

Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»



**Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	<b>Ανάπτυξη εκπαιδευτικής εφαρμογής για περιβάλλον Windows Phone</b>
Όνοματεπώνυμο Φοιτητή	<b>Σταθουλόπουλος Βασίλης</b>
Πατρώνυμο	<b>Εμμανουήλ</b>
Αριθμός Μητρώου	<b>ΜΠΠΛ/09021</b>
Επιβλέπων	<b>Βίρβου Μάρια , Καθηγητής</b>

Ημερομηνία Παράδοσης :

**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

(υπογραφή)

(υπογραφή)

Όνομα Επώνυμο  
Βαθμίδα

Όνομα Επώνυμο  
Βαθμίδα

Όνομα Επώνυμο  
Βαθμίδα

Μαρία Βίρβου

Γεώργιος Τσιχριντζής

Φούντας Ευάγγελος

Καθηγητής

Καθηγητής

Καθηγητής

## Περίληψη

Στην παρούσα μεταπτυχιακή διατριβή θα επικεντρωθούμε στις νέες τεχνολογίες στις αλλαγές που έχουν φέρει στον τρόπο με τον οποίο γίνεται η εκπαίδευση στις μέρες καθώς και τα σημαντικά πλεονεκτήματα που προσφέρουν σε σχέση με την παραδοσιακή εκπαίδευση και όποια μειονεκτήματα μπορεί να έχουν. Επίσης θα δούμε τα βασικά στάδια ανάπτυξης μια εκπαιδευτικής εφαρμογής για smartphones και όλη την διαδικασία που απαιτείται για την διάθεση της σε κάποιο online store .

## Abstract

In this thesis we will focus on new technologies to bring changes in the way education is today and the important advantages compared to traditional education and any disadvantages they may have. We will also look at key stages of development an educational application for smartphones and all the procedures required for its distribution across an online store.

# Περιεχόμενα

<b>1. ΕΙΣΑΓΩΓΗ</b>	<b>5</b>
<b>2. ΑΝΑΣΚΟΠΗΣΗ ΠΕΔΙΟΥ</b>	<b>7</b>
2.1 Ορισμός των τεχνολογιών στην εκπαίδευση	7
2.2 Διδασκαλία με νέες τεχνολογίες	9
2.3 Πλεονεκτήματα - Μειονεκτήματα	9
2.4 Εκπαίδευση και κινητή τηλεφωνία	11
2.5 Υφιστάμενα προγράμματα για smartphones	12
<b>3. ΠΑΡΟΥΣΙΑΣΗ ΣΥΣΤΗΜΑΤΟΣ</b>	<b>15</b>
3.1 Ενότητα "Μαθήματα"	15
3.2 Ενότητα "Παιχνίδι"	18
3.3 Ενότητα "Οδηγίες" και "BlueShyMonkey Apps"	22
<b>4. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ</b>	<b>25</b>
4.1 Προγραμματιστικές Πλατφόρμες	25
4.1.1 Γλώσσα επισήμανσης XAML	27
4.1.2 Isolated Storage	29
4.1.3 Γλώσσα σχεδίασης METRO	30
4.1.4 Hardware	40
4.2 Διαγράμματα UML	44
4.2.1 Διαγράμματα περιπτώσεων χρήσης	44
4.2.2 Διαγράμματα τάξεων	45

4.2.3	Ανάλυση - Σχεδιασμός	46
4.2.4	Φάση εκπόνησης μελέτης	48
4.3	Ανάπτυξη Εφαρμογής	56
4.3.1	Γενικά	56
4.3.2	Δοκιμή της εφαρμογής στον Emulator	62
4.3.3	Δημιουργώντας την "Κύρια Σελίδα"	66
4.3.4	Δημιουργώντας την "Σελίδα Οδηγιών"	72
4.3.5	Δημιουργώντας την "Πρώτη σελίδα αριθμών"	74
4.3.6	Δημιουργώντας τα εφέ κίνησης	83
4.3.7	Δημιουργώντας τις απαραίτητες κλάσεις	87
4.3.8	Δημιουργώντας την σελίδα για το "πλακάκι"	98
4.3.9	Δημιουργώντας την σελίδα του "Παιχνιδιού"	100
4.3.10	Πλοήγηση	110
4.4	Διαφημίσεις	112
4.4.1	Γενικά	112
4.4.2	Δημιουργία διαφημιστικής μονάδας	113
4.5	Marketplace	119
4.5.1	Εγγραφή στο Marketplace	119
4.5.2	Ανέβασμα Εφαρμογής	124
4.5.3	Έλεγχος εφαρμογών (Dashboard)	126
<b>5.</b>	<b>ΣΥΜΠΕΡΑΣΜΑΤΑ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ</b>	<b>132</b>
-	Βιβλιογραφία	134

# 1. ΕΙΣΑΓΩΓΗ

Στην παρούσα μεταπτυχιακή διατριβή διαπραγματευόμαστε την κατασκευή ενός εκπαιδευτικού λογισμικού για φορητές συσκευές και συγκεκριμένα για την νέα πλατφόρμα της Microsoft στο χώρο των κινητών τηλεφώνων τα **Windows Phone**.

Με το συγκεκριμένο πρόγραμμα θέλουμε να δείξουμε την συνεισφορά των νέων τεχνολογιών στην εκπαίδευση ακόμα και σε νηπιακό επίπεδο , καθώς η εφαρμογή μας απευθύνεται σε μικρά παιδιά. Το πρόγραμμα εστιάζει στην ευκολία χρήσης , στην παιδική διάθεση καθώς και στο παιχνίδι αφού απευθύνεται σε παιδιά μικρής ηλικίας.

Το πρόγραμμα το οποίο θα παρουσιάσουμε είναι το "**Αριθμάρια Παζλάκια**" είναι ένα διττό πρόγραμμα αφού προσπαθεί να παντρέψει την εκπαίδευση και το παιχνίδι μέσα από την σκοπιά της φορητής τεχνολογίας. Το πρόγραμμα μας χωρίζεται ουσιαστικά σε δύο μέρη το ένα είναι το μέρος των μαθημάτων όπου δίνει την δυνατότητα στο χρήστη να μάθει του αριθμούς από το ένα μέχρι και το δέκα εκμεταλλευόμενος όλων των νέων δυνατοτήτων που παρέχουν τα σύγχρονα κινητά. Το δεύτερο μέρος είναι ένα παιχνίδι τύπου jigsaw puzzle το οποίο προσπαθεί μέσω της διασκέδασης να δώσει στο χρήστη την δυνατότητα να εξασκήσει τις γνώσεις του.

Σκοπός του προγράμματος μας είναι να μεταφέρει την εκπαίδευση σε ένα μέσο το οποίο είναι ευρέως γνωστό τα τελευταία χρόνια και πρόκειται να γνωρίσει περαιτέρω ανθίσει τα επόμενα, τις οθόνες των κινητών τηλεφώνων. Αυτό που προσπαθήσαμε να επιτύχουμε είναι να δώσουμε την δυνατότητα να μπορεί κάποιος να εκπαιδευτεί δηλαδή να είναι ο μαθητής αλλά και να εκπαιδεύσει να έχει δηλαδή το ρόλο του δασκάλου. Ως προγράμματα το τελευταίο διάστημα έχουν αρχίσει να παρουσιάζονται αρκετά στο χώρο της τεχνολογίας και ειδικά των υπολογιστών, αλλά προσπαθήσαμε να προχωρήσουμε ένα βήμα παραπέρα και να αναπτύξουμε ένα απλό πρόγραμμα διδασκαλίας των αριθμών από το ένα μέχρι το δέκα για smart phones.

Απώτερος σκοπός είναι ουσιαστικά να ενσωματώσουμε την εκπαίδευση σε ένα μέσο το οποίο είναι κοινώς αποδεκτό και να δώσουμε την δυνατότητα σε αυτήν από οποιονδήποτε χρήστη και από οποιαδήποτε μέρος στο οποίο μπορεί να βρίσκεται .

Το πρόγραμμα μας όπως αναφέραμε δίνει την δυνατότητα για εκμάθηση των αριθμών από το ένα μέχρι το δέκα . Ουσιαστικά απευθύνεται κυρίως σε παιδιά προσχολικής και σχολικής ηλικίας. Ο λόγος που διαλέξαμε το συγκεκριμένο είναι να δείξουμε ότι η τεχνολογία δεν απευθύνεται μόνο σε γνώστες αλλά μπορεί να χρησιμοποιηθεί πάντα φυσικά με την επίβλεψη κάποιου γονέα μέχρι και από παιδιά και να τα βοηθήσει μέσω της άμεσης αλληλεπίδρασης που παρέχει όπως με την χρήση οθόνων αφής , ήχου , κινούμενης εικόνας κτλ να μάθουν καινούρια πράγματα . Η εφαρμογή μας χρησιμοποιεί όλες τις παραπάνω δυνατότητες των σύγχρονων κινητών καθώς κάνει χρήση της οθόνης αφής για άμεση και πιο φυσική αλληλεπίδραση του χρήστη με τις ενέργειες που θέλει να διεκπεραιώσει. Χρησιμοποιεί διάφορα ηχητικά εφέ τόσο για να δώσουμε μια χαρούμενη νότα καθώς η εφαρμογή απευθύνεται για παιδιά αλλά και ως μέσω εξόδου για να δώσει πληροφορίες στο χρήστη και στην συγκεκριμένη περίπτωση την δυνατότητα να ακούσει ως λέξη ο χρήστης τον κάθε αριθμό. Επιπλέον κάνει χρήση της εικόνας για να απεικονίσει στο χρήστη τον κάθε αριθμό τόσο ως λέξη όσο και ως νούμερο αλλά και κινούμενη εφέ για περισσότερο αλληλεπίδραση .

Στο δεύτερο μέρος της εφαρμογής είναι ένα εκπαιδευτικό παιχνίδι τύπου πάζλ στο οποίο ο χρήστης πρέπει να βάλει στην σειρά τα εννέα κομμάτια του αριθμού δέκα που είναι ανακατεμένα για να σχηματιστεί πάλι ο αριθμός και εδώ έχουμε πλήρη αλληλεπίδραση χρήστη κινητού καθώς ο χρήστης κατευθύνει τα κομμάτια μέσω του δαχτύλου ακουμπώντας απλά την οθόνη .

Ως σύστημα ανάπτυξης της εφαρμογής μας χρησιμοποιήσαμε την νέα πλατφόρμα της Microsoft τα Window Phone το οποίο είναι το καινούριο της λειτουργικό σύστημα για smart phones. Στην συνέχεια θα αναφερθούμε πιο αναλυτικά στο κεφάλαιο 2ο στις αλληλεπίδραση των νέων τεχνολογιών και της εκπαίδευσης . Στο κεφάλαιο 3ο θα κάνουμε μια πιο εκτενή και με αρκετά screen shots παρουσίαση του συστήματος μας . Στο κεφάλαιο 4ο θα αναφερθούμε στην πλατφόρμα βάσει της οποίας αναπτύξαμε την εφαρμογή, τις σχεδιαστικές αρχές του λειτουργικού της Microsoft καθώς και αναλυτικά παραδείγματα του κώδικα που "γράψαμε" για την εφαρμογή. Τέλος στο τελευταίο κεφάλαιο το 5ο θα κάνουμε μια συνολική αξιολόγηση του προγράμματος μας βελτιώσεις και προσθήκες που μπορούν να γίνουν μελλοντικά καθώς και σε διάφορα συμπεράσματα στα οποία έχουμε καταλήξει.

# 2. ΑΝΑΣΚΟΠΗΣΗ ΠΕΔΙΟΥ

## 2.1 Ορισμός των νέων τεχνολογιών στην εκπαίδευση

Ζούμε σε μια εποχή που τα πάντα γύρω μας αλλάζουν. Αντιλαμβανόμαστε ότι αυτό έχει να κάνει με τον αυξανόμενο ρυθμό της τεχνολογίας, τις νέες εφευρέσεις, τις νέες επινοήσεις. Συνέπεια όλων αυτών είναι η αλλαγή στον τρόπο με τον οποίο ο άνθρωπος σκέφτεται και πράττει. Έτσι ενεργεί με την μεγαλύτερη αυτονομία, πιο γρήγορα και με νέα θεμέλια αποδοτικότητας. Όλοι οι τομείς της ζωής είναι επηρεασμένοι απ' αυτό το πνεύμα της αλλαγής.

Όλο και περισσότερο παρατηρούμε την μεγάλη επιρροή που ασκούν οι νέες τεχνολογίες στον τομέα της εκπαίδευσης. Οι υπολογιστές, το δίκτυο World Wide Web, και η διαχυτικότητα που προσφέρει το Internet, έχει παρακινήσει πολλούς ερευνητές να εξετάσουν τις δυνατότητες και τις προκλήσεις που τυχόν να έχουν στον τομέα της παιδείας. Ο εκπαιδευτικός πρέπει να είναι ενήμερος γι' αυτά τα παιδαγωγικά και εκπαιδευτικά πλεονεκτήματα που προσφέρουν οι νέες τεχνολογίες πριν αποφασίσει να τις χρησιμοποιήσει, και να μην καταλήξει σ' αυτές μόνο εξαιτίας της άμετρης αγάπης για την τεχνολογία ή απλώς επειδή σήμερα η τεχνολογία έχει μπει στη ζωή μας.

Πραγματικά όμως στον τομέα αυτό, η τεχνολογία έχει πολλά να προσφέρει προς όφελος και των μαθητών και των εκπαιδευτικών και της κοινωνίας. Οι μέσοι μαθητές θα αποκομίσουν πολλά οφέλη, ενώ οι ιδιοφυίες δεν θα περιορίζονται μόνο στην συγκεκριμένη διδακτέα ύλη, ούτε θα χρειάζεται να ακολουθούν αργούς ρυθμούς προς χάριν μερικών άλλων μαθητών. Στα παιδιά με ειδικές ανάγκες θα ανοιχτούν νέοι οπτικοί οδοί, ενώ ο αναλφαβητισμός σε ορισμένες περιοχές θα πάψει να υπάρχει.

