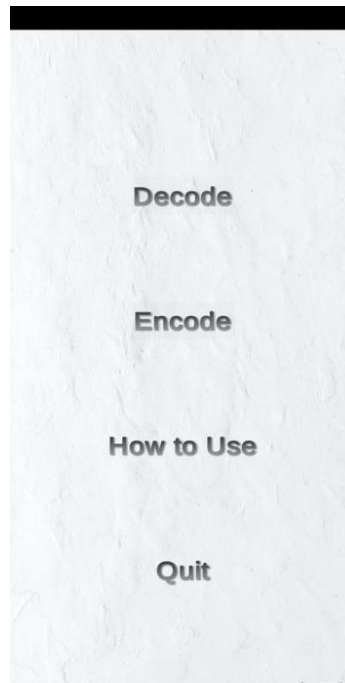
4.1 Εισαγωγή

Το εργαλείο που επιλέχθηκε για την υλοποίηση της εφαρμογής είναι το Unity. Η εφαρμογή έχει υλοποιηθεί με αναγνώριση εικόνας μέσω Augmented Reality, το οποίο επιτυγχάνεται μέσω του software development kit «Vuforia Engine» του Unity. [17] Το Vuforia Engine είναι ένα SDK (Software Development Kit), το οποίο χρησιμοποιείται για την δημιουργία εφαρμογών επαυξημένης πραγματικότητας. Μέσω αυτού, οι προγραμματιστές μπορούν να εισάγουν επιπλέον λειτουργικότητες στην εφαρμογή τους, οι οποίες επιτρέπουν στην εφαρμογή να αναγνωρίζει εικόνες ή αντικείμενα και να αλληλοεπιδρά με τον πραγματικό κόσμο.

4.2 Λειτουργίες Εφαρμογής

Όταν ο χρήστης εκκινήσει την εφαρμογή, εμφανίζεται ένα pop-up μήνυμα, ζητώντας από τον χρήστη έγκριση για την χρήση της κάμερας της συσκευής του. Αυτό γίνεται λόγω της επιλογής της αποκωδικοποίησης, όπου ο χρήστης πρέπει να σαρώσει με την κάμερα του την κωδικοποιημένη ιστορία.

Όταν ο χρήστης επιτρέψει στην εφαρμογή την χρήση της κάμερας, βλέπει το κυρίως μενού, το οποίο περιέχει τις επιλογές της αποκωδικοποίησης (Decode), της κωδικοποίησης (Encode), της επιλογής να διαβάσει πώς να χρησιμοποιήσει την εφαρμογή (How to Use) και της εξόδου από την εφαρμογή (Quit). Βλ. Σχήμα 4.2.



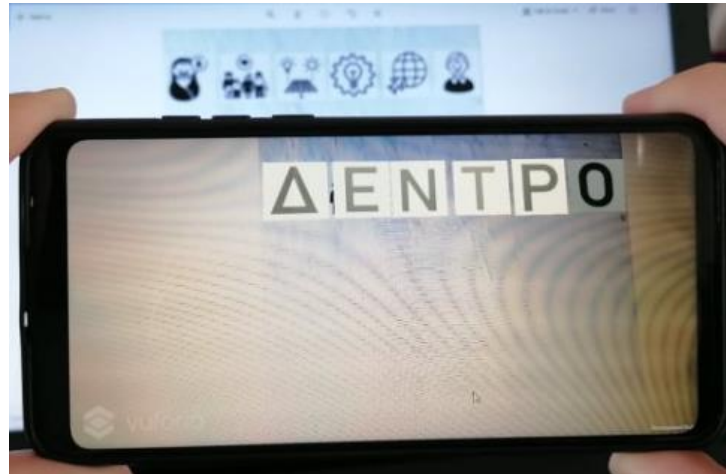
Σχήμα 4.2: Κυρίως Μενού

4.2.1 Αποκωδικοποίηση Ιστορίας

Ο στόχος της λειτουργίας αυτής είναι να εμφανίσει στον χρήστη την ιστορία (σε γράμματα του Ελληνικού αλφάβητου) που αντιστοιχεί στην κωδικοποιημένη ιστορία.

Ο χρήστης πρέπει να ακουμπήσει το κουμπί Decode και αυτό θα τον παραπέμψει στην κάμερα της κινητής συσκευής. Ο χρήστης τότε πρέπει να κρατήσει την συσκευή του πάνω από την επιφάνεια η οποία περιέχει την κωδικοποιημένη ιστορία (εκτυπωμένη σε

χαρτί, οθόνη υπολογιστή ή κινητού ή tablet). Τα σύμβολα πρέπει να φαίνονται ολόκληρα στην οθόνη και δεν είναι αναγκαίο να φαίνονται όλα τα σύμβολα που αποτελούν την ιστορία. Τότε τα αντίστοιχα γράμματα του αλφάβητου θα αντικαταστήσουν τα σύμβολα στην οθόνη του χρήστη. Η σάρωση γίνεται με την βοήθεια της AR κάμερας (Augmented Reality Camera) της βιβλιοθήκης Vuforia του Unity.



Σχήμα 4.3: Αποκωδικοποίηση μηνύματος «ΔΕΝΤΡΟ»

4.2.2 Κωδικοποίηση Ιστορίας

Ο στόχος της λειτουργίας αυτής είναι να κωδικοποιήσει ο χρήστης την δική του ιστορία. Ο χρήστης πρέπει να ακουμπήσει το κουμπί Encode και αυτό θα τον παραπέμψει σε μία καινούρια σελίδα η οποία περιέχει πληκτρολόγιο, και άλλα 3 κουμπιά. Ο χρήστης εδώ πρέπει αρχικά να πληκτρολογήσει την ιστορία του (ο μέγιστος αριθμός χαρακτήρων πρέπει να είναι 66) και έπειτα να την αποθηκεύσει μέσω του κουμπιού «Αποθήκευση Ιστορίας».

Τότε ο χρήστης για να κωδικοποιήσει την ιστορία του, πρέπει να ακουμπήσει το κουμπί «Κωδικοποίηση Ιστορίας» και αυτό θα τον παραπέμψει σε μία καινούρια σελίδα η οποία περιέχει την κωδικοποιημένη ιστορία του χρήστη (περισσότερα στην υποκατηγορία «Προβολή Ιστορίας»).

Το κουμπί «Κωδικοποίηση Ιστορίας» κωδικοποιεί την τελευταία ιστορία που αποθηκεύτηκε. Δηλαδή, αν ο χρήστης πληκτρολογήσει την ιστορία του, αλλά δεν την αποθηκεύσει, η ιστορία που θα κωδικοποιηθεί δεν θα είναι αυτή που ο χρήστης έχει πληκτρολογήσει, αλλά η ιστορία που είχε αποθηκευτεί τελευταία. Αν ο χρήστης δεν είχε αποθηκεύσει καμία ιστορία, η ιστορία που κωδικοποιείται είναι κενή.

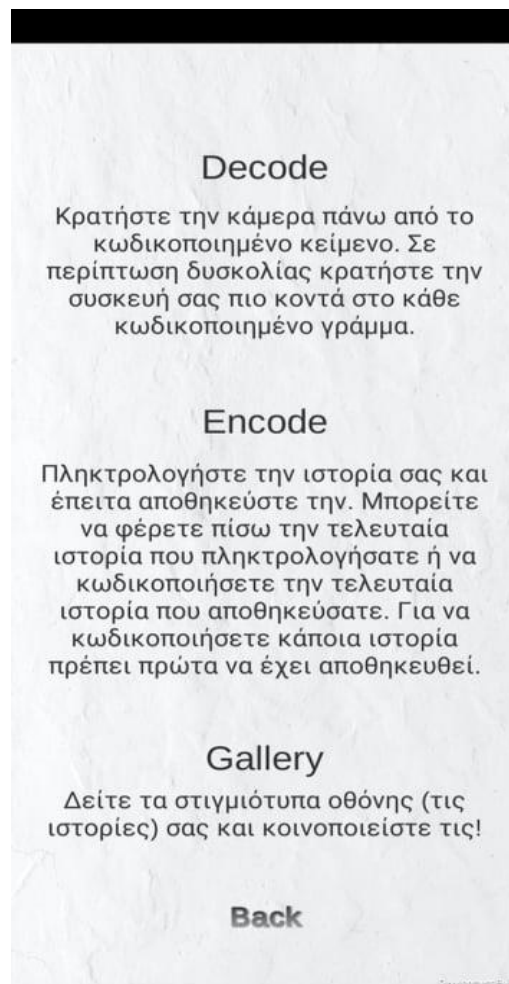
Ακόμη, ο χρήστης σε αυτή τη σελίδα έχει την επιλογή να εμφανίσει ξανά την τελευταία ιστορία που αποθήκευσε, μέσω του κουμπιού «Εμφάνιση τελευταίας Ιστορίας». Αυτή η λειτουργία μπορεί να φανεί χρήσιμη όταν ο χρήστης κλείσει την εφαρμογή, και όταν επιστρέψει θα ήθελε να θυμηθεί ή να κωδικοποιήσει ξανά την τελευταία του ιστορία. Η λειτουργία αυτή γίνεται με την βοήθεια δυαδικών αρχείων-binary files (περισσότερα στην υποκατηγορία 4.4.2 Αποθήκευση τελευταίας ιστορίας).

