

Εθνικό Μετσοβίο Πολγτεχνείο

Σχολή Ηλεκτρολογών Μηχανικών και Μηχανικών Υπολογιστών Τομέας Επικοινώνιων, Ηλεκτρονικής και Σύστηματών Πληροφορικής

«ΧΡΗΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΜΕС ΓΙΑ ΕΦΑΡΜΟΓΕΣ AUGMENTED REALITY»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΜΜΑΝΟΥΗΛ ΡΕΜΟΥΝΤΑΚΗΣ

Επιβλέπων: Παναγιώτης Γ. Κωττής

Καθηγητής Ε.Μ.Π

Αθήνα, Μάρτιος 2021



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ Σχολή Ηλεκτρολογών Μηχανικών και Μηχανικών Υπολογιστών Τομέας Επικοινώνιων, Ηλεκτρονικής και Σύστηματών Πληροφορικής

«ΧΡΗΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΜΕС ΓΙΑ

ΕΦΑΡΜΟΓΕΣ AUGMENTED REALITY»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΜΜΑΝΟΥΗΛ ΡΕΜΟΥΝΤΑΚΗΣ

Επιβλέπων: Παναγιώτης Γ. Κωττής Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 8^η Μαρτίου 2021

Π. ΚωττήςΧ. ΚαψάληςΓ. ΦικιώρηςΚαθηγητής Ε.Μ.Π.Καθηγητής Ε.Μ.Π.Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....

Εμμανουήλ Ρεμουντάκης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός & Μηχανικός Υπολογιστών Ε.Μ.Π

© 2020 – All rights reserved



Εθνικό Μετσόβιο Πολυτεχνείο Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

Copyright ©-All rights reserved Ρεμουντάκης Εμμανουήλ, 2021.

Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

ΠΕΡΙΛΗΨΗ

Η Εθνική Πινακοθήκη ιδρύθηκε το 1900 και σήμερα περικλείει στις συλλογές της περισσότερα από 20.000 έργα ζωγραφικής και αποτελεί το θησαυροφυλάκιο της νεότερης ελληνικής καλλιτεχνικής δημιουργίας. Την 25^η Μάρτιου 2021 αναμένονται τα εγκαίνια του νέου κτηρίου που θα στεγάσει όλα της τα εκθέματα. Οι νέοι αυτοί χώροι σε συνδυασμό με τα έργα τέχνης αποτελούν μια ανεπανάληπτη εμπειρία για τον κάθε επισκέπτη.

Στόχος της παρούσας διπλωματικής εργασίας είναι η βελτίωση αυτής της εμπειρίας με τη δημιουργία μιας εφαρμογής για κινητά τηλεφωνά (Android και iOS), χρησιμοποιώντας τεχνολογίες Augmented Reality (AR). Η εφαρμογή είναι ικανή να αναγνωρίσει οποιοδήποτε πίνακα βρίσκεται εντός της Εθνικής Πινακοθήκης και να παρέχει με τεχνολογίες AR πληροφορίες στον επισκέπτη για το έργο που βλέπει εκείνη τη στιγμή σε πραγματικό χρόνο. Όλα αυτά επιτυγχάνονται με τη χρήση δικτύων MEC υψηλής ταχύτητας.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:

Mobile Edge Computing (MEC), Επαυξημένη πραγματικότητα, Vuforia, Unity, Python, Java, Vuforia cloud database, C Sharp, Εθνική Πινακοθήκη, Android, iOS

7

.

ABSTRACT

The National Gallery was founded in 1900 and today includes in its collections more than 20,000 paintings and is the treasury of modern Greek art. The inauguration of the new building that will house all its exhibits is expected on March 25, 2021. These new spaces in combination with the works of art are a unique experience for every visitor.

The aim of this dissertation is to enhance this experience by creating a mobile application (Android and iOS), using Augmented Reality (AR) technologies. The application is able to identify any painting inside the National Gallery and provide information with AR technologies to the visitor about the work he is currently seeing, in real time. All this is achieved by using high speed MEC networks.

KEY WORDS:

Mobile Edge Computing (MEC), Augmented Reality (AR), Vuforia, Unity, Python, Java, Vuforia cloud database, C Sharp, National gallery, Android, iOS

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της διπλωματικής εργασίας κ. Παναγιώτη Κωττή για την υποστήριξη και την καθοδήγηση που μου προσέφερε, καθώς και για την ευκαιρία που μου έδωσε να εκπονήσω την διπλωματική μου στο συγκεκριμένο θέμα. Η καθοδήγησή, οι επισημάνσεις και οι διορθώσεις του συνέβαλαν καθοριστικά στο τελικό αποτέλεσμα. Καταλήγοντας, θα ήθελα να ευχαριστήσω τους γονείς μου, που ήταν πάντα δίπλα μου και με υποστήριζαν.

Κατάλογος Σχημάτων

2χημα 1.1 10 συνεχες εικονικοτητάς – πραγματικοτητάς	25
Σχήμα 1.2 Διάγραμμα απλού AR συστήματος	26
Σχήμα 2.1 Αρχιτεκτονική LTE	32
Σχήμα 2.2 Αριχτεκτονική E-UTRAN	33
Σχήμα 2.3 Αρχιτεκτονική ΕΡC	34
Σχήμα 2.4 Επισκόπηση συστήματος ΜΕC	35
Σχήμα 2.5 5G Εφαρμογές	38
Σχήμα 2.6 Ένταξη του MEC στο δίκτυο 5G	39
Σχήμα 2.7 Τεχνολογία ΜΕC στις ΑR εφαρμογές	41
Σχήμα 3.1 Τοπολογίες δικτύων Wi-Fi και MEC	43
Σχήμα 3.2 Λειτουργίες πελάτη και εξυπηρετητή	43
Σχήμα 3.3 Αρχιτεκτονική MEC για τις AR εφαρμογές	44
Σχήμα 3.4 Κατανομή εργασιών στο ΜΕC εξυπηρετητή	45
Σχήμα 3.5 Εφαρμογή AR σε εκθέματα μουσείων	46
Σχήμα 4.1 Επίσκεψη σε χώρο με αξιοθέατα	49
Σχήμα 4.2 Επίσκεψη στην Εθνική Πινακοθήκη με χρήση εφαρμογής	50
Σχήμα 4.3 Νέο κτήριο Εθνικής Πινακοθήκης	52
Σχήμα 4.4 QR code για τη λήψη της εφαρμογής	54
Σχήμα 4.5 Ιστοσελίδα λήψης της εφαρμογής	55
Σχήμα 5.1 Αναγνώριση στόχου εικόνας	58
Σχήμα 5.2 Αναγνώριση τρισδιάστατου μοντέλου	58
Σχήμα 5.3 Στόχος επίπεδης επιφάνειας	59
Σχήμα 5.4 Στόχος τύπου VuMark	60
Σχήμα 6.1 Λογότυπο ARt εφαρμογής	69
Σχήμα 6.2 Ερωτηματολόγιο διεπαφής χρήστη ARt	71
Σχήμα 6.3 Αποτελέσματα βαθμολογιών ανά σχεδιασμό διεπαφής χρήστη	72
Σχήμα 6.4 Τελικά αποτελέσματα βαθμολογιών κατά την οριστικοποίηση διεπαφής	73
Σχήμα 6.5 Εικονίδια εφαρμογής ARt	73
Σχήμα 7.1 Αρχική οθόνη	75
Σχήμα 7.2 Οθόνη αναμονής	76
Σχήμα 7.3 Οθόνη σάρωσης	78

Σχήμα 7.4 Δισδιάστατο πάνελ, κατόπιν σάρωσης του πίνακα "Figures in Space n.1"	79
Σχήμα 8.1 Τοπικό δίκτυο ΜΕC - Ροή της πληροφορίας	86
Σχήμα 8.2 Υπολογισμός εκτάσεως Εθνικής Πινακοθήκης	87
Σχήμα 8.3 Πακέτα χρέωσης της Vuforia Engine	88
Σχήμα 9.1 Επίσημη ιστοσελίδα Εθνικής Πινακοθήκης	91
Σχήμα 9.2 Κατηγορίες έργων τέχνης Εθνικής Πινακοθήκης	92
Σχήμα 9.3 Έργα Εθνικής Πινακοθήκης (Ηλεκτρονική συλλογή)	92
Σχήμα 9.4 Παράδειγμα ιστοσελίδας πίνακα του επίσημου ιστότοπου της Εθνικής	
Πινακοθήκης	93
Σχήμα 9.5 Κώδικας συλλογής δεδομένων	98
Σχήμα 9.6 Τελικά δεδομένα μετά την ολοκλήρωση εκτέλεσης του κώδικα Python	99
Σχήμα 9.7 Οι φωτογραφίες των πινάκων εντός του φακέλου "Image targets"	100
Σχήμα 9.8 Όλες οι πληροφορίες για κάθε πίνακα εντός του φακέλου "Metadata"	100
Σχήμα 9.9 Παράδειγμα αρχείου Metadata	101
Σχήμα 9.10 Μέρος από το αρχείο "painting_urls.txt"	101
Σχήμα 9.11 Μέρος του αρχείου "paintings_width.txt"	102
Σχήμα 9.12 Περιεχόμενα του αρχείου "paintings_with_exception.txt"	102
Σχήμα 9.13 Κώδικας ανάλυσης διαστάσεων πίνακα	103
Σχήμα 9.14 Σχηματική απεικόνιση μιας Vuforia Cloud Database	105
Σχήμα 9.15 Σχηματική απεικόνιση του τρόπου λειτουργίας της Vuforia Cloud Database	e 106
Σχήμα 9.16 Σημεία αναγνώρισης πίνακα	107
Σχήμα 9.17 Βαθμολογία πινάκων στο πάνελ της Vuforia Cloud Database	108
Σχήμα 9.18 Χαρακτηριστικά στόχου	109
Σχήμα 9.19 Παράδειγμα επεξεργασίας εικόνας για επίτευξη καλύτερης βαθμολογίας	111
Σχήμα 9.20 Αρχική σελίδα της Vuforia Engine	114
Σχήμα 9.21 Φόρμα δημιουργίας λογαριασμού Vuforia	115
Σχήμα 9.22 Είσοδος στην πλατφόρμα της Vuforia Engine	116
Σχήμα 9.23 Vuforia Get Development Key	117
Σχήμα 9.24 Δημιουργία Vuforia Development License Key	117
Σχήμα 9.25 Καρτέλα "License Manager" της Vuforia	118
Σχήμα 9.26 Καρτέλα "License Key" της αδείας χρήσης "Art_Development_License_K	ley"
	119
Σχήμα 9.27 Καρτέλα "Usage" της αδείας χρήσης "Art_Development_License_Key"	120
Σχήμα 9.28 Vuforia Add Database	120

Σχήμα 9.29 Δημιουργία Vuforia Cloud Database	121
Σχήμα 9.30 Καρτέλα "Target Manager" της Vuforia	
Σχήμα 9.31 "ARt_Cloud_Database" πάνελ	
Σχήμα 9.32 Εισαγωγή Image target στην Cloud Database μέσω της π	τλατφόρμας Vuforia124
Σχήμα 9.33 Ποσοστιαία μπάρα κατά τη δημιουργία νέου Image targ	get125
Σχήμα 9.34 Καρτέλα Vuforia Cloud Database	
Σχήμα 9.35 Κλειδιά πρόσβασης σε Vuforia Cloud Database	
Σχήμα 9.36 Αρχείο CreateCloudDatabase.java	131
Σχήμα 9.37 Αρχείο PostNewTarget.java	134
Σχήμα 9.38 Παράδειγμα http post request	135
Σχήμα 9.39 Παράδειγμα response	135
Σχήμα 9.40 Μέρος από Vuforia Cloud Database με τους πίνακες της	; πινακοθήκης136
Σχήμα 9.41 Αρχεία java για τη διαχείριση της Vuforia Cloud Databa	se με χρήση Vuforia
Web Services	137
Σχήμα 9.42 Βιβλιοθήκες που απαιτούνται για την εκτέλεση του Vufe	oria Web Service137
Σχήμα 9.43 Λήψη Unity Hub	
Σχήμα 9.44 Unity Hub Professional License	139
Σχήμα 9.45 Εγκατεστημένη Έκδοση Unity: 2020.1.11f1	139
Σχήμα 9.46 Unity asset store Vuforia package	140
Σχήμα 9.47 Unity Package Manager: Vuforia Engine	141
Σχήμα 9.48 Εργαλεία πακέτου Vuforia Engine	142
Σχήμα 9.49 Game Object δέντρου ως Prefab	144
Σχήμα 9.50 Unity Engine: Διεπαφή χρήστη (UI)	145
Σχήμα 9.51 Καρτέλα σκηνής Unity	146
Σχήμα 9.52 Καρτέλα ιεραρχίας Unity	147
Σχήμα 9.53 Καρτέλα Inspector Unity	148
Σχήμα 9.54 Καρτέλα Έργου (Project) Unity	149
Σχήμα 9.55 Καρτέλα κονσόλας Unity	149
Σχήμα 9.56 Γραμμή εργαλείων Unity	151
Σχήμα 9.57 Σκηνές που υλοποιούν την εφαρμογή στο Unity	
Σχήμα 9.58 Καρτέλα Ιεραρχίας της σκηνής "Splash"	
Σχήμα 9.59 Καρτέλα σκηνής "Splash"	154
Σχήμα 9.60 Κώδικας σκηνής "Splash"	155
Σχήμα 9.61 Καρτέλα ιεραρχίας σκηνής "Loading"	

Σχήμα 9.62 Καρτέλα σκηνής "Loading"	157
Σχήμα 9.63 Κώδικας σκηνής "Loading"	159
Σχήμα 9.64 Καρτέλα ιεραρχίας σκηνής "About"	160
Σχήμα 9.65 Κώδικας αρχείου "AboutManager.cs"	161
Σχήμα 9.66 Κώδικας αρχείου "AboutScreenInfo.cs"	164
Σχήμα 9.67 Καρτέλα σκηνής "About"	166
Σχήμα 9.68 Κώδικας αρχείου SafeAreaManager.cs	170
Σχήμα 9.69 Καρτέλα ιεραρχίας σκηνής "Paintings"	171
Σχήμα 9.70 Καρτέλα σκηνής "Paintings" Game Object	172
Σχήμα 9.71 Καρτέλα Inspector του "ARCamera" Game Object	173
Σχήμα 9.72 ARCamera (Vuforia configuration)	174
Σχήμα 9.73 Κώδικας αρχείου SceneOrientation.cs	175
Σχήμα 9.74 Κώδικας αρχείου AnimationsManager.cs	178
Σχήμα 9.75 Καρτέλα σκηνής "ScanLine" Game Object	179
Σχήμα 9.76 Κώδικας αρχείου CloudRecoScanLine.cs	181
Σχήμα 9.77 Καρτέλα Inspector του "CloudRecognition" Game Object	182
Σχήμα 9.78 Κώδικας αρχείου CloudRecoEventHandler.cs	185
Σχήμα 9.79 Κώδικας αρχείου CloudErrorHandler.cs	188
Σχήμα 9.80 Cloud icon	189
Σχήμα 9.81 Cloud Recognition GameObject	189
Σχήμα 9.82 Κώδικας αρχείου CloudErrorHandler.cs	191
Σχήμα 9.83 Καρτέλα Inspector του "CloudContentManager" Game Object	192
Σχήμα 9.84 Κώδικας αρχείου CloudContentManager.cs	195
Σχήμα 9.85 Ιεραρχία αντικειμένων σύνθεσης πάνελ	196
Σχήμα 9.86 Καρτέλα σκηνής "Canvas" Game Object	198
Σχήμα 9.87 Κώδικας αρχείου TapHandler.cs	201
Σχήμα 9.88 Κώδικας αρχείου NavigationHandler.cs	202
Σχήμα 9.89 Ιεραρχία αντικειμένων σύνθεσης μπάρας ενημέρωσης χρήστη	203
Σχήμα 9.90 Καρτέλα σκηνής "TitleBar" Game Object	204
Σχήμα 9.91 Unity Engine Build Settings	206
Σχήμα 10.1 Σενάριο εικονικής χρήσης της εφαρμογής ARt στο νέο κτήριο της Εθνική	IS
Πινακοθήκης	209
Σχήμα 10.2 Λήψη οδηγιών για την εύρεση εκθέματος με χρήση τεχνολογιών επαυξημ	ιένης
πραγματικότητας	210

Κατάλογος Πινάκων

Πίνακας 1.1 Σύγκριση διαφορετικών τύπων ασύρματων δικτύων για MAR εφαρμογές	28
Πίνακας 5.1 Εργαλεία ανάπτυξης εφαρμογής	56
Πίνακας 8.1 Κόστη εξοπλισμού και δικτύου	88
Πίνακας 8.2 Χρόνος εγκατάστασης	89
Πίνακας 9.1 Γραμμή εργαλείων Unity Engine	150
Πίνακας 10.1 Χρήσιμοι σύνδεσμοι	.208

Κατάλογος Συντμήσεων

AR	Augmented Reality
AV	Augmented Virtuality
CN	Core Network
CUPS	Control Plane User Plane Separation
eNB	evolved Node B
eMBB	Enhanced Mobile broadband
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
GUI	Graphic User Interface
HMD	Head Mounted Display
НТТР	Hypertext Transfer Protocol
HUD	Head Up Display
ІоТ	Internet of Things
IP	Internet Protocol
LAN	Local Area Network
LTE	Long Term Evolution
MAR	Mobile Augmented Reality
ME	Mobile Edge
MEC	Mobile Edge Computing
MEO	Mobile Edge Orchestrator
MME	Mobility Management Entity
NG-RAN	Next Generation Radio Access Network
PDA	Personal Digital Assistant
RAN	Radio Access Network
SDK	Software Development Kit
S-GW	Serving Gateway
UE	User equipment
UI	User Interface
UPF	User Plane Function
URL	Uniform Resource Locator
URLLC	Ultra-reliable low latency Communications

VR	Virtual Reality
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WWAN	Wireless Wide Area Network
3GPP	3rd Generation Partnership Project

Πίνακας περιεχομένων

Π	ΕΡΙΛΗ	ΙΨΗ	6
A	BSTR	ACT	8
E	υχαρισ	τίες	10
K	ατάλο	γος Σχημάτων	12
K	ατάλο	γος Συντμήσεων	18
1	ΕΙΣ	ΑΓΩΓΗ – ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ	24
	1.1	Εισαγωγή στην επαυξημένη πραγματικότητα	24
	1.1.	1 Μικτή Πραγματικότητα	24
	1.2	Χαρακτηριστικά συστήματος AR	25
	1.3	Συσκευές απεικόνισης	26
	1.4	Ασύρματη συνδεσιμότητα για τις AR εφαρμογές	27
	1.4.	1 Wireless Wide Area Network (WWAN)	27
	1.4.	2 Wireless Local Area Network (WLAN)	28
	1.4.	3 Wireless Personal Area Network (WPAN)	28
	1.5	Εφαρμογές του AR	29
2	ΗT	ΈΧΝΟΛΟΓΙΑ ΜΕC ΣΤΗΝ ΥΠΗΡΕΣΙΑ ΤΩΝ AR ΕΦΑΡΜΟΓΩΝ	32
	2.1	Τεχνολογία LTE	32
	2.1.	1 Αρχιτεκτονική LTE	32
	2.2	Mobile Edge Computing	35
	2.3	Πλεονεκτήματα τεχνολογίας ΜΕC	36
	2.4	Περί δικτύων 5G	37
	2.5	Εφαρμογή της τεχνολογίας MEC στα δίκτυα 5G	38
	2.6	Η τεχνολογία MEC στις AR εφαρμογές	40
	2.7	Επισκόπηση του ΜΕC συστήματος	40
	A T A		40
3	$\Delta I A$	$\Delta I \Delta I \Delta I O A O I H \Delta H \Delta Y \Delta I H M A I \Omega N E \Xi Y II H P E I H \Delta H \Delta A K E \Psi A P M O I \Omega N$	42

3.1.1	Επικοινωνία με το δίκτυο	42
3.2	Επίπεδα ΜΕC στις AR εφαρμογές	44
3.2.1	Ι Σενάριο εξυπηρέτησης AR εφαρμογών σε σύστημα MEC	45
3.3	AR εφαρμογές σε χώρους μουσείων	46
4 OPE	ΣΜΟΣ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΛΥΣΗ	48
4.1	Ορισμός προβλήματος	48
4.2	Λύση	49
4.3	Εθνική Πινακοθήκη	51
4.4	Ανάλυση απαιτήσεων	53
5 ЕРГ	ΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΗΣ	56
5.1	Πλατφόρμες	57
5.1.1	Vuforia Engine	57
5.1.2	2 Unity	60
5.1.3	3 Unity assets store	61
5.1.4	Adobe Photoshop	62
5.1.5	5 Marvel App	63
5.2	Γλώσσες προγραμματισμού	63
5.2.1	Python	63
5.2.2	2 Java	64
5.2.3	3 C# (C Sharp)	65
6 ΣXE	ΔΙΑΣΜΟΣ ΕΦΑΡΜΟΓΗΣ	66
6.1	Αρχική σκέψη	66
6.2	Στόχος	66
6.3	Αναγνώριση πινάκων	67
6.4	Ονομασία της εφαρμογής	68
6.5	Λογότυπο	69
6.6	Διαδικασία εύρεσης της καταλληλότερης διεπαφής χρήστη	70
7 Δ IEI	ΠΑΦΗ ΧΡΗΣΤΗ (USER INTERFACE) ΚΑΙ ΤΡΟΠΟΣ ΛΕΙΤΟΥΡΓΙΑΣ	74 21

	7.1	Αρχ	χική οθόνη	75
	7.2	Οθά	όνη Αναμονής	76
	7.3	Οθά	όνη σάρωσης	77
	7.4	Πάν	νελ πληροφοριών	78
	7.5	Bor	ίθεια χρήστη	80
	7.6	Παβ	ράδειγμα χρήσης	83
8	АП	APA	ΙΤΗΤΟΙ ΠΟΡΟΙ ΓΙΑ ΤΗ ΛΕΙΤΟΥΡΓΙΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	84
	8.1	Απο	οκλειστική χρήση δικτύου ΜΕC	84
	8.2	Про	ραπαιτούμενη εγκατάσταση	84
	8.2.	1	Διακομιστής	85
	8.2.	2	Δίκτυο ΜΕС	86
	8.3	Υλι	κοτεχνικός εξοπλισμός και κόστη	86
	8.4	Χρο	όνος εγκατάστασης	89
9	YΛ	ОПС	ΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ	90
	9.1	Βήμ	ιατα κατά τη δημιουργία εφαρμογής	90
	9.2	Συλ	λογή δεδομένων	91
	9.2.	1	Ιστότοπος Εθνικής Πινακοθήκης	91
	9.2.	2	Διαδικασία συλλογής δεδομένων	93
	9.2.	3	Αποτελέσματα συλλογής δεδομένων	99
	9.2.	4	Ανάλυση δεδομένων	103
	9.3	Δημ	1ιουργία βάσης δεδομένων και εισαγωγή στο Vuforia Engine	104
	9.3.	1	Vuforia Cloud Database	104
	9.3.	2	Τρόπος λειτουργίας της Vuforia Cloud Database	106
	9.3.	.3	Σημεία αναγνώρισης στόχου	107
	9.3.	4	Βαθμολογία στόχου	108
	9.3.	5	Βασικά χαρακτηριστικά που καθορίζουν τη βαθμολογία	109
	9.3.	6	Vuforia Cloud Recognition: Ορισμός και λόγοι επιλογής	111
	9.3.	7	Δημιουργία Vuforia Cloud Database	112 22

9.3.8	Vuforia Web Services (VWS): Κώδικας υλοποίησης	129
9.4 Σχ	ζεδιασμός εφαρμογής στο Unity	138
9.4.1	Λήψη και εγκατάσταση Unity Engine	
9.4.2	Εγκατάσταση Vuforia Engine	140
9.4.3	Unity: Βασικές έννοιες	143
9.4.4	Σκηνές εφαρμογής	152
9.4.5	Δημιουργία εκτελέσιμων αρχείων	
10 EПI	ΛΟΓΟΣ	208
10.1 Y	λικό και χρήσιμοι σύνδεσμοι	
10.2 Σι	ομπεράσματα και μελλοντική επέκταση	209
11 BIB	ΛΙΟΓΡΑΦΙΑ	212

1 ΕΙΣΑΓΩΓΗ – ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

1.1 Εισαγωγή στην επαυξημένη πραγματικότητα

Ο όρος «επαυξημένη πραγματικότητα» (Augmented Reality - AR) αναφέρεται στην τεχνολογία η οποία επιτρέπει την προσθήκη ψηφιακής πληροφορίας στο πραγματικό-φυσικό περιβάλλον μέσω κατάλληλων συσκευών. Το πραγματικό περιβάλλον δεν υποκαθίσταται όπως συμβαίνει στην εικονική πραγματικότητα (Virtual Reality - VR), αλλά εμπλουτίζεται με πρόσθετες ψηφιακές πληροφορίες που ενσωματώνονται σε αυτό. Ο όρος χρησιμοποιήθηκε για πρώτη φορά από τον Tom Caudell to 1990.

Ένα σύστημα επαυξημένης πραγματικότητας έχει τα εξής χαρακτηριστικά:

- Συνδυάζει το πραγματικό περιβάλλον με την εικονική πληροφορία
- Επιτρέπει την αλληλεπίδραση σε πραγματικό χρόνο
- Λειτουργεί αναφορικά με το πραγματικό περιβάλλον

Η επαυξημένη πραγματικότητα χρησιμοποιείται για να βελτιώσει την προοπτική του φυσικού κόσμου και να προσφέρει εμπλουτισμένη εμπειρία της πραγματικότητας στο χρήστη. Οι τεχνολογίες της όρασης και της γραφικής υπολογιστών μετατρέπουν το πραγματικό περιβάλλον σε ψηφιακή πληροφορία παρέχοντας στο χρήστη τη δυνατότητα της διαδραστικότητας και του χειρισμού των ψηφιακών δεδομένων.

Η αλματώδης εξέλιξη της τεχνολογίας έχει προσφέρει στον καταναλωτή πληθώρα έξυπνων συσκευών που χρησιμεύουν στη διευκόλυνσή της καθημερινότητας του, καθώς και στην ψυχαγωγία του. Η τεχνολογία αυτή, σε συνδυασμό με τη διαρκή αναβάθμιση των δικτύων τηλεπικοινωνιών, έχουν δώσει στο χρήστη την ευκαιρία να εξερευνήσει την επαυξημένη πραγματικότητα μέσω ποικίλων εφαρμογών και τελικά να την ενσωματώσει στις καθημερινές του δραστηριότητες.

1.1.1 Μικτή Πραγματικότητα

To 1994 οι Paul Milgram και Fumio Kishino όρισαν το «συνεχές πραγματικότητας- εικονικότητας» το οποίο εκτείνεται από το πρώτο άκρο, που είναι το πραγματικό περιβάλλον, στο δεύτερο, που είναι ένα πλήρως εικονικό περιβάλλον. Ενδιάμεσα βρίσκεται η «μικτή πραγματικότητα» (mixed reality)

που μπορεί να διακριθεί σε «επαυξημένη πραγματικότητα» και «επαυξημένη εικονικότητα» (Augmented Virtuality - AV). Η AR βρίσκεται κοντινότερα στο πραγματικό περιβάλλον, αφού έχει ως βάση της τον πραγματικό κόσμο, στον οποίο προστίθενται εικονικά αντικείμενα. Αντίθετα, η AV βρίσκεται πλησιέστερα στο εικονικό περιβάλλον και ορίζεται ως υποκατηγορία της μικτής πραγματικότητας, στην οποία το εικονικό περιβάλλον εμπλουτίζεται με δεδομένα από τον πραγματικό κόσμο.



Σχήμα 1.1 Το συνεχές εικονικότητας – πραγματικότητας

1.2 Χαρακτηριστικά συστήματος AR

Ένα απλό σύστημα επαυξημένης πραγματικότητας περιλαμβάνει μια κάμερα, μία μονάδα επεξεργασίας και μία οθόνη. Η λειτουργία του συστήματος αυτού είναι να τοποθετεί εικονικά αντικείμενα στο σημείο που υποδεικνύει μία εικόνα-δείκτης (marker). Το σύστημα λαμβάνει μία εικόνα από το πραγματικό περιβάλλον και κατόπιν εντοπίζει την εικόνα-δείκτη χρησιμοποιώντας μία μονάδα εντοπισμού, η οποία υπολογίζει την τοποθεσία και τον προσανατολισμό της κάμερας βασιζόμενη στη λήψη. Η μονάδα επίστρωσης (rendering module) χρησιμοποιεί τη θέση και τον προσανατολισμό της κάμερας σε σχέση με την εικόνα-δείκτη για να τοποθετήσει το εικονικό αντικείμενο πάνω στην εικόνα και να παρουσιάσει το αποτέλεσμα στην οθόνη. Αυτό είναι το απλούστερο σύστημα επαυξημένης πραγματικότητας και χρησιμοποιείται ευρέως στις εφαρμογές παιχνιδιών. Χρησιμοποιώντας την ανίχνευση αντικειμένων, το σύστημα ΑR γνωρίζει τόσο την τοποθεσία όσο και τον τύπο ενός αντικειμένου στην σκηνή και-βάσει αυτών-μπορεί να παρουσιάσει τα κατάλληλα εικονικά αντικείμενα ή την ψηφιακή πληροφορία.



Σχήμα 1.2 Διάγραμμα απλού AR συστήματος

1.3 Συσκευές απεικόνισης

Ο πλέον απαραίτητος εξοπλισμός ενός συστήματος επαυξημένης πραγματικότητας είναι η συσκευή απεικόνισης, η οποία εμφανίζει τη μίξη του πραγματικού και του εικονικού περιβάλλοντος. Μεγάλες εταιρίες τεχνολογίας έχουν κατασκευάσει τα τελευταία έτη συσκευές που προορίζονται αποκλειστικά για χρήση εφαρμογών AR, όπως τα έξυπνα γυαλιά (smart glasses). Καθώς όμως η τεχνολογία της επαυξημένης πραγματικότητας εξελισσόταν και αποκτούσε κοινό, οι μεγαλύτερες εταιρίες κατασκευής έξυπνων κινητών τηλεφώνων (smartphones) και tablets, ενσωμάτωσαν στα προϊόντα τους τα κατάλληλα εργαλεία και το λογισμικό που χρειάζονταν, ώστε κάθε τέτοια συσκευή να υποστηρίζει εφαρμογές AR. Βάσει της θέσης τους μεταξύ του χρήστη και του πραγματικού περιβάλλοντος, οι συσκευές αυτές διακρίνονται στις εξής κατηγορίες:

- Φορετές (head-worn): Οι φορετές συσκευές απεικόνισης (head-mounted display-HMD) προσαρτώνται στο κεφάλι του χρήστη και συνδυάζουν την εικόνα του πραγματικού κόσμου με εικονικά αντικείμενα μέσω οπτικής ή βίντεο τεχνολογίας. Μία ιδιαίτερα δημοφιλής συσκευή αυτής της κατηγορίας είναι το Google Glass.
- Φορητές (hand-held): Οι φορητές συσκευές απεικόνισης (hand-held) έχουν μία οθόνη στην οποία απεικονίζεται η επαυξημένη σκηνή και μια ενσωματωμένη κάμερα. Έχουν μέγεθος τέτοιο ώστε ο χρήστης να μπορεί να τις κρατήσει στο χέρι του και να τις μεταφέρει. Τέτοιες συσκευές είναι τα κινητά τηλέφωνα, τα PDA (Personal Digital Assistant) και τα tablets.
- Χωρικές (spatial): Κατανέμονται στατικά στο χώρο. Συνήθως, χρησιμοποιούνται για μεγάλες παρουσιάσεις και εκθέσεις, που δεν απαιτούν αλληλεπίδραση με το χρήστη. Οι συσκευές αυτές μπορεί να είναι: είτε προβολείς, είτε συσκευές που διαθέτουν μια συμβατική οθόνη (υπολογιστής, τηλεόραση) που προβάλλει τη ροή μιας κάμερας με μια επίστρωση επαυξημένης πραγματικότητας.

1.4 Ασύρματη συνδεσιμότητα για τις AR εφαρμογές

Τα ασύρματα δίκτυα χρησιμοποιούνται ευρέως για την επικοινωνία και τη μεταφορά δεδομένων στην κινητή επαυξημένη πραγματικότητα (mobile augmented reality–MAR). Ως αποτέλεσμα της συνεχούς αναβάθμισης στις τεχνολογίες ασύρματων δικτύων και των επενδύσεων σε υποδομές, έχουν δομηθεί πολυάριθμα συστήματα βασιζόμενα σε αρχιτεκτονική πελάτη-διακομιστή (client-server) αξιοποιώντας τα ασύρματα δίκτυα για την επικοινωνία. Οι κινητές συσκευές διαθέτουν ποικίλες διεπαφές δικτύου και μπορούν να συνδεθούν με έναν απομακρυσμένο διακομιστή (remote server) είτε μέσω του κυψελωτού δικτύου, είτε με την τεχνολογία Wi-Fi. Συνήθως, οι φορετές συσκευές δεν έχουν κάποια διεπαφή που να τις συνδέει άμεσα με το δίκτυο αλλά συνδέονται σε αυτό μέσω άλλων συσκευών με την τεχνολογία bluetooth.

Ο μέγιστος εκτιμώμενος όγκος δεδομένων που θεωρητικά μεταφέρονται σε μία ροή βίντεο είναι 9-12Gbps. Αυτή η εκτίμηση αντιπροσωπεύει το ανώτατο όριο δεδομένων που μπορούν να παραχθούν ανά δευτερόλεπτο. Ακόμα και αν αυτός ο αριθμός μπορεί να ελαττωθεί σημαντικά λόγω συμπίεσης, επιλογής συγκεκριμένων περιοχών κ.α., μερικές εφαρμογές μπορεί να απαιτήσουν εκατοντάδες megabits ή ακόμα και gigabits ανά δευτερόλεπτο.

Σχετικά με τις απαιτήσεις στο μέγεθος της καθυστέρησης και σύμφωνα με εμπειρικές μελέτες σε realtime εφαρμογές, αν θεωρηθεί μια από άκρο σε άκρο σύνδεση μονής κατεύθυνσης, η καθυστέρηση κυμαίνεται από 75ms έως και 250ms, με τη μέση καθυστέρηση να προκύπτει γύρω στα 100ms. Εντούτοις, λόγω πολλών ζητημάτων στις AR εφαρμογές, όπως η ευθυγράμμιση του εικονικού επιπέδου με τον πραγματικό κόσμο, η καθυστέρηση πρέπει να είναι μικρότερη από 20ms, ενώ έχει υποστηριχθεί ότι ιδανικό μέγεθος είναι περίπου 7ms, ώστε να διατηρηθεί η ακεραιότητα του εικονικού περιβάλλοντος και να αποφευχθούν φαινόμενα, όπως η ναυτία λόγω κίνησης, στους χρήστες. Υπάρχουν τρεις σημαντικές κατηγορίες δικτύων που χρησιμοποιούνται στις AR εφαρμογές.

1.4.1 Wireless Wide Area Network (WWAN)

Τα ασύρματα δίκτυα ευρείας περιοχής εκτείνονται σε μεγάλες γεωγραφικές περιοχές και συνδέουν τοπικά δίκτυα. Κάθε σταθμός βάσης στα κυψελωτά WWAN καλύπτει αποστάσεις της τάξης των km. Οι σταθμοί βάσης συνδέονται μεταξύ τους με ένα δίκτυο κορμού που είναι συνήθως ενσύρματο. Το είδος αυτών των δικτύων είναι κατάλληλο για εφαρμογές που υποστηρίζουν κινητικότητα μεγάλης κλίμακας. Υπάρχουν πολλές εφαρμογές των WWAN βασιζόμενες σε διαφορετικές τεχνολογίες, όπως το 2G GSM και το CDMA, το 2.5G GPRS, το 3G UMTS και το 4G LTE. Τα δίκτυα υψηλότερης

γενιάς έχουν αρκετά μεγαλύτερο εύρος ζώνης και μικρότερη καθυστέρηση σε σχέση με παλαιότερων γενεών. Πολλές AR εφαρμογές χρησιμοποιούν τα WWAN για επικοινωνία και μεταφορά δεδομένων. Ένα μεγάλο πρόβλημα των WWAN είναι το μεγάλο κόστος για την αρχική επένδυση των υποδομών. Εντούτοις, η WWAN τεχνολογία είναι πολύ δημοφιλής και μερικές φορές αποτελεί τη μοναδική λύση για κινητές AR εφαρμογές, καθώς είναι η μόνη διαθέσιμη τεχνολογία για μεγάλους δημόσιους χώρους.

1.4.2 Wireless Local Area Network (WLAN)

Τα ασύρματα τοπικά δίκτυα εκτείνονται μερικές εκατοντάδες μέτρα. Λειτουργούν σε πολύ μικρότερη περιοχή από αυτή των WWAN, αλλά προσφέρουν μεγαλύτερο εύρος ζώνης και μικρότερες καθυστερήσεις. Οι τεχνολογίες Wi-Fi και MIMO είναι παραδείγματα τυπικών WLAN. Το είδος των δικτύων αυτών είναι πολύ δημοφιλές και κατάλληλο για εφαρμογές εσωτερικού χώρου. Όμως, η μικρή κάλυψη μπορεί να αποτελέσει πρόβλημα όταν ο χρήστης κινείται σε μεγάλες αποστάσεις.

1.4.3 Wireless Personal Area Network (WPAN)

Τα δίκτυα προσωπικής περιοχής επιτρέπουν σε συσκευές να επικοινωνούν εντός της εμβέλειας ενός ατόμου. Οι ηλεκτρονικοί υπολογιστές, τα έξυπνα τηλέφωνα και τα PDA μπορούν να συνδεθούν μεταξύ τους σε ένα WPAN. Υπάρχουν πολλές εφαρμογές WPAN, όπως η τεχνολογία Bluetooth, το ZigBee και το UWB. Οι δύο πρώτες χρησιμοποιούνται συνήθως για ανίχνευση της θέσης, καθώς και για μεταφορά δεδομένων, ενώ το UWB χρησιμοποιείται κυρίως στην ανίχνευση. Το WPAN έχει αρκετά πλεονεκτήματα, όπως ο μικρός όγκος, η μικρή κατανάλωση ενέργειας και το μεγάλο εύρος ζώνης, αλλά δεν ενδείκνυται για εφαρμογές σε μεγάλες περιοχές.

Туре	Technology	Coverage	Bandwidth(bps)	Latency(ms)	Power(mw)
	GSM	\sim 35km	60K	high	1000~2000
	CDMA	12km~50km	384K	high	200~100
WWAN	GPRS	$\sim 10 \text{km}$	56K~114K	high	~ 1000
	UMTS	1km~2km	2M	medium	~ 250
	LTE	$5 \text{km} \sim 100 \text{ km}^a$	5~50M	low	~ 500
W/LAN	Wi-Fi	~90m	11M~54M	low	~ 100
WLAN	MIMO	$\sim 100 \mathrm{m}$	300M	medium	unknown
	UWB	10m~100m	20M~1G	low	20~1000
WPAN	Bluetooth	$\sim 10 \text{m}$	$1M \sim 3M$	medium	1~2.5
	ZigBee	~75m	20K~250K	low	1~100

Πίνακας 1.1 Σύγκριση διαφορετικών τύπων ασύρματων δικτύων για MAR εφαρμογές

1.5 Εφαρμογές του AR

Καθώς η τεχνολογία εξελίσσεται συνεχώς και οι έξυπνες συσκευές αποτελούν αναπόσπαστο τμήμα της καθημερινότητας των ανθρώπων, έχουν αναπτυχθεί πολλές AR εφαρμογές με σκοπό την εξυπηρέτηση του χρήστη σε διάφορους τομείς. Αναφέρονται κάποιες από τις βασικότερες:

Πλοήγηση

Η επαυξημένη πραγματικότητα μπορεί να βελτιώσει σημαντικά την αποτελεσματικότητα των συσκευών πλοήγησης. Οι πληροφορίες μπορούν να εμφανίζονται στον ανεμοθώρακα του οχήματος, υποδεικνύοντας τις κατευθύνσεις και τις αποστάσεις, τις καιρικές συνθήκες, την ποιότητα του εδάφους, την κυκλοφοριακή κίνηση, καθώς επίσης και προειδοποιήσεις για ενδεχόμενο κίνδυνο.

• Παιχνίδια

Η επαυξημένη πραγματικότητα έχει διεισδύσει στον τομέα της ψυχαγωγίας και έχει τύχει ιδιαίτερης αποδοχής από τη βιομηχανία παιχνιδιών. Τα πρώτα AR παιχνίδια ήταν σχεδιασμένα για εσωτερικούς χώρους και ο χρήστης είχε τη δυνατότητα να αλληλεπιδρά σε πραγματικό χρόνο, τόσο με πραγματικά αντικείμενα του περιβάλλοντός του όσο και με εικονικά. Η αίσθηση του ρεαλισμού που προσφέρει η επαυξημένη πραγματικότητα προσέλκυσε το κοινό και αύξησε κατακόρυφα τη δημοτικότητα των AR παιχνιδιών.

Εκπαίδευση

Στον τομέα της εκπαίδευσης, η επαυξημένη πραγματικότητα έχει χρησιμοποιηθεί για να βελτιώσει την εμπειρία των μαθητών κάνοντας το αντικείμενο μελέτης περισσότερο δελεαστικό και ενδιαφέρον. Εφαρμογές AR χρησιμοποιούνται για την προσομοίωση εργαστηριακών πειραμάτων, τη διδασκαλία γεωμετρικών εννοιών, την περιβαλλοντική εκπαίδευση, τη γεωγραφία, την αναπαράσταση ιστορικών γεγονότων και γενικότερα την κατανόηση δυσνόητων εννοιών και φαινομένων. Μέσω των AR εφαρμογών, οι μαθητές μπορούν να κατανοούν περίπλοκες και αφηρημένες επιστημονικές έννοιες καθώς πλέον εμφανίζονται στις οθόνες τους.

Ιατρική

Η επαυξημένη πραγματικότητα μπορεί να δώσει λύσεις στον τομέα της υγείας, διευκολύνοντας τόσο τους γιατρούς στο έργο τους όσο και τους φοιτητές ιατρικής στην εκμάθηση της επιστήμης τους. Η αρχή για τις AR εφαρμογές στον τομέα της υγείας έγινε το 2006, όταν κυκλοφόρησε μια συσκευή για τους παρόχους υγείας, η οποία μπορεί να απεικονίζει το σύστημα των φλεβών κάθε ασθενή. Σήμερα, οι γιατροί μπορούν να χρησιμοποιήσουν την επαυξημένη πραγματικότητα για τη βοήθεια στη διάγνωση και τη θεραπεία μιας ασθένειας ή ακόμα και στη χειρουργική. Τα δεδομένα που συλλέγονται σε μία εξέταση, όπως για παράδειγμα μια μαγνητική τομογραφία ή ένας υπέρηχος, μπορούν να συνδυαστούν με την επιτόπια εξέταση του ασθενούς και μέσω μίας συσκευής απεικόνισης, ο γιατρός είναι σε θέση να έχει μια ρεαλιστική εικόνα του εσωτερικού του ασθενούς. Ανάλογη διαδικασία γίνεται και στη λαπαροσκοπική χειρουργική, όπου ο γιατρός μπορεί να έχει οπτική πληροφορία για την χειρουργούμενη περιοχή χωρίς να χρειάζεται να κάνει περαιτέρω τομές.

Εμπόριο

Οι AR εφαρμογές μπορούν να προσφέρουν έξυπνες λύσεις στις καθημερινές αγορές των καταναλωτών αυξάνοντας έτσι την ποιότητα της εμπειρίας του αγοραστή. Τα τελευταία χρόνια έχουν αναπτυχθεί πολλές AR εφαρμογές στον τομέα του ηλεκτρονικού εμπορίου, οι οποίες προσφέρουν στον καταναλωτή τη δυνατότητα να δοκιμάσει προϊόντα μέσω της οθόνης της συσκευής του. Ο χρήστης χρησιμοποιώντας την κάμερα της συσκευής του μπορεί να επιλέξει τα προϊόντα της αρεσκείας του, όπως ρούχα, παπούτσια, κοσμήματα και να τα δει φορεμένα πάνω του. Σε κάποιες περιπτώσεις, μπορεί να απαιτείται η εκτύπωση κάποιας εικόνας, η οποία λειτουργεί ως δείκτης, ώστε η εφαρμογή να γνωρίζει το σημείο όπου θα τοποθετήσει το εικονικό αντικείμενο. Η επαυξημένη πραγματικότητα μπορεί να διευκολύνει τους καταναλωτές και κατά την παρουσία τους στα καταστήματα. Ο χρήστης της AR εφαρμογής μπορεί να στοχεύσει με την κάμερα της έξυπνης συσκευής του το αντικείμενο του ενδιαφέροντός του. Αμέσως, εμφανίζονται συ, κριτικές άλλων αγοραστών, σύγκριση με προϊόντα της ίδιας κατηγορίας του και πληροφορίες σχετικά με την τιμή του. Επίσης, μία AR εφαρμογή μπορεί να δώσει στον αγοραστή τη δυνατότητα να «δει» στο εσωτερικό μιας συσκευσίας, χωρίς να την ανοίξει.

• Τουρισμός και αξιοθέατα

Στον τομέα του τουρισμού έχουν αναπτυχθεί AR εφαρμογές που σχετίζονται με τη γεωγραφική τοποθεσία του χρήστη και έχουν κυρίως πληροφοριακό χαρακτήρα. Χρησιμοποιώντας μια τέτοια AR εφαρμογή, ο ταξιδιώτης μπορεί να αναζητήσει κατάλυμα, σημεία διασκέδασης, αξιοθέατα της περιοχής, ενώ όλες αυτές οι πληροφορίες εμφανίζονται στη συσκευή του σε πραγματικό χρόνο. Με την τεχνολογία της επαυξημένης πραγματικότητας παρέχεται η δυνατότητα στο χρήστη να στοχεύσει με την κάμερά του (tablet, smartphone κλπ) πάνω σε ένα αξιοθέατο, είτε πραγματικό, είτε εικονικό (χάρτη, φωτογραφία κλπ), και να λάβει πληροφορίες στην οθόνη του για το συγκεκριμένο σημείο ενδιαφέροντος. Οι πληροφορίες που λαμβάνει ο χρήστης έχουν τη μορφή κειμένου, φωτογραφίας, video ή παραπομπών για διευθύνσεις στο Διαδίκτυο. Τα αξιοθέατα μπορούν είτε να βρίσκονται σε εξωτερικό χώρο, είτε να είναι εκθέματα μουσείων. Οι προηγμένες AR εφαρμογές του είδους περιλαμβάνουν προσομοιώσεις ιστορικών γεγονότων, τοποθεσιών και αντικειμένων που σχετίζονται με συγκεκριμένο περιβάλλον.