Σύμφωνα με τον Niccolo Machiavelli "Δεν υπάρχει τίποτα πιο δύσκολο από το να σχεδιάσεις, τίποτα πιο αβέβαιο από την επιτυχία, τίποτα πιο αδύνατο στην υλοποίηση από την δημιουργία ενός νέου συστήματος. Γιατί ο δημιουργός θα συναντήσει μεγάλη εχθρότητα απ' αυτούς που ωφελούνται από τη διατήρηση του παλιού συστήματος, ενώ από την άλλη θα συναντήσει



απλώς αδιάφορους υποστηρικτές, εάν ορισμένοι πρόκειται να αποκομίσουν κάποια οφέλη από το καινούριο."

Τους υπολογιστές ως απλή τεχνολογία τους συναντάμε σε όλη τη διάρκεια της σχολικής ζωής. Με την εισβολή των υπολογιστών στις τάξεις, οι μαθητές αποκτούν πρόσβαση σε νέες πηγές γνώσης, έρχονται σε συνεργασία με άλλους μαθητές, και ουσιαστικά έχουν την δυνατότητα να αποκτήσουν ότι θέλουν, από όπου θέλουν και όποτε το θέλουν, και με μορφή πιο προσιτή σε αυτούς, πάντοτε όμως με την βοήθεια και καθοδήγηση του δασκάλου.

Το επόμενο στάδιο εξέλιξης της εκπαίδευσης είναι η Mobile εκπαίδευση, δηλαδή η χρήση συσκευών όπως tablet και κινητών τηλεφώνων για την διδασκαλία των μαθητών.

Ο υπολογιστής δεν πρόκειται να καταργήσει ούτε το βιβλίο, ούτε τον δάσκαλο. Αντίθετα, οι τελευταίοι είναι απαραίτητοι για την επιτυχία του νέου είδους διδασκαλίας. Η χρήση των νέων τεχνολογιών στην εκπαίδευση μεταφέρει σε άλλη διάσταση την μορφή διδασκαλίας καθώς πλέον είναι πολύ πιο εύκολη η αναπαράσταση διάφορων παραδειγμάτων. Ερευνητές στο τομέα προσφέρουν συνέχεια νέα επιτεύγματα που διευκολύνουν πολύ τους μαθητές. Για παράδειγμα η νέα εξέλιξη στο τομέα του video είναι το λεγόμενο **interactive video** που αποτελείται από ένα υπολογιστή, ένα video με Compact Disc και μια τηλεόραση. Το σύστημα ελέγχεται από τον υπολογιστή και ο χρήστης δεν παρακολουθεί παθητικά το μάθημα, αλλά έχει την δυνατότητα να επικοινωνήσει μέσω του υπολογιστή, να ρωτήσει ή να ζητήσει πληροφορίες για ένα συγκεκριμένο θέμα.

Το **Videotex** μας προσφέρει μια απεριόριστη ποσότητα πληροφοριών. Ο υπολογιστής με την βοήθεια ενός modem και μέσα από το τηλεπικοινωνιακό δίκτυο, μπορεί να επικοινωνήσει οποιαδήποτε στιγμή με οποιαδήποτε τράπεζα πληροφοριών, αρκεί να σχηματίσει το τηλέφωνο της. Το πιο γνωστό είναι το Γαλλικό Teletel ( με περισσότερες από 10000 τράπεζες πληροφοριών) και το Αγγλικό Prestel.

Επίσης το **Video-Conferencing** δίνει την δυνατότητα σε άτομα απομακρυσμένα να επικοινωνούν, μεταξύ τους, μέσω ειδικών τηλεφωνικών γραμμών ή άλλων καναλιών. Παρέχει ένα πρόγραμμα "βλέπω και ακούω" γιατί χρησιμοποιεί δύο τρόπους μετάδοσης ηχητικών και οπτικών σημάτων.

Η εξέλιξη αυτή δεν έχει σταματήσει. Οι επιστήμονες υπόσχονται πολλά για το μέλλον, ενώ η τεχνική νοημοσύνη, τα έμπειρα δηλαδή συστήματα που αναπτύσσουν διάλογο με το χρήστη είναι πλέον γεγονός.[7]

## 2.2 Διδασκαλία με νέες τεχνολογίες

Μια άλλη καινοτομία στο ρόλο της εκπαίδευσης είναι εκείνη του ρόλου του δασκάλου. Πριν από λίγο διάστημα κυριαρχούσε η δάσκαλο-κεντρική αντίληψη για το σχολείο. Ο δάσκαλος ήταν αυτός που "ήξερε" και οι μαθητές αυτοί που έπρεπε να "μάθουν". Αυτή η νοοτροπία έχει αρχίσει να υποχωρεί και αυτό κατορθώθηκε με την είσοδο των νέων τεχνολογιών στα σχολεία, οι οποίες κατέργησαν το μονοπώλιο του δασκάλου και διεκδίκησαν να πάρουν τη θέση του βιβλίου, την μοναδική έως τότε πηγή γνώσης.

Βάσει πρέπει να δοθεί όχι απλά στην χρήση, αλλά στην σωστή χρήση των νέων τεχνολογιών για να έχουμε πραγματική διδακτική αξία. Το να υπάρχει για παράδειγμα ένας υπολογιστής στην αίθουσα που θα τον χειρίζεται ο καθηγητής δείχνοντας κάποια πράγματα στην τάξη, και οι μαθητές να έχουν ένα βιβλίο που θα πρέπει να αποστηθίσουν προκειμένου να "μάθουν" τον υπολογιστή, δεν είναι η λύση. Τα παιδιά θα πρέπει να περάσουν πολλές ώρες μπροστά στο πληκτρολόγιο χωρίς την παρέμβαση κανενός. Οι καθηγητές θα πρέπει να αλλάξουν νοοτροπία και να σταματήσουν να ελέγχουν την ροή των πληροφοριών όπως κάνουν τώρα, γιατί έτσι και θα μπλοκάρουν την δύναμη των νέων τεχνολογιών και οι ίδιοι θα μείνουν εγκλωβισμένοι στο σημερινό ανεπαρκές σύστημα. Σκοπός τους πρέπει να είναι η εξασφάλιση των συνθηκών εκείνων, ώστε η εκπαίδευση να διαμορφώνει και να αναπτύσσει ένα ολοκληρωμένο άτομο και όχι μόνο την διανοητική του πλευρά. Άρα ο νέος ρόλος του εκπαιδευτικού είναι ρόλος συντονιστή και συμβούλου και όχι εκείνος του μεταδότη της γνώσης. Δεν προσφέρει έτοιμη λύση, αλλά οργανώνει έτσι το μάθημα ώστε να την ανακαλύπτουν οι ίδιοι οι μαθητές. Ο δάσκαλος απλώς παρακολουθεί, ελέγχει και συζητεί το πρόβλημα. Αυτός ο ρόλος του εκπαιδευτικού αντιστοιχεί παιδαγωγικά στην ενεργητική μάθηση μέσα από την ανακάλυψη. [7]

## 2.3 Πλεονεκτήματα - Μειονεκτήματα

Όπως με συντομία αναφέραμε, τα οφέλη της νέας τεχνολογίας είναι πολλά. Αρχικά καταφέρνει να τραβήξει την προσοχή των παιδιών, κάνει την εκπαίδευση πιο ενδιαφέρουσα, ενώ μετατρέποντας την σε παιχνίδι, αυξάνει την αποδοτικότητα. Αναντίρρητα είναι ανεξίτηλη πηγή γνώσης εύκολη στη χρήση, διευκολύνει πολύ την ζωή όλων μας, ενώ ταυτόχρονα ανοίγει

νέου ορίζοντες. Το κυριότερο πλεονέκτημα είναι ότι απευθύνεται σε όλους και ότι ο καθένας μπορεί να βρει οτιδήποτε τον ενδιαφέρει αμέσως. Καλύπτει κενά, καθοδηγεί σωστά, εκμηδενίζει τις αποστάσεις και φέρνει σε επαφή ανθρώπους που βρίσκονται χιλιόμετρα μακριά.

Στην ασύγχρονη επικοινωνία ( e-mail, World Wide Web κτλ ) το πλεονέκτημα είναι ότι δεν υπάρχουν χρονικοί περιορισμοί. Οι άνθρωποι μπορούν να συζητούν άνετα, χρησιμοποιώντας όσο χρόνο θέλουν για να σκεφτούν και να απαντήσουν . Ένα αδύνατο όμως σημείο είναι ότι τέτοιου είδους επικοινωνία, τουλάχιστον στα αρχικά στάδια είναι απρόσωπη. Επιπλέον οι άνθρωποι δεν έχουν συνηθίσει αυτόν τον τρόπο επικοινωνίας γι' αυτό είναι λίγο συγκρατημένοι. Εφόσον όμως γνωριστούν μεταξύ τους, έρθουν σε επαφή μέσω της τηλεσυνδιάλεξης, τότε η συζήτηση εξελίσσεται ομαλά.

Επίσης υπάρχει μια διαφωνία για το κατά πόσο η συνεργασία μέσω των νέων τεχνολογιών μπορεί να προαχθεί κανονικά. Στην πραγματικότητα είναι δύσκολο να πεισθούν οι άνθρωποι να συνεργαστούν με ξένους κυρίως από έλλειψη εμπιστοσύνης. Όταν κάποιος δεν γνωρίζει τον τρόπο με τον οποίο ο συνεργάτης του δουλεύει, τις ικανότητες του, τις γνώσεις του, τότε αυτή η συνεργασία είναι καταδικασμένη να αποτύχει. Υπάρχει βέβαια ο αντίλογος εκείνων που υποστηρίζουν ότι άτομα που είναι ντροπαλά και διστάζουν να πάρουν μέρος σε συζητήσεις, νιώθουν πιο ελεύθεροι όταν καλύπτονται πίσω από την νέα τεχνολογία.

Στην σύγχρονη επικοινωνία πάλι ( τηλέφωνο, video-audio conference κ.α. ), έχουμε την αίσθηση της παγκόσμιας παρουσίας. Η άμεση παγκόσμια παρουσία ανθρώπων από διαφορετικές πόλεις είναι πολύ συναρπαστική εμπειρία. Σοβαρό όμως μειονέκτημα εδώ έχουμε τον λιγοστό χρόνο που έχουμε στην διάθεση μας, και στον οποίο δεν μπορεί να γίνει πραγματικός διάλογος.

Επίσης έχει σημειωθεί ότι δουλεύοντας τα άτομα τει α τει για μια εργασία που χρειάζεται για παράδειγμα μία ώρα, εάν δουλέψουν με τηλε-συνδιάσκεψη, στο ίδιο χρονικό διάστημα δεν θα έχουν φτάσει στο ίδιο επίπεδο. Επιπρόσθετα δεν θα πρέπει να ξεχνάμε και το μεγάλο κόστος της όλης σύνδεσης που παίζει καταλυτικό ρόλο.

Πολλοί βέβαια έχουν κατηγορήσει τις νέες τεχνολογίες ότι δημιουργούν αποξένωση και έλλειψη οποιασδήποτε επικοινωνίας με άλλα πρόσωπα. Αν κάτι τέτοιο αλήθευε τότε το σχολείο ως θεσμός θα είχε αποτύχει γιατί δεν θα είχε καταφέρει να καλλιεργήσει στους μαθητές την συνεργασία, την αλληλεγγύη και την ευγενή άμιλλα. Σκοπός του σχολείου είναι να βοηθάει τα παιδιά να πάρουν τα απαραίτητα εφόδια και να τα εντάξει ομαλά στους κόλπους της κοινωνίας.

Τα παιδιά έχουν ανάγκη να μάθουν να διαμορφώνουν και να αναπτύσσουν διαπροσωπικές σχέσεις, όση ανάγκη έχουν να μορφωθούν, ίσως και περισσότερη. Με τις νέες τεχνολογίες

όμως η επικοινωνία έχει αλλάξει σελίδα. Μέσω του Διαδικτύου μπορούν οι μαθητές-σπουδαστές να μην περιορίζουν τις συναναστροφές τους σε άτομα της συνοικίας ή της πόλης τους, αλλά μπορούν να συνομιλήσουν με άτομα που βρίσκονται χιλιάδες χιλιόμετρα μακριά, σε οποιοδήποτε άλλο κράτος. Το μέγεθος του κόσμου γίνεται αισθητά μικρότερο και οι άνθρωποι διαφόρων πολιτισμών έρχονται σε επαφή μαθαίνοντας καθένας για τον άλλον και από τον άλλον.