The screenshot shows a mobile application interface with a light gray background. At the top, there is a black header bar. Below it, the text "Εισάγετε την ιστορία σας:" is displayed in a bold, black font. Underneath this text is a large, empty white rectangular box for text input. To the right of the bottom of this box, the text "Απομένουν 66 χαρακτήρες" is visible. Below the input box is a keyboard with Greek letters arranged in four rows: Α Β Γ Δ Ε Ζ Η, Θ Ι Κ Λ Μ Ν Ξ, Ο Π Ρ Σ Τ Υ Φ, and Χ Ψ Ω ! . ;. Below the keyboard is a gray rectangular button labeled "Delete". At the bottom of the screen, there are four buttons arranged in two rows: "Αποθήκευση Ιστορίας" and "Εμφάνιση τελευταίας Ιστορίας" in the first row, and "Κωδικοποίηση Ιστορίας" and "Back" in the second row. The "Quit" button is located at the bottom right of the screen.

Σχήμα 4.4: Κωδικοποίηση

4.2.3 Οδηγίες Χρήσης

Ο στόχος της λειτουργίας αυτής είναι η σύντομη επεξήγηση της λειτουργίας της εφαρμογής, έτσι ώστε ο χρήστης να την χρησιμοποιήσει σωστά. Ο χρήστης πρέπει να ακουμπήσει το κουμπί How to Use και αυτό θα τον παραπέμψει σε μία καινούρια σελίδα η οποία περιέχει σύντομες περιγραφές για την κάθε λειτουργία της εφαρμογής.



Σχήμα 4.5: How to Use

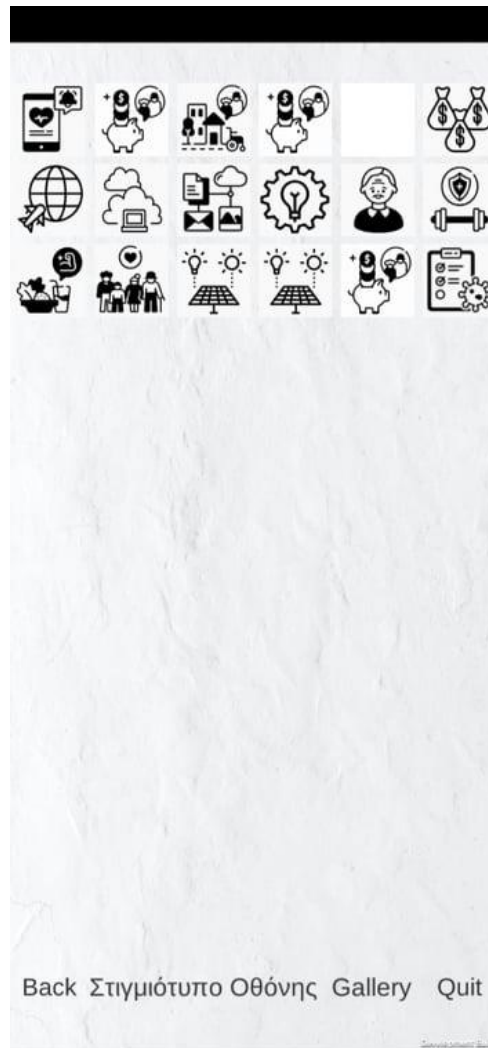
4.2.4 Προβολή Ιστορίας

Ο στόχος της λειτουργίας αυτής είναι ο χρήστης να δει την ιστορία που κωδικοποίησε τελευταία. Ο χρήστης καταλήγει σε αυτή τη λειτουργία όταν ακουμπήσει το κουμπί «Κωδικοποίηση Ιστορίας». Αν η ιστορία που ο χρήστης αποθήκευσε ήταν κενή (βλ. Σχήμα 4.6), τότε η σελίδα αυτής της λειτουργίας εμφανίζεται κενή (χωρίς κάποιο σύμβολο), αφού δεν υπάρχει κάποιο γράμμα για να κωδικοποιηθεί.



Σχήμα 4.6: Κενή Ιστορία

Στην περίπτωση που ο χρήστης έχει αποθηκεύσει κάποια μη κενή ιστορία, τότε η ιστορία αυτή εμφανίζεται στον χρήστη.

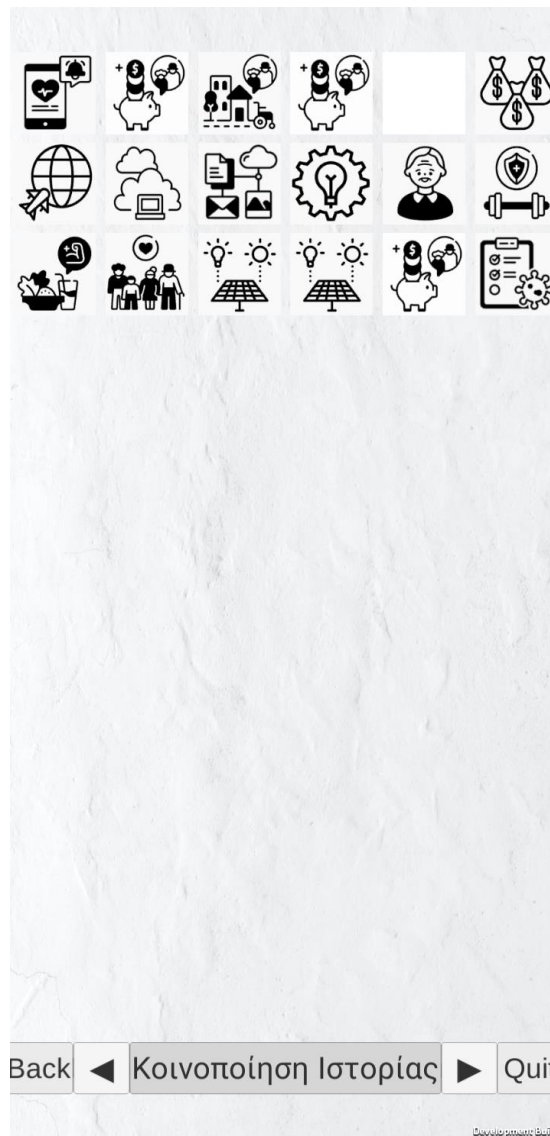


Σχήμα 4.7: Μη κενή ιστορία

Το Σχήμα 4.7 αποτελεί παράδειγμα κωδικοποιημένης ιστορίας που περιέχει το μήνυμα «ΚΑΛΑ ΧΡΙΣΤΟΥΓΕΝΝΑ!». Στη σελίδα αυτής της λειτουργίας δίνεται ακόμη στον χρήστη η επιλογή να αποθηκεύσει την ιστορία του στην γκαλερί της εφαρμογής, ακουμπώντας το κουμπί «Στιγμιότυπο Οθόνης» (περισσότερα στην υποκατηγορία «Γκαλερί»). Ακουμπώντας αυτό το κουμπί, ο χρήστης βγάζει φωτογραφία την οθόνη του, η οποία περιέχει την ιστορία που κωδικοποίησε, και αυτή η φωτογραφία αποθηκεύεται στην γκαλερί της εφαρμογής.

4.2.5 Γκαλερί

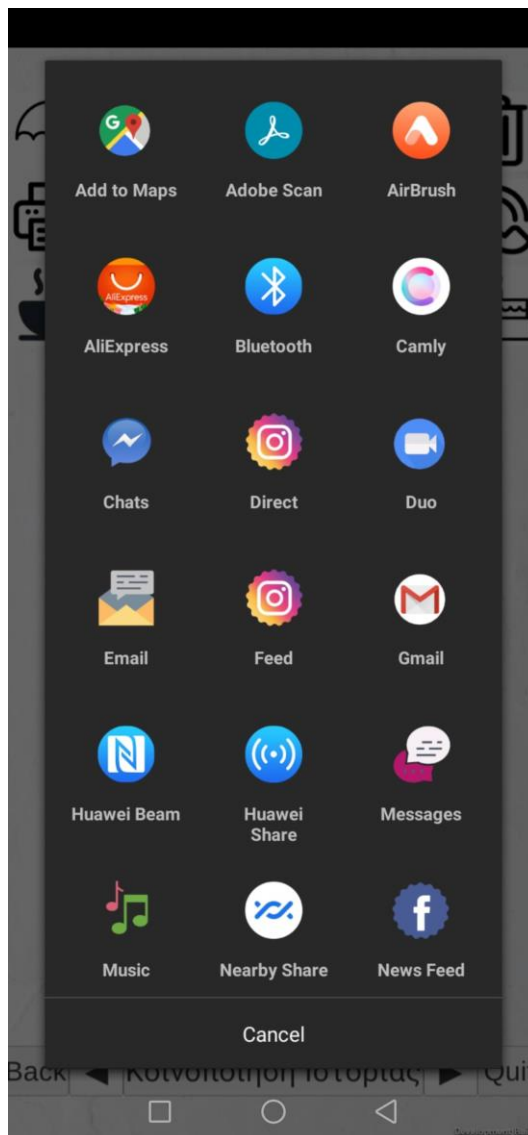
Ο στόχος της λειτουργίας αυτής είναι ο χρήστης να δει όλες τις ιστορίες που κωδικοποίησε. Ο χρήστης καταλήγει σε αυτή τη λειτουργία όταν ακουμπήσει το κουμπί «Γκαλερί», από την προηγούμενη σελίδα, η οποία ήταν αυτή που περιγράφεται στην υποκατηγορία «Προβολή Ιστορίας».



