ΑτομικήΔιπλωματικήΕργασία

Tour Guide With the Use of Augmented Reality

Χρίστος Βασιλείου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2013

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Tour Guide With the Use of Augmented Reality

Χρίστος Βασιλείου

Επιβλέπων Καθηγητής
Γιώργος Χρυσάνθου

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

Μάιος 2013

Ευχαριστίες

Θα ήθελα να ευχαριστήσω των επιβλέποντα καθηγητή μου κ. Γιώργο Χρυσάνθου για την πολύτιμη βοήθεια που μου παρείχε σ' όλη την διάρκεια της διπλωματικής μου εργασίας. Επίσης, θα ήθελα να τον ευχαριστήσω για τις πολύτιμες συμβουλές που μου έδινε.

Περίληψη

Το θέμα της Ατομικής Διπλωματικής μου Εργασίας είναι η ανάπτυξη μιας εφαρμογής για Android συσκευές όπου θα δίνει πληροφορίεςστους τουρίστες κατά την ξενάγησή τους σε αρχαιολογικούς χώρους. Η εφαρμογή αυτή αποτελείται από δύο μέρη, το χάρτη και το AugmentedReality.

Το πρώτο μέρος αποτελείται από ένα χάρτη. Σε αυτό το χάρτη βρίσκονται με σημεία όλα τα αξιοθέατα που μπορεί να δει ένας τουρίστας. Αν πατήσει πάνω σε ένα από αυτά τα σημεία τότε βλέπει το όνομα του αξιοθέατου με μια σύντομη περιγραφή του τι είναι καθώς και την επιλογή να δει περισσότερες πληροφορίες. Μπορεί επιπλέων να δει πια διαδρομή μπορεί να πάρει για να πάει από το ένα σημείο στο άλλο.

Το δεύτερο μέρος αποτελείται από το AugmentedReality. Εδώ, μέσα από την κάμερα της συσκευής του, ο τουρίστας μπορεί να δει τα κοντινά του αξιοθέατα πού βρίσκονται με βάση την δική του τοποθεσία. Τα αξιοθέατα εδώ αντιπροσωπεύονται με εικόνες από το πώς ήταν παλιά. Και πάλι, ο τουρίστας μπορεί να επιλέξει ένα από αυτά και να δει μια σύντομη περιγραφή και την επιλογή να δει περισσότερες πληροφορίες.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Γραφικά Υπολογιστών

1.1.1 Τι είναι τα Γραφικά Υπολογιστών;

Τα Γραφικά Υπολογιστών είναι ένας κλάδος της Επιστήμης της Πληροφορικής. Ασχολούνται με την διαδικασία απεικόνιση ενός εικονικού περιβάλλοντος σε μια εικόνα με την χρήση υπολογιστών και ψηφιακών μέσων.

1.1.2 Κατηγορίες Γραφικών

Μπορούμε να κατατάξουμε τα γραφικά με δύο τρόπους. Με βάση τις διαστάσεις και με βάση το χρόνο στον οποίο γίνεται η απόδοσή τους[10, 11].
Γραφικά με βάση τις διαστάσεις:

- Δισδιάστατα γραφικά: στην κατηγορία αυτή ανήκουν τα γραφικά που έχουν δύο διαστάσεις. Παραδείγματα τέτοιων γραφικών είναι γραφικά για την δημιουργία διεπαφών χρήστη και διάφορων έντυπων μέσων.
- Τρισδιάστατα γραφικά: στην κατηγορία αυτή ανήκουν τα γραφικά που έχουν τρις διαστάσεις. Σήμερα η χρήση των τρισδιάστατων γραφικών αυξάνεται ολοένα και περισσότερο γιατί κάνουν τις εικόνες που αποδίδουμε πιο ρεαλιστικές. Τέτοιου είδους γραφικά συναντούμε κυρίως σε ηλεκτρονικά παιχνίδια και ταινίες.

Γραφικά με βάση το χρόνο απόδοσης τους:

- Στατικά γραφικά: στην κατηγορία αυτή ανήκουν τα γραφικά τα οποία δεν αποδίδονται κατά την διάρκεια της χρήσης τους. Τέτοιου είδους γραφικά δεν αλλάζουν όσες φορές και να τα χρησιμοποιήσουμε. Είναι γνωστά και ως μη διαδραστικά γραφικά., δηλ. δεν μπορούμε να αλληλεπιδράσουμε μαζί τους. Τέτοια γραφικά συναντούμε σε ταινίες.
- Γραφικά πραγματικού χρόνου: στην κατηγορία αυτή ανήκουν τα γραφικά τα οποία αποδίδονται κατά την διάρκεια της χρήσης τους. Τα γραφικά αυτά αλλάζουν συνεχώς κατά την εκτέλεση της εφαρμογής που τα χρησιμοποιεί και το πώς θα τροποποιηθούν δεν είναι γνωστό από πριν. Είναι γνωστά και ως διαδραστικά γραφικά, δηλ. μπορούμε να αλληλεπιδράσουμε μαζί τους και να καθορίσουμε το πώς θα αποδοθούν στο τέλος. Τέτοια γραφικά συναντάμε σε ηλεκτρονικά παιχνίδια.

1.1.3 Εφαρμογές

Όπως αναφέρθηκε και στις κατηγορίες των γραφικών, συναντάμε τα γραφικά σε αρκετούς τομείς. Κυριότεροι από αυτούς είναι [10]:

- Ηλεκτρονικά παιχνίδια: μια βιομηχανία που αυξάνεται ολοένα και περισσότερο τα τελευταία χρόνια. Πολύ μεγάλο μέρος των ηλεκτρονικών παιχνιδιών αποτελείται πλέων από γραφικά και οι λάτρεις των παιχνιδιών απαιτούν ολοένα και καλύτερη ποιότητα σε αυτά.
- Κινηματογράφος: η εξέλιξη της τεχνολογία έδωσε την ευκαιρία στον κινηματογράφο να εισαγάγει νέες ταινίες και να τις κάνει πιο ρεαλιστικές. Ταινίες σαν το Avatar και ο HarryPottereίναι παραδείγματα τέτοιων ταινιών αφού σου δίνουν την εντύπωση ότι βλέπεις πράγματα της καθημερινής ζωής. Μια πλέων γνωστή κατηγορία ταινιών, κυρίως δημοφιλή στα μικρά παιδία που πλέων έχει κατακτήσει και τις καρδιές των μεγάλων, είναι τα κινούμενα σχέδια.
- Επιστήμες: σε διάφορες επιστήμες, όπως η φυσική και η ιατρική, τα γραφικά έχουν καταφέρει να γίνονται πιο διαδεδομένα. Παρουσιάζοντας μια γραφική απεικόνιση των δεδομένων τους, επιτρέπει στους επιστήμονες να κατανοούν περισσότερο αυτά που έχουν να αναλύσουν, βοηθώντας τους να βγάλουν πιο ακριβή συμπεράσματα.
- Βιομηχανίες παραγωγής: οι βιομηχανίες αυτές, με την βοήθεια των γραφικών, μπορούν να σχεδιάσουν προϊόντα πολύ πιο γρήγορα και πιο φθηνά. Η χρονοβόρα διαδικασία του να φτιαχτεί ένα προϊόν από την αρχή μέχρι το τέλος για να αποφασιστεί κατά πόσο είναι ικανοποιητικό αποτελεί παρελθόν.

1.2 AugmentedReality

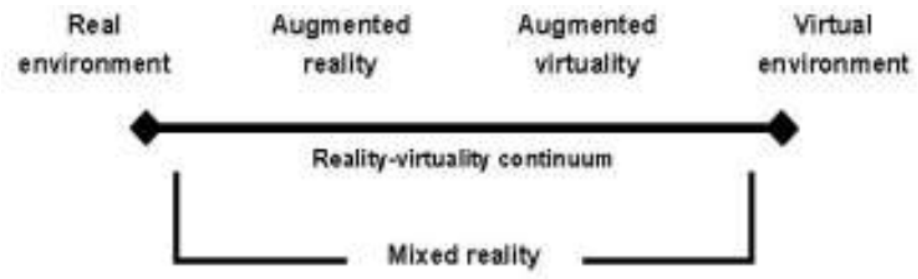
1.2.1 Τι είναι AugmentedReality;

Το AugmentedReality (AR)[12]είναι ένας κλάδος των γραφικών υπολογιστών, όπου στοιχεία και αντικείμενα του εικονικού κόσμου ενσωματώνονται και αναμειγνύονται με τον πραγματικό κόσμο. Για να μπορεί να θεωρηθεί μια εφαρμογή ως μια AR εφαρμογή θα πρέπει να έχει τα πιο κάτω χαρακτηριστικά[7, 4]:

- Συνδυάζει πραγματικά και εικονικά αντικείμενα.
- Είναι αλληλεπιδραστική (interactive).
- Τρέχει σε πραγματικό χρόνο (realtime).
- Στοιχίζει πραγματικά και εικονικά αντικείμενα μεταξύ τους.

- Χρησιμοποιεί τον τρισδιάστατο χώρο.

Αν και πιο πάνω φαίνονται τα στοιχεία που περιγράφουν τι καθορίζει μια εφαρμογή ως μια AR εφαρμογή θα πρέπει να αναφέρουμε ότι μια τέτοια εφαρμογή δεν περιορίζεται στο μέσο που χρησιμοποιεί για να προβάλει τα εικονικά αντικείμενα αλλά ούτε περιορίζεται μόνο στην όραση. Αντιθέτως υπάρχουν πολλά μέσα για την προβολή AugmentedReality, ενώ μπορεί να συνδυάζει ταυτόχρονα αντικείμενα εικονικού και πραγματικού κόσμου σε όλες τις ανθρώπινες αισθήσεις.



Εικόνα1.1: Reality - VirtualityContinuum

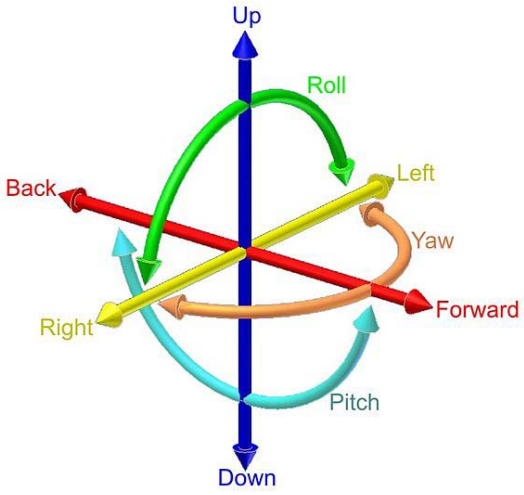
Στηνεικόνα 1.1 μπορούμε να δούμε πως μεταβαίνουμε από το πραγματικό περιβάλλον στο εικονικό. Οτιδήποτε υπάρχει ανάμεσα σε αυτά τα δύο άκρα ενοικεί την ανάμεικτη πραγματικότητα (Mixedreality).Σε αυτή την ανάμεικτη πραγματικότητα υπάρχουν δύο άλλα μέρη. Το πρώτο είναι το AugmentedRealityενώ το δεύτερο είναι το AugmentedVirtuality. Η διαφορά τους είναι ότι στο πρώτο προσθέτουμε εικονικά αντικείμενα σε ένα πραγματικό περιβάλλον ενώ στο δεύτερο προσθέτουμε πραγματικά αντικείμενα σε ένα εικονικό περιβάλλον.

Για να έχουμε ένα ολοκληρωμένο ARσύστημα θα πρέπει να έχουμε ένα μέσο καταγραφής του πραγματικού περιβάλλοντος, ένα μέσο στο οποίο θα παρουσιάζονται τα εικονικά αντικείμενα και μια συσκευή για να κάνει οποιαδήποτε επεξεργασία χρειάζεται.

1.2.2 Μέθοδοι Tracking

Ένα ARσύστημα για να μπορέσει να παρουσιάσει στο χρήστη τις εικονικές πληροφορίες σωστά θα πρέπει να γνωρίζει με ακρίβεια την τοποθεσία και τη κίνησή

του. Χωρίς την σωστή καταγραφή το σύστημα θα είναι δύσκολο να πείσει το χρήστη ότι τα εικονικά αντικείμενα είναι μέρος του πραγματικού κόσμου. Για τον λόγο αυτό χρησιμοποιείται συνήθως το 6 degreesoffreedom (6DOF),δηλ. καταγραφή της θέσης (positionx, y, z) και των γωνιών (yaw, pitch, roll). Οι μέθοδοι trackingδεν μπορούν να μπουν σε συγκεκριμένες κατηγορίες αφού μπορούν να κατηγοριοποιηθούν με διάφορους τρόπους όπως με βάση τις διαφορετικές προσεγγίσεις (με σχήματα ή όχι) ή με βάση το περιβάλλον (υπαίθριο - εσωτερικό).



Εικόνα-1.1: yaw, pitch, roll

Οι πιο κύριες μέθοδοι trackingφαίνονται πιο κάτω.[1,2,5]

GPS, GSM:

ToGPSαναφέρεταιστοGlobalPositioningSystem (GPS) μεβάση το οποίο αναγνωρίζεται η τοποθεσία του χρήστη. Είναι ιδανικό για εφαρμογές που χρησιμοποιούνται σε υπαίθριους χώρους μιας και η απόκλιση του είναι τρία με δέκα μέτρα περίπου. Υπολογίζεται ότι στο μέλλον θα μπορεί να γίνει και χρίση του GSM (GSM)για την αναγνώριση της τοποθεσίας του χρήστη.

Outside-in / Inside-in:

Με την μέθοδο outside-inγίνεται χρήση εξωτερικής κάμερας για την παρακολούθηση του χρήστη. Η ακρίβεια της τοποθεσίας του χρήστη αυξάνεται όταν η παρακολούθηση γίνεται με πολλαπλές κάμερες. Με την μέθοδο inside-inη κάμερα τοποθετείται στο κεφάλι του χρήστη και οποιαδήποτε κίνηση του χρήστη μεταφέρεται και στην κάμερα.

VisualMarker-based:

Μια από τις πιο γνωστές και συνηθισμένες προσεγγίσεις. Στην μέθοδο αυτή γίνεται χρίση εικόνων που απεικονίζουν απλά σχήματα και μοτίβα. Η κάμερα αναγνωρίζει τις εικόνες και τις αντιστοιχεί με συγκεκριμένα αντικείμενα τα οποία εμφανίζει στην θέσει των εικόνων. Τα εικονικά αντικείμενα μετακινούνται και περιστρέφονται σύμφωνα με την κίνηση της εικόνας.

VisualMarkerless:

Σε αντίθεση με την προηγούμενη μέθοδο, η αναγνώριση δεν γίνεται με εικόνες αλλά επεξεργαζόμαστε τον πραγματικό κόσμο και αναγνωρίζουμε διάφορα σημεία (φυσικά χαρακτηριστικά). Το πρόβλημα με αυτή την μέθοδο είναι η μεγάλη υπολογιστική δύναμη που απαιτείται για την αναγνώριση των σημείων.

Wireless-LAN

Με την ολοένα και αυξανόμενη χρήση της ασύρματης δικτύωσης μας παρέχεται η δυνατότητα να υπολογίζουμε την τοποθεσία του χρήστη. Αυτό οφείλεται στο γεγονός ότι κάποια πρωτόκολλα υποστηρίζουν τον υπολογισμός της απόστασης της συσκευής από το σημείο ασύρματης πρόσβασης. Ένα από τα πλεονεκτήματα αυτής της μεθόδου είναι ότι μπορεί να χρησιμοποιηθεί για εσωτερικούς χώρους. Υστερεί όμως στους υπαίθριους χώρους από είναι αδύνατη η παροχή ασύρματης δικτύωσης.

Hybrid

Στην μέθοδο αυτή γίνεται ένας συνδυασμός από διαφόρους αισθητήρες όπως το γυροσκόπιο (gyroscope) και το επιταχυνσιόμετρο (accelerometer).

1.2.3 Displays

Για την παρουσίαση των εικονικών αντικειμένων μπορούμε να επιλέξουμε μεταξύ δύο κατηγοριών συσκευών. Η πρώτη κατηγορία είναι οι opticalsee-throughσυσκευές όπου ο χρήστης μπορεί να δει απευθείας τον πραγματικό κόσμο όπως ένα είδος γυαλιών (πχ EyeTop). Η δεύτερη κατηγορία είναι οι videosee-through συσκευές όπου ο χρήστης βλέπει τον κόσμο μέσα από μια κάμερα.

1.2.4 Εφαρμογές

Το AugmentedRealityβρίσκει εφαρμογές σε αρκετούς τομείς. Κάποιοι από αυτούς αναφέρονται ποιο κάτω[1,2,3,5,6]:

Εκπαίδευση:
Οι μαθητές, και μάλιστα τα μικρά παιδιά, θα μπορούν μέσα από ένα εύκολο και διασκεδαστικό τρόπο να μάθουν πράγματα, όπως το αλφάβητο, τα ζώα, τα φυτά κτλ.

Πλοήγηση και Εύρεση Διαδρομών:
Μπορεί να χρησιμοποιηθεί στο να παρέχει κατευθύνσεις από ένα μέρος σε ένα άλλο, δίνοντας και πληροφορίες για τις διαδρομές.

Αρχαιολογία και Πολιτιστική κληρονομία:
Αποτελεί μια ενδιαφέρον εφαρμογή αφού μπορεί να χρησιμοποιηθεί σε μουσεία για να δείξει πως ήταν διάφορα ευρήματα και αρχαιολογικοί χώροι πριν πολλά χρόνια.

Διασκέδαση και Ηλεκτρονικά Παιχνίδια:
Ολοένα και περισσότερα παιχνίδια χρησιμοποιούν AR για να προσφέρουν περισσότερη διασκέδαση προσθέτοντας μια δόση πρόκλησης σε κάποια από αυτά.

Σχεδίαση εσωτερικού χώρου:
Με την χρήση ιδικών καρτελών ο χρήστης μπορεί να δει από προηγουμένως πως θα φαίνεται ένας χώρος έτσι ώστε να επιλέξει την καλύτερη τοποθέτηση.

Βιομηχανικές και στρατιωτικές εφαρμογές:
Σε αυτούς τους τομείς η εκπαίδευση των ατόμων αλλά και η εύρεση αντικειμένων σε βιομηχανικές ή στρατιωτικές αποθήκες μπορεί να γίνει πολύ πιο εύκολη με την χρήση ARεφαρμογών.

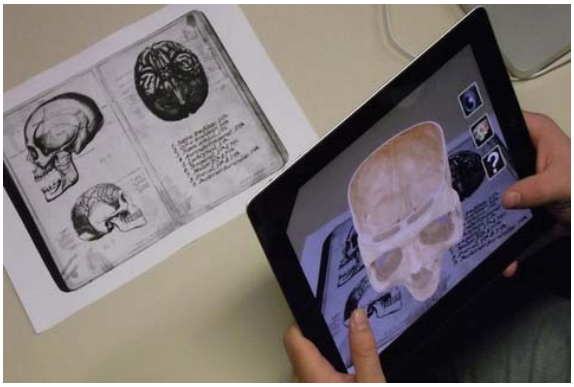
Ιατρική:
Με την βοήθεια τέτοιων εφαρμογών οι νοσοκόμοι και οι γιατροί θα μπορούν να έχουν εύκολη πρόσβαση στους φακέλους των ασθενών κατά την επίσκεψη τους αλλά και

κατά την διάρκεια επεμβάσεων, όπως η λαπαροσκόπηση.

Συντήρηση:

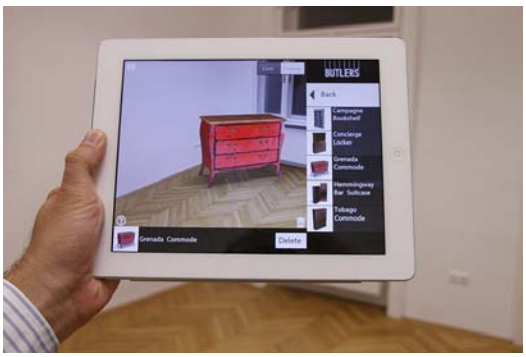
Πολλές φορές χρειάζεται να συντηρήσουμε μια συσκευή ή ένα μηχάνημα με το οποίο για πρώτη φορά θα έρθουμε σε επαφή μαζί του αλλά φοβόμαστε να το κάνουμε μόνοι λόγο έλλειψης γνώσης. Τη λύση μπορεί να μας την δώσει μια εφαρμογή που θα μας εξηγά τα μέρη αλλά και την πορεία που πρέπει να ακολουθήσουμε.