2 Η ΤΕΧΝΟΛΟΓΙΑ ΜΕC ΣΤΗΝ ΥΠΗΡΕΣΙΑ ΤΩΝ AR ΕΦΑΡΜΟΓΩΝ

2.1 Τεχνολογία LTE

Η τεχνολογία LTE (Long Term Evolution) αποτελεί εξέλιξη των συστημάτων τρίτης γενιάς (3G) και αναπτύχθηκε από τη συνεργασία 3GPP (3rd Generation Partnership Project). Η πρώτη εφαρμογή της τεχνολογίας πραγματοποιήθηκε το 2009, ως υλοποίηση της έκδοσης 8 της 3GPP και έγινε γνωστή παγκοσμίως ως η απαρχή της τεχνολογίας τέταρτης γενιάς (4G). Έκτοτε, ακολούθησαν οι εκδόσεις 9 έως 14, με σκοπό τη βελτίωση της απόδοσης του δικτύου. Οι εκδόσεις 10 έως 12 είναι γνωστές ως LTE-Advanced (LTE-A), ενώ οι επόμενες δύο, 13 και 14, ονομάζονται LTE-Advanced Pro. Το 2018 ακολούθησε η έκδοση 15, η οποία σηματοδοτεί την εισαγωγή στα δίκτυα πέμπτης γενιάς (5G).

2.1.1 Αρχιτεκτονική LTE

Η αρχιτεκτονική ενός δικτύου LTE μπορεί να διαχωριστεί στις εξής επιμέρους μονάδες:

- Το δίκτυο ραδιοπρόσβασης (Radio Access Network RAN), το οποίο στην ορολογία του LTE ονομάζεται "E-UTRAN" (Evolved Universal Terrestrial Radio Access Network) και είναι υπεύθυνο για την επικοινωνία των τερματικών με τους σταθμούς βάσης.
- Το δίκτυο κορμού (Core Network CN), το οποίο ονομάζεται "EPC" (Evolved Packet Core) και συνδέει τους σταθμούς βάσης με το σταθερό δίκτυο.



Σχήμα 2.1 Αρχιτεκτονική LTE

2.1.1.1 E-UTRAN



Σχήμα 2.2 Αριχτεκτονική E-UTRAN

Το E-UTRAN αποτελείται από τους σταθμούς βάσης που ονομάζονται eNB (evolved Node B) και τα τερματικά που στην ορολογία της 3GPP καλούνται UE (User Equipment). Η επικοινωνία μεταξύ των UEs με τον εκάστοτε eNB γίνεται μέσω της διεπαφής Uu, ενώ οι eNBs συνδέονται μεταξύ τους μέσω των διεπαφών X2. Η επικοινωνία μεταξύ των eNBs και του EPC γίνεται μέσω της διεπαφής S1. Η διεπαφή Uu μεταφέρει δεδομένα χρήστη και σηματοδοσίας στις κατευθύνσεις Uplink (UL) και Downlink (DL), ενώ ο κύριος ρόλος της διεπαφής X2 είναι η σηματοδοσία και η προώθηση πακέτων μεταξύ γειτονικών eNBs σε περιπτώσεις μεταπομπής. Η διεπαφή S1 μεταφέρει τόσο δεδομένα χρήστη όσο και ελέγχου μεταξύ του E-UTRAN και του EPC.

2.1.1.2 EPC

To EPC αποτελείται κυρίως από τη μονάδα διαχείρισης κινητικότητας (Mobility Management Entity – MME), την πύλη εξυπηρέτησης (Serving Gateway – S-GW), την πύλη του δικτύου πακέτων (Packet Data Network Gateway – P-GW) και τον οικείο εξυπηρετητή συνδρομητών (Home Subscriber Server – HSS).

Η διεπαφή S1-U μεταφέρει τα δεδομένα του στρώματος χρήστη και από τη διεπαφή S1-MME διέρχονται μηνύματα σηματοδοσίας του στρώματος ελέγχου. Συνεπώς, για το στρώμα χρήστη η S-GW αποτελεί το σημείο επαφής μεταξύ των οντοτήτων E-UTRAN και EPC, με κύριο ρόλο την προώθηση πακέτων από και προς τα τερματικά

Μέσω της διεπαφής SGi συνδέεται η P-GW και επομένως ο EPC με τα εξωτερικά δίκτυα. Κύρια λειτουργία της P-GW είναι η απόδοση διευθύνσεων και προθεμάτων IP στα UEs, ώστε αυτά να αποκτήσουν πρόσβαση στο Διαδίκτυο. Η διεπαφή που συνδέει τις S-GW και P-GW είναι η S5 αν οι δύο συσκευές βρίσκονται στο ίδιο δίκτυο ή η S8 αν ανήκουν σεδιαφορετικό.

Η ΜΜΕ συνδέει το E-UTRAN με τον EPC σχετικά με τις λειτουργίες του στρώματος ελέγχου. Επεξεργάζεται τη σηματοδοσία μεταξύ του UE και του EPC σχετικά με την κινητικότητα και την ασφάλεια της πρόσβασης στο E-UTRAN. Για την υποστήριξη κινητικότητας, η MME επικοινωνεί με κάποια S-GW μέσω της διεπαφής S11, ενώ συνδέεται και με άλλες MMEs με διεπαφές S10, οι οποίες συμμετέχουν κατά τη μεταπομπή ενός UE μεταξύ διαφορετικών MMEs.

Ο HSS αποτελεί μία βάση δεδομένων, η οποία περιέχει τα δεδομένα των συνδρομητών που εξυπηρετούνται από το δίκτυο, όπως το προφίλ QoS(Quality of Service) και πληροφορίες για τις P-GW και τις MME, με τις οποίες μπορεί να συνδεθεί το UE.



Σχήμα 2.3 Αρχιτεκτονική ΕΡC

2.2 Mobile Edge Computing

Οι περισσότερες εφαρμογές που χρησιμοποιούνται σήμερα από τους χρήστες έξυπνων συσκευών χρησιμοποιούν την τεχνολογία υπολογιστικού νέφους (cloud computing). Η τεχνολογία αυτή παρέχει τη δυνατότητα στους χρήστες να αποθηκεύουν, να επεξεργάζονται και να διαχειρίζονται δεδομένα, τα οποία βρίσκονται σε κάποια απομακρυσμένη τοποθεσία που είναι προσβάσιμη μέσω Διαδικτύου. Αν και το cloud computing προσφέρει απεριόριστους πόρους στη διάθεση του χρήστη, δημιουργεί σημαντική καθυστέρηση στην επικοινωνία, αφού οι εξυπηρετητές cloud είναι συνήθως απομονωμένοι από το υπόλοιπο δίκτυο. Το γεγονός αυτό λειτουργεί αρνητικά σε real-time εφαρμογές, όπως υπηρεσίες video streaming, όπου η ανάγκη ελαχιστοποίησης των καθυστερήσεων είναι σημαντική.

Η τεχνολογία Mobile Edge Computing (MEC) αναπτύχθηκε με σκοπό τη μετακίνηση των εξυπηρετητών του δικτύου από τα data centers, ώστε να βρίσκονται όσο το δυνατόν εγγύτερα στον τελικό χρήστη. Με τον τρόπο αυτό, μειώνονται οι καθυστερήσεις στην από άκρο σε άκρο επικοινωνία, αφού η πληροφορία ακολουθεί συντομότερη διαδρομή. Οι εφαρμογές που είναι ευαίσθητες σε καθυστέρηση μπορούν να εξυπηρετούνται από τους MEC εξυπηρετητές, ενώ όσες είναι ανεκτικές σε θέμα καθυστέρησης δρομολογούνται προς κάποιον εξυπηρετητή cloud.



Σχήμα 2.4 Επισκόπηση συστήματος MEC
2.3 Πλεονεκτήματα τεχνολογίας ΜΕC

Στη συνέχεια παρουσιάζονται συνοπτικά τα πλεονεκτήματα του ΜΕC:

- Μείωση της συνολικής καθυστέρησης που αντιλαμβάνεται ο χρήστης.
- Μείωση της συνολικής επιβάρυνσης του δικτύου κορμού, καθώς λιγότερα μηνύματα διέρχονται εντός αυτού.
- Ανάπτυξη καινοτόμων εφαρμογών με υψηλές απαιτήσεις σε καθυστέρηση και ποσοστό απώλειας πακέτων.
- Φιλοξενία εφαρμογών με βάση την τοποθεσία. Αφορά στην περίπτωση όπου οι χρήστες
 επιθυμούν να καταναλώνουν υπηρεσίες που τους ενδιαφέρουν, σε μια συγκεκριμένη
 περιοχή.
- Διευκόλυνση της εύρεσης της τοποθεσίας των τερματικών, κάτι που επιτρέπει την ανάπτυξη εφαρμογών βασισμένων στην τοποθεσία των χρηστών.
- Οικονομικό κέρδος για τους παρόχους δικτύων, αφού υπάρχει η δυνατότητα μίσθωσης πόρων σε τρίτους συνεργάτες για την παροχή δικών τους υπηρεσιών. Αυτό ισχύει σε συνδυασμό με την ευελιξία που υπάρχει στην κατανομή των διαθέσιμων πόρων του συστήματος (αποθηκευτικών, εύρους ζώνης, υπολογιστικής ισχύος) μέσω του MEC.

2.4 Περί δικτύων 5G

Το 5G είναι η επόμενη γενιά της 3GPP τεχνολογίας, η οποία ακολουθεί το 4G/LTE. Η εισαγωγή στα δίκτυα πέμπτης γενιάς έγινε με την έκδοση 15 της 3GPP. Τα 5G δίκτυα σκοπεύουν στην επίτευξη των ακόλουθων στόχων:

- Υψηλή διεκπεραιωτική ικανότητα (1-20Gbps)
- Πολύ μικρές καθυστερήσεις (<1ms)
- 1000x εύρος ζώνης ανά μονάδα περιοχής
- Μαζική συνδεσιμότητα
- Υψηλή διαθεσιμότητα
- Πυκνή κάλυψη
- Χαμηλή κατανάλωση ενέργειας
- Διάρκεια ζωής μπαταρίας συσκευών M2M (Machine-to-Machine) έως 10 έτη

Οι περιπτώσεις εφαρμογής του 5G μπορούν να διαχωριστούν σε τρείς επιμέρους κατηγορίες:

Enhanced Mobile broadband (eMBB): Πρόκειται για εφαρμογές πρόσβασης των χρηστών σε υπηρεσίες με βελτιωμένη απόδοση ως προς την ταχύτητα, τη διεκπεραιωτική ικανότητα και τη χωρητικότητα του δικτύου.

Ultra-reliable low latency Communications (URLLC): Εντάσσονται εφαρμογές που έχουν υψηλές απαιτήσεις σε καθυστέρηση και διαθεσιμότητα του δικτύου, όπως εφαρμογές επαυξημένης και εικονικής πραγματικότητας, τηλε-χειρουργικής, έξυπνων μετακινήσεων και βιομηχανικού αυτοματισμού.

Massive IoT: Περιλαμβάνει εφαρμογές που χρησιμοποιούνται από εξαιρετικά μεγάλο πλήθος συσκευών που επικοινωνούν μεταξύ τους και σχηματίζουν ένα σύστημα IoT.



Σχήμα 2.5 5G Εφαρμογές

2.5 Εφαρμογή της τεχνολογίας ΜΕС στα δίκτυα 5G

Όπως συμβαίνει και στο LTE, η αρχιτεκτονική δικτύων 5G διαχωρίζεται στο δίκτυο κορμού και το δίκτυο ραδιοπρόσβασης. Το NG-RAN (Next Generation RAN) του 5G αποτελείται από δύο στοιχεία: τους gNB, οι οποίοι παρέχουν υπηρεσίες στο επίπεδο χρήστη και στο επίπεδο ελέγχου νέας γενιάς και τους ng-eNB, που επικοινωνούν με πρωτόκολλα E-UTRAN του LTE. Το NG-RAN συνδέεται με το δίκτυο κορμού (5G Core – 5GC) μέσω της διεπαφής NG. Η μονάδα AMF (Access and Mobility Management Function) του 5GC συνδέει το RAN με το 5GC και εκτελεί λειτουργίες σχετικές με την εγγραφή των χρηστών, τη διαχείριση κινητικότητας και την ασφάλεια. Επίσης, η μονάδα UPF (User Plane Function) συνδέει το NG-RAN με το στρώμα δεδομένων.

To 5GC αποτελείται από ένα σύνολο δικτυακών λειτουργιών (Network Functions – NF), οι οποίες μπορούν να πραγματωθούν ως εικονικές λειτουργίες δικτύου (Virtual NF – VNF). Το 5G εφαρμόζει την τεχνική διαχωρισμού των CP και UP επιπέδων μέσω της ανάθεσης διακριτών λειτουργιών σε κάθε NF. Η τεχνική αυτή ονομάζεται "CUPS" (Control Plane User Plane Separation) και επιτρέπει την κατανομή των στοιχειών του κορμού, οδηγώντας σε αποδοτικότερη χρησιμοποίηση των πόρων του δικτύου.

Η αρχιτεκτονική 5G μπορεί να αναπαρασταθεί βάσει της μορφής SBA (Service Based Architecture), όπου κάθε NF συνδέεται με τις υπόλοιπες μέσω μιας μοναδικής διεπαφής, ώστε να

επιτρέπεται η ανταλλαγή υπηρεσιών. Στο Σχ.2.6 απεικονίζεται η σύνδεση ενός 5G δικτύου σε μορφή SBA, με ένα σύστημα MEC.

Το σύστημα MEC λειτουργεί ως εξωτερικό δίκτυο δεδομένων, άρα ο ME host συνδέεται με τη μονάδα λειτουργιών του στρώματος δεδομένων (User Plane Function - UPF). Ο ρόλος της UPF είναι να κατευθύνει την κίνηση δεδομένων χρήστη μεταξύ του 5G δικτύου και των εφαρμογών MEC βάσει κανόνων που λαμβάνει από τη μονάδα διαχείρισης συνόδου (Session Management Function – SMF). Η κίνηση επηρεάζεται επίσης από τη μονάδα ελέγχου συμπεριφοράς δικτύου (Policy Control Function – PCF), σύμφωνα με τους κανόνες χρέωσης των χρηστών.



Σχήμα 2.6 Ένταζη του ΜΕС στο δίκτυο 5G

Οι μονάδες του MEC, όπως ο ενορχηστρωτής και η πλατφόρμα, αντιμετωπίζονται από το 5G δίκτυο ως λειτουργίες εφαρμογών (Application Functions – AF) και επικοινωνούν με αυτό μέσω της διεπαφής Naf. Με αυτήν την υλοποίηση, υπηρεσίες που προέρχονται από εφαρμογές που τρέχουν στο host, μπορούν να διαφημιστούν στο 5G. Η μονάδα NEF (Network Exposure Function) εκθέτει τις διαθέσιμες δυνατότητες και γεγονότα σε άλλες NF και παρέχει ασφαλή πρόσβαση σε εφαρμογές εκτός του δικτύου 3GPP. Στο σύστημα MEC, η NEF χρησιμοποιείται για την επικοινωνία του Mobile Edge Orchestrator (MEO) με τις υπόλοιπες NF. Ωστόσο, στην περίπτωση όπου δεν τίθενται θέματα ασφάλειας, ο MEO επικοινωνεί απευθείας με τις NF του 5G. Επίσης, είναι δυνατή η τοποθέτηση μιας NEF εντός του MEC host, η οποία θα παρέχει άμεσα τοπικές πληροφορίες ραδιοπρόσβασης στις εφαρμογές MEC.

2.6 Η τεχνολογία ΜΕC στις ΑR εφαρμογές

Οι δύο σημαντικότερες απαιτήσεις των AR εφαρμογών είναι η μεγάλη ταχύτητα εξυπηρέτησης και η μικρή καθυστέρηση. Αυτά τα δύο προβλήματα καλείται να λύσει μια νέα τεχνολογία δικτύου που ονομάζεται MEC (Mobile Edge Computing). Σύμφωνα με την αρχιτεκτονική του MEC, οι εξυπηρετητές δικτύου έρχονται εγγύτερα στον τελικό χρήστη. Με αυτόν τον τρόπο, μειώνεται η διαδρομή ανάμεσα στο χρήστη της εφαρμογής και στον εξυπηρετητή, με αποτέλεσμα να ελαχιστοποιείται η καθυστέρηση και να αποφεύγονται φαινόμενα συμφόρησης.

Η AR τεχνολογία επωφελείται από τα πλεονεκτήματα του Mobile Edge Computing, καθώς παρουσιάζει υψηλές απαιτήσεις σε εύρος ζώνης και χαμηλή καθυστέρηση. Παράλληλα, το MEC υποστηρίζει real-time μετάδοση δεδομένων, γιατί οι πληροφορίες που απαιτούνται από τις εφαρμογές βρίσκονται στην άκρη του δικτύου κοντά στον τελικό χρήστη. Οι τοπικοί MEC εξυπηρετητές που εγκαθίστανται στο δίκτυο πρόσβασης, επιτυγχάνουν χαμηλή καθυστέρηση και υψηλές ταχύτητες, χαρακτηριστικά που ευνοούν την απρόσκοπτη-υψηλής ποιότητας-λειτουργία των AR εφαρμογών.

2.7 Επισκόπηση του ΜΕC συστήματος

Οι υπηρεσίες επαυξημένης πραγματικότητας απαιτούν την επεξεργασία υψηλού όγκου δεδομένων σε πραγματικό χρόνο, αφού το περιβάλλον που αντιλαμβάνεται ο χρήστης μέσω της κάμεράς του πρέπει να ανανεώνεται συνεχώς, όσο αυτός κινείται. Σε αυτό συμβάλλει η τεχνολογία MEC, σύμφωνα με την οποία οι εξυπηρετητές των σταθμών βάσης φιλοξενούν την απαιτούμενη πληροφορία και αναλαμβάνουν την επεξεργασία των μηνυμάτων που ανταλλάσσει η συσκευή του χρήστη με το δίκτυο. Συγκεκριμένα, ο χρήστης προωθεί στο δίκτυο την έξοδο (output) της κάμεράς του, ώστε να εντοπιστεί η ακριβής θέση και ο προσανατολισμός του. Ο εξυπηρετητής MEC επεξεργάζεται τα δεδομένα που λαμβάνει από την AR εφαρμογή και επιστρέφει στο χρήστη πληροφορίες, που βρίσκονται αποθηκευμένες σε κάποιο τοπικό data center. Παράλληλα, ο εξυπηρετητής MEC επικοινωνεί με ένα κεντρικό εξυπηρετητή, όταν χρειάζεται, προκειμένου να αποκτήσει δεδομένα που δεν έχει στη διάθεσή του.

Οι πληροφορίες που αναζητούν οι χρήστες των AR εφαρμογών είναι, συνήθως, άμεσα συσχετισμένες με μια φυσική περιοχή ενδιαφέροντος. Επομένως, η χρήση της τεχνολογίας MEC υπερτερεί σε σχέση με την παραδοσιακή cloud αρχιτεκτονική, αφού ο τοπικός MEC server διαθέτει τους απαιτούμενους υπολογιστικούς πόρους και τα δεδομένα της AR εφαρμογής, ενώ παράλληλα βρίσκεται εγγύτερα στον τελικό χρήστη. Επιπλέον, οι απαιτήσεις των AR εφαρμογών σε μικρή καθυστέρηση και μεγάλο εύρος ζώνης ικανοποιούνται πλήρως μέσω της MEC τεχνολογίας. Όπως φαίνεται στο παρακάτω σχήμα (Σχ. 2.7), μια κεντρική προσωρινή μνήμη μπορεί να αναπτυχθεί στο δίκτυο κινητής τηλεφωνίας και να αποθηκεύει AR αντικείμενα. Η κεντρική προσωρινή μνήμη επικοινωνεί με τον διακομιστή MEC χρησιμοποιώντας το αναγνωριστικό "object ID".



Σχήμα 2.7 Τεχνολογία MEC στις AR εφαρμογές

3 ΔΙΑΣΤΑΣΙΟΛΟΓΗΣΗ ΣΥΣΤΗΜΑΤΩΝ ΕΞΥΠΗΡΕΤΗΣΗΣ ΑR ΕΦΑΡΜΟΓΩΝ

3.1 Γενικά στοιχεία

Στο παρόν κεφάλαιο θα εξεταστούν δύο σενάρια εξυπηρέτησης AR εφαρμογών βάσει των τεχνολογιών δικτύου που αναπτύχθηκαν στις προηγούμενες ενότητες. Για το λόγο αυτό, πρέπει προηγουμένως να εκτιμηθούν βασικές παράμετροι, όπως το μήκος των μηνυμάτων που ανταλλάσσει ένα τερματικό με το δίκτυο κατά την εκτέλεση AR εφαρμογών και το πλήθος των τερματικών που βρίσκονται στην υπό εξέταση περιοχή.

Σε ένα τυπικό σύστημα επαυξημένης πραγματικότητας, όπου το πραγματικό περιβάλλον που αντιλαμβάνεται ο χρήστης μέσω της κάμεράς του επαυξάνεται με εικονικά αντικείμενα, η ανταλλαγή μηνυμάτων μεταξύ του τερματικού και του δικτύου πραγματοποιείται ως εξής:

- Ο χρήστης αποστέλλει στο δίκτυο τη ροή βίντεο από την έξοδο της κάμεράς του, ώστε να γίνει ανίχνευση της θέσης και του προσανατολισμού του.
- Τα δεδομένα που εξάγονται από τη ροή βίντεο υποβάλλονται σε επεξεργασία από κάποιον εξυπηρετητή. Ακολούθως, προωθούνται στο χρήστη πληροφορίες υπό τη μορφή εικόνας, βίντεο κλπ.
- Η διαδικασία αυτή επαναλαμβάνεται συνεχώς, καθώς ο χρήστης αλλάζει θέση και προσανατολισμό.

Για τη μελέτη της απόδοσης του συστήματος, τόσο χρησιμοποιώντας την τεχνολογία Wi-Fi όσο και αυτή του MEC, πρέπει να εξεταστούν αλληλένδετα μεγέθη, όπως ο χρόνος εξυπηρέτησης των AR εφαρμογών και η διεκπεραιωτική ικανότητα του χρήστη.

3.1.1 Επικοινωνία με το δίκτυο

Η εκτέλεση των AR εφαρμογών θέτει υψηλές απαιτήσεις, τόσο σε υπολογιστική ικανότητα όσο και στη διαθεσιμότητα μεγάλου όγκου πληροφοριών. Για το λόγο αυτό, είναι απαραίτητη

η ύπαρξη εξυπηρετητή, ο οποίος αναλαμβάνει: (i) την εκτέλεση εργασιών επεξεργασίας των δεδομένων που προωθούνται από το χρήστη και (ii) την παροχή των απαιτούμενων πληροφοριών στις AR εφαρμογές. Η τοποθεσία των εξυπηρετητών διαφέρει αντίστοιχα από το σχεδιασμό του δικτύου στο οποίο συνδέεται ο χρήστης.



Σχήμα 3.1 Τοπολογίες δικτύων Wi-Fi και MEC

Στο πλαίσιο της εργασίας, μελετάται η περίπτωση σύνδεσης του χρήστη σε δίκτυο Wi-Fi, μέσω του οποίου επικοινωνεί με κάποιο cloud server. Στη συνέχεια, εξετάζεται η περίπτωση εξυπηρέτησης των AR εφαρμογών βάσει του συστήματος MEC.

Οι λειτουργίες που ανατίθενται από τις AR εφαρμογές στους εξυπηρετητές φαίνονται στο Σχ.3.2. Αρχικά, γίνεται ανίχνευση αντικειμένου (object detection), δηλαδή αναγνωρίζονται οι δείκτες και τα μοτίβα που βρίσκονται στο οπτικό πεδίο του χρήστη, όπως επίσης και οι περιοχές ενδιαφέροντος που ανταποκρίνονται στους δείκτες. Το επόμενο βήμα είναι η εξαγωγή χαρακτηριστικών (feature extraction) από τις περιοχές ενδιαφέροντος. Ακολούθως, γίνεται χρήση της λειτουργίας αναγνώρισης αντικειμένου (object recognition), ώστε να αναγνωριστεί η πρότυπη εικόνα, η οποία ανταποκρίνεται στη μορφή του δείκτη και βρίσκεται αποθηκευμένη σε κάποια βάση δεδομένων.



Σχήμα 3.2 Λειτουργίες πελάτη και εζυπηρετητή

Αφού αναγνωριστεί το αντικείμενο, πραγματοποιείται η αντιστοίχιση προτύπων (template maching) προκειμένου να γίνει επαλήθευση των αποτελεσμάτων και να προσδιοριστεί η θέση και ο προσανατολισμός της κάμερας. Τα δεδομένα που προκύπτουν από την αντιστοίχιση προτύπων χρησιμοποιούνται ως είσοδος στην ιχνηλασία αντικειμένου (object tracking) που πραγματοποιείται από το τερματικό. Τα χαρακτηριστικά που εξάγονται από τη ροή βίντεο και τους χαρακτηριστικούς δείκτες προσδιορίζουν την πληροφορία που θα σταλεί στο χρήστη και θα ενσωματωθεί στην οπτική της κάμεράς του (annotation rendering).

3.2 Επίπεδα ΜΕC στις ΑR εφαρμογές

Σε ένα σύστημα MEC που αποσκοπεί στην εξυπηρέτηση AR εφαρμογών διακρίνονται τρία επίπεδα: το επίπεδο των χρηστών, το επίπεδο edge και το επίπεδο cloud. Όπως φαίνεται στο Σχ.3.3, στο επίπεδο χρηστών βρίσκονται πολλαπλές AR συσκευές, οι οποίες συνδέονται με το επίπεδο edge μέσω ασύρματων συνδέσεων. Το επίπεδο cloud αποτελείται από απεριόριστους αποθηκευτικούς και υπολογιστικούς πόρους. Η αρχιτεκτονική MEC εισάγει το επίπεδο edge, το οποίο μεσολαβεί μεταξύ των χρηστών και του cloud, ώστε να παρέχει ταχύτερα τους απαιτούμενους πόρους στις εφαρμογές.



Σχήμα 3.3 Αρχιτεκτονική MEC για τις AR εφαρμογές

Συγκεκριμένα, οι χρήστες των AR συσκευών αποστέλλουν αίτημα στον εξυπηρετητή cloud, ο οποίος περιέχει τη ροή βίντεο από το περιβάλλον τους. Ο εξυπηρετητής εκτελεί τους απαραίτητους υπολογισμούς-όταν αυτό απαιτείται-ενώ επικοινωνεί με τον cloud εξυπηρετητή για να αποκτήσει δεδομένα που δεν διαθέτει ήδη. Κατόπιν, απαντά στο τερματικό με τα δεδομένα επαύξησης.

3.2.1 Σενάριο εξυπηρέτησης AR εφαρμογών σε σύστημα MEC

Στην περίπτωση εξυπηρέτησης των AR εφαρμογών από το δίκτυο MEC, οι χρήστες αποστέλλουν αίτημα το οποίο περιέχει δεδομένα βίντεο στον εξυπηρετητή MEC. Ο εξυπηρετητής αναλαμβάνει να εκτελέσει τις κατάλληλες διεργασίες, ενώ επικοινωνεί με τον cloud εξυπηρετητή για να αποκτήσει δεδομένα που δεν διαθέτει ήδη. Αφού ενημερωθεί ο MEC εξυπηρετητής, αποστέλλει στο χρήστη τα δεδομένα επαύξησης.

Για να υπάρχει αντιστοιχία με την προηγούμενη μέθοδο, επιλέγεται σαφώς μεγαλύτερος ρυθμός μετάδοσης δεδομένων, καθώς σε αυτήν την περίπτωση δεν υπάρχει ομαδοποίηση των χρηστών και εξυπηρέτηση από διαφορετικά σημεία. Όλοι οι χρήστες επικοινωνούν με τον ίδιο MEC εξυπηρετητή, έχοντας ταχύτητες LTE της τάξης των 150Mbps στη διεύθυνση downlink και 50Mbps στη διεύθυνση uplink. Η καθυστέρηση ζεύξης μεταξύ των χρηστών και του MEC εξυπηρετητή είναι σχεδόν μηδενική. Σε αυτή την περίπτωση, οι χρήστες αντιλαμβάνονται πρακτικά μόνο την καθυστέρηση που εισάγεται για την ανταλλαγή πληροφοριών μεταξύ του cloud και του MEC εξυπηρετητή.



Σχήμα 3.4 Κατανομή εργασιών στο ΜΕC εξυπηρετητή

3.3 ΑR εφαρμογές σε χώρους μουσείων

Η τεχνολογία της επαυξημένης πραγματικότητας εφαρμόζεται σήμερα επιτυχώς σε αρχαιολογικούς χώρους και μουσεία, παρέχοντας στους χρήστες των AR εφαρμογών οπτικοακουστική εμπειρία υψηλής ποιότητας. Οι επισκέπτες των μουσείων έχουν τη δυνατότητα να αλληλεπιδράσουν με τα εκθέματα μέσω των AR εφαρμογών, λαμβάνοντας πρόσθετες πληροφορίες σε μορφή εικόνας ή βίντεο. Με αυτό τον τρόπο ο χρήστης της AR εφαρμογής μπορεί να λάβει πληροφορίες σχετικές με την προέλευση, την κατασκευή και τη χρήση των αντικειμένων, να δει προσομοιώσεις των εκθεμάτων στο περιβάλλον προέλευσής τους ή ακόμα να δει μέσω της οθόνης του τα εκθέματα να κινούνται στο χώρο του μουσείου.



Σχήμα 3.5 Εφαρμογή AR σε εκθέματα μουσείων

Το δίκτυο εξυπηρέτησης στο οποίο συνδέονται οι επισκέπτες πρέπει να είναι ικανό να εξυπηρετήσει τους χρήστες ανεξαρτήτως προσέλευσης. Ενδεικτικά, ο αριθμός των επισκεπτών του Μουσείου της Ακρόπολης είναι κατά μέσο όρο 4.000 ημερησίως, ενώ τους θερινούς μήνες ο αριθμός των επισκεπτών πολλαπλασιάζεται, φθάνοντας σε 1.000 επισκέπτες ανά ώρα.

Για τη μελέτη της απόδοσης του δικτύου στην εξυπηρέτηση των επισκεπτών ενός μουσείου, θεωρείται AR εφαρμογή, η οποία σαρώνοντας με την κάμερα της συσκευής το έκθεμα, εμφανίζει στο χρήστη πληροφορίες σχετικές με αυτό σε μορφή εικόνας. Συγκεκριμένα, ο χρήστης σαρώνει το αντικείμενο και αποστέλλει στο δίκτυο ροή βίντεο που περιέχει το συγκεκριμένο έκθεμα. Ο εξυπηρετητής δικτύου εκτελεί την απαραίτητη επεξεργασία για την εξαγωγή των χαρακτηριστικών της εικόνας και επιστρέφει στο χρήστη εικόνα, που τοποθετείται ως επίστρωση στη ροή βίντεο της κάμεράς του. Το βίντεο που αποστέλλει ο χρήστης στο δίκτυο συνήθως υπόκειται σε προεπεξεργασία στο τερματικό του χρήστη. Η διαδικασία αυτή απαιτεί χρόνο και έχει ως αποτέλεσμα τη συρρίκνωση του βίντεο που θα αποσταλεί στο δίκτυο λόγω της συμπίεσής του. Με τον τρόπο αυτό, μειώνεται η καθυστέρηση μετάδοσης καθώς και η συμφόρηση στο δίκτυο που προκαλείται λόγω των πολλαπλών, μεγάλων πακέτων της ροής βίντεο.

Ως υπόθεση της εργασίας θεωρείται ότι από τους 1.000 επισκέπτες που υποτίθεται ότι συνυπάρχουν στο μουσείο, το 1/8 κάνει ταυτόχρονη χρήση μιας AR εφαρμογής, δηλαδή 125 επισκέπτες. Το δίκτυο και οι εξυπηρετητές θα πρέπει να έχουν μεριμνήσει για την κάλυψη τουλάχιστον 200 αιτημάτων σχεδόν την ίδια χρονική στιγμή.

Ορισμένα παραδείγματα μουσείων που χρησιμοποιούν τεχνολογία επαυξημένης πραγματικότητας είναι τα παρακάτω:

- Μουσείο του Λούβρου, Παρίσι, Γαλλία
- Μητροπολιτικό μουσείο, Νέα Υόρκη
- Μουσείο της σύγχρονης τέχνης, Νέα Υόρκη
- Εθνική Πινακοθήκη, Λονδίνο Αγγλία
- Μουσείο του Πράδο, Μαδρίτη, Ισπανία

4 ΟΡΙΣΜΟΣ ΠΡΟΒΛΗΜΑΤΟΣ ΚΑΙ ΛΥΣΗ

4.1 Ορισμός προβλήματος

Η διαδικασία επίσκεψης σε αρχαιολογικούς χώρους και αξιοθέατα μέχρι στιγμής κρύβει αρκετά προβλήματα. Τα βασικότερα από αυτά αναφέρονται παρακάτω.

Ο επισκέπτης χρειάζεται να έχει μελετήσει (Διαδίκτυο, βιβλία κ.α.) πριν επισκεφτεί ένα αξιοθέατο προκειμένου να γνωρίζει πληροφορίες γι' αυτό. Με αυτόν τον τρόπο θα μπορέσει να αξιοποιήσει και να κατανοήσει καλυτέρα τη νέα οπτική πληροφορία που παίρνει με την επίσκεψη του στο χώρο των αξιοθέατων. Η νέα αυτή πληροφορία, σε συνδυασμό με τις γνώσεις που είχε καταφέρει να συλλέξει, του επιτρέπουν να δημιουργήσει μία ολοκληρωμένη εμπειρία που θα καταφέρει να αποτυπωθεί στη μνήμη του. Η παραπάνω διαδικασία είναι χρονοβόρα και όχι τόσο ευχάριστη για τον επισκέπτη.

Σε περίπτωση που ο επισκέπτης δεν ασχοληθεί με την προεργασία απόκτησης γνώσεων που απαιτείται, τότε η ύπαρξη κάποιου ξεναγού καθίσταται αναγκαία. Διαφορετικά, ενδέχεται να βρεθεί σε μια κατάσταση πλήρους άγνοιας σχετικά με το χώρο που έχει επισκεφθεί. Η λύση, ωστόσο, του ξεναγού δεν είναι πάντα εφικτή σε όλους τους χώρους με αξιοθέατα, καθώς αποτελεί μια κοστοβόρα διαδικασία για τον επισκέπτη και συνεπώς δεν αποτελεί ουσιαστική λύση του προβλήματος.

Αναμφισβήτητα, η σωστή καθοδήγηση και η στοχευμένη μετάδοση πληροφοριών αναφορικά με το εκάστοτε έκθεμα συνεπάγεται αύξηση των γνώσεων του επισκέπτη σε σχέση με τις γνώσεις που είχε πριν την επίσκεψη σε χώρους με αξιοθέατα. Το ποσοστό γνώσεων που μπορεί να συγκρατήσει κάποιος κατά τη διάρκεια μιας επίσκεψης είναι μόλις κατά 30%.

Συνοψίζοντας όλα τα παραπάνω, δημιουργείται το ακόλουθο διάγραμμα ροής, το οποίο παρουσιάζει την επίσκεψη σε χώρους με αξιοθέατα από τη σκοπιά της γνώσης και της διάθεσης του επισκέπτη.



Σχήμα 4.1 Επίσκεψη σε χώρο με αζιοθέατα

4.2 Λύση

Για τη βελτιστοποίηση της παραπάνω διαδικασίας, δημιουργήθηκε εφαρμογή ικανή να δώσει απάντηση σε όλα τα παραπάνω προβλήματα. Η εφαρμογή είναι σε θέση να σαρώσει οποιοδήποτε αξιοθέατο και να παρέχει σε πραγματικό χρόνο πληροφορίες για ο,τιδήποτε βλέπει ο επισκέπτης τη δεδομένη χρονική στιγμή με χρήση τεχνολογιών επαυξημένης πραγματικότητας. Τα πλεονεκτήματα αυτής της λύσης είναι όλα τα παρακάτω:

- Είναι δωρεάν. Η απόκτηση της εφαρμογής δεν στοιχίζει τίποτα για κανένα επισκέπτη.
- Είναι προσβάσιμη σε όλους. Η μονή προϋπόθεση για την απόκτηση της εφαρμογής είναι η κατοχή μιας κινητής συσκευής ανεξαρτήτως λειτουργικού συστήματος (Android/iOS).
- Για την επίσκεψη σε οποιοδήποτε χώρο με αξιοθέατα, δεν απαιτείται εκ των προτέρων καμία γνώση από τον επισκέπτη. Η πληροφορία παρέχεται σε πραγματικό χρόνο μέσω της εφαρμογής.
- Ο χρόνος εκμάθησης μειώνεται αισθητά, καθώς ο επισκέπτης είναι σε θέση να σαρώσει και να λάβει άμεσα πληροφορίες για τα έργα τέχνης που τον ενδιαφέρουν.

- Με την καινοτόμα αυτή διαδικασία επιτυγχάνεται ο συνδυασμός ψυχαγωγίας και μάθησης, καθώς η σάρωση αντικειμένων αποτελεί ενός είδος παιχνιδιού για τον επισκέπτη, ενώ η προβολή πληροφοριών με τεχνολογίες επαυξημένης πραγματικότητας που αντιλαμβάνεται ο χρήστης, διατηρούν αμείωτο το ενδιαφέρον καθ' όλη τη διάρκεια της εκπαίδευσης.
- Οι νέες γνώσεις/πληροφορίες που καταφέρνει να συγκρατήσει ο μέσος επισκέπτης είναι υψηλότερες σε σχέση με εκείνες χωρίς την εφαρμογή, γεγονός που καθιστά την περιήγηση περισσότερο αποδοτική.

Συνοψίζοντας όλα τα παραπάνω, δημιουργείται το ακόλουθο διάγραμμα ροής για την επίσκεψη στο χώρο της Εθνικής Πινακοθήκης με χρήση της εφαρμογής.



Σχήμα 4.2 Επίσκεψη στην Εθνική Πινακοθήκη με χρήση εφαρμογής

Σε αυτό το σημείο, είναι απαραίτητο να αναφερθεί ότι στα πλαίσια της παρούσας διπλωματικής η εφαρμογή αναπτύχθηκε με τρόπο που στοχεύει στην κάλυψη των παραπάνω αναγκών αποκλειστικά για το χώρο της Εθνικής Πινακοθήκης. Η ίδια εφαρμογή μπορεί να λειτουργήσει εξίσου αποτελεσματικά σε οποιαδήποτε πινακοθήκη.

4.3 Εθνική Πινακοθήκη

Η Εθνική Πινακοθήκη αποτελεί ένα από τα σημαντικότερα μουσεία της χώρας. Περικλείει στις συλλογές της περισσότερα από 20.000 έργα ζωγραφικής, γλυπτικής, χαρακτικής και άλλων μορφών τέχνης και αποτελεί το θησαυροφυλάκιο της νεότερης ελληνικής καλλιτεχνικής δημιουργίας, καλύπτοντας μια περίοδο από τα μεταβυζαντινά χρόνια ως τις μέρες μας. Διαθέτει, επίσης, μία αξιόλογη συλλογή δυτικοευρωπαϊκής ζωγραφικής.

Ιδρύθηκε στις 10 Απριλίου 1900 με νόμο, ο οποίος προέβλεπε θέση εφόρου του Ιδρύματος, στην οποία διορίστηκε ο γνωστός ζωγράφος Γεώργιος Ιακωβίδης. Στις 28 Ιουνίου του 1900 θεσμοθετήθηκε ο κανονισμός λειτουργίας της. Οι πρώτες συλλογές της προέρχονταν από το Εθνικό Μετσόβιο Πολυτεχνείο και το Πανεπιστήμιο Αθηνών. Σε αυτές προστέθηκαν αμέσως μετά σημαντικές δωρεές. Το 1954 η Εθνική Πινακοθήκη συγχωνεύτηκε με το κληροδότημα Αλεξάνδρου Σούτζου, από όπου προέρχεται και η διπλή της ονομασία.

Ο θεσμικός της ρόλος είναι η συλλογή, διαφύλαξη, συντήρηση, μελέτη και έκθεση έργων τέχνης, με σκοπό την αισθητική καλλιέργεια του κοινού, τη δια βίου εκπαίδευση μέσα από την τέχνη και την ψυχαγωγία που αυτή προσφέρει, αλλά και την αυτογνωσία των Ελλήνων με τη βοήθεια της ιστορίας της τέχνης, η οποία εκφράζει σε συμβολικό επίπεδο τον εθνικό βίο. [6]

Το κεντρικό κτήριο της Εθνικής Πινακοθήκης, επί της οδού Βασιλέως Κωνσταντίνου 50, έχει κλείσει από το Μάρτιο 2013 λόγω των εργασιών επέκτασης

Οι υπηρεσίες του μουσείου στεγάζονται προσωρινά στους χώρους της Εθνικής Γλυπτοθήκης, στο Άλσος Στράτου στο Γουδή. Το 2021 αναμένεται να ολοκληρωθεί το νέο κτήριο που θα στεγάσει πλέον μόνιμα την Εθνική Πινακοθήκη, γεγονός που επιβεβαιώνεται και από την τωρινή κυβέρνηση της χωράς. Όπως δήλωσε η Λίνα Μενδώνη (Αρχαιολόγος και Υπουργός Πολιτισμού): «εξετάζουμε την εξέλιξη του έργου, ακόμη και στις λεπτομέρειές του, με τις εβδομαδιαίες επισκέψεις μας, καθώς έχουμε μπει πλέον στην τελική ευθεία. Αντιμετωπίσαμε τις δυσκολίες, αλλά και τις καθυστερήσεις που προέκυψαν στην προμήθεια των υλικών λόγω της πανδημίας. Το αυστηρό χρονοδιάγραμμα, που έχουμε θέσει, τηρείται, προκειμένου το κτήριο της Εθνικής Πινακοθήκης να παραδοθεί εγκαίρως και να προχωρήσει το στήσιμο της έκθεσης. Στα εγκαίνια τα οποία έχουν καθοριστεί από τον Πρωθυπουργό για τις 25 Μαρτίου 2021, θα είναι όλα έτοιμα. Η ανακαινισμένη Πινακοθήκη θα τοποθετήσει την Αθήνα στις ευρωπαϊκές πρωτεύουσες, των οποίων οι πινακοθήκες αφηγούνται την ιστορία της σύγχρονης ζωγραφικής των χωρών τους». [7]



Σχήμα 4.3 Νέο κτήριο Εθνικής Πινακοθήκης

4.4 Ανάλυση απαιτήσεων

Για την ομαλή λειτουργία της εφαρμογής είναι απαραίτητη η ικανοποίηση κάποιων βασικών προϋποθέσεων τόσο από τον επισκέπτη όσο και από την Εθνική Πινακοθήκη.

Ο επισκέπτης είναι απαραίτητο να φέρει κατά την περιήγησή του στην Εθνική Πινακοθήκη το κινητό του τηλέφωνο. Το λειτουργικό σύστημα της συσκευής μπορεί να είναι Android ή iOS. Όσον αφορά το hardware της συσκευής, οι ελάχιστες απαιτούμενες προδιαγραφές που πρέπει να πληρούνται είναι: 1GHz CPU, 2GB RAM, 35MB ελεύθερος χώρος μνήμης. Η συσκευή πρέπει επίσης να είναι εξοπλισμένη με τα παρακάτω στοιχεία:

- Οθόνη: Για την προβολή των αποτελεσμάτων της εφαρμογής με χρήση επαυξημένης πραγματικότητας.
- Κάμερα: Για τη συνεχομένη λήψη εικόνων προκειμένου να επιτευχθεί η σάρωση των πινάκων, τουλάχιστον 5Mpixels.
- Κεραία: Για τη σύνδεση της συσκευής στο τοπικό δίκτυο της πινακοθήκης.

Το κτήριο της Εθνικής Πινακοθήκης πρέπει να είναι εξοπλισμένο με τα ακόλουθα:

Πάνελ: Στην είσοδο της πινακοθήκης θα υπάρχει πάνελ το οποίο θα περιέχει το QR code που θα σκανάρουν οι επισκέπτες προκειμένου να πραγματοποιήσουν λήψη της εφαρμογής.



Σχήμα 4.4 QR code για τη λήψη της εφαρμογής

Σαρώνοντας το παραπάνω QR code γίνεται αυτόματη μετάβαση στην ιστοσελίδα λήψης της εφαρμογής. Η ιστοσελίδα βρίσκεται στον παρακάτω σύνδεσμο:

<u>ARt – Λήψη εφαρμογής</u>



Σχήμα 4.5 Ιστοσελίδα λήψης της εφαρμογής

- Διακομιστές: Οι διακομιστές πρέπει να διαθέτουν επαρκή επεξεργαστική ισχύ ώστε να εξυπηρετούν ταυτόχρονα όλα τα αιτήματα των επισκεπτών. Επίσης, θα περιέχουν την ειδικά σχεδιασμένη βάση δεδομένων που απαιτείται για τη λειτουργία της εφαρμογής (περισσότερες πληροφορίες στο κεφάλαιο «Δημιουργία βάσης δεδομένων»).
- Κατάλληλο δίκτυο: Δίκτυο MEC υψηλής ταχύτητας ικανό να καλύψει τις ανάγκες μιας εφαρμογής που χρησιμοποιεί τεχνολογίες επαυξημένης πραγματικότητας.

5 ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΗΣ

Κατά τη διάρκεια υλοποίησης της εφαρμογής, χρησιμοποιήθηκαν τα ακόλουθα εργαλεία:

Εργαλείο	Σκοπός χρήσης
Vuforia Engine	Δημιουργία επαυξημένης πραγματικότητας
Unity Engine	Σχεδιασμός εφαρμογής
Adobe Photoshop	Σχεδιασμός γραφικών εφαρμογής
Marvel App	Σχεδιασμός προτύπου διεπαφής χρήστη
Python	Συλλογή δεδομένων
Java	Δημιουργία Vuforia Cloud Database
C Sharp	Προγραμματισμός λογικής της εφαρμογής στο Unity

Πίνακας 5.1 Εργαλεία ανάπτυζης εφαρμογής

5.1 Πλατφόρμες

5.1.1 Vuforia



Το Vuforia είναι ένα εργαλείο ανάπτυξης λογισμικού επαυξημένης πραγματικότητας (AR) για φορητές συσκευές που επιτρέπει τη δημιουργία εφαρμογών επαυξημένης πραγματικότητας. Χρησιμοποιεί τεχνολογία όρασης υπολογιστή για την αναγνώριση και παρακολούθηση συγκεκριμένων «στόχων» (αναφέρονται αναλυτικά παρακάτω) σε πραγματικό χρόνο.