Σχήμα 4.8: Γκαλερί

Σε αυτή τη σελίδα ο χρήστης έχει τη δυνατότητα να δει όλες τις κωδικοποιημένες του ιστορίες. Στο Σχήμα 4.8 φαίνεται ένα παράδειγμα εμφάνισης ιστορίας στη σελίδα της γκαλερί. Με τα κουμπιά «βέλη», ο χρήστης μπορεί να μετακινηθεί μπροστά και πίσω στην γκαλερί για να δει την προηγούμενη ιστορία (αριστερό βέλος) ή την επόμενη (δεξί

βέλος). Με το κουμπί «Κοινοποίηση Ιστορίας», ο χρήστης μπορεί να κοινοποιήσει την ιστορία που βλέπει στην οθόνη του σε κάποιο φίλο του.



Σχήμα 4.9: Κοινοποίηση

Στο Σχήμα 4.9 φαίνεται η οθόνη του χρήστη αφού ακουμπήσει το κουμπί «Κοινοποίηση Ιστορίας», όπου ο χρήστης επιλέγει την εφαρμογή με την οποία επιθυμεί να κοινοποιήσει την ιστορία του.

4.2.6 Άλλες λειτουργίες

Σε κάθε σελίδα της εφαρμογής, εκτός από την σελίδα που γίνεται χρήση της κάμερας, υπάρχει το κουμπί «Quit». Με αυτό το κουμπί ο χρήστης έχει την επιλογή να κλείσει την εφαρμογή οποιαδήποτε στιγμή επιθυμεί, ανεξάρτητα της λειτουργίας που χρησιμοποιεί.

Ακόμη, σε όλες τις σελίδες, εκτός από την σελίδα που γίνεται χρήση της κάμερας και το κυρίως μενού (αρχική σελίδα), υπάρχει το κουμπί «Back». Με αυτό το κουμπί ο χρήστης έχει την επιλογή να επιστρέψει στην προηγούμενη σελίδα από αυτή που βρισκόταν. Για παράδειγμα, αν ο χρήστης βρίσκεται στην σελίδα «How to Use» , τότε πατώντας του κουμπί «Back» θα επιστρέψει πίσω στο κυρίως μενού.

4.3 Φραγμοί Εφαρμογής

Κατά την υλοποίηση της εφαρμογής, συναντήσαμε προβλήματα σε πολλαπλούς τομείς. Λόγω της χρήσης της κάμερας, ο χρήστης αφού αποκωδικοποιήσει το μήνυμα που επιθυμεί, δεν έχει την δυνατότητα να επιστρέψει απευθείας στην εφαρμογή. Αντίθετα, πρέπει να κλείσει την εφαρμογή και να την ξανανοίξει, έτσι ώστε να έχει πρόσβαση και στις υπόλοιπες λειτουργίες της. Μετά από μελέτη, δεν βρέθηκε κάποιος τρόπος για την υλοποίηση ενός ανάλογου κουμπιού, έτσι η εφαρμογή περιορίστηκε σε αυτό τον τομέα. Ακόμη, στην λειτουργία της κωδικοποίησης ιστορίας, ήταν αναγκαίο να υλοποιήσουμε το δικό μας πληκτρολόγιο, αφού η επιλογή χρήσης πληκτρολογίου στο Unity είναι δυνατή μόνο με την χρήση υπολογιστή (με λειτουργικό σύστημα Windows), και όχι σε οποιοδήποτε άλλο λειτουργικό σύστημα (π.χ. Android).

Επίσης, κατά την εμφάνιση της κωδικοποιημένης ιστορίας του χρήστη, κληθήκαμε να εμφανίσουμε τις εικόνες-σύμβολα στην οθόνη σε ένα ικανοποιητικό μέγεθος, έτσι ώστε η προβολή τους σε άλλα μέσα (π.χ εκτυπωμένο χαρτί ή άλλη οθόνη) να είναι αναγνωρίσιμη από την εφαρμογή για την αποκωδικοποίησή τους.

Αυτό αποτέλεσε αιτία για τον περιορισμό του μεγέθους της ιστορίας που καλείται να δημιουργήσει ο χρήστης. Η ιστορία περιορίζεται σε 66 χαρακτήρες, αφού η οθόνη είναι σε θέση να εμφανίσει 66 σύμβολα ικανοποιητικού μεγέθους.

Επιπλέον, σχετικά με την αναγνώριση των συμβόλων, η ευκολία αναγνώρισης τους δεν είναι η ίδια για όλα τα σύμβολα, αφού η αναγνωσιμότητα του κάθε συμβόλου βαθμολογείται από την βάση δεδομένων της μηχανής Vuforia (περισσότερα στην υποκατηγορία 4.4 Υπόβαθρα Λειτουργιών). Έτσι, κάποια σύμβολα μπορεί να αναγνωρίζονται με περισσότερη δυσκολία από άλλα. Λύση για αυτό μπορεί να αποτελέσει η εστίαση σε λιγότερα σχήματα (αντί η κάμερα να βλέπει ολόκληρη την ιστορία). Για παράδειγμα, ο χρήστης μπορεί να δείχνει στην κάμερα ένα ή δύο σχήματα κάθε φορά.

4.4 Υλοποίηση Λειτουργιών

Σε αυτό το υποκεφάλαιο περιγράφονται οι διαδικασίες υλοποίησης των λειτουργιών της εφαρμογής.

4.4.1 Vuforia AR Camera

Η κάμερα επαυξημένης πραγματικότητας της μηχανής Vuforia [4] επιτρέπει στον χρήστη να «αντικαταστήσει» με την κινητή του συσκευή, ένα αντικείμενο που βρίσκεται στο φυσικό περιβάλλον, με ένα άλλο αντικείμενο. Δηλαδή, η κάμερα αυτή «ψάχνει» τους στόχους που ορίστηκαν, και όταν τους εντοπίσει, αντικαθιστά στην οθόνη του χρήστη τον στόχο αυτό με μία άλλη εικόνα ή αντικείμενο που ορίζεται μέσω του Unity. Αρχικά εισάγουμε στο περιβάλλον μας τα αντικείμενα (Image Targets) που η κάμερα επιχειρεί να εντοπίσει (αυτά είναι τα σύμβολα που έχουν αποθηκευτεί στην βάση δεδομένων- περισσότερα στην υποκατηγορία 4.4.3 Βάση Δεδομένων).

Έπειτα, προσαρμόζουμε σε κάθε Image Target την εικόνα-σύμβολο που επιθυμούμε, έτσι ώστε να υπάρχουν όλα τα σύμβολα στο περιβάλλον του Unity. Αυτό γίνεται μέσω της επιλογής Image Target Behavior του κάθε Image Target, αφού η βάση έχει προστεθεί στο project (περισσότερα στην υποκατηγορία 4.4.3 Βάση Δεδομένων).

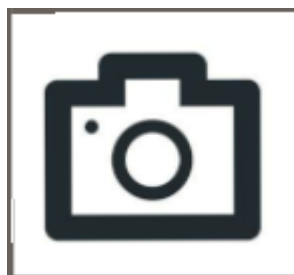
Το επόμενο βήμα είναι να δημιουργήσουμε το αντικείμενο το οποίο αντιπροσωπεύει την εικόνα-γράμμα που θα εμφανίζεται πάνω από κάθε σύμβολο. Αυτό γίνεται με τον εξής τρόπο:

Αρχικά γίνεται η δημιουργία του αντικειμένου, έστω «MyMaterial» στον φάκελο «Resources», όπου υπάρχουν και οι εικόνες με τα γράμματα του αλφάβητου.

Στη συνέχεια το αντικείμενο αυτό πρέπει να προσαρμοστεί έτσι ώστε να έχει την εμφάνιση που επιθυμούμε, και αυτό γίνεται αλλάζοντας την επιλογή Albedo του αντικειμένου. Για την παρούσα εφαρμογή αρκεί να σύρουμε στην επιλογή Albedo την εικόνα με το γράμμα που επιθυμούμε.

Τότε, απομένει να τοποθετήσουμε το κάθε αντικείμενο στον «κόσμο» μας και η τοποθεσία του κάθε αντικειμένου θα είναι πάνω από το αντίστοιχο σύμβολο. Αυτό γίνεται δημιουργώντας για κάθε ένα Image Target, ένα Quad αντικείμενο και σε αυτό αναθέτουμε το αντικείμενο MyMaterial (σύροντας το στο Quad αντικείμενο).

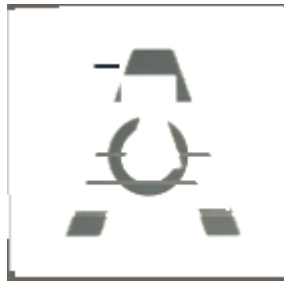
Επομένως μέχρι τώρα, υπάρχουν στον «κόσμο» μας η εικόνα-σύμβολο και το Quad αντικείμενο το οποίο προβάλλει την αντίστοιχη εικόνα-γράμμα της εικόνας - συμβόλου. Το αντικείμενο Quad που έχει δημιουργηθεί, έχει τις ίδιες συντεταγμένες στον κόσμο μας με το αντικείμενο Image Target, αφού αποτελεί «υπό-αντικείμενο» αυτού. Αυτό που απομένει για να γίνεται σωστά η εμφάνιση του αντικειμένου είναι να συμπίπτει η πρόσοψη των δύο αυτών αντικειμένων. Αυτό γίνεται γυρίζοντας το Quad αντικείμενο κατά 90 μοίρες έτσι ώστε η εμφάνιση του γράμματος να γίνεται ακριβώς πάνω από το σύμβολο.