Τέλος πλεονεκτήματα υπάρχουν και για τους εκπαιδευτικούς που φεύγουν από την απομόνωση, πλησιάζουν τους μαθητές και γίνονται πιο ενεργητικοί και η βοήθεια τους πιο ουσιαστική, αφού αφήνουν τις διαλέξεις και περνάνε στην πράξη, καθοδηγώντας και βοηθώντας τους μαθητές χωρίς να τους δίνουν έτοιμες λύσεις. Πολλοί καθηγητές ίσως νιώσουν το νέο αυτό ρόλο να τους ταιριάζει περισσότερο, αφού τους δίνει μεγαλύτερη απόλαυση, άρα έτσι μεγαλύτερη επιθυμία για δουλειά. [7]

## 2.4 Εκπαίδευση και κινητή τηλεφωνία

Μετά την σύντομη ανάλυση μας για την χρησιμοποίηση των νέων τεχνολογιών στην εκπαίδευση θα επικεντρωθούμε σε μια "νέα" τεχνολογία που αρχίζει να συμμετέχει στην εκπαίδευση αυτή των κινητών τηλεφώνων. Ως τεχνολογία δεν είναι κάτι το καινούριο καθώς η χρήση της τόσο στην επικοινωνία όσο και στην ψυχαγωγία είναι ήδη ευρέως γνωστή. Ένα καινούριο ρόλο που έχει ξεκινήσει να εμφανίζεται είναι αυτός της εκπαίδευσης. Όπως κάθε νέα τεχνολογία έχει φανατικούς πολέμιους αλλά και θερμούς υποστηρικτές. Μπορεί η χρήση των υπολογιστών να προχώρησε ένα βήμα παρά πέρα τους τρόπους εκπαίδευσης αλλά η συγκεκριμένη τεχνολογία συνεπάγεται κάτι τελείως διαφορετικό, καθώς η εκπαίδευσης δεν περιορίζεται σε ένα σχολείο, σπίτι, υπολογιστή αλλά σε μία φορητή συσκευή που οποιοσδήποτε την έχει μαζί του ανά πάσα στιγμή, καταργώντας ουσιαστικά γεωγραφικούς περιορισμούς άλλων τεχνολογιών. Αυτό είναι και το μεγαλύτερο της πλεονέκτημα καθώς συνδυάζει όλες τις προηγούμενες τεχνολογίες ( υπολογιστές, Διαδίκτυο, video-conference κτλ) με την γεωγραφική ελευθερία. Από την άλλη πολλοί είναι οι πολέμοι της και κυρίως στην χρήση της από μαθητές νεαρής ηλικίας ακόμα, οι οποίοι δεν έχουν την ανάλογη εξοικείωση από την χρήση τέτοιων τεχνολογιών. Φυσικά με την κατάλληλη καθοδήγηση όλα αυτά είναι δυνατόν να ξεπεραστούν. Τέλος το μοναδικό σημαντικό μειονέκτημα της συγκεκριμένης τεχνολογίας είναι οι επιδράσεις της στην υγεία καθώς γίνεται χρήση ασύρματων τεχνολογιών που επιβαρύνουν την σωματική υγεία των ανθρώπων και ειδικά των ατόμων νεαρής ηλικίας.

Προς το παρόν γίνονται μελέτες με διφορούμενα αποτελέσματα οπότε δεν μπορούμε να πάρουμε πιο συγκεκριμένη θέση πάνω στο συγκεκριμένο ακόμα .

## 2.5 Υφιστάμενα προγράμματα για smartphones

Φυσικό είναι με την ραγδαία αύξηση των smart phones να παρατηρείται αντίστοιχη αύξηση και των εφαρμογών στα διαδικτυακά μαγαζιά των αντίστοιχων φορητών λειτουργικών συστημάτων . Η εφαρμογή που αναπτύξαμε αποτελείται από δύο μέρη ένα που είναι καθαρά το εκπαιδευτικό κομμάτι, όπου το παιδί έχει την δυνατότητα να δει τους αριθμούς και να αλληλεπιδράσει μαζί τους . Το δεύτερο μέρος της εφαρμογής αφορά ένα παιχνίδι το οποίο δίνει την δυνατότητα στο παιδί παίζοντας να μάθει τους αριθμούς ( αναλυτικά θα αναφερθούμε στην εφαρμογή σε επόμενη ενότητα ) . Όπως είναι λογικό η εφαρμογή μας θα βρεθεί "αντιμέτωπη" με τις ήδη υπάρχουσες εφαρμογές στο online κατάστημα της **Microsoft** για τα **Windows Phone** το λεγόμενο **Marketplace**. Παρόλο που η προσπάθεια της Microsoft στο συγκεκριμένο τομέα ότι έχει ξεκινήσει το marketplace αυτή την στιγμή αριθμεί πάνω από 100.000 εφαρμογές και παρουσιάζει τους μεγαλύτερους ρυθμούς ανάπτυξης αυτή την στιγμή με τα αντίστοιχα της Google και Apple .

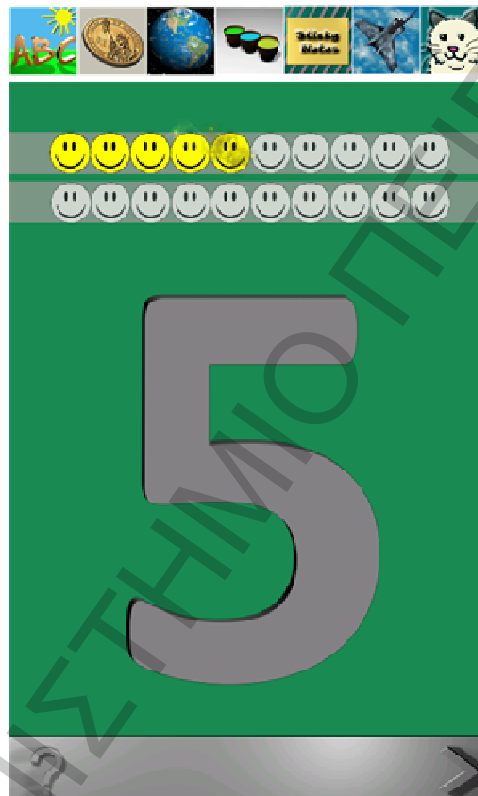
Όπως είναι λογικό αυτή την στιγμή υπάρχουν διαθέσιμες εφαρμογές που ασχολούνται με το θέμα της εκπαίδευσης στο **marketplace** . Παρακάτω θα προσπαθήσουμε να αναλύσουμε που προσπαθούμε να στοχεύσουμε με την εφαρμογή που αναπτύξαμε και φυσικά τα πλεονεκτήματα της και όποια μειονεκτήματα υπάρχουν έναντι των άλλων εφαρμογών.

Η εφαρμογή μας "Αριθμάκια Παζλάκια" μας απευθύνεται σε μικρά παιδιά και έχει ως κύριο σκοπό να τους μάθει τους αριθμούς. Αντίστοιχες εφαρμογές που υπάρχουν αυτή την στιγμή στο marketplace είναι οι ακόλουθες : "KiddieApps - Learn Numbers" , "Play With Numbers" , "Numbers" . [11]

Το μεγαλύτερο πλεονέκτημα της εφαρμογής μας σε σχέση με τις υπόλοιπες είναι ότι αυτή τη στιγμή είναι η μοναδική εφαρμογή με πλήρως ελληνικό περιεχόμενο, οι περισσότερες αυτή την στιγμή που είναι διαθέσιμες είναι κυρίως στα αγγλικά με κάποιες ελάχιστες εξαιρέσεις σε κάποιες άλλες γλώσσες .

Ένα ακόμα πλεονέκτημα της εφαρμογής μας είναι ότι παρουσιάζει με πλήρη τρόπο τον κάθε αριθμό δηλαδή δίνει την δυνατότητα στο χρήστη να δει τον αριθμό ως νούμερο, ως λέξη και φυσικά να τον ακούσει ως ηχητικό αρχείο . Ειδικά το τελευταίο κομμάτι της εφαρμογής η δυνατότητα της ακουστικής μάθησης είναι ίσως η πιο σημαντική καθώς σχεδόν όλες οι

υπόλοιπες εφαρμογές αναλώνονται κυρίως μόνο στην παρουσίαση του αριθμού καθώς και στην αλληλουχία των αριθμών κάτι το οποίο περιέχει ούτως η άλλως και η δική μας εφαρμογή. Σημαντικό είναι επίσης το ότι έχουμε διαλέξει ένα πιο οικείο περιβάλλον για τα παιδιά με την χρησιμοποίηση του πίνακα αλλά ταυτόχρονα και με τους αριθμούς σε πολύχρωμα σχέδια κάτι το οποίο δεν ισχύει για τις δύο παραπάνω εφαρμογές όπου το περιβάλλον τους είναι πιο λιτό και μη φιλικό σε μικρά παιδιά. (εικόνα 2.5-1)



εικόνα 2.5-1

Το σημαντικότερο πλεονέκτημα ίσως της εφαρμογής μας είναι το εκπαιδευτικό παιχνίδι το οποίο περιέχει το οποίο δίνει μια άλλη νότα στην εκμάθηση των αριθμών. Σε αντίθεση με τις υπόλοιπες εφαρμογές που είτε δεν περιέχουν καθόλου παιχνίδι είτε αρέσκονται σε παιχνίδια του τύπου ο χρήστης να πρέπει να μετρήσει πόσα αντικείμενα είναι στην οθόνη του κινητού και να επιλέξει τον σωστό αριθμό, στα "Αριθμάκια Παζλάκια" το εκπαιδευτικό παιχνίδι είναι πιο διαδραστικό και εκπαιδευτικό καθώς ο χρήστης μέσω ενός ruzzle στο οποίο έχουν μπει τα κομμάτια σε λάθος σειρά με τη χρήση της οθόνης αφής, πρέπει να σύρει τα κομμάτια τα οποία

έχουν πάνω τους αριθμούς από το 1 μέχρι το 8 , να τα βάλει στην σωστή σειρά για να εμφανιστεί ο τελευταίος αριθμός το δέκα.

Τελευταίο αλλά αρκετά σημαντικό είναι ότι η εφαρμογή μας είναι δωρεάν με μόνη επιβάρυνση στο χρήστη την εμφάνιση κάποιων διαφημιστικών μηνυμάτων σε συγκεκριμένα σημεία της εφαρμογής.

Αν θέλαμε να παρουσιάσουμε κάποια μειονεκτήματα της εφαρμογής μας θα ήταν τα ακόλουθα :

- το περιορισμένο εύρος των αριθμών καθώς δίνει την δυνατότητα εκμάθησης των αριθμών από το ένα μέχρι το δέκα ενώ υπάρχουν εφαρμογές με μεγαλύτερο αριθμητικό εύρος για παράδειγμα η "KiddieApps - Learn Numbers".[11]
- καθώς και η απουσία των αριθμητικών πράξεων ( όπως πρόσθεση και αφαίρεση ) οι οποίες υπάρχουν σε άλλες εφαρμογές όπως στην "Play With Numbers" (εικόνα 2.5-2) [11].



εικόνα 2.5-2

## 3. ΠΑΡΟΥΣΙΑΣΗ ΣΥΣΤΗΜΑΤΟΣ

### 3.1 Ενότητα "Μαθήματα"

Η εφαρμογή μας "**Αριθμάκια Παζλάκια**" όπως ήδη έχουμε αναφέρει συνδυάζει την εκπαίδευση με το παιχνίδι και σκοπός της είναι μέσα από ένα απλό και φιλικό σε μικρά παιδιά περιβάλλον να τους μάθει τους αριθμούς από το ένα μέχρι και το δέκα , δίνοντας τους παράλληλα την δυνατότητα να διασκεδάσουν μέσω ενός εκπαιδευτικού παιχνιδιού που



περιλαμβάνει. Στην συγκεκριμένη ενότητα θα αναλύσουμε το **περιβάλλον** της εφαρμογής μας και τα ιδιαίτερα χαρακτηριστικά του γνωρίσματα. Καταρχήν η εφαρμογή μας αποτελείται από τρεις κυρίως ενότητες . Τα "**Μαθήματα**" που όπως ορίζει και το όνομα της είναι η ενότητα που περιλαμβάνει το διδακτικό περιεχόμενο θα μπορούσαμε να πούμε. Το "**Παιχνίδι**" όπου περιλαμβάνεται το εκπαιδευτικό παιχνίδι της εφαρμογής μας και τέλος οι "**Οδηγίες**" που παρουσιάζονται εν συντομία οι οδηγίες χρήσης της εφαρμογής μας.[6]

Αφού τρέξει η εισαγωγική εικόνα της εφαρμογής μεταφερόμαστε στην κύρια σελίδα της εφαρμογής μας ( εικόνα 3.1-1 ) . Αυτό που προσπαθήσαμε να πετύχουμε είναι να δώσουμε ένα εκπαιδευτικό τόνο στην εφαρμογή μας ορίζοντας ως background ένα σχολικό πίνακα αλλά παράλληλα όχι τόσο αυστηρό ύφος κάνοντας χρήση των αριθμών - cartoon.



Εικόνα 3.1-1

Περιλαμβάνονται τέσσερα κουμπιά όπου το καθένα μας οδηγεί στην αντίστοιχη ενότητα μεγάλη προσοχή έχει δοθεί στο να είναι όσο το δυνατόν πιο σαφής και κατανοητή η επιλογή του κάθε κουμπιού.

Πατώντας στο κουμπί "**Μαθήματα**" μεταφερόμαστε στην αντίστοιχη ενότητα . Έχουμε διατηρήσει την ίδια εικόνα στο background για να υπάρχει μια συνοχή στην εφαρμογή μας. Στην συγκεκριμένη ενότητα παρουσιάζεται μέσω εικόνας τόσο ο αριθμός που θέλουμε να απεικονίσουμε ως σύμβολο τόσο και ως λέξη ακριβώς από πάνω (εικόνα 3.1-2 ).



Εικόνα 3.1-2

Επιπλέον ο χρήστης έχει τις εξής δυνατότητες να μετακινηθεί στον επόμενο ή προηγούμενο αριθμό πατώντας το αντίστοιχο βελάκι και εδώ όπως παρατηρούμε έχουν χρησιμοποιηθεί εικόνες με cartoon για τα βελάκια πλοήγησης καθώς και χρήση ηχητικού εφέ όταν ο χρήστης το πατήσει με ήχο ενός ελατηρίου για να τονίσει το παιδικό ύφος της εφαρμογής μας. Αξίζει να σημειωθεί ότι έχει χρησιμοποιηθεί διαφορετικός ήχος αναλόγως με το αν ο χρήστης πατήσει το βελάκι για να μετακινηθεί στην επόμενη σελίδα ή στην προηγούμενη. Άλλες δυνατότητες που έχει ο χρήστης είναι να μεταβεί στο αρχικό μενού πατώντας την αντίστοιχη λέξη στο πάνω μέρος της εφαρμογής ή να ακούσει τον αριθμό ως ηχητικό αρχείο πατώντας πάνω στην εικόνα ή στην λέξη του ενώ παράλληλα αναπαράγεται και ένα κινούμενο εφέ. Το ίδιο ισχύει και για τους επόμενους αριθμούς από το 2 μέχρι και το 10 (εικόνα 3.1-3) .