Εικόνα 1.7: Wikitude World Browser on the iPhone 3GS uses GPS and a solid state compass



Εικόνα1.2: App iSkull, an augmented human skull for education

Εικόνα 1.4: Augmented reality map on iPhone

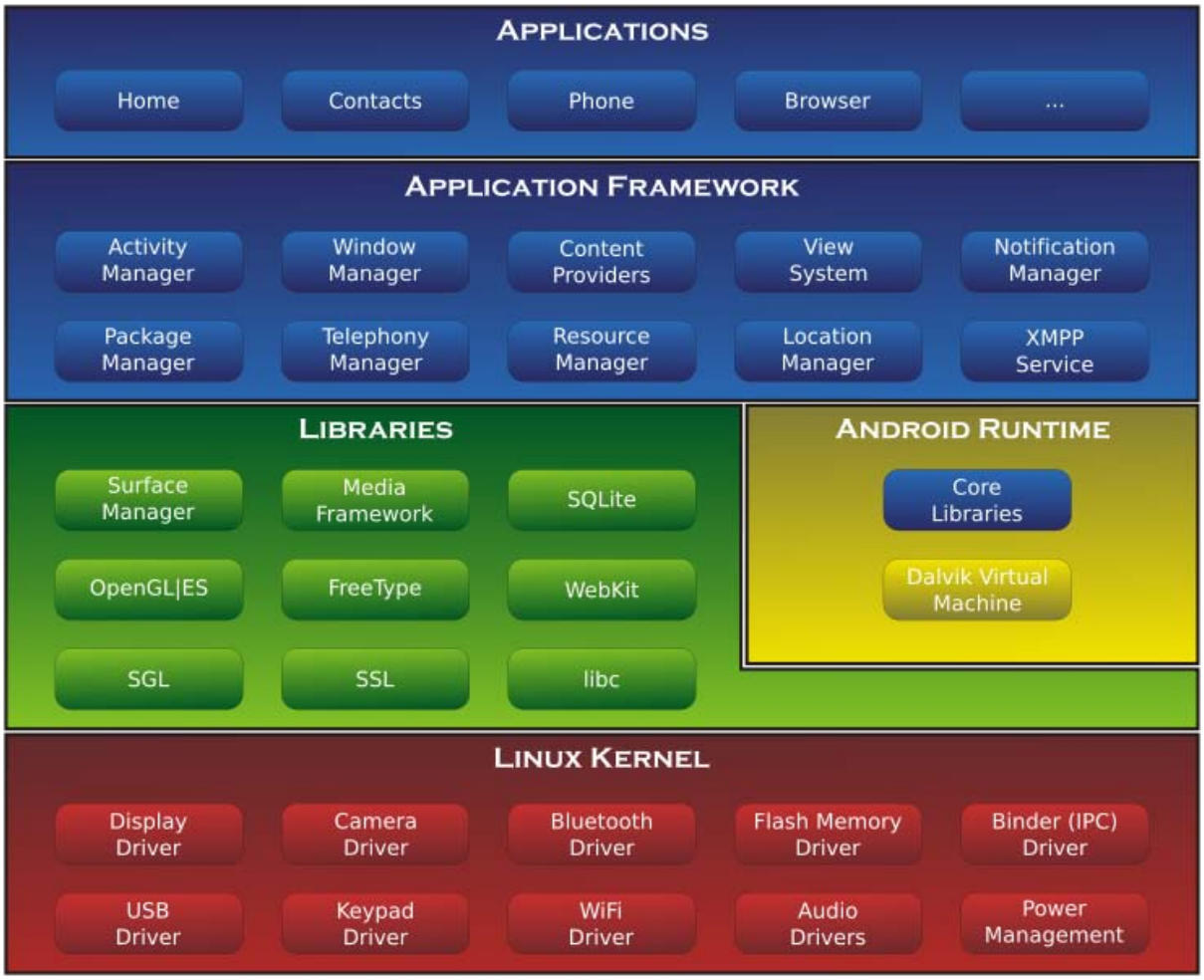


Εικόνα 1.5: ViewAR BUTLERS App - Placing furniture using AR

Εικόνα 1.6: Samsung SARI AR SDK markerless tracker used in the AR EdiBear game (Android OS)

1.3 Android

Το Androidείναι ένα λειτουργικό σύστημα ανοικτού πηγαίου κώδικα, το οποίο αναπτύχθηκε από την GoogleInc.[13]Εχει σχεδιαστεί κυρίως για κινητές συσκευές με οθόνες αφής καιtablets. Όπως φαίνεται και στην εικόνα 1.8 το Androidείναι βασισμένο πάνω σε ένα Linuxkernel. Παρόλα αυτά υπάρχουν κάποιες διαφορές με την παραδοσιακή Linuxαρχιτεκτονική, αφού τεχνολογίες πάνω από τον πυρήνα, όπως η udevκαι glibc, απουσιάζουν, καθιστώντας αδύνατη τη χρίση παραδοσιακών Linux εφαρμογών.Ένα από τα πλεονεκτήματα του Androidείναι ότι επιτρέπει στους προγραμματιστές να δημιουργούν εφαρμογές με τι γλώσσα προγραμματισμούJava.Με τη χρίση του AndroidSDK τροποποιείται ο τρόπος συγγραφής προγραμμάτων με τη χρίση της Javaπαρέχοντας επιπλέων δυνατότητες, χωρίς να ξεφεύγει από τον παραδοσιακή Java.



Εικόνα 1.8: AndroidSystemArchitecture

Αν και για την ανάπτυξη εφαρμογών χρησιμοποιείται η Java, για την εκτέλεση των προγραμμάτων δεν συμβαίνει το ίδιο και με το JavaVirtualMachine (JVM). Λόγω του ότι το AndroidOS προορίζεται για κινητές συσκευές και tables, τα οποία έχουν πιο αργή υπολογιστική δύναμη και λιγότερη RAM, το JVMδεν ανταποκρίνεται σε αυτές τις απαιτήσεις. Για τον σκοπό αυτό ανατήχθηκε μια άλλη εικονική μηχανή που ονομάζεται DalvikVirtualMachine (DVM)[14].

Αρκετά από τα χαρακτηριστικά του DVM το κάνει να είναι αρκετά διαφορετικό από το JVM. Το DVM[15] δεν μεταγλωττίζει τον κώδικα κατά τη εκτέλεση γιατί θα έκανε την εκτέλεση πολύ αργή. Το JVM μεταγλωττίζει τις κλάσεις σε .class αρχεία, ένα για κάθε κλάση, τα οποία αργότερα συμπεριλαμβάνονται στο jarαρχείο το οποίο εκτελείται. Σε αντίθεση το DVM κωδικοποιεί τα .class αρχεία σε ένα .dexαρχείο, το οποίο συνθέτει όλες τις κλάσεις μαζί και περιλαμβάνει όλες τις δηλώσεις σταθερών μεταβλητών και των υπογραφών των συναρτήσεων μόνο μια φορά. Αυτό οδηγεί στο να δημιουργείται ένα αρχείο με το μισό μέγεθος από το συνολικό μέγεθος των .class αρχείων. Το dexαρχείο μαζί με όλες τις βιβλιοθήκες και τους πόρους τοποθετούνται σε ένα apkαρχείο, αντίστοιχο του jar.

1.4 Στόχος Διπλωματικής Εργασίας

Στόχος αυτής της Διπλωματικής Εργασίας είναι η ανάπτυξη μια εφαρμογής, για Android κινητές συσκευές, που θα βοηθάει τους τουρίστες στις περιηγήσεις τους σε αρχαιολογικούς χώρους.

Συγκεκριμένα θα παρουσιάζει στον χρήστη ένα χάρτη της περιοχής που θα επισπευτεί (στη συγκεκριμένη περίπτωση για την περιοχή Ιδαλίου). Στο χάρτη αυτό θα παρουσιάζονται σαν σημεία όλα τα αρχαιολογικά μνημεία και χώροι. Όταν ο χρήστης επιλέξει ένα από αυτά πατώντας πάνω του, τότε θα του παρουσιάζεται το όνομα του χώρου ή μνημείου μαζί με μια σύντομη περιγραφή. Την ίδια ώρα με το πάτημα του κουμπιού “More” θα μπορεί να δει περισσότερες πληροφορίες για τον χώρο που επέλεξε. Θα δίνεται επίσης η δυνατότητα στον τουρίστα να λάβει οδηγίες για το πώς μπορεί να πάει από τον ένα χώρο στον άλλο ή από τη θέση που βρίσκεται.

Ένα άλλο σημαντικό μέρος της εφαρμογής θα είναι το AugmentedReality. Αυτό σημαίνει ότι, όταν ο χρήστης βρίσκεται κοντά σε κάποια από τα σημεία, θα μπορεί να ανοίξει την κάμερα της συσκευής του και να δει που βρίσκονται γύρω του αυτά τα μνημεία. Πατώντας πάνω σε κάποιο από αυτά θα μπορεί να δει στο κάτω μέρος της οθόνης του μια σύντομη περιγραφή και πατήσει πάνω στην περιγραφή, θα βλέπει περισσότερες πληροφορίες.

1.5 Overview

Μέχρι τώρα έχουμε δει ποιο είναι το θέμα της παρούσας Διπλωματικής Εργασίας καθώς και κάποια εισαγωγικά στοιχεία που αφορούν τα Γραφικά Υπολογιστών, το AugmentedReality και το Android. Στην συνέχεια θα δούμε στο κεφάλαιο δύο κάποιες AugmentedReality βιβλιοθήκες για Android και κάποιες εφαρμογές που υπάρχουν. Στο κεφάλαιο 3θα δούμε θέματα σχεδίασης όπως την ροή το δεδομένων και την αρχιτεκτονική της εφαρμογής. Στο κεφάλαιο 4 θα δούμε αναλυτικά τον τρόπο υλοποίησης της εφαρμογής και στο κεφάλαιο 5 θα παρουσιάσουμε τα αποτελέσματα και τα συμπεράσματα.

Κεφάλαιο 2

Προηγούμενη Εργασία

2.1 AR Βιβλιοθήκες για Android

Wikitude

Το Wikitudeείναι μια συλλογή από εργαλεία τα οποία δίνουν την δυνατότητα ανάπτυξης εφαρμογών επαυξημένης πραγματικότητας βασισμένες στην γεωγραφική θέση ή στην αναγνώριση εικόνων. Ένα από τα πλεονεκτήματά του, είναι η απλότητα της χρίσης που βασίζεται σε HTML5, CSSκαι JavaScript. Προσφέρεται τόσο για ανάπτυξη σε λειτουργικό Androidόσο και για iOS. Προσφέρεται δωρεάν για την ανάπτυξη μη εμπορικών εφαρμογών, ενώ για την ανάπτυξη εμπορικών εφαρμογών απαιτείται η αγορά άδειας.

Vuforia

Σε αντίθεση με το Wikitudeτο Vuforiaειδικεύεται στις εφαρμογές οι οποίες χρησιμοποιούν ComputerVisionτεχνολογίες. Προσφέρεται δωρεάν για Androidκαι iOS για τη χρήση του για εμπορικούς και μη σκοπούς.

ARToolKit

Μια βιβλιοθήκη ανοικτού πηγαίου κώδικα. Βασίζεται στην αναγνώριση μοτίβων και την αναπαράσταση τους σε τρισδιάστατα μοντέλα. Επιπλέων έχει την δυνατότητα να

αναγνωρίζει την κίνηση των μοτίβων και να κινεί των μοντέλο που προβάλλεται. ΕπιτρέπειτηνανάπτυξηεφαρμογώνσεSGIIRIX, MicrosoftWindows, MacOSX,Linux-basedsystems, Symbian, iPhone, Androidκαι WindowsPhone.

Mixare

Είναι πρόγραμμα ανοικτού πηγαίου κώδικα, το οποίο με μερικές τροποποιήσεις μπορεί να ενσωματωθεί στην εφαρμογή που θέλουμε. Χρησιμοποιεί geo-locationtracking και προσφέρεται για Androidκαι iOSεφαρμογές.

ARLab

Το ARLab είναι μια σουίτα λογισμικού για την παροχή χρίσης AugmentedReality σε εφαρμογές κατάλληλες για smartphones, desktopsκαι άλλες πλατφόρμες. Κύριο χαρακτηριστικό είναι η αναγνώριση αντικειμένων και εικόνων.

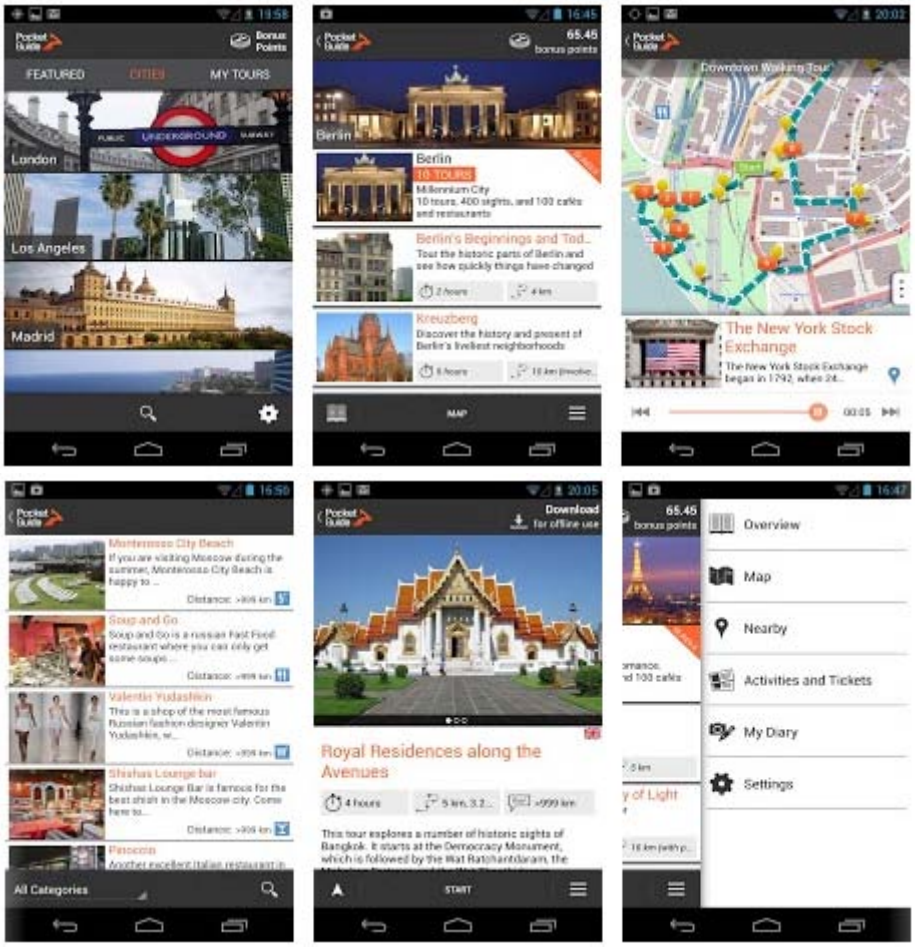
ARmsk

Όπως και το ARLab έτσι και το ARmskαναγνωρίζει εικόνες και αντικείμενα χωρίς τη χρίση μοτίβων. Διαφέρουν στο γεγονός ότι το ARmsk είναι πρόγραμμα ανοικτού πηγαίου κώδικα και επιτρέπει την ανάπτυξη εφαρμογών για Androidσυσκευές.

2.2 AR Εφαρμογέςγια Android

Pocket Guide

Το PocketGuide είναι μια εφαρμογή που καθοδηγεί του τουρίστες σε μια πόλη με τη χρήση ήχου (φωνής). Είναι διαθέσιμο για 64 περίπου χώρες παγκοσμίως. Η εφαρμογή αυτή εντοπίζει την τοποθεσία του χρήστη με GPS και του δίνει οδηγίες για το που μπορεί να βρει αξιοθέατα σε μια πόλη και του προτείνει εστιατόρια και καφετέριες που βρίσκονται κοντά του. Επίσης ο χρήστης μπορεί να δημιουργήσει ένα βίντεο με υλικό που έχει μαζέψει κατά την περιήγησή του και να το μοιραστεί με φίλους. Οι πληροφορίες και τα δεδομένα που χρειάζεται η εφαρμογή δεν βρίσκονται μαζί με την εφαρμογή και για αυτό απαιτείται ηχρήση Wi-Fi. Παρέχεται όμως η δυνατότητα, ο χρήστης να μπορεί να κατεβάσει δωρεάν το χάρτη για offlineχρήση, ωστόσο τον τουριστικό οδηγό θα πρέπει να τον αγοράσει για να μπορεί να τον χρησιμοποιήσει offline.



Εικόνα 2.1: Εικόνες από screenshots της εφαρμογής PocketGuide στο GooglePlay

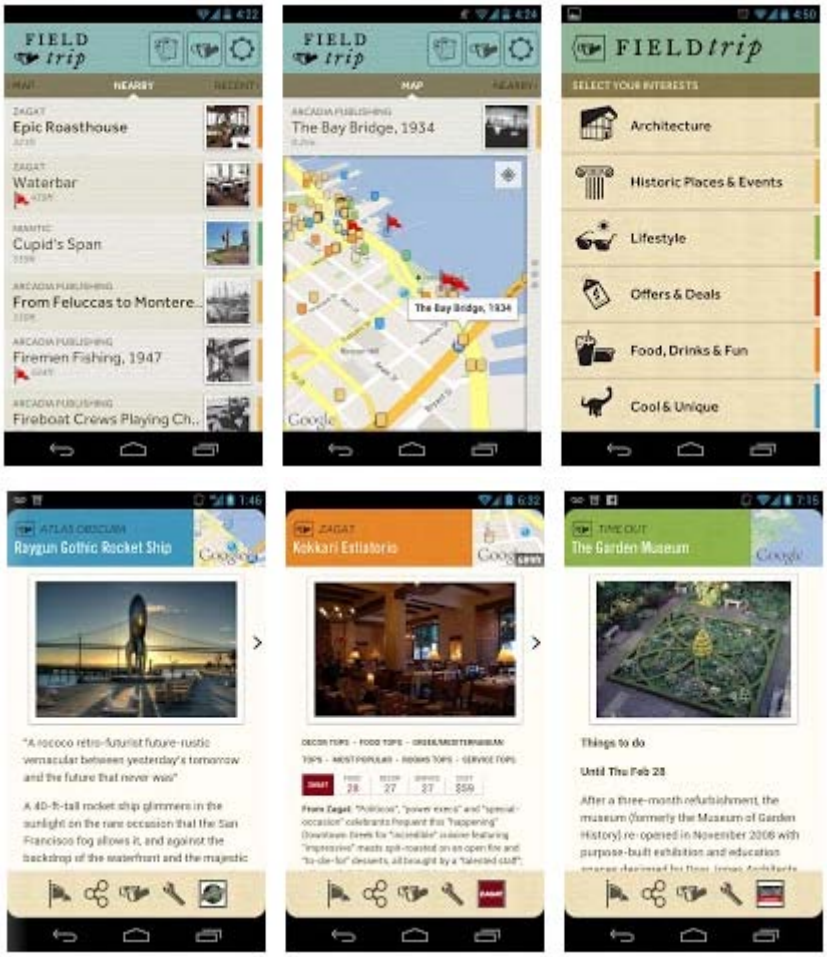
mTrip

Είναι μια offlineεφαρμογή που βοηθά τους χρήστες να οργανώσουν τα ταξίδια τους. Τους παρέχει πληροφορίες για μνημεία, εστιατόρια, καφετέριες και άλλα, μαζί με σχόλια και κριτικές, καθώς επίσης και τις ώρες λειτουργίας τους. Με την χρήση AugmentedRealityη εφαρμογή δείχνει τα πιο κοντινά σημεία ενδιαφέροντος μαζί με πληροφορίες όπως η απόσταση του χρήστη από αυτά.

Εικόνα 2.2: Εικόνες από screenshots της εφαρμογής mTrip στο GooglePlay

FieldTrip

ToFieldTripείναι μιαεφαρμογή σχεδιασμένοι για όσους τους αρέσει να κάνουν βόλτες. Είναι ιδικά σχεδιασμένη για smartphones.Τρέχει στο παρασκήνιο και όταν ο χρήστης πλησιάσει σε κάποιο σημείο ενδιαφέροντος η εφαρμογή το ειδοποιεί. Ο χρήστης μπορεί να επιλέξει κατηγορία με σημεία που τον ενδιαφέρουν περισσότερο.



Εικόνα 2.3: Εικόνες από screenshots της εφαρμογής FieldTrip στο GooglePlay

TouristGuideforGreece

Ένας τουριστικός οδηγός για την Ελλάδα. Συνδυάζει χαρακτηριστικά από τις προηγούμενες εφαρμογές, όπως το να ενημερώνει τον χρήστη πότε βρίσκεται κοντά σε ένα σημείο ενδιαφέροντος, να παρέχει πληροφορίες για κάποιο από αυτά με τη χρήση φωνής και να προτείνει εστιατόρια, καταστήματα και ξενοδοχεία.

Εικόνα 2.4: Εικόνες από screenshots της εφαρμογής TouristGuideforGreece στο GooglePlay

Κεφάλαιο 3

Σχεδίαση Εφαρμογής

3.1 Εργαλεία

Για την υλοποίηση του προγράμματός χρειάστηκε να χρησιμοποιήσω μια ομάδα εργαλείων τόσο για την συλλογή των δεδομένων όσο και για την ανάπτυξη του κώδικα.

MobileAtlasCreator 1.9.10(MOBAC) [16]

Με αυτή τη μικρή εφαρμογή μπορούμε να πάρουμε τα maptiles που χρειαζόμαστε για την χρήση του χάρτη offline. Το μόνο που χρειάζεται να κάνουμε είναι να επιλέξουμε την πηγή από όπου θα πάρουμε τα maptiles, να επιλέξουμε τα επίπεδα μεγέθυνσης και τη μορφή στην οποία θα πρέπει να τα εξάγουμε. Για την συγκεκριμένη εφαρμογή χρησιμοποιήσαμε ως πηγή για τα tilesto OpenStreetMapMapQuestκαι εξάγαμε τα maptilesσε MBTiles που είναι μια SQLite βάση.

OpenStreetMapExportTool [17]

Εκτός από τα maptiles για να μπορεί να παρουσιαστεί ο χάρτης, πρέπει να έχουμε και τα δεδομένα των δρόμων για να μπορεί η εφαρμογή να δίνει κατευθύνσεις στους χρήστες. Τα δεδομένα αυτά τα πήραμε από την επίσημη σελίδα του OpenStreetMap από την επιλογή Export. Στο σημείο αυτό επιλέγουμε το είδος του χάρτη (MapQuest), τα όρια του χάρτη που μπορούν να δοθούν με δύο τρόπους, δύοντας συγκεκριμένα όρια (πρέπει να ταυτίζονται με αυτά που φαίνονται στο MOBAC κατά την εξαγωγή των maptiles) ή επιλέγοντας από τον χάρτη την περιοχή που θέλουμε. Από τις επιλογές για εξαγωγή επιλέγουμε OpenStreetMapXMLData. Το αρχείο αυτό είναι ένα XMLαρχείο που περιγράφει τους κόμβους και τις συνδέσεις μεταξύ τους.

OSMParser

Εργαλείο αυτό αναπτύχθηκε στα πλαίσια αυτής της Διπλωματικής Εργασίας για την επεξεργασία των OpenStreetMapXMLδεδομένων. Η ανάπτυξη του εργαλείου αυτού οφείλεται στο ότι είναι μια επεξεργασία η οποία μπορεί να γίνει μια φορά μόνο ανεξάρτητα της εκτέλεσης της εφαρμογής. Με τον τρόπο αυτό μειώνεται η επεξεργασία που πρέπει να κάνει η εφαρμογή και γίνεται πιο γρήγορη. Η υλοποίηση του εργαλείου θα παρουσιαστεί στο επόμενο κεφάλαιο (Κεφάλαιο 4: Υλοποίηση Εφαρμογής).