Σχήμα 4.10: Image Target Σύμβολου Γράμματος Α



Σχήμα 4.11: Quad Αντικείμενο του Image Target του σύμβολου του γράμματος Α



Σχήμα 4.12: Το Image Target του σύμβολου του γράμματος Α μετά την εφαρμογή του Quad Αντικειμένου του

Με την ολοκλήρωση αυτών των βημάτων, η κάμερα της μηχανής Vuforia θα αναγνωρίζει ένα – ένα τα σύμβολα, αλλά δεν θα είναι σε θέση να αναγνωρίσει περισσότερα από ένα σύμβολα κάθε φορά, αφού η προκαθορισμένη επιλογή του αριθμού των αντικειμένων που μπορούν να αναγνωριστούν ταυτόχρονα είναι 1. Για να διορθωθεί αυτό αρκεί να αλλάξουμε τον αριθμό αυτό σε αυτό που επιθυμούμε εμείς, μέσω AR Camera → Open Vuforia Engine Configuration.

4.4.2 Αποθήκευση τελευταίας ιστορίας

Η αποθήκευση της τελευταίας ιστορίας γίνεται με την βοήθεια binary files. Η δημιουργία ενός ξεχωριστού δυαδικού αρχείου για την αποθήκευση πληροφοριών όχι μόνο καθιστά την αποθήκευση πληροφοριών σε κινητή συσκευή δυνατή, αλλά αυξάνει την ασφάλεια δεδομένων του χρήστη, αφού το αρχείο είναι κωδικοποιημένο. Στην παρούσα εργασία η διαδικασία έγινε δυνατή με τον εξής τρόπο:

Αρχική προϋπόθεση αποτελεί η ύπαρξη μίας κλάσης η οποία περιέχει τα στοιχεία τα οποία επιθυμούμε να αποθηκευτούν (σε αυτή την περίπτωση είναι η ιστορία). Έπειτα τα στοιχεία αυτά περνούν από έναν Binary Formatter, ο οποίος μετατρέπει τα στοιχεία

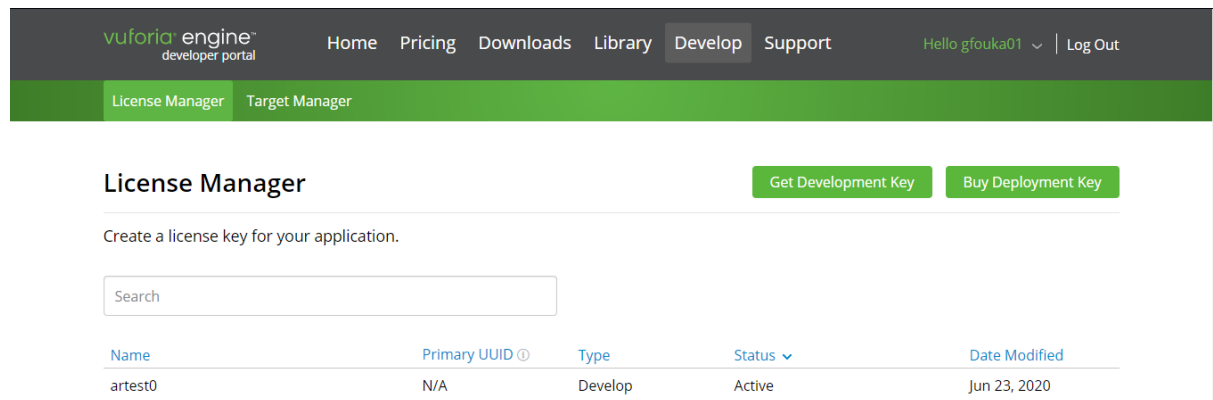
αυτά σε binary (0 και 1) και δημιουργούμε ένα αρχείο μέσω κώδικα, όπου τα στοιχεία αυτά (κωδικοποιημένα σε binary) θα αποθηκευτούν μέσω του Binary Formatter. Αυτό είναι το δυαδικό αρχείο.

Για να επαναφέρουμε τα στοιχεία στην κανονική τους μορφή, το αρχείο πρέπει να ανοιχτεί και ο Binary Formatter τα μετατρέπει από binary στον τύπο που είχαν, και τα επιστρέφει στην κλάση.

4.4.3 Βάση Δεδομένων

Οι εικόνες οι οποίες περιέχουν τα σύμβολα, εισάγονται στην βάση δεδομένων της μηχανής Vuforia, και έπειτα ακολουθεί η λήψη τους ως package στην συσκευή στην οποία αναπτύσσεται η εφαρμογή. Η διαδικασία έχει ως εξής:

Αρχικά ο χρήστης λαμβάνει ένα «development key» για την εφαρμογή του (βλ. Σχήμα 4.13).



The screenshot shows the 'vuforia engine developer portal' with a navigation bar including Home, Pricing, Downloads, Library, Develop, and Support. The 'Develop' tab is active. Below the navigation bar, there are links for 'License Manager' and 'Target Manager'. The 'License Manager' section has buttons for 'Get Development Key' and 'Buy Deployment Key'. It prompts the user to 'Create a license key for your application.' and includes a search bar. Below the search bar is a table with the following data:

Name	Primary UUID	Type	Status	Date Modified
artest0	N/A	Develop	Active	Jun 23, 2020

Σχήμα 4.13: Development Key

Αφού ο χρήστης δημιουργήσει το κλειδί του, το επόμενο βήμα είναι να δημιουργήσει την βάση δεδομένων του. Η βάση δεδομένων της παρούσας εφαρμογής είναι η «artest0» (βλ. Σχήμα 4.14).

vuforia engine[™]
developer portal

[Home](#)
[Pricing](#)
[Downloads](#)
[Library](#)
[Develop](#)
[Support](#)

Hello gfouka01
|
Log Out

[License Manager](#)
[Target Manager](#)

Target Manager

[Add Database](#)

Use the Target Manager to create and manage databases and targets.

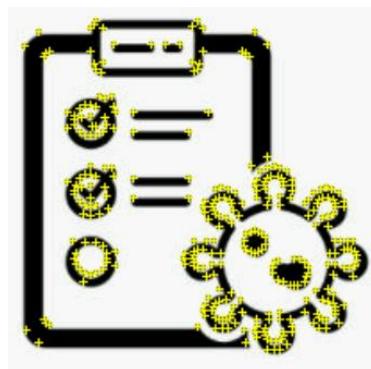
Search

Database	Type	Targets	Date Modified
artest0	Device	34	Dec 14, 2020

Σχήμα 4.14: Database

Έπειτα ο χρήστης πρέπει να προσθέσει στην βάση τις εικόνες που επιθυμεί και στην παρούσα εφαρμογή, επιλέχθηκαν εικόνες-σύμβολα από την ιστοσελίδα IconFinder [6]. Η κάθε εικόνα, βαθμολογείται αυτόματα από την βάση δεδομένων, βάσει των χαρακτηριστικών της (βλ. Σχήμα 4.15), και η βαθμολογία της εμφανίζεται κάτω από την στήλη «Rating». Οι εικόνες-σύμβολα που έχουμε χρησιμοποιήσει, επιλέχθηκαν με βασικό κριτήριο τη βαθμολογία τους, και όλες οι εικόνες έχουν βαθμολογία 4 στο ελάχιστο.

Η βαθμολογία μίας εικόνας προσδιορίζει τον βαθμό ευκολίας στον οποίο η εικόνα θα μπορεί να αναγνωριστεί. Ακόμη, οι εικόνες οι οποίες δέχεται η βάση, πρέπει να είναι τύπου .jpg ή .png με μέγιστο μέγεθος 2mb. Ο χρήστης δικαιούται να εισάγει εικόνες gray scale (8 bits) ή RGB (24 bits).



Σχήμα 4.15: Image Features

Target Manager > artest0

artest0 [Edit Name](#)

Type: Device

Targets (64)

Add Target

Download Database (All)

☐	Target Name	Type	Rating ①	Status ▾	Date Modified
☐	 60	Single Image	★★★★★	Active	Jan 05, 2021 15:38
☐	 59	Single Image	★★★★★	Active	Jan 05, 2021 15:35
☐	 58	Single Image	★★★★★	Active	Jan 05, 2021 15:34
☐	 57	Single Image	★★★★☆	Active	Jan 05, 2021 15:33

Σχήμα 4.16: Database

Όταν ο χρήστης επιθυμεί να προσθέσει μία καινούργια εικόνα στην βάση δεδομένων, πρέπει να επιλέξει την επιλογή «Add Target» και τότε του εμφανίζεται το παράθυρο που φαίνεται στο Σχήμα 4.17, στο οποίο ο χρήστης καλείται να επιλέξει την εικόνα που επιθυμεί από την συσκευή του, να διευκρινίσει το μέγεθος της εικόνας όταν βρίσκεται στο πραγματικό περιβάλλον, καθώς και το όνομα που θέλει να της αναθέσει.

Add Target

Type:

Single Image

Cuboid

Cylinder

3D Object

File:

Choose File

Browse...

.jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

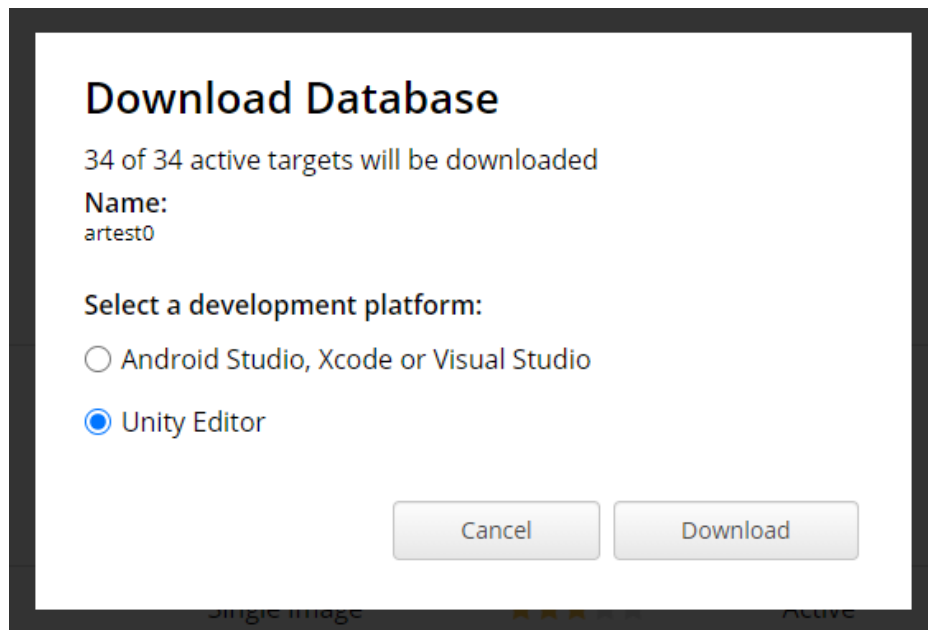
Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Cancel

Add

Σχήμα 4.17: Add Target

Όταν ο χρήστης τελειώσει με την εισαγωγή αντικειμένων στην βάση δεδομένων, απομένει η βάση αυτή να ενωθεί με το Unity Project. Για να γίνει αυτό, ο χρήστης πρέπει να κατεβάσει την βάση δεδομένων στην συσκευή του (βλ. Σχήμα 4.18) για την πλατφόρμα του Unity, πατώντας το κουμπί «Download Database (All)». Η βάση εμφανίζεται ως package στον υπολογιστή, και έπειτα ο χρήστης πρέπει να την προσθέσει στο Project.



Σχήμα 4.18: Download Database

Κεφάλαιο 5

Αξιολόγηση

5.1 Εισαγωγή	37
5.2 Διαδικασία Αξιολόγησης	37
5.3 Ερωτήσεις Ερωτηματολογίου	38
5.4 Αποτελέσματα Ερωτηματολογίου	39

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφεται και αναλύεται η αξιολόγηση της εφαρμογής, η οποία διεξήχθη μέσω ερωτηματολογίου από χρήστες οι οποίοι κατέβασαν και χρησιμοποίησαν την εφαρμογή. Το ερωτηματολόγιο εστάλη σε αρκετά άτομα με την βοήθεια μέσων κοινωνικής δικτύωσης. Το ερωτηματολόγιο ήταν διαθέσιμο για μια εβδομάδα, από τις 18 μέχρι τις 25 Δεκεμβρίου 2020 και απαντήθηκε από 25 χρήστες.

5.2 Διαδικασία Αξιολόγησης

Οι χρήστες κλήθηκαν να κατεβάσουν και να χρησιμοποιήσουν την εφαρμογή μέσω δικής τους κινητής συσκευής. Αυτό έγινε μέσω κοινοποίησης φακέλου στο Google Drive [11], ο οποίος φάκελος περιέχει:

- Το αρχείο apk, το οποίο ο χρήστης καλείται να εγκαταστήσει στην συσκευή του.
- Ένα pdf αρχείο το οποίο περιέχει ένα κωδικοποιημένο μήνυμα το οποίο ο χρήστης ζητείται να αποκωδικοποιήσει με την εφαρμογή.
- Ένα pdf αρχείο (Εγχειρίδιο Χρήσης), το οποίο επεξηγεί στον χρήστη την διαδικασία που πρέπει να ακολουθήσει για να κατεβάσει και να χρησιμοποιήσει σωστά την εφαρμογή.

- Το ερωτηματολόγιο αξιολόγησης το οποίο ο χρήστης καλείται να απαντήσει, αφού ολοκληρώσει την χρήση της εφαρμογής.

Το ερωτηματολόγιο στο οποίο βασιστήκαμε για την αξιολόγηση είναι το User Experience Questionnaire [15]. Αποτελείται από 26 ερωτήσεις, οι οποίες είναι σε μορφή ζεύγους αντιθέσεων (βλ. Σχήμα 5.1).

	1	2	3	4	5	6	7		
ενοχλητικό	☐	☐	☐	☐	☐	☐	☐	απολαυστικό	1
δυσνόητο	☐	☐	☐	☐	☐	☐	☐	κατανοητό	2
δημιουργικό	☐	☐	☐	☐	☐	☐	☐	αναποτελεσματικό	3
εύκολο στη μάθηση	☐	☐	☐	☐	☐	☐	☐	δύσκολο στη μάθηση	4
πολύτιμο	☐	☐	☐	☐	☐	☐	☐	υποδεέστερο	5
βαρετό	☐	☐	☐	☐	☐	☐	☐	συναρπαστικό	6
αδιάφορο	☐	☐	☐	☐	☐	☐	☐	ενδιαφέρον	7
απρόβλεπτο	☐	☐	☐	☐	☐	☐	☐	προβλέψιμο	8
γρήγορο	☐	☐	☐	☐	☐	☐	☐	αργό	9
εφευρετικό	☐	☐	☐	☐	☐	☐	☐	συμβατικό	10
παρελκυστικό	☐	☐	☐	☐	☐	☐	☐	υποστηρικτικό	11
καλό	☐	☐	☐	☐	☐	☐	☐	κακό	12
περίπλοκο	☐	☐	☐	☐	☐	☐	☐	εύκολο	13
αντιπαθητικό	☐	☐	☐	☐	☐	☐	☐	συμπαθητικό	14
συνηθισμένο	☐	☐	☐	☐	☐	☐	☐	πρωτοπόρο	15
δυσάρεστο	☐	☐	☐	☐	☐	☐	☐	ευχάριστο	16
ασφαλές	☐	☐	☐	☐	☐	☐	☐	ανασφαλές	17
ενθαρρυντικό	☐	☐	☐	☐	☐	☐	☐	αποθαρρυντικό	18
ανταποκρίνεται στις προσδοκίες	☐	☐	☐	☐	☐	☐	☐	δεν ανταποκρίνεται στις προσδοκίες	19
ανεπαρκές	☐	☐	☐	☐	☐	☐	☐	επαρκές	20
σαφές	☐	☐	☐	☐	☐	☐	☐	μπερδεμένο	21
μη πρακτικό	☐	☐	☐	☐	☐	☐	☐	πρακτικό	22
οργανωμένο	☐	☐	☐	☐	☐	☐	☐	ανοργάνωτο	23
ελκυστικό	☐	☐	☐	☐	☐	☐	☐	απωθητικό	24
φιλικό	☐	☐	☐	☐	☐	☐	☐	εχθρικό	25
συντηρητικό	☐	☐	☐	☐	☐	☐	☐	καινοτόμο	26

Σχήμα 5.1: Ερωτήσεις ερωτηματολογίου

5.3 Ερωτήσεις Ερωτηματολογίου

Το ερωτηματολόγιο που παρουσιάστηκε στους χρήστες αποτελείται από 30 ερωτήσεις, από τις οποίες οι 26 είναι αυτές που φαίνονται στο Σχήμα 5.1. Οι πρώτες 3 ερωτήσεις στο ερωτηματολόγιο που κλήθηκαν να απαντήσουν οι χρήστες αφορούν τον τρόπο χρήσης της εφαρμογής από τον κάθε χρήστη, καθώς και την ευκολία σάρωσης και κατανόησης των κωδικοποιημένων και αποκωδικοποιημένων φράσεων αντίστοιχα. Οι επόμενες 27 ερωτήσεις (από τις οποίες οι 26 είναι αυτές που φαίνονται στο Σχήμα 5.1) αφορούν την εμπειρία του χρήστη κατά την χρήση της εφαρμογής στην συσκευή του. Οι ερωτήσεις και η μορφή του ερωτηματολογίου παρουσιάζονται στο Παράρτημα Ζ.

5.4 Αποτελέσματα Ερωτηματολογίου

Από τις απαντήσεις του ερωτηματολογίου, παρατηρείται ότι 60% των χρηστών χρησιμοποίησαν υπολογιστή για την προβολή της κωδικοποιημένης φράσης, 28% χρησιμοποίησαν κινητή συσκευή και 12% εκτύπωσαν την κωδικοποιημένη φράση σε χαρτί.

Το 80% των χρηστών βρήκε την σάρωση της κωδικοποιημένης φράσης αρκετά εύκολη. Το 40% των χρηστών βαθμολόγησαν την αντίστοιχη ερώτηση με 6, ενώ το υπόλοιπο 40% αντιστοιχεί σε αυτούς που την βαθμολόγησαν με 5 (20%) και με 7 (20%). Το υπόλοιπο 20% των απαντήσεων των χρηστών αντιστοιχεί στα ποσοστά 8% στη βαθμολογία 2, 8% στη βαθμολογία 3 και 4% στη βαθμολογία 4.

Το 84% των χρηστών βρήκε την κατανόηση της αποκωδικοποιημένης φράσης αρκετά εύκολη. Το 44% των χρηστών βαθμολόγησαν την αντίστοιχη ερώτηση με 7, ενώ το υπόλοιπο 40% αντιστοιχεί σε αυτούς που την βαθμολόγησαν με 5 (12%) και με 6 (28%). Το υπόλοιπο 16% των απαντήσεων των χρηστών αντιστοιχεί στα ποσοστά 4% στη βαθμολογία 1, 8% στη βαθμολογία 3 και 4% στη βαθμολογία 4.