Εικόνα 3.1-3

## 3.2 Ενότητα "Παιχνίδι"

Πατώντας το κουμπί "Παιχνίδι" μεταφερόμαστε στην αντίστοιχη σελίδα ( Εικόνα 3.2-1 ) .

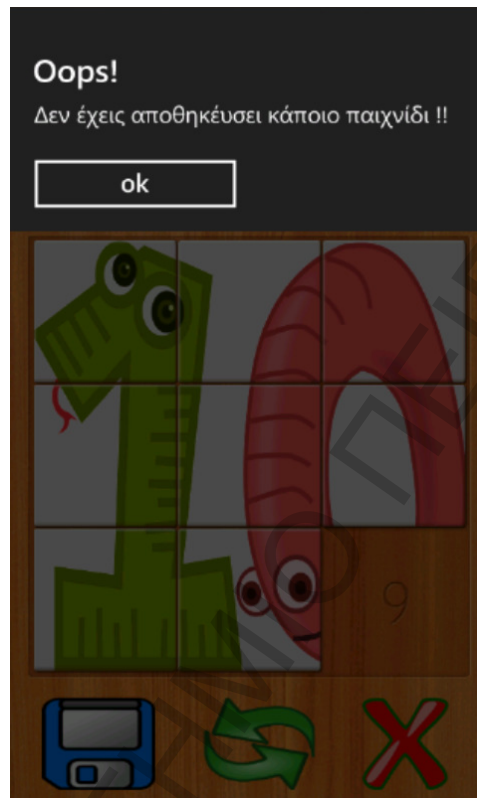


Εικόνα 3.2-1

Στην συγκεκριμένη σελίδα λαμβάνει χώρα το εκπαιδευτικό μέρος της εφαρμογής μας με τους αριθμούς. Σκοπός του παιχνιδιού είναι ο χρήστης να βάλει τα κομμάτια του ruzzle στην σωστή σειρά από το ένα μέχρι το εννιά έτσι ώστε να σχηματιστεί η εικόνα του επόμενου αριθμού του δέκα. Στην συγκεκριμένη σελίδα διαλέξαμε να χρησιμοποιήσουμε άλλη εικόνα για το background έτσι ώστε να υπάρχει διαχωρισμός με την ενότητα των "**Μαθημάτων**". Στην συγκεκριμένη σελίδα ο χρήστης έχει τις εξής δυνατότητες :

- Να μεταβεί στο αρχικό μενού πατώντας την επιλογή "μενού".
- Να ξεκινήσει το παιχνίδι πατώντας πάνω στην εικόνα υπάρχει και σχετικό μήνυμα που το διευκρινίζει.
- Να αποθηκεύσει την τρέχουσα κατάσταση του παιχνιδιού πατώντας την εικόνα με την κασέτα.
- Να διαγράψει την τρέχουσα αποθηκευμένη κατάσταση του παιχνιδιού πατώντας την εικόνα με το σύμβολο "X".

- Να φορτώσει την τελευταία κατάσταση παιχνιδιού που έχει σώσει πατώντας την εικόνα με το σύμβολο του refresh. Αν δεν έχει σώσει του βγάζει σχετικά μήνυμα (Εικόνα 3.2-2 )



Εικόνα 3.2-2

Καθώς η εφαρμογή απευθύνεται σε παιδιά όπως και στο υπόλοιπο κομμάτι της προσπαθήσαμε να διατηρήσουμε το καρτουνίστικο περιεχόμενο της με την χρησιμοποίηση των αντίστοιχων εικόνων ενώ παράλληλα να χρησιμοποιήσουμε σύμβολα τα οποία να εκφράζουν όσο είναι το δυνατόν πιο άμεσα την λειτουργία τους πχ. χρήση κασέτας υπολογιστή για την αποθήκευση.

Όταν ο χρήστης πατήσει την εικόνα με τον αριθμό αμέσως η εικόνα "σπάει" σε μικρότερα τετραγωνάκια με τυχαία σειρά. Ο χρήστης ακουμπώντας το κάθε τετράγωνο και μετακινώντας το χέρι μπορεί να το μεταφέρει όπου υπάρχει κενό σημείο με σκοπό να ξανασχηματίσει την αρχική εικόνα του αριθμού Δέκα. Παράλληλα το μήνυμα για την έναρξη του παιχνιδιού έχει

αντικατασταθεί με ένα νέο το οποίο καταγράφει τον αριθμό των κινήσεων που έχει κάνει μέχρι τώρα. ( Εικόνα 3.2-3 )



Εικόνα 3.2-3

Τέλος εφόσον ο χρήστης καταφέρει να ξανασηματίσει το αριθμό δέκα έχει δηλαδή ολοκληρώσει το μίνι-παιχνίδι έχουμε χρησιμοποιήσει ένα εφέ περιστροφής της εικόνας του

αριθμού και της εμφάνισης ενός banner με μήνυμα επιβράβευσης το οποίο αναφέρει "Συγχαρητήρια" ( Εικόνα 3.2-4 ).



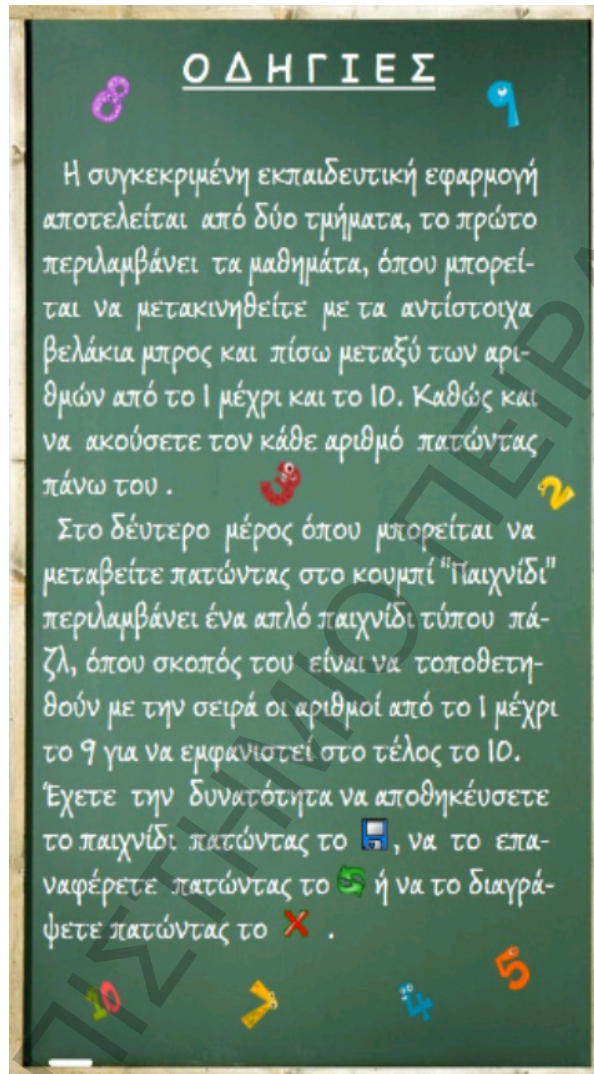
Εικόνα 3.2-4

### 3.3 Ενότητα "Οδηγίες" και "BlueShyMonkey Apps"

Πατώντας την επιλογή "**Οδηγίες**" μεταφερόμαστε στην σελίδα με τις συνοπτικές οδηγίες της εφαρμογής μας ( Εικόνα 3.3-1 ) όπως και στο υπόλοιπο κομμάτι της εφαρμογής μας έχουμε επιλέξει να χρησιμοποιήσουμε τα ίδια νούμερα και τα σύμβολα έτσι ώστε να υπάρχει το

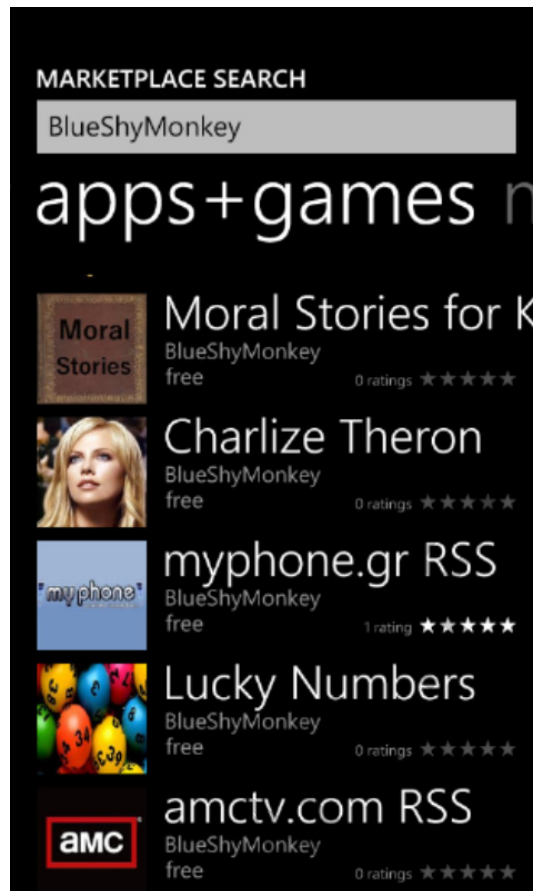


αίσθημα της συνοχής και να μην νομίζει ο χρήστης ότι κάθε σελίδα είναι και μια διαφορετική εφαρμογή.



Εικόνα 3.3-1

Τέλος έχουμε αφήσει την επιλογή "**BlueShymonkey Apps**" όπου ο χρήστης πατώντας στο συγκεκριμένο κουμπί τον μεταφέρει στο marketplace ( προϋποθέτει να υπάρχει πρόσβαση στο Διαδίκτυο ) των Windows Phone με κριτήριο αναζήτησης εφαρμογών τον εκδότη "**BlueshyMonkey**" στην προκειμένη περίπτωση εμάς (Εικόνα 3.3-2 ).



Εικόνα 3.3-2

## 4. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ

### 4.1 Προγραμματιστικές Πλατφόρμες

Το windows phone 7 υποστηρίζει δύο μοντέρνες και δημοφιλή προγραμματιστικές πλατφόρμες το Silverlight και το XNA.

- Το Silverlight είναι ένα υπό-προϊόν του client-based Windows Presentation Foundation (WPF) το οποίο δίνει την δύναμη στους προγραμματιστές να δημιουργούν user interface με μια μίξη παραδοσιακών κουμπιών , υψηλού περιεχομένου κείμενο, διανυσματικά γραφικά ( vector graphics ) , πολυμέσα , animation και σύνδεση δεδομένων ( data binding ) . Το Windows Phone 7 επεκτείνει αυτές τις δυνατότητες σε φορητές συσκευές .
- XNA είναι η πλατφόρμα της Microsoft για τα παιχνίδια υποστηρίζει τόσο 2D γραφικά όσο και 3D με παραδοσιακές αρχιτεκτονικές παιχνιδιών ( π.χ. game-loop ) .

Όλα τα προγράμματα για Windows Phone 7 γράφονται σε .NET . Προς το παρόν υποστηρίζονται ως γλώσσες προγραμματισμού η C# και η Visual Basic αν και υπάρχει μεγάλο ενδεχόμενο στο μέλλον να υποστηριχθεί και η C++ . Για να αναπτύξουμε προγράμματα σε Windows Phone 7 θα χρειαστούμε το Visual Studio 2010 , το Windows Phone SDK καθώς προαιρετικά και το Microsoft Expression Blend με το οποίο μπορούμε να αναπτύξουμε πολύ εύκολα εφέ και animation για το Silverlight. Η Microsoft διαθέτει εντελώς δωρεάν το πακέτο Microsoft Visual Studio Express for Windows Phone το οποίο περιλαμβάνει όλα τα απαραίτητα

εργαλεία για να αναπτύξουμε εφαρμογές για την συγκεκριμένη πλατφόρμα , καθώς και τον απαραίτητο εξομοιωτή ( emulator ) .

Το Silverlight και το XNA για Windows phone 7 μοιράζονται μερικές βιβλιοθήκες (libraries ) , δηλαδή μπορούμε να χρησιμοποιήσουμε XNA βιβλιοθήκες σε Silverlight προγράμματα και το αντίθετο . Πριν δημιουργήσουμε ένα project στο Visual Studio το πρώτο πράγμα που θα πρέπει να αποφασίσουμε είναι αν θα είναι ένα Silverlight πρόγραμμα ή ένα XNA πρόγραμμα .

Γενικά επιλέγουμε Silverlight για να γράψουμε προγράμματα τα οποία μπορούμε να τα κατηγοριοποιήσουμε ως εφαρμογές. Αυτά τα προγράμματα είναι σχεδιασμένα από ένα συνδυασμό σημάτων ( markup ) και κώδικα . Οι σημάσεις είναι η Extensible Application Markup Language ή XAML . Η XAML κυρίως ορίζει μια διάταξη από user-interface controls και panels. Πίσω από αυτά τα controls τρέχει κώδικας ο οποίος χειρίζεται διάφορα γεγονότα .

Από την άλλη το XNA χρησιμοποιείται κυρίως για να αναπτύξουμε υψηλού επιπέδου παιχνίδια. Για δύο διαστάσεων παιχνίδια (2D) ορίζουμε sprites και background περιβάλλοντα , ενώ για τριών διαστάσεων παιχνίδια (3D) ορίζουμε μοντέλα σε 3D χώρους . Οι ενέργειες του παιχνιδιού , οι οποίες περιλαμβάνουν κινούμενα γραφικά αντικείμενα στην οθόνη και εντολές του χρήστη , συγχρονίζονται μέσω ένθετων XNA game-loop .