Αυτή η δυνατότητα εγγραφής εικόνας επιτρέπει στους προγραμματιστές να τοποθετούν και να προσανατολίζουν εικονικά αντικείμενα, όπως τρισδιάστατα μοντέλα και άλλα μέσα (εικόνες, βίντεο κ.α.), σε σχέση με αντικείμενα πραγματικού κόσμου όταν προβάλλονται μέσω της κάμερας μιας κινητής συσκευής. Στη συνέχεια, το εικονικό αντικείμενο παρακολουθεί τη θέση και τον προσανατολισμό της εικόνας σε πραγματικό χρόνο, έτσι ώστε η προοπτική του θεατή στο αντικείμενο να αντιστοιχεί στην προοπτική του στόχου. Με αυτόν τον τρόπο, φαίνεται λοιπόν ότι το εικονικό αντικείμενο είναι μέρος της πραγματικής σκηνής. [2]

Το Vuforia SDK (Software Development Kit / Εργαλεία ανάπτυξης λογισμικού) υποστηρίζει μια ποικιλία τύπων στόχων τόσο δύο διαστάσεων όσο και τριών διαστάσεων, όπως «στόχοι εικόνας χωρίς σήμανση», στόχος 3D μοντέλου και μια μορφή διαστάσεων Fiducial Marker, γνωστού ως VuMark. Πρόσθετα χαρακτηριστικά του SDK περιλαμβάνουν εντοπισμό συσκευών με 6 βαθμούς ελευθερίας και δυνατότητα δημιουργίας και αναδιάρθρωσης συνόλων στόχων μέσω προγραμματισμού κατά το χρόνο εκτέλεσης. Συνοψίζοντας όλα τα παραπάνω, καταλήγουμε στις ακόλουθες κατηγορίες αναγνωρίσιμων στόχων από το Vuforia Engine.

Οι άμεσα αναγνωρίσιμοι στόχοι από το Vuforia Engine είναι οι ακόλουθοι:

• Εικόνα (Image target)

Οι στόχοι εικόνας είναι κοινές φωτογραφίες, ασπρόμαυρες ή πολύχρωμες, κατάλληλες για καλή αντίθεση, πλούσιες σε λεπτομέρειες και χωρίς επαναλαμβανόμενα μοτίβα.



Σχήμα 5.1 Αναγνώριση στόχου εικόνας

• Τρισδιάστατα μοντέλα

Οι στόχοι τρισδιάστατων μοντέλων είναι κοινά αντικείμενα πλούσια σε λεπτομέρειες, χωρίς επαναλαμβανόμενα μοτίβα και με καλή αντίθεση. Οι στόχοι αυτοί μπορεί να είναι αρκετά παράξενα αντικείμενα, υποθέτοντας ότι πληρούνται τα παραπάνω κριτήρια. Αυτό δίνει στον προγραμματιστή μία ποικιλία στόχων ικανών να υποστηρίξουν τις εφαρμογές AR.



Σχήμα 5.2 Αναγνώριση τρισδιάστατου μοντέλου

Επίπεδη επιφάνεια

Πρόκειται για οποιαδήποτε επίπεδη επιφάνεια. Κατά την αναγνώριση της προβάλλονται διαφορά αντικείμενα πάνω σε αυτήν, όπως για παράδειγμα στο Σχ. 5.3.



Σχήμα 5.3 Στόχος επίπεδης επιφάνειας

• VuMarks

Τα VuMarks είναι ο παλαιότερος τύπος στόχων, αλλά χρησιμοποιείται ακόμα σε αρκετές περιπτώσεις. Πρόκειται για ειδικά σχεδιασμένους στόχους που είναι παρόμοιοι των QR codes. Το σχήμα τους μπορεί να ποικίλλει, αλλά σε γενικές γραμμές αποτελείται από ένα μονόχρωμο πλαίσιο με τετράγωνα χαμηλής λεπτομέρειας. Ακόμα κι αν χρησιμοποιούνται ως επί το πλείστον για δευτερεύουσες εφαρμογές, όπως η παροχή απλής διεύθυνσης URL και όχι για AR, τα VuMarks είναι γνωστό παράδειγμα στόχων που χρησιμοποιούνται από κινητές συσκευές. Το κύριο πλεονέκτημά τους είναι ότι, παρά την απλότητά τους, γίνονται εύκολα αντιληπτοί από το Vuforia Engine.



Σχήμα 5.4 Στόχος τύπου VuMark

Εφόσον η δημιουργία της εφαρμογής απαιτεί αναγνώριση εικόνων και προβολή επαυξημένης πραγματικότητας, η επιλογή του Vuforia Engine αποτελεί ένα κατάλληλο περιβάλλον για την ανάπτυξη της εφαρμογής.

5.1.2 Unity **{**) unity



To Unity είναι μια μηχανή παιχνιδιών πολλαπλών εφαρμογών που αναπτύχθηκε από την Unity Technologies και κυκλοφόρησε για πρώτη φορά τον Ιούνιο του 2005 στο Worldwide Developers Conference της Apple Inc. ως αποκλειστική μηχανή παιχνιδιών Mac OS X. Από το 2018, το λογισμικό είχε επεκταθεί για να υποστηρίξει περισσότερες από 25 πλατφόρμες. Το λογισμικό μπορεί να χρησιμοποιηθεί για τη δημιουργία τρισδιάστατων, δισδιάστατων, εικονικής πραγματικότητας και επαυξημένης πραγματικότητας παιχνιδιών, καθώς και προσομοιώσεων και άλλων εμπειριών. [1] Ξεκινώντας από τη χρήση του στο περιβάλλον των βιντεοπαιχνιδιών, το συγκεκριμένο λογισμικό έχει υιοθετηθεί από πολλές βιομηχανίες φιλμ και αυτοκινήτων καθώς και από άλλους επιστημονικούς κλάδους, όπως είναι η αρχιτεκτονική και η μηχανική.

Αρκετές σημαντικές εκδόσεις του Unity έγουν κυκλοφορήσει από την πρώτη κυκλοφορία. Η τελευταία σταθερή έκδοση είναι η «2020.2.0», κυκλοφόρησε τον Δεκέμβριο του 2020. Η έκδοση που χρησιμοποιήθηκε για τη δημιουργία της εφαρμογής είναι η 2020.1.11f1.

5.1.3 Unity assets store Sunity Asset Store

To Unity Asset Store είναι ένα ηλεκτρονικό κατάστημα ενσωματωμένο στο Unity. Πρόκειται για μια αναπτυσσόμενη βιβλιοθήκη με πακέτα. Τόσο η Unity Technologies όσο και τα μέλη της κοινότητας δημιουργούν αυτά τα πακέτα και τα δημοσιεύουν στο κατάστημα. Υπάρχουν διάφοροι τύποι πακέτων, από κινούμενα σχέδια και τρισδιάστατα μοντέλα έως ολόκληρα παραδείγματα έργων, φροντιστήρια και επεκτάσεις προγραμμάτων επεξεργασίας. Κάποια από αυτά τα πακέτα είναι δωρεάν και κάποια αλλά επί πληρωμή. Το Vuforia Engine αποτελεί ένα τέτοιο πακέτο, βρίσκεται εντός του Unity assets store και διατίθεται δωρεάν.

Οι λόγοι που συνετέλεσαν στην επιλογή του Unity ως πλατφόρμα ανάπτυξης της εφαρμογής σχετίζονται σε μεγάλο βαθμό με την ευελιξία και την παραγωγικότητα που προσφέρει στο χρήστη. Συγκεκριμένα:

- Το Unity δίνει τη δυνατότητα εφόσον ολοκληρωθεί η εφαρμογή, να εξαχθεί το τελικό • αποτέλεσμα και στις δύο πλατφόρμες (και όχι μόνο). Ο στόχος ήταν η δημιουργία μιας εφαρμογής για όλα τα κινητά τηλεφωνά, τόσο εκείνα με λειτουργικό σύστημα Android όσο και εκείνα με λειτουργικό σύστημα iOS. Κάτι τέτοιο είναι αναγκαίο προκειμένου όλοι οι πιθανοί επισκέπτες της Εθνικής πινακοθήκης να μπορούν να εξυπηρετηθούν.
- Η γλώσσα προγραμματισμού που χρησιμοποιεί το Unity είναι η C# (C Sharp). • ευελιξία Πρόκειται αρκετά σύγχρονη γλώσσα που επιτρέπει για μια προγραμματιστικών κινήσεων και πληθώρα δυνατοτήτων στο πλαίσιο της ανάπτυξης εφαρμογής. Στο πέρας του χρόνου, θα συνεχίσει να αναπτύσσεται και το πρόγραμμα θα παραμείνει συντηρήσιμο για αρκετό καιρό ακόμα.
- Το Vuforia Engine συμπληρώνει αποτελεσματικά τη λειτουργία του Unity και η εισαγωγή του είναι αρκετά απλή, κατεβάζοντας το πακέτο από το Unity assets store, εντελώς δωρεάν.

5.1.4 Adobe Photoshop



Top Adobe Photoshop, ή απλά Photoshop, είναι πρόγραμμα επεξεργασίας γραφικών που αναπτύχθηκε και κυκλοφόρησε από την Adobe Systems. Αυτή τη στιγμή αποτελεί ηγέτη της αγοράς (market leader) των προγραμμάτων επεξεργασίας εικόνων και είναι το προϊόν - σήμα κατατεθέν-της Adobe Systems. Χαρακτηρίζεται ως «απαραίτητο εργαλείο για τους επαγγελματίες γραφίστες» και θεωρείται πως προώθησε τις αγορές των Macintosh και στη συνέχεια των Windows.

Δημιουργήθηκε αρχικά το 1988 από τους Thomas και John Knoll. Από τότε, το λογισμικό έχει γίνει το βιομηχανικό πρότυπο όχι μόνο στην επεξεργασία γραφικών, αλλά και στην ψηφιακή τέχνη στο σύνολό της. Το Photoshop μπορεί να επεξεργαστεί και να συνθέσει εικόνες σε πολλαπλά επίπεδα υποστηρίζοντας μάσκες, συνθέσεις άλφα και διάφορα χρωματικά μοντέλα όπως RGB, CMYK, CIELAB, spot color και duotone. Το Photoshop χρησιμοποιεί τις δικές του μορφές αρχείων PSD και PSB για να υποστηρίζει αυτές τις δυνατότητες. Εκτός από τα γραφικά, το Photoshop έχει περιορισμένες δυνατότητες επεξεργασίας, απόδοσης κειμένου και διανυσματικών γραφικών, τρισδιάστατων γραφικών και βίντεο. Το σύνολο χαρακτηριστικών του μπορεί να επεκταθεί με προσθήκες-προγράμματα που αναπτύχθηκαν και διανεμήθηκαν ανεξάρτητα από το Photoshop προσφέρουν νέες ή βελτιωμένες δυνατότητες.

Το σχήμα ονομάτων του Photoshop βασίστηκε αρχικά σε αριθμούς έκδοσης. Ωστόσο, τον Οκτώβριο του 2002 (μετά την καθιέρωση της επωνυμίας Creative Suite), κάθε νέα έκδοση του Photoshop χαρακτηρίστηκε με "CS" μαζί με έναν αριθμό (π.χ. η όγδοη κύρια έκδοση του Photoshop ήταν το Photoshop CS και η ένατη ήταν το Photoshop CS2). Το Photoshop CS3 έως CS6 διανεμήθηκε επίσης σε δύο διαφορετικές εκδόσεις, Standard και Extended. Με την καθιέρωση της δημιουργίας επωνυμίας Creative Cloud τον Ιούνιο του 2013 (και με τη σειρά του, η αλλαγή του επιθήματος "CS" σε "CC"), το σύστημα αδειοδότησης του Photoshop άλλαξε σε αυτό του λογισμικού ως μοντέλο ενοικίασης υπηρεσιών. Ιστορικά, το Photoshop συνδυάστηκε με πρόσθετο λογισμικό όπως το Adobe ImageReady, το Adobe Fireworks, το Adobe Bridge, το Adobe Device Central και το Adobe Camera RAW.

Παράλληλα με το Photoshop, η Adobe αναπτύσσει και δημοσιεύει Photoshop Elements, Photoshop Lightroom, Photoshop Express, Photoshop Fix, Photoshop Sketch και Photoshop Mix. Από τον Νοέμβριο του 2019, η Adobe κυκλοφόρησε επίσης μια πλήρη έκδοση του Photoshop για το iPad και, ενώ αρχικά ήταν περιορισμένη, η Adobe σκοπεύει να φέρει περισσότερες δυνατότητες στο Photoshop για iPad. Συλλογικά, χαρακτηρίζονται ως "Η οικογένεια Adobe Photoshop". [10]

5.1.5 Marvel App Marvel

Το Marvel App είναι μια διαδικτυακή πλατφόρμα δημιουργίας πρωτότυπων σχεδίων και διεπαφών χρήστη. Επιτρέπει τη δημιουργία διαδραστικών εφαρμογών χωρίς τη χρήση προγραμματισμού και κώδικα. Διαθέτει εργαλεία σχεδίασης, πολλαπλά επίπεδα και ανέβασμα σχεδίων. Με αυτό το εργαλείο δημιουργήθηκε το πρωτότυπο διεπαφής χρήστη για την εφαρμογή.

5.2 Γλώσσες προγραμματισμού

Οι κυρίες γλώσσες που χρησιμοποιήθηκαν για την ολοκλήρωση αυτής της εργασίας είναι: Python, Java και C#. Ακολουθεί μια σύντομη περιγραφή και μερικές πληροφορίες για κάθε μια από αυτές.

5.2.1 Python

Η Python είναι διερμηνευόμενη (interpreted), γενικού σκοπού και υψηλού επιπέδου, γλώσσα προγραμματισμού. Ανήκει στις γλώσσες προστακτικού προγραμματισμού (Imperative programming) και υποστηρίζει τόσο το διαδικαστικό (procedural programming) όσο και το αντικειμενοστραφές (object-oriented programming) προγραμματιστικό υπόδειγμα (programming paradigm). Είναι δυναμική γλώσσα προγραμματισμού (dynamically typed) και υποστηρίζει συλλογή απορριμμάτων (garbage collection ή GC).

Δημιουργήθηκε από τον Ολλανδό Γκίντο βαν Ρόσσουμ (Guido van Rossum) στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) το 1989 και κυκλοφόρησε για πρώτη φορά το 1991. Ο κύριος στόχος της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της. Το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα από ότι θα ήταν δυνατόν σε γλώσσες όπως η C++ ή η Java. Διακρίνεται λόγω των πολλών βιβλιοθηκών που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες και για την ταχύτητα εκμάθησης της. Ωστόσο, η διερμηνευόμενη φύση της την καθιστά αργή, σε σχέση με τις μεταγλωττιζόμενες (compiled) γλώσσες όπως η C και η C++. Για αυτόν τον λόγο δεν είναι κατάλληλη για γραφή λειτουργικών συστημάτων.

Οι διερμηνευτές της Python είναι διαθέσιμοι για εγκατάσταση σε πολλά λειτουργικά συστήματα, επιτρέποντας στην Python την εκτέλεση κώδικα σε ευρεία γκάμα συστημάτων. Χρησιμοποιώντας εργαλεία τρίτων, όπως το Py2exe ή το Pyinstaller, ο κώδικας της Python μπορεί να πακεταριστεί σε αυτόνομα εκτελέσιμα προγράμματα για μερικά από τα δημοφιλέστερα λειτουργικά συστήματα. Με αυτό τον τρόπο, επιτρέπουν τη διανομή του βασισμένου σε Python λογισμικού για χρήση σε αυτά τα περιβάλλοντα χωρίς να απαιτείται εγκατάσταση του διερμηνευτή της Python.

Η Python αναπτύσσεται ως ανοικτό λογισμικό (open source) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License, η οποία είναι συμβατή με την GPL. Το όνομα της γλώσσας προέρχεται από την ομάδα των Άγγλων κωμικών Μόντυ Πάιθον και δεν έχει καμιά σχέση με το φίδι πύθωνα, παρότι το λογότυπό της παραπέμπει σε κάτι τέτοιο.

5.2.2 Java

Η γλώσσα προγραμματισμού Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού, δηλαδή έχει ως βάση της κλάσεις οι οποίες αποτελούν αντικείμενα. Η κεντρική ιδέα του αντικειμενοστραφούς προγραμματισμού είναι ότι το αντικείμενο ξέρει να κάνει μια ενέργεια για τον εαυτό του και όχι πως το πρόγραμμα ξέρει πως να κάνει μια εργασία για το αντικείμενο. Έχει σχεδιαστεί με σκοπό να υπάρχουν όσο το δυνατό λιγότερες υλοποιήσεις εξαρτήσεων και να μπορεί να τρέχει σε όλες τις πλατφόρμες χωρίς να χρειάζεται η επαναμεταγλώττιση του κώδικα. Οι εφαρμογές που είναι γραμμένες στην Java μπορούν να τρέχουν σε οποιαδήποτε εικονική μηχανή Java (Java Virtual Machine – JVM), χωρίς να εξαρτάται από το λειτουργικό σύστημα ή την αρχιτεκτονική του εκάστοτε συστήματος.

Η Java μοιάζει αρκετά με τη γλώσσα προγραμματισμού C++, έχοντας όμως λιγότερες lowlevel δυνατότητες. Αυτός είναι ο λόγος που η Java απαιτεί περισσότερο χρόνο και μνήμη από τη C++. Παρόλα αυτά, γίνονται προσπάθειες βελτίωσης, με σκοπό να μειωθούν ο χρόνος εκτέλεσης και οι απαιτήσεις μνήμης. Τα αντικείμενα στην Java δημιουργούνται με την κλήση της new. Όμως, δεν υπάρχει άμεσος τρόπος να καταστραφούν αυτά τα αντικείμενα ή να αποδεσμεύσουν τη μνήμη που καταλαμβάνουν. Αυτό γίνεται αυτόματα μέσω της συλλογής σκουπιδιών (garbage collection). Η Java χρησιμοποιεί αυτή τη λειτουργία, ώστε να διαχειρίζεται τη μνήμη καθ' όλη τη διάρκεια ζωής ενός αντικειμένου. Στη C++, όπου δεν υπάρχει συλλέκτης σκουπιδιών, είναι υποχρέωση του προγραμματιστή να καταστρέψει τα αντικείμενα και να αποδεσμεύσει τη μνήμη που αυτά κατέχουν. Αν δε γίνει αυτό, είναι εξαιρετικά πιθανή η υπερχείλιση μνήμης. Ο προγραμματιστής καθορίζει ποια αντικείμενα δημιουργούνται και η Java, μέσω του συλλέκτη σκουπιδιών, είναι υπεύθυνη να καταστρέψει τα αντικείμενα όταν δε χρησιμοποιούνται ελευθερώνοντας μνήμη.

Ένα από τα σημαντικότερα πλεονεκτήματα της Java είναι οι βιβλιοθήκες που υπάρχουν, παρέχοντας στους προγραμματιστές πολλές από τις κοινές λειτουργίες που χρησιμοποιούνται συχνά, έτοιμες για χρήση. Επιπλέον, προσφέρονται βιβλιοθήκες για κάποιες λειτουργίες οι οποίες είναι άκρως εξαρτώμενες από το λειτουργικό σύστημα ή την αρχιτεκτονική του συστήματος, όπως πρόσβαση στο δίκτυο ή σε αρχεία. Αυτός είναι ο λόγος, όπως αναφέρθηκε και προηγουμένως, για τον οποίο η Java μπορεί να τρέχει με τον ίδιο τρόπο σε οποιοδήποτε σύστημα χωρίς να χρειάζεται αλλαγή στον κώδικα και επαναμεταγλώττιση. Παρά το γεγονός ότι δεν είναι δυνατό να μετρηθούν οι προγραμματιστές για κάθε γλώσσα προγραμματισμού, η Java θεωρείται μια από τις περισσότερο διαδεδομένες και ευρέως χρησιμοποιούμενες γλώσσες προγραμματισμού. [4]

5.2.3 C# (C Sharp) C*



Το όνομα C sharp εμπνεύστηκε από μια μουσική σημειογραφία. Εξαιτίας διαφόρων τεχνικών περιορισμών προβολής και επειδή το σύμβολο sharp # δεν υφίσταται στα απλά πληκτρολόγια χρησιμοποιήθηκε το αποτέλεσμα του συνδυασμού shift + 3 (#) για την αναπαράσταση της γλώσσας. Η συγκεκριμένη κατάληξη έχει χρησιμοποιηθεί από αρκετές ακόμα γλώσσες της .NET όπως η A# η F# και η Eiffel#.

Η C# είναι μια γλώσσα προγραμματισμού Η/Υ. Δημιουργήθηκε από την Microsoft μέσα από την πλατφόρμα .NET και αργότερα αναγνωρίστηκε επισήμως από την Ecma (ECMA-334) και την ISO (ISO/IEC 23270:2018). Αποσκοπεί στην εισαγωγή μιας απλής αντικειμενοστρεφούς γλώσσας για γενική χρήση και είναι μια από τις γλώσσες προγραμματισμού που δημιουργήθηκαν για την Κοινή Υποδομή Γλώσσας (Common Language Infrastructure). Ο υπεύθυνος της ομάδας που διαχειρίζεται την γλώσσα ονομάζεται Άντερς Χάιλσμπεργκ. Στις 15 Αυγούστου 2012 κυκλοφόρησε η έκδοση 5.0 η οποία είναι η περισσότερο πρόσφατη μέχρι σήμερα. Αυτή είναι και η έκδοση που χρησιμοποιήθηκε για τη δημιουργία της εφαρμογής. [5]

6 ΣΧΕΔΙΑΣΜΟΣ ΕΦΑΡΜΟΓΗΣ

6.1 Αρχική σκέψη

Βλέποντας τα χιλιάδες έργα τέχνης που βρίσκονται σήμερα στην Εθνική Πινακοθήκη, σε συνδυασμό με την προσπάθεια του κράτους (σε συνεργασία με την Ευρωπαϊκή Ένωση) να δημιουργηθεί ένα νέο και πλήρες εξοπλισμένο κτήριο το οποίο θα στεγάζει από εδώ και πέρα αυτή τη μοναδική καλλιτεχνική κληρονομιά, δημιουργήθηκε η ιδέα για το σχεδιασμό μιας εφαρμογής για κινητά τηλεφωνά που θα βοηθούσε οποιοδήποτε επισκέπτη να απολαύσει το ταξίδι του εντός της Εθνικής Πινακοθήκης. Μια τέτοια καινοτομία που αξιοποιεί στο έπακρο πληθώρα τεχνολογικών επιτευγμάτων μετατρέπει μια στείρα διαδικασία απόκτησης γνώσης σε μια ευχάριστη εμπειρία που ταυτόχρονα βοηθάει τον επισκέπτη να αποκομίσει το μέγιστο όφελος από την επίσκεψης.

Αξίζει να σημειωθεί πως ο ημερήσιος μέσος ορός επισκεπτών είναι αρκετά υψηλός σύμφωνα με τα δεδομένα που αντλούνται από της υπηρεσία "Google Maps", γεγονός που προμηνύει ότι η δημιουργία μιας τέτοιας εφαρμογής θα βοηθήσει αρκετό κόσμο.

6.2 Στόχος

Ο απώτερος σκοπός ενός τέτοιου εγχειρήματος είναι να ξεφύγει ο επισκέπτης από τους περιορισμούς μιας απλής περιήγησης στο χώρο της Πινακοθήκης, ώστε να μπορέσει να μάθει περισσότερες πληροφορίες για το έκθεμα που έχει κάθε φορά μπροστά του. Κάτι τέτοιο θα έπρεπε να γίνει χωρίς τη χρήση κάποιου επιπλέον εξοπλισμού. Για το λόγο, αυτό κρίθηκε αναγκαία η ανάπτυξη μιας εφαρμογής για κινητά τηλεφωνά, συσκευές που αποτελούν αναπόσπαστο κομμάτι κάθε επισκέπτη κατά τη διάρκεια της περιήγησης του στην Εθνική Πινακοθήκη.

Η εφαρμογή είναι σε θέση να αναγνωρίζει σε πραγματικό χρόνο τον πίνακα που σαρώνει ο χρήστης μέσω της κάμερα του κινητού τηλεφώνου και στη συνέχεια να προβάλλει με χρήση επαυξημένης πραγματικότητας ένα τρισδιάστατο πάνελ, πάνω στο οποίο αναγράφονται συγκεκριμένες πληροφορίες για τον πίνακα, όπως ο τίτλος, μια σύντομη περιγραφή, το όνομα του δημιουργού και οι διαστάσεις. Αυτές οι πληροφορίες αποτελούν ένα αρκετά καλό σύνολο, ώστε ο επισκέπτης να είναι σε θέση να γνωρίζει πλέον το έργο τέχνης που έχει την οποιαδήποτε στιγμή μπροστά του. Σε περίπτωση όπου επιθυμεί να λάβει ακόμα περισσότερες πληροφορίες, στο πάνελ που εμφανίζεται, υπάρχει ένα πλήκτρο το οποίο ονομάζεται «Περισσότερες πληροφορίες». Πατώντας το, οδηγείται αυτόματα σε διαδικτυακή ιστοσελίδα που αφορά το συγκεκριμένο πίνακα, πάντα εντός της επίσημης ιστοσελίδας της Εθνικής Πινακοθήκης. Ολοκληρώνοντας την ανάγνωση όλων των πληροφοριών, μπορεί να κλείσει το πάνελ και να συνεχίσει να σαρώνει το χώρο προκειμένου να επαναλάβει τη διαδικασία για επόμενο πίνακα ζωγραφικής. Κάθε επισκέπτης μπορεί να σαρώσει όσους πίνακες επιθυμεί.

Το πλέον ενδιαφέρον σημείο κατά τη διάρκεια χρήσης της εφαρμογής είναι ότι οι πληροφορίες δεν εμφανίζονται σαν απλό κείμενο στην οθόνη του χρήστη, αλλά σαν ένα τρισδιάστατο αντικείμενο το οποίο μοιάζει να "αιωρείται" σε θέση πάντα σχετική με την τοποθεσία του πίνακα στον πραγματικό χώρο. Με αυτό τον τρόπο, επιτυγχάνεται η λήψη περισσότερης πληροφορίας από τον επισκέπτη, αλλά με ένα τρόπο πρωτότυπο και μοναδικό, γεγονός που παρακινεί τον επισκέπτη να μάθει ακόμα περισσότερα, χωρίς να αισθάνεται ότι κουράζεται ή ότι βαριέται. Με τον τρόπο αυτό, επιτυγχάνεται ο συνδυασμός μάθησης και ψυχαγωγίας, ολοκληρώνοντας επιτυχώς το στόχο δημιουργίας της εφαρμογής.

Συνοψίζοντας όλα τα παραπάνω, μέσα στην πραγματικότητα που αντιλαμβάνεται ο επισκέπτης κατά τη διάρκεια περιήγησής του στην Εθνική Πινακοθήκη, η εφαρμογή έρχεται να προσθέσει μια δεύτερη εικονική πραγματικότητα, επαυξάνοντας την ήδη υπάρχουσα με τη δημιουργία πληροφοριών σε τρισδιάστατων πάνελ (επαυξημένη πραγματικότητα).

6.3 Αναγνώριση πινάκων

Το πλέον απαιτητικό τμήμα κατά τη διαδικασία υλοποίησης της εφαρμογής είναι η εύρεση αποτελεσματικού τρόπου αναγνώρισης κάθε πίνακα ξεχωριστά, ανάμεσα στους χιλιάδες που διαθέτει η Εθνική Πινακοθήκη. Για το λόγο αυτό, κάθε πίνακας απαιτείται να έχει ένα μοναδικό στοιχείο, το οποίο θα τον κάνει να ξεχωρίζει από όλους τους υπολοίπους που βρίσκονται στον ίδιο χώρο.

Η αρχική σκέψη ήταν να δημιουργηθεί ένας μοναδικός κωδικός VuMark (προσιδιάζει σε QR code αναγνωρίσιμο από το Vuforia) για κάθε πίνακα και να τοποθετηθεί σε μορφή

αυτοκόλλητου διπλά του, με τη συναίνεση της Εθνικής Πινακοθήκης . Όμως, κάτι τέτοιο δεν αποτελεί λύση, για τους παρακάτω λογούς:

- Απαιτεί τη δημιουργία πολλών χιλιάδων QR codes στη μορφή αυτοκόλλητων.
- Η τοποθέτησή τους και η συσχέτισή τους με τη βάση δεδομένων είναι μια αρκετά χρονοβόρα διαδικασία.
- Η ύπαρξη των QR codes υποβαθμίζει την αισθητική του χώρου και το τελικό αποτέλεσμα.

Αναπόφευκτα, έπρεπε να οριστεί ένα άλλο μέσο ταυτοποίησης του κάθε πίνακα. Το στοιχείο εκείνο που καθιστά κάθε πίνακα μοναδικό είναι η εμφάνιση του, δηλαδή ο,τιδήποτε αυτός απεικονίζει. Η εφαρμογή, λοιπόν, εκμεταλλεύεται τη δυνατότητα από το πακέτο της Vuforia Engine και έχει ορίσει ως «στόχο» την εικόνα κάθε πίνακα. Συνεπώς, είναι σε θέση να διαφοροποιήσει, να αναγνωρίσει και να ταυτοποιήσει οποιονδήποτε πίνακα, χρησιμοποιώντας ο,τιδήποτε απεικονίζει κάθε πίνακας, χωρίς να απαιτείται κάποιο επιπλέον μέσω ταυτοποίησης.

6.4 Ονομασία της εφαρμογής

Η εφαρμογή ονομάζεται "ARt". Πήρε το όνομα αυτό επειδή σχετίζεται με τους πίνακες της Εθνικής Πινακοθήκης και κατ' επέκταση με την τέχνη. Όπως λοιπόν είναι γνωστό σε όλους, ο αγγλικός όρος που χρησιμοποιείται για να αποδώσουμε την έννοια της τέχνης είναι η λέξη "art". Όμως τα δύο πρώτα γράμματα είναι κεφαλαία, κάτι που δεν έχει επιλεγεί τυχαία.

Η εφαρμογή προκειμένου να προβάλει την πληροφορία για κάθε πίνακα της Εθνικής Πινακοθήκης χρησιμοποιεί τεχνικές επαυξημένης πραγματικότητας. Στα Αγγλικά χρησιμοποιείται ο όρος "Augmented Reality", έχοντας συντομογραφία AR. Για το λόγο αυτό, χρησιμοποιήθηκαν τα δύο πρώτα γράμματα της λέξης art κεφάλαια, υποδηλώνοντας έτσι τη χρήση της τεχνολογίας επαυξημένης πραγματικότητας (Augmented Reality).

Η επιλογή της ονομασίας ARt, αποτέλεσε με τη σειρά της πηγή έμπνευσης για το λογότυπο της εφαρμογής.

6.5 Λογότυπο

Τίποτα σε αυτή την εφαρμογή δεν έχει επιλεγεί τυχαία. Το λογότυπο που χρησιμοποιήθηκε για την εφαρμογή είναι το πρώτο γράμμα της Αγγλικής αλφάβητου, το "A", προερχόμενο από το πρώτο γράμμα του ονόματος της, "ARt". Ένα αρκετά απλό λογότυπο, που, εξαιτίας αυτού του χαρακτηριστικού, μπορεί να μείνει στη μνήμη των χρηστών.

Έχει επιλεγεί πράσινο χρώμα ως φόντο, καθώς ένα τέτοιο χρώμα ξεχωρίζει πολύ εύκολα ανάμεσα σε άλλες εφαρμογές που μπορεί να έχει ο χρήστης στο κινητό του τηλέφωνο, γεγονός που καθιστά την εφαρμογή ως κάτι διαφορετικό. Επίσης, το πράσινο χρώμα προσδίδει ένα χαρακτήρα που κρύβει ζωηράδα και ευαισθησία. Παρόμοια στρατηγική έχει χρησιμοποιηθεί και από άλλες εταιρείες, όπως Spotify, BP, Starbucks, android, WhatsApp κ.α. που χρησιμοποιούν πράσινο χρώμα στα λογότυπά τους. Το χρώμα που έχει το γράμμα "A" είναι μαύρο προκειμένου να επιτευχθεί αντίθεση με το πράσινο του φόντου. Η γραμματοσειρά που έχει επιλεγεί, μοιάζει με μονοκοντυλιά από πινέλο, γεγονός που παραπέμπει σε ζωγραφική. Τέλος, προστέθηκε μια πολύ μικρή σκιά από τη δεξιά πλευρά στο γράμμα "A" θέλοντας να αποδώσει ότι το λογότυπο φωτίζεται.

Το σχήμα είναι κυκλικό αντί για τετράγωνο, μια τάση που εμφανίζεται στις περισσότερο πρόσφατες εφαρμογές για κινητά τηλεφωνά, αποδίδοντας έτσι ένα σύγχρονο χαρακτήρα. Συνοψίζοντας όλα τα παραπάνω, οδηγούμαστε στο ακόλουθο αισθητικό αποτέλεσμα του Σχ. 6.1.



Σχήμα 6.1 Λογότυπο ARt εφαρμογής

6.6 Διαδικασία εύρεσης της καταλληλότερης διεπαφής χρήστη

Πριν ξεκινήσει η οποιαδήποτε υλοποίηση και ο οποιοσδήποτε προγραμματισμός της εφαρμογής, είναι αναγκαίο να δημιουργηθεί η κατάλληλη διεπαφή χρήστη. Κύριο μέλημα αποτελεί η δημιουργία μιας διεπαφής χρήστη, αρκετά εύκολης στην κατανόηση. Είναι πολύ σημαντικό οι επισκέπτες, κατεβάζοντας την εφαρμογή στο κινητό τους τηλέφωνο, να μπορούν, χωρίς να χρειαστεί να διαβάσουν τις οδηγίες χρήσης, να είναι σε θέση να τη χρησιμοποιήσουν. Επιπλέον, η διεπαφή χρήστη πρέπει να ανταποκρίνεται στην εμφάνιση που διέπει μία τόσο καινοτόμα και σύγχρονη εφαρμογή, χωρίς ο χρήστης να αισθάνεται ότι έχει εγκαταστήσει στο κινητό του τηλέφωνο κάτι παλιό.

Για να εξασφαλιστούν οι παραπάνω απαιτήσεις, δημιουργήθηκε αρχικά ένα πρωτότυπο της διεπαφής στη διαδικτυακή πλατφόρμα του ιστότοπου "www.marvelapp.com". Με την ολοκλήρωση του μοντέλου, δημιουργήθηκε ένα σύνδεσμος ο οποίος επιτρέπει στους χρήστες την είσοδό τους στην πλατφόρμα προκειμένου να δοκιμάζουν τη συγκεκριμένη διεπαφή χρήστη.

Το πρωτότυπο της διεπαφής χρήστη βρίσκεται στον παρακάτω σύνδεσμο:

<u>ARt - Πρωτότυπο διεπαφής χρήστη</u>

Στην συνέχεια, αυτός ο σύνδεσμος στάλθηκε σε σαράντα άτομα διαφορετικής ηλικίας που προθυμοποιήθηκαν να δοκιμάσουν τη διεπαφή και να εκφράσουν τη γνώμη τους. Για το σκοπό αυτό, δημιουργήθηκε ερωτηματολόγιο με χρήση των φορμών της Google. Ειδικότερα μετά τη δοκιμή, τέθηκαν οι ακόλουθες ερωτήσεις:

Αφού δοκιμά (<u>https://marv</u>	σετε τ elapp.c	ο πρω com/p	Τότυπ rototy	Ο της pe/fhf:	ΟΙ εφαρμ <u>3791</u>)	ΕΙΙ ογής '	αφ _{ARt"}	ης	хр	noi	η
συμπληρώστ Ευχαριστώ π	ε το πο ολύ νιο	αρακά α το γι	τω ερι ρόνο σ	ωτημα ας.	τολόγ	ιο.					
* Υποχρεωτι	κό πεί	δίο		umuditi 🕊 kal S							
Πόσο πιθαν επισκέπτες	νό είνα της Ε	αι να τ Θνική	τροτε ς Πινα	ίνατε ακοθή	דη συ וκης; *	γκεκρ	οιµέ∨η	ιεφαι	oμoγŕ	και σε	άλλους
	1	2	3	4	5	6	7	8	9	10	
Απίθανο	0	0	0	0	0	0	0	С	C	0	Σίγουρα
Πόσο ικανο	ποιημ	ένοι ε	ίστε ό	όσον α	αφορά	א דח∨	ευκολ	ία χρι	ήσης ΄	της εφι	αρμογής; *
	1	2	3	4	5	6	7	8	9	10	
Καθόλου	0	0	0	0	0	0	0	0	0	0	Πάρα πολύ

Σχήμα 6.2 Ερωτηματολόγιο διεπαφής χρήστη ARt
Το παραπάνω ερωτηματολόγιο βρίσκεται στον ακόλουθο σύνδεσμο:

<u>ARt - Ερωτηματολόγιο διεπαφής χρήστη</u>

Αρχικά, οι βαθμολογίες ήταν χαμηλές (μέσος όρος 7,5) και η ερώτηση «Πως θα μπορούσε να βελτιωθεί η διεπαφή χρήστη κατά τη γνώμη σας;» ήταν συμπληρωμένη. Αξιοποιώντας την ανατροφοδότηση, πραγματοποιήθηκε μια σειρά αλλαγών και βελτιώσεων. Η διαδικασία επαναλαμβάνονταν μέχρις ότου οι αξιολογήσεις έφτασαν σε βαθμολογία πάνω από εννέα κατά μέσο όρο στο 97% όλων εκείνων που δοκίμασαν την εφαρμογή και δεν υπήρχαν άλλες προτάσεις βελτίωσης. Τότε οριστικοποιήθηκε η διεπαφή που θα χρησιμοποιηθεί.



Σχήμα 6.3 Αποτελέσματα βαθμολογιών ανά σχεδιασμό διεπαφής χρήστη



Σχήμα 6.4 Τελικά αποτελέσματα βαθμολογιών κατά την οριστικοποίηση διεπαφής

Όλα τα γραφικά και όλα τα εικονίδια που έχουν χρησιμοποιηθεί, σχεδιάστηκαν στο Photoshop. Τίποτα δεν έχει αντιγράφει από έτοιμο υλικό που υπάρχει διαθέσιμο στο Διαδίκτυο. Ορισμένα από τα σχέδια που έγιναν στο Photoshop, παρατίθενται παρακάτω:



Σχήμα 6.5 Εικονίδια εφαρμογής ARt

7 ΔΙΕΠΑΦΗ ΧΡΗΣΤΗ (USER INTERFACE) ΚΑΙ ΤΡΟΠΟΣ ΛΕΙΤΟΥΡΓΙΑΣ

Σε αυτήν την ενότητα, θα παρουσιαστεί η διεπαφή χρήστη και θα αναλυθεί ο τρόπος λειτουργίας της εφαρμογής.

Λαμβάνοντας υπόψη τις διορθώσεις που έγιναν κατά τη διαδικασία εύρεσης της καταλληλότερης διεπαφής χρήστη, η διεπαφή αποτελείται από τέσσερις διαφορετικές «οθόνες».

7.1 Αρχική οθόνη

Η αρχική οθόνη παρουσιάζει το κείμενο "National Gallery by ARt", έχει πράσινο χρώμα για φόντο και χρησιμοποιεί άσπρα γράμματα για το κείμενο. Εμφανίζεται αυτόματα, αμέσως μετά το άνοιγμα της εφαρμογής. Παραμένει στην οθόνη για τρία δευτερόλεπτα και στη συνέχεια δίνει τη θέση της στην επόμενη «οθόνη».

Με την εισαγωγή της, επιτυγχάνεται πρωτίστως η ομαλή εισαγωγή του χρήστη στην εφαρμογή, διότι το απευθείας άνοιγμα της κάμερας είναι αρκετά απότομο και πιθανόν να μπερδέψει ορισμένους χρήστες. Όμως, δευτερευόντως δίνει το χρόνο στη συσκευή να προετοιμαστεί για τις διεργασίες που είναι να εκτελεστούν προκειμένου να λειτουργήσει η εφαρμογή.

Η Αρχική οθόνη εμφανίζεται στην παρακάτω εικόνα:



Σχήμα 7.1 Αρχική οθόνη

7.2 Οθόνη Αναμονής

Στην οθόνη αναμονής παρουσιάζεται ένα περιστρεφόμενο εικονίδιο (κάθε 1/16 δευτερολέπτου περιστρέφεται δεξιόστροφα κατά 5 μοίρες) που στόχος του είναι να υποδηλώσει στο χρήστη ότι η εφαρμογή ετοιμάζεται να ξεκινήσει. Η οθόνη αυτή εμφανίζεται όταν οι διεργασίες που απαιτούνται να εκτελεστούν στην κινητή συσκευή του χρήστη για την επιτυχή εκκίνηση της εφαρμογής, δεν έχουν προλάβει να εκτελεστούν στα τρία δευτερόλεπτα που διαρκεί η «Αρχική οθόνη». Όσο ταχύτερος είναι ο επεξεργαστής του κινητού τηλεφώνου τόσο λιγότερες είναι οι πιθανότητες εμφάνισης αυτής της οθόνης στο χρήστη.

Η Οθόνη αναμονής εμφανίζεται παρακάτω:



Σχήμα 7.2 Οθόνη αναμονής

7.3 Οθόνη σάρωσης

Η οθόνη σάρωσης αποτελεί το κύριο μέρος αυτής της εφαρμογής. Σε αυτή την οθόνη εκτελούνται ουσιαστικά όλες οι βασικές λειτουργίες, όπως σάρωση πινάκων και προβολή πληροφοριών με τεχνολογίες επαυξημένης πραγματικότητας. Σε αυτήν την οθόνη γίνεται χρήση της κάμερας του κινητού τηλεφώνου. Σε κάθε χρονική στιγμή προβάλλονται συνέχεια εικόνες που λαμβάνονται από την κάμερα της συσκευής. Οι χρήστες αντιλαμβάνονται αυτή την οθόνη ως εφαρμογή της κάμερας της συσκευής τους, με μόνη διαφορά ότι η εκκίνηση της κάμερας έγινε με σκοπό την αναγνώριση έργων τέχνης και όχι τη λήψη φωτογραφιών ή βίντεο. Σε αυτή την οθόνη η εφαρμογή σκανάρει συνεχόμενα το χώρο για τον εντοπισμό έργων τέχνης στο οπτικό πεδίο της κάμερας.

Μια οριζόντια γραμμή που κινείται ομαλά κατά μήκος της οθόνης, ξεκινώντας από την πάνω πλευρά καταλήγοντας στην κάτω και το αντίστροφο υποδηλώνει στο χρήστη ότι η συσκευή σαρώνει συνεχώς το χώρο και είναι έτοιμη να αναγνωρίσει οποιοδήποτε έργο τέχνης. Στο πάνω μέρος της συγκεκριμένης οθόνης βρίσκεται η μπάρα ενημέρωσης του χρήστη. Η μπάρα αυτή έχει σχεδιαστεί με υψηλή διαφάνεια προκειμένου να επιτευχθεί η εκμετάλλευση όλου του εύρους της οθόνης και χωρίζεται σε τρία τμήματα: το αριστερό, το κεντρικό και το δεξιό.

- Στο αριστερό τμήμα βρίσκεται το κουμπί για την ανακατεύθυνση στις οδηγίες χρήσης της εφαρμογής (θα αναλυθεί παρακάτω)
- Στο κεντρικό τμήμα αναγράφεται το κείμενο "National Gallery"
- Στο δεξιό τμήμα υπάρχει ένα εικονίδιο που παρομοιάζει ένα μικρό σύννεφο. Σε αυτό το τμήμα δίνεται η δυνατότητα στο χρήστη να καταλάβει κάθε πότε η εφαρμογή επικοινωνεί με τον κεντρικό διακομιστή (server), καθώς κάθε φορά που γίνεται διακίνηση δεδομένων (αποστολή ή λήψη) εμφανίζεται ένα βελάκι στο εσωτερικό του σύννεφου.

Μόλις η κάμερα αναγνωρίσει ένα έργο τέχνης, τότε η οριζόντια γραμμή παύει να υπάρχει και εμφανίζεται ένα τρισδιάστατο πάνελ που αναγραφεί ολες τις βασικές πληροφορίες για τον πίνακα. Σε περίπτωση που ο χρήστης επιθυμεί να συνεχίσει με νέα σάρωση, απλά κλείνει το πάνελ πατώντας πάνω δεξιά το κουμπί "Χ" και η εφαρμογή αυτομάτως μπαίνει σε λειτουργία νέας σάρωσης (η γραμμή σάρωσης εμφανίζεται εκ νέου).



Σχήμα 7.3 Οθόνη σάρωσης

7.4 Πάνελ πληροφοριών

Το πάνελ πληροφοριών αποτελεί μέρος της «Οθόνης σάρωσης», αλλά αναφέρεται σαν ξεχωριστή ενότητα, καθώς αποτελεί το αποτέλεσμα μιας επιτυχούς αναγνώρισης. Σε αυτό προβάλλονται όλες οι πληροφορίες για κάθε πίνακα που έχει επιλέξει να σαρώσει ο χρήστης. Για να πραγματοποιηθεί αυτή η διαδικασία, η κάμερα λαμβάνει εικόνες από τον πραγματικό κόσμο και επικοινωνεί συνεχώς με το διακομιστή. Ο διακομιστής, αναλαμβάνει να αναζητήσει αν η εικόνα που έχει λάβει τη δεδομένη χρονική στιγμή, αντιστοιχεί σε κάποια από τις εικόνες που έχει ήδη αποθηκευμένες (Vuforia Cloud Database). Σε περίπτωση αντιστοιχίας, αποστέλλει στη συσκευή τις πληροφορίες που έχει αποθηκευμένες για τη συγκεκριμένη εικόνα (έργο τέχνης). Η συσκευή λαμβάνει τις πληροφορίες αυτές και τις προβάλλει στο πάνελ.

Το πάνελ πληροφοριών έχει δύο τρόπους προβολής:

Τρισδιάστατη προβολή

Όσο ο πίνακας είναι στο οπτικό πεδίο της κάμερας οι διαστάσεις, η περιστροφή και η θέση του πάνελ είναι πάντα σε σχέση με τον πίνακα. Για κάθε χρονική στιγμή ο πίνακας λειτουργεί ως σημείο αναφοράς.