Το 72% των χρηστών βρήκε την εφαρμογή απολαυστική. Το 16% των χρηστών βαθμολόγησαν την αντίστοιχη ερώτηση με 7, το 32% με 6 και το 24% με 5. Το υπόλοιπο 28% αντιστοιχεί σε αυτούς που την βαθμολόγησαν με 4 (16%) και με 3 (12%). Το 60% των χρηστών βρήκαν την εφαρμογή απόλυτα κατανοητή, ενώ το υπόλοιπο 40% αντιστοιχεί σε αυτούς που την βαθμολόγησαν με 6 (24%) και με 5 (16%).

Η εφαρμογή, επίσης, φαίνεται να είναι αρκετά δημιουργική, αφού 24% των χρηστών την βαθμολόγησαν με 7, 36% με 6 και 30% με 5. Το υπόλοιπο 20% των απαντήσεων των χρηστών αντιστοιχεί στα ποσοστά 4% στη βαθμολογία 1, 4% στη βαθμολογία 2, 8% στη βαθμολογία 3 και 4% στη βαθμολογία 4.

Το 64% των χρηστών δήλωσαν ότι η εφαρμογή ήταν εντελώς εύκολη στην μάθηση, ενώ 24% βαθμολόγησαν την αντίστοιχη ερώτηση με 6, και 8% με 5. Στην ερώτηση 8, όπου οι χρήστες κρίνουν κατά πόσο η εφαρμογή είναι πολύτιμη, 20% βαθμολόγησαν την αντίστοιχη ερώτηση με 7, 12% με 6, 24% με 5, 32% με 4, 8% με 3 και 4% με 1. Το 72% των χρηστών βρήκε την εφαρμογή συναρπαστική καθώς βαθμολόγησε την αντίστοιχη ερώτηση από 5-7. Το υπόλοιπο 28% αντιστοιχεί σε αυτούς που την βαθμολόγησαν με 4 (16%), με 3 (8%) και με 1 (4%).

Το 84% των χρηστών βρήκε την εφαρμογή ενδιαφέρουσα καθώς βαθμολόγησε την αντίστοιχη ερώτηση από 5-7. Το υπόλοιπο 16% των απαντήσεων των χρηστών αντιστοιχεί στα ποσοστά 4% στις βαθμολογίες 1, 2, 3 και 4. Ακόμη, το 96% των χρηστών βρήκε την εφαρμογή γρήγορη, καθώς βαθμολόγησε την αντίστοιχη ερώτηση από 5-7. Το υπόλοιπο 4% αντιστοιχεί στην βαθμολογία 2.

Επίσης, οι περισσότεροι χρήστες βρήκαν την εφαρμογή εφευρετική. Το 20% των χρηστών βαθμολόγησαν την αντίστοιχη ερώτηση με 7, το 20% με 6, το 28% με 5, το 24% με 4, το 4% με 3 και το 4% με 2. Επιπλέον, οι περισσότεροι χρήστες βρήκαν την εφαρμογή υποστηρικτική, αφού το 72% των χρηστών βαθμολόγησαν την αντίστοιχη ερώτηση από 5-7.

Το 80% δήλωσαν ότι η εφαρμογή ήταν καλή, ενώ όλοι οι χρήστες δήλωσαν ότι η εφαρμογή ήταν εύκολη. Αυτό φαίνεται από το γεγονός ότι 80% των χρηστών βαθμολόγησαν την ερώτηση σχετικά με το πόσο καλή ήταν η εφαρμογή από 5 – 7, ενώ όλοι οι χρήστες βαθμολόγησαν την ερώτηση σχετικά με το πόσο εύκολη ήταν η εφαρμογή από 5 – 7.

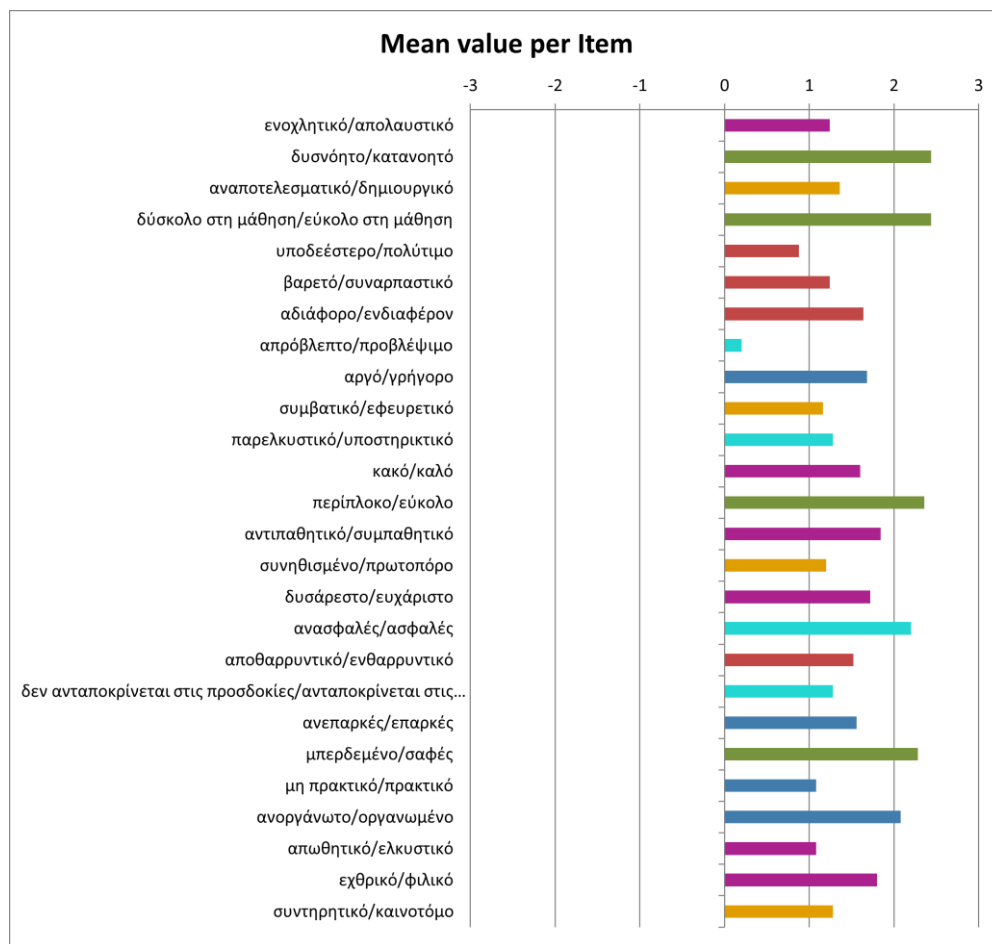
Το 84% δήλωσαν ότι η εφαρμογή ήταν συμπαθητική, ενώ το 72% δήλωσαν ότι η εφαρμογή ήταν πρωτόπορα. Αυτό φαίνεται από το γεγονός ότι 80% των χρηστών βαθμολόγησαν την ερώτηση σχετικά με το πόσο συμπαθητική ήταν η εφαρμογή από 5 – 7, ενώ 72% των χρηστών βαθμολόγησαν την ερώτηση σχετικά με το πόσο εύκολη ήταν η εφαρμογή από 5 – 7.

Η εφαρμογή περιγράφεται ως ευχάριστη και ασφαλής, καθώς το 84% βαθμολόγησαν την ερώτηση σχετικά με το πόσο ευχάριστη ήταν η εφαρμογή από 5 – 7 και 88% των χρηστών βαθμολόγησαν την ερώτηση σχετικά με το πόσο ασφαλής ήταν η εφαρμογή από 5 – 7.

Ακόμη, η εφαρμογή φαίνεται να είναι επαρκής και να ανταποκρίνεται στις προσδοκίες, αφού 76% βαθμολόγησαν τις αντίστοιχες ερωτήσεις από 5 -7. Επιπλέον, η εφαρμογή βαθμολογήθηκε ως σαφής, πρακτική και οργανωμένη, αφού το 84% των χρηστών την βαθμολόγησαν από 5 -7.

Παρ' όλα αυτά, από τα σχόλια των χρηστών, φαίνονται κάποια προβλήματα που συναντήθηκαν κατά την χρήση της εφαρμογής. 50% των σχολίων αναφέρουν την έλλειψη ενός «Back» κουμπιού στην οθόνη της AR κάμερας, το οποίο θα τους παρέπεμπε πίσω στο κυρίως μενού. Άλλα σχόλια επίσης αναφέρουν την εισαγωγή κάποιας ειδοποίησης στην οθόνη η οποία θα αναδεικνύει την επιτυχή λειτουργία του στιγμιότυπου οθόνης και την εμφάνιση του πληκτρολογίου σε άλλη μορφή. Ακόμη, παρουσιάστηκε σχόλιο το οποίο προτείνει τα στιγμιότυπα να αποθηκεύονται και στην συλλογή φωτογραφιών της κινητής συσκευής, και όχι μόνο στην γκαλερί της εφαρμογής.

Από τα αποτελέσματα της έρευνας, φαίνεται, γενικά, ότι η εφαρμογή που αναπτύχθηκε είναι ένα χρήσιμο εργαλείο για κωδικοποίηση και αποκωδικοποίηση φράσεων μέσω ασπρόμαυρων συμβόλων (βλ. Σχήμα 5.2). Παρ' όλο που, στα αποτελέσματα του ερωτηματολογίου φαίνονται μεικτές απαντήσεις, η γενική εικόνα των αποτελεσμάτων παραμένει θετική.