Latitude and Longitude of a Point [18]

Αυτό το εργαλείο είναι ένα διαδικτυακός χώρος, όπου ο χρήστης πατώντας πάνω στο χάρτη τοποθετεί markers τα οποία του δείχνουν τη θέση του σε ένα σύστημα γεωγραφικών συντεταγμένων. Συγκεκριμένα χρησιμοποιεί το γεωγραφικό μήκος και πλάτος.

ElevationFinder [19]

ΌπωςκαιτοLatitudeandLongitudeofaPointέτσικαιτοElevationFindereίναιέναςδιαδικτυακόςχώρος. Λειτουργεί με τον ίδιο ακριβώς τρόπο, με τη διαφορά ότι εδώ παίρνουμε το υψόμετρο.

Eclipse [20]

Το Eclipse είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών (Integrateddevelopmentenvironment - IDE). Αυτό το περιβάλλον θα χρησιμοποιήσουμε για να γράψουμε τον κώδικα της εφαρμογής. Για να μπορεί τοEclipse να αναγνωρίσει ότι γράφου κώδικα Androidθα πρέπει να εγκατασταθεί το αντίστοιχο plug-in. Αυτό θα επιτρέψει στην διεπαφή του Eclipse να τροποποιηθεί ώστε να χρησιμοποιήσουμε τα εργαλεία του Android.

AndroidSDK[21]

Το AndroidSDK παρέχει τις βιβλιοθήκες και τα εργαλεία για την ανάπτυξη, την δοκιμή και την αποσφαλμάτωση Android εφαρμογών. Η εγκατάσταση του γίνεται ξεχωριστά από την εγκατάσταση του Eclipse και ι σύνδεσή τους γίνεται μέσο του plug-inπου εγκατασταίνεταιστοEclipse. Η Googleδίνει την δυνατότητα της εγκατάστασης όλων των απαιτούμενο εργαλείων (EclipseandADTplug-in, AndroidSDKTools, AndroidPlatform-tools, thelatestAndroidplatform, thelatestAndroidsystemimagefortheemulator)σαν ένα δέμα χωρίς περεταίρω ρυθμίσεις.

3.2 Βιβλιοθήκες

osmdroid-android-3.0.8 καιslf4j-android-1.5.8 [22, 23]

Η βιβλιοθήκη osmdroidείναι μια δωρεάν βιβλιοθήκη που αντικαθιστά σχεδόν πλήρως την κλάση MapView που προσφέρεται από το Android SDK. Η βιβλιοθήκη slf4j (Simple Logging Facade for Java) είναι μια βοηθητική βιβλιοθήκη για να μπορεί να χρησιμοποιηθεί η osmdroid.

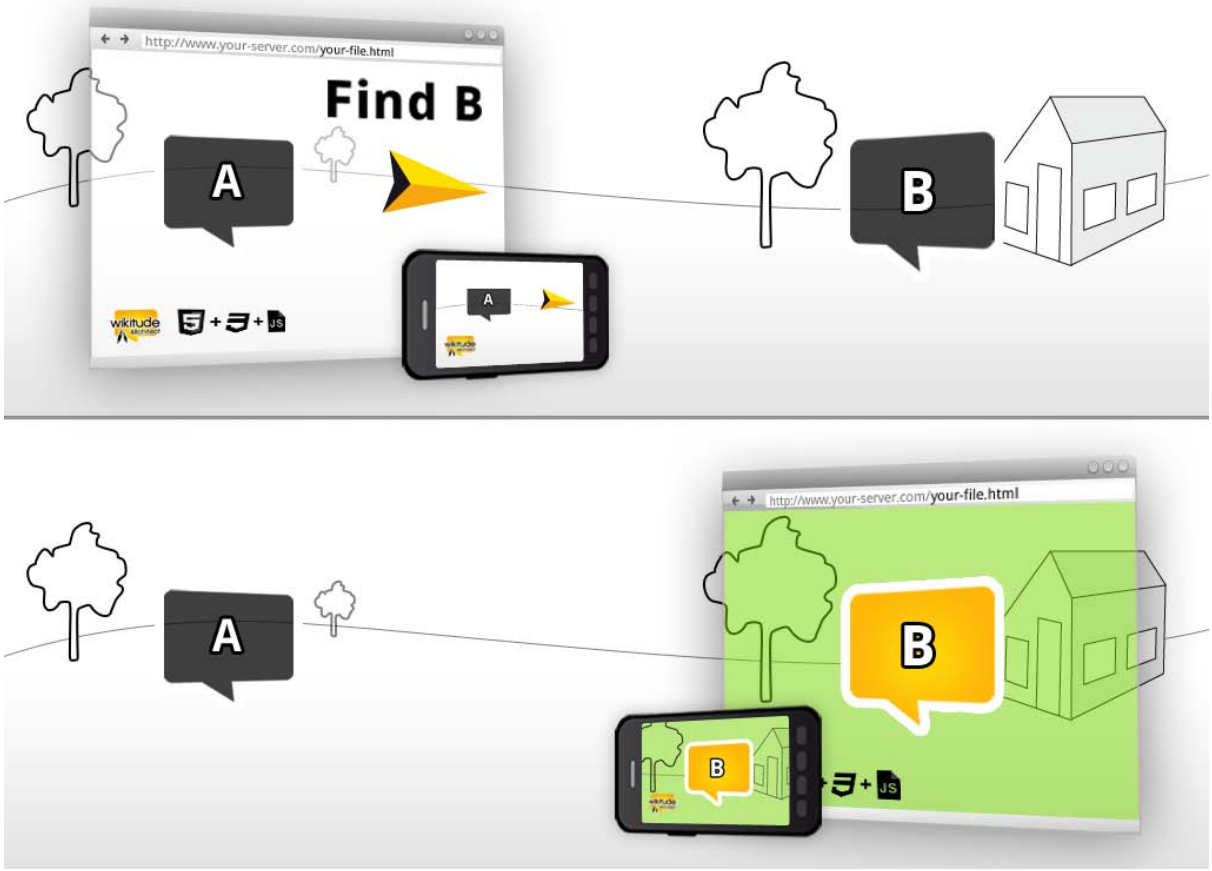
JGraphTJDK 1.6 [24]

Είναι μια δωρεάν βιβλιοθήκη για την γλώσσα προγραμματισμού Java.Προσφέρει υλοποίηση κατευθυνόμενου και μη γράφου μαζί με αλγόριθμους που σχετίζονται με γράφο. Η χρήση του είναι απλή γιατί το μόνο που απαιτείται είναι η δημιουργία του

γράφου με μια εντολή, η προσθήκη των κόμβων και ακολούθως των ακμών. Πάνω στο γράφο αυτό μπορούμε να τρέξουμε προ-υλοποιημένες συναρτήσεις για την εύρεση μονοπατιών.

WikitudeSDK 2.0 [26]

ToWikitudeείναι η βιβλιοθήκη που επιλέξαμε για την υλοποίηση του AugmentedRealityκομματιού της εφαρμογής. Όπως αναφέρθηκε και πιο πάνω στα πλαίσια της Javaαρχικοποιούμε το «κόσμο» με τα σημεία ενδιαφέροντος. Έχοντας δημιουργήσει τον κόσμο των σημείων, με την χρήση HTML, CSSκαι JavaScriptμορφοποιούμε τα σημεία μας και την εμφάνιση το υς και τους δίνο υμε ιδιότητες, όπως το να μπορούμε να τα επιλέξουμε.



Εικόνα 3.1: Τρόπος λειτουργίας Wikitude

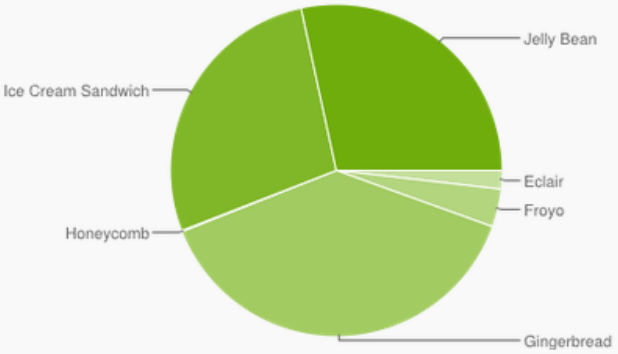
6.3 Απαιτήσεις Συστήματος

6.3.1 Απαιτήσεις Λογισμικού

Κατά την δημιουργία της εφαρμογής πρέπει να επιλέξουμε την έκδοση του Androidγια το οποίο προορίζεται η εφαρμογή. Για τον λόγο αυτό έγινε μια σύντομη έρευνα για την έκδοση του Android που είναιεγκατεστημένο σε κινητές συσκευές και tablet. Με βάση τις τελευταίες μελέτες και στατιστικές από την Google, όπως φαίνεται για στην εικόνα 3.2, το μεγαλύτερο μέρος το κατέχει η έκδοση 2.3 Gingerbread με ποσοστό 38% και ακολουθούν οι εκδόσεις 4.0 IceCreamSandwich με ποσοστό 27% και 4.1 JellyBeanμε ποσοστό 26%. Για το λόγο αυτό σαν έκδοση Androidγια την εφαρμογή επιλέξαμε το 2 3 . Ένα άλλο πλεονέκτημα πο υ μας δίνεται είναι ότι η εφαρμογή θα μπορεί να εγκατασταθεί σε όλες τις συσκευές με έκδοση Android 2.3 και άνω χωρίς καμιά τροποποίηση.

Version	Codename	API	Distribution
1.6	Donut	4	0.1%
2.1	Eclair	7	1.7%
2.2	Froyo	8	3.7%
2.3 - 2.3.2	Gingerbread	9	0.1%
2.3.3 - 2.3.7		10	38.4%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	27.5%
4.1.x	Jelly Bean	16	26.1%
4.2.x		17	2.3%

Data collected during a 14-day period ending on May 1, 2013.
Any versions with less than 0.1% distribution are not shown.



Εικόνα 3.2: Στατιστική ανάλυση των εκδόσεων Android που χρησιμοποιούνται σε συσκευές

3.3.2 Απαιτήσεις Υλικού

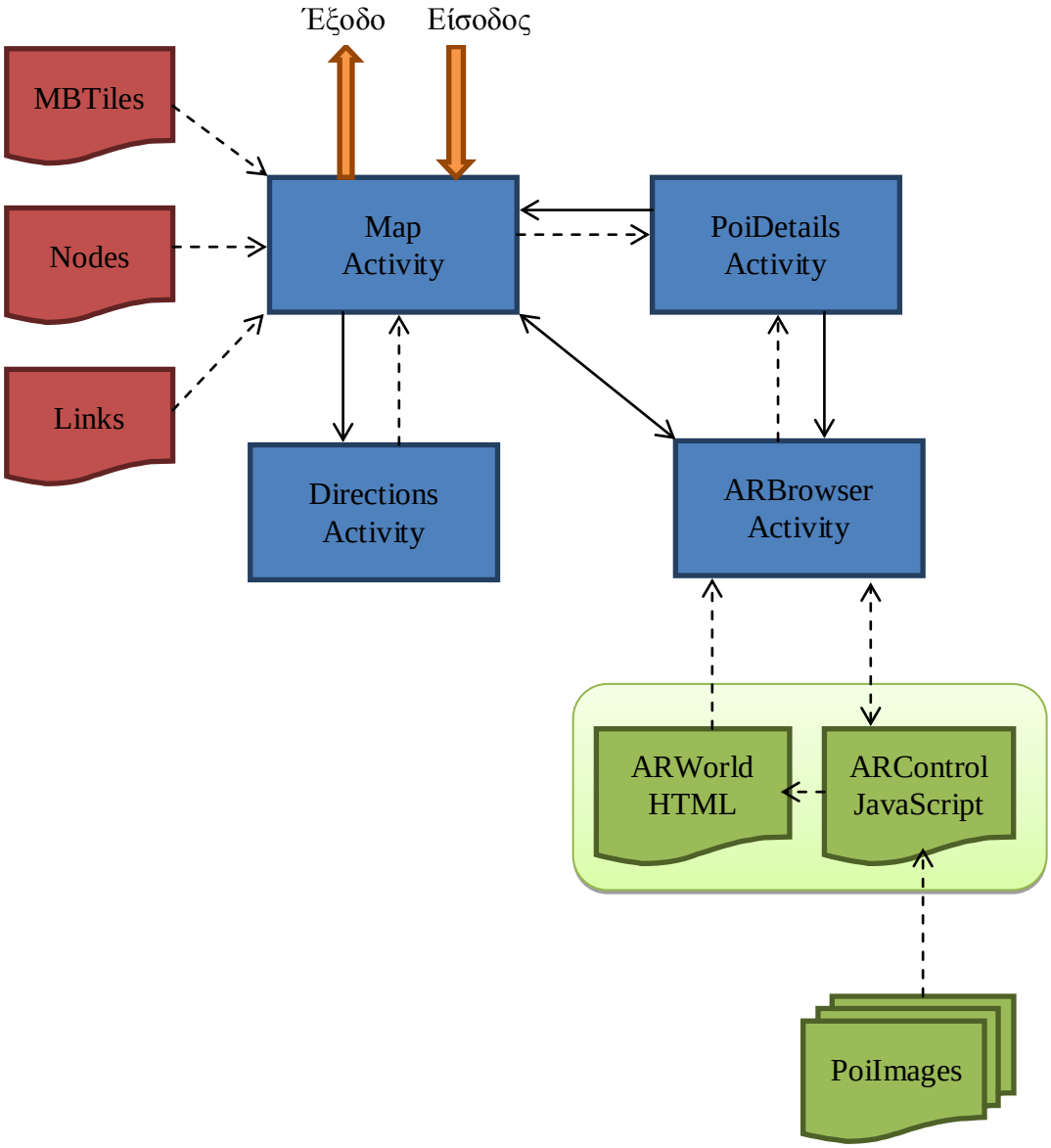
Για την σωστή και καλή λειτουργία της εφαρμογής θα πρέπει να λάβουμε υπόψη μας κάποιες απαιτήσεις υλικού που πρέπει να ικανοποιούνται.. Αυτές οι απαιτήσεις σχετίζονται με την χωρητικότητα που πρέπει να διατεθεί για την εφαρμογή και τους αισθητήρες που πρέπει να διαθέτει η συσκευή.

Για την χωρητικότητα απαιτείται ελεύθερος χώρος στην συσκευή το ελάχιστο 31MB, ενώ θα πρέπει να υπάρχει και μια SDCardγια την αποθήκευση των maptiles. Για τα tilesστην SDCardαπαιτούνται περίπου 30MB.

Από την πλευρά των αισθητήρων, η συσκευή θα πρέπει να διαθέτει γυροσκόπιο και επιταχυνσιόμετρο. Το γυροσκόπιο είναι ο αισθητήρας που αναγνωρίζει αν η κάμερα της συσκευής δείχνει προς τα πάνω, κάτω, δεξιά ή αριστερά. Το επιταχυνσιόμετρο αναγνωρίζει πόσο γρήγορα κινείται η κάμερα. Αν από τη συσκευή απουσιάζει το γυροσκόπιο τότε τα σημεία δεν θα μετακινούνται από την οθόνη εκτός και αν η κάμερα αλλάζει απότομα κίνηση όπου θα ανιχνεύεται από το επιταχυνσιόμετρο. Το πρώτο μέρος της εφαρμογής θα λειτουργεί κανονικά γιατί είναι ανεξάρτητο από αυτό το κομμάτι.

Και για τα δύο μέρη απαιτείται όπως η συσκευή διαθέτει GPS και είναι ενεργοποιημένο σε όλη την διάρκεια τη ς εφαρμογής.

3.4 Συστατικά Συστήματος



Εικόνα 3.3: Ροές Δεδομένων και Κλήσεις των Activities

Στη εικόνα του 3.3 φαίνεται διαγραμματικά τα διάφορα μέρη της εφαρμογής. Με μπλε χρώμα αναπαριστούνται το activities της εφαρμογής, δηλ. οι διάφορες οθόνες που μπορεί να δει ο χρήστης. Με κόκκινο και πράσινο χρώμα αναπαριστούνται τα αρχεία που χρησιμοποιούνται στην εφαρμογή και είτε δίνουν δεδομένα είτε προσφέρουν κάποιες λειτουργίες. Τα διακεκομμένα βέλη από κάποιο αρχείο σε κάποιο activity δείχνουν ροή δεδομένων, ενώ από ένα activity σε ένα άλλο δείχνουν κλήση ή επιστροφή σε αυτό το activity με την μεταφορά δεδομένων. Τα συνεχή βέλη

αντιπροσωπεύουν απλές κλήσεις ή επιστροφές σε activity. Τα πορτοκαλί βέλη δηλώνουν τα σημεία από όπου μπορεί να ξεκινήσει και να τελειώσει η εφαρμογή.

Συγκεκριμένα, η εφαρμογή ξεκινά με το MapActivity. Όπως παρουσιάζεται στην εικόνα 3.3, σε αυτό το activityπαίρνουμε δεδομένα από τρία αρχεία. Το πρώτο αρχείο (MBTiles)είναι το αρχείο με τα tilesγια την δημιουργία του χάρτη. Τα άλλα δύο αρχεία (Nodesκαι Links) χρησιμοποιούνται για την δημιουργία του γράφου για τις διαδρομές. Στο αρχείο Nodeseίναι αποθηκευμένοι όλοι οι κόμβοι του γράφου μαζί με το γεωγραφικό πλάτος και γεωγραφικό μήκος τους στην μορφή <nodeid><space><latitude><space><longitude>, π.χ. 06192001 35.14469 33.411499. Στο αρχείο Linkseίναι αποθηκευμένες όλες οι ακμές του γράφου που αναπαριστούνται με ζευγάρια κόμβων οι οποίοι ανήκουν στην ακμήμε την μορφή <nodeid><nodeid>, π.χ. 1887912610 1887912560.

Όταν ο χρήστης βρίσκεται στο MapActivity και έχει επιλέξει να του δώσουμε οδηγίες για το πώς μπορεί να φτάσει σε ένα μέρος τότε καλείται το DirectionsActivity. Το activityαυτό δεν παίρνει δεδομένα ως είσοδο. Ωστόσο, αφού ο χρήστης επιλέξει από ποιο μέρος σε ποιο θέλει να λάβει οδηγίες, το DirectionsActivity τερματίζει και στέλνει πίσω στο MapActivityτην επιλογή του χρήστη.

Πατώντας από το μενού την επιλογή ARView στο MapActivity τότε γίνεται κλήση στο ARBrowserActivity.Επειδή ο χάρτης και το AugmentedRealityείναι εντελώς ανεξάρτητα το ένα από το άλλο, ούτε κατά την κλήση του ARBrowserActivity, ούτε κατά την επιστροφή στο MapActivityέχουμε ανταλλαγή δεδομένων.

Με την κλήση του ARBrowserActivity δημιουργείται και ο επαυξημένος κόσμος. Για την δημιουργία του κόσμου αυτού, το ARBrowserActivity χρησιμοποιεί το ARWorldHTML αρχείο για να παρουσιάσει την διεπαφή του χρήστη πάνω από την εικόνα που λαμβάνει από την κάμερα της συσκευής του. Αφού έχουμε ετοιμάσει τα δεδομένα για τα σημεία ενδιαφέροντος τα περνούμε στο αρχείο ARControlJavaScriptπου αναλαμβάνει να παρουσιάσει τα σημεία στην οθόνη και να τα διαχειριστεί ανάλογα με τις επιλογές του χρήστη. Το JavaScriptαρχείο ενημερώνει στοιχεία του ARWorldHTML αρχείου για την παρουσίαση κάποιων πληροφοριών για

το σημείο που επέλεξε ο χρήστης ή ενημερώνει το ARBrowserActivity για πιο σημείο θέλει να δει περισσότερες πληροφορίες ο χρήστης, δύοντας του την ταυτότητα του σημείου.

Όταν βρισκόμαστε στο MapActivity ή στο ARBrowserActivity μπορούμε να δείξουμε περισσότερες πληροφορίες στον χρήστη για κάποιο σημείο ενδιαφέροντος που επιθυμεί. Αυτό γίνεται με την κλήση του PoiDetailsActivityκαι δίνοντας του ως δεδομένα εισόδου την ταυτότητα του σημείου Πατώντας το κουμπί BACK για να φύγουμε από το activity, τότε επιστρέφουμε στο activityαπό το οποίο έγινε η κλήση του PoiDetailsActivit.

Για την έξοδο από την εφαρμογή ο χρήστης πρέπει να βρίσκεται στο MapActivity.Αυτό συμβαίνει γιατί το MapActivity είναι το κομβικό σημείο από όπου γίνεται η κλήση όλων των άλλων activities.

Κεφάλαιο 4

Υλοποίηση Εφαρμογής

4.1 OSMParser

Στο κεφάλαιο 3 τμήμα 3.1 μιλήσαμε για το εργαλείοOSMParser. Το εργαλείο αυτό όπως αναφέρθηκε και πιο πάνω αναπτύχθηκε στα πλαίσια αυτής της Διπλωματικής Εργασίας. Σκοπός του είναι η επεξεργασία του αρχείου OpenStreetMapXMLData αρχείου. Για την υλοποίηση του χρησιμοποιήσαμε τον XMLParserπου προσφέρει η Java.

Το πρώτο πράγμα που χρειάστηκε να κάνουμε είναι να δημιουργήσουμε μια κλάση με προκαθορισμένες εντολές για την κλήση του Parserπου τροποποιήσαμε. Το αρχείο που

θα επεξεργαστούμε επιλέξαμε να δίνεται ως παράμετρος ώστε να μπορεί να χρησιμοποιηθεί και μελλοντικά για τηνεπεξεργασία και άλλον τέτοιων αρχείων. Αυτός ο κώδικας μας έδωσε το αρχείο Main.java.