Δισδιάστατη προβολή

Σε περίπτωση που ο πίνακας πάψει να υπάρχει στο οπτικό πεδίο της κάμερας, η προβολή μετατρέπεται αυτομάτως σε δισδιάστατη, καθώς η εφαρμογή χάνει το σημείο αναφοράς που είχε, σύμφωνα με το οποίο καθόριζε την εμφάνιση του πάνελ (μέγεθος, περιστροφή και θέση). Η δισδιάστατη αυτή προβολή φέρνει στο κέντρο της οθόνης το πάνελ, έχοντας πλέον σταθερή θέση και μέγεθος, με τον προσανατολισμό του να είναι παράλληλος στην οθόνη της συσκευής. Μόλις ο πίνακας επιστρέψει στο οπτικό πεδίο της κάμερας, γίνεται εκ νέου αυτόματη επιστροφή σε τρισδιάστατή προβολή

Ένα παραδειγμα δισδιαστατου πανελ, φαινεται στο Σχ. 7.4:



Σχήμα 7.4 Δισδιάστατο πάνελ, κατόπιν σάρωσης του πίνακα "Figures in Space n.1"

Οι πληροφορίες που λαμβάνονται από τη βάση δεδομένων και προβάλλονται στο πάνελ, με τη σειρά που εμφανίζονται, είναι οι εξής:

- Όνομα δημιουργού/καλλιτέχνη/ζωγράφου
 Fertis Elias
- Όνομα έργου τέχνης και χρονολογία δημιουργίας
 Figures in Space n.1, 1963
- Σύντομη περιγραφή και διαστάσεις (σε εκατοστά)
 Oil on hardboard, 90 x 60 cm
- Σύνδεσμος που αντιστοιχεί στην ιστοσελίδα του επιλεγμένου έργου τέχνη, εντός του επίσημου ισότοπου της Εθνικής Πινακοθήκης (πλήκτρο "More Information) https://www.nationalgallery.gr/en/all-collections/collection/collections/figures-in-space-n-1.html

7.5 Βοήθεια χρήστη

Η οθόνη «Βοήθεια χρήστη» εμφανίζεται μόλις ο χρήστης πατήσει πάνω αριστερά στην μπάρα της οθόνης σάρωσης. Είναι μία οθόνη που έχει ως στόχο να βοηθήσει το χρήστη να κατανοήσει περισσότερο τη λειτουργία της εφαρμογής. Η επιστροφή στην οθόνη σάρωσης επιτυγχάνεται μόλις ο χρήστης πατήσει το κουμπί επιστροφής που βρίσκεται πάνω δεξιά. Αναλυτικότερα η «Βοήθεια χρήστη» περιλαμβάνει τις ακόλουθες πληροφορίες:

Περιγραφή:

Μια σύντομη και γενική περιγραφή της εφαρμογής. Το κείμενο που εμφανίζεται ως περιγραφή είναι:

«Σκανάρετε οποιοδήποτε πίνακα της Εθνικής Πινακοθήκης για προβολή περισσοτέρων πληροφοριών για το συγκεκριμένο έργο τέχνης.»

• Οδηγίες χρήσης:

Σύντομες οδηγίες χρήσης για την εφαρμογή, χωρισμένες σε δύο απλά βήματα:

Στοχεύστε με την κάμερα τον πίνακα για προβολή πληροφοριών

 Πατήστε πάνω στις πληροφορίες που εμφανίστηκαν προκειμένου να σαρώσετε εκ νέου

• Πληροφορίες συσκευής:

Αναφορά τεχνικών χαρακτηριστικών της συσκευής στην οποία εκτελείται η εφαρμογή. Μια τέτοια πληροφορία είναι πάρα πολύ χρήσιμη, προκειμένου να γνωρίζουμε κατά πόσο η συσκευή του χρήστη ανταποκρίνεται στις απαιτήσεις της εφαρμογής, επιτρέποντας την ομαλή της λειτουργία.

> Πληροφορίες συστήματος

Ονομα συσκευής Μοντέλο συσκευής Λειτουργικό σύστημα Μνήμη συστήματος Αριθμός επεξεργαστών Συχνότητα επεξεργαστή

Πληροφορίες γραφικών

Μνήμη γραφικών

Όνομα συσκευής γραφικών

Προμηθευτής συσκευής γραφικών

Τύπος συσκευής γραφικών

Έκδοση συσκευής γραφικών

Υποστήριξη πολλαπλών εργασιών

Σύνδεσμοι:

Όροι και προϋποθέσεις χρήσης της εφαρμογής (όροι με τους οποίους κάθε χρήστης της εφαρμογής συναινεί κατά τη διάρκεια χρήσης).

Οι όροι χρήσης βρίσκονται στον ακόλουθο σύνδεσμο: ΑRt - Όροι και προϋποθέσεις

Η οθόνη βοήθειας χρήστη εμφανίζεται στο Σχ. 7.5.

ARt: Help



Description:

Scan any painting of National Gallery for more information on that work of art.

Instructions:

• Point camera at a painting to view info

• Click on the information that appeared to scan again

More Info:

System:

- Device Name: POCO X3 NFC
- Device Model: Xiaomi M2007J20CG
- Operating System: Android OS 10 / API-29
- (QKQ1.200512.002/V12.0.4.0.QJGEUXM)
- System Memory: 5608
- Processor Count: 8
- Processor Frequency: 2304

Graphics:

- Graphics Memory: 2048
- Device Name: Adreno (TM) 618
- Device Vendor: Qualcomm
- Device Type: OpenGLES3
- Device Version: OpenGL ES 3.2
- V@415.0 (GIT@29668c6, I386ada412e,
- 1599643163) (Date:09/09/20)
- MultiThreaded: True

Links:

Terms of Use

Σχήμα 7.5 Οθόνη βοήθειας χρήστη

7.6 Παράδειγμα χρήσης

Προκειμένου να γίνει περισσότερο αντιληπτός ο τρόπος λειτουργίας, ακολουθεί ένα παράδειγμα χρήσης της εφαρμογής. Όλα ξεκινάνε με την επίσκεψη κάποιου χρήστη στην Εθνική Πινακοθήκη. Στη συνέχεια, εκτελούνται με τη σειρά τα ακόλουθα βήματα:

Βήμα 1: Ο χρήστης συνδέεται στο δίκτυο της πινακοθήκης. Αυτομάτως πραγματοποιείται η μετάβαση του στην ιστοσελίδα λήψης της εφαρμογής. Εναλλακτικά σκανάρει με την κάμερα του το QR code που βρίσκεται στην είσοδο της πινακοθήκης.

Βήμα 2: Ο χρήστης κατεβάζει και εγκαθιστά στο κινητό του τηλέφωνο την εφαρμογή "ARt".

Βήμα 3: Ανοίγει την εφαρμογή και ξεκινάει τη σάρωση πίνακα.

Βήμα 4: Η συσκευή επικοινωνεί με το διακομιστή, στέλνοντας την εικόνα που έλαβε μέσω της κάμερας.

Βήμα 5: Ο διακομιστής (*Bλ. κεφ. 8*) αναζητεί στη βάση δεδομένων, έχοντας ως κλειδί αναζήτησης την εικόνα που μόλις στάλθηκε. Η αναζήτηση θα είναι ή επιτυχημένη ή αποτυχημένη. Σε περίπτωση αποτυχίας, δεν εκτελείται καμία ενέργεια. Σε περίπτωση επιτυχίας, επιστρέφει στη συσκευή όλα τα δεδομένα του Image target (εικόνα, πλάτος και μεταδεδομένα)

Βήμα 6: Η συσκευή συλλεγεί τα δεδομένα από το διακομιστή, προβάλλει ένα τρισδιάστατο πάνελ με τις πληροφορίες του πίνακα κάνοντας χρήση επαυξημένης πραγματικότητας και σταματά την επικοινωνία της με το διακομιστή. Μόλις ο χρήστης κλείσει το πάνελ, η διαδικασία επαναλαμβάνεται από το «Βήμα 4».

Στο παρακάτω διάγραμμα εμφανίζεται συνοπτικά η παραπάνω διαδικασία:



Σχήμα 7.6 Διάγραμμα σεναρίου χρήσης της εφαρμογής

8 ΑΠΑΡΑΙΤΗΤΟΙ ΠΟΡΟΙ ΓΙΑ ΤΗ ΛΕΙΤΟΥΡΓΙΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

8.1 Αποκλειστική χρήση δικτύου ΜΕC

Πατώντας το πλήκτρο «Περισσότερες πληροφορίες», ο χρήστης μεταφέρεται στην ιστοσελίδα του πίνακα που έχει σαρώσει προκειμένου να λάβει περισσότερες πληροφορίες για αυτόν. Το πρόβλημα, όμως, με μια τέτοια ενέργεια είναι ότι απαιτείται η σύνδεση του χρήστη στο Διαδίκτυο. Ασφαλώς, η εισαγωγή όλων των πληροφοριών στο πάνελ και η κατάργηση του κουμπιού «Περισσότερες πληροφορίες» θα οδηγήσει στη δημιουργία εκτενών κειμένων. Στόχος είναι η εμφάνιση βασικών πληροφοριών στο πάνελ. Για το λόγο αυτό, η παραμονή του πλήκτρου «Περισσότερες πληροφορίες» κρίνεται αναγκαία. Για τη λύση του παραπάνω προβλήματος, υπάρχουν δύο δυνατές επιλογές:

- Επικοινωνία του κεντρικού διακομιστή στο cloud, προκειμένου να λάβει την απαραίτητη πληροφορία από τον επίσημο ιστότοπο της Εθνικής Πινακοθήκης και στη συνέχεια, διαμοιρασμό της πληροφορίας μέσω του δικτύου MEC στο χρήστη.
- Ενσωμάτωση του ιστότοπου της Εθνικής Πινακοθήκης στη βάση δεδομένων του διακομιστή και απευθείας μετάδοση της απαραίτητης πληροφορίας στους χρήστες.

Η πρώτη λύση απαιτεί τη σύνδεση του κεντρικού διακομιστή στο Διαδίκτυο. Επιπροσθέτως, επιβαρύνει το διακομιστή αυξάνοντας την επεξεργαστική ισχύ, καθώς προστίθενται επιπλέον λειτουργίες, όπως επικοινωνία με το cloud και απόκτηση νέας πληροφορίας. Συνεπώς, η βέλτιστη υλοποίηση είναι η ενσωμάτωση του ιστότοπου της Εθνικής Πινακοθήκης στη βάση δεδομένων του τοπικού διακομιστή. Με αυτόν τον τρόπο, η αποθήκευση και η επεξεργασία της πληροφορίας έρχονται πλησιέστερα στον τελικό χρήστη, μειώνονται οι καθυστερήσεις, ελαχιστοποιείται η επεξεργαστική ισχύς και η διακίνηση της σχετικής πληροφορίας πραγματοποιείται με αποκλειστική χρήση τοπικού δικτύου MEC, χωρίς την ανάγκη εξωτερικής σύνδεσης στο Διαδίκτυο.

8.2 Προαπαιτούμενη εγκατάσταση

Όσον αφορά τον υλικοτεχνικό εξοπλισμό, η λειτουργία της εφαρμογής απαιτεί:

- Η εγκατάσταση διακομιστή
- Η ύπαρξη τοπικού δικτύου ΜΕC υψηλής ταχύτητας

Σύμφωνα με τα δεδομένα του Google Maps, η επισκεψιμότητα στην Εθνική Πινακοθήκη είναι αρκετά υψηλή, ιδιαίτερα τους καλοκαιρινούς μήνες, κατά τους οποίους ο μέγιστος αριθμός επισκεπτών είναι χίλιοι πεντακόσιοι ημερησίως. Το σύστημα πρέπει να είναι σε θέση να υποστηρίζει την αδιάκοπη και απρόσκοπτη λειτουργία της εφαρμογής, διασφαλίζοντας τη μέγιστη ταχύτητα και επίδοση. Λαμβάνοντας υπόψη ότι η μέγιστη τιμή επισκεπτών ανά ημέρα φτάνει τους χιλίους πεντακόσιους, υπάρχει πιθανότητα ύπαρξης χιλίων ατόμων εντός της πινακοθήκης. Από τα χίλια άτομα, οι εξακόσιοι είναι πιθανό να χρησιμοποιούν την εφαρμογή ταυτόχρονα, απαιτώντας επαρκή χωρητικότητα για την κάλυψη των αναγκών επαυξημένης πραγματικότητας. Συνοψίζοντας όλα τα παραπάνω δεδομένα, κρίνεται απαραίτητη η δημιουργία συστήματος που είναι σε θέση να εξυπηρετήσει τουλάχιστον εξακόσιους επισκέπτες ταυτόχρονα. Αυτό μεταφράζεται σε ικανοποιητική επεξεργαστική ισχύ του διακομιστή και υψηλή ταχύτητα δικτύου.

8.2.1 Διακομιστής

Ο διακομιστής βρίσκεται εντός του κτηρίου της Εθνικής Πινακοθήκης με κύριο μέλημα την αποστολή των απαραίτητων δεδομένων στους επισκέπτες. Λαμβάνοντας υπόψη τις προτάσεις της Vuforia Engine, καθώς και τον αριθμό των επισκεπτών, οι ελάχιστες δυνατότητες που πρέπει να διαθέτει προκειμένου να καλύψει τις απαιτήσεις των χρηστών σε μία εφαρμογή επαυξημένης πραγματικότητας, είναι οι παρακάτω:

- Επεξεργαστές 10 X Intel(R) Xeon(R) CPU X5675 @ 3.07GHz
- Μνήμη 24 GB DDR3
- 1ΤΒ σκληρών δίσκων τύπου M2 με συνδεσμολογία σε RAID10

Εντός των σκληρών δίσκων, βρίσκονται δύο ξεχωριστές βάσεις δεδομένων. Η πρώτη περιέχει τον πλήρη ιστότοπο της Εθνικής Πινακοθήκης και η δεύτερη περιέχει όλα τα δεδομένα που αποστέλλονται μετά την επιτυχημένη αναγνώριση στόχου για προβολή στο πάνελ. Όλα τα δεδομένα βρίσκονται εντός του διακομιστή και η κίνησή τους θα περιορίζεται μόνο εντός της πινακοθήκης.

Σε περίπτωση αύξησης της επισκεψιμότητας ή των αναγκών της εφαρμογής, η προσθήκη ενός διακομιστή με ίδια χαρακτηριστικά λύνει άμεσα το πρόβλημα. Σε ένα τέτοιο σενάριο, είναι αναγκαία η ύπαρξη ενός τρίτου διακομιστή με ελάχιστη επεξεργαστική ισχύ, ο οποίος θα έχει ως μοναδική λειτουργία την ίση κατανομή των χρηστών ανά διακομιστή (traffic manager). Με αυτό τον τρόπο, οι διεργασίες που απαιτούνται για την εκτέλεση της εφαρμογής μοιράζονται εξίσου στους διακομιστές.

8.2.2 Δίκτυο MEC

Το MEC είναι ένα τοπικό δίκτυο υψηλής ταχύτητας. Το δίκτυο αυτό θα εκκινεί από ένα κεντρικό modem που συνδέεται στο διακομιστή. Στη συνέχεια, μέσω πολλαπλών κεραιών το δίκτυο επεκτείνεται σε ολόκληρη την έκταση της Εθνικής Πινακοθήκης. Με αυτό τον τρόπο, εξασφαλίζεται η ασύρματη σύνδεση για όλους τους επισκέπτες. Το δίκτυο θα είναι ελεύθερο, χωρίς προστασία κωδικού και προσβάσιμο από όλες τις ηλεκτρονικές συσκευές. Μόλις συνδεθεί κάποιος σε αυτό, θα γίνεται άμεσα ανακατεύθυνση στην ιστοσελίδα λήψης της εφαρμογής (Βλ. Σχ. 4.5). Στη συνέχεια, μέσω αυτού του δικτύου, ο επισκέπτης θα είναι σε θέση να σαρώσει οποιονδήποτε πίνακα και μέσω της εφαρμογής και να λάβει άμεσα τη διαθέσιμη πληροφορία.



Σχήμα 8.1 Τοπικό δίκτυο ΜΕС - Ροή της πληροφορίας

8.3 Υλικοτεχνικός εξοπλισμός και κόστη

Σύμφωνα με μέτρηση που πραγματοποιήθηκε με τη χρήση του Google Earth (Βλ. Σχ. 8.2), το κτήριο της Εθνικής Πινακοθήκης καλύπτει συνολική έκταση χιλίων τετραγωνικών μέτρων. Τα εκθέματα εντός της πινακοθήκης εκτείνονται σε τέσσερις ορόφους. Συνεπώς, το δίκτυο καλύπτει συνολική έκταση τεσσάρων χιλιάδων τετραγωνικών μέτρων.





Σχήμα 8.2 Υπολογισμός εκτάσεως Εθνικής Πινακοθήκης

Η χρήση του λογισμικού επαυξημένης πραγματικότητας κοστολογείται ανάλογα με τη μηνιαία χρήση της εφαρμογής και τις δυνατότητες που προσφέρονται από τον πάροχο του λογισμικού. Για το λόγο αυτό, η Vuforia έχει δημιουργήσει συγκεκριμένα πακέτα χρέωσης:

- Pro
- Agency Package
- Basic + Cloud
- Lab Pack
- University Pack
- Developer

Οι δυνατότητες που παρέχονται ανά πρόγραμμα, φαίνονται στο σχήμα 8.3.

Vuforia Engine	Licenses						
Vuforia Engine offers a	range of license	e and pricing options to	fit the needs of your	project.			
Features	Pro	Agency Package	Basic + Cloud	Basic	Lab Pack	University Pack	Developer
Publishing for Commercial Use	\checkmark	\checkmark	\checkmark	\checkmark	50 Licenses	500 Licenses	Can't Publish
lmage Targets VuMarks Multi-Targets Cylinder Targets	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	100 VuMarks
Object Targets	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark	\checkmark
Model Targets	\checkmark	Add on	$\overline{}$	$\overline{}$	\checkmark	\checkmark	20
Area Targets	\checkmark	Add on	$\overline{}$	$\overline{}$	\checkmark	\checkmark	10
Advanced and External Camera APIS	\checkmark	Add on	$\overline{}$	\bigcirc	\checkmark	\checkmark	\checkmark
Cloud Database	Custom	Custom	10,000 recos per month 100,000 images	-	\checkmark	-	1,000 recos per month 1,000 images
Production Support	\checkmark	\checkmark	$\overline{}$	$\overline{}$	$\overline{}$	$\overline{}$	-

Σχήμα 8.3 Πακέτα χρέωσης της Vuforia Engine

Συνοψίζοντας όλα τα παραπάνω, δημιουργείται ο Πιν. 8.1 με τις δαπάνες (σε εύρω) για τον απαραίτητο εξοπλισμό, συμπεριλαμβανομένων και των εξόδων εγκατάστασης.

Διακομιστής	7,000
Κεντρικό modem	3,000
Κεραίες (1τμχ/30τ.μ.)	4,700
Καλωδίωση	500
Άδεια χρήσης λογισμικού AR	2,500
Εγκατάσταση δικτύου	2,300
Σύνολο	20,000€

Πίνακας 8.1 Κόστη εξοπλισμού και δικτύου

8.4 Χρόνος εγκατάστασης

Για την εγκατάσταση του απαραίτητου εξοπλισμού εντός του κτηρίου της Εθνικής Πινακοθήκης και την ομαλή λειτουργία της εφαρμογής συνιστώνται οι ακόλουθες ενέργειες:

- Δημιουργία βάσης δεδομένων: Αυτοματοποιημένη διαδικασία (Βλ. κεφ. 9.3)
- Παραμετροποίηση διακομιστή: Βάσεις δεδομένων και σύνδεση με modem
- Εγκατάσταση κατάλληλου δικτύου: Καλωδίωση και τοποθέτηση κεραιών

Σύμφωνα με τα παραπάνω, στον Πιν. 8.2 φαίνονται οι εκτιμώμενοι χρόνοι υλοποίησης.

Σύνολο	22 εργάσιμες ημέρες*
Εγκατάσταση κατάλληλου δικτύου	22 ημέρες
Παραμετροποίηση διακομιστή	5 ημέρες
Δημιουργία βάσης δεδομένων	3 ημέρες

Πίνακας 8.2 Χρόνος εγκατάστασης

*Θεωρείται πως υπάρχει παράλληλη εκτέλεση εργασιών.

9 ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ

9.1 Βήματα κατά τη δημιουργία εφαρμογής

Έχοντας αναλύσει σε προηγουμένη ενότητα τη διεπαφή χρήστη της εφαρμογής καθώς επίσης και τον τρόπο λειτουργίας, μπορούμε να προχωρήσουμε στην περιγραφή του τρόπου υλοποίησης της εφαρμογής.

Η δημιουργία της εφαρμογής, μπορεί να χωριστεί σε τρία βασικά στάδια:

- Συλλογή δεδομένων (σε Python)
- Δημιουργία βάσης δεδομένων (σε Java)
- Σχεδιασμός εφαρμογής στο Unity (σε C#)

Σε κάθε στάδιο, χρειάστηκε να δημιουργηθούν διαφορετικά προγράμματα σε διαφορετικές γλώσσες προγραμματισμού, ανάλογα με τις ανάγκες του προβλήματος. Θα ακολουθήσει λεπτομερής ανάλυση όλων των σταδίων, καθώς επίσης και παρουσίαση του κώδικα υλοποίησης της εφαρμογής στις τρεις επόμενες υποενότητες.

9.2 Συλλογή δεδομένων

9.2.1 Ιστότοπος Εθνικής Πινακοθήκης

Η Εθνική Πινακοθήκη, εκτός από τη φυσική έκθεση των έργων τέχνης, διαθέτει και ηλεκτρονική προβολή αυτών. Για την πρόσβαση στα έργα της Εθνικής Πινακοθήκης απαιτείται η πλοήγηση στην επίσημη ιστοσελίδα (<u>www.nationalgallery.gr</u>) και στη συνέχεια μετάβαση στην ενότητα «ΣΥΛΛΟΓΕΣ».



Σχήμα 9.1 Επίσημη ιστοσελίδα Εθνικής Πινακοθήκης

Εκεί βρίσκονται ανά κατηγορία όλα τα διαθέσιμα έργα τέχνης. Για την προβολή όλων επιλέγεται η καρτέλα «Όλες οι κατηγορίες», όπως φαίνεται στο στιγμιότυπο οθόνης που απεικονίζεται στο Σχ. 9.2.



Σχήμα 9.2 Κατηγορίες έργων τέχνης Εθνικής Πινακοθήκης

Στη συνέχεια, εμφανίζονται όλα τα έργα της Εθνικής Πινακοθήκης (Σχ. 9.3). Από εκεί μπορεί να επιλεγεί οποιοδήποτε έργο για προβολή περισσοτέρων πληροφοριών. Στο παράδειγμα του Σχ. 9.3 έχει επιλεγεί το έργο με όνομα «Ο υπηρέτης του Αβραάμ προσφέρει δώρα στην οικογένεια της Ρεβέκκας».



Σχήμα 9.3 Έργα Εθνικής Πινακοθήκης (Ηλεκτρονική συλλογή)



Σχήμα 9.4 Παράδειγμα ιστοσελίδας πίνακα εντός του επίσημου ιστότοπου της Εθνικής Πινακοθήκης

Στο Σχ. 9.4 φαίνονται οι πληροφορίες που παρέχει η Εθνική Πινακοθήκη για κάθε έργο. Αυτές είναι και οι πληροφορίες που θα συλλεχθούν για τη δημιουργία της εφαρμογής.

9.2.2 Διαδικασία συλλογής δεδομένων

Για να ολοκληρωθεί επιτυχώς ένα τέτοιο εγχείρημα, είναι απαραίτητη η συλλογή όλων των δεδομένων για κάθε πίνακα της Εθνικής πινακοθήκης ξεχωριστά.

Τα δεδομένα που είναι αναγκαία να συλλεχθούν εντάσσονται σε δύο κατηγορίες:

• Μια φωτογραφία του πίνακα (πρόσοψη, μπροστινή του όψη)

Η φωτογραφία είναι απαραίτητη, καθώς θα αποθηκευτεί στη βάση δεδομένων του διακομιστή και θα αποτελέσει κλειδί αναζήτησης σε όλα τα αιτήματα που αποστέλλουν οι χρήστες.

Πληροφορίες για τον πίνακα σε μορφή κειμένου

Συγκεκριμένα, για κάθε έργο τέχνης έγινε συλλογή των ακόλουθων στοιχείων:

- Όνομα δημιουργού/καλλιτέχνη/ζωγράφου
- Όνομα έργου τέχνης
- > Σύντομη περιγραφή του πίνακα
- Διαστάσεις πίνακα (σε εκατοστά)
- Σύνδεσμος που αντιστοιχεί στην ιστοσελίδα του έργου τέχνη, εντός του επίσημου ιστότοπου της Εθνικής Πινακοθήκης

Οι πληροφορίες αυτές θα παρουσιαστούν ως το αποτέλεσμα της αναζήτησης στο τρισδιάστατο πάνελ.

Τα έργα που συλλέγονται μέχρι στιγμής στην Εθνική Πινακοθήκη είναι περισσότερα από 20.000, οπότε προκύπτει ένας αρκετά μεγάλος όγκος δεδομένων. Τα δεδομένα αυτά είναι όλα συγκεντρωμένα εντός του επίσημου ισότοπου της Εθνικής Πινακοθήκης. Για να μπορέσουν, όμως, να χρησιμοποιηθούν στην εφαρμογή, είναι απαραίτητη εκτός από τη συλλογή τους, η κατάλληλη οργάνωση σε συγκεκριμένη μορφή, προκειμένου να εισαχθούν στη βάση δεδομένων (*Bλ. κεφ. 9.3*).

Για την επίλυση του παραπάνω προβλήματος δημιουργήθηκε πρόγραμμα σε Python, το οποίο σαρώνει ολόκληρο τον ιστότοπο της Εθνικής Πινακοθήκης, συλλεγεί για κάθε έργο τέχνης όλες τις σχετικές πληροφορίες και τις οργανώνει σε φακέλους.

Οι λόγοι που οδήγησαν στην επιλογή της Python ως γλώσσα προγραμματισμού για το συγκεκριμένο στάδιο είναι οι παρακάτω:

- Διαθέτει πληθώρα έτοιμων βιβλιοθηκών που μπορούν να χρησιμοποιηθούν εύκολα και άμεσα, βοηθώντας την υλοποίηση.
- Είναι δωρεάν, αποτελώντας μια αρκετά ανεπτυγμένη γλώσσα προγραμματισμού.
- Είναι αρκετά απλή γλώσσα προγραμματισμού με αποτέλεσμα τη δημιουργία καθαρότερου κώδικα. Η σύνταξή της είναι κομψή και οι τύποι της δυναμικοί.
- Είναι συμβατή στα περισσότερα λειτουργικά συστήματα (Windows, Linux, Unix, MacOS).
- Διαθέτει αποδοτικές δομές δεδομένων υψηλού επιπέδου.

Ο κώδικας που συλλέγει και οργανώνει όλα τα απαραίτητα δεδομένα για την υλοποίηση της εφαρμογής από τον επίσημο ιστότοπο της Εθνικής Πινακοθήκης παρουσιάζεται ακολούθως.

```
1 jimport urllib.request
       import requests
 3
       import json
 4
      dimport os
 5
       # Constant variables
 6
 7
       NATIONAL_GALLERY_URL = "https://www.nationalgallery.gr/el/sulloges.html"
 8
       PAINTING_URL
                            = "https://www.nationalgallery.gr/el/sulloges/collection/sulloges/"
 9
 10
        # Set Working Directory
                                = os.getcwd().replace("\\", "/") + "/"
        WD
13
        # Set folder and file names
                       = "National_Gallery_Data"
       FOLDER_NAME
       PAINTINGS_FILE = FOLDER_NAME + "/painting_urls.txt"
       WIDTHS_FILE = FOLDER_NAME + "/paintings_width.txt"
18
       # String crop keys
19
       START_STRING = "$scope.paintings "
                       = "$scope.annexes "
       END_STRING
       PAGE_HTML_START = "<meta property=\"og:title\""
       PAGE_HTML_END = "itemprop"
       IMAGE_SEARCH_KEY = "<meta property=\"og:image\" content=\""
PAINTING_TITLE_KEY = "<meta property=\"og:title\" content=\""
PAINTING_DESCRIPTION_KEY = "<meta property=\"og:description\" content=\""</pre>
       PAINTING_WIDTH_KEY = " x "
 26
28
29
        # Print with different styles
      eclass bcolors:
           HEADER = '\033[95m'
           OKBLUE = '\033[94m'
           OKCYAN = '\033[96m'
34
           OKGREEN = '\033[92m'
35
           WARNING = ' \\ 033[93m'
36
           FAIL = '\033[91m'
           ENDC = ' 033[Om'
           BOLD = '\033[1m'
38
 39
           UNDERLINE = ' \\ 033[4m'
40
41
42
       # Creates a file and save the paintings names
43
      def save_names(names_list):
          file = open(WD + PAINTINGS_FILE, "w+", encoding='utf8')
44
           names_str = ""
           for name in names_list:
           names_str += name + "\n"
47
           names_str = names_str[:-1] # Remove last \n
48
49
50
           # Remove broken pages
51
           names_str = names_str.replace(PAINTIN6_URL + "figoures-1.html\n", "") # Broken page
 52
           file.write(names_str)
53
54
            file.close()
55
```

```
56
```

```
58
59
      def get_painting_names():
            print(bcolors.OKBLUE + "Processing the painting urls file..." + bcolors.ENDC)
60
 61
       🖕 🛛 # Category types: 0 = General category, 1 = Paintings, 2 = Sculpture, 3 = Prints,
62
            # 4 = Drawings, 5 = Constructions Installations
            accepted_categories = [0, 1, 3, 4] # Categories 2 and 5 is 3D art
 64
            painting_names = []
            html_string = urllib.request.urlopen(NATIONAL_GALLERY_URL).read().decode("utf8")
            all paintings ison array = html string[html_string.find(START_STRING) + 47:html_string.find(END_STRING) - 13]
67
            all_paintings_json_array = all_paintings_json_array.replace("}, {", "} , {").split(" , ")
            for painting_json_string in all_paintings_json_array:
 68
                painting_json = json.loads(painting_json_string)
 69
                if painting_json["type"] in accepted_categories:
                    painting_names.append(PAINTING_URL + painting_json["alias"] + ".html")
            save_names(painting_names)
            print(bcolors.OKGREEN + "The file with painting names has successfully created\n" + bcolors.ENDC)
 74
 75
        # Gets a substring from a given string which starts from "starting_position" string and ends at character "end_char"
 76
      def get_substring(my_string, starting_position, end_char):
 78
            substring = ""
            for char in my_string[my_string.find(starting_position) + len(starting_position):]:
 79
80
                if char == end_char:
81
                   break
82
                substring += char
83
            return substring
84
85
      def save_image(image_name, url):
 86
            with open(WD + FOLDER_NAME + "/Image_targets/" + image_name + ".jpg", 'wb') as handle:
87
                response = requests.get(url, stream=True)
88
                if not response.ok:
 90
                    print(response)
 91
                for block in response.iter_content(1024):
                    if not block:
                       break
                    handle.write(block)
 97
      def save_metadata(name, text):
 98
            file = open(WD + FOLDER_NAME + "/Metadata/" + name + ".txt", "w+", encoding='utf8')
99
            file.write(text)
            file.close()
      def save_width(width):
            with open(WD + WIDTHS_FILE, "a", encoding='utf8') as widths_file:
                try:
                   widths_file.write(str(round(float(width.replace(",", ".")) / 100, 2)) + "\n")
                except:
108
                   widths_file.write("0.59") # Replace the painting width with the mean of all paintings
            widths_file.close()
```

```
def download_images_and_descriptions():
            print("Begin the extract process...")
            # Open files
            file = open(WD + PAINTINGS_FILE, 'r')
            widths_file = open(WD + FOLDER_NAME + "/paintings_width.txt", "w+")
            counter = 0
            for painting_url in file:
                painting_url = painting_url.rstrip("\n") # Remove new line char
                painting_name = get_substring(painting_url, PAINTING_URL, ".")
                print("Processing \"" + bcolors.WARNING + painting_name + bcolors.ENDC + "\"")
                trv:
                    html_string = urllib.request.urlopen(painting_url).read().decode("utf8")
                    html_string = html_string[html_string.find(PAGE_HTML_START):html_string.find(PAGE_HTML_END)] # Crop
                    painting_title = get_substring(html_string, PAINTING_TITLE_KEY, ">")[:-2]
                    painting_description = get_substring(html_string, PAINTING_DESCRIPTION_KEY, ">")[:-2]
                    painting_width = get_substring(painting_description, PAINTING_WIDTH_KEY, " ")
                    image_url = get_substring(html_string, IMAGE_SEARCH_KEY, ">")[:-2]
                    # painting_url = PAINTING_URL + painting_name + ".html"
                    save_image(painting_name, image_url)
                    save_metadata(painting_name, painting_title[:painting_title.index(",")] + "\n" +
                                  painting_title[painting_title.index(",") + 2:] + "\n" +
                                  painting_description + "\n" +
                                  image_url + "\n" +
                                  painting_url)
                    save_width(painting_width)
                    counter += 1
                    print("The painting: \"" + painting_name + "\" has been extracted.\n" + bcolors.BOLD +
                          "[ Processed images: " + str(counter) + " ]\n" + bcolors.ENDC)
                except:
                    with open(WD + FOLDER_NAME + "/paintings_with_exception.txt", "a") as error_file:
                        error_file.write(painting_url + "\n")
                    print(bcolors.FAIL + "The painting: \"" + painting_name + "\" has raised an exception.\n" + bcolors.ENDC)
            file.close()
            widths_file.close()
            print(bcolors.OKGREEN + "Process completed. All information has been successfully " +
                  "extracted from National Gallery site" + bcolors.ENDC)
        def create_directories():
            os.chdir(WD)
            if not os.path.isdir(FOLDER_NAME):
158
                os.mkdir(FOLDER NAME)
            if not os.path.isdir(FOLDER_NAME + "/Image_targets"):
                os.mkdir(FOLDER_NAME + "/Image_targets")
            if not os.path.isdir(FOLDER_NAME + "/Metadata"):
                os.mkdir(FOLDER_NAME + "/Metadata")
164
```

```
97
```

```
165 def main():
166 create_directories()
167 get_painting_names()
168 download_images_and_descriptions()
169
170
171 ▶ if __name__ == "__main__":
172 main()
173
```

Σχήμα 9.5 Κώδικας συλλογής δεδομένων

Για την εκτέλεση του παραπάνω κώδικα, απαιτείται η εγκατάσταση της εξωτερικής βιβλιοθήκης "requests" (εντολή: *pip install requests*). Οι υπόλοιπες βιβλιοθήκες εμπεριέχονται ενσωματωμένες στη βασική έκδοση της Python 3.

Το πρόγραμμα εκκινεί την εκτέλεσή του από τη main() συνάρτηση η οποία με τη σειρά της καλεί τρεις συναρτήσεις :

- create_directories()
- get_painting_names()
- download_images_and_descriptions()

Η συνάρτηση "create_directories()" δημιουργεί τους φακέλους "National_Gallery_Data", "Image targets" και "Metadata". Στον φάκελο "National_gallery_Data" εισάγονται όλα τα δεδομένα μετά την εκτέλεση του προγράμματος, στο φάκελο "Image targets" γίνεται η εισαγωγή όλων των εικόνων από τους πίνακες, ενώ στο φάκελο "Metadata" δημιουργούνται αρχεία κειμένου με τις πληροφορίες που απαιτούνται για κάθε πίνακα.

Η συνάρτηση "get_painting_names()" δημιουργεί ένα αρχείο κειμένου (.txt) με όνομα "painting_urls", στο οποίο κάθε γραμμή περιέχει το σύνδεσμο (URL) που παραπέμπει στην ιστοσελίδα κάθε πίνακα εντός του επίσημου ιστοτόπου της Εθνικής Πινακοθήκης. Η μόνη διαφορά σε κάθε σύνδεσμο είναι το όνομα του πίνακα, γι' αυτό και η συνάρτηση λέγεται "get_painting_names()"

Η συνάρτηση "download_images_and_descriptions()" κατεβάζει όλες τις εικόνες και τις περιγραφές των πινάκων. Επισκέπτεται για κάθε πίνακα την ιστοσελίδα που περιέχει τις πληροφορίες του, στην συνέχεια παίρνει τον html κώδικα της σελίδας και μέσα από αυτόν συλλέγει τα δεδομένα (όνομα ζωγράφου, περιγραφή πίνακα κτλ.). Τέλος, δημιουργεί ένα ακόμα αρχείο κειμένου, με όνομα "paintings_width", μέσα στο οποίο βρίσκονται όλα τα πλάτη

των πινάκων (θα χρειαστεί να γνωρίζουμε το πλάτος κατά τη διαδικασία δημιουργίας της βάσης δεδομένων). Σε περίπτωση που για κάποιο πίνακα αποτύχει η συλλογή δεδομένων (δημιουργία exception), τότε το URL του αποθηκεύεται σε αρχείο κειμένου με όνομα "paintings_with_exception.txt".

Επειδή ο αριθμός των έργων τέχνης είναι αρκετά μεγάλος, η διάρκεια εκτέλεσης του παραπάνω προγράμματος διαρκεί από 5 έως 20 ώρες κατά μέσο όρο. Η διάρκεια εκτέλεσης ποικίλει ανάλογα την επεξεργαστική ισχύ, καθώς επίσης και την ταχύτητα δικτύου (κατέβασμα πολλών εικόνων) που διαθέτει το υπολογιστικό σύστημα που θα τον εκτελέσει. Για τους λογούς αυτούς, το παραπάνω πρόγραμμα, ανέβηκε και εκτελέστηκε σε απομακρυσμένη πλατφόρμα ικανή όχι μόνο να εκτελεί κώδικα Python, αλλά και να αποθηκεύει το σύνολο των αρχείων που παράγει ο κώδικας. Η διαδικτυακή πλατφόρμα που χρησιμοποιήθηκε γι' αυτό το σκοπό, βρίσκεται στον ιστότοπου"<u>www.pythonanywhere.com</u>".

9.2.3 Αποτελέσματα συλλογής δεδομένων

Μετά την ολοκλήρωση της εκτέλεσης, έγινε λήψη των αρχείων σε προσωπικό υπολογιστή για τη δημιουργία κατάλληλης βάσης δεδομένων. Το συνολικό μέγεθος των δεδομένων περιορίζεται σε μερικά Gigabytes. Τα τελικά δεδομένα που συλλέχθηκαν, είχαν τη μορφή του Σχ 9.6.

Name	Date modified	Туре	Size
Image_targets	5/12/2020 1:26 πμ	File folder	
📙 Metadata	5/12/2020 1:27 πμ	File folder	
🗾 paintings_urls.txt	25/11/2020 10:34 μμ	TXT File	540 KI
paintings_width.txt	26/11/2020 12:05 μμ	TXT File	24 KI
🧧 paintings with exception.txt	26/11/2020 11:32 πμ	TXT File	90 KE

Σχήμα 9.6 Τελικά δεδομένα μετά την ολοκλήρωση εκτέλεσης του κώδικα Python

Ακολουθούν στιγμιότυπα οθόνης από όλα τα παραγόμενα αρχεία και φακέλους (στα στιγμιότυπα εμφανίζεται μέρος αυτών).

Φάκελος "Image targets":



Σχήμα 9.7 Οι φωτογραφίες των πινάκων εντός του φακέλου "Image targets"

Φάκελος "Metadata":

ws (C:) > PycharmProjects > ARt > National_Gallery_Data > Metadata				
Name	Date modified	Туре	Size	
💋 a-bishop-s-miraclethe-blessing.txt	26/11/2020 12:20 πμ	TXT File	1 KB	
💋 a-bishop-s-miraclethe-catechism.txt	26/11/2020 12:20 πμ	TXT File	1 KB	
💋 a-bishop-s-miraclethe-eucharist.txt	26/11/2020 12:20 πμ	TXT File	1 KB	
🥖 abraham-s-servant-offering-gifts-to-rebecca-s	26/11/2020 10:17 πμ	TXT File	1 KB	
🥖 abstract-2.txt	26/11/2020 4:55 πμ	TXT File	1 KB	
🥖 abstract-77209.txt	26/11/2020 4:55 πμ	TXT File	1 KB	
≶ abstract-brushstrokes.txt	26/11/2020 4:56 πμ	TXT File	1 KB	
abstract-composition.txt	26/11/2020 4:50 πμ	TXT File	1 KB	
🔰 abstract-composition-two-black-dots-on-a-gra	26/11/2020 4:56 πμ	TXT File	1 KB	
abstract-compositionwhite-gray-black.txt	26/11/2020 4:56 πμ	TXT File	1 KB	
≶ abstract-portrait.txt	26/11/2020 4:56 πμ	TXT File	1 KB	
≶ abstract-portrait-2.txt	26/11/2020 4:57 πμ	TXT File	1 KB	
≶ abstract-portrait-2-3.txt	26/11/2020 4:57 πμ	TXT File	1 KB	
≶ abstract-portraiture.txt	26/11/2020 4:57 πμ	TXT File	1 KB	
🗾 a-bundle-of-rays-in-the-water-1.txt	26/11/2020 4:06 πμ	TXT File	1 KB	
academy-of-sciencessina-academycross-sec	26/11/2020 11:03 πμ	TXT File	1 KB	
🔰 academy-of-sciencessina-academylongitudi	26/11/2020 11:03 πμ	TXT File	1 KB	

Σχήμα 9.8 Όλες οι πληροφορίες για κάθε πίνακα εντός του φακέλου "Metadata"

Κάθε αρχείο του φακέλου Metadata έχει τις πληροφορίες που θα παρουσιαστούν στο τρισδιάστατο πάνελ, όπως φαίνεται στο Σχ. 9.9.

•	acropolis.bt ×	
	Kogevinas Lykourgos	
2	Acropolis,	
	Oil on canvas, 60 x 82 cm	
4	https://www.nationalgallery.gr/images/paintings/71534_2000_2000.jpg	
5	https://www.nationalgallery.gr/en/all-collections/collection/collections/acropolis.html	

Σχήμα 9.9 Παράδειγμα αρχείου Metadata

••	paintings_urls.bdt ×
	1 https://www.nationalgallery.gr/en/all-collections/collection/collections/surroundings-of-grenoble.html
	2 https://www.nationalgallery.gr/en/all-collections/collection/collections/nights-in-graphite-no-3.html
	3 https://www.nationalgallery.gr/en/all-collections/collection/collections/three-female-figures-in-the-space.html
	4 https://www.nationalgallery.gr/en/all-collections/collection/collections/men-on-rocks.html
	5 https://www.nationalgallery.gr/en/all-collections/collection/collections/sounion-in-a-tempest.html
	6 https://www.nationalgallery.gr/en/all-collections/collection/collections/storm-2-3.html
	7 https://www.nationalgallery.gr/en/all-collections/collection/collections/the-artist-drawing-at-the-lake-shore.html
	8 https://www.nationalgallery.gr/en/all-collections/collection/collections/the-blowing-up-of-the-arcadi-monastery.html
	9 https://www.nationalgallery.gr/en/all-collections/collection/collections/the-sea-at-sounion.html
1	0 https://www.nationalgallery.gr/en/all-collections/collection/collections/the-wreck.html
1	1 https://www.nationalgallery.gr/en/all-collections/collection/collections/tsiki-tsiki.html
1	2 https://www.nationalgallery.gr/en/all-collections/collection/collections/horse-drawn-carriage.html
1	3 https://www.nationalgallery.gr/en/all-collections/collection/western-european-painting/incident-from-genesis.html
1	4 https://www.nationalgallery.gr/en/all-collections/collection/collections/medea-delivering-the-penates-to-jason.html
1	5 https://www.nationalgallery.gr/en/all-collections/collection/collections/unchastity.html
1	6 https://www.nationalgallery.gr/en/all-collections/collection/collections/road-with-cypress-trees.html
1	7 https://www.nationalgallery.gr/en/all-collections/collection/collections/rock-in-the-sea-with-constantinople-in-the-background.html
1	8 https://www.nationalgallery.gr/en/all-collections/collection/collections/capturing-enemy-positions-shouting-aera-1.html
1	9 https://www.nationalgallery.gr/en/all-collections/collection/collections/judith-with-the-head-of-holophernes.html
2	% https://www.nationalgallery.gr/en/all-collections/collection/collections/boat-dragged-on-the-shore.html

Σχήμα 9.10 Μέρος από το αρχείο "painting_urls.txt"

+	pai	ntings_width.txt ×
) g	1	0.63
i i	2	2.0
	3	1.18
, i	4	0.83
i.	5	0.18
	6	0.83
į	7	1.14
	8	0.42
	9	2.35
1	0	2.23
1	1	0.89
1	2	1.45
1	3	0.86
1	4	1.1
1	5	0.43
1	6	0.07
1	7	0.06
1	8	0.3
1	9	0.24
2	0	1.5

Σχήμα 9.11 Μέρος του αρχείου "paintings_width.txt"

Κάθε γραμμή του συγκεκριμένου αρχείου αντιστοιχεί με το πλάτος του πίνακα του οποίου το URL βρίσκεται στην ίδια γραμμή του αρχείου "painting_urls.txt"

Για τους πίνακες που δεν έγινε επιτυχώς η συλλογή δεδομένων έχουν αποθηκευτεί τα URL τους στο αρχείο " paintings_with_exception.txt". Λόγω προβλήματος στη σελίδα ενός πίνακα στον ιστότοπο της Εθνικής Πινακοθήκης (δεν ευθύνεται το πρόγραμμα Python), βρέθηκε πίνακας για τον οποίο δεν ολοκληρώθηκε επιτυχώς η συλλογή δεδομένων.



Σχήμα 9.12 Περιεχόμενα του αρχείου "paintings_with_exception.txt"

9.2.4 Ανάλυση δεδομένων

Ο λόγος συλλογής των δεδομένων έγινε για τη δημιουργία αντικειμένων επαυξημένης πραγματικότητας στη συνέχεια. Επειδή, λοιπόν, τα εικονικά αντικείμενα που θα προβάλλονται μέσω της εφαρμογής έχουν διαστάσεις που εξαρτώνται άμεσα από τις πραγματικές διαστάσεις του αντικειμένου με βάση το οποίο θα δημιουργηθούν, είναι πολύ σημαντικό να αναλυθούν τα μεγέθη όλων των πινάκων, κυρίως η εύρεση του μέγιστου και του ελάχιστου πλάτους (περισσότερες πληροφορίες στο κεφάλαιο δημιουργίας βάσης). Για το λόγο αυτό δημιουργήθηκε ένα σύντομο πρόγραμμα σε Python. Ο κώδικας που αναλύει τα πλάτη των πινάκων, παρουσιάζεται παρακάτω:

🛃 Analyze_widths.py 🛛

1	import os
2	
3	# Set working directory, folder and painting widths file
4	<pre>WD = os.getcwd().replace("\\", "/") + "/"</pre>
5	FOLDER_NAME = "National_Gallery_English"
6	WIDTHS_FILE = WD + FOLDER_NAME + "/paintings_width.txt"
7	
8	# Set starting values
9	<pre>min = 10000 # Probably the biggest width for a painting</pre>
10	max = -1
11	sum = 0
12	counter = 0
13	
14	etry:
15	<pre>widths_file = open(WIDTHS_FILE, "r")</pre>
16	<pre>for width in widths_file:</pre>
17	if width == "\n":
18	break
19	<pre>width = float(width.rstrip("\n"))</pre>
20	if width < min:
21	min = width
22	if width > max:
23	max = width
24	counter += 1
25	sum += width
26	print("Paintings widths file results:" +
27	"\nMax = " + str(max) +
28	"\nMin = " + str(min) +
29	"\nSum = " + str(sum) +
30	"\nCount = " + str(counter) +
31	<pre> "\nMean = " + str(sum / counter)) </pre>
32	except:
33	print("An exception occurred")
34	

Σχήμα 9.13 Κώδικας ανάλυσης διαστάσεων πίνακα

9.3 Δημιουργία βάσης δεδομένων και εισαγωγή στο Vuforia Engine

9.3.1 Vuforia Cloud Database

Προτού αναλυθεί ο τρόπος δημιουργίας της Vuforia Cloud Database, είναι χρήσιμο να δοθεί ένας σύντομος και ταυτόχρονα σαφής ορισμός. Πρόκειται για μια βάση δεδομένων ειδικά σχεδιασμένη προκειμένου να επιτευχθεί η δημιουργία επαυξημένης πραγματικότητας μέσω της Vuforia Engine. Είναι μια διαδικτυακή βάση δεδομένων που φιλοξενείται σε διακομιστές της Vuforia Engine. Σε αυτήν αποθηκεύονται εικόνες, ή αλλιώς Image targets (σύμφωνα με την ορολογία που προτείνει η Vuforia). Για μια σωστότερη διατύπωση, θα ακολουθηθεί η υιοθέτηση αυτού του ορού παρακάτω.

