

Κεφάλαιο 8: Ανάπτυξη Περιεχομένου για Κινητές Συσκευές: Εργαστηριακές Ασκήσεις

Σύνοψη

Στο κεφάλαιο αυτό αναλύονται αρχικά οι κατηγορίες ανάπτυξης περιεχομένου για κινητές συσκευές και στη συνέχεια παρουσιάζεται ένα σύνολο εργαστηριακών ασκήσεων (εκφωνήσεις και οι υποδειγματικές λύσεις αυτών), με σκοπό την κατανόηση τεχνικών προγραμματισμού ανάπτυξης περιεχομένου για κινητές συσκευές, κυρίως από την πλευρά του πελάτη (*client-side scripting*). Οι επιμέρους τεχνολογίες που αξιοποιούνται είναι η τεχνολογία JavaScript με έμφαση στο πλαίσιο *mobile jQuery* για κατασκευή ιστοσελίδων *mobile Web*, οι 'διευκολύνσεις' του περιβάλλοντος ανάπτυξης υβριδικών εφαρμογών *PhoneGap* και η γλώσσα προγραμματισμού Java για εγγενείς κινητές εφαρμογές περιβάλλοντος *Android*.

Προαπαιτούμενη γνώση

Τα κεφάλαια 1, 5 και 6 του παρόντος συγγράμματος, και επιπλέον θα είναι χρήσιμη κάποια προηγούμενη εμπειρία σε ζητήματα Προγραμματισμού Υπολογιστών.

1. Εισαγωγή

Μπορούμε να διακρίνουμε (Γαβαλάς και άλλοι, 2015; Firtman, 2013) τρεις μεγάλες κατηγορίες περιεχομένου για κινητές συσκευές, ως προς τον τρόπο ανάπτυξής του: α) οι *ιστοσελίδες κινητού Ιστού* (*mobile Web*), οι οποίες χρησιμοποιούν τις κυρίαρχες τεχνολογίες ανάπτυξης περιεχομένου για τον παγκόσμιο Ιστό (κυρίως τις τεχνολογίες HTML5, CSS3, JavaScript και jQuery) και οι οποίες «προσφέρονται» στον κινητό χρήστη μέσω του προγράμματος περιήγησης που παρέχει η κινητή συσκευή (*mobile browser*), β) οι *υβριδικές κινητές εφαρμογές* (*hybrid mobile applications*), οι οποίες χρησιμοποιούν και αυτές τις ίδιες τεχνολογίες ανάπτυξης περιεχομένου που χρησιμοποιούν οι κινητές ιστοσελίδες της προηγούμενης κατηγορίας, αλλά διαφοροποιούνται στο ότι η σχεδίασή της διάδρασής τους δε στοχεύει στη χρήση του browser από τον κινητό χρήστη, διότι μέσω κάποιων επιπλέον διαδικασιών ελέγχου τελικά οι υβριδικές αυτές εφαρμογές 'συσκευάζονται' και διατίθενται ως κινητές εφαρμογές μέσω του λειτουργικού συστήματος της κινητής συσκευής, και γ) οι *γνήσιες, εγγενείς κινητές εφαρμογές* (*native mobile applications*), για τις οποίες η ανάπτυξη του κώδικα βασίζεται στο συγκεκριμένο λειτουργικό σύστημα που διαθέτει η κινητή συσκευή, και οι οποίες βέβαια, όπως και οι υβριδικές εφαρμογές της προηγούμενης κατηγορίας, διατίθενται ως κινητές εφαρμογές μέσω του λειτουργικού συστήματος της κινητής συσκευής.

1.1. Ιστοσελίδες κινητού Ιστού (*mobile Web development*)

Πλεονεκτήματα-μειονεκτήματα:

- Δεν υπάρχουν πολλές διαδικασίες και έλεγχοι για να γίνει διαθέσιμη μια εφαρμογή/ιστοσελίδα κινητού Ιστού. όπως συμβαίνει στις άλλες κατηγορίες εφαρμογών που στοχεύουν στη διάθεση μέσω του app store του λειτουργικού συστήματος της εφαρμογής. Φτιάχνουμε την εφαρμογή, την ανεβάζουμε και τρέχει χωρίς περιττές και χρονοβόρες διαδικασίες έγκρισης.
- Μπορούμε να έχουμε έναν κοινό κώδικα που εξυπηρετεί πολλές διαφορετικές εφαρμογές. Επομένως το κόστος ανάπτυξης και συντήρησης είναι μικρότερο.
- Το τελικό προϊόν δε θα είναι ποτέ τέλει όπως μια καθαρά native εφαρμογή. Δε θα υπάρχει ομαλότητα (*smoothness*) όπως συμβαίνει με τον native κώδικα που ταιριάζει κατάλληλα σε ένα κινητό με το αντίστοιχο λειτουργικό.

Τι είναι το *mobile Web* από την πλευρά του προγραμματιστή (*developer*);

Είναι ένα σύνολο βέλτιστων πρακτικών, πρότυπων σχεδίασης - ένα νέο 'πλαίσιο' που πρέπει να μάθει για να αναπτύσσει εφαρμογές για τον κινητό Ιστό (ή αλλιώς κινητές εφαρμογές Ιστού)

- Μερικές δυσκολίες που συναντάει ο προγραμματιστής που αναπτύσσει ιστοσελίδες για σταθερούς υπολογιστές και μετά πηγαίνει στο επόμενο βήμα της ανάπτυξης για τον κινητό Ιστό:
 - ο Πιο αργά δίκτυα με υψηλότερη λανθάνουσα κατάσταση
 - ο Πιο αργό υλικό και λιγότερη διαθέσιμη μνήμη
 - ο Διαφορετική εμπειρία περιήγησης
 - ο Διαφορετικά πλαίσια χρηστών
 - ο Διαφορετική συμπεριφορά περιηγητών
 - ο Πάρα πολλά κινητά στην αγορά με διαφορετικούς περιηγητές, με διαφορετικές εκδοχές κατά την ίδια στιγμή

Μύθοι για το Mobile Web:

- **Ο κινητός Ιστός είναι απλώς το Web:** ο κινητός Ιστός μπορεί να χρησιμοποιεί τα ίδια πρωτόκολλα δικτύου όπως το κλασικό Διαδίκτυο (HTTP, HTTPS, POP3, Wireless LAN ακόμα και TCP/IP). Μπορεί τα πρωτόκολλα υποδομής να είναι τα ίδια με το κλασικό Web. Όταν αναπτύσσουμε όμως κώδικα για εφαρμογές mobile Web στοχεύουμε σε πολύ διαφορετικές συσκευές σε σχέση με αυτές που χρησιμοποιούν το κλασικό Web. Ένα ευρέως γνωστό χαρακτηριστικό είναι το μήκος της οθόνης αλλά υπάρχουν πολλές άλλες όχι και τόσο εμφανείς διαφορές.
- **Δε χρειάζεται να κάνετε κάτι ιδιαίτερο στην επιτραπέζια ιστοσελίδα σας για τους κινητούς χρήστες:** πλέον αρκετές ιστοσελίδες είναι διαθέσιμες και στο mobile Web όπως είναι στο κλασικό Web για το οποίο έχουν φτιαχτεί. Ωστόσο είναι δύσχρηστες γιατί οι χρήστες θα πρέπει να κάνουν συνεχώς zoom in και zoom out και η πλοήγηση γίνεται πολύ δύσκολη. Έτσι υπάρχουν οι εφαρμογές που αναπτύσσονται καθαρά για mobile Web και διευκολύνουν την εμπειρία του χρήστη.
- **Μια ιστοσελίδα θα πρέπει να λειτουργεί για όλες τις συσκευές:** υπάρχουν τεχνικές που μας επιτρέπουν να δημιουργήσουμε μόνο ένα αρχείο, αλλά αυτό το αρχείο θα παρέχει διαφορετικές εμπειρίες σε μια ποικιλία συσκευών, συμπεριλαμβανομένων των επιτραπέζιων υπολογιστών, κινητών τηλεφώνων, ταμπλετών, τηλεοράσεων και κονσόλων παιχνιδιών. Το όραμα αυτό ονομάζεται "one Web" και υπάρχει η τεχνική που ονομάζεται Responsive Web Design που έχει ως στόχο να το επιτύχει. Ωστόσο μέχρι τώρα δεν έχει καταστεί πλήρως εφικτό και έτσι δεν έχουν εκλείψει εντελώς οι λόγοι που οδηγούν στο ότι χρειάζεται αρκετές φορές να αναπτύσσεται διαφορετικός κώδικας για κάθε πλατφόρμα.
- **Απλά να δημιουργήσετε ένα αρχείο HTML με πλάτος 320 pixels και έχετε μια κινητή ιστοσελίδα:** Αυτός είναι ο άλλος τρόπος fast-food για να σκεφτεί κανείς το mobile Web. Σήμερα, υπάρχουν περισσότερες από 3.000 κινητές συσκευές στην αγορά με σχεδόν 30 διαφορετικά προγράμματα περιήγησης (στην πραγματικότητα περισσότερα από 300 διαφορετικά προγράμματα περιήγησης αν τα χωρίσουμε σύμφωνα με τον αριθμό έκδοσης). Δημιουργώντας ένα αρχείο HTML ως κινητή ιστοσελίδα θα είναι ένα αποτυχημένο σχέδιο. Επιπλέον, το γεγονός αυτό συμβάλλει στην πεποίθηση ότι η κινητή περιήγηση στο Διαδίκτυο είναι οδυνηρή.
- **Οι εγγενείς κινητές (native mobile) εφαρμογές θα 'σκοτώσουν' το mobile Web:** Κάθε λύση έχει πλεονεκτήματα και μειονεκτήματα. Το κινητό Διαδίκτυο έχει πολλά να προσφέρει με τις εγγενείς εφαρμογές. Από την άλλη μεριά, το mobile Web (και η σχετικά νέα έννοια της υβριδικής εφαρμογής - native Web app σε αντιδιαστολή του όρου native mobile app), μας προσφέρει μια μεγάλη πλατφόρμα πολλαπλών εφαρμογών, συμπεριλαμβανομένων και τοπικών εφαρμογών που δεν απαιτούν πάντα σύνδεση στο Διαδίκτυο με διευθύνσεις URL και προγράμματα περιήγησης.
- **Οι άνθρωποι δε χρησιμοποιούν τους κινητούς περιηγητές τους:** Οι συνδέσεις στο Διαδίκτυο σε όλο τον κόσμο είναι: 3,035,749,340 (42% του παγκόσμιου πληθυσμού)

[τελευταία στοιχεία: 30/6/2014, <http://www.internetworldstats.com/stats.htm>]
Οι συνδρομές για κινητές συσκευές είναι: 7,000,000,000 (υπολογίζονται στο τέλος του 2014, U.N. Telecommunications Agency, <http://www.un.org/apps/news/story.asp?NewsID=47729#.VPd5aC7Vqhk>)
Προσοχή: 2, 300,000,000 mobile broadband συνδέσεις, 32% του παγκ. πληθυσμού, υπολογίζονται στο τέλος του 2014, http://www.itu.int/net/pressoffice/press_releases/2014/23.aspx#.VPeAZy7Vqhl **ΤΕΛΙΚΑ:** Οι mobile λογαριασμοί αποτελούν το 25% της όλης χρήσης του Web (2014) (<http://www.smartinsights.com/digital-marketing-strategy/internet-trends-2014-mary-meeker/>)

1.2. Υβριδικές κινητές εφαρμογές (hybrid ή cross - platform development)

Πλεονεκτήματα-μειονεκτήματα:

- Απλή λογική: γράφω σε HTML5 (και κατ' επέκταση σε CSS3 και JavaScript κλπ.) αλλά το 'συσκευάζω' ως native εφαρμογή.
- Η εφαρμογή πρέπει να περάσει από κάποιες διαδικασίες ελέγχου σε σύγκριση με τις εφαρμογές/ιστοσελίδες mobile Web της προηγούμενης κατηγορίας.
- Ένα μεγάλο πλεονέκτημα είναι ότι γράφουμε μια φορά τον κώδικά μας και μπορούμε να τον τρέξουμε σε πολλές πλατφόρμες. Αυτό συνεπάγεται μειωμένο κόστος ανάπτυξης.
- Η πρόσβαση σε εγγενείς δυνατότητες είναι καλύτερη από ότι αυτές που υπάρχουν στην κατηγορία των mobile Web ιστοσελίδων.
- Το τελικό προϊόν που παράγεται ως εφαρμογή δεν είναι πάλι τέλειο (σε σύγκριση με μια εγγενή κινητή εφαρμογή), αλλά είναι αισθητικά καλύτερο από ότι θα ήταν το αποτέλεσμα με χρήση mobile Web ιστοσελίδας.