Για να επεξεργαστούμε τα δεδομένα μας δημιουργήσαμε την κλάση OsmParser στο αντίστοιχο java αρχείο και υλοποιήσαμε τις αφηρημένες συναρτήσεις του προκαθορισμένου Parser. Οι συναρτήσεις που υλοποιήσαμε είναι η startDocument, η endDocument, η startElement και η endElement.

startDocument

Στην συνάρτηση startDocument αναγνωρίζουμε ότι έχει ξεκινήσει το αρχείο που θέλουμε να επεξεργαστούμε. Είναι η πρώτη συνάρτηση που θα καλεστεί και αυτό κάνουμε είναι να αρχικοποιήσουμεδύο BufferedWritersγια την δημιουργία των αρχείων Nodes.txtκαι Links.txt.

startElement

Ακολούθως θα καλείται η συνάρτηση startElement μόλις αναγνωρίσουμε ότι έχει ξεκινήσει ένα xmlστοιχείο (π.χ. <node>)και θα ακολουθείται από την συνάρτηση endElement μόλις αναγνωρίσουμε ότι έχει τελειώσει ένα xmlστοιχείο (π.χ. </node>). Η σειρά κλήσεων των δύο συναρτήσεων δεν ακολουθεί πάντα αυτό το μοτίβο γιατί έχουμε στοιχεία που στο σώμα τους έχουν άλλα στοιχεία (π.χ. <way><node/></way>).

startElement

Στη συνάρτηση startElement μόλις αναγνωρίσουμε ένα στοιχείο node τότε περνούμε από τα χαρακτηρίστηκα του το id, το lat και το lon και τα γράφουμε απευθείας στο αρχείο Nodes.txt. Για την περίπτωση του στοιχείου wayχρειάζεται να κάνουμε περισσότερη επεξεργασία. Το στοιχείο wayέχει στο σώμα του μια λίστα από κόμβους που το αποτελούν και μια λίστα από tags που δηλώνουν το είδος του δρόμου. Από αυτά τα tagsχρειαζόμαστε μόνο αυτά που δηλώνουν το wayείναι δρόμος (highway)και αν είναι μονόδρομος (oneway).Οι ενέργειες που κάνουμε για αυτό το στοιχεία είναι μόλις βρούμε τη αρχή του στοιχείου wayνα δημιουργούμε μια προσωρινή λίστα που θα αποθηκεύει τους κόμβους και θέτουμε μια μεταβλητή ίσο με truepου σημαίνει ότι έχει ξεκινήσει ένα .στοιχείο way. Όταν βρούμε το στοιχείο nd, δηλ. κόμβο, τοποθετούμε το

idτου στην προσωρινή λίστα. Όταν αναγνωρίσουμε το στοιχείο tagελέγχουμε τη μεταβλητή που μαw λέει αν επεξεργαζόμαστε το waykαι αν ναι τότε ελέγχουμε αν βρήκαμε onewayή highway και ενημερώνουμε τις ανάλογες μεταβλητές.

endElement

Στη συνάρτηση endElement μας ενδιαφέρει μόνο πότε τελειώνει το στοιχείο way. Σε αυτή την περίπτωση αν έχουμε αναγνωρίσει ότι το στοιχείο way αποτελεί ένα δρόμο (highway), τότε αποθηκεύουμε σε μια τελική λίστα τους κόμβους ανά ζευγάρια. Στη λίστα αποθηκεύονται τα ζευγάρια μόνο μια φορά και με την αντίθετη φορά λόγω του ότι οι δρόμοι είναι διπλής κατεύθυνσης. Αυτό θα μας βοηθήσει αργότερα για να δημιουργήσουμε ένα κατευθυνόμενο γράφο. Τα ζευγάρια δεν αποθηκεύονται από την αντίθετη φορά μόνο όταν ο δρόμος είναι μονόδρομος (oneway). Στο τέλος της συνάρτησης μηδενίζουμε όλες της μεταβλητές που χρησιμοποιήσαμε.

endDocument

Τέλος, στη συνάρτηση endDocument που καλείται στο τέλος του επεξεργαζόμενου αρχείου, κλείνει τα αρχεία Nodes.txtκαι Links.txt αφού πρώτα αποθηκεύσει τη τελική λίστα με τις ακμές στο αρχείο Links.txt.

Με την εκτέλεση του εργαλείου OSMParser έχουμε ως έξοδο δύο αρχεία (Nodes.txtκαι Links.txt).

4.2 Χάρτης

Μεγάλο μέρος της εφαρμογής αποτελεί το κομμάτι του χάρτη που χρειάστηκε και το περισσότερο χρόνο για να υλοποιηθεί. Για την υλοποίηση του μέρους το υ χάρτη απαιτούνται τα εξής αρχεία:

MapActivity.java,	MBTileProvider.java*
DirectionsActivity.java,	MBTileSource.java*
PoiDetailsActivity.java,	PointData.java
BoundedMapView.java*	Nodes.txt
EdgeData.java	Links.txt

MBTileModuleProvider.java* Map.mbtiles

*Note: οι κλάσεις αυτές είναι μέρος ενός opensourcetutorial [27,28]

4.2.1 Δημιουργία MapView

Για την δημιουργία του χάρτη μόνο χρησιμοποιήσαμε ελάχιστες γραμμές κώδικα κώδικα. Η διαδικασία της δημιουργίας του χάρτη είναι σταθερή και το μόνο που χρειάστηκε να καθορίσουμε ήταν το μονοπάτι για τατiles και οι τρίτη και τέταρτη παράμετροι στην συνάρτηση XYTileSource(). Το μονοπάτι για τα tiles πρέπει να δείχνει στην κάρτα SD. Για να βεβαιωθούμε ότι το αρχείο βρίσκεται όντος στη κάρτα ελέγχουμε με την συνάρτηση exists αν υπάρχει. Σε περίπτωση που δεν υπάρχει (π.χ. πρώτη φορά που τρέχει η εφαρμογή) το μεταφέρουμε και μετά το χρησιμοποιούμε. Οι τρίτη και τέταρτη παράμετροι που καθορίζουμε στη συνάρτηση XYTileSource() αντιπροσωπεύουν το μικρότερο και μεγαλύτερο επίπεδο μεγέθυνσης που επιτρέπεται να έχει ο χάρτης.

Μέχρι αυτό το σημείο έχουμε ένα στατικό χάρτη που δεν κινείται ούτε χρησιμεύει σε κάτι. Για να του δώσο με περισσότερη λειτουργικότητα. Δημιουργούμε ένα mapcontrollerγια να μπορούμε να διαχειριζόμαστε το χάρτη και ένα locationlistener που θα ανακαλύπτει τη θέση του χρήστη. Βάζουμε το locationlistener να ενημερώνεται με τη θέση του χρήστη κάθε δευτερόλεπτο. Στη συνέχεια προσθέτουμε σαν ένα σημείο τη θέση του χρήστη στο χάρτη και κεντράρουμε το χάρτη σε ένα προκαθορισμένο σημείο με ένα προκαθορισμένο επίπεδο μεγέθυνσης

4.2.2Σημεία Ενδιαφέροντος

Έχοντας δημιουργήσει .ένα αλληλεπιδραστικό χάρτη που δείχνει τη θέση του χρήστη είναι ώρα να προσθέσουμε τα σημεία ενδιαφέροντος Για να μπορεί ο χρήστη να επιλέγει αυτά τα σημεία ώστε να βλέπει κάποιες πληροφορίες προεκτείναμε την κλάση ItemizedIconOverlay από την βιβλιοθήκη Osmdroid και φτιάξαμε την κλάση OverlayMarker.Δίνουμε στην κλάση ένα πίνακα με όλα τα στοιχεία των σημείων που χρειάζονται σαν OverlayItemαντικείμενα, τα οποία περιλαμβάνουν τη θέση του κάθε

σημείου, τον τίτλο και μια σύντομη περιγραφή. Για να παρουσιάσουμε στο χρήστη τις πληροφορίες υλοποιούμε τη συνάρτηση onSingleTapUpHelper η οποία δημιουργεί ένα παράθυρο διαλόγου (AlterDialog) με τίτλο το όνομα του σημείου, περιεχόμενο τη σύντομη περιγραφή και ένα κουμπί MORE. Όταν ο χρήστης πατήσει το κουμπί MORE καλείται το PoiDetailsActivity. Για να καταλάβει το activityγια πιο σημείο θέλουμε περισσότερες πληροφορίες περνάμε την ταυτότητα του σημείου (τον τίτλο του), με τη χρήση της κλάσης Intentκαι τις συνάρτησης putextra.

Για να εμφανιστούν τα σημεία πάνω στο χάρτη καλούμε τη συνάρτηση addPoi. Ευθύνη της συνάρτησης είναι να δημιουργήσει όλα τα σημεία ενδιαφέροντος ως OverlayItem, να τα βάλει σε ένα πίνακα και να δημιουργήσει ένα αντικείμενο OverlayMarkerμε τον τρόπο που περιγράψαμε πιο πάνω. Όταν έχει δημιουργηθεί το OverlayMarker αντικείμενο το προσθέτουμε στο χάρτη.

4.2.3 Οδηγίες από σημείο σε σημείο

Ένα άλλο στοιχείο που έχει το κομμάτι του χάρτη είναι η δυνατότητα να μπορεί να καθοδηγήσει το χρήστη από ένα σημείο σε ένα άλλο βρίσκοντας το πιο σύντομο μονοπάτι και εμφανίζοντας το στο χάρτη. Εδώ είναι το σημείο όπου χρησιμοποιούμε το γράφο (κατευθυνόμενο με βάρη). Ο γράφος αρχικοποιείται και γεμίζεται από τις συναρτήσεις addVerticesκαι addEdges. Παράλληλα με τη χρήση του γράφου γίνεται χρήση δύο βοηθητικών λιστών pointsκαι edges.Σκοπός τους είναι να κρατούν κάποια δεδομένα για τους κόμβους και τις ακμές που δεν μπορούμε να έχουμε στο γράφο.

Η λίστα pointsκρατάει αντικείμενα τύπου PointData. Στην κλάση PointData κρατάμε απλός την ταυτότητα του κόμβου μαζί με τη γεωγραφική του θέση.

Η λίστα edges κρατάει αντικείμενα τύπου EdgesData. Η κλάσηEdgesDataείναι πιο σύνθετη από την PointData. Τα δεδομένα που κρατάει είναι οι ταυτότητες των δύο κόμβων που αποτελούν την ακμή, και τις παραμέτρους που μας δίνουν την ευθεία που ανήκει η ακμή. Συγκεκριμένα παίρνουμε την εξίσωση ευθείας δύο σημείων και τη φέρνουμε στην μορφή $\alpha x + \beta y + \gamma = 0$. Από την εικόνα 4.1παρατηρούμε ότι μόνο ο συντελεστής του x και η μεταβλητή γ αλλάζουν με βάση τις συντεταγμένες των

σημείων ενώ ο συντελεστής του β παραμένει σταθερός (-1). Ως συνέπεια τα δεδομένα της εξίσωσης που κρατάμε είναι ο συντελεστής του χ και η μεταβλητή γ οι οποίες υπολογίζονται κατά την δημιουργία του αντικειμένου.

$$\frac{\psi - \psi_1}{\psi_2 - \psi_1} = \frac{x - x_1}{x_2 - x_1} \Leftrightarrow \frac{\psi_2 - \psi_1}{x_2 - x_1} x - \psi + \left(\psi_1 - \frac{\psi_2 - \psi_1}{x_2 - x_1} \cdot\right) x_1 = 0$$

Εικόνα 4.1: Εξίσωση ευθείας δύο σημείων

addVertices

Η συνάρτηση addVerticesδιαβάζει το αρχείο Nodes.txtκαι προσθέτει την ταυτότητα των κόμβων κε τις γεωγραφικές τους συντεταγμένες στον πίνακα points και στο γράφο με τη συνάρτηση addVertex του γράφου προσθέτει σε αυτόν τον κόμβο. Στο τέλος της συνάρτησης προσθέτουμε ακόμα ένα κόμβο με ταυτότητα mylocation ο οποίος αναπαριστά τη θέση του χρήστη στο γράφο.

addEdges

Η συνάρτηση addEdges διαβάζει το αρχείο Links.txt. και για κάθε ζευγάρι κόμβων που διαβάζει, δημιουργεί μια ακμή με τις δύο ταυτότητες και θέτει ως το βάρος της ακμής την γεωγραφική απόσταση των δύο σημείων. Η απόσταση των σημείων υπολογίζεται από την Haversineformula όπως φαίνεται στην εικόνα 4.2.Ακολούθως υπολογίζονται οι μεταβλητές α και γ και προστίθεται η ακμή στον πίνακα edges.

$$\Delta\text{lat} = \text{lat}_2 - \text{lat}_1$$

$$\Delta\text{long} = \text{long}_2 - \text{long}_1$$

$$a = \sin^2(\Delta\text{lat}/2) + \cos(\text{lat}_1).\cos(\text{lat}_2).\sin^2(\Delta\text{long}/2)$$

$$c = 2a\tan 2[\sqrt{a}, \sqrt{(1-a)}]$$

then

$$d = Rc.$$

Εικόνα 4.2: Haversineformula

Εύρεση κοντινότερου μονοπατιού

Όταν ο χρήστης έχει δώσει τα σημεία για τα οποία θέλει να πάρει οδηγίες χρησιμοποιούμε τη συνάρτηση addPath για να υπολογίσουμε το κοντινότερο μονοπάτι και να το εμφανίσουμε στο χάρτη. Υπάρχουν δύο περιπτώσεις που πρεπεί να ελέγξουμε. Αν θέλει να πάει από την τοποθεσία που βρίσκεται ή από ένα προκαθορισμένο σημείο. Αν επιλέξει προκαθορισμένα σημεία τότε απλά τον αλγόριθμο Dijkstra και αφού μας επιστρέψει την διαδρομή τη δείχνουμε στο χάρτη με τη χρίση μιας μεταβλητής τύπου Path.

Στην περίπτωση που ο χρήστης επιθυμεί να λάβει οδηγίες από την τοποθεσία του πρέπει να ελέγξουμε τρις περιπτώσεις. Η πρώτη αν το GPSέχει μια θέση για τον χρήστη. Σε περίπτωση που δεν έχει εμφανίζει το κατάλληλο μήνυμα. Η δεύτερη περίπτωση είναι αν ο χρήστης είναι μέσα στην ορατή περιοχή του χάρτη. Πάλι, στη περίπτωση που δεν είναι εμφανίζεται αντίστοιχο μήνυμα. Αν καμιά από τις δύο πρώτες περιπτώσεις ισχύουν τότε μπορούμε να προχωρήσουμε να υπολογίσουμε το μονοπάτι.

Το πρώτο βήμα που κάνουμε είναι να ενημερώσουμε τη θέση του χρήστη στον πίνακα points. Βρίσκουμε τον κόμβο με την ταυτότητα mylocation και θέτουμε το γεωγραφικό μήκος και πλάτος με αυτά που παίρνουμε από το locationmanager.

Στη συνέχεια ψάχνουμε στο γράφο να δούμε αν υπάρχουν ακμές που ένα από τους κόμβους έχει ταυτότητα mylocation. Αν βρούμε τους αφαιρούμε για να εισάγουμε αργότερα καινούργιους.

Για να βρούμε το κοντινότερο μονοπάτι από τον χρήστη πρέπει ο κόμβος του χρήστη να ενωθεί με τον υπόλοιπο γράφο μέσο ακμών. Οι ακμές αυτές είναι πάντα δύο και έχουν ως κόμβους τους τη θέση του χρήστη και τους κόμβους της πιο κοντινής ακμής προς τη θέση του χρήστη.

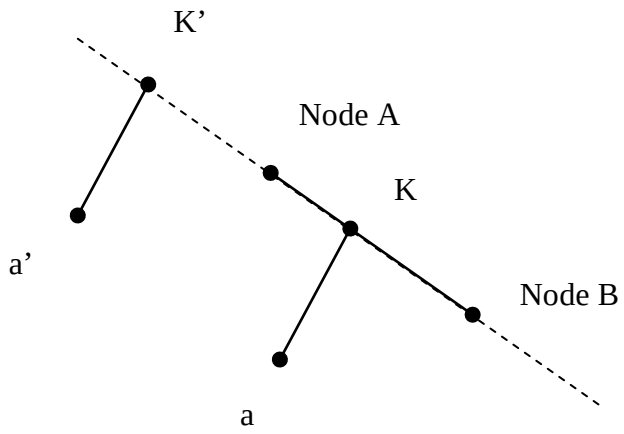
Την πιο κοντινή ακμή θα τη βρούμε με το διασχίσουμε όλες τις ακμές του γράφου βρίσκοντας την απόσταση τους με την τοποθεσία του χρήστη. Για τον υπολογισμό της απόσταση θα πρέπει να αντικαταστήσουμε τη μεταβλητή χ με το γεωγραφικό μήκος, τη μεταβλητή ψ με το γεωγραφικό πλάτος, και τις μεταβλητές α, β και γ με τις μεταβλητές

της εγθείας που παίρνουμε για κάθε ακμή από τον πίνακα edgesστην εξίσωση της εικόνας 4.3.

$$d = \frac{|A\chi_1 + B\psi_1 + \Gamma|}{\sqrt{A^2 + B^2}}$$

Εικόνα 4.3: Απόσταση σημείου από ευθεία

Πρέπει να λάβουμε υπόψη ότι οι ακμές είναι ευθύγραμμα τμήματα με αρχή και τέλος. Αν και ο πιο πάνω τύπος για τον υπολογισμό της απόστασης μας πει ότι ένα σημείο είναι πιο κοντά στην ακμήαυτό δεν ισχύει κατά ανάγκη.Στην εικόνα 4.4βλέπουμε μια τέτοια περίπτωση. Ο τύπος της εικόνας 4.3 θα μας πει ότι τα σημεία a και a’ έχουν την ίδια απόσταση ενώ στην πραγματικότητα το σημείο a δεν είναι.



Εικόνα 4.4: Ακρίβεια απόστασης σημείου ευθείας

Για να αποφύγουμε αυτό το λάθος πρέπει να βρούμε το σημείο K. Το σημείου δηλ. η κάθετος από το σημείο ατέμνει την ευθεία αν το σημείο K βρίσκεται πάνω στο ευθύγραμμο τμήμα της ακμής τότε δεχόμαστε την απόσταση.

$$\lambda_1 \cdot \lambda_2 = -1$$

Εικόνα 4.5: Κλίσεις κάθετων ευθειών

Θα χρησιμοποιήσουμε το τύπο της εικόνας 4.5 για να βρούμε την κλίση της ευθείας aK. Η κλίση της ακμής είναι ο συντελεστής το υ χ. Αφο ύ βρούμε την κλίση aK χρησιμοποιούμε την εξίσωση της ευθείας αχ +βψ +γ = 0 και το σημείο aγια να βρούμε τη μεταβλητή γ. έτσι έχουμε βρει και την εξίσωση τηςευθείας aK. Η λύση του συστήματος των δύο ευθειών θα μας δώσει το σημείο K.

$$P(t) = P_0 + t(P_1 - P_0) = \begin{cases} x(t) = x_0 + t(x_1 - x_0) \\ y(t) = y_0 + t(y_1 - y_0) \end{cases}$$

Εικόνα 4.6: Εξίσωση ευθείας

Η εξίσωση της εικόνας 4 6 υπολογίζει τις συντεταγμένες ενός σημείουPόταν έχουμε δύο σημεία της ευθείας. Ο συντελεστής τείναι μικρότερος από 0 αν το σημείο Pείναι πριν το σημείο P0 της ευθείας, 0 αν είναι το σημείο P0, κυμαίνεται μεταξύ 0και 1 αν το σημείο Pείναι μεταξύ των δύο σημείων της ευθείας, 1 αν είναι το σημείο P1 και μεγαλύτερο από 1 αν είναι μετά το σημείο P1.

Αυτό μπορούμε να το χρησιμοποιήσουμε για να δο ύμε αν το τγια το χ και ψ το υ σημείου Kείναι μεταξύ του 0 και 1. Αν το K είναι μεταξύ του 0και 1 και η απόσταση είναι μικρότερη από την προηγούμενη τότε έχο ψμε βρει μια πιο κοντινή ακμή και κρατάμε τους κόμβους που την αποτελούν.

Μόλις βρο ύμε την πιο κοντινή ακμή τότε δημιουργούμε δύο καινούργιες ακμές με τη θέση του χρήστη με το ν ίδιο τρόπο που αναφέραμε πιο πριν. Στη συνέχεια χρησιμοποιούμε τον αλγόριθμο Dijkstra για να βρούμε το μονοπάτι και να το δείξουμε στο χάρτη.

4.2.4 Μέθοδοι ελέγχου

Στην υλοποίηση το υ χάρτη ήταν απαραίτητη η δημιουργία μεθό δων που θα διασφάλιζαν ότι στη συσκευή υπάρχει ενεργή η χρήση του GPS. Για να μπορέσουμε να βεβαιωθούμε ότι το GPS είναι ενεργοποιημένο, ελέγχουμε από το locationmanager με τη συνάρτηση isProviderEnable αν υπάρχει παροχέας ενεργός. Στη περίπτωση που δεν υπάρχει ρωτάμε τον χρήστη αν επιθυμεί να ενεργοποίηση την υπηρεσία. Σε θετική απάντηση μεταφερόμαστε στις ρυθμίσεις για την ενεργοποίηση. Σε αρνητική απάντηση τερματίζουμε την εφαρμογή. Αν ο χρήστης επιλέξει να ενεργοποιήσει την υπηρεσία αλλά δεν το κάνει, επιστρέφοντας στην εφαρμογή, με την συνάρτηση onActivityResult ελέγχουμε ξανά και επαναλαμβάνουμε την διαδικασία.