Σχήμα 5.2: Γραφική Ανάλυση Αποτελεσμάτων

Κεφάλαιο 6

Συμπεράσματα

6.1 Εισαγωγή	43
6.2 Τελικά Συμπεράσματα	43
6.3 Μελλοντική Εργασία	44

6.1 Εισαγωγή

Μετά την ολοκλήρωση της έρευνας στο Κεφάλαιο 1 και την μελέτη των διαφόρων τεχνολογιών που θα μπορούσαν να βοηθήσουν στην υλοποίηση της εφαρμογής με κάθε έναν από τους τρόπους αναγνώρισης εικόνας που αναφέρονται στο Κεφάλαιο 1, η υλοποίηση της εφαρμογής έχει ολοκληρωθεί. Η εφαρμογή είναι σε θέση να αποκωδικοποιήσει μία φράση ή μια ιστορία η οποία παρουσιάζεται με εικόνες σύμβολα, και να κωδικοποιήσει μία καινούργια ιστορία του χρήστη.

Σε αυτό το κεφάλαιο παρουσιάζονται τα τελικά συμπεράσματα της εργασίας, καθώς και το ενδεχόμενο συνέχειας της εργασίας, με βάση την παρούσα της εξέλιξη. Τα τελικά συμπεράσματα βασίζονται στα αποτελέσματα του ερωτηματολογίου από την χρήση της εφαρμογής.

6.2 Τελικά Συμπεράσματα

Μέσα από την αξιολόγηση της εφαρμογής, έχουν εξαχθεί κάποια συμπεράσματα για αυτήν.

Αρχικά, παρατηρείται ότι οι περισσότεροι χρήστες βρήκαν την εφαρμογή ευχάριστη και σαφή, γεγονός που συνεπάγεται ότι η οργάνωση και ο σκοπός της εφαρμογής ήταν ξεκάθαρα.

Σημαντικό κομμάτι για την εξασφάλιση της ευχάριστης αλληλεπίδρασης των χρηστών με την εφαρμογή, ήταν ο τρόπος με τον οποίο έγινε η σάρωση. Εξήχθη το συμπέρασμα ότι οι χρήστες οι οποίοι έκαναν την σάρωση με την κινητή τους συσκευή και ως δεύτερη οθόνη (η οποία προβάλλει την κωδικοποιημένη ιστορία) αυτή ενός υπολογιστή, δεν είχαν τόσο καλά αποτελέσματα όσο οι χρήστες που χρησιμοποίησαν ως δεύτερη οθόνη αυτή ενός δεύτερης κινητής συσκευής. Παρ' όλα αυτά, η σάρωση αποδείχτηκε να είναι βέλτιστη όταν ο χρήστης έκανε την προβολή της κωδικοποιημένης ιστορίας μέσω εκτύπωσης σε χαρτί, αφού η ποιότητα της εικόνας δεν αλλοιώνεται λόγω της επιπρόσθετης οθόνης.

Ακόμη, φάνηκε ότι η εφαρμογή θα μπορούσε να βελτιωθεί σε μερικά σημεία. Για παράδειγμα, το πληκτρολόγιο θα μπορούσε να υλοποιηθεί στοχεύοντας περισσότερο στην πρακτικότητα του προς τον χρήστη, δηλαδή να έχει λειτουργίες όπως την παρατεταμένη χρήση του κουμπιού «Delete», ή την εμφάνιση του σε «qwerty» μορφή. Οι περισσότεροι χρήστες ηλικιών 20-30 έδειξαν περισσότερο ενδιαφέρον και κατανόηση προς την εφαρμογή, με αποτέλεσμα την καλύτερη ανταπόκρισή τους στο ερωτηματολόγιο.

Παρ' όλα αυτά, η εφαρμογή αποτελεί παράδειγμα χρήσης της επαυξημένης πραγματικότητας, τομέας ο οποίος σήμερα αποτελεί σημαντικό και δημοφιλές κομμάτι στον κόσμο της Πληροφορικής.

6.3 Μελλοντική Εργασία

Η εφαρμογή θα μπορούσε να αναπτυχθεί περαιτέρω για σκοπούς ευχρηστίας και πρακτικότητας. Για παράδειγμα, η εισαγωγή χαρακτήρων θα μπορούσε να γίνεται δυναμικά, σε αντίθεση με την παρούσα εργασία η οποία γίνεται στατικά (αυτό φαίνεται από τον κώδικα). Δηλαδή, στην παρούσα εργασία ο κάθε χαρακτήρας από τους 66, έχει την θέση του ως εικόνα-σύμβολο στην οθόνη που θα παρουσιαστεί. Κατ' επέκταση

αυτής της βελτίωσης, το όριο χαρακτήρων της φράσης που εισάγει ο χρήστης, θα μπορούσε να είναι μεγαλύτερο από 66.

Επίσης, όπως ανέφεραν και κάποιοι χρήστες κατά την αξιολόγηση της εφαρμογής, θα μπορούσε να μελετηθεί περισσότερο το ενδεχόμενο της ύπαρξης ενός «Back» κουμπιού στην οθόνη με την AR κάμερα και η εισαγωγή κάποιας ένδειξης κατά την λήψη του στιγμιότυπου οθόνης.

Ακόμη, με περαιτέρω έρευνα, ίσως να ήταν βοηθητική η εύρεση διαφορετικών εικόνων-συμβόλων, τα οποία να είναι ευκολότερα στην αναγνώριση. Αυτό θα βοηθούσε στην γρηγορότερη σάρωση της ιστορίας και κατ' επέκταση στην ολική εμπειρία του χρήστη με την εφαρμογή.

Επιπλέον, παρ' όλο που στην παρούσα εργασία επιλέχθηκε η μέθοδος της Επαυξημένης Πραγματικότητας, μία παρόμοια εφαρμογή θα μπορούσε να υλοποιηθεί με κάποιο άλλο τρόπο αναγνώρισης εικόνας-συμβόλου σε εικόνα, για την οποία εφαρμογή η παρούσα εργασία θα αποτελούσε βάση.

Βιβλιογραφία

- [1] A Gentle Introduction to Object Recognition With Deep Learning,
<https://machinelearningmastery.com/object-recognition-with-deep-learning/>
- [2] Anuroop Katiyar, Karan Kalra and Chetan Garg, “Marker Based Augmented Reality”
- [3] Convolutional Neural Network,
<https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>
- [4] Getting Started with Vuforia Engine in Unity,
<https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html>
- [5] How to choose between a marker-based and markerless AR app,
<https://vr-tigers.com/blog/marker-based-vs-markerless-ar-apps/>
- [6] Iconfinder Website, <https://www.iconfinder.com/>
- [7] Influence of temporal delay and display update rate in an augmented reality application scenario
<https://dl.acm.org/doi/abs/10.1145/2836041.2836070>
- [8] Marker vs Markerless AR,
<https://appreal-vr.com/blog/markerless-vs-marker-based-augmented-reality/>
- [9] Mobile Augmented Reality
https://www.elsevier.com/_data/assets/pdf_file/0009/96966/Mobile_Augmented_Reality.pdf

- [10] Object Recognition 3 things you need to know,
<https://www.mathworks.com/solutions/image-video-processing/object-recognition.html>
- [11] The Google Drive Folder with the apk file and instructions for the application,
https://drive.google.com/drive/folders/16PSJV6h7iZQdPr0I20bubgL_sUkDLuIQ?usp=sharing
- [12] Σωκράτης Στρατής, «The “Recollecting Collective Memory” Project»
- [13] The Ultimate Guide to Markerless Augmented Reality,
<https://www.marxentlabs.com/what-is-markerless-augmented-reality-dead-reckoning/>
- [14] Tobias H. Höllerer, Steven K. Feiner, “Mobile Augmented Reality”
- [15] User Experience Questionnaire, <https://www.ueq-online.org/>
- [16] Visual Search Engine,
https://en.wikipedia.org/wiki/Visual_search_engine
- [17] Vuforia Developer Library, <https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html>
- [18] What Is Object Detection? 3 things you need to know,
<https://www.mathworks.com/discovery/object-detection.html>
- [19] Wikipedia Website, Augmented Reality,
https://en.wikipedia.org/wiki/Augmented_reality
- [20] Wikipedia Website, Template Matching,
https://en.wikipedia.org/wiki/Template_matching

Παράρτημα Α

Ακολουθεί ο κώδικας του αρχείου MainMenu.cs, ο οποίος αντιπροσωπεύει το κυρίως μενού της εφαρμογής.

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

public class MainMenu : MonoBehaviour
{
    public void Decode()
    {
        SceneManager.LoadScene("artest0");
    }

    public void Quit()
    {
        Application.Quit();
    }

    public void Encode()
    {
        SceneManager.LoadScene("encode");
    }

    public void HowToUse()
    {
        SceneManager.LoadScene("HowToUse");
    }
}
```

Παράρτημα Β

Ακολουθεί ο κώδικας του αρχείου ButtonText.cs, ο οποίος αντιπροσωπεύει την εμφάνιση των απομενόντων χαρακτήρων του χρήστη, καθώς και την αποθήκευση και την φόρτωση της ιστορίας του χρήστη.