Ένα "Image target", αποτελείται από τα ακόλουθα στοιχεία:

Εικόνα

Η μορφή της εικόνας πρέπει να είναι .jpg ή .png, έχοντας μέγιστο μέγεθος τα 2 Megabyte.

Πλάτος εικόνας

Η μονάδα μέτρησης που θα χρησιμοποιηθεί για το πλάτος διαλέγεται αυθαίρετα από το δημιουργό της βάσης. Η Vuforia προτείνει να χρησιμοποιηθεί ως μονάδα μέτρησης ίδια με εκείνη της σκηνής στο Unity. Μπορεί να είναι μέτρα, εκατοστά ή ο,τιδήποτε άλλο. Θα πρέπει, όμως, σε κάθε περίπτωση να τηρηθεί κοινή μονάδα μέτρησης για όλους τους "Image targets" της βάσης. Το ύψος της εικόνας, υπολογίζεται αυτόματα κατά τη διάρκεια δημιουργίας του "Image target".

Πακέτο μεταδιδόμενων (Metadata)

Το πακέτο αυτό, δεν θα πρέπει να ξεπερνάει σε μέγεθος τα 2 Megabyte, ενώ η μόνη μορφή που επιτρέπεται, είναι ένα αρχείο κειμένου (.txt). Σε αυτό το πακέτο, συνήθως, υπάρχει πληροφορία για τα δεδομένα επαυξημένης πραγματικότητας που θα προβληθούν αμέσως μετά την αναγνώριση στόχου. Για παράδειγμα, μπορεί να είναι ένας σύνδεσμος που να παραπέμπει στη λήψη ενός τρισδιάστατου μοντέλου, επιπλέον μπορεί να είναι ένας σύνδεσμος από κάποιο βίντεο του YouTube, μια εικόνα σε μορφή κειμένου, ο,τιδήποτε επιθυμεί ο δημιουργός του πακέτου.

> Ονομασία "Image target"

Οποιοδήποτε όνομα που θα αντιπροσωπεύει τον "Image target".

Αποτελείται από νούμερα (0-9), οποιοδήποτε λατινικό χαρακτήρα (a-z) και από τα σύμβολα κάτω παύλα (_) και παύλα (-).

Για τη δημιουργία ενός "Image target" είναι απαραίτητα όλα τα παραπάνω, εκτός από το πακέτο δεδομένων Metadata.

Στο παρακάτω σχήμα εμφανίζεται η τελική μορφή μιας ολοκληρωμένης Vuforia Cloud Database, ειδικά σχεδιασμένη ώστε να ανταποκρίνεται στις ανάγκες της εφαρμογής. Για το λόγο αυτό κάθε image target περιέχει metadata που αφορούν έργα τέχνης.



Σχήμα 9.14 Σχηματική απεικόνιση μιας Vuforia Cloud Database

9.3.2 Τρόπος λειτουργίας της Vuforia Cloud Database

Μια Vuforia Cloud Database εκτός από μια βάση δεδομένων εικόνων και πληροφοριών, εμπεριέχει και μια ενσωματωμένη λειτουργία αναζήτησης, η οποία εκτελείται αυτόματα στους διακομιστές της Vuforia Engine, χωρίς να επιβαρύνει τον επεξεργαστή της κινητής συσκευής. Για να γίνει περισσότερο κατανοητή αυτή η λειτουργία, θα εξετάσουμε τον τρόπο που η εφαρμογή την χρησιμοποιεί. Κάθε φορά που κάποιος χρήστης σαρώνει τον περιβάλλοντα του χώρο, στέλνεται μια σειρά από στιγμιότυπα αυτού του χώρου (φωτογραφίες) στο διακομιστή της Vuforia. Εκείνος με τη σειρά του έχοντας ως δεδομένο το απεσταλμένο στιγμιότυπο, ξεκινά μια διαδικασία αναζήτησης σε όλους του "Image targets" που διαθέτει. Μόλις βρει κάποια αντιστοιχία, τότε επιστρέφει το συγκεκριμένο "Image target" μαζί με το πακέτο δεδομένων που το συνοδεύει (Metadata).



Σχήμα 9.15 Σχηματική απεικόνιση του τρόπου λειτουργίας της Vuforia Cloud Database

9.3.3 Σημεία αναγνώρισης στόχου

Προκειμένου η παραπάνω διαδικασία αναζήτησης να είναι επιτυχής, κατά τη διάρκεια προσθήκης ενός νέου στόχου (image target) στη βάση δεδομένων, υπολογίζονται αυτόματα κάποια σημεία αναγνώρισης. Με βάση αυτά τα ειδικά σημεία υπολογίζεται ενός βαθμός ομοιότητας μεταξύ της εικόνας που αποστέλλεται στο διακομιστή της Vuforia και του image target που υπάρχει ήδη εντός της Vuforia Cloud Database. Αυτός ο βαθμός ομοιότητας βοηθάει το διακομιστή να αποφασίσει εάν η αναζήτηση θα πετύχει ή θα αποτύχει. Όσα περισσότερα σημεία υπολογιστούν για ένα στόχο, τόσο μεγαλώνει η πιθανότητα αναγνώρισης του από τη βάση δεδομένων και κατ' επέκταση δημιουργούνται ευκολότερες και γρήγορες αναγνωρίσεις στόχων. Μία βάση δεδομένων που έχει στόχους με υψηλή βαθμολογία, οδηγεί στην ανάπτυξη μιας λειτουργικής εφαρμογής, καθώς σε αυτή την περίπτωση όλοι οι στόχοι θα αναγνωρίζονται επιτυχώς.

Υπάρχει συγκεκριμένος αλγόριθμος που εντοπίζει τα σημεία αναγνώρισης για κάθε στόχο. Στο ακόλουθο σχήμα εμφανίζονται τα σημεία αναγνώρισης για έναν πίνακα της Εθνικής Πινακοθήκης με το σχήμα σταυρού (*****).



Σχήμα 9.16 Σημεία αναγνώρισης πίνακα

Αριστερά εμφανίζεται ο πίνακας όπως είναι στην πραγματικότητα, ενώ δεξιά εμφανίζεται η εικόνα του ιδίου πίνακα μετά τον υπολογισμό των ειδικών σημείων, προκειμένου να επιτευχθεί η διαδικασία της αναζήτησης του.

Η Vuforia Database αντιλαμβάνεται κάθε image target ως την εικόνα που παρουσιάζεται στα δεξιά.

Η διαδικασία υπολογισμού ειδικών σημείων κοστίζει σε υπολογιστική ισχύ και εκτελείται αποκλειστικά στους διακομιστές της Vuforia Engine. Συνήθως διαρκεί μερικά δευτερόλεπτα.
Αυτός είναι και ο λόγος που από τη στιγμή δημιουργίας ενός νέου image target υπάρχει κάποια καθυστέρηση μέχρι τη στιγμή εμφάνισης του στη βάση δεδομένων.

9.3.4 Βαθμολογία στόχου

Τα σημεία αναγνώρισης εντοπίζονται βάσει φυσικών χαρακτηριστικών που εξάγονται από την εικόνα που καταχωρείται στη βάση δεδομένων και στη συνέχεια συγκρίνονται κατά το χρόνο εκτέλεσης με χαρακτηριστικά στη ζωντανή εικόνα της κάμερας.

Όσα περισσότερα σημεία αναγνώρισης δημιουργηθούν τόσο καλύτερος θεωρείται ο στόχος, αφού η αναγνώριση του έχει περισσότερες πιθανότητες να είναι επιτυχής. Για το λόγο αυτό η Vuforia Engine βαθμολογεί κάθε image target ξεχωριστά με κατώτερη βαθμολογία το μηδέν και ανώτερη το πέντε. Η βαθμολογία αυτή καθορίζει το πόσο αποτελεσματικά μπορεί να εντοπίζεται και να παρακολουθείται μια εικόνα χρησιμοποιώντας το Vuforia Engine, καθώς αναπαρίσταται με αστέρια που εμφανίζονται στο πάνελ των image targets για κάθε βάση δεδομένων.

Targe	et Name	Rating 🛈	Recos 🗸	Status 🗸	Date Modified 🗸
te	ta-krebatia-tou-thanatou	****	3	Active	Nov 15, 2020 03:10
W.	tin-trabaei-sosta	****	3	Active	Nov 08, 2020 01:26
	<u>ti-allo-prepei-na-kanoun</u>	*****	1	Active	Nov 08, 2020 01:26
13	ti-kouragio	*****	0	Active	Nov 08, 2020 01:26
23	tantalos	*****	45	Active	Nov 08, 2020 01:26
图	ti-thusia	****	0	Active	Nov 08, 2020 01:26
12	ti-na-sou-kanei-mia-koupa	*****	0	Active	Nov 08, 2020 01:26
æ	sodeia-apo-nekrous	****	0	Active	Nov 08, 2020 01:25
	stasou-na-se-hrisoun	****	0	Active	Nov 08, 2020 01:25
戀	ta-epakoloutha	****	5	Active	Nov 08, 2020 01:25
巡	spoudaio-katorthomame-n	****	0	Active	Nov 08, 2020 01:25
St.	sto-kunigi-gia-dontia	****	0	Active	Nov 08, 2020 01:25
6	sout	****	3	Active	Nov 08, 2020 01:25

Σχήμα 9.17 Βαθμολογία πινάκων στο πάνελ της Vuforia Cloud Database

Όσο υψηλότερη είναι η βαθμολογία ενός στόχου εικόνας τόσο ισχυρότερη είναι η ικανότητα ανίχνευσης και παρακολούθησης που περιέχει. Βαθμολογίες από δύο αστέρια και πάνω θεωρούνται αρκετά καλές και καθιστούν το στόχο άμεσα αναγνωρίσιμο. Οι στόχοι που έχουν μηδέν ή ένα αστέρι, αναγνωρίζονται κανονικά. Απλά απαιτείται από το χρήστη να στείλει στο διακομιστή μια αρκετά καλή εικόνα του στόχου, προκειμένου να αναγνωριστεί. Μια αρκετά καλή εικόνα του στόχου, προκειμένου να αναγνωριστεί. Μια αρκετά καλή εικόνα του στόχου, προκειμένου να αναγνωριστεί. Μια αρκετά καλή εικόνα στόχου, ορίζεται ως ένα στιγμιότυπο που περιλαμβάνει ολόκληρο το στόχο και όχι μέρος από αυτόν, σε συνδυασμό με τις καλές συνθήκες φωτισμού. Ο χώρος της Εθνικής Πινακοθήκης ενδείκνυται για αυτή τη λειτουργία, καθώς τόσο οι πίνακες όσο και ο χώρος φωτίζονται επαρκώς και η απόσταση τους από το χρήστη είναι ικανοποιητική προκειμένου να παρθεί ένα στιγμιότυπο που θα περιλαμβάνει ολόκληρό το στόχο. Τέλος, επειδή και το υλικό που αποτυπώνεται η εικόνα αποτελεί σημαντικό παράγοντα, ο καμβάς βελτιώνει την απόδοση των χρωμάτων αποφεύγοντας αντανακλάσεις και θαμπώματα.

9.3.5 Βασικά χαρακτηριστικά που καθορίζουν τη βαθμολογία

Το βασικότερο χαρακτηριστικό που καθορίζει τη δημιουργία των σημείων αναγνώρισης και συνεπώς τη βαθμολογία, είναι τα σχήματα που απαρτίζουν την εικόνα. Όσες περισσότερες οι λεπτομέρειες τόσα περισσότερα και τα σημεία αναγνώρισης. Στην παρακάτω εικόνα (Σχ. 9.18) γίνεται απολύτως κατανοητό το πως επηρεάζουν οι λεπτομέρειες και τα σχήματα που απαρτίζουν την εικόνα στον υπολογισμό των σημείων αναγνώρισης.



Σχήμα 9.18 Χαρακτηριστικά στόχου

Για τη δημιουργία ενός στόχου που θα ανιχνεύεται με ακρίβεια, πρέπει να χρησιμοποιούνται εικόνες σύμφωνα με τα παρακάτω χαρακτηριστικά για έναν ιδανικό Image target.

Πλούσιο σε λεπτομέρειες

Οδός δρόμου, ομάδα ανθρώπων, κολλάζ, μείγματα αντικειμένων και αθλητικές σκηνές είναι καλά παραδείγματα.

Καλή αντίθεση

Οι εικόνες με φωτεινές και σκοτεινές περιοχές λειτουργούν αρκετά καλά.

Χωρίς επαναλαμβανόμενα μοτίβα

Προτείνεται η χρήση μοναδικών δυνατοτήτων και διακριτών γραφικών που καλύπτουν όσο το δυνατόν περισσότερο το στόχο για την αποφυγή συμμετρίας, επαναλαμβανόμενων μοτίβων και των περιοχών χωρίς χαρακτηριστικά.

Κατάλληλη μορφή

Η μορφή της εικόνας θα πρέπει να είναι 8- ή 24-bit PNG και JPG, μικρότερο από 2 MB σε μέγεθος. Οι εικόνες JPG πρέπει να είναι RGB ή κλίμακα του γκρι (όχι CMYK). [8]

Η πλειονότητα των πινάκων της Εθνικής Πινακοθήκης πληροί τα παραπάνω χαρακτηρίστηκα, συνεπώς οι βαθμολογίες που σημειωθήκαν μετά τη δημιουργία της βάση δεδομένων ήταν αρκετά υψηλές. Σε περιπτώσεις που η βαθμολογία ήταν μηδέν (περίπου 50 πίνακες), έγινε αφαίρεση τους από τη βάση δεδομένων και ακολουθήθηκε ειδική επεξεργασία εικόνας, προκειμένου να επιτευχθεί υψηλότερη βαθμολογία.

Υπάρχουν αρκετοί τρόποι και τεχνικές πριν την δημιουργία ενός νέου image target, με τους οποίους μπορεί να διασφαλιστεί η επίτευξη μιας καλής βαθμολογίας αναγνώρισης, αλλά στην περίπτωση μας δεν είναι όλοι διαθέσιμοι. Κατά τη διάρκεια της επεξεργασίας, απαγορεύεται η προσθήκη λεπτομερειών και μικροαντικειμένων που βοηθούν στη δημιουργία νέων σημείων αναγνώρισης, καθώς με αυτό τον τρόπο αλλοιώνεται η πραγματική εμφάνιση του έργου τέχνης. Η διαδικασία που ακολουθήθηκε, ήταν η αύξηση της αντίθεσης και των φωτεινών σημείων με ειδικά προγράμματα επεξεργασίας εικόνας, όπως το Photoshop. Με την ολοκλήρωση της επεξεργασίας οι πίνακες ανέβηκαν εκ νέου στη βάση δεδομένων πετυχαίνοντας πολύ καλύτερη βαθμολογία.

Στο ακόλουθο παράδειγμα, παρουσιάζεται η σταδιακή αύξηση αντίθεσης της εικόνας με συνοπτική περιγραφή των βημάτων που απαιτούνται.



Σχήμα 9.19 Παράδειγμα επεξεργασίας εικόνας (αύζηση αντίθεσης) για επίτευζη καλύτερης βαθμολογίας

Αρχικά απαιτείται φόρτωση της εικόνας στο Adobe Photoshop (ή οποιοδήποτε άλλο πρόγραμμα επεξεργασίας) και μετατροπή του επιπέδου της εικόνας σε «έξυπνο αντικείμενο». Στη συνέχεια από την επιλογή "Filter → Sharpen → Unsharp Mask" προσαρμόζονται κατάλληλα οι ρυθμίσεις Amount, Radius και Threshold για βελτίωση της αντίθεσης. Με αυτό τον τρόπο από βαθμολογία ένα, επιτεύχθηκε βαθμολογία πέντε στα πέντε. Παρόμοια διαδικασία εφαρμόστηκε για την πλειονότητα των πινάκων που είχαν χαμηλή βαθμολογία.

9.3.6 Vuforia Cloud Recognition: Ορισμός και λόγοι επιλογής

Vuforia Cloud Recognition ονομάζεται η τεχνική που χρησιμοποιεί το Vuforia Engine προκειμένου να επιτευχθεί η αναγνώριση στόχων με χρήση διαδικτυακής βάσης δεδομένων (Vuforia Cloud Database).

Η Vuforia Engine, δίνει τη δυνατότητα, εκτός από διαδικτυακή βάση, τη δημιουργία τοπικής βάσης, χωρίς παρουσία απομακρυσμένου διακομιστή. Μια τοπική βάση δημιουργείται στον επίσημο ιστότοπο της Vuforia και στη συνέχεια αφού οι προγραμματιστές την κατεβάσουν τοπικά στον υπολογιστή τους, την ενσωματώνουν εντός της εφαρμογής που δημιουργούν. Αυτή η υλοποίηση επιτρέπει την αναγνώριση στόχων ακόμα κι αν δεν υπάρχει σύνδεση δικτύου. Κάτι τέτοιο, όμως, είναι αδύνατο στην περίπτωση μας, διότι τα έργα τέχνης είναι χιλιάδες και η έκταση των δεδομένων καταλαμβάνει αρκετά Gigabyte. Συνεπώς, για να κατεβάσει οποιοσδήποτε χρήστης την εφαρμογή, θα ήταν αναγκαίο να κατεβάσει ταυτόχρονα και μια αρκετά μεγάλη βάση δεδομένων που θα περιείχε όλα τα image targets (ένα για κάθε πίνακα της Εθνικής Πινακοθήκης), γεγονός που καθιστά τη χρήση της εφαρμογής μη πρακτική. Συνήθως, η τοπική βάση δεδομένων δημιουργείται και ενσωματώνεται σε εφαρμογές όταν πρόκειται για μικρό αριθμό στόχων. Επιπροσθέτως, σε περίπτωση που η εφαρμογή ανεβεί σε κάποιο ηλεκτρονικό κατάστημα εφαρμογών (τύπου PlayStore/AppStore), θα πρέπει να πληροί τους κανόνες που διέπουν μια τέτοια ενέργεια. Ένας από τους βασικότερους κανόνες είναι ότι το μέγεθος της εφαρμογής δεν θα πρέπει να ξεπερνάει τα 100 Megabyte. Συνυπολογίζοντας όλα τα παραπάνω, καταλήγουμε εύκολα στο συμπέρασμα ότι η δημιουργία μιας Cloud Database είναι μονόδρομος.

Με τη δημιουργία μιας Cloud Database, όλα οι Image targets είναι αποθηκευμένοι σε κάποιο/ους διακομιστή/ες και ο χρήστης λαμβάνει δεδομένα μόνο για τους στόχους που επιθυμεί να αναγνωρίσει. Συνεπώς, η εφαρμογή αποτελείται μόνο από τα δομικά της στοιχεία, χωρίς να περιέχει καμία ενσωματωμένη βάση δεδομένων. Ένας τέτοιος σχεδιασμός αποτελεί μια αποδοτική και κομψή υλοποίηση με μονή προϋπόθεση την ύπαρξη δικτύου υψηλής ταχύτητας για την αποστολή και λήψη των δεδομένων από και προς το διακομιστή.

Τα πλεονεκτήματα χρήσης Vuforia Cloud Database, είναι τα παρακάτω:

- Κλίμακα: Περισσότεροι από ένα εκατομμύριο στόχοι μπορούν να χρησιμοποιηθούν σε μια εφαρμογή.
- Ευελιξία: Ενσωμάτωση με υπάρχοντα συστήματα διαχείρισης περιεχομένου.
- Time to Market: Παράδοση σε πραγματικό χρόνο, δυναμικά μεταβαλλόμενο περιεχόμενο και επιτάχυνση του χρόνου στην αγορά. [9]

9.3.7 Δημιουργία Vuforia Cloud Database

Με την ολοκλήρωση της συλλογής δεδομένων ολοκληρώνεται το πρώτο στάδιο υλοποίησης, γεγονός που μας επιτρέπει να προχωρήσουμε στο επόμενο στάδιο, τη δημιουργία κατάλληλης βάσης δεδομένων. Η βάση δεδομένων αποτελεί δομικό στοιχείο για την υλοποίηση της εφαρμογής. Είναι εκείνη που θα περαστεί εντός του διακομιστή με αποτέλεσμα την εξυπηρέτηση των χρηστών.

Κατά τη διάρκεια υλοποίησης της εφαρμογής δεν υπήρχε διαθέσιμος διακομιστής, συνεπώς η βάση δεδομένων δημιουργήθηκε σε διακομιστή της Vuforia Engine, εντός του επίσημου ιστότοπου της Vuforia και εξυπηρετεί τις ανάγκες της εφαρμογής απομακρυσμένα. Σε περίπτωση ύπαρξης διακομιστή εντός της Εθνικής Πινακοθήκης, η βάση θα μεταφερθεί σε αυτόν.

Η δημιουργία της βάσης δεδομένων χωρίζεται σε δύο βήματα:

• Καταχώρηση νέας Vuforia Cloud Database

• Εισαγωγή των δεδομένων στην Vuforia Cloud Database

Στις επόμενες δύο υποενότητες θα αναλυθεί κάθε βήμα χωριστά

Καταχώρηση νέας Vuforia Cloud Database

Για τη δημιουργία μιας Vuforia Cloud Database, είναι απαραίτητες οι ακόλουθες ενέργειες:

- Δημιουργία λογαριασμού στον επίσημο ιστότοπου της Vuforia Engine
- Δημιουργία κλειδιού χρήσης.

Το κλειδί χρήσης, χρησιμοποιείται από τη Vuforia προκειμένου να γνωρίζει πόσες αναγνωρίσεις αντικειμένων πραγματοποιήθηκαν από τις διασυνδεδεμένες εφαρμογές, προκειμένου να υπολογίσει τα κόστος χρήσης της υπηρεσίας.

• Προσθήκη νέας Cloud Database στο λογαριασμό

Δημιουργία λογαριασμού Vuforia Engine

Για τη δημιουργία λογαριασμού, είναι απαραίτητα τα ακόλουθα βήματα:

• Επίσκεψη του επίσημου ιστότοπου της Vuforia (<u>https://developer.vuforia.com</u>)

Nov 25, 2020

Vuforia Engine 9.6 is Available!

The Vuforia Engine team is happy to announce that version 9.6 is now live! We've made some quality of life improvements to our Model Targets development process and have added support for the new Iphone 12 for scanning Area Targets. Below are the key updates in this release, please be sure to check out the release notes for the full list.

Model Target Generator

- When creating a close-up guide view with Advanced Model Targets the Model Target Generator will now automatically calculate a suitable target extent for optimal detection.
- Detection of Standard Model Targets has been enhanced to include a wider range of positions around the defined guide view for snapping to the model.

Vuforia Area Target Creator App

- Developers can now create Area Targets quickly and easily on the new iPhone Pro
 12!
- The amount of scanning time supported in the application has been increased to 5 minutes to help developers capture larger spaces and include more details.

Unite Now 2020 Session:

• Check out our Unite Now 2020 session on the value of spatial AR and how to create a Vuforia Engine Area Target! Watch here.

If you have questions about this release or other Vuforia Engine features, feel free to post on our community forums. Have an AR experience you'd like to share with us? Please send any videos of your Vuforia Engine application to vuforia-feedback@ptc.com.



Thanks, Vuforia Engine Team

Σχήμα 9.20 Αρχική σελίδα της Vuforia Engine

 Επιλογή "Register" και συμπλήρωση των απαραίτητων στοιχείων στη φόρμα «Δημιουργίας λογαριασμού».

Register for a Vuforia Developer Account

With an account you can download development tools, get license keys, and participate in the Vuforia community.

First Name * Emmanouil		Last Name * Remountakis	
Company *		Country *	
ARt		Greece	
Email Address *		Username *	
el13527@ntua.gr	(?)	el13527	(?)
Password *		Confirm Password *	



✓ I agree to the terms of the Vuforia Developer Agreement.

I acknowledge that my personal details will be processed in accordance with PTC's privacy policy and may be used for marketing purposes by PTC Inc. its subsidiaries and members of the PTC Partner Network, solely for the promotion of PTC's products and associated services.

Create account

Σχήμα 9.21 Φόρμα δημιουργίας λογαριασμού Vuforia

Μετά τη συμπλήρωση της παραπάνω φόρμας, είναι απαραίτητη η επιβεβαίωση της διεύθυνσης ηλεκτρονικού ταχυδρομείου.

 Είσοδο στον ιστότοπο της Vuforia με username τη διεύθυνση ηλεκτρονικού ταχυδρομείου και password τον κωδικό που συμπληρώθηκε κατά τη διάρκεια εγγραφής.

LOGIN Login with your Vuforia developer account to download software, get license keys and participate in the Vuforia community.					
Email Address el13527@ntua.gr					
Not Registered? Create account now					
Password					
	<u></u>				

Σχήμα 9.22 Είσοδος στην πλατφόρμα της Vuforia Engine

Δημιουργία "Development key" (Κλειδί / Άδεια χρήσης)

Login

Με την ολοκλήρωση του λογαριασμού, επιτρέπεται η δημιουργία κλειδιού χρήσης. Η διαδικασία δημιουργίας κλειδιού χρήσης περιγράφεται παρακάτω.

Μετά την είσοδο στην πλατφόρμα της Vuforia Engine, γίνεται πλοήγηση στην καρτέλα "License Manager" και εκεί επιλέγεται το κουμπί "Get Development Key."

License Manager	Target Manager		
License Ma	inager	Get Development Key	Buy Deployment Key
Create a license ke	ey for your application.		
Search			

Σχήμα 9.23 Vuforia Get Development Key

Για τη δημιουργία κλειδιού χρήσης απαιτείται η συμπλήρωση ονόματος του κλειδιού και η αποδοχή των όρων χρήσης της Vuforia Engine (checkbox). Στη συνέχεια επιλέγεται το κουμπί "Confirm".

License Manager Target Manager

Back To License Manager

Add a free Development License Key

License Name *
ARt_Development_License_Key

You can change this later

License Key

Develop Price: No Charge Reco Usage: 1,000 per month Cloud Targets: 1,000 VuMark Templates: 1 Active VuMarks: 100

By checking this box, I acknowledge that this license key is subject to the terms and conditions of the Vuforia Developer Agreement.



Σχήμα 9.24 Δημιουργία Vuforia Development License Key

Σημείωση: Το κλειδί που δημιουργείται με την παραπάνω διαδικασία, αποτελεί δωρεάν κλειδί χρήσης και η απόκτηση του γίνεται χωρίς καμία χρέωση. Για το λόγο αυτό επιτρέπονται μόνο χίλιες αναγνωρίσεις στόχων κάθε μηνά. Ο αριθμός αυτός είναι σε θέση να καλύψει τις ανάγκες της εφαρμογής όσο βρίσκεται σε ακαδημαϊκό επίπεδο. Σε περίπτωση που η εφαρμογή εγκατασταθεί στην Εθνική Πινακοθήκη, δημιουργείται η ανάγκη ανανέωσης του κλειδιού χρήσης σε καταλληλότερη κατηγορία, που θα επιτρέπει απεριόριστο αριθμό αναγνωρίσεων. Το κόστος για μια τέτοια αλλαγή εξαρτάται από τις αναγνωρίσεις των στόχων που απαιτούνται, συνεπώς από την επισκεψιμότητα της πινακοθήκης.

Περισσότερες λεπτομέρειες σχετικά με τα κλειδιά χρήσης και το κόστος απόκτησης, βρίσκονται στο κεφ. 8.3 και στον ακόλουθο σύνδεσμο: <u>Vuforia - Τιμολόγηση</u>.

Στη συνέχεια γίνεται αυτόματη πλοήγηση στην καρτέλα "License Manager" όπου εκεί υπάρχει το κλειδί που δημιουργήθηκε ("ARt_Development_License_Key") μαζί με όλα τα υπόλοιπα κλειδιά που έχουν δημιουργηθεί στο παρελθόν και ανήκουν στον ίδιο λογαριασμό.

Vuforia engine developer portal	Pricing Downloads	Library	Develop Su	pport	Hello eRemountakis 🗸 🕴 Log Out				
License Manager Target Manager									
License Manager				Get Developme	nt Key Buy Deployment Key				
Create a license key for your application.									
Search									
Name	Primary UUID 🛈	Туре	Statu	s 🗸	Date Modified				
ARt_Development_License_Key	N/A	Develop	Activ	9	Jan 24, 2021				
licenseV5	N/A	Develop	Active	2	Nov 22, 2020				
IcenseV4	N/A	Develop	Active	9	Nov 20, 2020				
new.	N/A	Develop	Active	5	Nov 15, 2020				
/licenseV3	N/A	Develop	Active	9	Nov 15, 2020				
LicenseV2	N/A	Develop	Active	9	Nov 07, 2020				
↓ z_licenseV1	N/A	Develop	Active	5	Nov 07, 2020				
25 per page Showing 1-7 of 7 Last updated: Today 2:47 PM Refresh			1 >>>						

Σχήμα 9.25 Καρτέλα "License Manager" της Vuforia

Πατώντας πάνω στο κλειδί χρήσης που μόλις δημιουργήθηκε ανοίγει η σελίδα με τις πληροφορίες του κλειδιού. Αυτή η σελίδα, έχει δύο βασικές καρτέλες: "License Key" και "Usage". Η καρτέλα "License Key" περιέχει το μοναδικό κλειδί χρήσης που θα εισαχθεί στην εφαρμογή προκειμένου η Vuforia Engine να γνωρίζει πόσες επιτυχείς αναγνωρίσεις έγιναν και πότε, ενώ η καρτέλα "Usage" περιέχει πληροφορίες για τον αριθμό τον αναγνωρίσεων που απομένουν για τη συγκεκριμένη άδεια χρήσης. Στις παρακάτω φωτογραφίες εμφανίζονται τα στιγμιότυπα από μια δωρεάν άδεια χρήσης και για τις δύο καρτέλες.

License Manager Target Manager

License Manager > ARt_Development_License_Key

ARt_Development_License_Key Edit Name Delete License Key

License Key

Usage

Please copy the license key below into your app

AfDOAw3////AAABmYwcZVb9IUucvoDljFal8GsRRZ1obUlobZRKyS3bGaXjzR358i6FA5PYWjbJ+WlwczDRWWunrh2NFvdJGxGEEm5H3x QuvDm5g7XgWXssXVsVc56D14wmOhigIs9wzDlwjiYeXdZ0fy/uRF6REv5khjEmUQl3fZ6FXXRtQEAyxMpcxhAMwNBo9YaFQLS4GVhQ9FL+ 6QGWWhWrJUg1jlRUoNsSVsDATpBF/RKMud/VwhiULW6iUrdolBKWH11mVY+d7J4f7CmrlJnjSyNwHL8VFDzKuRUdEAZ5u2u3smjB5Gv5xP 7SghBPffiTfMWvDWoa6KWMENmCvqsgd9tKsCchovuQYtvbRvUgknVR2BvQ08v9

Plan Type: Develop Status: Active Created: Jan 24, 2021 14:47 License UUID: 81a43c7fa4c44a92a5c04a2debc2cf19

Permissions:

- Advanced Camera
- External Camera
- Model Targets
- Watermark

History:

Cloud Database Associated (ARt_Cloud_Database33) - Jan 24, 2021 15:13 License Created - Jan 24, 2021 14:47

Σχήμα 9.26 Καρτέλα "License Key" της αδείας χρήσης "Art_Development_License_Key"

License Manager Target Manager	
License Manager > ARt_Development_License_Key	
ARt_Development_License_Key Edit Name	2 Delete License Key
License Key Usage	
Cloud Recos	VuMarks generated
0 of 1000 (0%)	0 of 100 (0%)
Cloud Databases: ARt_Cloud_Database33	VuMark Databases: None
Last Usage: Today 00:37	VuMark Templates: 1 active
Reco Threshold: 1,000 recos / month Cloud Targets: 1 of 1000	VuMarks: 100

Statement History

Σχήμα 9.27 Καρτέλα "Usage" της αδείας χρήσης "Art_Development_License_Key"

Προσθήκη νέας Cloud Database στο λογαριασμό

Με την ολοκλήρωση του κλειδιού χρήσης (ARt_Development_License_Key), επιτρέπεται η δημιουργία της Vuforia Cloud Database. Η διαδικασία περιγράφεται παρακάτω.

Επιλογή καρτέλας "Target Manager" και επιλογή του κουμπιού "Add Database".



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

Σχήμα 9.28 Vuforia Add Database

Για τη δημιουργία νέας βάσης δεδομένων, απαιτείται η εισαγωγή ονόματος βάσης (ARt_Cloud_Database), επιλογή "Cloud" στην ενότητα "Type" και η σύνδεση της βάσης με

το κλειδί χρήσης που δημιουργήθηκε σε προηγούμενο βήμα (ARt_Development_License_Key). Με την ολοκλήρωση της εισαγωγής αυτών των πεδίων, επιλέγεται το κουμπί "Create".

Database Name * ARt_Cloud_Database		
Туре:		
O Device		
Cloud		
○ VuMark		
ARt_Development_License_Key		
To create a license key please go to the License Manager		
To create a license key please go to the License Manager	Cancel	Creat

Σχήμα 9.29 Δημιουργία Vuforia Cloud Database

Στη συνέχεια, γίνεται αυτόματη πλοήγηση στην καρτέλα "Target Manager" όπου εκεί υπάρχει η βάση δεδομένων που μόλις δημιουργήθηκε (ARt_Vuforia Cloud_Database) μαζί με όλες τις υπόλοιπες βάσεις δεδομένων που έχουν δημιουργηθεί στο παρελθόν και ανήκουν στον ίδιο λογαριασμό.

Vuforia engine Home Pricing developer portal		Pricing	Downloads	Library	Develop	Support	Hello eRemountakis 🐱 Log Ou	t		
License Manager	Target Manager									
Target Mar	ager						Add Database			
Use the Target Manager to create and manage databases and targets.										
Search										
Database			Туре			Targets	Date Modified			
ARt_Cloud_Databas	se		Cloud			1000	Nov 15, 2020			
ARt_cloud_DB_V3			Cloud			961	Nov 15, 2020			
ARt_Cloud_DB_V4			Cloud			961	Nov 20, 2020			
ARt_Cloud_DB_V5			Cloud			950	Nov 26, 2020			
25 per page 👻 Last updated: Today 3:07	Showing 1-4 of 4 7 PM Refresh				< 1 >>>					



Εισαγωγή των δεδομένων στην Vuforia Cloud Database

Έχοντας δημιουργήσει τη βάση δεδομένων, επιτρέπεται η εισαγωγή δεδομένων σε αυτήν.

Υπάρχουν δύο τρόποι εισαγωγής δεδομένων σε μια Vuforia Cloud Database:

- Χειροκίνητα, μέσω της πλατφόρμας στον επίσημο ιστότοπο της Vuforia
- Μέσω του Vuforia Web Services API

Θα παρουσιαστούν και οι δύο τρόποι, καθώς χρησιμοποιήθηκαν και οι δύο για την υλοποίηση της εφαρμογής.

Εισαγωγή δεδομένων μέσω της Vuforia πλατφόρμας

Αυτός ο τρόπος χρησιμοποιήθηκε για μικρό αριθμό πινάκων, διότι είναι αρκετά χρονοβόρος. Ειδικότερα, χρησιμοποιήθηκε μόνο για τους πίνακες με χαμηλή βαθμολογία, οι οποίοι αφαιρέθηκαν από τη βάση δεδομένων, επεξεργάστηκαν, ώστε να πετύχουν μεγαλύτερη βαθμολογία και στη συνέχεια δημιουργήθηκαν εκ νέου. Τα βήματα δημιουργίας νέου image target μέσω της Vuforia πλατφόρμας περιγράφονται παρακάτω. Από την καρτέλα "Target Manager", επιλέγεται η βάση δεδομένων που δημιουργήθηκε σε προηγουμένη ενότητα (Βλ. Σχήμα: «Καρτέλα "Target Manager" της Vuforia»).

vuforia [®] engin _{developer po}	Ie [™] Ho _{ortal}	ome Pric	ing Downloads	s Library	Develop	Support	Hello eRemountakis $ \sim ig $ Log Out
License Manager	Target Manag	ger					
Target Manager > ARt_	Cloud_Databas						
ARt_Cloud_ Type: Cloud License Key: ARt_De	Databas	e Edit Name					
Targets (0)	Database A	ccess Keys					
Add Target							Search by target name or target ID
Target Name			Rating (i)	Recos	~	Status 🗸	Date Modified 🗸
Last updated: Today 05:	00 PM Refresh						

Σχήμα 9.31 "ARt_Cloud_Database" πάνελ

Στο πάνελ της βάσης δεδομένων επιλέγεται το κουμπί "Add Target". Στη συνέχεια, γίνεται η εισαγωγή των δεδομένων που απαρτίζουν έναν Image target. Δίνεται ένα παράδειγμα εισαγωγής δεδομένων για τον πίνακα "Acropolis" στην εικόνα που ακολουθεί.

Τα στοιχεία που απαιτούνται για τη δημιουργία ενός νέου image target, είναι τα παρακάτω:

- Εικόνα (.jpg ή .png με μέγιστο μέγεθος 2 Megabyte)
 Το αρχείο "acropolis.jpg" αποτελεί την μπροστινή όψη του πίνακα «Ακρόπολη».
- Πλάτος (σε μονάδα μέτρησης της σκηνής στο Unity)
 Εχουν επιλεγεί 23 μονάδες σκηνής (για όλες τις σκηνές ορίστηκε αυθαίρετα μια κοινή μονάδα μέτρησης απόστασης)..

 Πακέτο Metadata (αρχείο κειμένου .txt με μέγιστο μέγεθος ορίζονται τα 2 Megabyte) Το αρχείο "acropolis.txt" περιέχει πληροφορίες για τον πίνακα (τίτλος, περιγραφή, όνομα ζωγράφου και URL για την ιστοσελίδα του έργου τέχνης).

4. Όνομα

Στο πεδίο «Όνομα» μπορεί να μπει ο,τιδήποτε, αλλά προς χάριν οργάνωσης έχει επιλεγεί αυτόματα το ίδιο όνομα με εκείνο της εικόνας του πίνακα ("acropolis").

acropolis.jpg	Browse
.jpg or .png (max file 2mb)	
Width:	
23	
Enter the width of your target in	i scene units. The size of the target should be of
same scale as your augmented unit scale. The target's height w Metadata Package: (Option	virtual content. Vuforia uses meters as the defa ill be calculated when you upload your image. nal)
same scale as your augmented unit scale. The target's height w Metadata Package: (Optior acropolis X	virtual content. Vuforia uses meters as the defa ill be calculated when you upload your image. ^{nal})
same scale as your augmented unit scale. The target's height w Metadata Package: (Optior acropolis X max file 2mb	virtual content. Vuforia uses meters as the defa ill be calculated when you upload your image. ^{nal})
same scale as your augmented unit scale. The target's height w Metadata Package: (Optior acropolis X max file 2mb Name:	virtual content. Vuforia uses meters as the defa ill be calculated when you upload your image. ^{nal})

Σχήμα 9.32 Εισαγωγή νέου Image target στην Cloud Database μέσω της πλατφόρμας Vuforia

Με την ολοκλήρωση εισαγωγής των στοιχείων επιλέγεται το κουμπί "Add". Με αυτή την ενέργεια ενεργοποιούνται όλες οι απαραίτητές λειτουργίες προκειμένου να εισαχθεί το νέο

Image target στη βάση δεδομένων. Οι βασικότερες λειτουργίες είναι το ανέβασμα των δεδομένων στη βάση και ο υπολογισμός των σημείων αναγνώρισης που συμβάλλει στον καθορισμό της τελικής βαθμολογίας του Image target. Όσο αυτές οι λειτουργίες εκτελούνται, ο χρήστης ενημερώνεται με μια μπάρα φόρτωσης.



Σχήμα 9.33 Ποσοστιαία μπάρα κατά τη δημιουργία νέου Image target

Με την ολοκλήρωση των ενεργειών δημιουργίας νέου Image target γίνεται αυτόματη μεταφορά στην καρτέλα της βάσης δεδομένων. Στο ακόλουθο σχήμα εμφανίζεται η κατάσταση της Vuforia Cloud Database με όνομα "ARt_Cloud_Database" και license key

vuforia [®] engin developer po	le [™] Home _{prtal}	Pricing Dov	nloads Libra	ry Develop	Support	Hello eRemountakis 🗸 Log Out					
License Manager	Target Manager										
Target Manager > ARt_	Target Manager > ARt_Cloud_Databas										
ARt_Cloud_Database Edit Name Type: Cloud License Key: ARt_Development_License_Key Targets (1) Database Access Keys											
Add Target						Search by target name or target ID					
Target Name		Rating	(i) Re	cos 🗸	Status 🗸	Date Modified 🗸					
🗆 🥌 acropo	lis	***	3		Active	Jan 24, 2021 20:17					

Last updated: Today 09:32 PM Refresh

Σχήμα 9.34 Καρτέλα Vuforia Cloud Database

"ARt_Development_License_Key" που έχει μόλις έναν Image target με όνομα "acropolis". Σε αυτήν εμφανίζονται όλοι οι Image target της βάσης.

Η καρτέλα μιας Vuforia Cloud Database έχει δύο υποκαρτέλες, εκείνη με το όνομα "Targets" και εκείνη με όνομα "Database Access Keys"

Στην καρτέλα "Targets" μιας Vuforia Cloud Database, εμφανίζονται οι ακόλουθες πληροφορίες χωρισμένες ανά στήλη:

- Target Name: Το όνομα κάθε Image Target. Αριστερά του ονόματος εμφανίζεται σε μικρογραφία η εικόνα του image target.
- Rating: Η βαθμολογία για κάθε Image Target σε μορφή αστεριών από μηδέν έως πέντε.
 Η βαθμολογία εξαρτάται αποκλειστικά από το πλήθος των σημείων αναγνώρισης της εικόνας του Image target.
- Recos: Αριθμός που αντιπροσωπεύει το πλήθος των επιτυχημένων αναγνωρίσεων του Image target.
- Status: Υποδηλώνει την κατάσταση του image target. Οι πιθανές τιμές αυτής της στήλης είναι "active", "inactive" και "failed". Η κατάσταση "active" δηλώνει ότι ο image target είναι ενεργός και έτοιμος προς αναγνώριση. Η κατάσταση "inactive" δηλώνει ότι ο image target έχει τεθεί εκτός λειτουργίας, υπάρχει κανονικά στη βάση δεδομένων, αλλά ο δημιουργός της βάσης έχει αποφασίσει ότι ο συγκεκριμένος στόχος δεν θα αναγνωρίζεται. Τέλος, η κατάσταση "failed" δηλώνει ότι ο image target απέτυχε να ενταχθεί επιτυχώς στη βάση (κατά τη διάρκεια εισαγωγής του, προέκυψε κάποιο σφάλμα). Όλοι οι στόχοι με κατάσταση "failed" καταλαμβάνουν χώρο στη βάση, αλλά δεν αναγνωρίζονται.
- Date Modified / Last Reco: Πρόκειται για μια επιπλέον στήλη που βοηθάει στην κατάταξη των image targets. Ο χρήστης μπορεί να επιλέξει ανάμεσα στις τιμές "Date Modified" και "Last Reco". Με την επιλογή "Date Modified", οι image target κατατάσσονται ανάλογα με την ημερομηνία και ώρα που έγινε η τελευταία τους επεξεργασία, ενώ με την επιλογή "Last Reco" οι image targets κατατάσσονται ανάλογα με την ημερομηνία και ώρα που έγινε η τελευταία τους επιτυχημένη αναγνώριση.

Η καρτέλα "Targets" της βάσης δεδομένων, είναι ένα πολύ χρήσιμο εργαλείο, καθώς επιτρέπει στον προγραμματιστή να έχει μία καθολική εικόνα των image target που βρίσκονται εντός της βάσης. Στην καρτέλα "Database Access Keys" βρίσκονται τα κλειδιά που επιτρέπουν την πρόσβαση στη βάση δεδομένων. Υπάρχουν δύο κατηγορίες κλειδιών, τα "Client Access Keys" και τα "Server Access Keys". Τα "Client Access Keys" επιτρέπουν στην οποιαδήποτε εφαρμογή επιθυμεί να χρησιμοποιήσει τη συγκεκριμένη βάση δεδομένων, να έχει πρόσβαση. Τα "Server Access Keys" επιτρέπουν την πρόσβαση στη βάση δεδομένων για εισαγωγή νέων image target μέσω REST API κλήσεις (ακολουθεί επεζήγηση του όρου σε επόμενη ενότητα).

vuforia engin developer po	le [™] Home	Pricing	Downloads	Library	Develop	Support
License Manager	Target Manager					
Target Manager > ARt_	Cloud_Databas					

ARt_Cloud_Database Edit Name

Type: Cloud License Key: ARt_Development_License_Key

Targets (1)

Database Access Keys

Access Keys are specific to each cloud database. You should never share your access keys with an untrusted party and should take the necessary steps to protect them within your application code.

Client Acc	ess Keys
Access Key:	f248f38lced2d746be1ad842434d3861898e995e
Secret Key:	d1ab73f5885aedaa801415d0174e8b60c43364f5
	The Client Access Keys must be passed to the Vuforia library within the app in order to authenticate itself with the server
Server Ac	cess Keys
Server Acc	1ded95775521f738d5d8eb47a01a0aed6d22f72c
Server Acc Access Key: Secret Key:	1ded95775521f738d5d8eb47a01a0aed6d22f72c 04be9a372eabba8e4accd3816cb7a88c988a61bf

Σχήμα 9.35 Κλειδιά πρόσβασης σε Vuforia Cloud Database

Σημείωση: Για λογούς ασφαλείας τα παραπάνω κλειδιά δεν αντιστοιχούν στα πραγματικά (έγινε επεζεργασία εικόνας).