1.3. Εγγενείς κινητές εφαρμογές (native mobile development)

Πλεονεκτήματα-μειονεκτήματα:

- Η πρόσβαση σε εγγενείς δυνατότητες είναι καλύτερη σε σύγκριση με τις άλλες κατηγορίες.
- Όταν αναφερόμαστε σε εγγενείς δυνατότητες σημαίνει ότι χρησιμοποιώντας native development αξιοποιούμε καθαρά την ανάπτυξη του κώδικα της εφαρμογής μας βασισμένη στο λειτουργικό σύστημα που τρέχει στο κινητό (π.χ. iPhone - iOS ανάπτυξη κώδικα με την ObjectiveC, Android ανάπτυξη κώδικα με Java, Windows ανάπτυξη κώδικα με .NET κλπ.). Τα πλεονεκτήματα είναι εμφανώς μεγαλύτερα γιατί το τελικό προϊόν είναι αισθητικά καλύτερο (τα χρώματα, σχεδίαση κουμπιών κλπ.) και αξιοποιεί όλες τις δυνατότητες της συσκευής.
- Το κόστος άδειας ανάπτυξης κώδικα είναι υψηλό (εξαιρέση το Android).
- Αν θέλουμε να κάνουμε την εφαρμογή μας διαθέσιμη σε περισσότερα από ένα λειτουργικά συστήματα ο κώδικας της εφαρμογής θα πρέπει να γραφτεί καθαρά για κάθε εφαρμογή στη γλώσσα ανάπτυξης που υπάρχει. Το κόστος είναι πολύ υψηλό, απαιτούνται ειδικευμένοι προγραμματιστές και το κόστος συντήρησης είναι επίσης πολύ υψηλό.
- Όταν ολοκληρωθεί η εφαρμογή θα πρέπει να περάσει όλες τις διαδικασίες και τους ελέγχους των app stores.

1.4. Σύγκριση

Ακολουθεί πίνακας με σύγκριση των τριών αυτών κατηγοριών περιεχομένου για κινητές συσκευές, βασισμένη στα χαρακτηριστικά αυτών:

	Mobile Web	Hybrid	Native
Γραφικά	HTML5, CSS3, JavaScript	HTML, CSS3, JavaScript	Native APIs
Εκτέλεση	Αργή	Όχι τόσο αργή	Γρήγορη
Native εμφάνιση και αίσθηση	Απομίμηση	Απομίμηση	Native
Διανομή	Διαδίκτυο	App Store	App Store
Συνδεσιμότητα	Κυρίως εντός σύνδεσης	Εντός και εκτός σύνδεσης	Εντός και εκτός σύνδεσης
Δεξιότητες Ανάπτυξης	HTML5, CSS3, JavaScript	HTML5, CSS3, JavaScript	Objective C, Java
Πρόσβαση συσκευής:			
Φωτογ. Μηχανή		√	√
Ειδοποιήσεις		√	√
Επαφές, Ημερολόγιο		√	√
Αποθήκευση χωρίς σύνδεση	Κοινόχρηστη SQL	Ασφαλής αποθήκευση αρχείων, Κοινόχρηστη SQL	Ασφαλής αποθήκευση αρχείων
Γεωγραφικοί χάρτες	√	√	√
Χειρονομίες:			
Άγγιγμα	√	√	√
Τσίμπημα, Διαδοχική επαφή		√	√

Πίνακας 8.1 Διαφορές μεταξύ κατηγοριών περιεχομένου για κινητές συσκευές

2. Ανάπτυξη ιστοσελίδων κινητού Ιστού

Όπως αναφέραμε προηγουμένως, οι mobile εφαρμογές για διαφορετικές συσκευές αναπτύσσονται χρησιμοποιώντας τελείως διαφορετικές γλώσσες προγραμματισμού. Για παράδειγμα, μια Android εφαρμογή είναι γραμμένη σε Java, μια iOS εφαρμογή είναι γραμμένη σε Objective-C, και σε περιβάλλον .NET αναπτύσσεται μια Windows Phone εφαρμογή. Κάθε εταιρεία έχει το δικό της λειτουργικό σύστημα με αποτέλεσμα να μην υπάρχει φορητότητα για τη χρήση όλων των mobile εφαρμογών σε όλες τις πλατφόρμες και συσκευές. Με τις τεχνολογίες HTML5, CSS3 και JavaScript μπορούμε να δημιουργήσουμε mobile applications χωρίς να υπάρχει αυτός ο περιορισμός. Μια εφαρμογή η οποία είναι γραμμένη με τις τρεις παραπάνω γλώσσες είναι το ίδιο λειτουργική τόσο σε μια Android όσο και σε μια iOS ή οποιουδήποτε άλλου λειτουργικού συστήματος συσκευή. Αυτό συμβαίνει διότι οι Web browsers, οι οποίοι δουλεύουν με αυτές τις Web γλώσσες, είναι συμβατοί και χρησιμοποιούνται από κάθε συσκευή και λειτουργικό σύστημα.

2.1. JavaScript και jQuery

Η χρήση της JavaScript από τους web browsers επιτρέπει την αλληλεπίδραση μεταξύ των client-side scripts και των χρηστών. Επίσης ελέγχει τους browsers, και μέσω ασύγχρονης επικοινωνίας κρατάει ενημερωμένο το Document Object Model (DOM) των ιστοσελίδων. Το DOM είναι μια πλατφόρμα μέσω της οποίας

αναπαριστάνονται όλα τα HTML (και γενικότερα όλα τα XML) αντικείμενα. Η αναπαράσταση των αντικειμένων αυτών αποτελείται από κόμβους οι οποίοι χωρίζονται σε δενδροειδή μορφή η οποία ονομάζεται DOM tree.

Η JavaScript είναι μια δυναμική γλώσσα προγραμματισμού η οποία έχει πολλές δυνατότητες κυρίως στην κατασκευή ιστοσελίδων και σε ότι έχει να κάνει με διαδικτυακές εφαρμογές. Θα λέγαμε ότι είναι η γλώσσα του Internet όσον αφορά την πλευρά του πελάτη (client-side scripting).

Αυτό που πετυχαίνει η JavaScript είναι ότι μπορεί να ελέγξει όλα τα στοιχεία του DOM μια ιστοσελίδας και να τα πυροδοτήσει προκαλώντας διάφορα συμβάντα. Η εξέλιξη του Διαδικτύου ανέβασε τις προσδοκίες για ένα πιο λειτουργικό Web, τόσο από την πλευρά του χρήστη όσο και από την πλευρά του developer. Αυτή τη στιγμή υπάρχουν πολλές βιβλιοθήκες οι οποίες είναι γραμμένες σε JavaScript και χρησιμοποιούνται αποκλειστικά για τη διευκόλυνση χρήσης των html στοιχείων μέσω του DOM. Μια τέτοια βιβλιοθήκη είναι η jQuery.

Η jQuery χρησιμοποιεί JavaScript κώδικα για να διευκολύνει τη διαχείριση των html στοιχείων, δημιουργεί πολύπλοκα animations, effect και ελέγχει events τα οποία χρησιμοποιούνται για τη δημιουργία δυναμικών ιστοσελίδων και web εφαρμογών. Επίσης μπορεί να χρησιμοποιηθεί για την ανάπτυξη AJAX εφαρμογών. Η JavaScript ελέγχει τα συμβάντα τα οποία ενεργοποιούνται από εξωτερικές συσκευές όπως είναι το ποντίκι και το πληκτρολόγιο, καθώς αναφερόμαστε σε desktop περιβάλλοντα. Βέβαια τα συμβάντα αυτά δεν περιορίζονται μόνο στις φυσικές ενέργειες του χρήστη αλλά και σε ενέργειες που αφορούν τη διαδραστικότητα του χρήστη σε μια ιστοσελίδα. Τα κυριότερα εξ' αυτών είναι τα Drag & Drop events, media, animation, frame και form events.

2.2. Mobile frameworks: mobile jQuery

Υπάρχουν πολλά frameworks τα οποία δίνουν τη δυνατότητα ανάπτυξης εφαρμογών χρησιμοποιώντας μόνο τις HTML5, CSS3 και JavaScript. Μια από αυτές είναι η mobile jQuery, η οποία είναι βασισμένη στην jQuery. Είναι ένα interface το οποίο διαχειρίζεται events των mobile συσκευών για τη δημιουργία mobile εφαρμογών (Matthews & Gliser, 2015; Firtman, 2012). Συμβάντα τέτοιου τύπου είναι τα tap, touch, scroll και orientation.

2.3. Ενδεικτικό παράδειγμα κώδικα σε mobile jQuery

Εκφώνηση:

Σε αυτή την εργαστηριακή άσκηση θα δούμε κάποια βασικά στοιχεία χρήσης του πλαισίου mobile jQuery για την κατασκευή mobile Web εφαρμογής. Στοιχεία, όπως η κεφαλίδα, το υποσέλιδο και οι σύνδεσμοι (προς μια εξωτερική σελίδα ή προς ένα άλλο τμήμα της ίδιας σελίδας, προς μια 'σελίδα' μέσα στη σελίδα).

Αρχείο mjQuery1.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Πολλαπλές Σελίδες</title>
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.css" />
    <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.js"></script>
  </head>

  <body>
    <div data-role="page" id="homePage">
      <div data-role="header"><h1>Καλημέρα</h1></div>
```

```

    <div data-role="content">
      <p> Welcome to our first mobile web site. It's going to be the best site you've ever seen. Once we
get some content. And a business plan. But the hard part is done!
    </p>
    <p>
      Find out about our wonderful <a href="tmimata.html" data-prefetch="true" data-
transition="pop" data-direction="reverse">departments</a>.
    </p>
    <p>
      You can also <a href="#aboutPage">learn more</a> about Πανεπιστήμιο Μακεδονίας.
    </p>
    </div>

    <div data-role="footer">
      <h4>Copyright UoM 2015</h4>
    </div>
</div>

<div data-role="page" id="aboutPage" data-title="ΠΑΜΑΚ">
  <div data-role="header"><h1>Σχετικά με ΠΑΜΑΚ</h1></div>

  <div data-role="content">
    <p>
      This text talks about ΠΑΜΑΚ and how interesting it is. Most likely though you want to <a
href="#homePage">return</a> to the home page.
    </p>
  </div>

  <div data-role="footer">
    <h4>Copyright UoM 2015</h4>
  </div>
</div>
</body>
</html>

```

Αρχείο *tmimata.html*

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Τμήματα ΠΑΜΑΚ</title>
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.css" />
    <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.js"></script>
  </head>

  <body>
    <div data-role="page" id="deptsPage">
      <div data-role="header"><h1>Departments</h1></div>
      <div data-role="content">
        <p>
          Our departments include:

```

```

</p>
<ul>
  <li>Εφαρμοσμένη Πληροφορική</li>
  <li>Οργάνωση και Διοίκηση Επιχειρήσεων</li>
  <li>Χρηματοοικονομικά και Λογιστική</li>
</ul>
</div>
</div>
</body>
</html>

```

2.4. Παράδειγμα φόρμας με κώδικα σε mobile jQuery

Εκφώνηση:

Δημιουργήστε μια φόρμα με το όνομα `mob_form.html`, που να έχει την ακόλουθη μορφή, κάνοντας χρήση του πλαισίου mobile jQuery. Στη φόρμα αυτή, τα πεδία ονοματεπώνυμο και email είναι απαραίτητα (κάντε τον έλεγχο αυτό καθώς και τον έλεγχο ορθότητας της εισαγωγής στο πεδίο email, με απλό τρόπο (μέσω ιδιοτήτων των ετικετών της φόρμας). Το πάτημα του πλήκτρου Υποβολή, όταν είναι εντάξει οι προηγούμενοι έλεγχοι, μας οδηγεί στη σελίδα `mob_stoixeia.php`, που εμφανίζει (κάνοντας χρήση `php` κώδικα) ότι έβαλε ο χρήστης στα πεδία της φόρμας. Δείτε την παρακάτω εικόνα. Σημειώστε ότι το δεύτερο κουμπί ‘καθαρίζει’ τις όποιες επιλογές/εισαγωγές στοιχείων έκανε ο χρήστης στη φόρμα.