4.3. AugmentedReality

Το δεύτερο μέρος της Διπλωματικής Εργασίας είναι το AugmentedReality.Για την υλοποίηση αυτού του μέρους χρειάζονται τα εξής αρχεία:
ARBrowserActivity.java
PoiBean.java
ARWorld.html
ARControls.js

4.3.1ΑρχικοποίησηARWorld και δεδομένων σημείων

Για να δημιουργήσουμε τον επαυξημένος κόσμο το μόνο που χρειάζεται να κάνουμε είναι να καλέσουμε των κατασκευαστή (constructor), πο υ μας δύνεται από την βιβλιοθήκη του Wikitide. Ο κόσμος που έχουμε δημιουργήσει όμως δεν είναι τίποτα από μια απλή οθόνη κάμερας. Θα πρέπει να χρησιμοποιήσουμε μία από τις πιο βασικές συναρτήσεις του ARBrowserActivity αρχείου, την loadWorld. Το πρώτο πράγμα που κάνουμε είναι να φορτώσουμε ένα htmlαρχείο το οποίο θα περιγράφει τι θα βλέπουμε στην οθόνη της κάμερας πέρα από τα σημεία ενδιαφέροντος.

Για την δική μας εφαρμογή φορτώνουμε το αρχείοARWorld.html. Σε αυτό το αρχείο έχουμε βάλει ένα footer που θα εμφανίζεται στο κάτω μέρος της οθόνης. Το χώρο που έχουμε δώσει στο footer το έχουμε χωρίσει σε 3 άλλα μέρη. Το πρώτο είναι ένα header στο οποίο θα φαίνεται ο τίτλος του επιλεγμένου σημείου. Το δεύτερο είναι μια παράγραφος στην οποία θα μπαίνει η σύντομη περιγραφή του σημείο και το τρίτο μέρος είναι ακόμα μια παράγραφος.Στην δεύτερη παράγραφο θα τοποθετούμεπάντα τις λέξεις "Readmore…" έτσι ώστε οχρήστης να καταλαβαίνει ότι πατώντας στο footerθα μπορεί να δει περισσότερες πληροφορίες για το σημείο.

Εκτός από το να καθορίσουμε πώς θα φαίνεται η διεπαφή του ARWorld θα πρέπει να δώσουμε στον επαυξημένο κόσμο τις πληροφορίες για τα σημεία. Οι πληροφορίες των σημείων θα αποθηκεύονται σε ένα πίνακα τύπου JavaScriptObject. Πριν μετατρέψουμε τα δεδομένα σε JavaScriptObject χρησιμοποιούμε την κλάση PoiBean για να αποθηκεύσουμε το όνομα, την περιγραφή και την τοποθεσία του σημείου. Μετά τα μετατρέπουμε σε JSONαντικείμενα και τα βάζουμε σε ένα πίνακα. Για να μεταφέρουμε αυτά τα δεδομένα στο JavaScriptαρχείο για να τα διαχειριστούμε πρέπει να μετατρέψουμε τον πίνακα JSON σε μια συμβολοσειρά. Με τη callJavascript τα στέλλουμε τα δεδομένα μας στη συνάρτηση newData του JsvaScriptαρχείου.

Η δεύτερη σημαντική συνάρτηση στο ARBrowserActivity αρχείο είναι η συνάρτηση urlWasInvoked. Στο κεφάλαιο3 τμήμα 3.4αναφέραμε ότι το JavaScriptαρχείο θα μπορεί να επικοινωνεί με το ARBrowserActivity. Αυτό επιτυγχάνεται με τη συνάρτηση urlWasInvoked. με την κλήση της, δέχεται ως παράμετρο τη ταυτότητα ενός σημείου. Αυτή με την σειρά της καλεί την κλάση PoiDetailsActivityγια να δείξει στο χρήστη περισσότερες πληροφορίες.

4.3.2 Διαχείριση Σημείων ενδιαφέροντος

Τα σημεία ενδιαφέροντος διαχειρίζονται εξολοκλήρου με τη χρήση JavaScript. Τα δεδομένα των σημείων μεταφέρονται από το περιβάλλον της Javaστο περιβάλλον του JavaScript με τη συνάρτηση newData.

Αφού λάβει τα δεδομένα, τα μετατρέπει σε ένα JSON πίνακα. Για κάθε ένα από τα σημεία, αναθέτει την εικόνα του σημείου και εγγράφει την μέθοδο onClickAnimation που δημιουργεί ένα μικρό animation (μια μικρή μεγέθυνση) όταν ο χρήστης το επιλέξει.

Εκτός από την μέθοδο onClickAnimation, για κάθε σημείο εγγράφεται και η συνάρτηση createClickTrogger. Ο ρόλος της createClickTrogger είναι κάθε φορά που ο χρήστη επιλέγει την εικόνα ενός σημείου να εμφανίζει στο footertις πληροφορίες του σημείου. Αν η εικόνα το σημείου κρύβεται πίσω από άλλα σημεία γιατί είναι πιο μακριά τότε τη φέρνει μπροστά από όλες τις άλλες.

Τα αντίθετα αποτελέσματα έχει η onScreenClickσυνάρτηση. Η onScreenClickκαλείται κάθε φορά που ο χρήστης πατήσει τον άδειο χώρο της οθόνης. Όπως είναι αναμενόμενο τα περιεχόμενα του σημείου που βρίσκονταν στοfooter φεύγουν και η εικόνα του σημείου παίρνει τη κανονική τους θέση.

4.4.Παρουσίαση πληροφοριών Σημείων Ενδιαφέροντος

Ένα activityπου χρησιμοποιείται και από τα δύο μέρη της εφαρμογής είναι το PoiDetailsActivity, δηλ το activity που παρουσιάζει περισσότερες πληροφορίες για τα μνημεία. Το PoiDetailsActivity παίρνει δεδομένα εισόδου και από το ARBrowserActivity και από το MapActivity.Και στις δύο περιπτώσεις το PoiDetailsActivity δέχεται σαν είσοδο την ταυτότητα του σημείου (όνομα).

Με βάση το όνομα του σημείου φορτώνεται και οι ανάλογες πληροφορίες του σημείου. Στο πάνω μέρος είναι με μεγάλα γράμματα το όνομα του σημείου. Από κάτω βρίσκεται ένα imagegalleryπου προβάλλει φωτογραφίες. Από κάτω είναι ένα textview που περιέχει μια πιο λεπτομερή περιγραφή του σημείου.

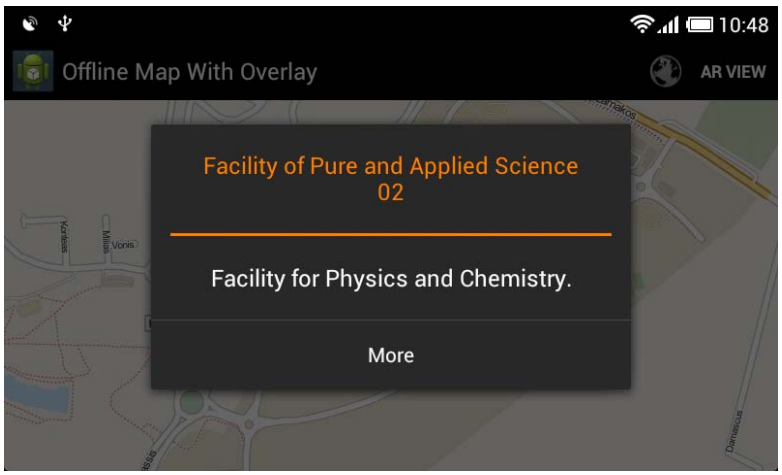
Κεφάλαιο 5

Αποτελέσματα

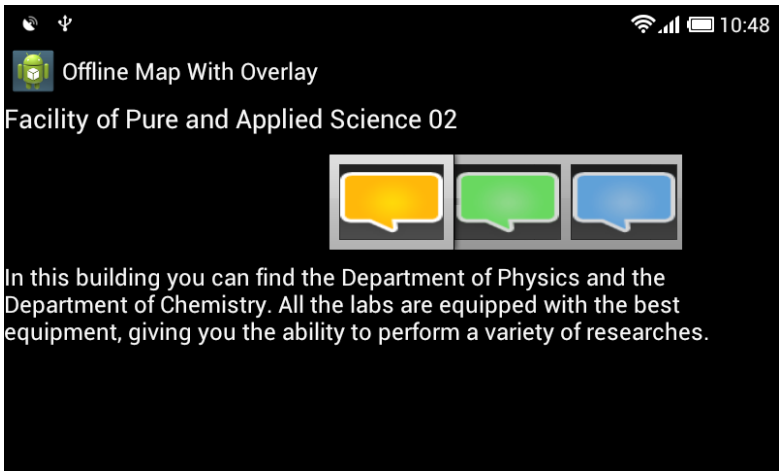
5.1 Αποτελέσματα

Η εφαρμογή έχει ολοκληρωθεί και συμπεριλαμβάνει τις βασικές απαιτήσεις ενόςAugmentedReality τουριστικού ξεναγού. Δίνει στο χρήστη τη δυνατότητα να βρεί τα σημαντικά σημεία γύρο του, να του δίνει πληροφορίες για αυτά και με την κάμερα να μπορεί να δει που γύρο του βρίσκονται.

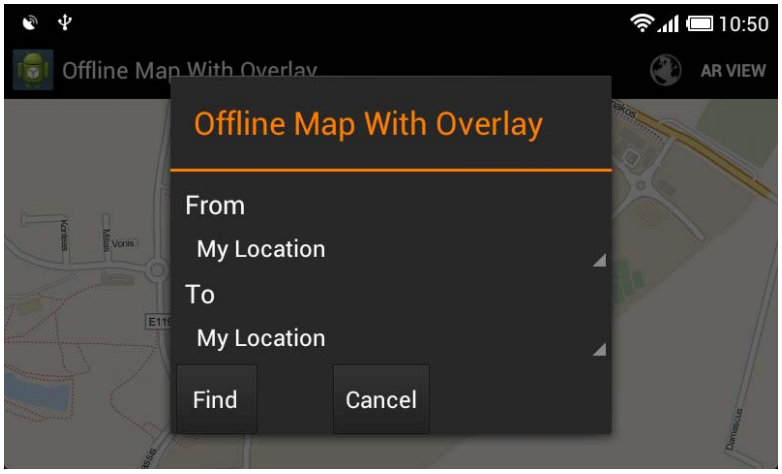
Πιο κάτω φαίνονται κάποιες εικόνες από την εφαρμογή.



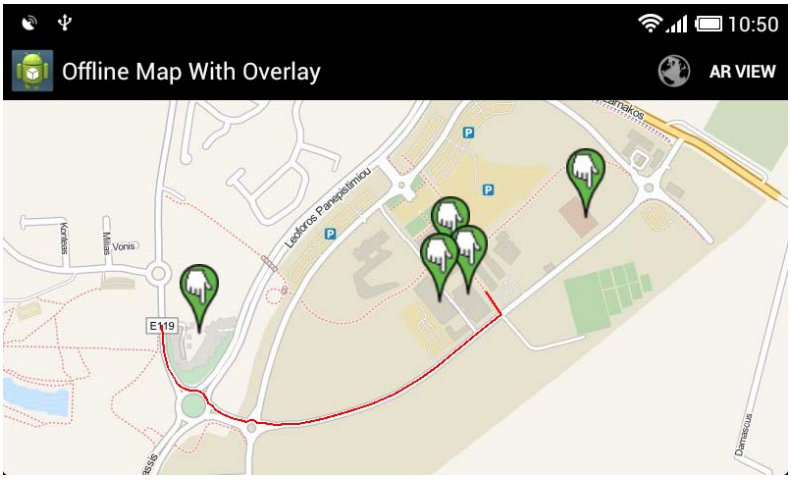
Εικόνα 5.1: Σύντομη περιγραφή σημείου



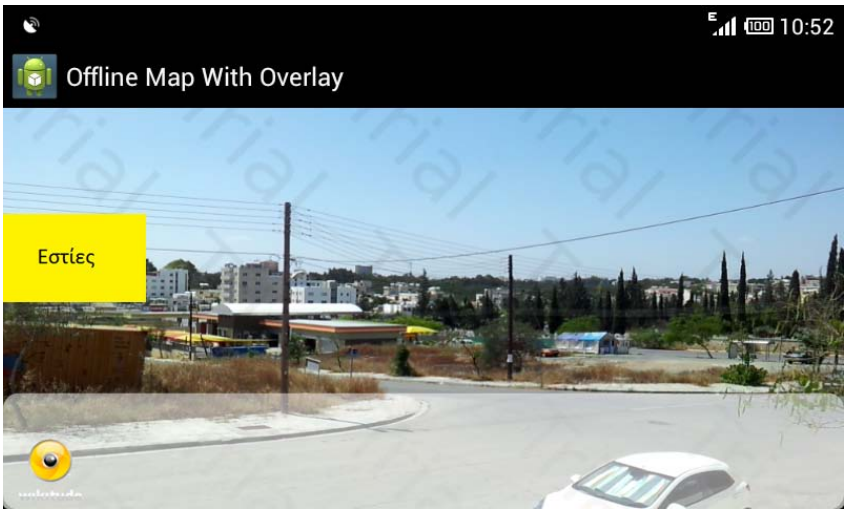
Εικόνα 5.2: Παρουσίαση περισσότερων πληροφοριών για ένα σημείο



Εικόνα 5.3: Φόρμα επιλογής οδηγιών από σημείο σε σημείο



Εικόνα 5.3: Το συντομότερο μονοπάτι που μπορεί να ακολουθήσει ο χρήστης.



Εικόνα 5.4: Πώς φαίνονται τα σημαίας στο ARView

Το μόνο που δεν έχει υλοποιηθεί μέχρι στιγμής είναι η ενσωμάτωση δεδομένων από αρχαιολογικούς χώρους.

5.2 Μελλοντική Εργασία

Παρατηρώντας τα χαρακτηρίστηκα παρόμοιων εφαρμογών θα μπορούσαμε να πούμε ότι σαν μελλοντικές εργασίες μπορούν να γίνουν τα εξής:

- Χρήση τρισδιάστατων μοντέλων ή κινούμενων εικόνων.
- Χρήση φωνής για να δίνει περισσότερες πληροφορίες και όχι μόνο κείμενο.
- Επέκταση για περισσότερο από ένα αρχαιολογικούς χώρους.
- Χρήση πολλών γλωσσών στην εφαρμογή.
- Βελτίωση της εμφάνισης των σύντομων περιγραφών.

Βιβλιογραφία

[1] Schall, Gerhard, Schöning, Johannes, “Asurveyonaugmentedmapsandenvironments: Approaches, interactions and applications”, Taylor and Francis Group, Pages: 207-226, 2011

[2] George Papagiannakis, Gurminder Singh, Nadia Magnenat-Thalmann, “A survey of mobile and wireless technologies for augmented reality systems”, Wiley Online Library, Pages: 3-22, 2008

[3] A.Y.C.Nee,S.K. Ong, G.Chryssolouris, D. Mourtzis, “Augmented reality applications in design and manufacturing”, CIRP Annals - Manufacturing Technology, 2012

[4] Tobias H Höllerer, Steven K Feiner, “Mobile Augmented Reality”, Taylor & Francis Books Ltd., Pages: 1-39, 2004

[5] Ronald T Azuma, “A Survey of Augmented Reality”, Citeseer, Pages: 355-385, 1997

[6] D W F Van Krevelen, R Poelman, “A Survey of Augmented Reality Technologies , Applications and Limitations”, International Journal, Volume: 9, Issue: 2, Pages: 1–20, 2010

[7] Paul Milgram, Haruo Takemura, Akira Utsumi, Fumio Kishio, “Augmented Reality: A class of displays on the reality-virtuality cotinuum”, SPIE, 2351:282-292, 1994

[8] Daniel Wagner, DieterScmalstieg, “Making augmented reality practical on mobile phones, part 1”, IEEE Comput. Graph.. Appl. , 29(3):12-15, 2009

[9] Daniel Wagner, DieterScmalstieg, “Making augmented reality practical on mobile phones, part 3”, IEEE Comput. Graph.. Appl. , 29(4):6-9, 2009

[10] “Computer graphics”, Wikipedia, http://en.wikipedia.org/wiki/Computer_graphics

[11] “What is computer Graphics? Explain Interactive and Non-interactive”, Computer Notes

[12] “Augmented reality”, Wikipedia http://en.wikipedia.org/wiki/Augmented_reality

[13] “Android, the world's most popular mobile platform”, Google Inc http://developer.android.com/about/index.html

[14] Dan Bornstein, “Dalvik VM internals.” Presentation, GoogleInc. May 2008, http://sites.google.com/site/io/dalvuk-vm-internals

[15] Patrick Brady, “Anatomy & physiology of an android”, Presentation, Google Inc, May 2008 http://sites.google.com/site/io/anatomy--physiology-of-an-android

[16] Mobile Atlas Creator http://mobac.sourceforge.net/

[17] Open Street Map http://www.openstreetmap.org/

[18] Latitude and Longitude of a Point
http://itouchmap.com/latlong.html

[19] Elevation Finder
http://www.freemaptools.com/elevation-finder.htm

[20] Eclipse
http://www.eclipse.org/

[21] Android ASD
http://developer.android.com/sdk/index.html

[22] Osmdroid
https://code.google.com/p/osmdroid/

[23] Simple Logging Facade for Java
http://www.slf4j.org/

[24] JGraphT
http://jgrapht.org/

[25] Wikitude
http://www.wikitude.com/

[27] Using MBTiles in osmdroid
http://www.sieswerda.net/2012/08/15/upping-the-developer-friendliness/

[28] BoundedMapView – A MapView with limits

http://www.sieswerda.net/2012/08/15/boundedmapview-a-mapview-with-limits/

ΠαράρτημαΑ

OSMParser

Main.java

```
importjava.io.IOException;
importjavax.xml.parsers.ParserConfigurationException;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import org.xml.sax.SAXException;
import org.xml.sax.XMLReader;


publicclass Main {

	publicstaticvoid main(String[] args) {

		OsmHandler osmHandler = null;


		try {
			/* 'standard' SAX way to parse xml */
			// create the factory
			SAXParserFactory factory =
SAXParserFactory.newInstance();

			// create a parser
			SAXParser parser = factory.newSAXParser();

			// create the reader (scanner)
			XMLReader xmlreader = parser.getXMLReader();

			/* end 'standard' SAX way to parse xml */

			// instantiate our handler
			osmHandler = new OsmHandler();

			// assign our handler
			xmlreader.setContentHandler(osmHandler);

			System.out.println("Parsing has started!\nPlease
wait...");

			// XMLReader parse() method
```

```
                    xmlreader.parse(args[0]);

                    System.out.println("Parsing has been completed!");

               } catch (ParserConfigurationException e) {
                    e.printStackTrace();
               } catch (IOException e) {
                    e.printStackTrace();
               } catch (SAXException e) {
                    e.printStackTrace();
               }

          }

}
```

OsmHandler.java

```
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;

import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

publicclass OsmHandler extends DefaultHandler {

	private BufferedWriter bwNodes = null;
	private BufferedWriter bwLinks = null;
	privatebooleanway = false;
	privatebooleanoneway = false;
	privatebooleanhighway = false;
	private ArrayList<String>tempLinks = null;
	private ArrayList<String>finalLinks = new ArrayList<String>();

	public OsmHandler() {

		super();

	}

	publicvoid startDocument() {

		// Opens Nodes.txt file to write the nodes
		try {
			bwNodes = new BufferedWriter(new FileWriter(new
File("Nodes.txt")));
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

		// Opens Links.txt file to write the links
		try {
			bwLinks = new BufferedWriter(new FileWriter(new
File("Links.txt")));
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

	}

	publicvoid endDocument() {

		for (int i = 0; i <finalLinks.size(); i++) {
			try {
				bwLinks.write(finalLinks.get(i) + "\n");
			} catch (IOException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
```

```
                }
        }

        try {
                bwNodes.close();
        } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }

        try {
                bwLinks.close();
        } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }

}

        publicvoid startElement(String uri, String name, String qName,
Attributes atts) throws SAXException {

        // Starting node tag was found
        if (qName.equals("node")) {
                try {
                        bwNodes.write(atts.getValue("id") + " " +
atts.getValue("lat") + " " + atts.getValue("lon") + "\n");
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
        }

        // Starting way tag was found
        if (qName.equals("way")) {
                way = true;
                tempLinks = new ArrayList<String>();
        }

        // Starting nd tag was found
        if (qName.equals("nd")) {
                tempLinks.add(atts.getValue("ref"));
        }

        // Starting tag tag was found
        if (qName.equals("tag") &&way == true) {

                // Oneway == yes
                if (atts.getValue("k").equals("oneway") &&
atts.getValue("v").equals("yes")) {
                        oneway = true;

                }

                // highway == yes
                if (atts.getValue("k").equals("highway")) {
                        highway = true;
                }
```

```
			}

		}

		publicvoid endElement(String uri, String name, String qName) {

			// Check if end element is way and is about highway
			if (qName.equals("way") &&highway == true) {
				String tempstr = null;

				for (int i = 0; i <tempLinks.size() - 1; i++) {

					tempstr = tempLinks.get(i) + " " + tempLinks.get(i
+ 1);
					if (finalLinks.indexOf(tempstr) == -1) {
						finalLinks.add(tempstr);
					}

					tempstr = tempLinks.get(i + 1) + " " +
tempLinks.get(i);
					if (oneway == false&&finalLinks.indexOf(tempstr) ==
-1) {

						finalLinks.add(tempLinks.get(i + 1) + " " +
tempLinks.get(i));
					}

				}

				way = false;
				oneway = false;
				highway = false;
				tempLinks = null;
			}

		}

}
```