Δημιουργία Vuforia Cloud Database με χρήση Vuforia Web Services (API)

Το Vuforia Engine δίνει τη δυνατότητα δημιουργίας, επεξεργασίας, διαγράφης και άλλων ενεργειών για τη διαχείριση εικόνων σε μια Cloud Database χρησιμοποιώντας REST APIs μέσω HTTPS. Το σύνολο αυτών των δυνατοτήτων ονομάζεται Vuforia Web Services (VWS) κατά τη Vuforia.

Προτού αναλυθεί η διαδικασία υλοποίησης της βάσης δεδομένων με χρήση Vuforia Web Services, είναι χρήσιμο να δοθούν σύντομοι ορισμοί των εννοιών API, REST κλήσεων και HTTPS πρωτοκόλλου.

Διεπαφή Προγραμματισμού Εφαρμογών – ΑΡΙ

Η Διεπαφή Προγραμματισμού Εφαρμογών [Application Programming Interface (API)], γνωστή και ως Διασύνδεση Προγραμματισμού Εφαρμογών (εν συντομία διεπαφή ή διασύνδεση), είναι η διεπαφή των προγραμματιστικών διαδικασιών που παρέχει ένα λειτουργικό σύστημα, βιβλιοθήκη ή εφαρμογή, προκειμένου να επιτρέπει την αποστολή αιτημάτων ή την ανταλλαγή δεδομένων με άλλα λογισμικά ή προγράμματα. Όταν ένα API χρησιμοποιείται για την ανάπτυξη διαδικτυακών εφαρμογών, αναφέρεται ως web API. Ένα web API περιέχει ένα σύνολο από μεθόδους που εκτελούν HTTP requests σε έναν server. Τα web API ανταλλάσσουν δεδομένα σε μορφή JSON ή σε XML.

Ένας από τους βασικούς σκοπούς μίας διεπαφής είναι να ορίζει και να διατυπώνει το σύνολο των λειτουργιών-υπηρεσιών που μπορεί να παρέχει μια βιβλιοθήκη ή ένα λειτουργικό σύστημα σε άλλα προγράμματα, χωρίς να επιτρέπει πρόσβαση στον κώδικα που υλοποιεί αυτές τις υπηρεσίες. Η διεπαφή, ένα «συμβόλαιο κλήσης» μεταξύ καλούντος και καλούμενου, διαχωρίζει την προγραμματιστική υλοποίηση κάποιων υπηρεσιών από τη χρήση τους. [11]

Αρχιτεκτονική Representational State Transfer - REST

Η αρχιτεκτονική REST (REpresentational State Transfer) αποτελεί ένα σύνολο από αρχές σχεδίασης μιας δικτυακής υπηρεσίας. Η μεταβολή της κατάστασης της βάσης δεδομένων του συστήματος περιγράφεται και μεταφέρεται στο σύστημα μέσω του πρωτοκόλλου HTTP από διάφορους πελάτες (clients) ανεξαρτήτως της γλώσσας στην οποία έχουν υλοποιηθεί. [12]

Πρωτόκολλο επικοινωνίας ΗΤΤΡ

Το HTTPS (Hypertext Transfer Protocol Secure) πρόκειται για ένα πρωτόκολλο επικοινωνίας και χρησιμοποιείται στην πληροφορική για να δηλώσει μία ασφαλή δικτυακή σύνδεση HTTP. Ένας σύνδεσμος (URL) που αρχίζει με το πρόθεμα https υποδηλώνει ότι θα χρησιμοποιηθεί κανονικά το πρωτόκολλο HTTP, αλλά η σύνδεση θα γίνει σε διαφορετική πόρτα (443 αντί 80) και τα δεδομένα θα ανταλλάσσονται κρυπτογραφημένα. Κάθε πρωτόκολλο https δεν είναι το ίδιο, καθώς δεν παρέχει την ίδια ασφάλεια και αποτελεσματικότητα στα αποτελέσματα των μηχανών αναζήτησης. Το σύστημα αυτό σχεδιάστηκε αρχικά από την εταιρία Netscape Communications Corporation για να χρησιμοποιηθεί σε ιστοσελίδες όπου απαιτείται επαλήθευση χρηστών και κρυπτογραφημένη επικοινωνία. Σήμερα, χρησιμοποιείται ευρέως στο Διαδίκτυο, όπου χρειάζεται αυξημένη ασφάλεια, διότι διακινούνται ευαίσθητες πληροφορίες (π.χ. αριθμοί πιστωτικών καρτών, κωδικοί κοκ). [13]

9.3.8 Vuforia Web Services (VWS): Κώδικας υλοποίησης

Η χρήση του Vuforia Web Services μέσω APIs, επιτρέπει τις ακόλουθες ενεργείς:

- Δημιουργία νέων image target
- Διαγραφή image target
- Ενημέρωση image target
- Δημιουργία ιστορικού αναγνώρισης image target
- Έλεγχος για παρόμοιους τύπους image target εντός ιδίας cloud database
- Αποστολή λίστας με όλους τους image target μιας cloud database
- Δημιουργία συνοπτικής αναφοράς image target
- Δημιουργίας συνοπτικής αναφοράς βάσης δεδομένων

Υπάρχει αναλυτική επεξήγηση για την κάθε κατηγορία (API) εντός του επίσημου ιστότοπου της Vuforia Engine (<u>How To Use Vuforia Web Services - API</u>) μαζί με παραδείγματα χρήσης.

Στη συγκεκριμένη ενότητα θα αναλυθεί μόνο η πρώτη κατηγορία (δημιουργία image target) του Vuforia Web Services, καθώς αυτή χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής. Η διαδικασία που ακολουθείται για τη δημιουργία ενός νέου image target, είναι η αποστολή ενός HTTP Post αιτήματος στην ιστοσελίδα <u>https://vws.vuforia.com/targets</u>. Η υλοποίηση όλων των αιτημάτων, έγινε με χρήση κώδικα java (έκδοση 15). Ο κώδικας που δημιουργεί ένα αντικείμενο τύπου image target και στέλνει για κάθε πίνακα ένα HTTP Post request για την ένταξη του στόχου στη βάση δεδομένων παρουσιάζεται παρακάτω.

```
🔮 CreateCloudDatabase.java 🛛
       package com.qualcomm.vuforia;
 1
 2
 3
      import java.io.BufferedReader;
       import java.io.File;
 4
 5
       import java.io.FileReader;
      import java.util.Scanner;
 6
       public class CreateCloudDatabase {
 8
 9
10 🕨 🖯
           public static void main(String[] args) {
               final String PROJECTPATH
                                                 = "C:/Users/PycharmProjects/ARt/";
               final String FOLDERNAME
                                                 = "National_Gallery_Data/";
13
               final String PAINTINGSFILE
                                                 = PROJECTPATH + FOLDERNAME + "paintings_urls.txt";
14
15
               final String WIDTHSFILE
                                                 = PROJECTPATH + FOLDERNAME + "paintings_width.txt";
                                                 = PROJECTPATH + FOLDERNAME + "Image_targets/";
16
               final String PAINTINGSPATH
               final String DESCRIPTIONSPATH
                                                = PROJECTPATH + FOLDERNAME + "Metadata/";
18
               final String SERVERACCESSKEYSFILE = PROJECTPATH + "Server_Access_Keys.txt";
19
               //Server Keys (Must be changed for new Database)
                                                = "e5bb5db6cb941b2be5a2d25e7f0b8919cab803a0";
               final String ACCESSKEY
               final String SECRETKEY
                                                = "d513a6176af4c7f80d93db53ec072411abf5267e";
               final int IMAGESLIMIT
                                                 = 1000; // Depends of Vuforia license
24
25
26
               // Width parameters, are used to normalize the width
28
               final int DESIREDMAXWIDTH
                                                = 500;
               final int DESIREDMINWIDTH
29
                                                 = 250;
               final float MAXWIDTH
                                                 = 12.15f;
               final float MINWIDTH
                                                 = 0.03f;
               BufferedReader paintingsFile, widthsFile, serverAccessKeysFile;
               PostNewTarget newTarget = new PostNewTarget();
34
35
               String accessKey, secretKey, url, name, width, metadata;
37
               try {
38
                   serverAccessKeysFile = new BufferedReader(new FileReader(SERVERACCESSKEYSFILE));
39
                   accessKey = serverAccessKeysFile.readLine();
40
                   secretKey = serverAccessKeysFile.readLine();
41
               } catch (Exception e) {
                   accessKey = ACCESSKEY;
42
                   secretKey = SECRETKEY;
44
               3
45
               newTarget.setAccessKey(accessKey);
               newTarget.setSecretKey(secretKey);
47
               try {
48
                   paintingsFile = new BufferedReader(new FileReader(PAINTINGSFILE));
49
                   widthsFile = new BufferedReader(new FileReader(WIDTHSFILE));
51
                   url = paintingsFile.readLine();
                   width = widthsFile.readLine();
52
```

53

```
int counter = 1;
                                                 while (url != null && width != null && counter <= IMAGESLIMIT) {
                                                           trv {
57
                                                                    name = url.substring(url.lastIndexOf( str "/") + 1, url.lastIndexOf( str "."));
58
                                                                    System.out.println("Processing \"" + name + "\"...");
                                                                    metadata = new Scanner(new File( pathname: DESCRIPTIONSPATH + name + ".txt")).useDelimiter("\\Z").next();
59
                                                                    //Normalize width between <a href="https://www.desired.com">DESIREDMAXWIDTH</a> and <a href="https://www.desired.com">dos</a> and <a href="https://www.desired.com">https://www.desired.com</a> and <a href="https://www.desired.com">https://www.desired.com</a> and <a href="https://www.desired.com"/>a</a> and <a href="https://wwww.desired.com"/>a</a> and <a href="https://wwww.desired.com"/>a</a> and <a href="htt
                                                                    width = String.valueOf(((DESIREDMAXWIDTH - DESIREDMINWIDTH) * ((Float.parseFloat(width) - MINWIDTH)
63
                                                                                         / (MAXWIDTH - MINWIDTH))) + DESIREDMINWIDTH);
64
                                                                    newTarget.setTargetName(name);
                                                                    newTarget.setTargetWidth(width);
                                                                    newTarget.setTargetMetadata(metadata);
                                                                    newTarget.setTargetLocation(PAINTINGSPATH + name + ".jpg");
69
                                                                    newTarget.postTargetThenPollStatus();
                                                                    System.out.println("Image: \"" + name + "\" has been added to database." +
                                                                                         "\n[ Processed images: " + <u>counter</u> + " ]\n");
                                                           } catch (Exception e) {
                                                                     System.out.println("There was an error. Tha painting could not be uploaded to database (line: "
75
                                                                                        + <u>counter</u> + ")\n");
                                                           }
                                                           counter++;
78
                                                           url = paintingsFile.readLine();
79
                                                           width = widthsFile.readLine();
80
81
                                                paintingsFile.close();
82
                                                 widthsFile.close();
83
                                                 System.out.println("All images has been successfully added to the Vuforia Cloud Database.");
84
                                                 System.exit( status: 0);
85
86
                                      } catch (Exception e) {
87
                                                e.printStackTrace();
89
90
                 }
91
```

Σχήμα 9.36 Αρχείο CreateCloudDatabase.java

Στο παραπάνω αρχείο δημιουργείται ένα αντικείμενο της κλάσης "PostNewTarget", θέτονται οι κατάλληλες ιδιότητες (όνομα, πλάτος, metadata και URL εικόνας) διαβάζοντας τα δεδομένα, όπως ακριβώς έχουν δημιουργηθεί μετά την εκτέλεση του προγράμματος Python και στη συνέχεια καλείται η συνάρτηση "PostTargetThenPollStatus()" που στέλνει ένα HTTP post request στον ιστότοπο της Vuforia με σκοπό τη δημιουργία ενός νέου image target.

Σημείωση: Για καλυτέρα αποτελέσματα, το πλάτος του πίνακα κανονικοποιείται μεταζύ των αριθμών 250 και 500 πριν εκείνος εισαχθεί στη βάση δεδομένων (γίνεται χρήση του μέγιστου και του ελάχιστου πλάτους πινάκων που βρέθηκε από την ανάλυση των δεδομένων σε προηγουμένη ενότητα).

Εχει παρατηρηθεί κατά τη διάρκεια των δοκίμων ότι σε αυτές τις τιμές επιτυγχάνεται πιο ρεαλιστικό αποτέλεσμα στο πάνελ επαυξημένης πραγματικότητας με τις πληροφορίες του πίνακα. 🔮 PostNewTarget.java 🛛

```
package com.qualcomm.vuforia;
2
3
        import java.io.File;
        import java.io.IOException;
4
        import java.net.URI;
5
        import java.net.URISyntaxException;
6
        import java.util.Date;
7
8
9
        import org.apache.commons.codec.binary.Base64;
        import org.apache.commons.io.FileUtils;
        import org.apache.http.HttpResponse;
        import org.apache.http.client.ClientProtocolException;
13
        import org.apache.http.client.HttpClient;
        import org.apache.http.client.methods.HttpPost;
14
        import org.apache.http.client.methods.HttpUriRequest;
        import org.apache.http.entity.StringEntity;
        import org.apache.http.impl.client.DefaultHttpClient;
18
        import org.apache.http.impl.cookie.DateUtils;
19
        import org.apache.http.message.BasicHeader;
        import org.apache.http.util.EntityUtils;
        import org.json.JSONException;
        import org.json.JSONObject;
       import com.qualcomm.vuforia.SignatureBuilder;
24
25
26
27
   public class PostNewTarget implements TargetStatusListener {
28
29
            //Server Keys
             private String accessKey;
             private String secretKey;
             private String targetName;
             private String targetWidth;
             private String targetMetadata;
             private String targetLocation;
             private final String url = "https://vws.vuforia.com";
             private TargetStatusPoller targetStatusPoller;
39
             private final float pollingIntervalMinutes = 60; //poll at 1-hour interval
41
             private String postTarget() throws URISyntaxException, ClientProtocolException,
                                          IOException, JSONException {
43
                 HttpPost postRequest = new HttpPost();
45
                 HttpClient client = new DefaultHttpClient();
                 postRequest.setURI(new URI( str: url + "/targets"));
46
                 JSONObject requestBody = new JSONObject();
48
                 setRequestBody(requestBody);
                 postRequest.setEntity(new StringEntity(requestBody.toString()));
                 setHeaders(postRequest); // Must be done after setting the body
51
```

```
52
53
                 HttpResponse response = client.execute(postRequest);
                 String responseBody = EntityUtils.toString(response.getEntity());
54
                 System.out.println(responseBody);
                 JSONObject jobj = new JSONObject(responseBody);
57
58
                 String uniqueTargetId = jobj.has( key: "target_id") ? jobj.getString( key: "target_id") : "";
59
                 System.out.println("Created target with id: " + uniqueTargetId);
                 return uniqueTargetId;
62
             }
64
65
             private void setRequestBody(JSONObject requestBody) throws IOException, JSONException {
                 File imageFile = new File(getTargetLocation());
66
                 if(!imageFile.exists()) {
                     System.out.println("File location does not exist!");
68
69
                     System.exit( status: 1);
70
                 }
                 byte[] image = FileUtils.readFileToByteArray(imageFile);
71
                 requestBody.put("name", getTargetName()); // Mandatory
                 requestBody.put("width", Float.parseFloat(getTargetWidth())); // Mandatory
                 requestBody.put("image", Base64.encodeBase64String(image)); // Mandatory
74
                 requestBody.put("active_flag", 1); // Optional
75
                 requestBody.put("application_metadata", Base64.encodeBase64String(getTargetMetadata().
                         getBytes()); // Optional
77
78
             }
79
80
    0
             private void setHeaders(HttpUriRequest request) {
                 SignatureBuilder sb = new SignatureBuilder();
81
82
                 request.setHeader(new BasicHeader( name: "Date", DateUtils.formatDate(new Date()).
                         replaceFirst( regex: "[+]00:00$", replacement: "")));
83
84
                 request.setHeader(new BasicHeader( name: "Content-Type", value: "application/json"));
                 request.setHeader( s: "Authorization", s1: "VWS " + getAccessKey() + ":" + sb.
85
                         tmsSignature(request, getSecretKey()));
86
             }
87
88
89
             /**
              * Posts a new target to the Cloud database;
90
91
              * then starts a periodic polling until 'status' of created target is reported as 'success'.
92
              */
             public void postTargetThenPollStatus() {
93
                 String createdTargetId = "";
94
                 try {
                     createdTargetId = postTarget();
96
                 } catch (URISyntaxException | IOException | JSONException e) {
97
98
                     e.printStackTrace();
99
                     return;
```

```
// Poll the target status until the 'status' is 'success'
                  // The TargetState will be passed to the OnTargetStatusUpdate callback
                  if (createdTargetId != null && !createdTargetId.isEmpty()) {
                      targetStatusPoller = new TargetStatusPoller(pollingIntervalMinutes, createdTargetId,
                              getAccessKey(), getSecretKey(), target_listener: this );
                      targetStatusPoller.startPolling();
108
109
              // Called with each update of the target status received by the TargetStatusPoller
              @Override
113 🎊 @
              public void OnTargetStatusUpdate(TargetState target_state) {
                  if (target_state.hasState) {
                      String status = target_state.getStatus();
                      System.out.println("Target status is: " + (status != null ? status : "unknown"));
                      if (target_state.getActiveFlag() == true && "success".equalsIgnoreCase(status)) {
                          targetStatusPoller.stopPolling();
118
                          System.out.println("Target is now in 'success' status");
                  }
              3
              // Constructor, Getters & Setters
              public PostNewTarget(){}
125
126
              public String getTargetName() { return targetName; }
              public void setTargetName(String targetName) { this.targetName = targetName; }
              public String getTargetWidth() { return targetWidth; }
138
              public void setTargetWidth(String targetWidth) { this.targetWidth = targetWidth; }
139
              public String getTargetMetadata() { return targetMetadata; }
146
              public void setTargetMetadata(String targetMetadata) { this.targetMetadata = targetMetadata; }
              public String getTargetLocation() { return targetLocation; }
              public void setTargetLocation(String targetLocation) { this.targetLocation = targetLocation; }
158
159
              public String getAccessKey() { return accessKey; }
              public void setAccessKey(String accessKey) { this.accessKey = accessKey; }
              public String getSecretKey() { return secretKey; }
              public void setSecretKey(String secretKey) { this.secretKey = secretKey; }
175
              public static void main(String[] args) throws URISyntaxException, ClientProtocolException,
          ÷
                     IOException, JSONException {
                 PostNewTarget p = new PostNewTarget();
                 p.postTargetThenPollStatus();
179
180
         }
```

Σχήμα 9.37 Αρχείο PostNewTarget.java

Στο παραπάνω αρχείο υπάρχει μια κλάση που ορίζονται οι συναρτήσεις για την αποστολή του http post request και την παραλαβή της κατάστασης του image target μετά τη δημιουργία.

Παράδειγμα κώδικα JSON POST

```
POST /targets HTTP/1.1
1
2
     Host: vws.vuforia.com
     Date: Mon, 23 Apr 2012 12:45:19 GMT
3
     Authorization: VWS df8d23140eb443505c0661c5b58294ef472baf64:jHX6oLeqTXpynyqcvVC2MSHarhU
4
     Content-Type: application/json
5
6
      "name":"tarmac",
7
       "width":32.0,
8
9
       "image":"0912ba39x...",
       "application_metadata": "496fbb6532b3863460a984de1d980bed5ebcd507"
10
11
    }
```

Σχήμα 9.38 Παράδειγμα http post request

Παράδειγμα κώδικα απόκρισης JSON



Σχήμα 9.39 Παράδειγμα response

Μετά την εκτέλεση του παραπάνω κώδικα, μια νέα Vuforia Cloud Database που περιέχει image targets για όλους τους πίνακες της Εθνικής Πινακοθήκης, έχει δημιουργηθεί επιτυχώς.

vuforio	engine Home	Pricing Downlo	oads Library	Develop Support	Hello eRemountakis 🗸 Log Out	
License M	Manager Target Manager					
Target Mana	Target Manager > ARt Cloud Databas					
ARt_Cloud_Database Edit Name Type: Cloud License Key: ARt_Development_License_Key						
Target	s (10000) Database Access	s Keys				
Add 1	Target				Search by target name or target ID	
🗆 Tar	get Name	Rating ①	Recos	✓ Status ✓	Date Modified 🗸	
	millu-apoto-imerologio-tis	5 *** *	• 0	Active	Nov 08, 2020 01:02	
	o-dromos-pros-ton-golgoth	a *** *	• 0	Active	Nov 08, 2020 01:02	
	odophragma-no-7	****	• 0	Active	Nov 08, 2020 01:02	
	ilios	****	• 0	Active	Nov 08, 2020 01:17	
	o-iraklis-sikonei-tis-stiles-tis	-g ****	• 0	Active	Nov 08, 2020 01:02	
. 10	kephali-gatasgatos-enface	****	0	Active	Nov 08, 2020 01:17	
	o-pilatos-nipton-tas-heiras-t	tou 🔶 📩 🚖 📩	0	Active	Nov 08, 2020 01:02	
0 🧖	peristeria	****	• 0	Active	Nov 08, 2020 01:17	
	peristeria-2	****	• 0	Active	Nov 08, 2020 01:17	
	o-sampson-me-tis-pules-tis-	-g ★★★★	• 0	Active	Nov 08, 2020 01:02	
	sunthesi-me-ahladia	****	• 0	Active	Nov 08, 2020 01:17	
	xerolithia	****	• 0	Active	Nov 08, 2020 01:02	
	nekri-phusi-me-psaria-6785	4	• 0	Active	Nov 08, 2020 01:08	
	o-psaras-2	****	0	Active	Nov 08, 2020 01:08	
	enorasi-1	****	• 0	Active	Nov 08, 2020 01:02	

Σχήμα 9.40 Μέρος από Vuforia Cloud Database με τους πίνακες της Εθνικής Πινακοθήκης

Έχει δημιουργηθεί αντίστοιχος κώδικας για την επεξεργασία, διαγραφή, έλεγχο κατάστασης, λήψη και ενημέρωση image target, αλλά δεν χρησιμοποιήθηκε.

Το σύνολο των αρχείων κώδικα java που χρησιμοποιεί Vuforia Web Services (μαζί με τις βιβλιοθήκες που είναι απαραίτητες για την εκτέλεση του) με σκοπό την άμεση επεξεργασία της βάσης δεδομένων, βρίσκεται στον ακόλουθο σύνδεσμο: <u>ARt VWS</u>.

This PC > Documents > IdeaProjects > ARt VWS > src > com > qualcomm > vuforia						
c.	Name	Date modified	Туре	Size		
,s	🖺 CreateCloudDatabase.java	28/1/2021 10:24 πμ	IntelliJ IDEA	5 KB		
~ ام	😰 DeleteTarget.java	28/8/2014 6:55 µµ	IntelliJ IDEA	5 КВ		
	😰 GetAllTargets.java	28/8/2014 6:55 µµ	IntelliJ IDEA	2 KB		
ts 🛪	🐑 GetTarget.java	28/8/2014 6:55 µµ	IntelliJ IDEA	2 КВ		
*	😰 PostNewTarget.java	28/1/2021 10:23 πμ	IntelliJ IDEA	6 KB		
*	🐑 SignatureBuilder.java	28/8/2014 6:55 µµ	IntelliJ IDEA	4 KB		
foria Cloud Databse	😰 Summary.java	28/8/2014 6:55 µµ	IntelliJ IDEA	2 KB		
ikh	😰 TargetState.java	28/8/2014 6:55 µµ	IntelliJ IDEA	3 КВ		
ess	😰 TargetStatusListener.java	28/8/2014 6:55 µµ	IntelliJ IDEA	1 KB		
	😰 TargetStatusPoller.java	28/8/2014 6:55 µµ	IntelliJ IDEA	4 KB		
	😰 UpdateTarget.java	28/8/2014 6:55 µµ	IntelliJ IDEA	3 КВ		
ts						

Σχήμα 9.41 Αρχεία java για τη διαχείριση της Vuforia Cloud Database με χρήση Vuforia Web Services

📙 > Т	📙 > This PC > Documents > IdeaProjects > ARt VWS > src > lib						
		Name	Date modified	Туре	Size		
55	<u>م</u>	🛓 commons.io_2.0.1.jar	28/8/2014 6:55 μμ	Executable Jar File	167 KB		
de	Â	🍰 commons-codec.jar	28/8/2014 6:55 μμ	Executable Jar File	258 KB		
,15	<u> </u>	🥌 commons-logging-1.1.3.jar	28/8/2014 6:55 μμ	Executable Jar File	61 KB		
ts	*	🛓 httpclient-4.3.2.jar	28/8/2014 6:55 μμ	Executable Jar File	574 KB		
	*	≜ httpcore-4.3.2.jar	28/8/2014 6:55 μμ	Executable Jar File	276 KB		
	*	🍰 org.json-20120521.jar	28/8/2014 6:55 μμ	Executable Jar File	40 KB		
foria Clou	ud Databse						
ikh							
ess							

Σχήμα 9.42 Βιβλιοθήκες που απαιτούνται για την εκτέλεση του Vuforia Web Service

9.4 Σχεδιασμός εφαρμογής στο Unity

9.4.1 Λήψη και εγκατάσταση Unity Engine

Μετά τη συλλογή δεδομένων και τη δημιουργία Vuforia Cloud Database, ακολουθεί η υλοποίηση της εφαρμογής στο Unity Engine. Αρχικά, έγινε λήψη του Unity Hub. Το Unity Hub είναι μια αυτόνομη εφαρμογή που βελτιστοποιεί τον τρόπο εύρεσης, λήψης και διαχείρισης των έργων (projects) και εκδόσεων του Unity.





Η εγγραφή σε αυτό ως μαθητής χρησιμοποιώντας τα φοιτητικά στοιχεία είναι ο ευκολότερος τρόπος για τη δωρεάν απόκτηση επαγγελματικής αδείας χρήσης.

Unity Hub 2.4.2 Х ← Preferences \$ ER License \$ General License Management Professional ∃≟ Advanced Unity Pro offers access to enhanced benefits designed to support professional teams and freelancers who create single or multiplayer games or other real-time 2D, 3D, VR and AR interactive experiences. Activation: 2020/10/18 Expiration: 2021/10/20 CHECK FOR UPDATES Help — FAQ RETURN LICENSE

Σχήμα 9.44 Unity Hub Professional License

Στη συνέχεια μέσω του Unity Hub έγινε λήψη της τελευταίας σταθερής έκδοσης του Unity Engine.



Σχήμα 9.45 Εγκατεστημένη Έκδοση Unity: 2020.1.11fl

9.4.2 Εγκατάσταση Vuforia Engine

Για την εγκατάσταση του Vuforia Engine, πραγματοποιείται πλοήγηση στο Unity Asset Store και είσοδος με το λογαριασμό του Unity. Αναζήτηση για "Vuforia Engine" και στο αναδυόμενο παράθυρο, επιλέγεται το κουμπί "Open in Unity".



The Vuforia Core Samples provide complete Unity scenes that show how to create AR experiences for everyday objects and environments using Vuforia targets and trackers.

Σχήμα 9.46 Unity asset store Vuforia package

Με αυτήν την ενέργεια ανοίγει στο "Package Manager" του Unity το πακέτο της Vuforia έτοιμο για εγκατάσταση και στη συνέχεια επιλέγεται το κουμπί "Download".

A Inspector V Novigation	aakaga Ma	Accel Store			;
+ Packages: My Assets Sort: Nam	ne ↓ ▼ Filte	ers Clear Filters		\$ 4	
▶ Awesome Stylized Mage Tower	1.0 🛓	Vuforia Engine			
Buried Memories Volume 1: Yggdrasi		PTC			
▶ Day and Night Cycle With Scattering		Version 9.6.4 - December 17, 2020 asset store]		
▶ Fantasy Defensive Structures					
▶ Fire & Spell Effe	1.0.3 <u>+</u>	See the Unity Getting Started guide to learn more a	about working with Vuforia in Unity.		
		Get started with augmented reality using Vuforia E	ngine, the world's most widely used AR platform. Vuforia Er	gine makes it easy to dev	elop
▶ Free HDR Sky	1.0 🛓	Activision and more.			
Game Bar Creation System		The Vuforia Core Samples provide complete Unity Vuforia targets and trackers.	scenes that show how to create AR experiences for every	lay objects and environm	ents using
▶ Gradient GUI Bars		This asset store pack contains the Vuforia Engine : project from scratch.	SDK and can be used to add Vuforia Engine to existing pro	ects or start a new Vuforia	Engine
▶ Helicopter Pack	v4.0.1 <u>+</u>	Register as a Vuforia Developer when you are read	dy to add your own targets.		
▶ Rain Maker - 2D and 3D Rain Particl.		Less	r support.		
▶ Sci-Fi		Images & Videos			
▶ Simple Tor Ve					
Simple Wooden Bridge	1.0 🛓	🖉 🛇 vuloria			
▶ Snaps Art HD Asian Residential		- A -			
▶ Snaps Art HD Buried Memories Volu.		2 Real Contraction			
▶ Snaps Art HD School		View images & videos on Asset Store			
▶ Snaps Prototype Asian Residential		Package Size	Supported Unity Versions		
▶ Snaps Prototype Buried Memories V		512e. 20,72 WB (Number of mes. 141)	2018.2.0 01 higher		
Snaps Prototype School		Purchased Date			
▶ Tanks! Tutorial	1.97 <u>↓</u>	Polease Details			
► The Night Sky		9.6.4 (Current) - released on December 17, 2020			
▶ Two-Swords Warrior		Original - released on March 18, 2020			
▶ Vuforia Engi	9.6.4 <u>+</u>	Assigned Labels			
25 of 28 packages Loa					
Last update Jan 30, 22:58	C				Download

Σχήμα 9.47 Unity Package Manager: Vuforia Engine

Ακολουθεί αυτόματη λήψη και εγκατάσταση του πακέτου Vuforia Engine.

Από αυτό το σημείο και μετά, είναι δυνατή η χρήση εργαλείων της Vuforia Engine.



Σχήμα 9.48 Εργαλεία πακέτου Vuforia Engine

9.4.3 Unity: Βασικές έννοιες

Το Unity χρησιμοποιείται κυρίως για την κατασκευή παιχνιδιών, χωρίς όμως αυτό να σημαίνει ότι η ανάπτυξη εφαρμογών είναι αποτρεπτική, καθώς ανταποκρίνεται στις απαιτήσεις των εφαρμογών εξαιρετικά.

Game Objects

Ο,τιδήποτε δημιουργείται στο Unity θεωρείται ως ένα "Game Object".

Τα "Game Objects" είναι τα θεμελιώδη αντικείμενα της Unity που αντιπροσωπεύουν χαρακτήρες, αντικείμενα και τοπία. Δεν επιτυγχάνουν πολλά από μόνα τους, αλλά ενεργούν ως κοντέινερ για εξαρτήματα (components), τα οποία εφαρμόζουν την πραγματική λειτουργικότητα. Για παράδειγμα, ένα αντικείμενο "Light" δημιουργείται συνδέοντας ένα στοιχείο "Light" σε ένα "Game Object". [14]

Ορισμένες από τις διαθέσιμες κατηγορίες component που μπορούν να ενταχθούν σε ένα Game Object είναι οι παρακάτω:

- AR
- Audio
- Effects
- Event
- Layout
- Mesh
- Miscellaneous
- Navigation
- Physics 2D
- Physics
- Playable
- Rendering
- Scripts
- Tilemap
- UI
- Video
- New script

Add Component						
٩						
Component						
Layout	````` ^					
Mesh	>					
Miscellaneous	>					
Navigation						
Physics 2D	>					
Physics	>					
Playables	>					
Rendering	>					
Scripts	>					
Tilemap	>					
UI	>					
Video	>					
New script	> -					
Οι κυρίες κατηγορίες που είναι απαραίτητες για την υλοποίηση μιας εφαρμογής είναι οι "UI" (User Interface) μέσω της οποίας ορίζονται τα αντικείμενα της διεπαφής χρήστη και "Scripts" που ορίζουν τα αρχεία κώδικα σε C Sharp μέσω των οποίων η διεπαφή χρήστη αποκτά λειτουργικότητα και λογική.

Prefabs

Για γρηγορότερο και ευκολότερο σχεδιασμό, το Unity δίνει τη δυνατότητα κατά το σχεδιασμό ενός Game Object, να οριστεί ως "Prefab", δηλαδή προσαρμοσμένα αντικείμενα που θα αποτελούν μεγάλο μέρος της εφαρμογής ή του παιχνιδιού. Το "Prefab" λειτουργεί ως πρότυπο από το οποίο μπορούν να δημιουργηθούν άλλες παρουσίες prefab εντός του παιχνιδιού ή της εφαρμογής. Πρόκειται ουσιαστικά για ένα επαναχρησιμοποιούμενο Game Object, με διαφορετικές ιδιότητες κάθε φορά κατά βούληση.

Για παράδειγμα, κατά το σχεδιασμό ενός δάσους, θα μπορούσε να οριστεί ως prefab ένα Game Object που έχει γραφική απεικόνιση δέντρου. Στο δάσος υπάρχουν πολλά δέντρα τα οποία σχεδιάστηκαν μία μόνο φορά, εισήχθησαν στο παιχνίδι ως prefabs και άλλαξε μόνο το ύψος και η κλίση τους. [14]



Σχήμα 9.49 Game Object δέντρου ως Prefab

Assets

"Asset" είναι ένα στοιχείο που μπορεί να χρησιμοποιηθεί στη δημιουργία του παιχνιδιού ή της εφαρμογής. Αυτό το στοιχείο μπορεί να προέρχεται από ένα αρχείο που δημιουργήθηκε εκτός του Unity, όπως ένα μοντέλο 3D, ένα αρχείο ήχου, μια εικόνα ή οποιοσδήποτε άλλος τύπος αρχείων υποστηρίζεται από το Unity Engine. Υπάρχουν ,επίσης, ορισμένοι τύποι "asset" που μπορούν να δημιουργηθούν μέσα στο Unity ("Animator Controller", "Audio Mixer", "Render Texture" κ.α.). Κύρια πηγή για έτοιμα asset αποτελεί το Asset Store του Unity. [14]

Διεπαφή χρήστη Unity Engine

Η διεπαφή χρήστη του Unity Engine χρησιμοποιεί καρτέλες προκειμένου να γίνει περισσότερο αποδοτική η χρήση του.



Σχήμα 9.50 Unity Engine: Διεπαφή χρήστη (UI)

Οι βασικότερες καρτέλες είναι οι ακόλουθες:

• Scene

Οι σκηνές περιλαμβάνουν το περιβάλλον μέσα στο οποίο μπορεί να αναπτυχθεί και να σχεδιαστεί η εφαρμογή ή το παιχνίδι με την βοήθεια των εργαλείων που προσφέρει το Unity. Σε κάθε σκηνή, τοποθετούνται και απεικονίζονται αντικείμενα, εμπόδια και διακοσμήσεις.

Κάθε ενότητα του παιχνιδιού/ της εφαρμογής, ορίζεται ως «σκηνή». Με αυτόν τον τρόπο η υλοποίηση επιμερίζεται σε μικρότερα και διακριτά τμήματα. Σε περίπτωση που η υλοποίηση αναφέρεται σε κατασκευή εφαρμογής, κάθε σκηνή ορίζεται ως ενός μέρος της διεπαφής χρήστη. Εντός της σκηνής δημιουργούνται "Game Objects". [14]



Σχήμα 9.51 Καρτέλα σκηνής Unity

• Hierarchy

Η Ιεραρχία (Hierarchy) περιέχει κάθε "Game Object" της τρέχουσας σκηνής. Ορισμένα από αυτά είναι χαρακτηριστικά παραδείγματα αρχείων, όπως είναι τα τρισδιάστατα μοντέλα, και άλλα είναι παρουσίες Prefabs, προσαρμοσμένων αντικειμένων που θα

αποτελούν μεγάλο μέρος της εφαρμογής ή του παιχνιδιού. Καθώς τα αντικείμενα προστίθενται και αφαιρούνται στη σκηνή, θα εμφανίζονται και θα εξαφανίζονται ταυτόχρονα από την Ιεραρχία. Από προεπιλογή, τα Game Objects εμφανίζονται στο παράθυρο Ιεραρχίας με τη σειρά που έχουν δημιουργηθεί. [14]

Γονείς και παιδιά Game Object

Το Unity χρησιμοποιεί μια έννοια που ονομάζεται "Parenting". Κατά τη δημιουργία μιας ομάδας από GameObjects, το κορυφαίο GameObject της σκηνής ονομάζεται «πατέρας» GameObject και όλα τα GameObjects που ομαδοποιούνται από κάτω του ονομάζονται «παιδιά» GameObjects ή «παιδιά». Τα Game Object που αποτελούν παιδιά από παιδιά Game Objects ονομάζονται «απόγονοι».

Για παράδειγμα στην παρακάτω εικόνα το "Particle System 1" Game Object αποτελεί παιδί του "SHIP 2" Game Object και απόγονος του "Animation 3 Root" Game Object. Το "SHIP 2" Game Object αποτελεί με τη σειρά του παιδί του "Animation 3 Root" Game Object, που θεωρείται και ως πατέρας Game Object.



Σχήμα 9.52 Καρτέλα ιεραρχίας Unity

• Inspector

Τα παιχνίδια και οι εφαρμογές στο Unity αποτελούνται από πολλά Game Objects που περιέχουν φυσική, αρχεία κώδικα, ήχους ή άλλα γραφικά στοιχεία όπως τα "Lights". Η καρτέλα "Inspector" χρησιμοποιείται για την προβολή και επεξεργασία των ιδιοτήτων και των ρυθμίσεων των "Game Object", των "asset" και άλλων προτιμήσεων και ρυθμίσεων στο πρόγραμμα επεξεργασίας. [14]



Σχήμα 9.53 Καρτέλα Inspector Unity

• Project

Στην καρτέλα "Έργο" (Project) εμφανίζονται όλα τα αρχεία που σχετίζονται με το σύνολο της εφαρμογής ή του παιχνιδιού και είναι ο κύριος τρόπος με τον οποίο επιτυγχάνεται η πλοήγηση και η εύρεση στοιχείων ενεργητικού και άλλα αρχεία έργου στην εφαρμογή σας. Όταν ξεκινάτε ένα νέο έργο από προεπιλογή, αυτό το παράθυρο είναι ανοικτό. [14]



Σχήμα 9.54 Καρτέλα Έργου (Project) Unity

• Console

Η καρτέλα της κονσόλας εμφανίζει σφάλματα, προειδοποιήσεις και άλλα μηνύματα που δημιουργούνται από το Unity. Για τον καλύτερο εντοπισμό σφαλμάτων, οι προγραμματιστές μπορούν να εμφανίσουν τα δικά τους μηνύματα στην Κονσόλα χρησιμοποιώντας τις συναρτήσεις "Debug.Log()", "Debug.LogWarning()" και "Debug.LogError()". Η γραμμή εργαλείων του παραθύρου της κονσόλας έχει πολλές επιλογές που επηρεάζουν τον τρόπο εμφάνισης των μηνυμάτων. [14]



Σχήμα 9.55 Καρτέλα κονσόλας Unity

9.4.3.1 Γραμμή εργαλείων

Η γραμμή εργαλείων αποτελείται από επτά βασικές ομάδες ελέγχου. Κάθε μια από αυτές σχετίζεται με διαφορετικά μέρη του επεξεργαστή (Unity Engine).

Εικονίδιο	Επεξήγηση
🖑 💠 S 🖂 🔟	"Transform Tools" - χρησιμοποιείται για την προβολή σκηνής.
Pivot 🕼 Local	Μεταφόρτωση Gizmo, γραφικά που σχετίζονται με τα GameObjects στη σκηνή – επηρεάζουν την καρτέλα σκηνής.
	Κουμπιά αναπαραγωγής / παύσης / βηματικής εκτέλεσης - χρησιμοποιούνται για την προβολή παιχνιδιού ή εφαρμογής.
	"Cloud Button" - ανοίγει το Παράθυρο Υπηρεσιών Unity .
Account 🔻	Μενού λογαριασμών - χρησιμοποιείται για πρόσβαση στο λογαριασμό Unity .
Layers -	Στρώματα - ελέγχει ποια αντικείμενα εμφανίζονται στη σκηνή.
Layout -	Μενού Layout – για ρύθμιση προβολής διεπαφής χρήστη Unity

Πίνακας 9.1 Γραμμή εργαλείων Unity Engine



Σχήμα 9.56 Γραμμή εργαλείων Unity

Το Unity Engine είναι ένα εξαιρετικά μεγάλο εργαλείο δημιουργίας τόσο παιχνιδιών όσο και εφαρμογών, το οποίο διαθέτει πάρα πολλές δυνατότητες και λειτουργίες. Παραπάνω αναφέρθηκαν μόνο τα βασικά εργαλεία, προκειμένου να γίνει κατανοητή η υλοποίηση της εφαρμογής. Περισσότερες πληροφορίες πέραν αυτών που αναφέρθηκαν, υπάρχουν στο εγχειρίδιο χρήσης του Unity Engine, ακολουθώντας τον παρακάτω σύνδεσμο: <u>Εγχειρίδιο</u> χρήσης Unity Engine. [14]

9.4.4 Σκηνές εφαρμογής

Η υλοποίηση της εφαρμογής πραγματοποιείται σε τέσσερις "σκηνές" [16] που έχουν πλήρη αντιστοιχία με τις "οθόνες" που αναλυθήκαν στην ενότητα της διεπαφής χρήστη της εφαρμογής:

- Splash: Η αρχική οθόνη της εφαρμογής (splash screen)
- About: Οθόνη πληροφοριών και βοήθειας (about/help page)
- *Loading*: Οθόνη Αναμονής (loading page)
- *Paintings*: Οθόνη σάρωσης (main application page)



Σχήμα 9.57 Σκηνές που υλοποιούν την εφαρμογή στο Unity

Σκηνή Splash

Η σκηνή "Splash screen" εμφανίζεται κατά την εκκίνηση της εφαρμογής. Τα Game Objects που απαρτίζουν αυτή τη σκηνή φαίνονται παρακάτω:



Σχήμα 9.58 Καρτέλα Ιεραρχίας της σκηνής "Splash"

Πρόκειται για ένα μεγάλο καμβά (Canvas Game Object) που αποκτά χρώμα φόντου με το "BackgroundPanel" Game Object. Αυτός ο καμβάς χωρίζεται σε τρία βασικά μέρη: το πάνω μέρος αποτελείται από το "TopRegion" Game Object, το μεσαίο μέρος αποτελείται από το "MiddleRegion" Game Object, ενώ το κάτω μέρος αποτελείται από το "BottomRegion" Game Object. Το μεσαίο μέρος με τη σειρά του χωρίζεται στο αριστερό μέρος (MiddleLeftRegion Game Object), το κεντρικό (LogoPanel Game Object) και το δεξιά μέρος (MiddleRightRegion Game Object). Στο κεντρικό μέρος έχει τοποθετηθεί μια εικόνα με κείμενο "National Gallery by ARt" (δημιουργήθηκε αρχικά με PhotoShop) μέσω Image Game Object. Το τελικό



Σχήμα 9.59 Καρτέλα σκηνής "Splash"

Ο κώδικας που εκτελείται σε αυτή την σκηνή έχει οριστεί ως "Script Component" στο αρχικό Game Object "SplashScreen" και εμφανίζεται στην παρακάτω εικόνα.

```
C# AsyncSceneLoader.cs ×
```

```
1
         using UnityEngine;
         using UnityEngine.UI;
 2
 3
        ousing System.Collections;
 4
 5
         public class AsyncSceneLoader : MonoBehaviour
 6
         {
 7
             #region PUBLIC_MEMBERS
 8
             public float loadingDelay = 5.0F;
 9
             #endregion //PUBLIC_MEMBERS
10
             #region MONOBEHAVIOUR_METHODS
             void Start()
13
14
             {
15
                 StartCoroutine(LoadNextSceneAfter(loadingDelay));
             }
16
             #endregion //MONOBEHAVIOUR_METHODS
17
18
19
20
             #region PRIVATE_METHODS
             private IEnumerator LoadNextSceneAfter(float seconds)
             {
23
                 yield return new WaitForSeconds(seconds);
24
25
                 UnityEngine.SceneManagement.SceneManager.
26
                 LoadScene(UnityEngine.SceneManagement.SceneManager.GetActiveScene().buildIndex + 2);
             }
             #endregion //PRIVATE_METHODS
28
29
         }
30
```

Σχήμα 9.60 Κώδικας σκηνής "Splash"

Αρχικά ορίζεται μια σταθερά "loadingDelay" που δηλώνει τα δευτερόλεπτα εκτέλεσης της σκηνής και στη συνέχεια εκτελείται η συνάρτηση "LoadNextSceneAfter(seconds)". Μετά το πέρας των προκαθορισμένων δευτερολέπτων, εκτελείται η μεθεπόμενη σκηνή (Loading).

Σκηνή Loading

Η σκηνή "Loading screen" εμφανίζεται αμέσως μετά τη σκηνή "Splash screen".

Τα Game Objects που απαρτίζουν αυτή τη σκηνή φαίνονται παρακάτω:



Σχήμα 9.61 Καρτέλα ιεραρχίας σκηνής "Loading"

Πρόκειται για ένα μεγάλο καμβά (Canvas Game Object) που αποκτά χρώμα φόντου με το "Background" Game Object. Αυτό επιτυγχάνεται καθώς στο "Background" Game Object έχει προστεθεί μια μονόχρωμη εικόνα (μαύρου χρώματος) ως "component". Στο κέντρο του καμβά έχει τοποθετηθεί ένα "Spinner" Game Object. Σε αυτό το Game Object έχει προστεθεί ως "component" μια εικόνα που θυμίζει στροβιλιστεί (spinner).

Σημείωση: Όλα τα εικονίδια που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής, όπως στην συγκεκριμένη σκηνή το "spinner", δημιουργήθηκαν αρχικά στο PhotoShop.

Το τελικό αποτέλεσμα φαίνεται στο παρακάτω στιγμιότυπο σκηνής.