Οι διαφοροποιήσεις μεταξύ των Silverlight εφαρμογών και των XNA είναι εμφανής αλλά όχι απαγορευτικές . Μπορούμε να χρησιμοποιήσουμε Silverlight για να γράψουμε παιχνίδια και μπορούμε ακόμα και να γράψουμε παραδοσιακές εφαρμογές χρησιμοποιώντας XNA , αν και αυτό μπορεί να αποδειχτεί πρόκληση.

Συγκεκριμένα το Silverlight μπορεί να είναι ιδανικό για την ανάπτυξη για παιχνίδια τα οποία χρησιμοποιούν ελάχιστα γραφικά , ή χρησιμοποιούν διανυσματικά γραφικά (vector) παρά bitmaps γραφικά .

Η Microsoft έχει ορίσει το Silverlight ως το "πρόσωπο" της στο cloud , έτσι οι υπηρεσίες στο cloud και το Windows Azure είναι σημαντικό κομμάτι στην ανάπτυξη των Windows Phone 7. Το Windows Phone είναι "cloud-ready" . Τα προγράμματα λαμβάνουν υπόψη την τοποθεσία (location-aware) έχουν πρόσβαση σε χάρτες και άλλα δεδομένα μέσω του Bing και του Windows Live. Μια από τις διαθέσιμες cloud υπηρεσίες είναι το Xbox Live το οποίο επιτρέπει σε XNA προγράμματα να συμμετέχουν σε online παιχνίδια πολλών παιχτών (multiplayer games) .

Τα προγράμματα που γράφονται για Windows Phone 7 μπορούν να διατεθούν είτε δωρεάν είτε με χρέωση από το Windows Phone MarketPlace, το οποίο παρέχει υπηρεσίες εγγραφής

και πιστοποιεί ότι τα προγράμματα τηρούν τα τις απαραίτητες προδιαγραφές αξιοπιστίας , αποτελεσματικότητας και καλής συμπεριφοράς . [8]

## 4.1.1 Γλώσσα επισήμανσης XAML

- Γενικά

Η XAML δεν είναι γλώσσα προγραμματισμού , αλλά μια γλώσσα σήμανσης . Εφαρμόζεται στο μοντέλο προγραμματισμού του **.NET Framework** και απλοποιεί την δημιουργία του περιβάλλοντος εργασίας (**UI**) για μια εφαρμογή του είδους . Μπορούμε να δημιουργήσουμε ορατά στοιχεία **User Interface (UI)** με την χρήση της **XAML** και στην συνέχεια να ορίσουμε τις λειτουργίες τους κατά τον χρόνο εκτέλεσης με την χρήση αρχείων που τρέχουν πίσω από αυτά (**code-behind files**) . Αυτό είναι και που την κάνει να ξεχωρίζει από άλλες γλώσσες σήμανσης το άμεσο δέσιμό της . Όταν αναπαριστάτε ως κείμενο , τα αρχεία **XAML** είναι αρχεία **XML** τα οποία έχουν συνήθως την επέκταση **.xaml** .

Το παρακάτω παράδειγμα (εικόνα 4.1.1-1 ) δείχνει πως μπορούμε να δημιουργήσουμε ένα κουμπί (**button**) ως μέρος ενός **User Interface**. Το συγκεκριμένο είναι απλώς για να μας δώσει μια γεύση από την χρήση της **XAML** (δεν είναι πλήρης παράδειγμα).[10]



```
XAML
<StackPanel>
  <Button Content="Click Me"/>
</StackPanel>
```

Εικόνα 4.1.1-1

- Σύνταξη της XAML

Οι ακόλουθες ενότητες εξηγούν τις βασικές μορφές της σύνταξης XAML και δίνουν ένα σύντομο παράδειγμα σύνταξης. Ένα αντικείμενο (**object**) ενός στοιχείου δηλώνει συνήθως ένα στιγμιότυπο (**instance**) του τύπου. Η σύνταξη ενός αντικειμένου ξεκινάει πάντα με μια αγκύλη ανοίγματος ( < ). Μετά από αυτό μπορούμε να δηλώσουμε προαιρετικά χαρακτηριστικά για το στοιχείο αντικειμένου. Για να ολοκληρωθεί το αντικείμενο χρησιμοποιούμε στο τέλος μια αγκύλη κλεισίματος ( > ). Μπορούμε επίσης να χρησιμοποιήσουμε μια αυτόματη μορφή κλεισίματος που δεν έχει κανένα περιεχόμενο συμπληρώνοντας την ετικέτα με μια κάθετο και το κλείσιμο αγκύλης διαδοχικά ( /> ) παρουσιάζεται στο παράδειγμα παραπάνω (εικόνα 4.1.1-1). Το συγκεκριμένο παράδειγμα παρουσιάζει δύο αντικείμενα το <StackPanel> ( με περιεχόμενο και αγκύλη κλεισίματος μετά ) και ένα <Button .../> ( με αυτόματο τρόπο κλεισίματος ) . [10]

- Ιδιότητες


Οι ιδιότητες (**Properties**) ενός στοιχείου μπορεί αρκετές φορές να αναφέρονται και ως χαρακτηριστικά (**attributes**) . Στη σύνταξη ενός χαρακτηριστικού καταρχήν ορίζουμε αρχικά το χαρακτηριστικό μέσα στο αντικείμενο το οποίο ακολουθείτε από το συντελεστή εκχώρησης (=) . Η τιμή ενός χαρακτηριστικού ορίζεται πάντα ως **string** και περιέχεται μέσα σε εισαγωγικά . Για παράδειγμα η ακόλουθη σύνταξη δημιουργεί ένα κουμπί (button) που έχει κόκκινο κείμενο, μπλε φόντο και το περιεχόμενο του (εικόνα 4.1.1-2) .

```
<Button Background="Blue" Foreground="Red" Content="This is a button"/>
```

Εικόνα 4.1.1-2

- XAML Root Elements and XAML Namespaces

Ένα αρχείο XAML πρέπει να έχει μόνο ένα ριζικό στοιχείο ( **root element** ) , προκειμένου να είναι τόσο καλοσχεδιασμένο **αρχείο XML** και ένα έγκυρο **αρχείο XAML**. Για τυπικά σενάρια **WPF**, μπορείτε να χρησιμοποιήσετε ένα στοιχείο ρίζας που έχει εξέχουσα σημασία στο μοντέλο εφαρμογή **WPF** (για παράδειγμα, στο παράθυρο ή σελίδας για μια σελίδα, ResourceDictionary για κάποιο εξωτερικό λεξικό). Το ακόλουθο παράδειγμα δείχνει το στοιχείο ρίζας ενός τυπικού αρχείου **XAML** για μια σελίδα **WPF**, με το ριζικό στοιχείο της σελίδας (εικόνα 4.1.1-3) . [10]



```
XAML
<Page
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  ...
</Page>
```

Εικόνα 4.1.1-3

Το στοιχείο ρίζας περιέχει επίσης τα χαρακτηριστικά xmlns και xmlns:x. Αυτά τα χαρακτηριστικά δείχνουν σε έναν επεξεργαστή XAML ποια XAML namespaces περιέχουν τους ορισμούς τύπου για την υποστήριξη των τύπων στα οποία θα αναφέρονται ως στοιχεία στην συνέχεια του project . [10]

## 4.1.2 ISOLATED STORAGE

Το Isolated Storage είναι ο μηχανισμός που μας επιτρέπει να αλληλεπιδράσουμε με το FileSystem στα WP7. Ο προγραμματιστής έχει πρόσβαση στο IsolatedStorage μέσω των δομών

που του παρέχει το namespace [System.IO.IsolatedStorage](#). Πριν ξεκινήσουμε, μερικές παρατηρήσεις :

- Κάθε εφαρμογή έχει τον δικό της "χώρο" στο filesystem. Ο χώρος αυτός δεν έχει συγκεκριμένο μέγεθος, αλλά περιορίζεται από την κάθε συσκευή, καθώς και τον χώρο που χρησιμοποιούν οι υπόλοιπες εφαρμογές.
- Κάθε εφαρμογή μπορεί να προσπελάσει μόνο τον χώρο που της αντιστοιχεί στο filesystem και δεν μπορεί να δει τον χώρο άλλων εφαρμογών.
- Σε αντίθεση με τις συσκευές, το Isolated Storage στον Windows Phone Emulator σβήνεται κάθε φορά που κλείνουμε τον Emulator.
- Αν βγάλουμε κάποιο update για την εφαρμογή μας, στο οποίο έχει αλλάξει η δομή που χρησιμοποιούμε για το Isolated Storage, τότε είμαστε εμείς υπεύθυνοι για να την αλλάξουμε "χειροκίνητα".
- Αν θέλουμε για λόγους debugging να δούμε το Isolated Storage της εφαρμογής μας (τόσο σε συσκευές, Emulator) μπορούμε να χρησιμοποιήσουμε τον *WP7 Isolated Storage Explorer*. [10]

### 4.1.3 Γλώσσα Σχεδίασης METRO

- Γενικά

Η **METRO** είναι μια γλώσσα σχεδίασης, οι βασικές αρχές της είναι οι εξής ακόλουθες:



- Γρήγορη και αποκριτική.
- Επικεντρώνεται στα βασικά έργα.
- Μειώνει την ύπαρξη περιττών στοιχείων.
- Αφαιρεί οπτικό υλικό που δεν αφορά περιεχόμενο.
- Επικεντρώνεται στις πληροφορίες και το περιεχόμενο και όχι στα χειριστήρια.
- Προσφέρει άμεση αλληλεπίδραση με το περιεχόμενο.
- Τα τυπογραφικά στοιχεία εκτός από ευανάγνωστα είναι και καλαίσθητα.
- Χρησιμοποιεί τυπογραφικά στοιχεία σε διαφορετικά μεγέθη και βάρη έτσι ώστε να προσφέρει δομημένη πληροφορία και μοναδικές οπτικές δυνατότητες σχεδιασμού.
- Χρησιμοποιεί πλαίσια για την βελτίωση της χρηστικότητας.
- Δίνει μεγάλη βαρύτητα στα εφέ μετάβασης.
- Εστιάζει στην ατομικότητα και τα καθήκοντα της.
- Καλή οργάνωση πληροφοριών και εφαρμογών.

Στην παρακάτω εικόνα ( εικόνα 4.1.3-1 ) , παρουσιάζονται δύο στιγμιότυπα από το METRO UI τόσο από την αρχική οθόνη αριστερά , όσο και από την οθόνη με την λίστα των εφαρμογών . Όπως παρατηρούμε συμβαδίζουν με τις βασικές αρχές που αναφέραμε παραπάνω.



Εικόνα 4.1.3-1

Στις παραπάνω αρχές στηρίζονται όμως εκτός από το ίδιο το λειτουργικό και οι εφαρμογές.

Γενικά η δομή μιας σελίδας διακρίνεται στους παρακάτω τομείς (εικόνα 4.1.3-2) :

- **Status bar** : Είναι μια μικρή μπάρα η οποία παρουσιάζει μερικές βασικές πληροφορίες για την λειτουργία του τηλεφώνου όπως η ώρα , η στάθμη της μπαταρίας , το σήμα λήψης κτλ.
- **Application space** : Ο βασικός χώρος που εμφανίζεται το περιεχόμενο της εφαρμογής.
- **Application bar** : Η βασική μπάρα της εφαρμογής στην οποία μπορούμε να τοποθετήσουμε μέχρι και 4 κουμπιά , στην συγκεκριμένη τοποθετούνται οι βασικές λειτουργίες της εφαρμογής, στις οποίες ο χρήστης πρέπει να έχει άμεση πρόσβαση.

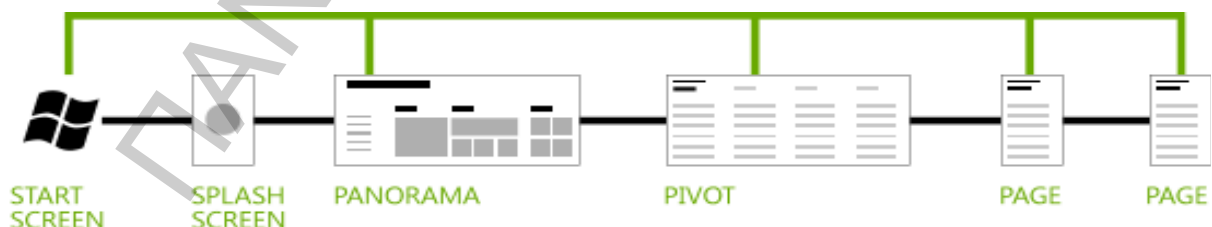
- **Application bar menu** : Ένα αναδυόμενο μενού το οποίο εμφανίζεται πατώντας πάνω στην application bar και περιέχει κάποιες επιπλέον λειτουργίες τις εφαρμογής.



Εικόνα 4.1.3-2

- **Βασικά εργαλεία του METRO UI**

Στην παρακάτω εικόνα ( εικόνα 4.1.3-3 ) παρουσιάζεται η πορεία από την αρχική οθόνη μέχρι και την πλοήγηση σε κάποια σελίδα μιας εφαρμογής . [10]



Εικόνα 4.1.3-3

Τα βασικά εργαλεία που χρησιμοποιούνται είναι συνοπτικά τα παρακάτω :

- Start Tiles
- Splash Screen
- Pivot
- Panorama
- Single Page
- Application Bar

## Start Tiles

Τα tiles ή πλακίδια ( εικόνα 4.1.3-4 ) αποτελούν μια εύκολα αναγνωρίσιμη οπτική συντόμευση για μια εφαρμογή ή το περιεχόμενο της , την οποία οι χρήστες μπορούν να την τοποθετήσουν στην αρχική οθόνη του κινητού τους. Μπορούμε να πούμε ότι αποτελούν έναν ανανεωτικό σχεδιασμό των γνωστών μας εικονιδίων. Τα βασικά τους γνωρίσματα είναι τα παρακάτω :

- Περιορίζουν την χρήση των λέξεων στις απολύτως απαραίτητες.
- Οι πληροφορίες της επιφάνειας είναι απαραίτητα σχετικές με την εφαρμογή.
- Δεν χρησιμοποιούμε άσπρο background.
- Δεν χρησιμοποιούμε στρογγυλεμένες γωνίες.
- Εάν χρησιμοποιούμε πολλαπλά tiles θα πρέπει να είναι οπτικά συνεκτικά μεταξύ τους και να έχουν αναγνωρίσιμο ύφος και θέμα.