ΠαράρτημαΒ

Pocket Museum

ARBrowserActivity.java

```
import java.io.IOException;
import java.net.URI;
import java.util.ArrayList;
import java.util.List;

import org.apache.http.NameValuePair;
import org.apache.http.client.utils.URLEncodedUtils;
import org.json.JSONArray;
import com.wikitude.architect.ArchitectUrlListener;
import com.wikitude.architect.ArchitectView;

import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.media.AudioManager;
import android.os.Bundle;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;

import android.content.res.Configuration;
```

```
public class ArBrowserActivity extends Activity implements ArchitectUrlListener, LocationListener {

	private ArchitectView architectView;
	private LocationManager locManager;
	private Location loc;
	private List<PoiBean> poiBeanList;

	@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);

		// in order to inform the ARchitect framework about the user's location
		// Androids LocationManager is used in this case
		locManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

		// check if the device fulfills the SDK'S minimum requirements
		if (!ArchitectView.isDeviceSupported(this)) {
			Toast.makeText(this, "Minimum requirements not fulfilled", Toast.LENGTH_LONG).show();
			this.finish();
			return;
		}
		setContentView(R.layout.activity_ar_broswer);

		// set the devices' volume control to music to be able to change the
		// volume of possible sound files to play
		this.setVolumeControlStream(AudioManager.STREAM_MUSIC);
		this.architectView = (ArchitectView) this.findViewById(R.id.architectView);
		// onCreate method for setting the license key for the SDK
		architectView.onCreate("1234");
```

```
        locManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0,
this);

        }

        @Override
        protected void onPostCreate(Bundle savedInstanceState) {
                super.onPostCreate(savedInstanceState);

                // IMPORTANT: creates ARchitect core modules
                if (this.architectView != null)
                        this.architectView.onPostCreate();

                // register this activity as handler of "architectsdk://" urls
                this.architectView.registerUrlListener(this);

                try {
                        loadWorld();
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }

        }

        @Override
        protected void onResume() {
                super.onResume();

                this.architectView.onResume();
                // get last known location from LocationManager
```

```
                loc                                                    =
locManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
                if (loc != null) {
                        this.architectView.setLocation((float)  (loc.getLatitude()),  (float)
(loc.getLongitude()), loc.getAccuracy(), loc.getTime());
                }

        }

        @Override
        protected void onPause() {
                super.onPause();
                if (this.architectView != null)
                        this.architectView.onPause();
        }

        @Override
        protected void onDestroy() {
                super.onDestroy();

                if (this.architectView != null)
                        this.architectView.onDestroy();
        }

        @Override
        public void onLowMemory() {
                super.onLowMemory();

                if (this.architectView != null)
                        this.architectView.onLowMemory();
        }

        /**
```

```
         * loads a sample architect world and creates a definable amount of pois in
         * beancontainers and converts them into a jsonstring that can be sent to
         * the framework
         *
         * @throws IOException
         *             exception thrown while loading an Architect world
         */
        private void loadWorld() throws IOException {
                this.architectView.load("ARWorld.html");

                JSONArray array = new JSONArray();
                poiBeanList = new ArrayList<PoiBean>();

                try {
                        PoiBean bean = new
PoiBean(getResources().getString(R.string.esties),

getResources().getString(R.string.esties_short_discription),

35.143608153115736,

33.40523958206177,
                                                                149.950);
                        array.put(bean.toJSONObject());
                        poiBeanList.add(bean);
                        PoiBean bean1 = new
PoiBean(getResources().getString(R.string.thee02),

        getResources().getString(R.string.thee02_short_discription),

        35.144459139006344,

        33.410303592681885,
```

```
                                                                    135.26);
                    array.put(bean1.toJSONObject());
                    poiBeanList.add(bean1);
                    PoiBean bean2 = new
PoiBean(getResources().getString(R.string.thee01),

    getResources().getString(R.string.thee01_short_discription),

    35.14461705293515,

    33.41095805168152,
                                                                    134.328);
                    array.put(bean2.toJSONObject());
                    poiBeanList.add(bean2);
                    PoiBean bean3 = new
PoiBean(getResources().getString(R.string.xod01),

    getResources().getString(R.string.xod01_short_discription),

    35.145117111688315,

    33.410550355911255,
                                                                    137.743);
                    array.put(bean3.toJSONObject());
                    poiBeanList.add(bean3);
                    PoiBean bean4 = new
PoiBean(getResources().getString(R.string.sports),

    getResources().getString(R.string.sports_short_discription),

    35.14549434696019,

    33.413758277893066,
```

```
                                                                    133.481);
                        array.put(bean4.toJSONObject());
                        poiBeanList.add(bean4);
                } catch (Exception e) {// Catch exception if any
                        System.err.println("Error: " + e.getMessage());
                }
                this.architectView.callJavascript("newData('" + array.toString() + "');");
        }

        /**
         * listener method called when the location of the user has changed used for
         * informing the ArchitectView about a new location of the user
         */

        public void onLocationChanged(Location loc) {
                // IMPORTANT:
                // use this method for informing the SDK about a location change by the
                // user
                // for simplicity not used in this example

                // inform ArchitectView about location changes
                if (this.architectView != null)
                        this.architectView.setLocation((float)  (loc.getLatitude()),  (float)
(loc.getLongitude()), loc.getAccuracy());

        }

        public void onProviderDisabled(String provider) {
                // TODO Auto-generated method stub

        }

        public void onProviderEnabled(String provider) {
```

```
            // TODO Auto-generated method stub

        }

        public void onStatusChanged(String provider, int status, Bundle extras) {
            // TODO Auto-generated method stub

        }

        /**
         * <p>
         * interface method of {@link ArchitectUrlListener} class called when an url
         * with host "architectsdk://" is discovered
         *
         * can be parsed and allows to react to events triggered in the ARchitect
         * World
         * </p>
         */
        public boolean urlWasInvoked(String url) {
            // parsing the retrieved url string
            List<NameValuePair> queryParams =
URLEncodedUtils.parse(URI.create(url), "UTF-8");

            String id = "";
            // getting the values of the contained GET-parameters
            for (NameValuePair pair : queryParams) {
                if (pair.getName().equals("id")) {
                    id = pair.getValue();
                }
            }

            // start a new intent for displaying the content of the bean
            Intent intent = new Intent(this, PoiDetailsActivity.class);
```

```
                intent.putExtra("id", id);
                this.startActivity(intent);
                return true;

        }

        @Override
        public void onConfigurationChanged(Configuration newConfig) {
                super.onConfigurationChanged(newConfig);
        }

}
```

BoundedMapView.java

```
import microsoft.mappoint.TileSystem;

import org.osmdroid.ResourceProxy;
import org.osmdroid.events.ScrollEvent;
import org.osmdroid.tileprovider.MapTileProviderBase;
import org.osmdroid.util.BoundingBoxE6;
import org.osmdroid.views.MapView;
import org.osmdroid.views.util.constants.MapViewConstants;

import android.content.Context;
import android.graphics.Point;
import android.graphics.Rect;
import android.widget.Scroller;

/**
 * Extension of MapView that limits scrolling to the area specified. Based on
 * code from MarcKurtz and ZoranNikolic (see
 * http://code.google.com/u/107017135012155810755/ for details)
 */
publicclass BoundedMapView extends MapView {

protected Rect mScrollableAreaLimit;
protected BoundingBoxE6 box;

public BoundedMapView(Context context, ResourceProxy resourceProxy,
MapTileProviderBase provider) {

super(context, provider.getTileSource().getTileSizePixels(), resourceProxy,
provider);

    }

/**
     * Set the map to limit it's scrollable view to the specified
BoundingBoxE6.
     * Note that, like North/South bounds limiting, this allows an overscroll
of
     * half the screen size. This means each border can be scrolled to the
     * center of the screen.
     *
     * @param box
     *            A lat/long bounding box to limit scrolling to, or null to
     *            remove any scrolling limitations
     */
publicvoid setScrollableAreaLimit(BoundingBoxE6 box) {

finalint worldSize_2 = TileSystem.MapSize(MapViewConstants.MAXIMUM_ZOOMLEVEL)
/ 2;

// Clear scrollable area limit if null passed.
if (box == null) {
mScrollableAreaLimit = null;
return;
        }
```

```
// Get NW/upper-left
final Point upperLeft = TileSystem.LatLongToPixelXY(box.getLatNorthE6() /
1E6,

box.getLonWestE6() / 1E6,

MapViewConstants.MAXIMUM_ZOOMLEVEL,
null);
        upperLeft.offset(-worldSize_2, -worldSize_2);

// Get SE/lower-right
final Point lowerRight = TileSystem.LatLongToPixelXY(box.getLatSouthE6() /
1E6,

box.getLonEastE6() / 1E6,

MapViewConstants.MAXIMUM_ZOOMLEVEL,
null);
        lowerRight.offset(-worldSize_2, -worldSize_2);
mScrollableAreaLimit = new Rect(upperLeft.x,
                                        upperLeft.y,
                                        lowerRight.x,
                                        lowerRight.y);
    }

@Override
publicvoid scrollTo(int x, int y) {
finalint worldSize_2 = TileSystem.MapSize(this.getZoomLevel(true)) / 2;
while (x < -worldSize_2) {
            x += worldSize_2 * 2;
        }
while (x > worldSize_2) {
            x -= worldSize_2 * 2;
        }
if (y < -worldSize_2) {
            y = -worldSize_2;
        }
if (y > worldSize_2) {
            y = worldSize_2;
        }

if (mScrollableAreaLimit != null) {
finalint zoomDiff = MapViewConstants.MAXIMUM_ZOOMLEVEL - getZoomLevel();
finalint minX = mScrollableAreaLimit.left>> zoomDiff;
finalint minY = mScrollableAreaLimit.top>> zoomDiff;
finalint maxX = mScrollableAreaLimit.right>> zoomDiff;
finalint maxY = mScrollableAreaLimit.bottom>> zoomDiff;
if (x < minX)
                x = minX;
elseif (x > maxX)
                x = maxX;
if (y < minY)
                y = minY;
elseif (y > maxY)
                y = maxY;
        }
super.scrollTo(x, y);
```

```
// do callback on listener
if (mListener != null) {
final ScrollEvent event = new ScrollEvent(this, x, y);
mListener.onScroll(event);
        }
    }

@Override
publicvoid computeScroll() {
final Scroller mScroller = getScroller();
finalint mZoomLevel = getZoomLevel(false);

if (mScroller.computeScrollOffset()) {
if (mScroller.isFinished()) {
/**
                * Need to jump through some accessibility hoops here Silly
                * enough the only thing MapController.setZoom does is call
                * MapView.setZoomLevel(zoomlevel). But noooo .. if I try
that
                * directly setZoomLevel needs to be set to "protected".
                * Explanation can be found at
                * http://docs.oracle.com/javase/tutorial
                * /java/javaOO/accesscontrol.html
                *
                * This also suggests that if the subclass is made to be part
of
                * the package, this can be replaced by a simple call to
                * setZoomLevel(mZoomLevel)
                */
// This will facilitate snapping-to any Snappable points.
                getController().setZoom(mZoomLevel);
            } else {
/* correction for double tap */
int targetZoomLevel = getZoomLevel();
if (targetZoomLevel == mZoomLevel)
                    scrollTo(mScroller.getCurrX(), mScroller.getCurrY());
            }
            postInvalidate(); // Keep on drawing until the animation has
// finished.
        }
    }

}
```

DirectionsActivity.java

```
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;
import android.widget.AdapterView.OnItemSelectedListener;
import android.app.Activity;
import android.content.Intent;
import android.content.res.Configuration;

publicclass DirectionsActivity extends Activity implements
OnItemSelectedListener, OnClickListener {

	private String from = null;
	private String to = null;
	private Spinner spinner1 = null;
	private Spinner spinner2 = null;
	private Button button1 = null;
	private Button button2 = null;


	@Override
	protectedvoid onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_directions);

		spinner1 = (Spinner) findViewById(R.id.spinner1);
		spinner2 = (Spinner) findViewById(R.id.spinner2);
		button1 = (Button) findViewById(R.id.button1);
		button2 = (Button) findViewById(R.id.button2);

		// Create an ArrayAdapter using the string array and a default
spinner
		// layout
		ArrayAdapter<CharSequence> adapter1 =
ArrayAdapter.createFromResource(this, R.array.destinationsFrom,
android.R.layout.simple_spinner_item);
		ArrayAdapter<CharSequence> adapter2 =
ArrayAdapter.createFromResource(this, R.array.destinationsTo,
android.R.layout.simple_spinner_item);

		// Specify the layout to use when the list of choices appears

	adapter1.setDropDownViewResource(android.R.layout.simple_spinner_dropd
own_item);

	adapter2.setDropDownViewResource(android.R.layout.simple_spinner_dropd
own_item);

		// Apply the adapter to the spinner
		spinner1.setAdapter(adapter1);
		spinner2.setAdapter(adapter2);
```

```
            spinner1.setOnItemSelectedListener(this);
            spinner2.setOnItemSelectedListener(this);
            button1.setOnClickListener(this);
            button2.setOnClickListener(this);



      }

      publicvoid onItemSelected(AdapterView<?> arg0, View arg1, int arg2,
long arg3) {

            switch (arg0.getId()) {
            caseR.id.spinner1:
                  from = spinner1.getSelectedItem().toString();
                  break;

            caseR.id.spinner2:
                  to = spinner2.getSelectedItem().toString();
                  break;

            }

      }

      publicvoid onNothingSelected(AdapterView<?> arg0) {

      }

      publicvoid onClick(View v) {

            // if "Find" was pressed
            if (v.getId() == R.id.button1) {

                  if (from.equals(to)) {

                        Toast toast =
Toast.makeText(getApplicationContext(), "\"From\" and \"To\" fields must be
different. ", Toast.LENGTH_SHORT);
                        toast.setGravity(Gravity.CENTER, 0, 0);
                        toast.show();

                  } else {

                        if (from.equals("My Location")) {
                              from = "mylocation";
                        } elseif (from.equals("Dormitory")) {
                              from = "1887912229";
                        } elseif (from.equals("Facility of Pure and Applied
Science 01")) {
                              from = "06192001";
                        } elseif (from.equals("Facility of Pure and Applied
Science 02")) {
                              from = "1887912377";
                        } elseif (from.equals("Teaching Facility 01")) {
                              from = "262813490";
                        } elseif (from.equals("Sports Center")) {
```

```
                        from = "1887912526";
                    }

                    if (to.equals("My Location")) {
                        to = "mylocation";
                    } elseif (to.equals("Dormitory")) {
                        to = "1887912229";
                    } elseif (to.equals("Facility of Pure and Applied
Science 01")) {
                        to = "06192001";
                    } elseif (to.equals("Facility of Pure and Applied
Science 02")) {
                        to = "1887912377";
                    } elseif (to.equals("Teaching Facility 01")) {
                        to = "262813490";
                    } elseif (to.equals("Sports Center")) {
                        to = "1887912526";
                    }


                    String path = from + " " + to;
                    Intent data = new Intent();
                    data.putExtra("path", path);
                    // Activity finished OK, return the data
                    setResult(RESULT_OK, data);
                    super.finish();

                }
                // if "Cancel" was pressed
            } else {
                setResult(RESULT_CANCELED, null);
                super.finish();
            }

        }

        @Override
        publicvoid onConfigurationChanged(Configuration newConfig) {
            super.onConfigurationChanged(newConfig);
        }

}
```

EdgeData.java

```
publicclass EdgeData {

	private String idA;
	private String idB;
	privatedoubleax;
	privatedoublec;

	public EdgeData(String inputIdA, String inputIdB) {

		setIdA(inputIdA);
		setIdB(inputIdB);

	}

	public String getIdA() {
		returnidA;
	}

	publicvoid setIdA(String idA) {
		this.idA = idA;
	}

	public String getIdB() {
		returnidB;
	}

	publicvoid setIdB(String idB) {
		this.idB = idB;
	}

	publicdouble getAx() {
		returnax;
	}

	publicvoid setAx(double lat1x, double lat2x, double lon1y, double
lon2y) {

		double d = lat2x - lat1x;

		if (d == 0) {
			this.ax = 0;
		} else {
			this.ax = (lon2y - lon1y) / d;
		}
	}

	publicdouble getC() {
		returnc;
	}

	publicvoid setC(double lat1x, double lat2x, double lon1y, double
lon2y) {

		double d = lat2x - lat1x;

		if (d == 0) {
```

```
                this.c = lon1y;
            } else {
                this.c = lon1y - ((lon2y - lon1y) / d) * lat1x;
            }

        }

    }
```

GalleryViewActivity.java

```
import java.util.Arrays;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.widget.Gallery;

publicclass GalleryViewActivity extends Activity {

	private Integer[] pics = null;
	// adapter for gallery view
	private ImageAdapter imgAdapt;
	// gallery object
	private Gallery picGallery;


	@Override
	publicvoid onCreate(Bundle savedInstanceState) {

		// call superclass method and set main content view
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_gallery_view);

		Intent mIntent = getIntent();
		Object[] s = (Object[]) mIntent.getSerializableExtra("gallery");
		pics = Arrays.copyOf(s, s.length, Integer[].class);

		// get the gallery view
		picGallery = (Gallery) findViewById(R.id.gallery);

		// create a new adapter
		imgAdapt = new ImageAdapter(this, pics);

		// set the gallery adapter
		picGallery.setAdapter(imgAdapt);

	}

}
```

ImageAdapter.java

```
import android.content.Context;
import android.content.res.TypedArray;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Gallery;
import android.widget.ImageView;

publicclass ImageAdapter extends BaseAdapter {

	Integer[] pics = null;
	private Context ctx;
	intimageBackground;

	public ImageAdapter(Context c, Integer[] images) {
		ctx = c;
		TypedArray ta =
ctx.obtainStyledAttributes(R.styleable.Gallery1);
		imageBackground =
ta.getResourceId(R.styleable.Gallery1_android_galleryItemBackground, 1);
		ta.recycle();
		pics = images;
	}

	@Override
	publicint getCount() {

		returnpics.length;
	}

	@Override
	public Object getItem(int arg0) {

		return arg0;
	}

	@Override
	publiclong getItemId(int arg0) {

		return arg0;
	}

	@Override
	public View getView(int arg0, View arg1, ViewGroup arg2) {
		ImageView iv = new ImageView(ctx);
		iv.setImageResource(pics[arg0]);
		iv.setScaleType(ImageView.ScaleType.FIT_XY);
		iv.setLayoutParams(new Gallery.LayoutParams(150, 120));
		iv.setBackgroundResource(imageBackground);
		return iv;
	}
}
```

MapActivity.java

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.List;
import java.util.Set;

import org.jgrapht.alg.DijkstraShortestPath;
import org.jgrapht.graph.DefaultWeightedEdge;
import org.jgrapht.graph.SimpleDirectedWeightedGraph;
import org.osmdroid.DefaultResourceProxyImpl;
import org.osmdroid.ResourceProxy;
import org.osmdroid.tileprovider.IRegisterReceiver;
import org.osmdroid.tileprovider.MapTileProviderArray;
import org.osmdroid.tileprovider.modules.IArchiveFile;
import org.osmdroid.tileprovider.modules.MBTilesFileArchive;
import org.osmdroid.tileprovider.modules.MapTileFileArchiveProvider;
import org.osmdroid.tileprovider.modules.MapTileModuleProviderBase;
import org.osmdroid.tileprovider.tilesource.XYTileSource;
import org.osmdroid.tileprovider.util.SimpleRegisterReceiver;
import org.osmdroid.util.BoundingBoxE6;
import org.osmdroid.util.GeoPoint;
import org.osmdroid.views.MapController;
import org.osmdroid.views.overlay.MyLocationOverlay;
import org.osmdroid.views.overlay.OverlayItem;
import org.osmdroid.views.overlay.PathOverlay;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.res.Configuration;
import android.graphics.Color;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.Environment;
import android.util.Log;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;

publicclass MapActivity extends Activity implements IRegisterReceiver {

	// Default map zoom level:
	privatefinalstaticintMAP_DEFAULT_ZOOM = 15;
	// Default map Latitude:
	privatefinalstaticdoubleMAP_DEFAULT_LATITUDE = 35.144766;
```

```
	// Default map Longitude:
	privatefinalstaticdoubleMAP_DEFAULT_LONGITUDE = 33.409520;
	// Default map North bound:
	privatefinalstaticdoubleMAP_DEFAULT_NORTH_BOUND = 35.155846;
	// Default map East bound:
	privatefinalstaticdoubleMAP_DEFAULT_EAST_BOUND = 33.425903;
	// Default map South bound:
	privatefinalstaticdoubleMAP_DEFAULT_SOUTH_BOUND = 35.133387;
	// Default map West bound:
	privatefinalstaticdoubleMAP_DEFAULT_WEST_BOUND = 33.398438;
	// Radius of the earth in km
	privatefinalstaticintRa = 6371;

	private DefaultResourceProxyImpl resProxy = null;
	private XYTileSource tSource = null;
	private SimpleRegisterReceiver sr = null;
	private MapTileModuleProviderBase moduleProvider = null;
	private MapTileModuleProviderBase[] pBaseArray = null;
	private MapTileProviderArray provider = null;
	private BoundingBoxE6 bBox = null;
	private BoundedMapView mymapView = null;
	private MapController myMapController = null;
	private MyLocationOverlay location = null;

	private LocationManager mylocmanager = null;
	private LocationListener myloclistener = null;

	private ArrayList<OverlayItem>OverlayItemArray = null;
	private OverlayMarker overlay = null;
	private PathOverlay myPath = null;
	private SimpleDirectedWeightedGraph<String, DefaultWeightedEdge>graph;
	private ArrayList<PointData>points;
	private ArrayList<EdgeData>edges;

	@Override
	publicvoid onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);

		mylocmanager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);

		// check if gps is turned on
		if
(!mylocmanager.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
			showGPSDisabledAlertToUser();
		}

		// Create the mapView with an MBTileProvider
		resProxy = new
DefaultResourceProxyImpl(this.getApplicationContext());

		tSource = new XYTileSource("mbtiles",
ResourceProxy.string.offline_mode, 16, 18, 256, ".png", "localhost");
		sr = new SimpleRegisterReceiver(this);

		String packageDir = "/osmdroid";
		String path = Environment.getExternalStorageDirectory() +
packageDir;
```

```
            File file = new File(path, "UcyMap.mbtiles");

            // Check if the MBTiles file is already on SD CARD
            // if not, make a copy on SD CARD
            if (!file.exists()) {

                  File dir = new File(path);
                  dir.mkdir();
                  try {

                        File f2 = new File(path, "UcyMap.mbtiles");
                        InputStream in =
getAssets().open("UcyMap.mbtiles");
                        OutputStream out = new FileOutputStream(f2);