Σχήμα 9.62 Καρτέλα σκηνής "Loading"

Ο κώδικας που εκτελείται σε αυτή την σκηνή έχει οριστεί ως "Script Component" στον πατέρα Game Object "LoadingManager" και εμφανίζεται στις παρακάτω εικόνες:

```
C# LoadingScreen.cs ×
          using UnityEngine;
          using UnityEngine.UI;
          using System.Collections;
 4
         using UnityEngine.SceneManagement;
 5
 6
          public class LoadingScreen : MonoBehaviour
 7
          {
              #region PRIVATE_MEMBER_VARIABLES
 8
 9
              RawImage m_SpinnerImage;
10
              AsyncOperation m_AsyncOperation;
              bool m_SceneReadyToActivate;
              #endregion // PRIVATE_MEMBER_VARIABLES
              #region PUBLIC_MEMBER_VARIABLES
14
              public static string SceneToLoad { get; set; }
              #endregion // PUBLIC_MEMBER_VARIABLES
16
18
              public static void Run()
19
              {
20
                  SceneManager.LoadSceneAsync("2-Loading");
              }
              #region MONOBEHAVIOUR_METHODS
              void Start()
24
              {
                  m_SpinnerImage = GetComponentInChildren<RawImage>();
26
                  Application.backgroundLoadingPriority = ThreadPriority.Low;
28
                  StartCoroutine(LoadNextSceneAsync());
29
              3
30
31
              void Update()
              ł
                  if (m_SpinnerImage)
34
                  ł
                      if (!m_SceneReadyToActivate)
36
                      {
37
                          m_SpinnerImage.rectTransform.Rotate(Vector3.forward, 90.0f * Time.deltaTime);
38
                      }
39
                      else
40
                      {
41
                          m_SpinnerImage.enabled = false;
42
                      }
43
                  }
44
45
                  if (m_AsyncOperation != null)
46
                  {
                      if (m_AsyncOperation.progress < 0.9f)</pre>
48
                      {
                          Debug.Log("Scene Loading Progress: " + m_AsyncOperation.progress * 100 + "%");
                      }
50
                      else
                      {
                          m_SceneReadyToActivate = true;
                          m_AsyncOperation.allowSceneActivation = true;
                      }
                  }
              }
              #endregion // MONOBEHAVIOUR_METHODS
58
59
60
```

```
61
              #region PRIVATE_METHODS
62
             IEnumerator LoadNextSceneAsync()
63
              {
                  int nextSceneIndex = SceneManager.GetActiveScene().buildIndex + 1;
64
65
                  if (string.IsNullOrEmpty(SceneToLoad))
66
67
                  {
                      m_AsyncOperation = SceneManager.LoadSceneAsync(nextSceneIndex);
68
69
                  }
70
                  else
71
                  {
72
                      m_AsyncOperation = SceneManager.LoadSceneAsync(SceneToLoad);
73
                  }
74
                  m_AsyncOperation.allowSceneActivation = false;
75
76
77
                  yield return m_AsyncOperation;
78
             }
79
              #endregion // PRIVATE_METHODS
80
         }
81
```

Σχήμα 9.63 Κώδικας σκηνής "Loading"

Η εκτέλεση του κώδικας ξεκινάει με τη συνάρτηση "Start". Εκεί αρχικοποιείται η εικόνα του "spinner" και ξεκινά την εκτέλεση της η συνάρτηση "LoadNextSceneAsync", η όποια αναλαμβάνει να ξεκινήσει την επόμενη σκηνή. Η συνάρτηση "Update" εκτελείται με κάθε ανανέωση της οθόνης, δηλαδή σε κάθε «καρέ» της εφαρμογής. Σε αυτή τη συνάρτηση εκτελείται περιστροφή του spinner μέχρις ότου φορτωθεί η επόμενη σκηνή.

Σκηνή About

Η σκηνή "About screen" εμφανίζεται μόλις ο χρήστης πατήσει το κουμπί για

πληροφορίες/βοήθεια ((). Τα Game Objects που απαρτίζουν αυτή τη σκηνή φαίνονται παρακάτω:



Σχήμα 9.64 Καρτέλα ιεραρχίας σκηνής "About"

Τα Game Objects "AboutManager" και "SafeAreaManager" είναι τα δύο βασικότερα της σκηνής, πατέρες όλων των υπολοίπων Game Objects και χρώματος μπλε, καθώς έχουν οριστεί ως prefab. Ακολουθεί ανάλυση για τα βασικότερα Game Objects της σκηνής.

AboutManager: Game Object που έχει ως component το script το αρχείο "AboutManager.cs". Ο κώδικας του αρχείου εμφανίζεται παρακάτω. C# AboutManager.cs \times

```
using UnityEngine;
         using UnityEngine.UI;
         using TMPro;
         public class AboutManager : MonoBehaviour
         {
             #region PUBLIC_MEMBERS
             public string m_AboutScreen = "Paintings";
8
             #endregion //PUBLIC_MEMBERS
             #region PRIVATE_METHODS
             public void LoadNextScene()
              {
                 {\tt UnityEngine.SceneManagement.SceneManager.LoadScene(UnityEngine.SceneManagement)} \\
                     .SceneManager.GetActiveScene().buildIndex + 1);
16
             #endregion //PRIVATE_METHODS
             #region MONOBEHAVIOUR_METHODS
20
             void Start()
              ł
                 UpdateAboutText();
              }
25
             void Update()
              {
                 if (Input.GetKeyUp(KeyCode.Return) || Input.GetKeyUp(KeyCode.JoystickButton0))
29
                  {
                      LoadNextScene();
                 if (Input.GetKeyUp(KeyCode.Escape))
                  ſ
                      if (Application.isEditor)
36
                      {
         #if UNITY_EDITOR
38
                          UnityEditor.EditorApplication.isPlaying = false;
39
         #endif
                      }
                      else
                      {
                          Application.Quit();
                      3
                 }
             3
47
             #endregion // MONOBEHAVIOUR_METHODS
48
             void UpdateAboutText()
50
              {
                  AboutScreenInfo m_AboutScreenInfo = new AboutScreenInfo();
53
                 string title = m_AboutScreenInfo.GetTitle(m_AboutScreen);
                 string description = m_AboutScreenInfo.GetDescription(m_AboutScreen);
55
                 Text[] textElements = GetComponentsInChildren<Text>();
                  textElements[0].text = title;
                 TextMeshProUGUI textMeshProUGUI = GetComponentInChildren<TextMeshProUGUI>();
58
                  textMeshProUGUI.text = description;
59
60
             }
61
      ________}
```

Σχήμα 9.65 Κώδικας αρχείου "AboutManager.cs"

Η εκτέλεση του παραπάνω κώδικα ξεκινάει από τη συνάρτηση "Start", η οποία με τη σειρά της καλεί τη συνάρτηση "UpdateAboutText" και εκεί δημιουργείται ένα αντικείμενο της κλάσης "AboutScreenInfo". Στην κλάση αυτή δημιουργούνται όλες οι πληροφορίες που θα εισαχθούν στην οθόνη «Βοήθεια» (Περιγραφή εφαρμογής, Οδηγίες χρήσης, Πληροφορίες κινητής συσκευής και Οροί χρήσης). Μετά τη δημιουργία των πληροφοριών, τα δεδομένα επιστρέφουν και αρχικοποιούν τα GameObjects της σκηνής κατά τη διάρκεια εκτέλεσης της. Ο κώδικας του αρχείου "AboutScreenInfo" παρουσιάζεται στη συνέχεια.

C# AboutScreenInfo.cs ×

using System.Collections.Generic; using UnityEngine; ousing Vuforia; 3 4 public class AboutScreenInfo 5 { 6 7 #region PRIVATE_MEMBERS 8 readonly Dictionary<string, string> titles; 9 readonly Dictionary<string, string> descriptions; 10 #endregion // PRIVATE_MEMBERS 14 #region PUBLIC_METHODS 16 public string GetTitle(string titleKey) 18 ł 19 return GetValuefromDictionary(this.titles, titleKey); 20 } public string GetDescription(string descriptionKey) { return GetValuefromDictionary(this.descriptions, descriptionKey); } 26 #endregion // PUBLIC_METHODS 28 29 30 #region PRIVATE_METHODS 31 string GetValuefromDictionary(Dictionary<string, string> dictionary, string key) { 34 if (dictionary.ContainsKey(key)) 35 { 36 string value; 37 dictionary.TryGetValue(key, out value); 38 return value; 39 } 40 41 return "Key not found."; 42 } 43 #endregion // PRIVATE_METHODS 44 45 46 #region CONSTRUCTOR 47 48 public AboutScreenInfo() 49 50 { // Init our Title Strings this.titles = new Dictionary<string, string>() 54 ł { "Paintings", "ARt: Help" }, 57 58 };

```
// Init our Common Cache Strings
                  string vuforiaVersion = Vuforia.VuforiaUnity.GetVuforiaLibraryVersion();
                  string unityVersion = UnityEngine.Application.unityVersion;
 63
                  UnityEngine.Debug.Log("Vuforia Engine " + vuforiaVersion + "\nUnity " + unityVersion);
                  string vuforia = Vuforia.VuforiaRuntime.Instance.InitializationState
                                   != Vuforia.VuforiaRuntime.InitState.NOT_INITIALIZED
67
                                           ? "<#23B200>Yes</color>"
68
                                           : "<color=red>No</color>";
                  string fusionProvider = Vuforia.VuforiaRuntimeUtilities.GetActiveFusionProvider().ToString();
70
                  string description = "\n<size=26>Description:</size>";
                  string keyFunctionality = "<size=26>Key Functionality:</size>";
                  string targets = "<size=26>Targets:</size>";
                  string instructions = "<size=26>Instructions:</size>";
74
75
                  string footer =
76
                      "<size=26>More Info:</size>" +
                      "\n<size=22>System:</size>" +
                      "\n• Device Name: " + SystemInfo.deviceName +
78
                      "\n• Device Model: " + SystemInfo.deviceModel +
                      "\n• Operating System: " + SystemInfo.operatingSystem +
80
81
                      "\n• System Memory: " + SystemInfo.systemMemorySize +
                      "\n• Processor Count: " + SystemInfo.processorCount +
82
                      "\n• Processor Frequency: " + SystemInfo.processorFrequency +
83
                      "\n" +
84
                      "\n<size=22>Graphics:</size>" +
85
                      "\n• Graphics Memory: " + SystemInfo.graphicsMemorySize +
86
                      "\n• Device Name: " + SystemInfo.graphicsDeviceName +
87
                      "\n• Device Vendor: " + SystemInfo.graphicsDeviceVendor +
88
                      "\n• Device Type: " + SystemInfo.graphicsDeviceType +
                      "\n• Device Version: " + SystemInfo.graphicsDeviceVersion +
                      "\n• MultiThreaded: " + SystemInfo.graphicsMultiThreaded +
                      "\n" +
93
                      "\n<size=22>Links:</size>" +
                      "\n• <link=https://arttermsandconditions.000webhostapp.com/terms><color=blue>"
94
                      +"<u>Terms of Use</u></color></link>" +
95
                      "\n":
96
                  string targetPDFsURL = "<link=https://library.vuforia.com/content/vuforia-library/"</pre>
97
                                         +"en/articles/Solution/sample-apps-target-pdfs.html>";
99
                  // Init our Description Strings
                  this.descriptions = new Dictionary<string, string>();
101
                  // Paintings
103
104
                  this.descriptions.Add(
105
                      "Paintings",
106
                      description +
107
                      "\nScan any painting of National Gallery for more information on that work of art." +
                      "\n\n" +
109
                      instructions +
                      "\n• Point camera at a painting to view info" +
110
                      "\n• Click on the information that appeared to scan again" +
                      "\n\n" +
                      footer + "\n");
              }
              #endregion // CONSTRUCTOR
        白}
```

Σχήμα 9.66 Κώδικας αρχείου "AboutScreenInfo.cs"

Όλα τα υπόλοιπα παιδιά του AboutManager Game Object συνθέτουν την οθόνη «Βοήθεια». Η ανάλυση των αντικειμένων που ακολουθεί εστιάζει στα βασικά χαρακτηριστικά των Game Object, καθώς δεν έχουν πολλά components και παρουσία αρχείων κώδικα.

- > AboutCanvas: Game Object τύπου καμβά που οριοθετεί την οθόνη «Βοήθεια».
- RootPanel: Ορίζει τον τρόπο εμφάνισης και την απόσταση που θα έχουν τα παιδιά αντικείμενα κατά την εμφάνιση τους. Κάτι τέτοιο στο Unity ονομάζεται "Vertical Layout Group".
- TitlePanel: Ορίζει το φόντο της πάνω μπάρας έχοντας ως component μια μονόχρωμη γκρι εικόνα.

			Т	Title				>
							-	

- Title: Game Object κειμένου, εκεί τοποθετείται ο τίτλος «ARt: Βοήθεια» κατά την εκτέλεση
- ForwardButton: Προσθέτει την εικόνα για το μαύρο βελάκι, καθώς επίσης λειτουργεί και ως κουμπί που πυροδοτεί τη συνάρτηση "LoadNextScene" από το αρχείο "AboutManager.cs".
- BodyPanel: Ορίζει τον τρόπο εμφάνισης και την απόσταση που θα έχουν τα παιδιά αντικείμενα (Vertical Layout Group).
- ScrollPanel: Παρέχει τη δυνατότητα της κυλιόμενης απεικόνισης (scroll) στην περιοχή εμφάνισης πληροφοριών.
- ContentPanel: Ορίζει το φόντο της οθόνης έχοντας ως component μια μονόχρωμη άσπρη εικόνα.

Description: Game Object κειμένου. Εκεί τοποθετούνται τα δεδομένα της οθόνης «Βοήθεια» κατά το χρόνο εκτέλεσης η περιγραφή κάθε πίνακα.

# Scene 👁 C	ame ▼ 20 ● who st ▼ and 0 共 ▼			
Shaded				x x
		Title	>	≪Persp
	Description: <loaded at="" runtime=""> Key Functionality: • <loaded at="" runtime=""></loaded></loaded>			
	Physical Targets: • <loaded at="" runtime=""></loaded>			
	• <loaded at="" runtime=""> Build Version Info:</loaded>			

Το τελικό αποτέλεσμα φαίνεται στο παρακάτω στιγμιότυπο σκηνής:

Σχήμα 9.67 Καρτέλα σκηνής "About"

SafeAreaManager: Game Object, το οποίο αναλαμβάνει τον τρόπο με τον οποίο η προβολή των αντικειμένων της σκηνής θα πραγματοποιηθεί εντός των περιθωρίων της οθόνης της κινητής συσκευής. Γι' αυτό το λόγο, έχει δημιουργηθεί ειδικό αρχείο script που επιτελεί τον παραπάνω σκοπό. Το αρχείο ονομάζεται "SafeAreaManager.cs" και ο κώδικας του φαίνεται παρακάτω.

Σημείωση: Ο παρακάτω κώδικας αποτελεί προϊόν της PTC (εταιρεία που δημιούργησε το Vuforia Engine) και χρησιμοποιήθηκε αυτούσιος εφόσον καλύπτει τις ανάγκες της εφαρμογής.[16]

C# SafeAreaManager.cs ×

```
using System;
         using UnityEngine;
         using UnityEngine.UI;
         public class SafeAreaManager : MonoBehaviour
         {
 6
              #region PRIVATE_MEMBERS
8
9
             [System.Serializable]
10
             class SafeAreaRect
             ſ
                  public RectTransform rectTransform = null;
                  [Header("Apply Safe Area Constraints")]
                 public bool top = false;
                  public bool bottom = false;
             }
18
             [Header("Global Unsafe Area Settings (Per-Scene)")]
             [Tooltip("Unsafe Area Colors can be changed programmatically at runtime.")]
19
20
              [SerializeField] RectTransform topArea = null;
             [SerializeField] RectTransform bottomArea = null;
              [SerializeField] Color topAreaColor;
              [SerializeField] Color bottomAreaColor;
              [Tooltip("Safe Area Margin reduces the Safe Area by the specified amount at the Top/Bottom boundaries. " +
                       "It is useful for testing Safe Area Behaviour in PlayMode.")]
              [Range(0,100)] // Max range value is arbitrary for example purposes
              [SerializeField] private int SafeAreaMargin = 0;
28
29
              [Header("Apply Safe Area Constraints to RectTransforms")]
              [SerializeField] SafeAreaRect[] safeAreaRects = null;
30
             ScreenOrientation lastOrientation;
              Rect lastSafeArea = new Rect(0, 0, 0, 0);
             Rect safeArea;
             Image topAreaImage = null;
36
              Image bottomAreaImage = null;
              bool colorsChanged => (topAreaColor != topAreaImage.color) || (bottomAreaColor != bottomAreaImage.color);
38
              #endregion // PRIVATE_MEMBERS
40
41
             #region MONOBEHAVIOUR_METHODS
42
              void Awake()
45
              {
46
                  if (!topArea || !bottomArea)
47
                  {
48
                      Debug.LogWarning("Either topArea or bottomArea is null."
49
                                       + " Programmatically getting the required references.");
50
                      SetAreaRectTransforms();
                  }
52
                  // cache our unsafe area image components
                  this.topAreaImage = this.topArea.GetComponent<Image>();
```

```
55
                   this.bottomAreaImage = this.bottomArea.GetComponent<Image>();
                  // Set the unsafe area colors using Inspector values
57
58
                  SetAreaColors(this.topAreaColor, this.bottomAreaColor);
59
                  this.safeArea = GetSafeArea();
61
              }
63
              void SetAreaRectTransforms()
              {
                  var images = GetComponentsInChildren<Image>();
65
                  if (images.Length != 2)
67
                  {
                      Debug.LogError($"SafeAreaManager must have exactly two children with Image components attached.");
68
69
                       return;
70
                  }
71
                  topArea = images[0].rectTransform;
                  bottomArea = images[1].rectTransform;
74
              }
              Rect GetSafeArea()
 76
              {
78
                  return new Rect(
                      Screen.safeArea.x,
79
                       Screen.safeArea.y + this.SafeAreaMargin,
80
81
                       Screen.safeArea.width,
                       Screen.safeArea.height - (this.SafeAreaMargin * 2));
82
83
              }
84
85
              void Start()
86
              {
87
                  this.lastOrientation = Screen.orientation;
89
                  Refresh();
              }
90
91
              void Update()
92
93
              {
94
                  Refresh();
95
              }
96
97
              #endregion // MONOBEHAVIOUR_METHODS
98
99
              #region PRIVATE_METHODS
100
101
              void Refresh()
103
              {
104
                  this.safeArea = GetSafeArea();
105
106
                  if ((this.safeArea != this.lastSafeArea) || (Screen.orientation != this.lastOrientation))
107
                  {
108
                       ApplySafeArea();
```

```
109
                       UpdateUnsafeArea();
110
                   }
                   if (this.colorsChanged)
                   {
114
                       SetAreaColors(this.topAreaColor, this.bottomAreaColor);
                   }
116
               }
               void ApplySafeArea()
118
119
               {
                   this.lastSafeArea = this.safeArea;
120
                   this.lastOrientation = Screen.orientation;
                   foreach (SafeAreaRect areaRect in this.safeAreaRects)
                   {
                       var anchorMin = this.safeArea.position;
                       var anchorMax = this.safeArea.position + this.safeArea.size;
                       anchorMin.x /= Screen.width;
128
129
                       anchorMin.y = areaRect.bottom ? anchorMin.y / Screen.height : 0;
130
                       anchorMax.x /= Screen.width;
                       anchorMax.y = areaRect.top ? anchorMax.y / Screen.height : 1;
                       if (Screen.orientation == ScreenOrientation.LandscapeLeft ||
                           Screen.orientation == ScreenOrientation.LandscapeRight)
                       {
136
                           anchorMin.x = 0;
                           anchorMax.x = 1;
138
                       }
139
140
                       areaRect.rectTransform.anchorMin = anchorMin;
                       areaRect.rectTransform.anchorMax = anchorMax;
                   }
               }
               void UpdateUnsafeArea()
               {
                   var anchorMin = this.safeArea.position;
148
                   var anchorMax = this.safeArea.position + this.safeArea.size;
149
150
                   anchorMin.x /= Screen.width;
                   anchorMin.y = anchorMin.y / Screen.height;
                   anchorMax.x /= Screen.width;
                   anchorMax.y = anchorMax.y / Screen.height;
                   SetUnsafeAreaSizes(anchorMin.y, anchorMax.y);
                   SetAreaColors(this.topAreaColor, this.bottomAreaColor);
               }
158
159
               void SetUnsafeAreaSizes(float safeAreaAnchorMinY, float safeAreaAnchorMaxY)
160
               {
                   this.topArea.anchorMin = new Vector2(0, safeAreaAnchorMaxY);
```

```
this.topArea.anchorMax = Vector2.one;
                  this.bottomArea.anchorMin = Vector2.zero;
                  this.bottomArea.anchorMax = new Vector2(1, safeAreaAnchorMinY);
              }
168
              #endregion // PRIVATE_METHODS
169
170
              #region PUBLIC_METHODS
              public void AddSafeAreaRect(RectTransform rect, bool applyTopConstraint, bool applyBottomConstraint)
              ł
                  Array.Resize(ref this.safeAreaRects, this.safeAreaRects.Length + 1);
176
                  this.safeAreaRects[this.safeAreaRects.Length - 1] = new SafeAreaRect
177
178
                  {
                      rectTransform = rect,
179
180
                      top = applyTopConstraint,
                      bottom = applyBottomConstraint
                  };
                  ApplySafeArea();
              }
187
              public void SetAreasEnabled(bool topAreaEnabled, bool bottomAreaEnabled)
              {
                  this.topAreaImage.enabled = topAreaEnabled;
190
                  this.bottomAreaImage.enabled = bottomAreaEnabled;
              }
             /// <summary>
194
              /// Sets the area colors programmatically and updates Inspector colors.
195
              /// </summary>
              /// <param name="topColor">Top color.</param>
              /// <param name="bottomColor">Bottom color.</param>
198
              public void SetAreaColors(Color topColor, Color bottomColor)
199
              {
200
                  // update Inspector-level colors to match programmatic ones
201
                  this.topAreaColor = topColor;
                  this.bottomAreaColor = bottomColor;
202
203
                  // assign the colors
204
                  this.topAreaImage.color = this.topAreaColor;
205
                  this.bottomAreaImage.color = this.bottomAreaColor;
207
              }
208
              #endregion // PUBLIC_METHODS
209
        ≙}
210
```

Σχήμα 9.68 Κώδικας αρχείου SafeAreaManager.cs

Σκηνή Painting

Η σκηνή "Painting screen" εμφανίζεται αμέσως μετά τη σκηνή "Loading screen". Πρόκειται για τη μεγαλύτερη σκηνή στην οποία πραγματοποιούνται σχεδόν όλες οι λειτουργίες της εφαρμογής. Τα Game Objects που απαρτίζουν αυτή τη σκηνή φαίνονται παρακάτω:



Σχήμα 9.69 Καρτέλα ιεραρχίας σκηνής "Paintings"



Το τελικό αποτέλεσμα φαίνεται στο παρακάτω στιγμιότυπο σκηνής:

Σχήμα 9.70 Καρτέλα σκηνής "Paintings" Game Object

Ακολουθεί αναλυτική περιγραφή για τα βασικότερα Game Objects σε συνδυασμό με όλα τα components που διαθέτουν, προκειμένου να γίνουν περισσότερο κατανοητά τα βήματα που ακολουθήθηκαν για την υλοποίηση της σκηνής "Paintings" (η σειρά ανάλυσης και περιγραφής είναι ίδια με εκείνη που εμφανίζονται στην καρτέλα της ιεραρχίας). Στα περισσότερα Game Objects, το βασικότερο component είναι η κατηγορία script (κώδικας σε C Sharp), καθώς με αυτό τον τρόπο εισάγεται η λειτουργικότητα και η λογική στη σκηνή.

ARCamera: Πρόκειται για Game Object της Vuforia Engine τύπου AR κάμερας, έχοντας κατά τη δημιουργία του ως components δύο βασικά scripts: "Vuforia Behavior" (εκτέλεση βασικών ενεργειών της Vuforia Engine) και "InitErrorHandler" (διαχείριση σφαλμάτων), τα οποία έχουν τοποθετηθεί σε σταθερή απόσταση από το πάνελ πληροφοριών. Παρακάτω φαίνονται οι ιδιότητες του συγκεκριμένου Game Object:

🛈 Inspector 🛛 🔀 Navigation 🛛 🖬 Package Manager	🛱 Asset Store			a :
ARCamera				Static 🗸
Tag Untagged	▼ Laye	er Default		
Transform				0 :± :
Position	x o	Y 300	7 0	
Potation	x 90	× 0	7 0	
Scale	x 30 x 1	Y 1	Z 0	
				⊎ .⊢:
Clear Flags	Solid Color			
Background	Evonything			<u></u>
	Lverytning			
Projection	Perspective			
FOV Axis	Vertical			•
Field of View		•		60
Physical Camera				
Clipping Planes	Near 0.05			
	Far 2000			
Viewport Rect	X 0	Y 0		
	w 1	н 1		
Depth				
Rendering Path	Use Graphics Settin	igs		
Target Texture	None (Render Textu	ıre)		
Occlusion Culling	~			
HDR	Off			
MSAA Allow Dunomia Desclution	Off			
Allow Dynamic Resolution				
🎧 🗹 Audio Listener				07≓ :
🔻 🕢 Vuforia Behaviour (Script)				07‡ :
Upgra	de to Vuforia Engine 9.6.3 via	a Package Manager		
World Center Mode	FIRST_TARGET			
	Open Vuforia Engine con	figuration		
# Init Error Handler (Script)				0 ‡ i
Script	# InitErrorHandler			
# Scene Orientation (Script)				0 ‡ ;
Script	SceneOrientation			
Scene Orientation	AUTOROTATION			
	Add Component			

Σχήμα 9.71 Καρτέλα Inspector του "ARCamera" Game Object

Πατώντας στο κουμπί "Open Vuforia Engine configuration" δίνεται η δυνατότητα εισαγωγής του κλειδιού χρήσης που δημιουργήθηκε σε προηγουμένη ενότητα (*Bλ. σχήμα «Καρτέλα* "License Key" της αδείας χρήσης "Art_Development_License_Key"»). Εδώ είναι πολύ σημαντικό να οριστεί η τιμή "1" στην παράμετρο "Max simultaneous Tracked Images".

0 Inspector	🔀 Navigation	🖬 Package Manager	Asset Store	a :
Vufori	a Configuration (\	/uforia Configuration)	0	***
			0	pen
Global Vuforia Vers	sion		9.5.4	
A new	Vuforia Engine version	is available: 9.6.3		
		Upgrade	e to Vuforia Engine 9.6.3 via Package Manager	-1
Learn More				
🚺 We str	rongly recommend devi	elopers to encrypt their key for	r enhanced security. For more information refer to the article below.	
Open Librai	ry Article			
App License	e Key		AazTVQH/////AAABmZ/l2llqD0mBhF0lwUgVTaUblRdMxy/ODadNSq91F3z/NvOuxU OOWxwQmOuR/APbz3aswgTRNk7eZCQGLpCenPwQr7zLwCludIbpVhaoT7J0+C/ es4xOVpJLvzC+JgYUjPWxHpvTBoeceqMhWFXhu1xdx7QbxZGWxyxvaTu2RVRsa r+AptmqE60+TzAr3v6C/W+tkTr/W0KEZQfuhio8tehLQKwkULayda4F6gobE0Bc/V NuRPeMvTLw2IBVM0PlusKuge3erx+ySGzSOwUIYViV9P9reN31AeM6/vPJWdWl/ BS3v15nB2UB4DINoWjBRIOxZp6f7+1kDIvfPydCls5uidj9	16aO (4fln H+p Xos 7LB
Delayed	ticlization		Add License	
Camera De	vice Mode		MODE DEFAULT	-
Max Simulta	aneous Tracked Ima	ages	1	
Max Simulta	aneous Tracked Ob	jects	1	
Virtual Scer	ne Scale Factor			
The fa	ictor above configures nt will not be scaled. No	how many Unity scene units co ote that on HoloLens and Magic	orrespond to one meter in the physical world. Poses reported by Vuforia will be transformed accordingly. S cLeap, a factor other than 1.0 is not supported.	
Load Objec	t Targets on Detect			
Trained Tar	rgets Continuous Se	arch	\checkmark	
▼ Databases	an found			
Disable mod	del extraction from d	databases	Add Database	
▼ Video Back	around			
Enable vide	eo background		~	
Video Back	ground Shader		S Custom/VideoBackground	
Number Div	visions		2	
Device Track Device	cker		7	
	opers looking for Extend	led Tracking functionality shoul	ld enable the Positional Device Tracker.	
Open Libra	rv Article			
Android Se	ttings			
ARCore Rec	quirement		OPTIONAL	•
Include ARC	Core library			
▼ Play Mode				
Open Play I	Mode Documentatio	on	WERCAM	
No we	bcam profile has been	found for your webcam model:	: 'HP Webcam'.	
A defa	ault profile will be used.	Diou Modo porferme well		
You ca	an promes ensure that an create a custom pro ts\National Gallery Apr	play mode performs well with y offile for your camera by editing o\Library\PackageCache\com r	your weocann. 'C:\Users\remou\Documents\Documents\Unity nc yuforia engine@9.5.4/Vuforia/Editor/Editor/Eesources\webcamprofiles.xm#	
Asset Labels		ore of a charge cache (com.)	And a construction of the	
				۲
AssetBundle	None			

Σχήμα 9.72 ARCamera (Vuforia configuration)

C# SceneOrientation.cs ×

```
1 😨
         using UnityEngine;
        public class SceneOrientation : MonoBehaviour {
 3
 4
 5
             #region PUBLIC_MEMBERS
 6
             public enum Orientation {
 7
 8
                 AUTOROTATION,
9
                 PORTRAIT,
10
                 LANDSCAPE
             }
13
             public Orientation sceneOrientation;
14
15
             #endregion // PUBLIC_MEMBERS
16
18
             #region MONOBEHAVIOUR_METHODS
19
             void Awake()
20
             {
                 SetSceneOrientation();
23
             }
24
25
             #endregion // MONOBEHAVIOUR_METHODS
26
27
             #region PRIVATE_METHODS
28
29
             private void SetSceneOrientation()
30
             {
                 switch (sceneOrientation)
                 {
                     case Orientation.AUTOROTATION:
35
                         Screen.orientation = ScreenOrientation.AutoRotation;
36
                         break;
                     case Orientation.PORTRAIT:
                         Screen.orientation = ScreenOrientation.Portrait;
38
39
                         break;
40
                     case Orientation.LANDSCAPE:
41
                         Screen.orientation = ScreenOrientation.LandscapeLeft;
42
                         break;
43
                 }
             }
44
             #endregion // PRIVATE_METHODS
46
48
       A}
```

Σχήμα 9.73 Κώδικας αρχείου SceneOrientation.cs

Το παραπάνω script καθορίζει τον προσανατολισμό της οθόνης. Στην εφαρμογή έχει τεθεί η επιλογή "AUTOROTATION", για αυτόματη περιστροφή.

PositionFor2D: Αποτελεί παιδί του ARCamera Game Object έχοντας ως μοναδικό component

το AnimationsManager script. Ο κώδικας του script αρχείου εμφανίζεται παρακάτω.

C# AnimationsManager.cs × using UnityEngine; 3 b/// <summary> /// This class handles the animations of the augmentation /// </summary> 6 public class AnimationsManager : MonoBehaviour { 8 #region PRIVATE_MEMBERS 9 // Reference to the augmentation object to animate 10 GameObject mAugmentationObject; bool mIsTracking; bool mDoAnimationTo2D; bool mDoAnimationTo3D; bool mIsShowingOverlay; #endregion //PRIVATE_MEMBERS 16 #region MONOBEHAVIOUR_METHODS 18 19 void Start() 20 { // Determine how far the Canvas should be from Camera in 2D mode float uiDistanceFromCamera = (Screen.dpi > 260) ? 550f : 700f; // Set our 2D mode location transform to be specified distance away transform.localPosition = new Vector3(0, 0, uiDistanceFromCamera); 3 26 void Update() 29 { if (mAugmentationObject == null) 30 { mDoAnimationTo2D = false; mDoAnimationTo3D = false; return; } // Performs Animation to 2D if (mDoAnimationTo2D) 38 ł // Updates AugmentedObject position and rotation in every frame 40 mAugmentationObject.transform.position = Vector3.Lerp(mAugmentationObject.transform.position, 41 transform.position, Time.deltaTime * 5.0f); mAugmentationObject.transform.rotation = Quaternion.Slerp(mAugmentationObject.transform.rotation, 46 transform.rotation, Time.deltaTime * 5.0f); 48 49 // Checks for object distance to check animation finish 50 if (Vector3.Distance(mAugmentationObject.transform.position, transform.position) < 0.001f) { mDoAnimationTo2D = false; 54

```
// Performs Animation to 3D
 56
                  if (mDoAnimationTo3D)
 58
                  {
 59
                      // Updates AugmentedObject position and rotation in every frame
 60
                      mAugmentationObject.transform.localPosition = Vector3.Lerp(mAugmentationObject.transform.localPosition,
 61
                                                                                 new Vector3(0, 0, 0),
                                                                                 Time.deltaTime * 5.0f);
                      mAugmentationObject.transform.localRotation = Quaternion.Slerp(mAugmentationObject.transform.localRotation,
                                                                                      Quaternion.identity,
                                                                                      Time.deltaTime * 5.0f);
 68
                      // Checks for object distance to check animation finish
 69
                      if (Vector3.Distance(mAugmentationObject.transform.localPosition, new Vector3(0, 0, 0)) < 0.001f)
 70
                      {
                          mDoAnimationTo3D = false;
                      }
              ļ
 75
              #endregion //MONOBEHAVIOUR_METHODS
76
 78
              #region PUBLIC_METHODS
 79
              // Starts playing animation to 2D
 80
              public void PlayAnimationTo2D(GameObject augmentationObject)
 81
              {
                  mAugmentationObject = augmentationObject;
 82
                  // Checks that the system is already tracking
 84
 85
                  if (!mIsTracking)
 86
                  {
 87
                      Debug.Log("Starting animation-to-2D from untracked target position.");
 88
                  }
 89
 90
                  mDoAnimationTo2D = true;
 91
                  mIsShowingOverlay = true;
                  // Updates state variables
 93
                  mDoAnimationTo3D = false;
 94
 95
                  mIsTracking = false;
 96
              }
 97
 98
              // Sets the initial animation flags
99
              public void SetInitialAnimationFlags()
100
              {
101
                  mDoAnimationTo2D = true;
102
                  mDoAnimationTo3D = false;
103
                  mIsShowingOverlay = true;
104
              }
105
```

```
106
              // Starts playing animation to 3D
107
              public void PlayAnimationTo3D(GameObject augmentedObject)
108
              ł
109
                  mAugmentationObject = augmentedObject;
110
                  mDoAnimationTo2D = false;
                  // Checks that the system is showing the overlay right now
                  if (mIsShowingOverlay)
                  {
                      mDoAnimationTo3D = true;
116
                     // Updates the augmented object initial position to the overlay position.
118
                      // Since the overlayPosition is child of the ARCamera, once the trcking
119
                      // starts again the ARCamera position is updated relative to the target position
120
                      augmentedObject.transform.position = transform.position;
                 }
                  // Updates state variables
124
                  mIsShowingOverlay = false;
                  mIsTracking = true;
126
              #endregion //PUBLIC_METHODS
128
        ≙}
129
```

Σχήμα 9.74 Κώδικας αρχείου AnimationsManager.cs

Το παραπάνω αρχείο κώδικα είναι ένα από τα σημαντικότερα για τη λειτουργία της εφαρμογής, καθώς καθορίζει πότε το πάνελ πληροφοριών εμφανίζεται σε δυσδιάστατη μορφή και πότε σε τρισδιάστατη μορφή. Επίσης, εντός της συνάρτησης "Update" ρυθμίζεται και η θέση του πάνελ σε πραγματικό χρόνο ανά καρέ σκηνής.

ScanLine: Πρόκειται για ένα "Quad" Game Object. Το quad είναι ένα από τα έτοιμα Game Objects της πλατφόρμας Unity Engine και ένας από τους τρόπους για την αναπαράσταση μιας επίπεδης επιφάνειας έχοντας μήκος μία μονάδα σκηνής ανά πλευρά και ανήκει στην κατηγορία "3D Objects". Στο quad έχει δοθεί "material" (καθορίζει την εμφάνιση σε Game Object) μια διαφανή σφαίρα με μια άσπρη γραμμή στη μέση, αποκτώντας με αυτόν τον τρόπο εμφάνιση όμοια με αυτή του παρακάτω σχήματος.



Σχήμα 9.75 Καρτέλα σκηνής "ScanLine" Game Object

Στο "ScanLine" Game Object υπάρχει μόνο ένα component τύπου script με όνομα "CloudRecoScanLine.cs". Ο κώδικας του script αρχείου εμφανίζεται παρακάτω:
C# CloudRecoScanLine.cs ×

```
1
       using UnityEngine;
       ousing Vuforia;
 3
       public class CloudRecoScanLine : MonoBehaviour
 4
         {
             #region PRIVATE_MEMBERS
 6
 7
             const float SCAN_DURATION = 4; //seconds
8
9
             float mTime;
10
             bool mMovingDown = true;
             bool cachedEnabledState;
             Camera m_Camera;
             Renderer m_Renderer;
14
             CloudRecoBehaviour m_CloudRecoBehaviour;
15
16
             bool CloudEnabled
             {
                 get { return m_CloudRecoBehaviour && m_CloudRecoBehaviour.CloudRecoEnabled; }
18
19
             }
20
             #endregion //PRIVATE_MEMBERS
             #region MONOBEHAVIOUR_METHODS
             void Start()
             {
26
                 m_Camera = Camera.main;
                 m_Renderer = GetComponent<Renderer>();
28
                 m_CloudRecoBehaviour = FindObjectOfType<CloudRecoBehaviour>();
29
30
                 // Cache the Cloud enable state so that we can reset the scanline
                 // when the enabled state changes
                 cachedEnabledState = CloudEnabled;
             }
34
             void Update()
36
             {
                 if (cachedEnabledState != CloudEnabled)
38
                 {
39
                     cachedEnabledState = CloudEnabled;
40
                     // Reset the ScanLine position when Cloud enabled state changes
41
                     mTime = 0;
42
                     mMovingDown = true;
43
                 }
44
                 m_Renderer.enabled = CloudEnabled; // show/hide scanline
45
47
                 if (CloudEnabled)
48
                 ł
                     float u = mTime / SCAN_DURATION;
49
                     mTime += Time.deltaTime;
50
                     if (u > 1)
51
52
                     {
53
                         // invert direction
54
                         mMovingDown = !mMovingDown;
55
                         u = 0;
56
                         mTime = 0;
57
```

```
58
                      // Get the main camera
                      float viewAspect = m_Camera.pixelWidth / (float)m_Camera.pixelHeight;
                      float fovY = Mathf.Deg2Rad * m_Camera.fieldOfView;
61
62
                      float depth = 1.02f * m_Camera.nearClipPlane;
                      float viewHeight = 2 * depth * Mathf.Tan(0.5f * fovY);
                      float viewWidth = viewHeight * viewAspect;
64
65
                      // Position the mesh
67
                      float y = -0.5f * viewHeight + u * viewHeight;
                      if (mMovingDown)
                      {
70
                          y *= −1;
                      }
73
                      transform.localPosition = new Vector3(0, y, depth);
74
                      // Scale the quad mesh to fill the camera view
76
                      float scaleX = 1.02f * viewWidth;
                      float scaleY = scaleX / 32;
77
                      transform.localScale = new Vector3(scaleX, scaleY, 1.0f);
78
79
                 }
80
             }
81
             #endregion // MONOBEHAVIOUR_METHODS
82
83
       _<u>_</u>}
```

Σχήμα 9.76 Κώδικας αρχείου CloudRecoScanLine.cs

Ο παραπάνω κώδικας αναλαμβάνει την κίνηση της γραμμής σάρωσης κατά την εκτέλεση της εφαρμογής. Επειδή η κίνηση είναι συνεχής, όλος ο κώδικας έχει δημιουργηθεί εντός της συνάρτησης "Update", προκειμένου να ανανεώνεται σε κάθε καρέ.

CloudRecognition: Πρόκειται για Game Object της βιβλιοθήκης Vuforia. Ανήκει στην κατηγορία "Cloud Recognition" και κατά τη δημιουργία του έχει ως component το script με όνομα Cloud Reco Behavior". Με αυτό πραγματοποιείται η σύνδεση της εφαρμογής με την Vuforia Cloud Database. Το συγκεκριμένο script παίρνει ως παραμέτρους τα "Client Access Keys" της βάσης που δημιουργήθηκε σε προηγουμένη υποενότητα (Bλ. σχήμα «Κλειδιά πρόσβασης σε Vuforia Cloud Database»). Επιπλέον, στο συγκεκριμένο Game Object προστέθηκαν δύο ακόμα components τύπου script: "CloudRecoEventHandler.cs" και "CloudErrorHandler.cs". Οι ιδιότητες του Game Object μαζί με τα πεδία εισαγωγής των κλειδιών για τη βάση δεδομένων, φαίνονται στην καρτέλα "Inspector".

🚯 Inspector 🛛 🔀 Navigation 🛛 🖬 Package Manager	🖀 Asset Store				9	:
CloudRecognition				Static 🔻		
Tag Untagged	▼ Lay	/er Default				
🔻 🙏 Transform				0		
Position	X 0	Y O	ZO			
Rotation	X 0	Y 0	ZO			
Scale	X 1					
🔻 👩 🗹 Cloud Reco Behaviour (Script)				0		
Upgrade	to Vuforia Engine 9.6.3 v	ia Package Manager				
The Access and Secret Keys are used to authenticate your app with the Cloud Reco service, and to identify which Cloud database is queried. Do not share your keys with untrusted 3rd parties and take appropriate steps to protect them within your application code. If you based your app on the Cloud Reco sample, be sure to replace the sample keys with your own.						
Access Key	cess Key 23563c92118f90eab4fada3c8cd26754eacfd6cc					
Secret Key	bb4fd15b0ff20655	54724873a47e3027ca9664	4de			
🔻 📕 🖌 Cloud Reco Event Handler (Script)				0		
Script	# CloudRecoEver					
Image Target Behaviour	CloudRecoTarg	et (Image Target Behaviour)				
Cloud Activity Icon	In Cloud_Activity (Image)				
Cloud Idle Icon	⊠Cloud_Idle (Imag	ge)				•
🔻 # 🗹 Cloud Error Handler (Script)				0		
Script	# CloudErrorHand					
	Add Componen	t				

Σχήμα 9.77 Καρτέλα Inspector του "CloudRecognition" Game Object

Ο κώδικας των δύο προστιθέμενων script αρχείων εμφανίζεται και επεξηγείται παρακάτω:

C# CloudRecoEventHandler.cs \times

1	US	ing UnityEngine;
2	⊖us	ing Vuforia;
3		
4	B//	/ <summary></summary>
5	11	/ This ManaRehaviour implements the Cloud Reco Event handling for this sample
6	11	/ It registers itself at the CloudRecoRebaviour and is notified of new search results as well as error messages
7		/ The support state is visualized and new pecults are enabled using the TangetEinden APT
,		I aloummanus
0		
9	epu	DLIC CLASS CLOUDKECOEVENTHANDLEF : MONOBENAVIOUR
10	1	
11		#region PRIVALE_MEMBERS
12		CloudRecoBehaviour m_CloudRecoBehaviour;
13		ObjectTracker m_ObjectTracker;
14		TargetFinder m_TargetFinder;
15	P	#endregion // PRIVATE_MEMBERS
16		
17		
18		<pre>#region PUBLIC_MEMBERS</pre>
19	Ð	/// <summary></summary>
20		/// Can be set in the Unity inspector to reference a ImageTargetBehaviour
21		/// that is used for augmentations of new cloud reco results.
22		///
23		[Tooltip("Here you can set the ImageTargetBehaviour from the scene that will be used to " +
24	Ă	"augment new cloud reco search results.")]
25		public ImageTargetBehaviour m ImageTargetBehaviour:
26		public UnityEngine UT.Image m CloudActivityIcon:
20		public UnityEngine UI Image m_CloudIdleIcon;
27		#endnergion // DURLTC MEMBERS
20		"endregion // roberte_nenders
29		
30		
51	F	#region munubehaviouk_meihous
52		/// <summary></summary>
33		/// Register for events at the cloudkecobenaviour
34	θ	///
35		void Start()
36		
37		// Register this event handler at the CloudRecoBehaviour
38		m_CloudRecoBehaviour = GetComponent <cloudrecobehaviour>();</cloudrecobehaviour>
39		if (m_CloudRecoBehaviour)
40		{
41		<pre>m_CloudRecoBehaviour.RegisterOnInitializedEventHandler(OnInitialized);</pre>
42		<pre>m_CloudRecoBehaviour.RegisterOnNewSearchResultEventHandler(OnNewSearchResult);</pre>
43		<pre>m_CloudRecoBehaviour.RegisterOnStateChangedEventHandler(OnStateChanged);</pre>
44	6	}
45		
46		<pre>if (m_CloudActivityIcon)</pre>
47		{
48		<pre>m_CloudActivityIcon.enabled = false;</pre>
49	A	}
50	Ā	}
51		
01		

52	Þ	void Update()
53		{
54	e e	<pre>if (m_CloudRecoBehaviour.CloudRecoInitialized && m_TargetFinder != null)</pre>
55		{
56		<pre>SetCloudActivityIconVisible(m_TargetFinder.IsRequesting());</pre>
57	þ	}
58		
59	e e	if (m_CloudIdleIcon)
60		{
61		<pre>m_CloudIdleIcon.color = m_CloudRecoBehaviour.CloudRecoEnabled ? Color.white : Color.gray;</pre>
62	4	}
63	þ	}
64	4	#endregion // MONOBEHAVIOUR_METHODS
65		
66		
67	Ð	<pre>#region INTERFACE_IMPLEMENTATION_ICloudRecoEventHandler</pre>
68		
69	Ð	/// <summary></summary>
70		/// called when TargetFinder has been initialized successfully
71	4	///
72	P	<pre>public void OnInitialized(TargetFinder targetFinder)</pre>
73		{
74		<pre>Debug.Log("Cloud Reco initialized successfully.");</pre>
75		
76		m_ObjectTracker = TrackerManager.Instance.GetTracker <objecttracker>();</objecttracker>
77		m_TargetFinder = targetFinder;
78	þ	}
79		
80	P	/// <summary></summary>
81		/// when we start scanning, unregister Trackable from the ImageTargetBehaviour,
82		/// then delete all trackables
83	þ	///
84	P	<pre>public void OnStateChanged(bool scanning)</pre>
85		{
86		<pre>Debug.Log("<color=blue>OnStateChanged(): " + scanning);</color=blue></pre>
87		
88	Þ	<pre>// Changing CloudRecoBehaviour.CloudRecoEnabled to false will call:</pre>
89		<pre>// 1. TargetFinder.Stop()</pre>
90		<pre>// 2. All registered ICloudRecoEventHandler.OnStateChanged() with false.</pre>
91		
92		<pre>// Changing CloudRecoBehaviour.CloudRecoEnabled to true will call:</pre>
93		// 1. TargetFinder.StartRecognition()
94	Ĥ	<pre>// 2. All registered ICloudRecoEventHandler.OnStateChanged() with true.</pre>
95	Ĥ	}
96		

```
/// <summary>
97
              /// Handles new search results
              /// </summary>
100
              /// <param name="targetSearchResult"></param>
101
              public void OnNewSearchResult(TargetFinder.TargetSearchResult targetSearchResult)
102
              ł
                  Debug.Log("<color=blue>OnNewSearchResult(): </color>" + targetSearchResult.TargetName);
103
104
                  TargetFinder.CloudRecoSearchResult cloudRecoResult = (TargetFinder.CloudRecoSearchResult)targetSearchResult;
106
107
                 // This code demonstrates how to reuse an ImageTargetBehaviour for new search results
108
                 // and modifying it according to the metadata. Depending on your application, it can
109
                 // make more sense to duplicate the ImageTargetBehaviour using Instantiate() or to
                  // create a new ImageTargetBehaviour for each new result. Vuforia will return a new
                 // object with the right script automatically if you use:
                 // TargetFinder.EnableTracking(TargetSearchResult result, string gameObjectName)
                 // Check if the metadata isn't null
                  if (cloudRecoResult.MetaData == null)
                  {
                      Debug.Log("Target metadata not available.");
118
                 }
                 else
                  {
                      Debug.Log("MetaData: " + cloudRecoResult.MetaData);
                      Debug.Log("TargetName: " + cloudRecoResult.TargetName);
                      Debug.Log("Pointer: " + cloudRecoResult.TargetSearchResultPtr);
                      Debug.Log("TrackingRating: " + cloudRecoResult.TrackingRating);
                      Debug.Log("UniqueTargetId: " + cloudRecoResult.UniqueTargetId);
128
                 // Changing CloudRecoBehaviour.CloudRecoEnabled to false will call TargetFinder.Stop()
                  // and also call all registered ICloudRecoEventHandler.OnStateChanged() with false.
130
                 m_CloudRecoBehaviour.CloudRecoEnabled = false;
                 // Clear any existing trackables
                 m_TargetFinder.ClearTrackables(false);
                  // Enable the new result with the same ImageTargetBehaviour:
                 m_TargetFinder.EnableTracking(cloudRecoResult, m_ImageTargetBehaviour.gameObject);
138
                 // Pass the TargetSearchResult to the Trackable Event Handler for processing
                 m_ImageTargetBehaviour.gameObject.SendMessage("TargetCreated", cloudRecoResult,
                      SendMessageOptions.DontRequireReceiver);
              #endregion // INTERFACE_IMPLEMENTATION_ICloudRecoEventHandler
              #region PRIVATE_METHODS
              void SetCloudActivityIconVisible(bool visible)
              {
                  if (!m_CloudActivityIcon) return;
                  m_CloudActivityIcon.enabled = visible;
              #endregion // PRIVATE_METHODS
         5}
```

Σχήμα 9.78 Κώδικας αρχείου CloudRecoEventHandler.cs

Ο παραπάνω κώδικας αναλαμβάνει να αναγνωρίζει τη χρονική στιγμή κατά την οποία η εφαρμογή επικοινωνεί με τον κεντρικό διακομιστή της Vuforia και πραγματοποιείται μια

επιτυχημένη αναζήτηση στη βάση δεδομένων. Στη συνέχεια ενημερώνει το εικονίδιο του σύννεφου στην εφαρμογή ανάλογα την περίπτωση.