```
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class ButtonText : MonoBehaviour
{
    public Text TextField;
    public Text RemainChar;
    public int characters = 66;

    public void SetText(string text)
    {
        if ((characters > 0)&&(characters < 67)){
            TextField.text = TextField.text + text;
            characters--;
        }
        RemainChar.text = "Απομένουν " +characters+ " χαρακτήρες";
    }

    public void RemoveText(string text)
    {
        TextField.text = TextField.text.Remove(TextField.text.Length - 1);
        if(characters<66)
            characters++;
        RemainChar.text = "Απομένουν " + characters + " χαρακτήρες";
    }
}
```



```

public void SaveStory()
{
    SaveSystem.SaveUser(this);
}

public void LoadStory()
{
    UserData data = SaveSystem.LoadUser();
    TextField.text = data.story;

    characters = 66-TextField.text.Length;
    RemainChar.text = "Απομένουν " + characters + " χαρακτήρες";

}

public void Submit()
{
    SceneManager.LoadScene("encoded_message");
}
}

```

Παράρτημα Γ

Ακολουθεί ο κώδικας του αρχείου SaveSystem.cs, όπου γίνεται η δημιουργία του δυαδικού αρχείου και η αποθήκευση της τελευταίας ιστορίας του χρήστη σε binary σε αυτό.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;

public static class SaveSystem
{
    public static void SaveUser(ButtonText textstory)
    {
        BinaryFormatter formatter = new BinaryFormatter();

        string path = Application.persistentDataPath + "/user.fun";

        FileStream stream = new FileStream(path, FileMode.Create);

        UserData data = new UserData(textstory);

        formatter.Serialize(stream, data);
        stream.Close();
    }

    public static UserData LoadUser()
    {

```

```

string path = Application.persistentDataPath + "/user.fun";

if (File.Exists(path))
{

    BinaryFormatter formatter = new BinaryFormatter();
    FileStream stream = new FileStream(path, FileMode.Open);

    UserData data = formatter.Deserialize(stream) as UserData;
    stream.Close();

    return data;

}

else
{
    Debug.LogError("Save File not found in" + path);
    return null;
}

}
}

```

Παράρτημα Δ

Ακολουθεί ο κώδικας του αρχείου LoadTexture.cs, όπου γίνεται η κωδικοποίηση από γράμμα σε εικόνα και η εμφάνιση των εικόνων στην οθόνη του χρήστη, κατά την λειτουργία «Κωδικοποίηση Ιστορίας».

```
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class LoadTexture : MonoBehaviour
{
    Texture2D myTexture;
    GameObject rawImage;
    bool b=true;
    public GameObject img;

    public void Start()
    {

        UserData data = SaveSystem.LoadUser();

        char[] arr = data.story.ToCharArray();

        string str;

        for (int i = 0; i < arr.Length; i++)
        {
            str = i.ToString();
            rawImage = GameObject.Find(str);
```

```

switch (arr[i])
{
    case 'A':
        {
            myTexture = Resources.Load("Images/53") as Texture2D;
            rawImage.GetComponent<RawImage>().texture = myTexture;
            break;
        }
    case 'B':
        {
            myTexture = Resources.Load("Images/32") as Texture2D;
            rawImage.GetComponent<RawImage>().texture = myTexture;

            break;
        }
    .....
}
}
public void Quit()
{
    Application.Quit();
}

public void Back()
{
    SceneManager.LoadScene("encode");
}

public void NextScene()
{
    SceneManager.LoadScene("gallery");
}
}

```

Παράρτημα Ε

Ακολουθεί ο κώδικας του αρχείου TakeScreenshot.cs, όπου γίνεται η λήψη του στιγμιότυπου οθόνης.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class TakeScreenshot : MonoBehaviour
{
    [SerializeField]

    public void TakeAShot()
    {
        StartCoroutine("CaptureIt");
    }

    IEnumerator CaptureIt()
    {
        string timeStamp = System.DateTime.Now.ToString("dd-MM-yyyy-HH-mm-ss");
        string fileName = "Screenshot" + timeStamp + ".png";
        string pathToSave = fileName;

        ScreenCapture.CaptureScreenshot(pathToSave);

        yield return new WaitForEndOfFrame();
    }
}
```

Παράρτημα ΣΤ

Ακολουθεί ο κώδικας του αρχείου ScreenshotPreview.cs, όπου γίνεται η προβολή του στιγμιότυπου οθόνης στην γκαλερί και η υπόλοιπη υλοποίηση της οθόνης γκαλερί.

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

public class ScreenShotPreview : MonoBehaviour
{
    [SerializeField]
    GameObject can;
    string[] files = { "ONE", "TWO" };
    int whichScreenShotIsShown = 0;

    // Start is called before the first frame update
    void Start()
    {
        files = Directory.GetFiles(Application.persistentDataPath + "/", "*.png");

        if (files.Length > 0)
        {
            GetPictureAndShowIt();
        }
    }
}
```

```

void GetPictureAndShowIt()
{

    string pathToFile = files[whichScreenShotIsShown];
    Texture2D texture = GetScreenshotImage(pathToFile);
    Sprite sp = Sprite.Create(texture, new Rect(0, 0, texture.width, texture.height), new
Vector2(0.5f, 0.5f));
    can.GetComponent<Image>().sprite = sp;
}

Texture2D GetScreenshotImage(string filePath)
{

    Texture2D texture = null;
    byte[] fileBytes;
    if (File.Exists(filePath))
    {
        fileBytes = File.ReadAllBytes(filePath);
        texture = new Texture2D(2, 2, TextureFormat.RGB24, false);
        texture.LoadImage(fileBytes);
    }
    return texture;
}

public void NextPicture()
{
    if (files.Length > 0)
    {
        whichScreenShotIsShown += 1;
        if (whichScreenShotIsShown > files.Length - 1)
            whichScreenShotIsShown = 0;
        GetPictureAndShowIt();
    }
}

```



```
}

public void PreviousPicture()
{
    if (files.Length > 0)
    {
        whichScreenShotIsShown -= 1;
        if (whichScreenShotIsShown < 0)
            whichScreenShotIsShown = files.Length - 1;
        GetPictureAndShowIt();
    }
}

}
```

Παράρτημα Z

Ακολουθούν οι ερωτήσεις του ερωτηματολογίου, καθώς και η μορφή τους.

1. Έχετε κατεβάσει το αρχείο που περιέχει την κωδικοποιημένη φράση σε συσκευή η οποία ήταν:

- Υπολογιστής
- Κινητή Συσκευή
- Other:

2. Βρήκατε την σάρωση της φράσης

Πολύ δύσκολη 1 2 3 4 5 6 7 Πολύ εύκολη

3. Βρήκατε την κατανόηση της φράσης

Πολύ δύσκολη 1 2 3 4 5 6 7 Πολύ εύκολη

4. Βρήκατε την εφαρμογή

Ενοχλητική 1 2 3 4 5 6 7 Απολαυστική

5. Βρήκατε την εφαρμογή

Δυσνόητη 1 2 3 4 5 6 7 Κατανοητή

6. Βρήκατε την εφαρμογή

Αναποτελεσματική 1 2 3 4 5 6 7 Δημιουργική

7. Βρήκατε την εφαρμογή

Δύσκολη στη μάθηση 1 2 3 4 5 6 7 Εύκολη στη μάθηση

8. Βρήκατε την εφαρμογή

Υποδεέστερη 1 2 3 4 5 6 7 Πολύτιμη

9. Βρήκατε την εφαρμογή

Βαρετή 1 2 3 4 5 6 7 Συναρπαστική

10. Βρήκατε την εφαρμογή

Αδιάφορη 1 2 3 4 5 6 7 Ενδιαφέρον

11. Βρήκατε την εφαρμογή

Προβλέψιμη 1 2 3 4 5 6 7 Απρόβλεπτη

12. Βρήκατε την εφαρμογή

Αργή 1 2 3 4 5 6 7 Γρήγορη

13. Βρήκατε την εφαρμογή

Συμβατική 1 2 3 4 5 6 7 Εφευρετική

14. Βρήκατε την εφαρμογή

Παρελκυστική 1 2 3 4 5 6 7 Υποστηρικτική

15. Βρήκατε την εφαρμογή

Κακή 1 2 3 4 5 6 7 Καλή

16. Βρήκατε την εφαρμογή

Περίπλοκη 1 2 3 4 5 6 7 Εύκολη

17. Βρήκατε την εφαρμογή

Αντιπαθητική 1 2 3 4 5 6 7 Συμπαθητική

18. Βρήκατε την εφαρμογή

Συνηθισμένη 1 2 3 4 5 6 7 Πρωτοπόρα

19. Βρήκατε την εφαρμογή

Δυσάρεστη 1 2 3 4 5 6 7 Ευχάριστη

20. Βρήκατε την εφαρμογή

Ανασφαλής 1 2 3 4 5 6 7 Ασφαλής

21. Βρήκατε την εφαρμογή

Αποθαρρυντική 1 2 3 4 5 6 7 Ενθαρρυντική

22. Βρήκατε την εφαρμογή

Δεν ανταποκρίνεται στις προσδοκίες 1 2 3 4 5 6 7 Ανταποκρίνεται στις
προσδοκίες

23. Βρήκατε την εφαρμογή

Ανεπαρκής 1 2 3 4 5 6 7 Επαρκής

24. Βρήκατε την εφαρμογή

Μπερδεμένη 1 2 3 4 5 6 7 Σαφής

25. Βρήκατε την εφαρμογή

Μη πρακτική 1 2 3 4 5 6 7 Πρακτική

26. Βρήκατε την εφαρμογή

Ανοργάνωτη 1 2 3 4 5 6 7 Οργανωμένη

27. Βρήκατε την εφαρμογή

Αποθητική 1 2 3 4 5 6 7 Ελκυστική

28. Βρήκατε την εφαρμογή

Εχθρική 1 2 3 4 5 6 7 Φιλική

29. Βρήκατε την εφαρμογή

Συντηρητική 1 2 3 4 5 6 7 Καινοτόμα

30. Έχετε κάτι να προσθέσετε σχετικά με την χρήση της εφαρμογής;

.....