Αρχείο `mob_form.html`

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Φόρμα</title>
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.css" />
    <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.js"></script>
  </head>

  <body>

    <div data-role="page" id="homePage">
      <div data-role="header"><h1>Φόρμα Επικοινωνίας</h1></div>
      <div data-role="content">
        <h3>Συμπληρώστε παρακαλώ τα παρακάτω στοιχεία</h3>
        <form name="gbForm" method = "post" action = "mob_stoixeia.php">
      <div>Ονοματεπώνυμο:</div>
        <div> <input type="text" name="name" size="50" required /></div>
        <br/>
      <div>Ηλικία (διαλέξτε ένα):</div>
        <div data-role="fieldcontain">
          <fieldset data-role="controlgroup">
            <input type="radio" name="age" id="OS30" value="Εως 30"/>
            <label for="OS30">Από 18 έως 30</label>

            <input type="radio" name="age" id="OS50" value="Εως 50" />
            <label for="OS50">Από 31 έως 50</label>
          </fieldset>
        </div>
      </div>
    </div>
  </body>
</html>

```

```

<input type="radio" name="age" id="OLDER" value="Πάνω από 50"/>
<label for="OLDER">Από 51 και πάνω</label>

</fieldset>
</div>
<br/>

```

Εικόνα 8.1 Παράδειγμα φόρμας επικοινωνίας

```

<div>Περιοχή που σας ενδιαφέρει:</div>
<div>
  <select name="region">
    <option value="Μακεδονία" selected="selected">Μακεδονία</option>
    <option value="Κυκλάδες">Κυκλάδες</option>
    <option value="Θράκη">Θράκη</option>
    <option value="Άλλη">Άλλη</option>
  </select>

```

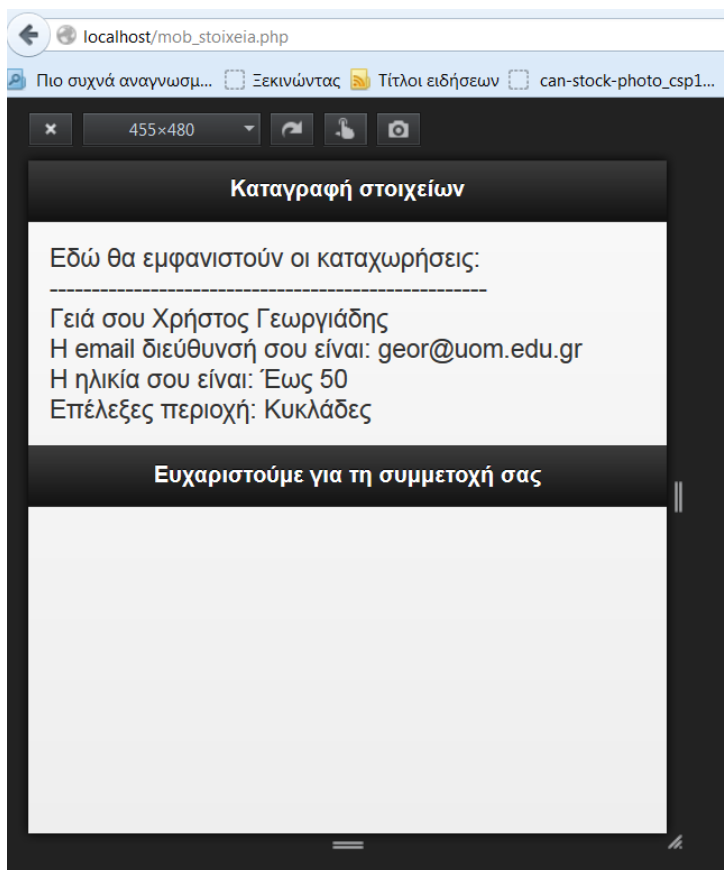


```

</div>
<br/>

<div>e-mail επικοινωνίας:</div>
<div ><input type="email" name="email" size="50" required /></div>
<br/>
<div style="clear: left;">
  <input type="submit" name = "submit" value="Υποβολή - Δείξε στοιχεία" />
  <input type="reset" value="Καθάρισμα" />
</div>
</form>
</div>
<div data-role="footer">
  <h4>Ευχαριστούμε για τη συμμετοχή σας</h4>
</div>
</div>
</body>
</html>

```



Εικόνα 8.2 Εμφάνιση των καταχωρήσεων σε φόρμα

Αρχείο *mob_stoixeia.php*

```

<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.css" />

```

```

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
<script src="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.js"></script>
<meta charset="utf-8"/>
<title>Apotelesmata</title>
<style>
    div#problemArea {border:solid 2px red; margin:7px;
        padding:5px;
        background-color:#000000;
        color: white;
        float:left;
        width:40%;}
</style>
</head>
<body>
    <div data-role="page">
        <div data-role="header">
            <h1>Καταγραφή στοιχείων</h1>
        </div>
        <div data-role="content">
            <div id="problemArea"> Εδώ θα εμφανιστούν οι καταχωρήσεις:
                <br/>----- </div>
                Γεία σου <?php echo $_POST["name"]; ?><br/>
                Η email διεύθυνσή σου είναι: <?php echo $_POST["email"]; ?><br/>
                Η ηλικία σου είναι: <?php echo $_POST["age"]; ?><br/>
                Επέλεξες περιοχή: <?php echo $_POST["region"]; ?><br/>
            </div>
            <div data-role="footer">
                <h4>Ευχαριστούμε για τη συμμετοχή σας</h4>
            </div>
        </div>
    </body>
</html>

```

3. Ανάπτυξη υβριδικών κινητών εφαρμογών

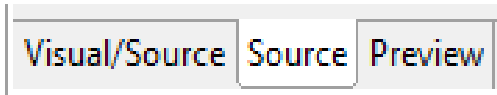
Ένα από τα πιο διαδεδομένα περιβάλλοντα ανάπτυξης υβριδικών κινητών εφαρμογών είναι το αποκαλούμενο Apache Cordova PhoneGap. Είναι ένα περιβάλλον προγραμματισμού που βασίζεται στο πολύ γνωστό περιβάλλον Eclipse (Eclipse JBoss).

3.1. Εισαγωγή

Σε αυτή την εργαστηριακή άσκηση θα αναφερθούμε στα βασικά στοιχεία του περιβάλλοντος PhoneGap, για την κατασκευή μιας υβριδικής εφαρμογής.

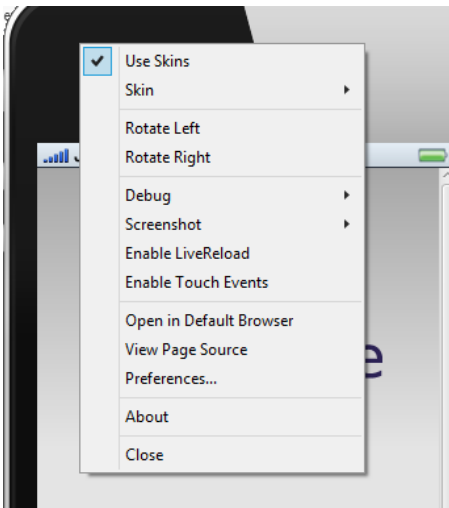
1. Δημιουργούμε το πρώτο μας project (File → New → Other → Mobile → Hybrid ...). Βλέπουμε τα αρχεία που δημιουργούνται - τις βασικές αλλαγές τις κάνουμε αρχικά στο αρχείο index.html.
2. Ανοίγουμε το αρχείο config.xml - εκεί υπάρχει το όνομα του project μας και ρυθμίσεις
3. Πηγαίνοντας πάλι στο αρχείο index.html κάτω στα tabs βλέπουμε:
 - o Visual / Source που έχει και τις δυο οπτικές τόσο του development όσο και της εκτέλεσης της εφαρμογής (πώς φαίνεται στο front-end).

- Source: βλέπουμε καθαρά μόνο τον κώδικα (αυτή είναι η κατάσταση που θα χρησιμοποιούμε γιατί τα αποτελέσματα θα τα βλέπουμε μέσα από τη χρήση προσομοιωτών)
- Preview: βλέπουμε μόνο την οθόνη εκτέλεσης της εφαρμογής



Εικόνα 8.3 Καρτέλες προβολής περιεχομένου στο αρχείο *index.html*

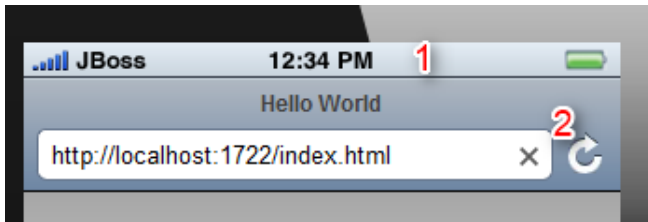
4. Ενδεικτικά, για να δούμε πώς φαίνεται και το αρχικό παράδειγμα της εφαρμογής μας πάμε και κάνουμε δεξί κλικ στο συνολικό μας project και μετά Run As και επιλέγουμε τον CordovaSim. Η κατασκευή αυτού του emulator βασίζεται ουσιαστικά στον Ripple emulator αλλά έχει προσαρμοστεί για να φαίνεται ως συσκευή.
5. Αυτή τη συσκευή που βλέπουμε, αν θέλουμε μπορούμε να την προσαρμόσουμε στις δικές μας προτιμήσεις. Έστω ότι κάνουμε δεξί κλικ επάνω στη συσκευή και βλέπουμε την παρακάτω λίστα:



Εικόνα 8.4 Λίστα παραμετροποίησης εικονικής συσκευής

6. Αν αφαιρέσουμε το tick από το skin θα δούμε ότι ουσιαστικά αυτός ο προσομοιωτής τρέχει σε localhost. Για να επαναφέρουμε το Skin πάμε Device → Use Skin.
7. Μετά μπορούμε να χρησιμοποιήσουμε το skin για να επιλέξουμε μια άλλη συσκευή. Ενδεικτικά, πειραματιστείτε με κάποιες αλλαγές.
8. Επίσης μπορούμε να εφαρμόσουμε και την περιστροφή χρησιμοποιώντας το Rotate Left ή Rotate Right.
9. Αφήνουμε ανοικτό τον προσομοιωτή, τον τακτοποιούμε σε μια άκρη της οθόνης και πάμε στη σελίδα *index.html*. Ρυθμίζουμε το JBoss παράθυρο να μην τον καλύπτει, και πάμε για να κάνουμε μια μικρή αλλαγή στο *h1*: αντί για το υπάρχον περιεχόμενο `<h1>Apache Cordova application powered by Eclipse ...</h1>`, βάζουμε ένα δικό μας κείμενο έστω “Η πρώτη μας εφαρμογή”.
10. Αποθηκεύουμε το περιεχόμενο της σελίδας και πάμε στον προσομοιωτή που έχουμε ανοικτό και πατάμε ένα κλικ επάνω στην μπάρα που γράφει JBoss μια φορά και βλέπουμε ότι

εμφανίζεται μια γραμμή με τη διεύθυνση και πατάμε του κουμπί της ανανέωσης για να δούμε την αλλαγή μας.



Εικόνα 8.5 Διαδικασία προβολής αλλαγών στην εφαρμογή μας

11. Ωστόσο μπορούμε να βλέπουμε τις αλλαγές μας και με μεγαλύτερη ευκολία: δηλαδή, αμέσως μόλις σώζουμε τις αλλαγές να ενημερώνεται αυτόματα ο προσομοιωτής χωρίς να χρειάζεται να ζητάμε τη διαδικασία της ανανέωσης. Αυτό μπορούμε να το καταφέρουμε αν χρησιμοποιήσουμε τον LiveReload Sever που τρέχει τοπικά. Για να τον δούμε πάμε στο tab Servers. Βλέπουμε εκεί ότι υπάρχει ο server. ΠΡΟΣΟΧΗ: Αν δε βλέπουμε το tab του server, πάμε (στο περιβάλλον του eclipse) από το επάνω οριζόντιο μενού Window → Show View → Other → Server. Κάνουμε expand και επιλέγουμε “Servers”.
12. Βλέπουμε ότι είναι σε stopped mode - για να τον ενεργοποιήσουμε πάμε και κάνουμε δεξί κλικ πάνω στον server και πατάμε start.
13. Χρειάζεται ένα ακόμα βήμα για να αρχίσει ο LiveReload Server να αλληλοεπιδρά με τον simulator μας. Πάμε στον προσομοιωτή μας και κάνουμε δεξί κλικ και επιλέγουμε Enable LiveReload.
14. Για να δούμε ότι λειτουργεί με επιτυχία πάμε να κάνουμε μια αλλαγή, έστω πάλι στο <h1> και να γράψουμε πχ. “η πρώτη δοκιμή με τον LiveReload server”. Αποθηκεύουμε την αλλαγή μας και βλέπουμε ότι η οθόνη του προσομοιωτή ενημερώθηκε αυτόματα.
15. Προσέξτε ότι στην πρώτη μας εφαρμογή υπάρχει μια ένδειξη που δείχνει “Device is Ready”. Όταν κάνουμε μια αλλαγή και την αποθηκεύουμε, στο παράθυρο Console καταγράφεται το «!JavaScript LOG: Received Event: deviceready» ενώ επίσης η ένδειξη στον προσομοιωτή προς στιγμή αναγράφει “Connection to Device”.
16. Αυτό οφείλεται σε έναν συνδυασμό κώδικα events στην JavaScript: στέλνεται ένα μήνυμα για να εξεταστεί ότι η συσκευή είναι ενεργή και αν αυτή είναι ενεργή επιστρέφει το αντίστοιχο μήνυμα. Πιο συγκεκριμένα στο index.html βλέπουμε ότι υπάρχει:

```
<div id="deviceready" class="blink">
  <p class="event listening">Connecting to Device</p>
  <p class="event received">Device is Ready</p>
</div>
```

Όταν αποθηκεύουμε το περιεχόμενό μας, συμβαίνει το event listening και στον προσομοιωτή μας καταγράφεται το Connecting to Device. Αφού επιστραφεί η τιμή true ότι η συσκευή μας είναι ενεργή, εκτελείται το event στην JavaScript “received” και έτσι εμείς βλέπουμε το μήνυμα “Device is Ready”.

17. Όλα αυτά τα events στην JavaScript που περιγράψαμε στο προηγούμενο βήμα, έχουν δημιουργηθεί στον φάκελο index.js που βρίσκεται μέσα στον φάκελο js. Στο αρχείο μας index.html συνδέεται με τη χρήση του κώδικα

```
<script type="text/javascript" src="js/index.js"></script>
```

Παρατηρήστε το κώδικα .js. Κάντε στη συνέχεια μια προσθήκη (εντολή console.log('Event listening: '); μέσα στην onDeviceReady), ώστε να έχουμε την παρακάτω μορφή:

```
...
onDeviceReady: function() {
    app.receiveEvent('deviceready');
    console.log('Event listening: ');
}
...
```

Τι άλλαξε στη συμπεριφορά της εφαρμογής;

18. Υπάρχει ο φάκελος res → icon που περιέχει την εικόνα έναρξης όταν ξεκίνησε η εικόνα μας. Εκεί αν φτιάχνουμε μια δικιά μας εφαρμογή μπορούμε να τοποθετούμε τις δικές μας εικόνες.
19. Παρατηρούμε ξανά το αρχείο index.html. Βλέπουμε ότι υπάρχει η μορφή που γνωρίζουμε για την κατασκευή σελίδων html. Υπάρχει το <title>, υπάρχει και η σύνδεση με ένα εξωτερικό css αρχείο index.css που βρίσκεται στον φάκελο css:

```
<link rel="stylesheet" type="text/css" href="css/index.css" />
```

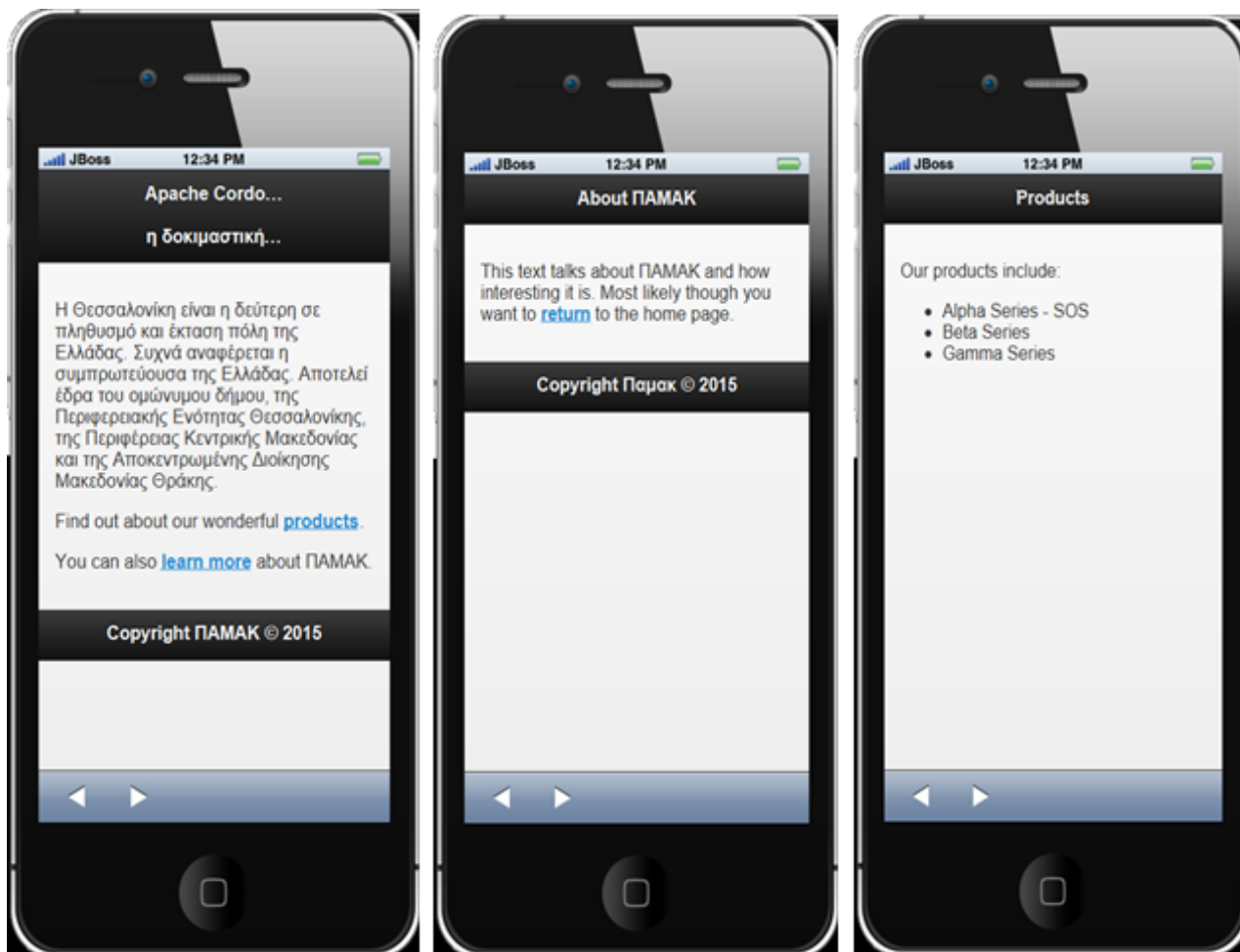
20. Βλέπουμε ότι το index.css δίνει τη μορφή που βλέπουμε στη συσκευή μας.
21. Αν θέλουμε να έχουμε έλεγχο στις μορφοποιήσεις css, στο index.css σβήνουμε από το body όλο το περιεχόμενο εκτός από το ακόλουθο:
body {
-webkit-touch-callout: none; /* prevent callout to copy image, etc when tap to hold */
-webkit-text-size-adjust: none; /* prevent webkit from resizing text to fit */
-webkit-user-select: none; /* prevent copy paste, to allow, change 'none' to 'text' */
}

3.2. Εσωτερικοί και εξωτερικοί σύνδεσμοι

Σε αυτή την εργαστηριακή άσκηση θα δούμε κάποια βασικά στοιχεία στην κατασκευή υβριδικής εφαρμογής, όπως η κεφαλίδα, το υποσέλιδο και οι σύνδεσμοι (προς μια εξωτερική σελίδα ή προς ένα άλλο τμήμα της ίδιας σελίδας, προς μια 'σελίδα' μέσα στη σελίδα). Προσέξτε την αναμενόμενη πληθώρα των 'κοινών στοιχείων' ανάμεσα στον απαιτούμενο κώδικα κατά την ανάπτυξη υβριδικής εφαρμογής και τον απαιτούμενο κώδικα για την ανάπτυξη ιστοσελίδας κινητού Ιστού.

1. Στο index.html πηγαίνουμε στο <div class="app"> και αλλάζουμε τον τίτλο της εφαρμογής μας, πχ. <h1>Apache Cordova

 η δοκιμαστική εφαρμογή μου</h1>



Εικόνα 8.6 Χρήση εσωτερικών και εξωτερικών συνδέσμων

2. Κάνουμε τις απαραίτητες προσθήκες/διαγραφές ώστε να φτάσουμε στο ακόλουθο περιεχόμενο:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.css" />
    <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.js"></script>
    <title>Δοκιμές σε PhoneGap</title>
  </head>
  <body>
    <div class="app" data-role="page" id="homePage">
      <div data-role="header">
        <h1>Apache Cordova <br/><br/> η δοκιμαστική εφαρμογή μου</h1>
      </div>
      <div data-role="content">
        <p>
          Η Θεσσαλονίκη είναι η δεύτερη σε πληθυσμό και έκταση πόλη της
          Ελλάδας. Συχνά αναφέρεται η συμπρωτεύουσα της Ελλάδας. Αποτελεί
        </p>

```

έδρα του ομώνυμου δήμου, της Περιφερειακής Ενότητας Θεσσαλονίκης, της Περιφέρειας Κεντρικής Μακεδονίας και της Αποκεντρωμένης Διοίκησης Μακεδονίας Θράκης.

```
</p>
<p>
  Find out about our wonderful <a href="products.html"
    data-prefetch="true" data-transition="pop"
    data-direction="reverse">products</a>.
</p>
<p>
  You can also <a href="#aboutPage">learn more</a> about ΠΑΜΑΚ.
</p>

</div>
<div data-role="footer">
  <h4>Copyright ΠΑΜΑΚ &copy; 2015</h4>
</div>
</div>
```

```
<div data-role="page" id="aboutPage">

  <div data-role="header"><h1>About ΠΑΜΑΚ</h1></div>

  <div data-role="content">
    <p>
      This text talks about ΠΑΜΑΚ and how interesting it is. Most likely
      though you want to <a href="#homePage">return</a> to the home page.
    </p>
  </div>

  <div data-role="footer">
    <h4>Copyright Παμακ &copy; 2015</h4>
  </div>
</div>
</body>
</html>
```

3. Παρατηρήστε την ενσωμάτωση των css κανόνων μορφοποίησης mobile jQuery και των βιβλιοθηκών jQuery και mobile jQuery με τις εντολές:

```
<link rel="stylesheet"
  href="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.css"/>
<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
<script src="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.js"></script>
```

4. Παρατηρήστε τη χρήση ετικετών <meta> για τα Ελληνικά και τη ρύθμιση για mobile συσκευές:

```
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1">
```

5. Παρατηρήστε τα τμήματα <div data-role="header">, <div data-role="content"> και <div data-role="footer"> που ορίζουν κεφαλίδα, περιεχόμενο και υποσέλιδο σε mobile web pages.