                        byte[] buf = newbyte[1024];
                        int len;
                        while ((len = in.read(buf)) > 0) {
                              out.write(buf, 0, len);
                        }
                        in.close();
                        out.close();
                        file = new File(path, "UcyMap.mbtiles");
                  } catch (FileNotFoundException ex) {
                        System.out.println(ex.getMessage() + " in the
specified directory.");
                        System.exit(0);
                  } catch (IOException e) {
                        System.out.println(e.getMessage());
                  }

            }

            IArchiveFile[] files = {
MBTilesFileArchive.getDatabaseFileArchive(file) };
            moduleProvider = new MapTileFileArchiveProvider(sr, tSource,
files);
            pBaseArray = new MapTileModuleProviderBase[] { moduleProvider };
            provider = new MapTileProviderArray(tSource, null, pBaseArray);

            mymapView = new BoundedMapView(this, resProxy, provider);

            // Set the bounds of the scrolable area on the map
            bBox = new BoundingBoxE6(MAP_DEFAULT_NORTH_BOUND,
MAP_DEFAULT_EAST_BOUND, MAP_DEFAULT_SOUTH_BOUND, MAP_DEFAULT_WEST_BOUND);

            mymapView.setScrollableAreaLimit(bBox);
            mymapView.setBuiltInZoomControls(true);
            mymapView.setMultiTouchControls(true);

            // Initialize Map Controller
            myMapController = mymapView.getController();

            // Create a location listener and
            // register for updates on the location
            myloclistener = new LocationListener() {
```

```
                    publicvoid onStatusChanged(String provider, int status,
Bundle extras) {
                          // TODO Auto-generated method stub

                    }

                    publicvoid onProviderEnabled(String provider) {
                          // TODO Auto-generated method stub

                    }

                    publicvoid onProviderDisabled(String provider) {
                          // TODO Auto-generated method stub

                    }

                    publicvoid onLocationChanged(Location location) {
                          // TODO Auto-generated method stub
                    }

              };

      mylocmanager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
1000, 0, myloclistener);

              // Get Location
              location = new MyLocationOverlay(getApplicationContext(),
mymapView);
              // View Location
              mymapView.getOverlays().add(location);
              location.enableMyLocation();
              location.disableFollowLocation();

              // Zoom and Center Map
              myMapController.setZoom(MAP_DEFAULT_ZOOM);
              myMapController.setCenter(new GeoPoint(MAP_DEFAULT_LATITUDE,
MAP_DEFAULT_LONGITUDE));

              // Add the poi on the map
              addPoi();

              // Create the graph
              graph = new SimpleDirectedWeightedGraph<String,
DefaultWeightedEdge>(DefaultWeightedEdge.class);
              points = new ArrayList<PointData>();
              edges = new ArrayList<EdgeData>();

              addVertices();
              addEdges();

              // Set the MapView as the root View for this Activity; done!
              setContentView(mymapView);

      }

      // Create Menu
      @Override
      publicboolean onCreateOptionsMenu(Menu menu) {
```

```
		getMenuInflater().inflate(R.menu.activity_main, menu);
		returntrue;
	}

	// Actions on Menu Options
	@Override
	publicboolean onOptionsItemSelected(MenuItem item) {

		if (item.getItemId() == R.id.arview) {
			Intent i = new Intent(this, ArBrowserActivity.class);
			startActivityForResult(i, 1);
		} elseif (item.getItemId() == R.id.directions) {
			Intent i = new Intent(this, DirectionsActivity.class);
			startActivityForResult(i, 1);
		} elseif (item.getItemId() == R.id.cleardirections) {
			if (mymapView.getOverlays().size() == 3) {
				mymapView.getOverlays().remove(2);
				mymapView.invalidate();
			}
		} elseif (item.getItemId() == R.id.exit) {
			this.finish();
		}
		returnsuper.onOptionsItemSelected(item);
	}

	//
	@Override
	publicvoid onConfigurationChanged(Configuration newConfig) {
		super.onConfigurationChanged(newConfig);
	}

	// Actions to be performed after an activity result is caught
	protectedvoid onActivityResult(int requestCode, int resultCode, Intent
data) {

		switch (requestCode) {
		case 0:
			mylocmanager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);

			// check if gps is turned on
			if
(!mylocmanager.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
				showGPSDisabledAlertToUser();
			}
			break;
		case 1:
			if (resultCode == RESULT_OK) {
				if (data.hasExtra("path")) {
					String path =
data.getExtras().getString("path");
					if (mymapView.getOverlays().size() == 3) {
						mymapView.getOverlays().remove(2);
						addPath(path);
					} else {
						addPath(path);
					}
				}
```

```
                }
                break;
            }

        }

        // Display the "Enable GPS Dialog box"
        privatevoid showGPSDisabledAlertToUser() {

            AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(this);

            alertDialogBuilder.setMessage("GPS is disabled in your device.
Would you like to enable it?");
            alertDialogBuilder.setCancelable(false);
            alertDialogBuilder.setPositiveButton("Enable GPS", new
DialogInterface.OnClickListener() {
                publicvoid onClick(DialogInterface dialog, int id) {
                    Intent callGPSSettingIntent = new
Intent(android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                    startActivityForResult(callGPSSettingIntent, 0);
                }
            });
            alertDialogBuilder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
                publicvoid onClick(DialogInterface dialog, int id) {
                    dialog.cancel();
                    System.exit(1);
                }
            });
            AlertDialog alert = alertDialogBuilder.create();
            alert.show();
        }

        // Add the Points of Interest on the map
        privatevoid addPoi() {

            OverlayItemArray = new ArrayList<OverlayItem>();
            // Points to be added
            GeoPoint point1 = new GeoPoint(35143923, 33405218);
            OverlayItem overlayitem1 = (OverlayItem) new
OverlayItem(getResources().getString(R.string.esties),
getResources().getString(R.string.esties_short_discription), point1);
            OverlayItemArray.add(overlayitem1);

            GeoPoint point2 = new GeoPoint(35144476, 33410475);
            OverlayItem overlayitem2 = new
OverlayItem(getResources().getString(R.string.thee02),
getResources().getString(R.string.thee02_short_discription), point2);
            OverlayItemArray.add(overlayitem2);

            GeoPoint point3 = new GeoPoint(35144617, 33411108);
            OverlayItem overlayitem3 = new
OverlayItem(getResources().getString(R.string.thee01),
getResources().getString(R.string.thee01_short_discription), point3);
            OverlayItemArray.add(overlayitem3);

            GeoPoint point4 = new GeoPoint(35145125, 33410711);
```

```
            OverlayItem overlayitem4 = new
OverlayItem(getResources().getString(R.string.xod01),
getResources().getString(R.string.xod01_short_discription), point4);
            OverlayItemArray.add(overlayitem4);

            GeoPoint point5 = new GeoPoint(35145985, 33413672);
            OverlayItem overlayitem5 = new
OverlayItem(getResources().getString(R.string.sports),
getResources().getString(R.string.sports_short_discription), point5);
            OverlayItemArray.add(overlayitem5);

            // Add Points to Overlay
            overlay = new OverlayMarker(this, OverlayItemArray);
            mymapView.getOverlays().add(overlay);

      }

      // Find the path between two Geo points
      // and add it to the map
      privatevoid addPath(String path) {

            double _locLat = 0;
            double _locLon = 0;
            String[] paths = path.split(" ");

            if (paths[0].equals("mylocation")) {
                  Location loc =
mylocmanager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
                  if (loc != null) {

                        // _locLat = loc.getLatitude();
                        // _locLon = loc.getLongitude();

                        _locLat = 35.144161;
                        _locLon = 33.409746;

                        // If the user location is out of the map bounds
                        // then he is out of the view area
                        if (_locLat >MAP_DEFAULT_NORTH_BOUND || _locLat
<MAP_DEFAULT_SOUTH_BOUND || _locLon >MAP_DEFAULT_EAST_BOUND || _locLon
<MAP_DEFAULT_WEST_BOUND) {
                              Toast toast =
Toast.makeText(getApplicationContext(), "You are out of the visible area",
Toast.LENGTH_SHORT);
                              toast.setGravity(Gravity.CENTER, 0, 0);
                              toast.show();
                        } else {
                              // update the "mylocation" node with the
current
                              // location lat and lon
                              int index = getIndex(paths[0]);
                              points.get(index).setLat(_locLat);
                              points.get(index).setLon(_locLon);

                              Set<DefaultWeightedEdge> locEdge =
graph.edgesOf("mylocation");

                              if (locEdge.size() != 0) {
```

```
                                        Log.d("Testing", locEdge.toString());
                                        String locEdgeStr =
locEdge.toString();
                                        locEdgeStr = locEdgeStr.replace("(",
"");
                                        locEdgeStr = locEdgeStr.replace(")",
"");
                                        locEdgeStr = locEdgeStr.replace(" ",
"");
                                        locEdgeStr = locEdgeStr.replace(",",
":");
                                        String[] locEdgeStrtokens =
locEdgeStr.split(":");
                                        graph.removeEdge(locEdgeStrtokens[0],
locEdgeStrtokens[1]);
                                        graph.removeEdge(locEdgeStrtokens[2],
locEdgeStrtokens[3]);
                                }

                                String nodes = findNearestEdge(_locLat,
_locLon);
                                String[] tokens = nodes.split(" ");

                                DefaultWeightedEdge e =
graph.addEdge("mylocation", tokens[0]);
                                int index1 = getIndex("mylocation");
                                int index2 = getIndex(tokens[0]);
                                double w =
getDistanceFromLatLonInKm(points.get(index1).getLat(),
points.get(index1).getLon(), points.get(index2).getLat(),
points.get(index2).getLon());
                                graph.setEdgeWeight(e, w);

                                e = graph.addEdge("mylocation", tokens[1]);
                                index1 = getIndex("mylocation");
                                index2 = getIndex(tokens[1]);
                                w =
getDistanceFromLatLonInKm(points.get(index1).getLat(),
points.get(index1).getLon(), points.get(index2).getLat(),
points.get(index2).getLon());
                                graph.setEdgeWeight(e, w);

                                List<DefaultWeightedEdge> shortest_path =
DijkstraShortestPath.findPathBetween(graph, "mylocation", paths[1]);
                                myPath = new PathOverlay(Color.RED, this);

                                for (int i = 0; i < shortest_path.size();
i++) {
                                        String tempstr =
shortest_path.get(i).toString();
                                        tempstr = tempstr.replace("(", "");
                                        tempstr = tempstr.replace(")", "");
                                        tempstr = tempstr.replace(" ", "");
                                        String[] temptokens =
tempstr.split(":");
                                        if (i == 0) {
                                                index1 =
getIndex(temptokens[0]);
```

```
                                                myPath.addPoint(new
GeoPoint(points.get(index1).getLat(), points.get(index1).getLon()));
                                          }
                                          index2 = getIndex(temptokens[1]);
                                          myPath.addPoint(new
GeoPoint(points.get(index2).getLat(), points.get(index2).getLon()));
                                    }

                                    mymapView.getOverlays().add(myPath);
                                    mymapView.invalidate();
                              }
                        } else {

                              Toast toast =
Toast.makeText(getApplicationContext(), "Now location was found.\nPlease wait
for satellite to respond.", Toast.LENGTH_SHORT);
                              toast.setGravity(Gravity.CENTER, 0, 0);
                              toast.show();

                        }

                  } else {

                        List<DefaultWeightedEdge> shortest_path =
DijkstraShortestPath.findPathBetween(graph, paths[0], paths[1]);
                        myPath = new PathOverlay(Color.RED, this);

                        for (int i = 0; i < shortest_path.size(); i++) {
                              String tempstr = shortest_path.get(i).toString();
                              tempstr = tempstr.replace("(", "");
                              tempstr = tempstr.replace(")", "");
                              tempstr = tempstr.replace(" ", "");
                              String[] temptokens = tempstr.split(":");
                              if (i == 0) {
                                    int index1 = getIndex(temptokens[0]);
                                    myPath.addPoint(new
GeoPoint(points.get(index1).getLat(), points.get(index1).getLon()));
                              }
                              int index2 = getIndex(temptokens[1]);
                              myPath.addPoint(new
GeoPoint(points.get(index2).getLat(), points.get(index2).getLon()));
                        }

                        mymapView.getOverlays().add(myPath);
                        mymapView.invalidate();

                  }

            }

            // Add the Vertices of the graph
            privatevoid addVertices() {

                  String line = null;
                  String[] tokens = null;
                  InputStream in = null;
                  BufferedReader br = null;
```

```
		try {
			in = getAssets().open("Nodes.txt");
			br = new BufferedReader(new InputStreamReader(in));

			while ((line = br.readLine()) != null) {

				tokens = line.split(" ");
				PointData p = new PointData(tokens[0],
Double.parseDouble(tokens[1]), Double.parseDouble(tokens[2]));
				points.add(p);
				graph.addVertex(p.getId());

			}

			PointData p = new PointData("mylocation", 0.0, 0.0);
			points.add(p);
			graph.addVertex(p.getId());

		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

		try {
			br.close();
		} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

	}

	// Add the Edges of the graph
	privatevoid addEdges() {

		String line = null;
		String[] tokens = null;
		InputStream in = null;
		BufferedReader br = null;

		try {
			in = getAssets().open("Links.txt");
			br = new BufferedReader(new InputStreamReader(in));

			while ((line = br.readLine()) != null) {

				tokens = line.split(" ");

				DefaultWeightedEdge e = graph.addEdge(tokens[0],
tokens[1]);

				int index1 = getIndex(tokens[0]);
				int index2 = getIndex(tokens[1]);
				double w =
getDistanceFromLatLonInKm(points.get(index1).getLat(),
points.get(index1).getLon(), points.get(index2).getLat(),
points.get(index2).getLon());
```

```
                        graph.setEdgeWeight(e, w);

                        EdgeData ed = new EdgeData(tokens[0], tokens[1]);
                        ed.setAx(points.get(index1).getLat(),
points.get(index2).getLat(), points.get(index1).getLon(),
points.get(index2).getLon());
                        ed.setC(points.get(index1).getLat(),
points.get(index2).getLat(), points.get(index1).getLon(),
points.get(index2).getLon());
                        edges.add(ed);

                  }

            } catch (IOException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }

            try {
                  br.close();
            } catch (IOException e) {
                  // TODO Auto-generated catch block
                  e.printStackTrace();
            }

      }

      // returns the index of a point based on its id
      privateint getIndex(String id) {

            for (int i = 0; i <points.size(); i++) {
                  if (id.equals(points.get(i).getId())) {
                        return i;
                  }
            }

            return -1;
      }

      // Find the distance between two Geo points in Km
      privatedouble getDistanceFromLatLonInKm(double lat1, double lon1,
double lat2, double lon2) {

            double dLat = deg2rad(lat2 - lat1);
            double dLon = deg2rad(lon2 - lon1);

            double a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +
Math.cos(deg2rad(lat1)) * Math.cos(deg2rad(lat2)) * Math.sin(dLon / 2) *
Math.sin(dLon / 2);

            double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
            double d = Ra * c; // Distance in km

            return d;

      }

      // Convert degrees to radiant
```

```
	privatedouble deg2rad(double deg) {

		return deg * (Math.PI / 180);

	}

	// Find the nearest Edge to the given lat-lon
	private String findNearestEdge(double lat, double lon) {

		double min = 100;
		String nodes = null;
		for (int i = 0; i <edges.size(); i++) {

			// The a and c of the link
			double a = edges.get(i).getAx();
			double c = edges.get(i).getC();

			// The ids of the nodes of the link
			String idA = edges.get(i).getIdA();
			String idB = edges.get(i).getIdB();

			// The distance of the point from the link
			double newMin = Math.abs(a * lon - lat + c) /
Math.sqrt(Math.pow(a, 2.0) + 1);

			// If the current edge is closest to the user location
			// check if the
			if (newMin <= min) {
				// a and c of the new line
				double a_ = -1 / a;
				double c_ = lat - a_ * lon;

				// Intersection point
				double x = 0;
				double y = 0;

				if ((a_ - a) != 0) {
					x = (c - c_) / (a_ - a);
				}

				y = a_ * x + c_;

				// Get the bounds of the edge
				double minlat = 0;
				double maxlat = 0;
				double minlon = 0;
				double maxlon = 0;

				// The lat and lot of the above nodes
				double lat1 = points.get(getIndex(idA)).getLat();
				double lon1 = points.get(getIndex(idA)).getLon();
				double lat2 = points.get(getIndex(idB)).getLat();
				double lon2 = points.get(getIndex(idB)).getLon();

				if (lat1 < lat2) {
					minlat = lat1;
					maxlat = lat2;
				} else {
```

```
                                    minlat = lat2;
                                    maxlat = lat1;
                             }

                             if (lon1 < lon2) {
                                    minlon = lon1;
                                    maxlon = lon2;
                             } else {
                                    minlon = lon2;
                                    maxlon = lon1;
                             }

                             // If the intersection point is in the bounds then
                             // the edge is nearest
                             if (x >= minlat && x <= maxlat && y >= minlon && y
<= maxlon) {
                                    min = newMin;
                                    nodes = idA + " " + idB;
                             }
                      }
               }

               return nodes;

       }

}
```

MBTilesModuleProvider.java

```
import java.io.File;
import java.io.InputStream;

import org.osmdroid.tileprovider.IRegisterReceiver;
import org.osmdroid.tileprovider.MapTile;
import org.osmdroid.tileprovider.MapTileRequestState;
import org.osmdroid.tileprovider.modules.MapTileFileStorageProviderBase;
import org.osmdroid.tileprovider.modules.MapTileModuleProviderBase;
import org.osmdroid.tileprovider.tilesource.ITileSource;
import org.osmdroid.tileprovider.util.StreamUtils;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import android.graphics.drawable.Drawable;

publicclass MBTileModuleProvider extends MapTileFileStorageProviderBase {

privatestaticfinal Logger logger =
LoggerFactory.getLogger(MBTileModuleProvider.class);

protected MBTileSource tileSource;

/**
     * Constructor
     *
     * @param pRegisterReceiver
     * @param file
     * @param tileSource
     */
public MBTileModuleProvider(IRegisterReceiver receiverRegistrar,
                                File file,
                                MBTileSource tileSource) {

// Call the super constructor
super(receiverRegistrar,
NUMBER_OF_TILE_FILESYSTEM_THREADS,
TILE_FILESYSTEM_MAXIMUM_QUEUE_SIZE);

// Initialize fields
this.tileSource = tileSource;

    }

@Override
protected String getName() {
return"MBTiles File Archive Provider";
    }

@Override
protected String getThreadGroupName() {
return"mbtilesarchive";
    }

@Override
protected Runnable getTileLoader() {
returnnew TileLoader();
```

```
    }

@Override
publicboolean getUsesDataConnection() {
returnfalse;
    }

@Override
publicint getMinimumZoomLevel() {
returntileSource.getMinimumZoomLevel();
    }

@Override
publicint getMaximumZoomLevel() {
returntileSource.getMaximumZoomLevel();
    }

@Override
publicvoid setTileSource(ITileSource tileSource) {
logger.warn("*** Warning: someone's trying to reassign MBTileModuleProvider's
tileSource!");
if (tileSource instanceof MBTileSource) {
this.tileSource = (MBTileSource) tileSource;
        } else {
logger.warn("*** Warning: and it wasn't even an MBTileSource! That's just
rude!");

        }
    }

privateclass TileLoader extends MapTileModuleProviderBase.TileLoader {

@Override
public Drawable loadTile(final MapTileRequestState pState) {

// if there's no sdcard then don't do anything
if (!getSdCardAvailable()) {
returnnull;
            }

            MapTile pTile = pState.getMapTile();
            InputStream inputStream = null;

try {
                inputStream = tileSource.getInputStream(pTile);

if (inputStream != null) {
                    Drawable drawable = tileSource.getDrawable(inputStream);

// Note that the finally clause will be called before
// the value is returned!
return drawable;
                }

            } catch (Throwable e) {
logger.error("Error loading tile", e);

            } finally {
```

```
if (inputStream != null) {
                    StreamUtils.closeStream(inputStream);
                }
            }

returnnull;
        }
    }

}
```

MBTileProvider.java

```
import java.io.File;
import java.util.Collections;

import org.osmdroid.tileprovider.IRegisterReceiver;
import org.osmdroid.tileprovider.MapTileProviderArray;
import org.osmdroid.tileprovider.modules.MapTileModuleProviderBase;

/**
 *
 *
 * This class is a simplification of the the MapTileProviderArray: it only
 * allows a single provider.
 *
 * @authormelle
 *
 */
publicclass MBTileProvider extends MapTileProviderArray {

public MBTileProvider(IRegisterReceiver receiverRegistrar, File file) {

/**
         * Call the super-constructor.
         *
         * MapTileProviderBase requires a TileSource. As far as I can tell it
is
         * only used in its method rescaleCache(...) to get the pixel size of
a
         * tile. It seems to me that this is inappropriate, as a
MapTileProvider
         * can have multiple sources (like the module array defined below)
and
         * therefore multiple tileSources which might return different
values!!
         *
         * If the requirement is that the tile size is equal across tile
         * sources, then the parameter should be obtained from a different
         * location, From TileSystem for example.
         */
super(MBTileSource.createFromFile(file), receiverRegistrar);

// Create the module provider; this class provides a TileLoader that
// actually loads the tile from the DB.
        MBTileModuleProvider moduleProvider;
        moduleProvider = new MBTileModuleProvider(receiverRegistrar,
                                                  file,
                                                  (MBTileSource)
getTileSource());

        MapTileModuleProviderBase[] pTileProviderArray;
        pTileProviderArray = new MapTileModuleProviderBase[] { moduleProvider
};

// Add the module provider to the array of providers; mTileProviderList
// is defined by the superclass.
        Collections.addAll(mTileProviderList, pTileProviderArray);
    }
```

```
// TODO: implement public Drawable getMapTile(final MapTile pTile) {}
//       The current implementation is needlessly complex because it uses
//       MapTileProviderArray as a basis.