Στην παρακάτω εικόνα βλέπουμε το μια αρχική οθόνη με διάφορα tiles που έχει προσθέσει ο χρήστης όπως του skype. [10]



Εικόνα 4.1.3-4

## Splash screen

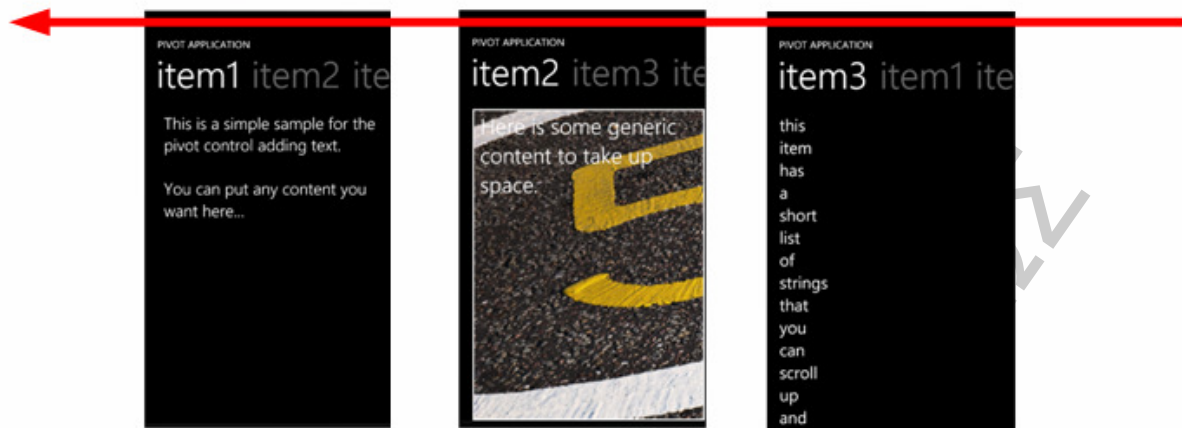
Η Splash Screen ή οθόνη εκκίνησης (εικόνα 4.1.3-5) αποτελεί την αρχική οθόνη της εφαρμογής αυτή που εμφανίζεται δηλαδή πρώτη στο χρήστη. Αυτό που θα πρέπει να προσέξουμε είναι να είναι όσο το δυνατόν πιο απλή και άμεση και να δίνει έμφαση στην εταιρεία ανάπτυξης και στην εφαρμογή που πρόκειται να χρησιμοποιήσει ο χρήστης. Παρακάτω βλέπουμε έναν πολύ καλό παράδειγμα από την οθόνη εκκίνησης της εφαρμογής skype. [10]



Εικόνα 4.1.3-5

## **Pivot**

Το εργαλείο Pivot ( εικόνα 4.1.3-6 ) παρέχει ένα γρήγορο τρόπο για να διαχειριστούμε τις όψεις και τις σελίδες μιας εφαρμογής. Μπορεί να χρησιμοποιηθεί για το φιλτράρισμα μεγάλων συνόλων δεδομένων, προβολή μεγάλων συνόλων δεδομένων ή την εναλλαγή των όψεων μιας εφαρμογής ή σχετικού περιεχομένου. Το εργαλείο Pivot είναι ουσιαστικά ένα δοχείο για ένα δευτερεύων στοιχείο ελέγχου τα PivotItem, τα συγκεκριμένα περιλαμβάνουν αποκλειστικό περιεχόμενο για κάθε σελίδα όπως χειριστήρια , grid , συνδέσμους κτλ. Το εργαλείο Pivot τοποθετεί τα παραπάνω PivotItem το ένα δίπλα στο άλλο και διαχειρίζεται το κάθε επιμέρους PivotItem μέσω της πλοήγησης αριστερά ή δεξιά. Παρακάτω παρουσιάζεται ένα παράδειγμα (εικόνα 3.4-6) ενός εργαλείου Pivot το οποίο περιλαμβάνει τρία PivotItem



Εικόνα 4.1.3-6

Οι βασικές αρχές του εργαλείου Pivot είναι οι ακόλουθες:

- Πρέπει να χρησιμοποιείται μόνο για να παρουσιάσει αντικείμενα ή δεδομένα του ίδιου ή παρόμοιου τύπου ( για παράδειγμα φιλτραρισμένες προβολές των ίδιων δεδομένων).
- Ποτέ δεν πρέπει να χρησιμοποιείται ένα εργαλείο Pivot μέσα σε ένα εργαλείο Panorama ( θα αναφερθούμε παρακάτω στο συγκεκριμένο εργαλείο ) ή σε ένα άλλο εργαλείο Pivot.
- Οι σελίδες Pivot δεν πρέπει να χρησιμοποιούνται όταν έχουμε εργασίες ροής.
- Οι σελίδες Pivot είναι κυκλικές. Αυτό σημαίνει ότι όταν είμαστε στην τελευταία σελίδα (PivotItem) αν συνεχίσουμε να σύρουμε θα εμφανιστεί πάλι η πρώτη σελίδα , αυτό έχει ως αποτέλεσμα να δημιουργείται μια αδιάλειπτη ροή δεδομένων.
- Η επικεφαλίδα του Pivot έχει δεδομένο μέγεθος και δεν μπορεί να αλλάξει.
- Δεν χρησιμοποιούμε εργαλεία τα οποία μπορούν να μετακινούνται ή να κάνουν scroll μέσα σε ένα Pivot , όπως είναι sliders , χάρτες κτλ. Καθώς αυτό έχει ως αποτέλεσμα να μπερδεύεται ο διαχειριστής χειρονομίας. Για παράδειγμα αν έχουμε τοποθετήσει ένα slider σε ένα χάρτη και προσπαθήσουμε να τον σύρουμε προς τα αριστερά δεν είναι σαφές αν θέλουμε να μετακινήσουμε το slider του χάρτη ή να μετακινηθούμε στο επόμενο PivotItem.

Παρακάτω βλέπουμε ένα παράδειγμα (εικόνα 4.1.3-7 ) από την εφαρμογή Skype όπου γίνεται χρήση του εργαλείου Pivot για να ομαδοποιησει την διαδικασία για log in (Σύνδεση) και register (Εγγραφή) ενός χρήστη στην εφαρμογή όπως βλέπουμε έχουμε δύο PivotItem. [10]



Εικόνα 4.1.3-7

## Panorama

Το εργαλείο Panorama εμφανίζει μια προβολή από αντικείμενα στα οποία μπορούμε να μετακινηθούμε από πλευρά σε πλευρά. Είναι ένας container πλήρους οθόνης και μοντέλο πλοήγησης για μια εφαρμογή. Η πανοραμική εμπειρία αποτελεί μέρος των Windows Phone. Σε αντίθεση με τις τυπικές εφαρμογές που έχουν σχεδιαστεί να χωρούν στα όρια της οθόνης του κινητού, οι πανοραμικές εφαρμογές χρησιμοποιούν ένα μοναδικό τρόπο για να δείτε τα χειριστήρια , τα δεδομένα κτλ χρησιμοποιώντας ένα μεγάλο οριζόντιο καμβά που εκτείνεται πέρα από τα όρια της οθόνης. Το user interface αποτελείται από τύπους στρώματος που λειτουργούν με την δική τους ανεξάρτητη λογική. Μια εικόνα φόντου, ένα τίτλο πανοράματος, τίτλοι ενοτήτων του πανοράματος και τμήματα του πανοράματος. Η εικόνα φόντου είναι το χαμηλότερο επίπεδο και έχει ως στόχο να δώσει μια πλούσια αίσθηση όπως τα εξώφυλλα σε



ένα περιοδικό. Συνήθως το background είναι το πιο ορατό μέρος μιας εφαρμογής. Ο τίτλος του πανόραμα έχει ως στόχο να προσδιορίσει την εφαρμογή και θα πρέπει να είναι ορατός ανεξάρτητα από το πόσο ο χρήστης έχει εμβαθύνει στην εφαρμογή. Τα τμήματα πανοράματος είναι συστατικά της εφαρμογής και ενσωματώνουν άλλα χειριστήρια και δεδομένα. Τα τμήματα του πανοράματος κινούνται στο ίδιο ρυθμό με την κίνηση του δαχτύλου.

Οι βασικές αρχές του panorama είναι οι ακόλουθες :

- Χρησιμοποιούμε ενός χρώματος background η μια μεμονωμένη εικόνα για να μην δημιουργήσουμε σύγχυση στο χρήστη.
- Τόσο για λόγους απόδοσης όσο και για ευκολία του χρήστη δεν πρέπει να χρησιμοποιούμε πάνω από πέντε τομείς σε ένα panorama.
- Υποστηρίζει μόνο landscape προσανατολισμό.
- Δεν χρησιμοποιούμε ποτέ ένα εργαλείο Pivot μέσα σε ένα Panorama.
- Δεν χρησιμοποιούμε εργαλεία τα οποία μπορούν να μετακινούνται ή να κάνουν scroll μέσα σε ένα Pivot , όπως είναι sliders , χάρτες κτλ. Καθώς αυτό έχει ως αποτέλεσμα να μπερδεύεται ο διαχειριστής χειρονομίας. Για παράδειγμα αν έχουμε τοποθετήσει ένα slider σε ένα χάρτη και προσπαθήσουμε να τον σύρουμε προς τα αριστερά δεν είναι σαφές αν θέλουμε να μετακινήσουμε το slider του χάρτη ή να μετακινηθούμε στο επόμενο Panoramatem.

Παρακάτω εμφανίζεται ένα παράδειγμα (εικόνα 4.1.3-8) όπου εμφανίζεται η χρήση του εργαλείου Panorama στην εφαρμογή Skype. [10]

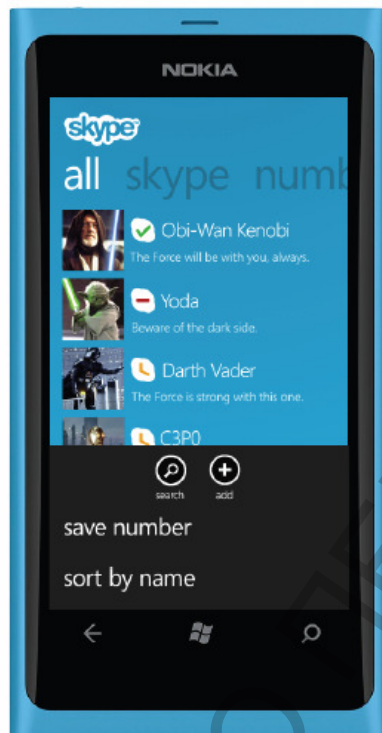


Εικόνα 4.1.3-8

## Application Bar

Η Application Bar παρέχει ένα χώρο για τις σημαντικότερες εργασίες μιας εφαρμογής. Για τις λιγότερες σημαντικές εργασίες, η εφαρμογή μπορεί να έχει ένα ενιαίο αναδυόμενο μενού που να περιλαμβάνονται σε αυτό. Η **applications bar** μένει πάντα στην ίδια μεριά της οθόνης όπως βρίσκονται τα πλήκτρα hardware και μπορεί να περιέχει μέχρι και **τέσσερα πλήκτρα** με λειτουργίες. Από την άλλη το μενού προσφέρει έναν εναλλακτικό τρόπο για τον χρήστη να έχει πρόσβαση σε συγκεκριμένες εργασίες. Το συγκεκριμένο μενού θα πρέπει να περιέχει δευτερεύουσες ή εξεζητημένες εργασίες.

Στο παρακάτω παράδειγμα (εικόνα 4.1.3-10 ) βλέπουμε ένα παράδειγμα από την application bar της εφαρμογής Skype. Η οποία έχει δυο βασικές λειτουργίες "search" και "add" στην application bar και δύο δευτερεύουσες στο application menu τις "save number" και "sort by name".[10]



Εικόνα 4.1.3-10

#### 4.1.4 Hardware

- Γενικά

Οι προγραμματιστές που έχουν εμπειρία από τις Windows Mobile συσκευές του παρελθόντος θα παρατηρήσουν σημαντικές αλλαγές στην στρατηγική της Microsoft για το Windows Phone 7. Η Microsoft είναι πολύ προσεχτική στα χαρακτηριστικά του hardware.

Οι αρχικές συσκευές με Windows Phone 7 θα έχουν μονό μία συγκεκριμένη ανάλυση οθόνης (περισσότερες αναλύσεις θα υποστηριχτούν με μελλοντικά update όπως το Tango και Apollo ). Πολλά ακόμα χαρακτηριστικά επιβάλλονται να υπάρχουν στις συσκευές.