6. Παρατηρήστε τα τμήματα `<div data-role="page" id="homePage">` και `<div data-role="page" id="aboutPage">` που ορίζουν δυο διαφορετικές σελίδες μέσα σε μία σελίδα html.
7. Παρατηρήστε πώς με απλές ετικέτες υπερσύνδεσης, `<a href="#aboutPage">` και `<a href="#homePage">` γίνεται η ανακατεύθυνση προς και από τέτοιες σελίδες.
8. Η σελίδα products.html έχει το παρακάτω περιεχόμενο:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Products</title>
  </head>
  <body>
    <div data-role="page" id="productsPage">

<div data-role="header"><h1>Products</h1></div>

<div data-role="content">
<p>
    Our products include:
</p>
<ul>
    <li>Alpha Series - SOS</li>
    <li>Beta Series</li>
    <li>Gamma Series</li>
</ul>
  </div>
</div>
</body>
</html>
```

4. Ανάπτυξη εγγενών κινητών εφαρμογών (Android programming)

4.1. Εισαγωγή

Σε αυτή την ενότητα θα ασχοληθούμε με την ανάπτυξη εγγενών εφαρμογών για το λειτουργικό σύστημα Android. Θα γίνει παρουσίαση της μεθοδολογίας ανάπτυξης εφαρμογών τόσο σε περιβάλλον Eclipse (σε συνδυασμό με το απαραίτητο Android plugin), όσο και στο νεότερο περιβάλλον Android Studio της Google.

4.2. Χρήση layouts και buttons

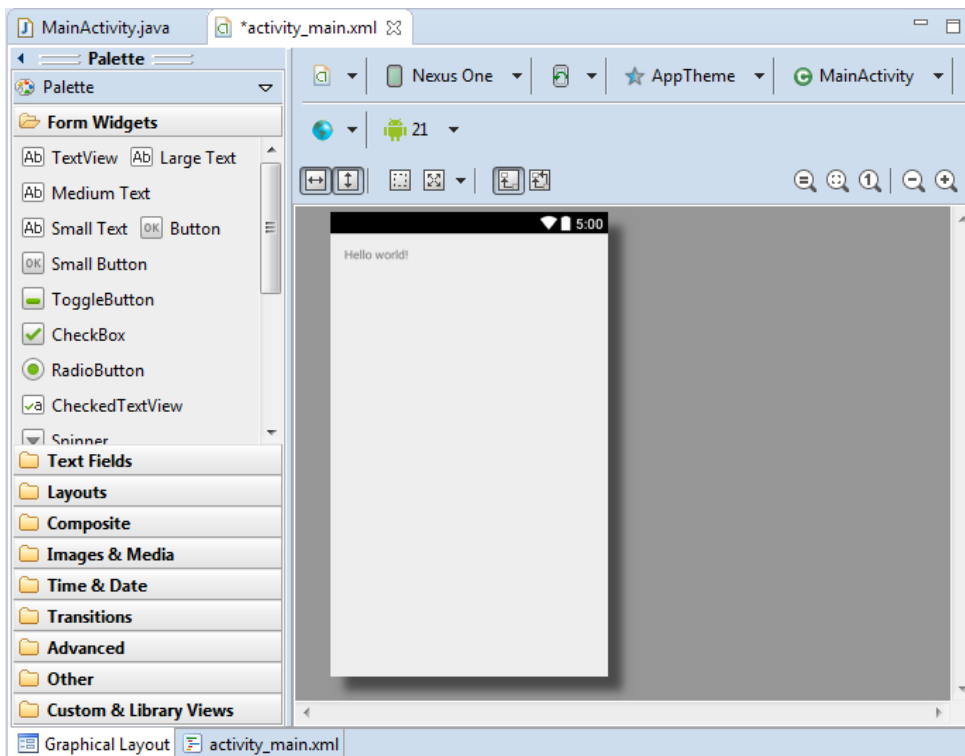
Σε αυτή την εργαστηριακή άσκηση θα ασχοληθούμε με τα layouts, τη χρήση buttons και button events, μέσα από την ανάπτυξη μιας σχετικής εφαρμογής. Το προγραμματιστικό περιβάλλον που θα χρησιμοποιήσουμε θα είναι το Eclipse IDE, έκδοση Luna.

1. Δημιουργούμε ένα νέο project. Επιλέγουμε File->New->Other->Android->Android Application Project.
2. Δίνουμε σαν όνομα “Application With Buttons And Layouts” και πατάμε Next.
3. Πατάμε Next στις επόμενες επιλογές και Finish.

4. Έχοντας ανοιχτό το project «Application With Buttons And Layouts» θα ανατρέξουμε στο αρχείο:

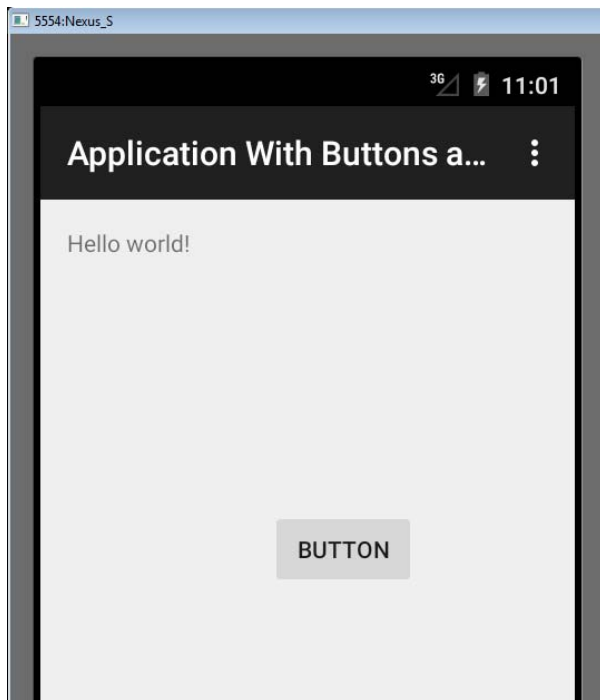
res->layout->activity_main.xml

Και θα το ανοίξουμε με διπλό κλικ. Έτσι θα ανοίξει στην οθόνη μας ο layout manager.



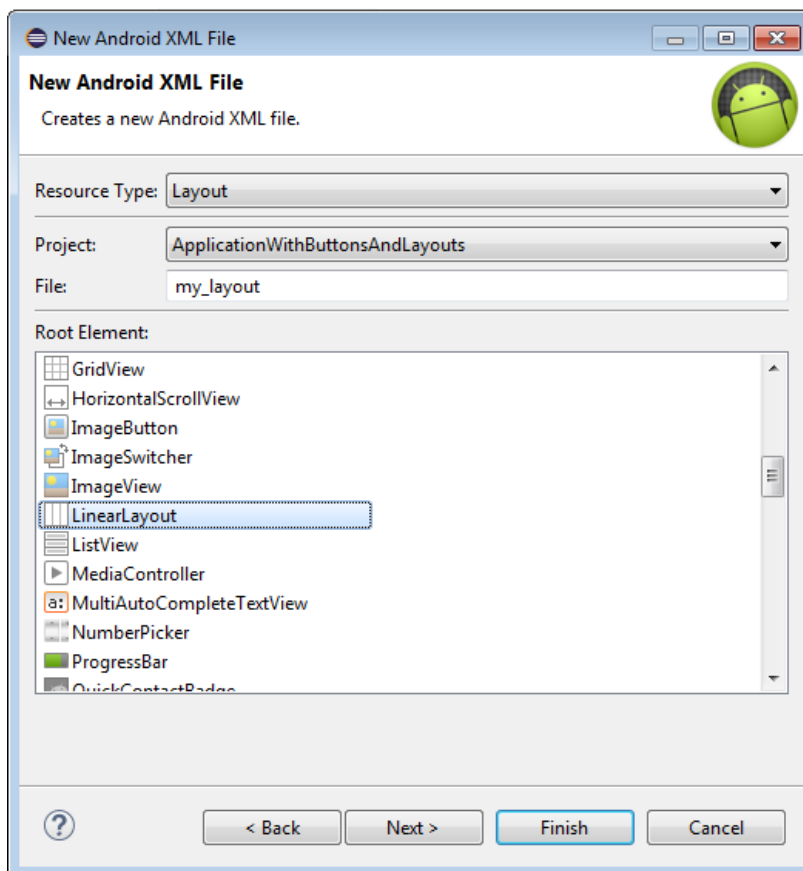
Εικόνα 8.7 Προβολή του Layout Manager στο περιβάλλον Eclipse IDE

5. Εδώ φαίνεται πως έχουμε ως root layout το relative layout, στο οποίο κάθε αντικείμενο έχει σχετική θέση σε σχέση με τα υπόλοιπα.
6. Από το μενού στα αριστερά μπορούμε να «σύρουμε» (με drag and drop) νέα στοιχεία όπως για παράδειγμα πλήκτρα (buttons).
7. Θα δούμε πως προσθέτοντας ένα πλήκτρο αυτό αποκτά σχετική θέση με το στοιχείο text που ήδη υπάρχει. Κάνοντας δεξί κλικ στο πλήκτρο που μόλις προσθέσαμε και επιλέγοντας “Edit Text”, μπορούμε να αλλάξουμε το κείμενο που εμφανίζεται στο εσωτερικό του πλήκτρου.
8. Κάνουμε save και εκτελούμε την εφαρμογή μας, για να δούμε πως αυτή εμφανίζεται στον emulator. Θα δούμε πως υπάρχει το πλήκτρο που μόλις προσθέσαμε, αλλά δεν έχει κάποια λειτουργικότητα ακόμη. Στη συνέχεια της άσκησης θα προσθέσουμε λειτουργικότητα μέσω των Button Events.



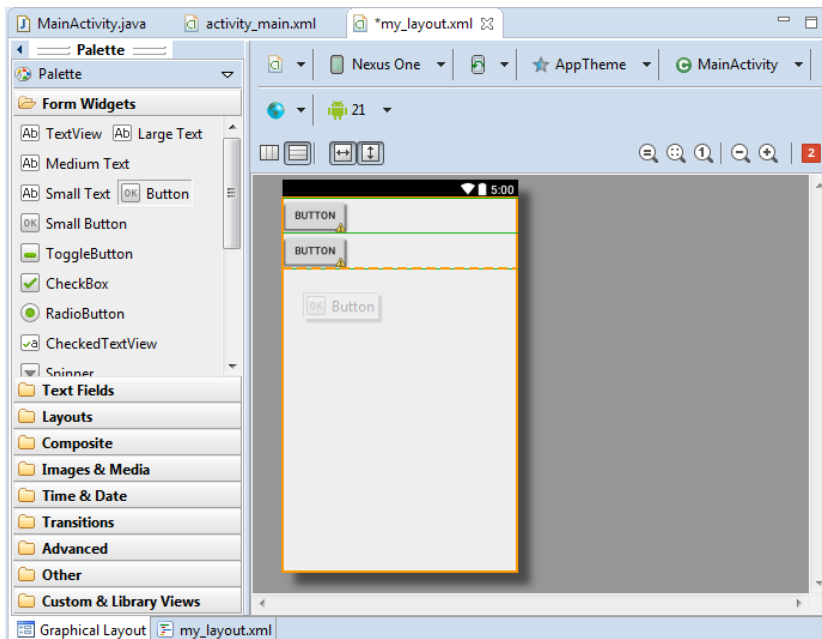
Εικόνα 8.8 Ενδεικτική εμφάνιση της εφαρμογής μας

9. Δημιουργούμε ένα νέο layout πηγαίνοντας στον φάκελο layouts και κάνοντας δεξί κλικ->new-> other->Android->Android XML file. Επιλέγουμε linear layout και δίνουμε ως όνομα mylayout.xml.



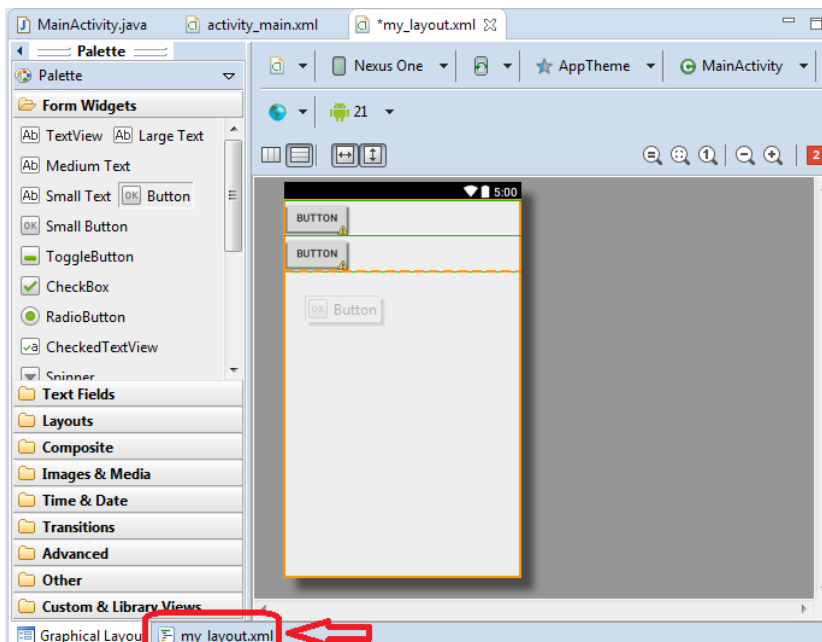
Εικόνα 8.9 Εισαγωγή ενός νέου Layout, τύπου Linear

10. Επιλέγοντας στο νέο layout να προσθέσουμε κάποια στοιχεία (πχ. Buttons) θα δούμε πως μας επιτρέπει να προσθέσουμε Button μόνο το ένα κάτω από το άλλο.



Εικόνα 8.10 Επεξεργασία ενός Linear Layout

11. Πέρα από το γραφικό περιβάλλον που μας προσφέρεται από την καρτέλα “Graphical Layout” μπορούμε να επεξεργαστούμε το xml αρχείο και μέσω κώδικα κάνοντας κλικ στην καρτέλα my_layout.xml.

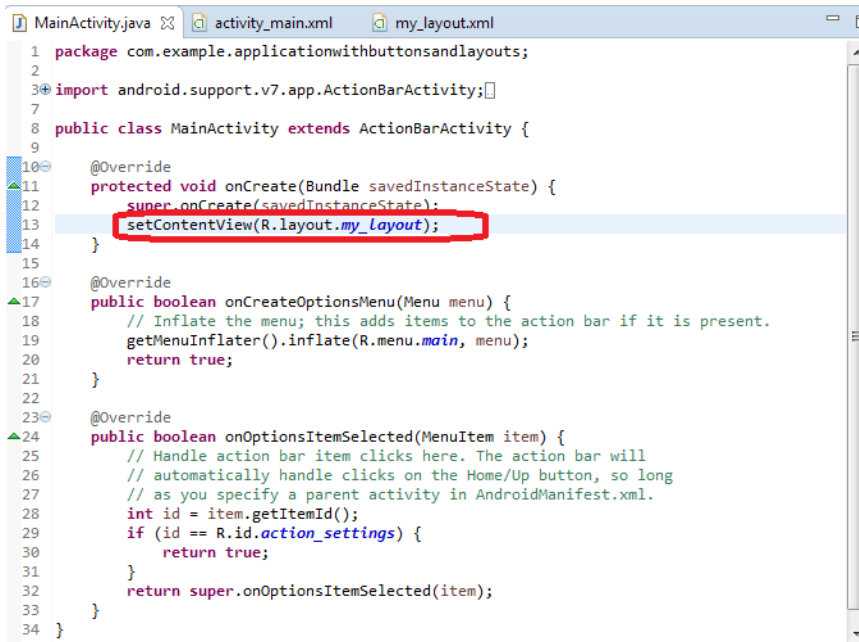


Εικόνα 8.11 Καρτέλα προβολής XML κώδικα

Στην καρτέλα αυτή μπορούμε με χρήση κώδικα να κάνουμε αλλαγές στην εμφάνιση της εφαρμογής. Για παράδειγμα για να αλλάξουμε το χρώμα του background προσθέτουμε τη σειρά:

```
android:background="#00cc00"
```

12. Στη συνέχεια τροποποιούμε το αρχείο MainActivity.java ώστε να δείχνει στο my_layout αρχείο. Τρέχουμε ξανά την εφαρμογή μας για να δούμε τις αλλαγές.



Εικόνα 8.12 Τροποποίηση του αρχείου MainActivity.java

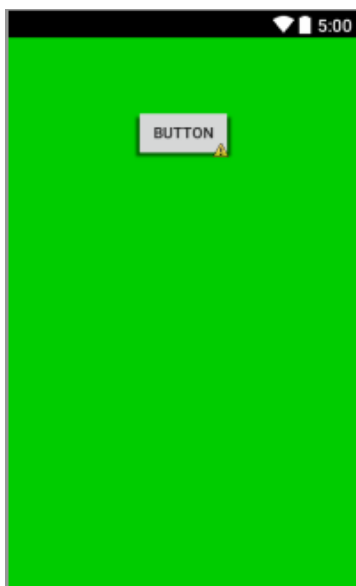
13. Εναλλακτικά μπορούμε να προσθέσουμε κάποια εικόνα. Θα πρέπει να έχουμε τροποποιήσει την εικόνα έτσι ώστε να λειτουργεί σε διάφορες αναλύσεις και σε διάφορα μεγέθη οθονών.

Για να γίνει αυτό τοποθετούμε την τροποποιημένη εικόνα σε καθένα από τους φακέλους drawable που βρίσκονται στον φάκελο res. Για να το κάνουμε αυτό μπορούμε να χρησιμοποιήσουμε το εργαλείο Android Asset Studio.

Link: <http://romannurik.github.io/AndroidAssetStudio/icons-launcher.html>

14. Στη συνέχεια θα προσθέσουμε λειτουργικότητα στο στοιχείο Button. Στο αρχείο activity_main, μέσα στο αντικείμενο Button, εισάγουμε την παρακάτω σειρά κώδικα: android:onClick="onClick"
15. Ενώ στο αρχείο java προσθέτουμε την παρακάτω μέθοδο:

```
public void onClick(View v){
    switch(v.getId()){
        case R.id.button1:
            String message = "Καλημέρα";
            Toast.makeText(this,message,Toast.LENGTH_SHORT).show();
            default:
            break;
        }
    }
```



Εικόνα 8.13 Εμφάνιση εφαρμογής Android

16. Αφήνοντας το ποντίκι πάνω από την πρώτη σειρά κώδικα, μας δίνεται η δυνατότητα να εισάγουμε τις βιβλιοθήκες view που απαιτούνται. Σε περίπτωση που δε γίνει αυτόματα η εισαγωγή τους τις προσθέτουμε χειροκίνητα.

```
import android.view.View;
import android.view.Menu;
import android.widget.Toast;
```

17. Εκτελούμε την εφαρμογή μας και βλέπουμε πως πατώντας το button εμφανίζεται το μήνυμα που ορίσαμε.

Αρχείο xml:

```
<RelativeLayout xmlns: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"
    android:background="#00cc00"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.appicationwithlayoutsandbuttons.MainActivity" >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="48dp"
        android:onClick="onClick"
        android:text="Button"/>

</RelativeLayout>
```

Αρχείο java:

```
package com.example.appicationwithlayoutsandbuttons;
import android.R.string;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Toast;