}
```

MBTileSource.java

```
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.InputStream;

import org.osmdroid.ResourceProxy;
import org.osmdroid.ResourceProxy.string;
import org.osmdroid.tileprovider.MapTile;
import org.osmdroid.tileprovider.tilesource.BitmapTileSourceBase;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;

publicclass MBTileSource extends BitmapTileSourceBase {

// Log log log log ...
privatestaticfinal Logger logger =
LoggerFactory.getLogger(MBTileSource.class);

// Database related fields
publicfinalstatic String TABLE_TILES = "tiles";
publicfinalstatic String COL_TILES_ZOOM_LEVEL = "zoom_level";
publicfinalstatic String COL_TILES_TILE_COLUMN = "tile_column";
publicfinalstatic String COL_TILES_TILE_ROW = "tile_row";
publicfinalstatic String COL_TILES_TILE_DATA = "tile_data";

protected SQLiteDatabase database;
protected File archive;

// Reasonable defaults ..
publicstaticfinalintminZoom = 8;
publicstaticfinalintmaxZoom = 15;
publicstaticfinalinttileSizePixels = 256;

// Required for the superclass
publicstaticfinal string resourceId = ResourceProxy.string.offline_mode;

/**
     * The reason this constructor is protected is because all parameters,
     * except file should be determined from the archive file. Therefore a
     * factory method is necessary.
     *
     * @param minZoom
     * @param maxZoom
     * @param tileSizePixels
     * @param file
     */
protected MBTileSource(int minZoom,
int maxZoom,
int tileSizePixels,
                           File file,
                           SQLiteDatabase db) {
super("MBTiles", resourceId, minZoom, maxZoom, tileSizePixels, ".png");
```

```
archive = file;
database = db;
    }

/**
     * Creates a new MBTileSource from file.
     *
     * Parameters minZoom, maxZoom en tileSizePixels are obtained from the
     * database. If they cannot be obtained from the DB, the default values
as
     * defined by this class are used.
     *
     * @param file
     * @return
     */
publicstatic MBTileSource createFromFile(File file) {
        SQLiteDatabase db;
int flags = SQLiteDatabase.NO_LOCALIZED_COLLATORS |
SQLiteDatabase.OPEN_READONLY;

int value;
int minZoomLevel;
int maxZoomLevel;
int tileSize = tileSizePixels;
        InputStream is = null;

// Open the database
        db = SQLiteDatabase.openDatabase(file.getAbsolutePath(), null,
flags);

// Get the minimum zoomlevel from the MBTiles file
        value = getInt(db, "SELECT MIN(zoom_level) FROM tiles;");
        minZoomLevel = value > -1 ? value : minZoom;

// Get the maximum zoomlevel from the MBTiles file
        value = getInt(db, "SELECT MAX(zoom_level) FROM tiles;");
        maxZoomLevel = value > -1 ? value : maxZoom;

// Get the tile size
        Cursor cursor = db.rawQuery("SELECT tile_data FROM images LIMIT 0,1",
new String[] {});

if (cursor.getCount() != 0) {
            cursor.moveToFirst();
            is = new ByteArrayInputStream(cursor.getBlob(0));

            Bitmap bitmap = BitmapFactory.decodeStream(is);
            tileSize = bitmap.getHeight();
logger.debug(String.format("Found a tile size of %d", tileSize));
        }

        cursor.close();
// db.close();

returnnew MBTileSource(minZoomLevel, maxZoomLevel, tileSize, file, db);
    }
```

```
protectedstaticint getInt(SQLiteDatabase db, String sql) {
        Cursor cursor = db.rawQuery(sql, new String[] {});
int value = -1;

if (cursor.getCount() != 0) {
            cursor.moveToFirst();
            value = cursor.getInt(0);
logger.debug(String.format("Found a minimum zoomlevel of %d", value));
        }

        cursor.close();
return value;
    }

public InputStream getInputStream(MapTile pTile) {

try {
            InputStream ret = null;
final String[] tile = { COL_TILES_TILE_DATA };
final String[] xyz = { Integer.toString(pTile.getX()),
                                    Double.toString(Math.pow(2,
pTile.getZoomLevel()) - pTile.getY() - 1),
                                    Integer.toString(pTile.getZoomLevel()) };

final Cursor cur = database.query(TABLE_TILES,
                                                    tile,
"tile_column=? and tile_row=? and zoom_level=?",
                                                    xyz,
null,
null,
null);

if (cur.getCount() != 0) {
                cur.moveToFirst();
                ret = new ByteArrayInputStream(cur.getBlob(0));
            }

            cur.close();

if (ret != null) {
return ret;
            }

        } catch (final Throwable e) {
logger.warn("Error getting db stream: " + pTile, e);
        }

returnnull;

    }

}
```

OverlayMarker.java

```
import java.util.List;

import org.osmdroid.views.MapView;
import org.osmdroid.views.overlay.ItemizedIconOverlay;
```

```
import org.osmdroid.views.overlay.OverlayItem;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.util.Log;

publicclass OverlayMarker extends ItemizedIconOverlay<OverlayItem> {

	private Context mContext;

	public OverlayMarker(final Context context, final List<OverlayItem>
aList) {
		super(context, aList, new OnItemGestureListener<OverlayItem>() {
			publicboolean onItemSingleTapUp(finalint index, final
OverlayItem item) {
				returnfalse;
			}

			publicboolean onItemLongPress(finalint index, final
OverlayItem item) {
				returnfalse;
			}
		});
		// TODO Auto-generated constructor stub
		mContext = context;
	}

	@Override
	protectedboolean onSingleTapUpHelper(finalint index, final OverlayItem
item, final MapView mapView) {

		AlertDialog.Builder dialog = new AlertDialog.Builder(mContext);

		dialog.setTitle(item.getTitle());
		dialog.setMessage(item.getSnippet());


		dialog.setPositiveButton("More", new
DialogInterface.OnClickListener() {
			publicvoid onClick(DialogInterface dialog, int id) {
				Intent poidetails = new Intent(mContext,
PoiDetailsActivity.class);
				poidetails.putExtra("id", item.getTitle());
				Log.d("Testing", item.getTitle());
				poidetails.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
				mContext.startActivity(poidetails);
			}
		});

		dialog.show();
		returntrue;
	}

}
```

PioBean.java

```
import org.json.JSONException;
import org.json.JSONObject;


/**
 * PoiBean class that serves as a container for poi information
 * contains basic information such as name, description, type and location
 */
publicclass PoiBean {
	private String name;
	private String description;
	private Point point;

	privateclass Point
	{
		privatedoublelatitude;
		privatedoublelongitude;
		privatedoublealtitude;

		public Point(double lat, double lon, double alt)
		{
			this.latitude = lat;
			this.longitude = lon;
			this.altitude = alt;
		}

		public JSONObject toJSONString() throws JSONException
		{
			JSONObject object = new JSONObject();
			object.put("latitude", this.latitude);
			object.put("longitude", this.longitude);
			object.put("altitude", this.altitude);
			return object;
		}
	}

	public PoiBean(String _name, String _desc, double lat, double lon,
double alt)
	{
		this.name = _name;
		this.description = _desc;
		this.point = new Point(lat, lon, alt);
	}

	public String getName() {
		returnname;
	}

	publicvoid setName(String name) {
		this.name = name;
	}

	public String getDescription() {
		returndescription;
	}
```

```
	publicvoid setDescription(String description) {
		this.description = description;
	}

	public JSONObject toJSONObject() throws JSONException
	{
		JSONObject object = new JSONObject();
		object.put("name", this.name);
		object.put("description", this.description);
		object.put("Point", this.point.toJSONString());

		return object;
	}

}
```

PoiDetailsActivity.java

```
import android.app.Activity;
import android.content.Intent;
import android.content.res.Configuration;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Gallery;
import android.widget.TextView;
import android.widget.AdapterView.OnItemClickListener;

publicclass PoiDetailsActivity extends Activity {

	private TextView title;
	private TextView discription;
	private Integer[] pics = null;
	// adapter for gallery view
	private ImageAdapter imgAdapt;
	// gallery object
	private Gallery picGallery;

	@Override
	protectedvoid onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_poi_details);

		Intent mIntent = getIntent();
		String value = mIntent.getExtras().getString("id");

		title = (TextView) findViewById(R.id.PoiTitle);
		discription = (TextView) findViewById(R.id.PoiDescription);

		picGallery = (Gallery) findViewById(R.id.samplegallery);


		if (value.equals("University Dormitory")) {
			title.setText(R.string.esties);
			discription.setText(R.string.esties_discription);
			pics = new Integer[3];
			pics[0] = R.drawable.marker1;
			pics[1] = R.drawable.marker2;
			pics[2] = R.drawable.marker3;


		} elseif (value.equals("Facility of Pure and Applied Science
01")) {
			title.setText(R.string.thee01);
			discription.setText(R.string.thee01_discription);
			pics = new Integer[3];
			pics[0] = R.drawable.marker1;
			pics[1] = R.drawable.marker2;
			pics[2] = R.drawable.marker3;

		} elseif (value.equals("Facility of Pure and Applied Science
02")) {
			title.setText(R.string.thee02);
			discription.setText(R.string.thee02_discription);
```

```
                pics = new Integer[3];
                pics[0] = R.drawable.marker1;
                pics[1] = R.drawable.marker2;
                pics[2] = R.drawable.marker3;

            } elseif (value.equals("Teaching Facility 01")) {
                title.setText(R.string.xod01);
                discription.setText(R.string.xod01_discription);
                pics = new Integer[3];
                pics[0] = R.drawable.marker1;
                pics[1] = R.drawable.marker2;
                pics[2] = R.drawable.marker3;

            } elseif (value.equals("Sports Center")) {
                title.setText(R.string.sports);
                discription.setText(R.string.sports_discription);
                pics = new Integer[3];
                pics[0] = R.drawable.marker1;
                pics[1] = R.drawable.marker2;
                pics[2] = R.drawable.marker3;

            }

            imgAdapt = new ImageAdapter(this, pics);
            picGallery.setAdapter(imgAdapt);

            picGallery.setOnItemClickListener(new OnItemClickListener() {
                // handle clicks
                publicvoid onItemClick(AdapterView<?> parent, View v, int
position, long id) {
                    // set the larger image view to display the chosen
bitmap
                    // calling method of adapter class
                    Intent gallery = new
Intent(getApplicationContext(), GalleryViewActivity.class);
                    gallery.putExtra("gallery", pics);
                }
            });

    }

    @Override
    publicvoid onConfigurationChanged(Configuration newConfig) {
        super.onConfigurationChanged(newConfig);
    }

}
```

PointData.java

```
publicclass PointData {

	private String id;
	privatedoublelat;
	privatedoublelon;

	publicPointData(String inputId, double inputLat, double inputlon) {

		setId(inputId);
		setLat(inputLat);
		setLon(inputlon);

	}

	public String getId() {
		returnid;
	}

	publicvoid setId(String id) {
		this.id = id;
	}

	publicdouble getLat() {
		returnlat;
	}

	publicvoid setLat(double lat) {
		this.lat = lat;
	}

	publicdouble getLon() {
		returnlon;
	}

	publicvoid setLon(double lon) {
		this.lon = lon;
	}

}
```

ARControls.js

```
// Create new images, which will be loaded right away
firstImage = new AR.ImageResource("esties.png", {onError: errorLoadingImage});
secondImage = new AR.ImageResource("thee01.png", {onError: errorLoadingImage});
thirdImage = new AR.ImageResource("thee02.png", {onError: errorLoadingImage});
fourthImage = new AR.ImageResource("xod.png", {onError: errorLoadingImage});
fifthImage = new AR.ImageResource("sportscenter.png", {onError:
errorLoadingImage});


// current selected object
var selectedObject = null;


//variable that keeps the jsonData received from the native app
var jsonObject;


//function that gets called when the displayed poi bubble is clicked
//sends the id of the selected poi to the native app
function generateOnPoiBubbleClickFunc(name){

   return function(){
      document.location = "architectsdk://opendetailpage?id="+name;
   }

}


// creates a property animation
function createOnClickAnimation(imageDrawable){

   var anim = new AR.PropertyAnimation( imageDrawable, 'scaling', 1.0, 1.2, 750, new
AR.EasingCurve(AR.CONST.EASING_CURVE_TYPE.EASE_OUT_ELASTIC,
{amplitude : 2.0}) );
   return anim;

}


// creates a function for assigning to label's and imageDrawable's onClickTrigger
function createClickTrigger(id){

         return function(){
      // hide the bubble
      document.getElementById("footer").style.display = 'block';
      document.getElementById("poiName").innerHTML = jsonObject[id].name;
```

```
        document.getElementById("poiDesc").innerHTML = "Description: " +
jsonObject[id].description;
                document.getElementById("more").innerHTML = "More"
        document.getElementById("footer").onclick=
generateOnPoiBubbleClickFunc(jsonObject[id].name);

        // reset the previous selected poi
        if(selectedObject != null){
                        // reset the property animation
                        selectedObject.animation.stop();

                        //selectedObject.arLabel.style.backgroundColor = '#FFFFFF80';
                        selectedObject.img.scaling = 1.0;
                        selectedObject.poiObj.renderingOrder = 0;
        }

        // set a new select status for the current selected poi
        selectedObject = jsonObject[id];
        //selectedObject.arLabel.style.backgroundColor = '#FFFFFFFF';
        selectedObject.poiObj.renderingOrder = 1;

        // start the assigned animation
        selectedObject.animation.start();

        return true;
    }
}

//function called from the native app fia callJavascript method
//receives json-data as string and processes the contained information
function newData(jsonData){
    jsonObject = JSON.parse(jsonData);

    for(var i = 0; i < jsonObject.length; i++)
    {
        var poidrawables = new Array();
        /*var label = new AR.Label(jsonObject[i].name,1.0, {offsetY : -1.5,
         triggers: {
         onClick:
         createClickTrigger(jsonObject[i].id)},
         style : {fontsize: '9px', textColor : '#FFC100',backgroundColor : '#FFFFFF80'}});

        jsonObject[i].arLabel = label;
        */
        var poiImage;
        if(jsonObject[i].name == "University Dormitory"){
                        poiImage = firstImage;
        }else if(jsonObject[i].name == "Facility of Pure and Applied Science 01"){
                        poiImage = secondImage;
```

```
        }else if(jsonObject[i].name == "Facility of Pure and Applied Science 02"){
                        poiImage = thirdImage;
        }else if(jsonObject[i].name == "Teaching Facility 01"){
                        poiImage = fourthImage;
        }else{
                        poiImage = fifthImage;
        }


        var img = new AR.ImageDrawable(poiImage, 2.0,{triggers:
{onClick:createClickTrigger(i)}});

        jsonObject[i].animation = createOnClickAnimation(img);
        jsonObject[i].img = img;

        //poidrawables.push(label);
        poidrawables.push(img);
        geoLoc = new
AR.GeoLocation(jsonObject[i].Point.latitude,jsonObject[i].Point.longitude,jsonObject[i
].Point.altitude);
        jsonObject[i].poiObj = new AR.GeoObject(geoLoc, {drawables: {cam:
poidrawables}});
    }



}

// Called if loading of the image fails.
function errorLoadingImage() {
    // set error message on HUD
    document.getElementById("footer").style.display
    document.getElementById("poiDesc").innerHTML = "Unable to load image!";
}


// hide bubble and reset the selected poi if nothing was hit by a display click
AR.context.onScreenClick = function(){
    // hide the bubble
    //document.getElementById("footer").style.display = 'none';
    document.getElementById("poiName").innerHTML = "";
    document.getElementById("poiDesc").innerHTML = "";
          document.getElementById("more").innerHTML = "";
    // and reset the current selected poi
    if(selectedObject != null){
        // reset the property animation
        selectedObject.animation.stop();

        //selectedObject.arLabel.style.backgroundColor = '#FFFFFF80';
```

```
        selectedObject.img.scaling = 1.0;
        selectedObject.poiObj.renderingOrder = 0;
        selectedObject = null;
    }
}
```

ARWorld.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!-- saved from
url=(0061)http://www.wikitude.com/doc/architect_tutorial/tutorial5.html -->
<html>
	<head>
		<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
		<title>Augmented Reality World</title>

		<!-- Include CSS Styles -->
		<link href="styles.css" rel="stylesheet" type="text/css">

		<!-- Include the ARchitect library -->
		<script src="architect://architect.js"></script>
		<script src="ARControl.js"></script>

	</head>
<body>
<div id="footer">
<h3><span id="poiName"></span></h3>
<span id="poiDesc"></span><br/>
<p id="more"style="text-align:right;"></p>
</div>
</body>
</html>
```

Activity_ar_browser.xml

```
<?xmlversion="1.0"encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent">


<com.wikitude.architect.ArchitectView
android:id="@+id/architectView"
android:layout_width="fill_parent"
android:layout_height="fill_parent">
      </com.wikitude.architect.ArchitectView>


</LinearLayout>
```

Activity_directions.xml

```
<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:orientation="vertical">

<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:paddingLeft="10dp"
android:paddingTop="10dp"
android:text="@string/from"
android:textAppearance="?android:attr/textAppearanceLarge"/>

<Spinner
android:id="@+id/spinner1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:paddingLeft="10dp"
android:layout_below="@+id/textView1"/>

<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:paddingLeft="10dp"
android:layout_below="@+id/spinner1"
android:text="@string/to"
android:textAppearance="?android:attr/textAppearanceLarge"/>

<Button
android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button1"
android:layout_alignBottom="@+id/button1"
android:layout_marginLeft="44dp"
android:layout_toRightOf="@+id/button1"
android:text="@string/cancle"/>

<Spinner
android:id="@+id/spinner2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_below="@+id/textView2"
android:paddingLeft="10dp"/>

<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_below="@+id/spinner2"
android:paddingLeft="10dp"
android:text="@string/find"/>

</RelativeLayout>
```

Activity_galery_view.xml

```
<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent">

<Gallery
android:id="@+id/gallery"
android:layout_width="fill_parent"
android:layout_height="wrap_content"/>

</RelativeLayout>
```

Activity_poi_details.xml

```
<?xmlversion="1.0"encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/LinearLayout1"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical">

<TextView
android:id="@+id/PoiTitle"
android:layout_width="fill_parent"
android:layout_height="30dp"
android:text="@string/empty"
android:textAppearance="?android:attr/textAppearanceLarge"/>

<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent">

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical">

<Gallery
android:id="@+id/samplegallery"
android:layout_width="fill_parent"
android:layout_height="wrap_content"/>

<TextView
android:id="@+id/PoiDescription"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:scrollbarStyle="insideInset"
android:scrollbars="vertical"
android:text="@string/empty"
android:textAppearance="?android:attr/textAppearanceMedium"/>
</LinearLayout>
</ScrollView>

</LinearLayout>
```

Activity_main.xml

```
menuxmlns:android="http://schemas.android.com/apk/res/android">

<itemandroid:id="@+id/arview"
android:title="@string/ArView"
android:orderInCategory="0"
android:icon="@drawable/ic_menu_globe"
android:showAsAction="ifRoom|withText">
</item>

<itemandroid:id="@+id/directions"
android:title="@string/directions"
android:orderInCategory="1"
android:icon="@drawable/ic_menu_car"
android:showAsAction="never">
</item>

<itemandroid:id="@+id/cleardirections"
android:title="@string/menu_settings"
android:orderInCategory="2"
android:icon="@drawable/ic_menu_stop"
android:showAsAction="never">
</item>

<itemandroid:id="@+id/exit"
android:title="@string/exit"
android:orderInCategory="3"
android:icon="@drawable/ic_menu_exit"
android:showAsAction="never">
</item>

</menu>
```

Attributes.xml

```
<?xmlversion="1.0"encoding="utf-8"?>
<resources>
	<declare-styleablename="Gallery1">
		<attrname="android:galleryItemBackground"/>
	</declare-styleable>
</resources>
```

Strings.xml

```
<?xmlversion="1.0"encoding="utf-8"?>
<resources>

<stringname="app_name">Pocket Museum</string>
<stringname="MapView">Map View</string>
<stringname="ArView">AR View</string>
<stringname="menu_settings">Clear Path</string>
<stringname="directions">Get Directions</string>
<stringname="exit">Exit</string>
<stringname="cancle">Cancel</string>
<stringname="from">From</string>
<stringname="to">To</string>
<stringname="find">Find</string>
<stringname="empty"/>
<stringname="esties">University Dormitory</string>
<stringname="esties_short_discription">University of Cyprus
dormitory.</string>
<stringname="esties_discription">University of Cyprus dormitory.</string>
<stringname="thee01">Facility of Pure and Applied Science 01</string>
<stringname="thee01_short_discription">Facility for Mathematics and Computer
Science.</string>
<stringname="thee01_discription">In this building you can find the Department
of Computer Science and the Department of Mathematics and Statistics. Here
you can find many labs related to all aspects of Computer Science. there are
also the Mathematic labs. </string>
<stringname="thee02">Facility of Pure and Applied Science 02</string>
<stringname="thee02_short_discription">Facility for Physics and
Chemistry.</string>
<stringname="thee02_discription">In this building you can find the Department
of Physics and the Department of Chemistry. All the labs are equipped with
the best equipment, giving you the ability to perform a variety of
researches.</string>
<stringname="xod01">Teaching Facility 01</string>
<stringname="xod01_short_discription">Facilities designed for
teaching.</string>
<stringname="xod01_discription">A building dedicated in the ease of
teaching.</string>
<stringname="sports">Sports Center</string>
<stringname="sports_short_discription">University of Sports Center.</string>
<stringname="sports_discription">University of Cyprus dormitory.</string>

<string-arrayname="destinationsFrom">
<item>My Location</item>
<item>Dormitory</item>
<item>Facility of Pure and Applied Science 01</item>
<item>Facility of Pure and Applied Science 02</item>
<item>Teaching Facility 01</item>
<item>Sports Center</item>
</string-array>

<string-arrayname="destinationsTo">
<item>Dormitory</item>
<item>Facility of Pure and Applied Science 01</item>
<item>Facility of Pure and Applied Science 02</item>
<item>Teaching Facility 01</item>
<item>Sports Center</item>
```

```
</string-array>

</resources>
```

Styles.xml

```
<resources>

<stylename="AppTheme"parent="android:Theme.Holo"/>
<stylename="AppThemeDialog"parent="android:Theme.Holo.Dialog"></style>

</resources>
```