C# CloudErrorHandler.cs \times

```
/*-----
        Copyright (c) 2018 PTC Inc. All Rights Reserved.
 4
        Vuforia is a trademark of PTC Inc., registered in the United States and other
        countries.
        -----*/
 6
        using UnityEngine;
 8
        using Vuforia;
 9
        public class CloudErrorHandler : MonoBehaviour
10
        {
            #region PRIVATE_MEMBERS
            bool mustRestartApp;
            string errorTitle;
            string errorMsg;
            CloudRecoBehaviour m_CloudRecoBehaviour;
18
            #endregion PRIVATE_MEMBERS
19
20
            #region MONOBEHAVIOUR_METHODS
            void Start()
            {
               // Register this event handler with CloudRecoBehaviour
               m_CloudRecoBehaviour = FindObjectOfType<CloudRecoBehaviour>();
               if (m_CloudRecoBehaviour)
28
               {
                   m_CloudRecoBehaviour.RegisterOnInitErrorEventHandler(OnInitError);
29
                   m_CloudRecoBehaviour.RegisterOnUpdateErrorEventHandler(OnUpdateError);
30
               }
               if (VuforiaConfiguration.Instance.Vuforia.LicenseKey == string.Empty && m_CloudRecoBehaviour)
                {
                   errorTitle = "Cloud Reco Init Error";
                   errorMsg = "Vuforia License Key not found. Cloud Reco requires a valid license.";
38
                   MessageBox.DisplayMessageBox(errorTitle, errorMsg, false, null);
                }
40
            }
            void OnDestroy()
            {
44
                if (m_CloudRecoBehaviour)
                {
46
                   m_CloudRecoBehaviour.UnregisterOnInitErrorEventHandler(OnInitError);
47
                   m_CloudRecoBehaviour.UnregisterOnUpdateErrorEventHandler(OnUpdateError);
48
                }
49
            }
50
51
            #endregion MONOBEHAVIOUR_METHODS
52
53
```

```
#region INTERFACE_IMPLEMENTATION_ICloudRecoEventHandler
55
56
             /// <summary>
             /// Called if Cloud Reco initialization fails
58
             /// </summary>
59
             public void OnInitError(TargetFinder.InitState initError)
60
              {
                  switch (initError)
                  ł
                      case TargetFinder.InitState.INIT_ERROR_NO_NETWORK_CONNECTION:
                         mustRestartApp = true;
                         errorTitle = "Network Unavailable";
                         errorMsg = "Please check your Internet connection and try again.";
67
                         break;
68
                      case TargetFinder.InitState.INIT_ERROR_SERVICE_NOT_AVAILABLE:
69
                         errorTitle = "Service Unavailable";
70
                         errorMsg = "Failed to initialize app because the service is not available.";
                         break:
                  }
                  // Prepend the error code in red
                  errorMsg = "<color=red>" + initError.ToString().Replace("_", " ") + "</color>\n\n" + errorMsg;
76
                 // Remove rich text tags for console logging
                 var errorTextConsole = errorMsg.Replace("<color=red>", "").Replace("</color>", "");
78
79
80
                 Debug.LogError("OnInitError() - Initialization Error: " + initError + "\n\n" + errorTextConsole);
81
82
                 MessageBox.DisplayMessageBox(errorTitle, errorMsg, true, CloseDialog);
83
             }
84
85
             /// <summary>
             /// Called if a Cloud Reco update error occurs
86
87
             /// </summary>
             public void OnUpdateError(TargetFinder.UpdateState updateError)
88
89
             {
90
                  switch (updateError)
91
                  {
92
                      case TargetFinder.UpdateState.UPDATE_ERROR_AUTHORIZATION_FAILED:
93
                         errorTitle = "Authorization Error";
                         errorMsg = "The cloud recognition service access keys are incorrect or have expired.";
94
95
                         break;
                      case TargetFinder.UpdateState.UPDATE_ERROR_NO_NETWORK_CONNECTION:
96
                         errorTitle = "Network Unavailable";
97
98
                         errorMsg = "Please check your Internet connection and try again.";
99
                         break;
                      case TargetFinder.UpdateState.UPDATE_ERROR_PROJECT_SUSPENDED:
101
                         errorTitle = "Authorization Error";
102
                         errorMsg = "The cloud recognition service has been suspended.";
103
                         break;
                      case TargetFinder.UpdateState.UPDATE_ERROR_REQUEST_TIMEOUT:
104
105
                         errorTitle = "Request Timeout";
                         errorMsg = "The network request has timed out, please check your Internet connection and try again.";
106
107
                          break;
```

```
108
                      case TargetFinder.UpdateState.UPDATE_ERROR_SERVICE_NOT_AVAILABLE:
                          errorTitle = "Service Unavailable";
109
                          errorMsg = "The service is unavailable, please try again later.";
110
                          break:
                      case TargetFinder.UpdateState.UPDATE_ERROR_TIMESTAMP_OUT_OF_RANGE:
                          errorTitle = "Clock Sync Error";
114
                          errorMsg = "Please update the date and time and try again.";
                          break;
116
                      case TargetFinder.UpdateState.UPDATE_ERROR_UPDATE_SDK:
                          errorTitle = "Unsupported Version";
118
                          errorMsg = "The application is using an unsupported version of Vuforia.";
                          break:
120
                      case TargetFinder.UpdateState.UPDATE_ERROR_BAD_FRAME_QUALITY:
                          errorTitle = "Bad Frame Ouality":
                          errorMsg = "Low-frame quality has been continuously observed.\n\nError Event Received on Frame: "
                                     + Time.frameCount;
                          break;
                  3
126
                  // Prepend the error code in red
                  errorMsg = "<color=red>" + updateError.ToString().Replace("_", " ") + "</color>\n\n" + errorMsg;
128
130
                  // Remove rich text tags for console logging
                  var errorTextConsole = errorMsg.Replace("<color=red>", "").Replace("</color>", "");
                  Debug.LogError("OnUpdateError() - Update Error: " + updateError + "\n\n" + errorTextConsole);
135
                  MessageBox.DisplayMessageBox(errorTitle, errorMsg, true, CloseDialog);
138
              #endregion INTERFACE_IMPLEMENTATION_ICloudRecoEventHandler
139
140
              #region PUBLIC_METHODS
              public void CloseDialog()
              {
                  if (mustRestartApp) RestartApplication();
              3
148
              #endregion PUBLIC_METHODS
149
              #region PRIVATE_METHODS
              // Callback for network-not-available error message
              void RestartApplication()
              ł
                  int startLevel = UnityEngine.SceneManagement.SceneManager.GetActiveScene().buildIndex - 2;
                  if (startLevel < 0) startLevel = 0;</pre>
                  UnityEngine.SceneManagement.SceneManager.LoadScene(startLevel);
              #endregion PRIVATE_METHODS
         b}
```

Σχήμα 9.79 Κώδικας αρχείου CloudErrorHandler.cs

Ο παραπάνω κώδικας αναλαμβάνει τη διαχείριση όλων των πιθανών σφαλμάτων που μπορεί να προκύψουν κατά τη διάρκεια της επικοινωνίας μεταξύ της εφαρμογής και του διακομιστή της Vuforia, όπως για παράδειγμα λανθασμένα κλειδιά πρόσβασης για τη βάση, διακοπή σύνδεσης στο Διαδίκτυο κ.α. Για κάθε πιθανό σφάλμα παράγει και εμφανίζει στο χρήστη το αντίστοιχο μήνυμα σφάλματος.

Canvas (παιδί του CloudRecognition): Πρόκειται για ένα Game Object, τύπου καμβά που απλώς περικλείει το "RootFrame" Game Object.

RootFrame: Game Object που έχει την ιδιότητα του "Rect Transform". Πρόκειται, δηλαδή, για ένα ορθογώνιο, το οποίο ορίζει μια περιοχή εντός της οποίας τοποθετούνται αντικείμενα διεπαφής χρήστη (UI). Το συγκεκριμένο Game Object είναι πατέρας του "Cloud_Activity" και "Cloud_Idle", δύο Game Objects τύπου εικόνας. Εμπεριέχουν τις αντίστοιχες εικόνες προκειμένου να ενημερώνουν το χρήστη τη χρονική στιγμή που γίνεται επικοινωνία με το διακομιστή ή διαφορετικά κάθε πότε γίνεται χρήση δικτύου (για την επικοινωνία). Οι εικόνες των αντικειμένων "Cloud_Activity" και "Cloud_Idle" φαίνονται παρακάτω:



Εικόνα του Cloud_Activity



Εικόνα του Cloud_Idle

Σχήμα 9.80 Cloud icon

CloudRecoTarget: Πρόκειται για το βασικότερο Game Object της σκηνής. Αυτό είναι το αντικείμενο που προβάλλεται σε κάθε επιτυχημένη αναγνώριση στόχου. Ανήκει στη βιβλιοθήκη αντικειμένων της Vuforia Engine.



Σχήμα 9.81 Cloud Recognition GameObject

Κατά τη δημιουργία του, αυτό το Game Object παρουσιάζεται έχοντας ως component το script "DefaultTrackableEventHandler.cs". Το αρχικό αυτό script έχει διαγραφεί και στη θέση του έχει δημιουργηθεί το "CloudTrackableEventHandler.cs", ο κώδικας του οποίου παρουσιάζεται παρακάτω:

```
C# CloudTrackableEventHandler.cs
       ousing UnityEngine;
       dusing Vuforia;
       public class CloudTrackableEventHandler : DefaultTrackableEventHandler
        {
             #region PRIVATE_MEMBERS
            CloudRecoBehaviour m_CloudRecoBehaviour;
8
            CloudContentManager m_CloudContentManager;
            AnimationsManager m_AnimationsManager;
10
             bool m_isAugmentationVisible;
            #endregion // PRIVATE_MEMBERS
             #region MONOBEHAVIOUR_METHODS
             protected override void Start()
                base.Start();
18
                m_CloudRecoBehaviour = FindObjectOfType<CloudRecoBehaviour>();
20
                m_CloudContentManager = FindObjectOfType<CloudContentManager>();
                m_AnimationsManager = FindObjectOfType<AnimationsManager>();
                // Hide the Canvas Augmentation
                base.OnTrackingLost();
             }
             #endregion // MONOBEHAVIOUR_METHODS
             #region BUTTON_METHODS
            public void OnReset()
30
                Debug.Log("<color=blue>OnReset()</color>");
                // Changing CloudRecoBehaviour.CloudRecoEnabled to true will call TargetFinder.StartRecognition()
                // and also call all registered ICloudRecoEventHandler.OnStateChanged() with true.
                m_CloudRecoBehaviour.CloudRecoEnabled = true;
                m_isAugmentationVisible = false;
38
                // Hide the Canvas Augmentation
40
                base.OnTrackingLost();
                 TrackerManager.Instance.GetTracker<ObjectTracker>().GetTargetFinder<ImageTargetFinder>()
                     .ClearTrackables(false);
             }
             #endregion BUTTON_METHODS
48
             #region PUBLIC_METHODS
             /// <summaru>
50
             /// Method called from the CloudRecoEventHandler when a new target is created
             /// </summary>
             public void TargetCreated(TargetFinder.CloudRecoSearchResult targetSearchResult)
                 m_AnimationsManager.SetInitialAnimationFlags();
                 \verb|m_CloudContentManager.HandleMetadata(targetSearchResult.MetaData, targetSearchResult.TargetName); \\
             #endregion // PUBLIC_METHODS
58
```



Σχήμα 9.82 Κώδικας αρχείου CloudErrorHandler.cs

Κατά τη δημιουργία νέου στόχου, εκτελείται η συνάρτηση "TargetCreated", η οποία θέτει τις σωστές παραμέτρους, ώστε να ξεκινήσει να προβάλλεται το πάνελ επαυξημένης πραγματικότητας με την κλήση της συνάρτησης "SetInitialAnimationFlags". Αυτή με τη σειρά της, καλεί τη συνάρτηση "HandleMetadata", η οποία ξεκινά τη διαχείριση των μεταδεδομένων του στόχου. Επιπλέον, σε αυτό το αρχείο ορίζονται οι συναρτήσεις "OnTrackingFound" και "OnTrackingLost", οι οποίες κατά την ύπαρξη του στόχου στο οπτικό πεδίο της κάμερας ορίζουν τρισδιάστατη προβολή για το πάνελ πληροφοριών, ενώ σε αντίθετη περίπτωση ορίζουν δυσδιάστατη προβολή.

CloudContentManager: Πρόκειται για το Game Object που διαχειρίζεται τα μεταδεδομένα (metadata) κάθε στόχου μέχρι να προβληθούν στο πάνελ. Διαθέτει μόνο ένα component τύπου script με όνομα "CloudContentManager.cs", το οποίο δέχεται ως ορίσματα άλλα Game Objects τύπου κειμένου και τους αρχικοποιεί τις τιμές τους σύμφωνα με εκείνες που θα λάβει από τα μεταδεδομένα κάθε στόχου. Για την καλύτερη κατανόηση, η καρτέλα Inspector του CloudContentManager Game Object παρουσιάζεται παρακάτω:

🚯 Inspector 🛛 🔀 Navigation 🖉 Package N	Aanager 🛛 🚔 Asset Store				a :
CloudContentManager				St	tatic 🔻
Tag Untagged		Layer Default			
■ I Transform				0	
	X O	N O	70	v	
Position	X U	YU	20		
Rotation	X 0	Y 0	Z 0		
Scale	X 172.9	Y 172.9	2 172.9		
🔻 📕 🗹 Cloud Content Manager (Script)				9	
Script	# CloudCont				
Loading Indicator	⊠Spinner (Im	nage)			\odot
Cover	囸Preview (In	nage)			\odot
Title	🖬 Title (Text)				\odot
Painter	🖬 Painter (Te	xt)			0
Painting Title	PaintingTit	le (Text)			0
Description	■ PaintingDe	scription (Text)			\odot
▼ 券 Event Trigger				0	
Pointer Click (BaseEventData)					
Runtime Only	CloudTrackableEventHandler	.OnReset			
😫 CloudRecoTarget (Cloud Trackable Eve 👁					
Runtime Only -	<missing cloudcontentmanag<="" td=""><td>ger.ClearBookData></td><td></td><td></td><td></td></missing>	ger.ClearBookData>			
CloudContentManager (Cloud Content ⊙					
				+	
	Add New Ev	ent Type			
	Add Comp	onent			

Σχήμα 9.83 Καρτέλα Inspector του "CloudContentManager" Game Object

C# CloudContentManager.cs \times

```
busing System.Collections;
 1
         using System.Globalization;
 3
         using UnityEngine;
         using UnityEngine.Networking;
 4
        ousing UnityEngine.UI;
 5
 6
 7
        public class CloudContentManager : MonoBehaviour
 8
         {
 9
             #region PROPERTIES
10
             string Painter { get; set; }
             string Title { get; set; }
             string Description { get; set; }
             string ImageUrl { get; set; }
14
             string BrowserURL { get; set; }
             #endregion // PROPERTIES
16
18
             #region PUBLIC_MEMBERS
19
20
             public Image m_LoadingIndicator;
             public Image m_Cover;
             public Text m_Title;
23
             public Text m_Painter;
             public Text m_PaintingTitle;
24
             public Text m_Description;
             #endregion PUBLIC_MEMBERS
28
29
             #region PRIVATE_MEMBERS
30
             bool webRequestInProgress;
             #endregion PRIVATE_MEMBERS
             #region MONOBEHAVIOUR_METHODS
             void Update()
36
             {
                 if (webRequestInProgress && m_LoadingIndicator)
38
                 {
                     m_LoadingIndicator.rectTransform.Rotate(Vector3.forward, 90.0f * Time.deltaTime);
39
40
                 }
             }
             #endregion // MONOBEHAVIOUR_METHODS
44
             #region PUBLIC_METHODS
             public void HandleMetadata(string metadata, string targetName)
             {
48
                 Debug.Log("<color=red>" + metadata + "</color>");
49
                 ClearPaintingData();
50
                 string[] metadataArray = metadata.Split('\n');
                 Painter = metadataArray[0];
                 Title = metadataArray[1];
                 Description = metadataArray[2];
                 ImageUrl = metadataArray[3];
                 BrowserURL = metadataArray[4];
56
```

```
57
                  StartCoroutine(WebRequest(ImageUrl));
58
                  UpdatePaintingTextData();
59
              }
60
61
              public void ClearPaintingData()
62
63
              {
64
                  m_Cover.sprite = null;
                  m_Cover.color = Color.black;
65
                  m_Title.text = "Art_recognition";
                  m_Painter.text = "Painter";
67
 68
                  m_PaintingTitle.text = "Title";
                  m_Description.text = "Description";
69
              }
70
              #endregion // PUBLIC_METHODS
71
72
74
              #region BUTTON_METHODS
              public void MoreInfoButton()
75
76
              {
77
                  Application.OpenURL(BrowserURL);
78
              }
              #endregion // BUTTON_METHODS
79
80
81
82
              #region PRIVATE_METHODS
              void UpdatePaintingTextData()
83
84
              {
85
                  Debug.Log("UpdatePaintingTextData() called.");
86
                  m_Title.text = "ARt recognition";
87
                  m_Painter.text = Painter;
88
89
                  m_PaintingTitle.text = Title;
90
                  m_Description.text = Description;
91
              }
92
              void ProcessWebRequest(UnityWebRequest webRequest)
93
              {
                  Debug.Log("ProcessWebRequest() called: \n" + webRequest.url);
94
                  var texture = DownloadHandlerTexture.GetContent(webRequest);
95
                  if (texture == null) return;
96
97
98
                  if (m_Cover)
99
                  {
                      m_Cover.sprite = Sprite.Create(texture, new Rect(0, 0, texture.width, texture.height),
100
                          new Vector2(0.5f, 0.5f));
101
102
                      m_Cover.color = Color.white;
103
                      webRequest.Dispose();
104
                  }
105
              }
106
```

```
107
              IEnumerator WebRequest(string url)
108
              {
109
                  Debug.Log("WebRequest() called: \n" + url);
110
                  webRequestInProgress = true;
                  m_LoadingIndicator.enabled = true;
                  if (string.IsNullOrEmpty(url))
114
                  {
                      Debug.LogError("WebRequest() failed. Your URL is null or empty.");
                      yield break;
118
                  }
119
                  UnityWebRequest webRequest;
120
                  if (url.Contains(".jpg") || url.Contains(".png"))
                  {
                      webRequest = UnityWebRequestTexture.GetTexture(url);
                  }
                  else
                  {
                      Debug.LogError("WebRequest() failed. The URL has to point to a .jpj or .png file.");
128
                      yield break;
                  }
129
130
                  yield return webRequest.SendWebRequest();
                  if (webRequest.isDone)
                  {
                      Debug.Log("Done Loading: \n" + webRequest.url);
                      webRequestInProgress = false;
                      m_LoadingIndicator.enabled = false;
138
                  }
139
140
                  if (string.IsNullOrEmpty(webRequest.error))
                  {
                      // If error string is null or empty, then request was successful
                      ProcessWebRequest(webRequest);
                  }
                  else
                  {
                      Debug.LogError("Error With WWW Request: " + webRequest.error);
148
149
                      string error = "<color=red>" + webRequest.error + "</color>" + "\nURL Requested: " + webRequest.url;
150
                      MessageBox.DisplayMessageBox( title: "WWW Request Error", body: error, dismissable: true, dismissAction: null);
                  }
              #endregion //PRIVATE_METHODS
         6}
```

Σχήμα 9.84 Κώδικας αρχείου CloudContentManager.cs

Με την αναγνώριση στόχου καλείται η συνάρτηση "HandleMetadata", όπου παίρνει σαν όρισμα τα μεταδεδομένα και το όνομα του στόχου. Με την εκτέλεση της συνάρτησης διαχωρίζονται τα μεταδεδομένα ανά κατηγορία (τίτλος πίνακα, όνομα ζωγράφου, περιγραφή πίνακα και σύνδεσμος εικόνας), ενώ οι τιμές τους τίθενται στα αντίστοιχα Game Objects προκειμένου να ολοκληρωθεί η προβολή τους στο πάνελ πληροφοριών. Τα Game Objects στα οποία αντιστοιχίζονται οι τιμές των μεταδιδόμενων έχουν οριστεί εκ των προτέρων, όπως φαίνονται στην καρτέλα Inspector. Τέλος, σε αυτό το αρχείο ορίζεται και η συνάρτηση "MoreInfoButton", η οποία εκτελείται κάθε φορά που θα πατηθεί το κουμπί «Περισσότερες πληροφορίες» από το πάνελ πληροφοριών, προκειμένου οι χρήστες να μετάβουν στην ιστοσελίδα του πίνακα.

Canvas (παιδί του CloudContentManager): Πρόκειται για ένα Game Object τύπου καμβά το οποίο συνθέτει το πάνελ πληροφοριών. Σε αυτό το αντικείμενο έχει εστιάσει το "ARCamera" Game Object.

Η δημιουργία του πάνελ πληροφοριών περιγράφεται συνοπτικά, καθώς απαρτίζεται από αρκετά αλλά καθόλου περίπλοκα Game Objects χωρίς πολλά components και παρουσία αρχείων κώδικα. Κατά κύριο λόγο, υπάρχουν game objects τύπου εικόνας, διεπαφής χρήστη (UI) και κειμένου. Τα αντικείμενα που συνθέτουν το πάνελ είναι όλα παιδιά του Canvas Game Object και η ιεραρχία τους εμφανίζεται παρακάτω εντός του κόκκινου πλαισίου.



Σχήμα 9.85 Ιεραρχία αντικειμένων σύνθεσης πάνελ

Η λειτουργία και ο σκοπός ύπαρξης των παραπάνω Game Objects, εξηγείται παρακάτω για το καθένα χωριστά:

- Root: Ορίζει το φόντο του πάνελ έχοντας ως component μια ημιδιάφανη μονόχρωμη μαύρη εικόνα.
- Panel: Ορίζει τον τρόπο εμφάνισης και την απόσταση που θα έχουν τα παιδιά αντικείμενα κατά την εμφάνιση τους. Κάτι τέτοιο στο Unity ονομάζεται "Vertical Layout Group".
- > Title: Game Object κειμένου, εκεί τοποθετείται ο γενικός τίτλος του πάνελ.
- Content: Ορίζει τον τρόπο εμφάνισης και την απόσταση που θα έχουν τα παιδιά αντικείμενα (Vertical Layout Group). Συγκεκριμένα, ορίζει το εσωτερικό κενό διάστημα που βρίσκεται στα αριστερά (padding left) κάθε παιδιού Game Object στα δέκα εικονοστοιχεία.
- Preview: Ορίζει το φόντο της εικόνας του πάνελ έχοντας ως μοναδικό component μια μονόχρωμη μαύρη εικόνα.
- > Spinner: Προσθέτει την εικόνα του spinner στο κέντρο του Preview Game Object.
- Details: Ορίζει τον τρόπο εμφάνισης και την απόσταση που θα έχουν τα παιδιά αντικείμενα (Vertical Layout Group).
- > Painter: Game Object κειμένου, εκεί τοποθετείται το όνομα του ζωγράφου.
- > PaintingTitle: Game Object κειμένου, εκεί τοποθετείται ο τίτλος του πίνακα.
- PaintingDescription: Game Object κειμένου, εκεί τοποθετείται η περιγραφή κάθε πίνακα.
- > **TopRightCorner:** Ορίζει την πάνω δεξιά περιοχή του πάνελ.

- CloseIcon: Προσθέτει το εικονίδιο για κλείσιμο του πάνελ στην περιοχή που ορίζεται από το TopRightCorner Game Object.
- Hyperlink: Κουμπί που πυροδοτεί τη συνάρτηση "MoreInfoButton" από το αρχείο "CloudContentManager.cs"
- Label: Game Object κειμένου με περιεχόμενο «Περισσότερες πληροφορίες»

Η σύνθεση όλων των παραπάνω Game Objects, οδηγούν στη δημιουργία του πάνελ πληροφοριών και το τελικό αποτέλεσμα απεικονίζεται στην καρτέλα σκηνής του Canvas Game Object.



Σχήμα 9.86 Καρτέλα σκηνής "Canvas" Game Object

CommonUI: Διαχειρίζεται τη μπάρα ενημέρωσης του χρήστη και είναι ο πατέρας όλων των Game Objects που την αποτελούν. Έχει ως components δύο script αρχεία: "NavigationHandler.cs" και "TapHandler.cs". Το αρχείο "NavigationHandler" αναλαμβάνει τη μετάβαση του χρήστη από την σκηνή "Paintings" στη σκηνή "About", όταν ο χρήστης πατήσει το πίσω κουμπί της κινητής του συσκευής ενώ το αρχείο "TapHandler" αναλαμβάνει να ξεχωρίσει το απλό πάτημα του χρήστη από το διπλό πάτημα στην οθόνη της κινητής συσκευής. Ο κώδικας των αρχείων παρουσιάζεται στις ακόλουθες φωτογραφίες:

```
C# TapHandler.cs \times
```

1	usin	g UnityEngine;
2	- nubl	ic class TanHandlen · MonoBohavioun
5	- hon r	
4		#pegion DETVATE MEMBERS
6		const float DOUBLE TAP MAY DELAY - $0.5f$ //seconds
7		float mTimeSinceLastTan:
0		MenuOntions m MenuOntions:
0		CameraSettings m CameraSettings:
7 1 M		<pre>#endregion //PRIVATE MEMBERS</pre>
11		
12		
13	_	#region PROTECTED MEMBERS
14	Y I	protected int mTanCount:
15		#endregion //PROTECTED MEMBERS
16		
17		
18		#region MONOBEHAVIOUR METHODS
19	Ľ.	void Start()
20	Ť	{
21		mTapCount = 0:
22		mTimeSinceLastTap = 0;
23		<pre>m MenuOptions = FindObjectOfTvpe<menuoptions>();</menuoptions></pre>
24		<pre>m_CameraSettings = FindObjectOfType<camerasettings>();</camerasettings></pre>
25		}
26		
27	L .	void Update()
28		{
29	e l	<pre>if (m_MenuOptions && m_MenuOptions.IsDisplayed)</pre>
30		{
31		mTapCount = 0;
32		<pre>mTimeSinceLastTap = 0;</pre>
33	þ	}
34	e e	else
35		{
36		HandleTap();
37	þ	}
38	P	}
39	P	<pre>#endregion //MONOBEHAVIOUR_METHODS</pre>
40		
41		
42	Þ	#region PRIVATE_METHODS
43	Þ	private void HandleTap()
44		
45	Ð	1+ (miapCount == 1)
46		
47		milmeSinceLastiap += ilme.deltailme;
48	Ę	<pre>if (miimesinceLastiap > DUUBLE_IAP_MAX_DELAY) </pre>
49		1
50	Ę	// too late for double tap,
51		// we confirm it was a single tap
52		UNSINGLETAPLONTITMEd();

```
53
54
                           // reset touch count and timer
                           mTapCount = 0;
                           mTimeSinceLastTap = 0;
56
58
                  }
59
                  else if (mTapCount == 2)
                  {
60
61
                      // we got a double tap
                      OnDoubleTap();
63
                      // reset touch count and timer
64
                      mTimeSinceLastTap = 0;
65
                      mTapCount = 0;
66
67
                  }
68
                  if (Input.GetMouseButtonUp(0))
69
70
                  {
                      mTapCount++;
                      if (mTapCount == 1)
73
                      {
74
                           OnSingleTap();
                      }
76
                  }
              }
              #endregion // PRIVATE_METHODS
78
79
80
81
              #region PROTECTED_METHODS
              /// <summary>
82
              /// This method can be overridden by custom (derived) TapHandler implementations,
83
              /// to perform special actions upon single tap.
              /// </summary>
85
              protected virtual void OnSingleTap() { }
86
87
              protected virtual void OnSingleTapConfirmed()
88
89
              {
                  if (m_CameraSettings)
90
91
                  {
                      m_CameraSettings.TriggerAutofocusEvent();
92
                  }
93
94
              }
95
              protected virtual void OnDoubleTap()
96
97
              {
                  if (m_MenuOptions && !m_MenuOptions.IsDisplayed)
98
99
                  {
                      m_MenuOptions.ShowOptionsMenu(true);
100
                  }
101
102
              }
              #endregion // PROTECTED_METHODS
103
104
        ₿}
```

Σχήμα 9.87 Κώδικας αρχείου TapHandler.cs

```
C# NavigationHandler.cs \times
```

```
1
        busing UnityEngine;
 2
        dusing UnityEngine.SceneManagement;
 3
 4
        bpublic class NavigationHandler : MonoBehaviour
         {
 5
             #region PUBLIC_MEMBERS
 6
             public string m_BackButtonNavigation = "[Name of Scene To Load]";
 7
             #endregion // PUBLIC_MEMBERS
 8
 9
10
11
             #region MONOBEHAVIOUR_METHODS
12
             void Update()
13
             {
                 // On Android, the Back button is mapped to the Esc key
14
15
                 if (Input.GetKeyUp(KeyCode.Escape))
16
                  {
                      HandleBackButtonPressed();
17
18
                  2
19
             #endregion // MONOBEHAVIOUR_METHODS
20
21
22
             #region PUBLIC_METHODS
23
             public void HandleBackButtonPressed()
24
25
             ł
                  Debug.Log("HandleBackButtonPressed() called.");
26
                  if (SceneManager.GetActiveScene().name != m_BackButtonNavigation)
27
28
                      LoadScene(m_BackButtonNavigation);
29
             2
             #endregion // PUBLIC_METHODS
30
31
32
33
             #region PRIVATE_METHODS
             void LoadScene(string sceneName)
34
35
             {
                  Debug.Log("LoadScene(" + sceneName + ") called.");
36
                  if (!string.IsNullOrEmpty(sceneName))
                  {
38
39
                      SceneManager.LoadScene(sceneName);
40
                  3
41
             }
             #endregion // PRIVATE_METHODS
42
        Δ}
43
```

Σχήμα 9.88 Κώδικας αρχείου NavigationHandler.cs

Η μπάρα ενημέρωσης του χρήστη, εκτός από το εικονίδιο του σύννεφου, το οποίο εισέρχεται από διαφορετικά Game Objects, αποτελείται από τα ακόλουθα Game Objects εντός του κόκκινου πλαισίου.



Σχήμα 9.89 Ιεραρχία αντικειμένων σύνθεσης μπάρας ενημέρωσης χρήστη

Κανένα από τα παραπάνω Game Objects δεν διαθέτει κάποια λογική ή κάποιο αρχείο κώδικα. Πρόκειται για εικόνες και κείμενα που συνθέτουν την μπάρα ενημέρωσης. Τα κυριότερα από αυτά είναι τα ακόλουθα:

- Canvas (παιδί του TitleBar): Ορίζει το φόντο του πάνελ έχοντας ως component μια ημιδιάφανη μονόχρωμη μαύρη εικόνα.
- Title: Game Object κειμένου με περιεχόμενο «Εθνική Πινακοθήκη».
- BackButton: Προσθέτει το εικονίδιο για την ενότητα «Βοήθεια», ενώ ταυτόχρονα λειτουργεί και ως κουμπί που πυροδοτεί τη συνάρτηση "HandleBackButtonPressed" από το αρχείο "NavigationHandler.cs".

Η σύνθεση όλων των παραπάνω Game Objects, οδηγούν στη δημιουργία της μπάρας ενημέρωσης και το τελικό αποτέλεσμα απεικονίζεται στην καρτέλα σκηνής του TitleBar Game Object.



Σχήμα 9.90 Καρτέλα σκηνής "TitleBar" Game Object

SafeAreaManager: Game Object, το οποίο έχει ακριβώς την ίδια λειτουργικότητα με εκείνο της σκηνής «About» (έχει περιγράφει παραπάνω).

Σε αυτό το σημείο έχει ολοκληρωθεί η δημιουργία όλων των απαιτούμενων σκηνών για την υλοποίηση της εφαρμογής, γεγονός που επιτρέπει τη δημιουργία εκτελέσιμων αρχείων.

9.4.5 Δημιουργία εκτελέσιμων αρχείων

Με την ολοκλήρωση των σκηνών που απαιτούνται για την υλοποίηση της εφαρμογής, ακολουθεί η δημιουργία εκτελέσιμων αρχείων ικανά να εγκαταστήσουν την εφαρμογή τόσο σε κινητά τηλεφωνά με λειτουργικό σύστημα "Android", όσο και σε συσκευές με λειτουργικό σύστημα "iOS". Ειδικότερα, πρόκειται για μια διαδικασία που ενσωματώνει την εφαρμογή σε πρόγραμμα αναπαραγωγής. Η διαδικασία αυτή ονομάζεται στο το Unity Engine ως "Build". Για την εκκίνηση αυτής της διαδικασίας, απαιτείται το άνοιγμα του παραθύρου με τις ρυθμίσεις του "Build", το οποίο ονομάζεται "Build Settings". Για την πρόσβαση στο παράθυρο "Build Settings", επιλέγεται από το κύριο μενού του Unity το "File" και στη συνέχεια το "Build Settings" (File → Build Settings).

Το παράθυρο "Build Settings" χρησιμοποιείται για την επιλογή της πλατφόρμας προορισμού και την προσαρμογή των ρυθμίσεων για την κατασκευή της εφαρμογής, προκειμένου να ξεκινήσει η διαδικασία κατασκευής εκτελέσιμων αρχείων (διαδικασία Build). Με αυτό τον τρόπο καθορίζονται οι σκηνές, η πλατφόρμα και οι ρυθμίσεις συσκευής σύμφωνα με τις οποίες θα εκτελεστεί η διαδικασία του Build. Μετά τη διαμόρφωση των ρυθμίσεων, υπάρχουν δύο επιλογές: "Build" και "Build and Run".

Η επιλογή "Build and Run" πραγματοποιεί αρχικά τη διαδικασία του "Build" και στη συνέχεια εκτελεί την εφαρμογή στον υπολογιστή, ενώ η επιλογή "Build" δημιουργεί τα αρχεία εγκατάστασης. Καθώς η εφαρμογή δεν απαιτείται να εκτελεστεί στον υπολογιστή, επιλέγεται το "Build".

Build Settings		×
Scenes In Build SamplesResources/Scenes/0-Splash SamplesResources/Scenes/1-About SamplesResources/Scenes/2-Loading SamplesResources/Scenes/3-Paintings		0 1 2 3
Dististant		Add Open Scenes
Platform PC, Mac & Linux Standalone	🛒 Android	
iOS ios	Texture Compression	Don't override 🗸 👻
📫 Android 📢	ETC2 fallback Export Project	32-bit ▼
Universal Windows Platform	Symlink Sources Build App Bundle (Google Play	
tvos tvos	Create symbols.zip	
гла PS4	Run Device Development Build	Android SDK and/or JDK directory is not set
🖄 Xbox One	Deep Profiling	
WebGL	Script Debugging Scripts Only Build Compression Method	Patch Patch And Run LZ4 ✓ Learn about Unity <u>Cloud Build</u>
Player Settings		Build Build And Run

Σχήμα 9.91 Unity Engine Build Settings

Για "Build" σε κινητά με λειτουργικό σύστημα Android, επιλέγεται ως πλατφόρμα το Android και στην συνέχεια το κουμπί Build, ενώ οι υπόλοιπες ρυθμίσεις μένουν στις προκαθορισμένες τιμές. Με την ολοκλήρωση του "Build" δημιουργείται ένα αρχείο .apk (αρχείο εγκατάστασης εφαρμογής). Η εφαρμογή είναι πλέον έτοιμη για εγκατάσταση σε συσκευές Android.

Σημείωση: Θα χρειαστεί η ενεργοποίηση εγκατάστασης εφαρμογών από άγνωστες πήγες στις ρυθμίσεις της συσκευής σε περίπτωση που δεν ανεβεί στο PlayStore.

Για "Build" σε κινητά με λειτουργικό σύστημα iOS ακολουθείται η ιδιά διαδικασία, με μόνη διαφορά την επιλογή επιλέγεται το iOS ως πλατφόρμα. Με την ολοκλήρωση του "Build" έχουν δημιουργηθεί όλα τα απαραίτητα αρχεία για την εγκατάσταση.

Σημείωση: Απαιτείται η λήψη του προγράμματος Xcode (περιβάλλον ανάπτυξης εφαρμογών της Apple), καθώς επίσης και δημιουργία λογαριασμού προγραμματιστή (Apple developer account).

10 ΕΠΙΛΟΓΟΣ

10.1 Υλικό και χρήσιμοι σύνδεσμοι

Με την ολοκλήρωση της εφαρμογής ARt, ακολουθεί συνοπτική παρουσίαση όλων των αρχείων που δημιουργήθηκαν κατά τα στάδια υλοποίησης, σε συνδυασμό με τα τελικά εκτελέσιμα για Android και iOS συσκευές. Οι παρακάτω σύνδεσμοι αντιστοιχούν σε συμπιεσμένα αρχεία (.zip) στο OneDrive (διαδικτυακός χώρος αποθήκευσης αρχείων της Microsoft).

```
Περιγραφή
```

Σύνδεσμος

Πρωτότυπο διεπαφής χρήστη στο Marvel <u>ARt – Marvel UI</u> Ερωτηματολόγιο διεπαφής χρήστης <u>ARt – Ερωτηματολόγιο</u> Συλλογή δεδομένων Εθνικής Πινακοθήκης ARt – Κώδικας Python * Δημιουργία Vuforia Cloud Database ARt – Κώδικας Java Ιστοσελίδα λήψης της εφαρμογής <u>ARt – Λήψη εφαρμογής</u> Αρχεία εγκατάστασης εφαρμογής <u>ARt – Android & ARt – iOS</u> Δοκιμαστικοί στόχοι ARt – Image targets Όροι χρήσης και προϋποθέσεις ARt - Όροι και προϋποθέσεις

Πίνακας 10.1 Χρήσιμοι σύνδεσμοι

Ο σύνδεσμος "ARt – Android & iOS" επιτρέπει τη λήψη της εφαρμογής για κινητές συσκευές. Μετά τη λήψη και εγκατάσταση της εφαρμογής, έχει δημιουργηθεί αρχείο με ορισμένους πίνακες της Εθνικής Πινακοθήκης (σύνδεσμος "ARt – Image targets"), το οποίο επιτρέπει τη δοκιμαστική λειτουργία της εφαρμογής σε πραγματικό χρόνο.

Επιτρέπεται η αποθήκευση και η διανομή του παραπάνω υλικού για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης.

10.2 Συμπεράσματα και μελλοντική επέκταση

Η εφαρμογή έχει χρησιμοποιηθεί και δοκιμαστεί μέχρι στιγμής μόνο σε ακαδημαϊκά πλαίσια, σημειώνοντας εξαιρετικά αποτελέσματα. Στόχος όμως, είναι η χρήση της από τους επισκέπτες της Εθνικής Πινακοθήκης. Για την πραγματοποίηση του στόχου απαιτείται η ύπαρξη κατάλληλων υποδομών εντός του νέου κτιρίου της Εθνικής Πινακοθήκης (δίκτυο υψηλής ταχύτητας MEC και διακομιστές), καθώς επίσης και έγκριση από τους αρμοδίους φορείς. Ένα τέτοιο εγχείρημα μπορεί να αποτελέσει το έναυσμα για την επέκταση της χρήσης τεχνολογιών επαυξημένης πραγματικότητας και σε άλλους παραπλήσιους τομείς.



Σχήμα 10.1 Σενάριο εικονικής χρήσης της εφαρμογής ARt στο νέο κτήριο της Εθνικής Πινακοθήκης

Μελλοντική επέκταση της εφαρμογής θα μπορούσε να αποτελέσει η προσθήκη οδηγιών όσον αφορά την τοποθεσία των εκθεμάτων στο χώρο και η πρόταση πιθανών διαδρομών για μια περισσότερο εξατομικευμένη ξενάγηση με χρήση τεχνολογιών επαυξημένης πραγματικότητας. Μια καινοτόμα μέθοδος προσέλκυσης και καθοδήγησης των επισκεπτών ικανή να επαναπροσδιορίσει την επίσκεψη σε χώρους μουσείων.



Σχήμα 10.2 Λήψη οδηγιών για την εύρεση εκθέματος με χρήση τεχνολογιών επαυζημένης πραγματικότητας

11 ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Unity (game engine) https://en.wikipedia.org/wiki/Unity_(game_engine) [ONLINE - Wikipedia]
- [2] Vuforia Augmented Reality SDK <u>https://en.wikipedia.org/wiki/Vuforia_Augmented_Reality_SDK</u> [ONLINE -Wikipedia]
- [3] Python https://el.wikipedia.org/wiki/Python [ONLINE - Wikipedia]
- [4] Java
 «Βελτιστοποίηση απόδοσης συστήματος προτάσεων βασισμένη σε τεχνικές εξόρυξης
 πληροφορίας»
 <u>http://artemis.cslab.ece.ntua.gr:8080/jspui/bitstream/123456789/17287/1/Andreas%20
 Christou%20Thesis.pdf</u> [Διπλωματική εργασία του Ανδρέα Χρήστου, σελίδες 37-38]
- [5] C Sharp https://el.wikipedia.org/wiki/C_Sharp [ONLINE - Wikipedia]
- [6] Εθνική Πινακοθήκη <u>www.nationalgallery.gr</u> [ONLINE]
- [7] Δήλωση για την έγκαιρη ολοκλήρωση του νέου κτηρίου της Εθνική Πινακοθήκης <u>https://ypodomes.com/stin-teliki-eytheia-to-ergo-tis-epektasis-tis-ethnikis-</u> <u>pinakothikis/amp</u> [ONLINE]
- [8] Βέλτιστες πρακτικές για το σχεδιασμό και την ανάπτυξη στόχων βάσει εικόνας <u>https://library.vuforia.com/features/images/image-targets/best-practices-for-</u> <u>designing-and-developing-image-based-targets.html</u> [ONLINE - Vuforia Official site]
- [9] Cloud Recognition
 <u>https://library.vuforia.com/articles/Training/Cloud-Recognition-Guide.html</u> [ONLINE
 - Vuforia Official site]
- [10] Adobe Photoshop <u>https://en.wikipedia.org/wiki/Adobe_Photoshop</u> [ONLINE - Wikipedia]
- [11] Διεπαφή προγραμματισμού εφαρμογών (API) <u>https://en.wikipedia.org/wiki/API</u> [ONLINE - Wikipedia]

- [12] Αρχιτεκτονική Rpresentational State Transfer REST <u>https://gkalmoukis.com/ti-einai-to-rest</u> [ONLINE]
- [13] Πρωτόκολλο επικοινωνίας HTTP <u>https://el.wikipedia.org/wiki/HTTPS</u> [ONLINE - Wikipedia]
- [14] Unity Manual https://docs.unity3d.com/Manual/index.html [Unity webpage]
- [15] Vuforia Web Services Samples https://developer.vuforia.com/downloads/samples [Vuforia Engine: developer portal]
- [16] Vuforia Books Sample https://assetstore.unity.com/packages/templates/packs/vuforia-books-sample-102254
- [17] Arif Ahmed, Ejaz Ahmed, "A Survey on Mobile Edge Computing", 2016
- [18] Dario Sabella, Alessandro Vaillant, Pekka Kuure, Uwe Rauschenbach, Fabio Giust, "Mobile Edge Computing Architecture - The role of MEC in the Internet of Things", 2016
- [19] ETSI GS MEC 012, Mobile Edge Computing (MEC); Radio Network Information API, V1.1.1, 2017-07
- [20] ETSI GS MEC 013, Mobile Edge Computing (MEC); Location API, V1.1.1, 2017-07
- [21] ETSI GS MEC 015, Mobile Edge Computing (MEC); Bandwidth Management API, V1.1.1 , 2017-10
- [22] ETSI White Paper NO. 11, Mobile Edge Computing A key kechnology towards 5G, First Edition- September 2015
- [23] ETSI White Paper No. 24, MEC Deployments in 4G and Evolution Towards 5G, First Edition - February 2018
- [24] Cisco, 5G Security Innovation with Cisco, Whitepaper, 2018
- [25] ETSI White Paper No. 28, MEC in 5G networks, First Edition June 2018
- [26] Qualcomm Technologies, VR and AR pushing connectivity limits, October 2018