public class MainActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    public void onClick(View v){
        switch(v.getId()){
            case R.id.button1:
                String message = "Καλημέρα";
                Toast.makeText(this,message,Toast.LENGTH_SHORT).show();
            default:
                break;
        }
    }
}
```

4.3. Άσκηση με αξιοποίηση του Google Maps API

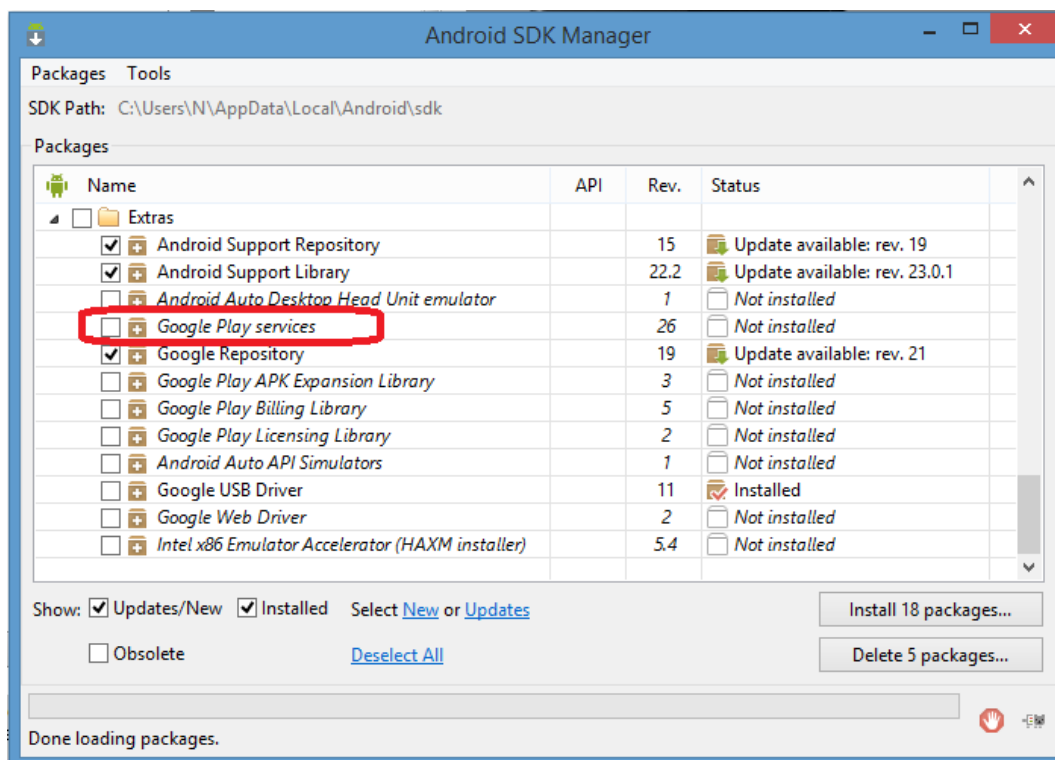
Σε αυτή την εργαστηριακή άσκηση, θα ασχοληθούμε με την ενσωμάτωση ενός χάρτη σε μια εφαρμογή, αξιοποιώντας το Google Maps API. Για την ενσωμάτωση και τη χρήση της υπηρεσίας των χαρτών της Google, σε μια android εφαρμογή, είναι απαραίτητη η λήψη ενός κλειδιού (Google Maps API key). Σε

περίπτωση χρήσης του Eclipse IDE και του Android plugin, είναι απαραίτητο να ακολουθηθεί η διαδικασία που περιγράφεται στο παρακάτω link:

<https://developers.google.com/maps/documentation/android-api/signup>

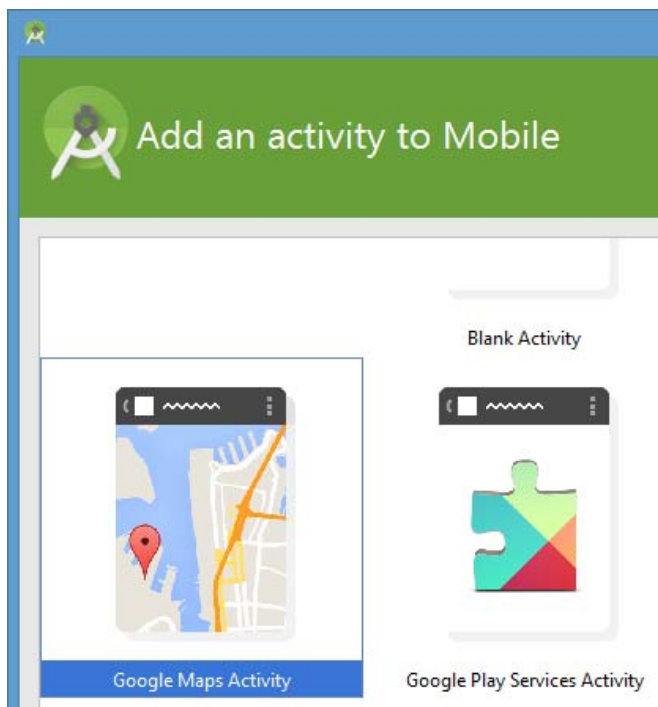
Παρακάτω θα δούμε μια πιο απλοποιημένη μέθοδο, με τη χρήση του προγραμματιστικού περιβάλλοντος Android Studio. Το Android Studio είναι μια πλατφόρμα που έχει αναπτύξει η Google, και σύντομα θα αποτελέσει τη μοναδική πλατφόρμα που θα λαμβάνει τις τελευταίες ενημερώσεις (πχ. ενημερώσεις ασφαλείας).

1. Αφού εκκινήσουμε το Android Studio, εγκαθιστούμε το πακέτο Google Play services μέσω του SDK manager.



Εικόνα 8.14 Επιλογή των Google Play Services, μέσω του Android SDK manager

2. Αφού έχουμε βεβαιωθεί πως έχουμε εγκατεστημένο το πακέτο Google Play services, δημιουργούμε ένα νέο project. Επιλέγουμε File -> New -> New Project.
3. Δίνουμε όνομα στην εφαρμογή μας και πατάμε το πλήκτρο Next.
4. Επιλέγουμε σαν τύπο συσκευών Phone and Tablet και πατάμε το πλήκτρο Next.
5. Επιλέγουμε ως activity ένα Google Maps Activity στο παράθυρο 'Add an activity to Mobile'. Έπειτα πατάμε το πλήκτρο Next.



Εικόνα 8.15 Επιλογή ενός Google Maps Activity κατά την αρχικοποίηση της εφαρμογής

6. Αντιγράφουμε τον σύνδεσμο που έχει δημιουργήσει αυτόματα το Android Studio και που βρίσκεται στο αρχείο google_maps_api.xml.



Εικόνα 8.16 Ο σύνδεσμος που πρέπει να δοθεί στην Google για τη λήψη του κατάλληλου κλειδιού

7. Επικολλούμε τον σύνδεσμο σε έναν φυλλομετρητή (web browser).

Register your application for Google Maps Android API v2 in Google Developers Console

Google Developers Console allows you to manage your application and monitor API usage.

You have no existing projects. A new project named "My Project" will be created.

☐ I agree that my use of any [services and related APIs](#) is subject to my compliance with the applicable [Terms of Service](#).

Agree and continue

Εικόνα 8.17 Διεπιφάνεια του Google Developers Console

8. Ακολουθούμε τις οδηγίες και δημιουργούμε ένα Android API key για το project που εμφανίζεται.

The API is enabled

The project has been created and Google Maps Android API v2 has been enabled.

Next, to use the API you'll need the right credentials.

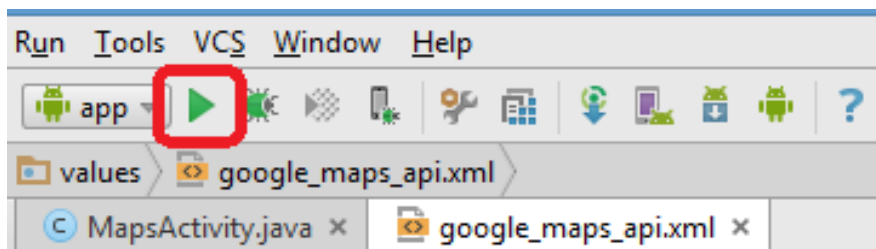
Go to credentials

Εικόνα 8.18 Λήψη διαπιστευτηρίων (κλειδιού)

9. Αντιγράφουμε το κλειδί που δημιουργήθηκε στο αρχείο google_maps_api.xml στο στοιχείο `<string>` (αντικαθιστώντας το κείμενο «YOUR_KEY_HERE»).
10. Ανοίγουμε το αρχείο MapsActivity.java και εντός της μεθόδου MapsActivity προσθέτουμε την παρακάτω μέθοδο (αντικαθιστώντας τις `setUpMapIfNeeded` και `setUpMap`) :

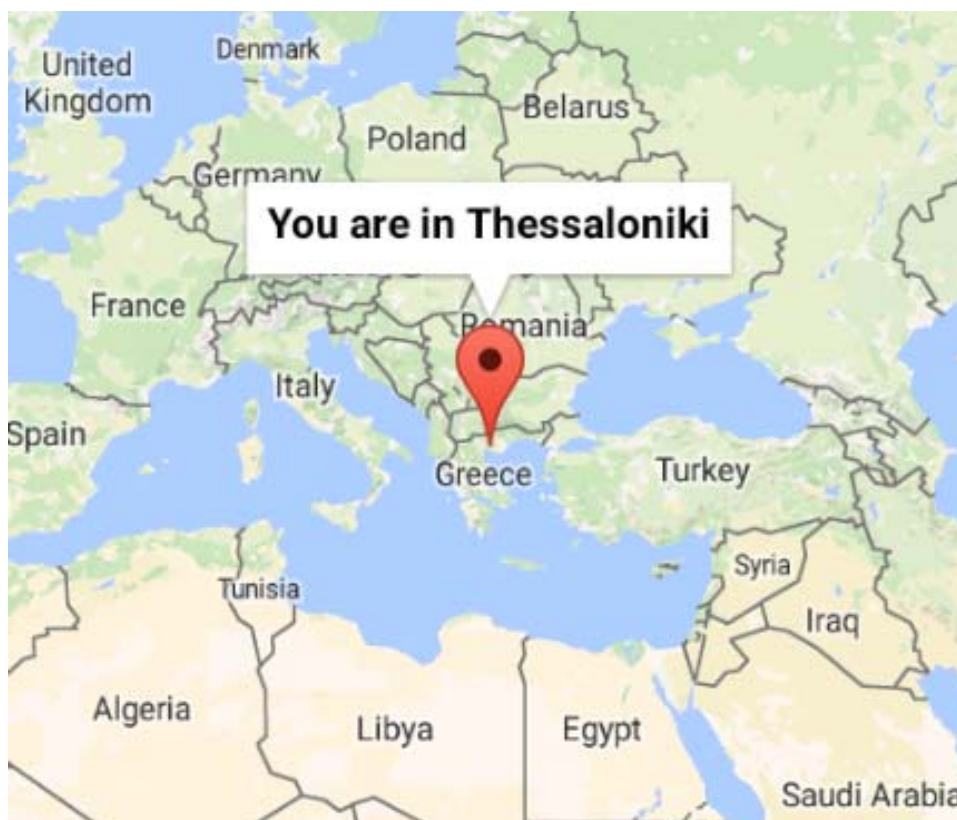
```
@Override
public void onMapReady(GoogleMap map) {
    LatLng thessaloniki = new LatLng(40.625161, 22.960073);
    map.addMarker(new MarkerOptions().position(thessaloniki).title("You are in Thessaloniki"));
    map.moveCamera(CameraUpdateFactory.newLatLng(thessaloniki));
}
```

11. Δημιουργούμε μια εικονική μηχανή που να περιέχει τα Google Play services
12. Πατούμε το πλήκτρο εκτέλεσης της εφαρμογής και επιλέγουμε να την εκτελέσουμε στην εικονική μηχανή που μόλις δημιουργήσαμε



Εικόνα 8.19 Πλήκτρο εκτέλεσης της εφαρμογής

13. Στην εικονική μηχανή που ανοίγει βλέπουμε πως η εφαρμογή μας φορτώνει έναν χάρτη και εστιάζει και τοποθετεί έναν marker στη Θεσσαλονίκη, με το μήνυμα “You are in Thessaloniki”. Κάνοντας zoom in θα δούμε λεπτομέρειες για το σημείο που φαίνεται στον χάρτη.



Εικόνα 8.20 Αποτέλεσμα εκτέλεσης της εφαρμογής



Εικόνα 8.21 Αποτέλεσμα «μεγέθυνσης» (zoom) στον χάρτη

Τελική μορφή αρχείου MapsActivity:

```
package com.example.uom.mymapapplication;
```

```
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
```

```
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

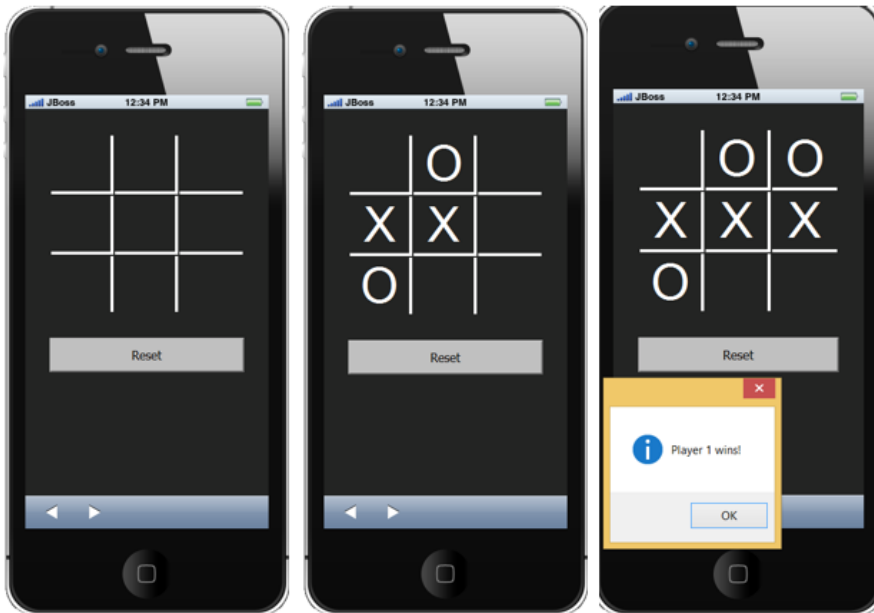
    @Override
    public void onMapReady(GoogleMap map) {
        LatLng thessaloniki = new LatLng(40.625161, 22.960073);
        map.addMarker(new MarkerOptions().position(thessaloniki).title("You are in Thessaloniki"));
        map.moveCamera(CameraUpdateFactory.newLatLng(thessaloniki));
    }
}
```

5. Ασκήσεις αυτοαξιολόγησης

5.1. Ανάπτυξη εφαρμογής για το παιχνίδι τρίλιζα

Εκφώνηση:

Να υλοποιήσετε στο περιβάλλον Phone Gap το γνωστό παιχνίδι τρίλιζα. Για να θεωρείται πλήρης η άσκηση, θα πρέπει όταν ένας παίχτης κερδίζει τον γύρο να εμφανίζεται το κατάλληλο μήνυμα. Ενδεικτικά screenshots για το επιθυμητό τρόπο εμφάνισης της εφαρμογής είναι τα παρακάτω:

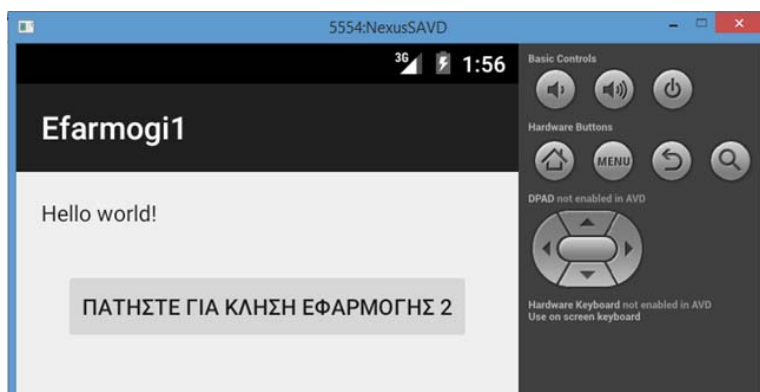


Εικόνα 8.22 Ενδεικτικές οθόνες της εφαρμογής

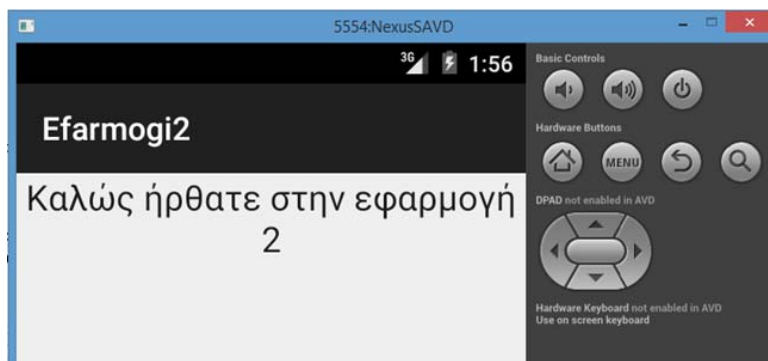
5.2. Άσκηση με αξιοποίηση Intents

Εκφώνηση:

Να αναπτυχθεί εφαρμογή που με χρήση Intents, θα είναι δυνατή η κλήση μιας εξωτερικής εφαρμογής. Συγκεκριμένα θα πρέπει να αναπτυχθούν δύο εφαρμογές. Η πρώτη θα περιλαμβάνει ένα στοιχείο τύπου button. Με την ενεργοποίηση αυτού του στοιχείου από τον χρήστη θα καλείται ένα layout μιας δεύτερης εφαρμογής που θα περιλαμβάνει ένα μήνυμα καλωσορίσματος. Ενδεικτικές εικόνες των δύο εφαρμογών είναι οι παρακάτω:



Εικόνα 8.23 Ενδεικτική οθόνη πρώτης εφαρμογής



Εικόνα 8.24 Ενδεικτική οθόνη δεύτερης εφαρμογής

Σημείωση: Στον Ιστότοπο του συγγραμματος (<http://ec-tech.uom.gr/WT-ECOM>), θα βρείτε τον πηγαίο κώδικα για όλες τις υποδειγματικά λυμένες ασκήσεις, καθώς και για τις ασκήσεις αυτοαξιολόγησης.

6. Συμπεράσματα

Στο κεφάλαιο αυτό αναλύθηκαν και περιγράφηκαν εκτενώς οι κατηγορίες ανάπτυξης περιεχομένου για κινητές συσκευές, ενώ παράλληλα παρουσιάστηκε ένα σύνολο εργαστηριακών ασκήσεων (εκφωνήσεις και οι υποδειγματικές λύσεις αυτών). Σκοπός των εργαστηριακών ασκήσεων ήταν η κατανόηση τεχνικών προγραμματισμού ανάπτυξης περιεχομένου για κινητές συσκευές, κυρίως από την πλευρά του πελάτη (client-side scripting). Οι ασκήσεις που αφορούν την ανάπτυξη υβριδικών εφαρμογών έχουν υποδειγματικά επιλυθεί στο περιβάλλον PhoneGap, ενώ οι ασκήσεις που αφορούν την ανάπτυξη εγγενών εφαρμογών έχουν επιλυθεί στα περιβάλλοντα Eclipse (σε συνδυασμό με το Android Plugin) αλλά και Android Studio, κάνοντας χρήση της γλώσσας προγραμματισμού Java. Μέσω των εργαστηριακών ασκήσεων παρουσιάζονται τεχνικές ανάπτυξης κινητών εφαρμογών ενσωματώνοντας στοιχεία δεδομένα σε εφαρμογές που υποστηρίζουν συναλλαγές Ηλεκτρονικού Εμπορίου (εισαγωγή στοιχείων, φόρμες, διάδραση με χάρτη κ.α.).

Βιβλιογραφία/Αναφορές

- Annuzzi, J. Jr., Darcey, L., Conder, S. (2014). *Introduction to Android Application Development, 4th edition*, Addison-Wesley.
- Cameron, D. (2013). *A Software Engineer Learns HTML5, JavaScript and jQuery*, CreateSpace.
- Firtman, M. (2013). *Programming the Mobile Web, 2nd edition*, O' Reilly.
- Firtman, M. (2012). *jQuery Mobile: Up and Running*, O' Reilly.
- Frederick, G. R., & Lal, R. (2009). *Beginning Smartphone Web Development*, Apress.
- Gasston, P. (2013). *The Modern Web: Multi-Device Web Development with HTML5, CSS3, and JavaScript*, No Starch Press.
- Harwani, B. (2013). *PhoneGap Build: Developing Cross Platform Mobile Applications in the Cloud*, Auerbach Publications.
- Jackson, W. (2013). *Learn Android App Development*, Apress.
- Matthews, A., & Gliser, S. (2015). *Creating Mobile Apps with jQuery Mobile, 2nd edition*, Packt Publishing.
- Γαβαλάς, Δ., Κασαπάκης, Β., & Χατζηδημήτρης, Θ. (2015). *Κινητές Τεχνολογίες*, Εκδόσεις Νέων Τεχνολογιών.

Χρήσιμοι δικτυακοί τόποι:

World Wide Web Consortium, W3C:

W3C Web Design (http://www.w3schools.com/website/web_design.asp)

W3C HTML5 Tutorial (http://www.w3schools.com/html/html5_intro.asp)

W3C CSS3 Tutorial (http://www.w3schools.com/css/css3_intro.asp)

W3C jQuery Mobile Tutorial (<http://www.w3schools.com/jquerymobile/default.asp>)

W3C AJAX Tutorial (<http://www.w3schools.com/ajax/default.asp>)

W3C Google Maps API Tutorial (<http://www.w3schools.com/googleAPI/default.asp>)

Official jQuery Mobile Resources (<http://jquerymobile.com/resources>)

Android tutorials:

<https://developer.android.com/training/basics/firstapp/index.html>

<https://developer.android.com/training/basics/fragments/index.html>

<https://developer.android.com/training/building-graphics.html>

<https://developer.android.com/training/building-location.html>

<https://developer.android.com/training/best-ux.html>

Κριτήρια αξιολόγησης

Σημείωση: Η διαβάθμιση δυσκολίας των κριτηρίων αξιολόγησης δίνεται με το πλήθος των αναγραφόμενων αστερίσκων.

Κριτήριο αξιολόγησης 1

[*] Οι υβριδικές κινητές εφαρμογές, χρησιμοποιούν τις ίδιες τεχνολογίες ανάπτυξης περιεχομένου που χρησιμοποιούν και οι κινητές ιστοσελίδες

A) Σωστό

B) Λάθος

Απάντηση/Λύση

A) Σωστό

Κριτήριο αξιολόγησης 2

[*] Οι εγγενείς κινητές εφαρμογές:

A) Γράφονται συνήθως σε HTML5 κώδικα σε συνδυασμό με τη χρήση CSS3 και Javascript

B) Αναπτύσσονται αποκλειστικά σε Java και έτσι εκτελούνται σε όλα τα γνωστά λειτουργικά συστήματα κινητών συσκευών.

Γ) Αναπτύσσονται σε κατάλληλη γλώσσα προγραμματισμού ανάλογα με το λειτουργικό σύστημα στο οποίο στοχεύουν.

Απάντηση/Λύση

Γ) Αναπτύσσονται σε κατάλληλη γλώσσα προγραμματισμού ανάλογα με το λειτουργικό σύστημα στο οποίο στοχεύουν.

Κριτήριο αξιολόγησης 3

[*] Ο τρόπος διανομής και η συνδεσιμότητα είναι παρόμοια για τις υβριδικές και τις εγγενείς κινητές συσκευές:

A) Σωστό

B) Λάθος

Απάντηση/Λύση

A) Σωστό

Κριτήριο αξιολόγησης 4

[*] Η εντολή «Toast.makeText(this,message,Toast.LENGTH_SHORT).show()», σε μια εφαρμογή android επιτρέπει:

A) Την εμφάνιση ενός toast μηνύματος στην οθόνη

B) Την είσοδο ενός sting από τον χρήστη και την αποθήκευση του στη μεταβλητή message

Γ) Τη μεταφορά ενός toast μηνύματος μέσω intent σε μια άλλη εφαρμογή

Απάντηση/Λύση

Α) Την εμφάνιση ενός toast μηνύματος στην οθόνη

Κριτήριο αξιολόγησης 5

[*] Για την ενσωμάτωση και τη χρήση της υπηρεσίας των χαρτών της Google, σε μια android εφαρμογή, είναι απαραίτητη η λήψη ενός κλειδιού (Google Maps API key).

Α) Σωστό

Β) Λάθος

Απάντηση/Λύση

Α) Σωστό

Κριτήριο αξιολόγησης 6

[**] Το παρακάτω τμήμα κώδικα:

```
<div style="clear: left;"> <input type="reset" value="Λειτουργία" /> </div>
```

Α) Χρησιμοποιείται για τη στοίχιση κειμένου στα αριστερά της οθόνης

Β) Χρησιμοποιείται για την υποστήριξη καθαρίσματος μιας φόρμας από τις επιλογές που έχει δώσει ο χρήστης

Γ) Χρησιμοποιείται για την υποβολή των στοιχείων που έχει δώσει ο χρήστης και στη συνέχεια την επαναφορά σε προηγούμενη οθόνη

Απάντηση/Λύση

Β) Χρησιμοποιείται για την υποστήριξη καθαρίσματος μιας φόρμας από τις επιλογές που έχει δώσει ο χρήστης

Κριτήριο αξιολόγησης 7

[**] Η εντολή «map.addMarker(new MarkerOptions().position(X).title("Message"))» χρησιμοποιείται για

την εμφάνιση της τρέχουσας τοποθεσίας του χρήστη στον χάρτη, εμφανίζοντάς του παράλληλα ένα μήνυμα.

α) Σωστό

β) Λάθος

Απάντηση/Λύση

Β) Λάθος

Κριτήριο αξιολόγησης 8

[*] Η JavaScript είναι μια δυναμική γλώσσα προγραμματισμού η οποία:

Α) χρησιμοποιείται ευρέως από τους Web browsers, όσον αφορά μάλιστα την πλευρά του πελάτη θεωρείται η κύρια γλώσσα του Διαδικτύου

B) είναι ικανή να ελέγξει όλα τα στοιχεία του Document Object Model μια ιστοσελίδας και να τα πυροδοτήσει προκαλώντας διάφορα συμβάντα

Γ) είναι ικανή να υποστηρίζει την έννοια της φορητότητας στις κινητές εφαρμογές για όλες τις πλατφόρμες και συσκευές

Δ) συνδυάζει όλα τα προαναφερόμενα χαρακτηριστικά

Ε) συνδυάζει τα προαναφερόμενα χαρακτηριστικά των επιλογών Α) και Γ)

Απάντηση/Λύση

Δ) συνδυάζει όλα τα προαναφερόμενα χαρακτηριστικά

Κριτήριο αξιολόγησης 9

[*] Η mobile jQuery είναι μια βιβλιοθήκη, ένα πλαίσιο (framework), η οποία:

A) δε βασίζεται στη βιβλιοθήκη jQuery

B) χρησιμοποιεί τις HTML5, CSS3 και JavaScript

Γ) διαχειρίζεται συμβάντα των κινητών συσκευών για τη δημιουργία κινητών εφαρμογών

Δ) συνδυάζει όλα τα προαναφερόμενα χαρακτηριστικά

Ε) συνδυάζει τα προαναφερόμενα χαρακτηριστικά των επιλογών B) και Γ)

Απάντηση/Λύση

Ε) συνδυάζει τα προαναφερόμενα χαρακτηριστικά των επιλογών B) και Γ)

Κριτήριο αξιολόγησης 10

[**] Ποια από τις ακόλουθες επιλογές δεν αποτελεί σημείο προβληματισμού στην ανάπτυξη κινητών ιστοσελίδων:

A) Είναι διαφορετικά τα πλαίσια κινητών χρηστών σε σχέση με τους χρήστες σταθερών υπολογιστών

B) Υπάρχουν στην κυκλοφορία πολλά κινητά, με διαφορετικούς περιηγητές, με διαφορετικές εκδοχές

Γ) Είναι διαφορετικές οι γλώσσες προγραμματισμού σε σχέση με αυτές που απαιτούνται για την ανάπτυξη ιστοσελίδων για σταθερούς υπολογιστές (δηλ. ιστοσελίδων του παραδοσιακού ενσύρματου Ιστού)

Δ) Τα ασύρματα/κινητά δίκτυα είναι πιο αργά, ενώ και το υλικό είναι πιο αργό και με λιγότερη διαθέσιμη μνήμη σε σχέση με τους αντίστοιχους διαθέσιμους πόρους των σταθερών υπολογιστών για πρόσβαση σε ιστοσελίδες

Απάντηση/Λύση

Γ) Είναι διαφορετικές οι γλώσσες προγραμματισμού σε σχέση με αυτές που απαιτούνται για την ανάπτυξη ιστοσελίδων για σταθερούς υπολογιστές (δηλ. ιστοσελίδων του παραδοσιακού ενσύρματου Ιστού).