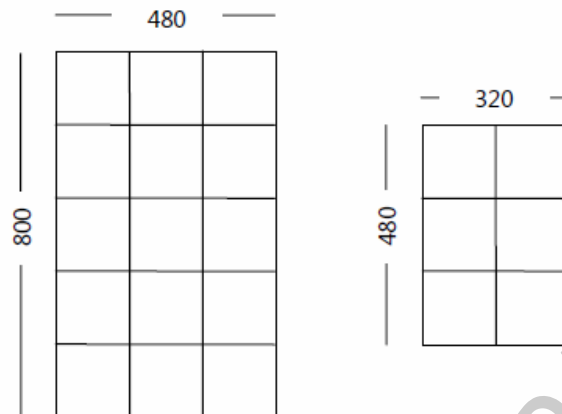
Το εμπρός μέρος της συσκευής αποτελείται από μια οθόνη πολλαπλής αφής (multi-touch) και τριών hardware πλήκτρων (εικόνα 4.1.4-1 ) τα οποία τοποθετούνται σε σειρά κάτω από την οθόνη. Από αριστερά προς δεξιά, τα πλήκτρα αυτά ονομάζονται **Πίσω (Back)** , **Έναρξη (Start)** και **Αναζήτηση (Search)** :



εικόνα 4.1.4-1

- **Back** : Τα προγράμματα μπορούν να χρησιμοποιούν αυτό το κουμπί για τις δικές τους ανάγκες πλοήγησης , όπως το πίσω κουμπί στους web browsers. Από την αρχική οθόνη της εφαρμογής, το πάτημα του κάνει την εφαρμογή να τερματιστεί.
- **Start** : Αυτό το κουμπί μεταφέρει το χρήστη στην αρχική οθόνη του τηλεφώνου.
- **Search** : Το λειτουργικό σύστημα χρησιμοποιεί αυτό το κουμπί για να πραγματοποιεί αναζητήσεις διάφορων χαρακτηριστικών.

Οι αρχικές συσκευές με Windows Phone 7 έχουν οθόνη ανάλυσης 480 X 800 εικονοστοιχεία (pixels) . Ήδη έχει κυκλοφορήσει το καινούριο update με κωδική ονομασία "Tango" το οποίο αναμένεται να υποστηρίξει αναλύσεις 320 X 480 εικονοστοιχείων ( εικόνα 4.1.4-2 ). Με την επόμενη έκδοση των windows phone την 8η θα υποστηρίζονται και High Definition αναλύσεις.



εικόνα 4.1.4-2

Στην παραπάνω εικόνα φαίνεται οι δύο υποστηριζόμενες αναλύσεις προς το παρόν σε portrait διάταξη. Φυσικά τα κινητά μπορούν να περιστραφούν και σε landscape διάταξη. Μερικά προγράμματα μπορεί να απαιτούν να κρατηθεί το κινητό σε μια συγκεκριμένη διάταξη ενώ άλλα είναι προσαρμόσιμα.

Από προεπιλογή οι Silverlight εφαρμογές εμφανίζονται σε portrait διάταξη, αλλά μπορούμε να γράψουμε εφαρμογές οι οποίες να υποστηρίζουν αλλαγές του προσανατολισμού. Σε αντίθεση οι προγραμματιστές παιχνιδιών χρησιμοποιούν συνήθως ένα συγκεκριμένο προσανατολισμό, στις περισσότερες περιπτώσεις landscape.

Όπως πολλά τελευταίας τεχνολογίας τηλέφωνα, τα Windows Phone χρησιμοποιούν οθόνες τύπου OLED τεχνολογίας εκπομπής φωτός μέσω οργανικής διόδου (organic light emitting diode), παρόλο που δεν είναι ένα προαπαιτούμενο του hardware. Οι οθόνες OLED διαφέρουν από τις κοινές επίπεδες οθόνες του παρελθόντος κυρίως στην χαμηλότερη κατανάλωση ενέργειας, το οποίο παίζει σημαντικό ρόλο στην αυτονομία μίας φορητής συσκευής. Για παράδειγμα μια οθόνη τύπου OLED καταναλώνει περίπου την μισή ενέργεια από μια αντίστοιχη τεχνολογίας LCD ίδιου μεγέθους, αλλά μόνο όταν η οθόνη είναι μαύρου φόντου. Για μια όλο άσπρη οθόνη, μια οθόνη OLED καταναλώνει τρεις φορές παραπάνω ενέργεια από μία LCD.

Επειδή η κατανάλωση μπαταρίας είναι εξαιρετικά σημαντική στις φορητές συσκευές, αυτό το χαρακτηριστικό των οθονών τύπου OLED επιβάλλει μια αισθητική με σκουρόχρωμα backgrounds και ανοιχτού χρώματος γραφικά και γραμματοσειρές. Παρόλο αυτά το Windows Phone υποστηρίζει δύο μεγάλα χρωματικά θέματα, ανοιχτού χρώματος κείμενο σε μαύρο background και σκούρου χρώματος κείμενο σε άσπρο background.

Το μεγαλύτερο ποσοστό διεπαφής του χρήστη με προγράμματα των Windows Phone 7 γίνεται μέσω της πολλαπλής αφής οθόνη. Οι οθόνες ενσωματώνουν τεχνολογίας χωρητικότητας-αφής (capacitance-touch technology), το οποίο σημαίνει ότι ανταποκρίνονται σε ένα ανθρώπινο δάχτυλο , αλλά όχι σε ένα stylus ή σε άλλη μορφή πίεσης. Οι οθόνες στα κινητά με Windows Phone 7 απαιτείται να ανταποκρίνονται σε τουλάχιστον τέσσερα συνεχόμενα σημεία αφής.

Το φυσικό πληκτρολόγιο είναι προαιρετικό. Συνήθως συναντάμε το on-screen πληκτρολόγιο , το οποίο θα πρέπει να ανταποκρίνεται στις αλλαγές προσανατολισμού του κινητού , καθώς και να μην αποκρύπτει χρήσιμο περιεχόμενο της εφαρμογής. [8]

## • Αισθητήρες

Ένα κινητό με windows phone λειτουργικό απαιτείται να έχει κάποια ελάχιστα τεχνικά χαρακτηριστικά όσο αφορά τους αισθητήρες και τις παρεχόμενες υπηρεσίες λογισμικού μέσω της βοήθειας του συγκεκριμένου hardware. Παρακάτω αναφέρονται τα σημαντικότερα hardware χαρακτηριστικά και υπηρεσίες από αυτά :

- **wi-fi/3G:** Το τηλέφωνο χρησιμοποιεί το wi-fi για πρόσβαση στο internet ή την τεχνολογία 3G που παρέχεται από την εταιρεία κινητής τηλεφωνίας που χρησιμοποιεί ο χρήστης . Το λογισμικό του τηλεφώνου χρησιμοποιεί μια φορητή έκδοση του Internet Explorer.
- **Κάμερα:** Το τηλέφωνο διαθέτει τουλάχιστον κάμερα 5 megapixels με flash . Τα προγράμματα μπορούν να "καλέσουν" το λογισμικό της κάμερας για εισαγωγή δεδομένων ( όπως φωτογραφίες, βίντεο , barcode κτλ. ) ή να χρησιμοποιηθούν για επεξεργασία των συγκεκριμένων δεδομένων που εισάγονται από την κάμερα.
- **Επιταχυνσιόμετρο:** Το επιταχυνσιόμετρο ανιχνεύει την επιτάχυνση, το οποίο στην φυσική ορίζεται ως η αλλαγή της ταχύτητας. Όταν η κάμερα είναι ακίνητη, το επιταχυνσιόμετρο αντιδρά στην βαρύτητα. Τα προγράμματα μπορούν να ανιχνεύσουν σε χώρο τριών-διαστάσεων τις κινήσεις και μπορούν να "μαντέψουν" πώς είναι τοποθετημένο το τηλέφωνο.
- **Τοποθεσία:** Εφόσον ο χρήστης το επιθυμεί, το τηλέφωνο μπορεί να χρησιμοποιήσει πολλαπλές στρατηγικές για να προσδιορίσει που βρίσκεται γεωγραφικά. Το τηλέφωνο μπορεί να διαθέτει ένα hardware GPS αλλά μπορεί να αντλεί και πληροφορίες από το Διαδίκτυο και τις κεραίες κινητής τηλεφωνίας.

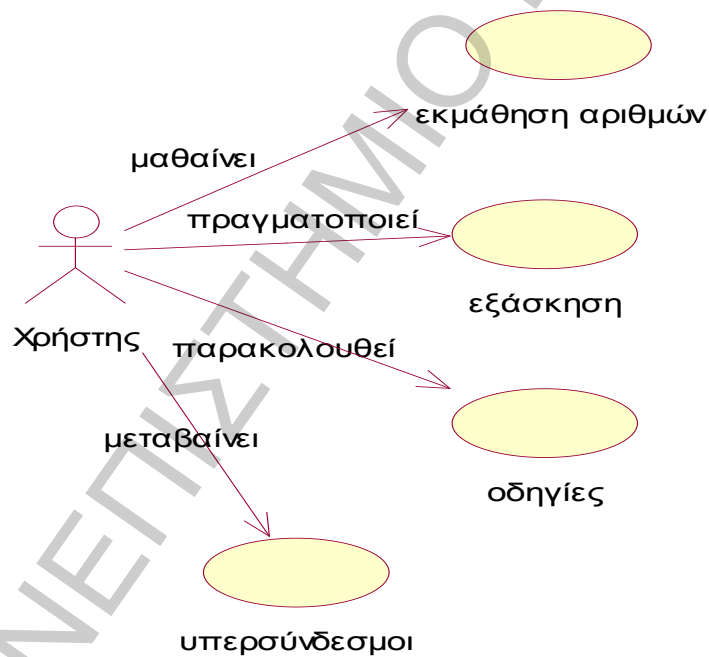
- **Δόνηση:** Το τηλέφωνο μπορεί να δονείται μέσω του προγράμματος ελέγχου.
- **FM Radio:** Ένας δέκτης ραδιοφώνου FM είναι διαθέσιμος και προσβάσιμος από το αντίστοιχο πρόγραμμα ελέγχου.
- **Push Notifications:** Μερικές υπηρεσίες που χρησιμοποιούν το Διαδίκτυο κανονικά απαιτούν από το τηλέφωνο να καταγράφουν ανανεωμένες πληροφορίες. Αυτό μπορεί να μειώσει δραματικά την διάρκεια της μπαταρίας. Για να ξεπεραστεί το συγκεκριμένο πρόβλημα έχει ανατηχθεί μια υπηρεσία η οποία ονομάζεται push notification, η οποία επιτρέπει να πραγματοποιηθεί σύνδεση από το τηλέφωνο και στο τηλέφωνο για λήψη δεδομένων μόνο όταν τα δεδομένα έχουν ανανεωθεί.[8]

## 4.2 Διαγράμματα UML

### 4.2.1 Διαγράμματα περιπτώσεων χρήσης

Τα use case diagrams μας επιτρέπουν να αναπαραστήσουμε ποιες λειτουργίες του συστήματος πραγματοποιούνται και από ποιους χρήστες, καθώς και ποιες εξαρτήσεις υπάρχουν μεταξύ των χρηστών. [6]

Οι δράστες (actors) του συστήματος μπορούν να διακριθούν στην εξής 1 κατηγορία (εικόνα 4.2.1-1) :



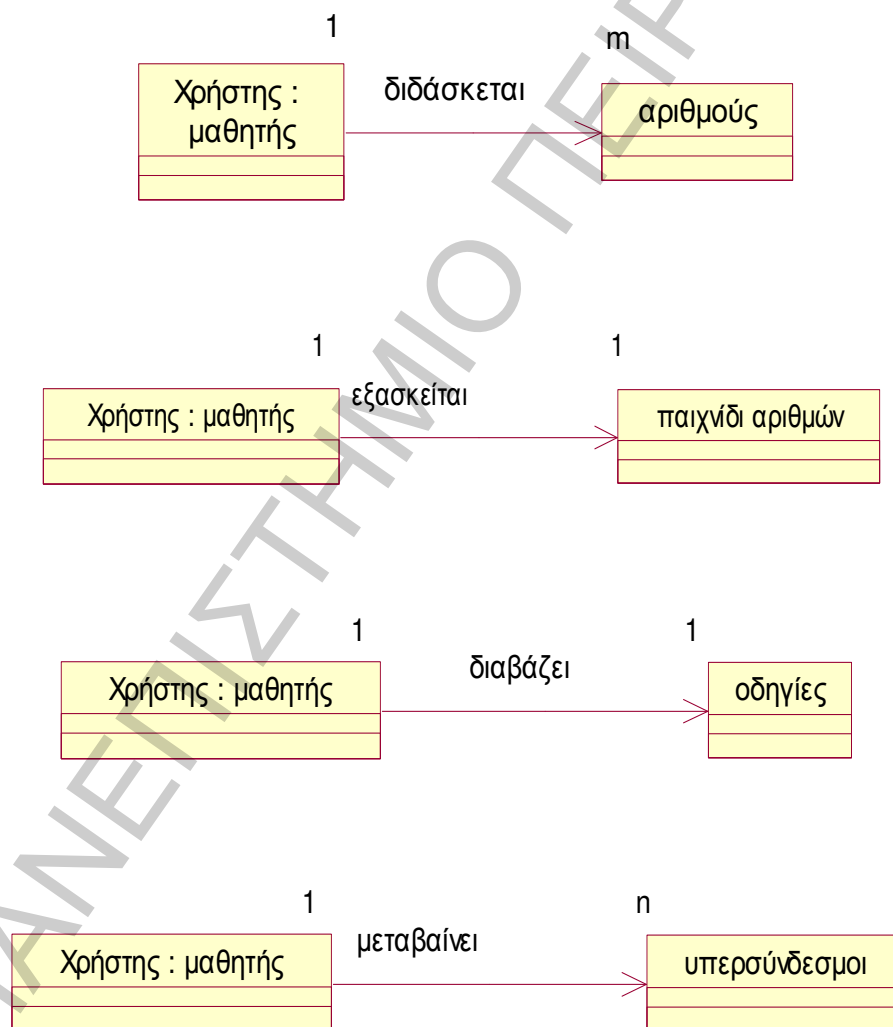
εικόνα 4.2.1-1



## 4.2.2 Διαγράμματα τάξεων

Τα διαγράμματα τάξεων (εικόνα 4.2.2-1) περιγράφουν τις οντότητες που απαρτίζουν ένα σύστημα και τις στατικές συσχετίσεις μεταξύ τους. Αποτελούνται από:

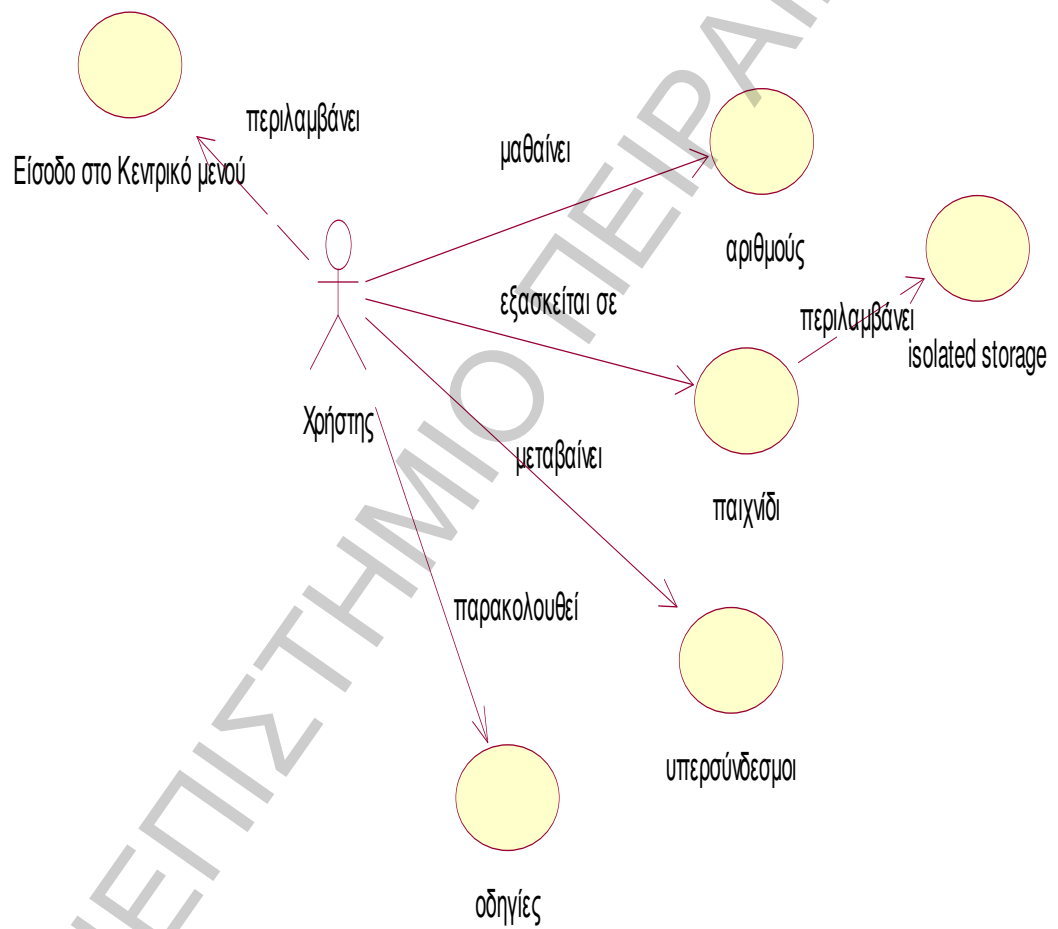
- Κλάσεις (*classes*)
- Σχέσεις (*relationships*)



εικόνα 4.2.2-1

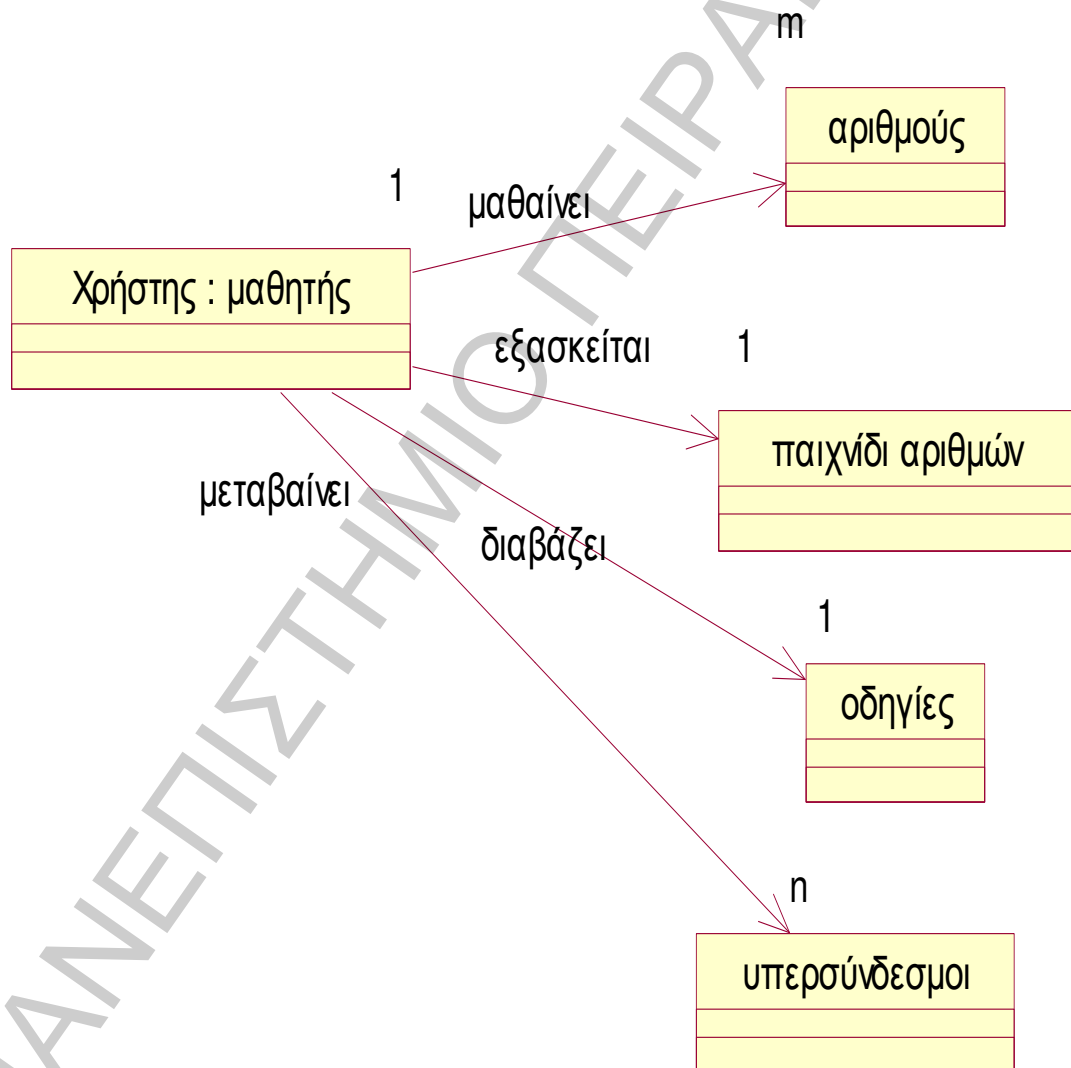
### 4.2.3 Ανάλυση – Σχεδιασμός

Σε ένα συγκεντρωτικό use case (εικόνα 4.2.3-1 ) [6]



εικόνα 4.2.3-1

Σε ένα συγκεντρωτικό class (εικόνα 4.2.3-2):

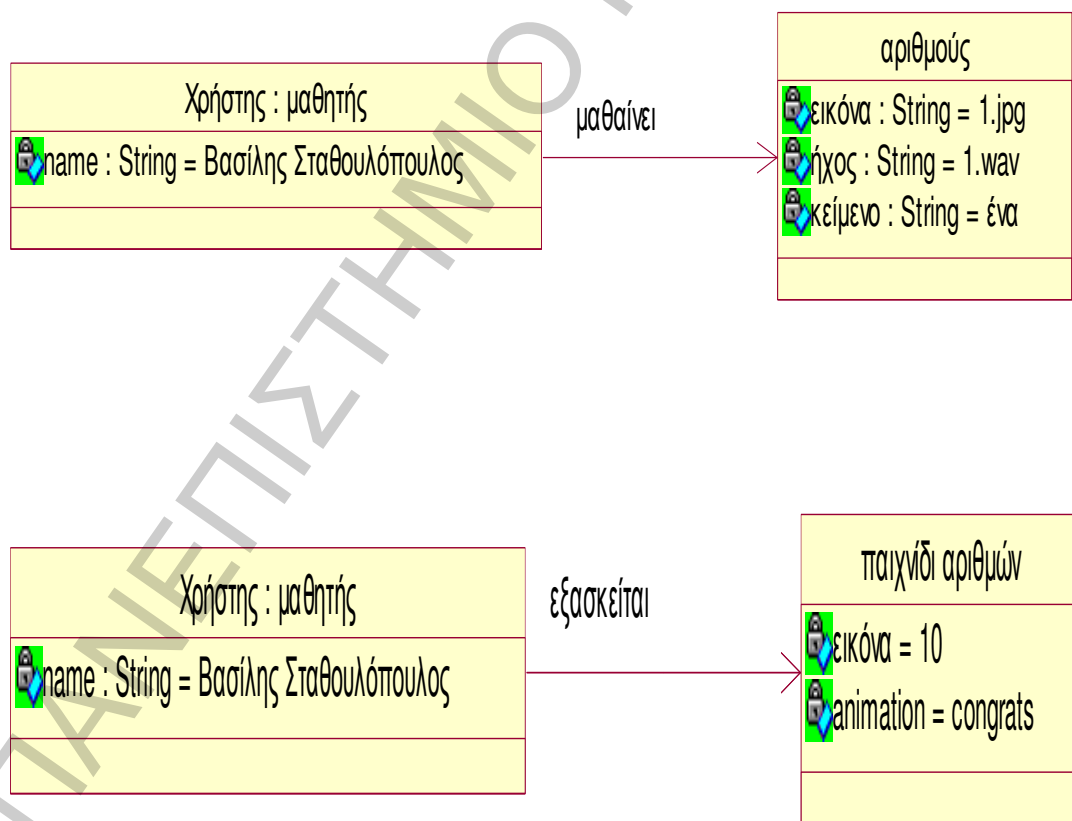


εικόνα 4.2.3-2

## 4.2.4 Φάση Εκπόνησης μελέτης

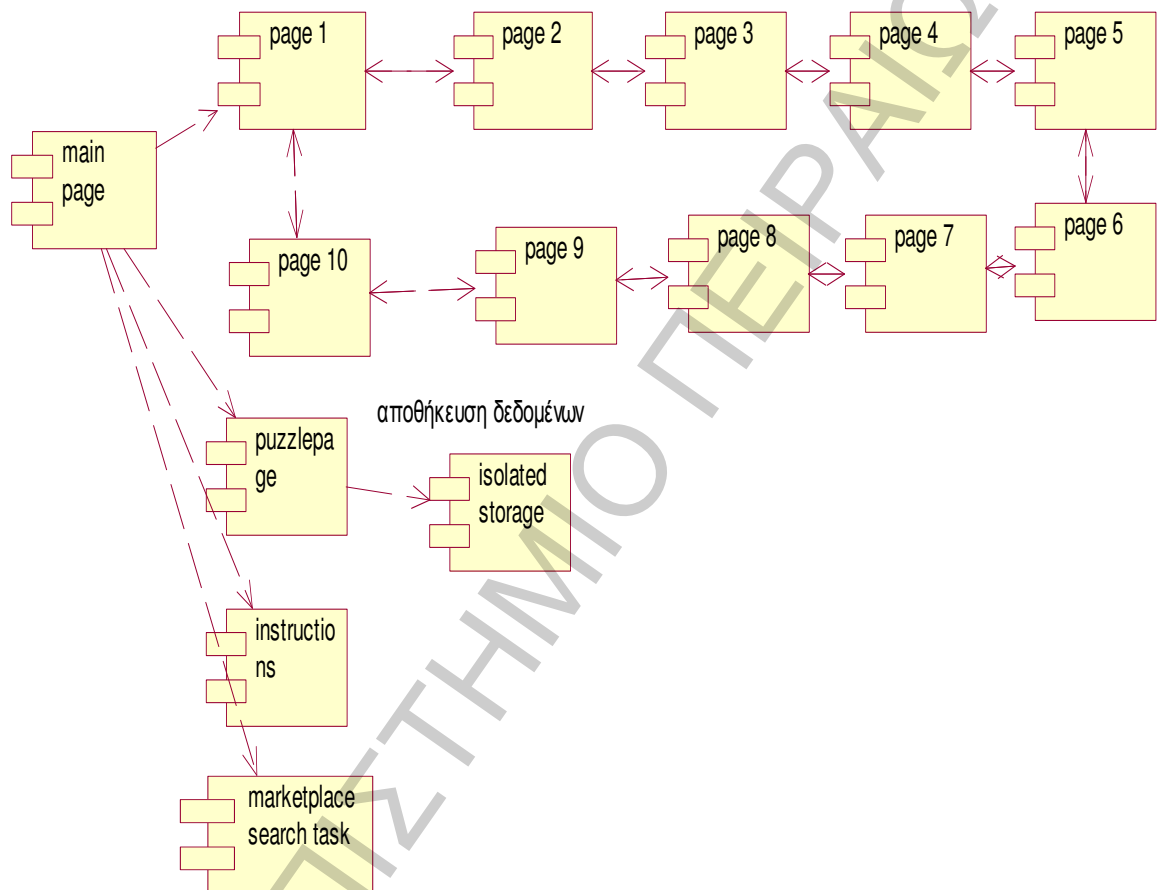
- *Διαγράμματα Αντικειμένων (object diagram) :*

Το διάγραμμα αντικειμένων (εικόνα 4.2.4-1 ) εξειδικεύει ένα στιγμιότυπο του διαγράμματος κλάσης, τη στιγμή που εκτελείται στον πραγματικό κόσμο. Για παράδειγμα, η κλάση δάσκαλος καθώς δημιουργεί και επεξεργάζεται ασκήσεις μπορεί να περιγραφεί ως εξής :[6]



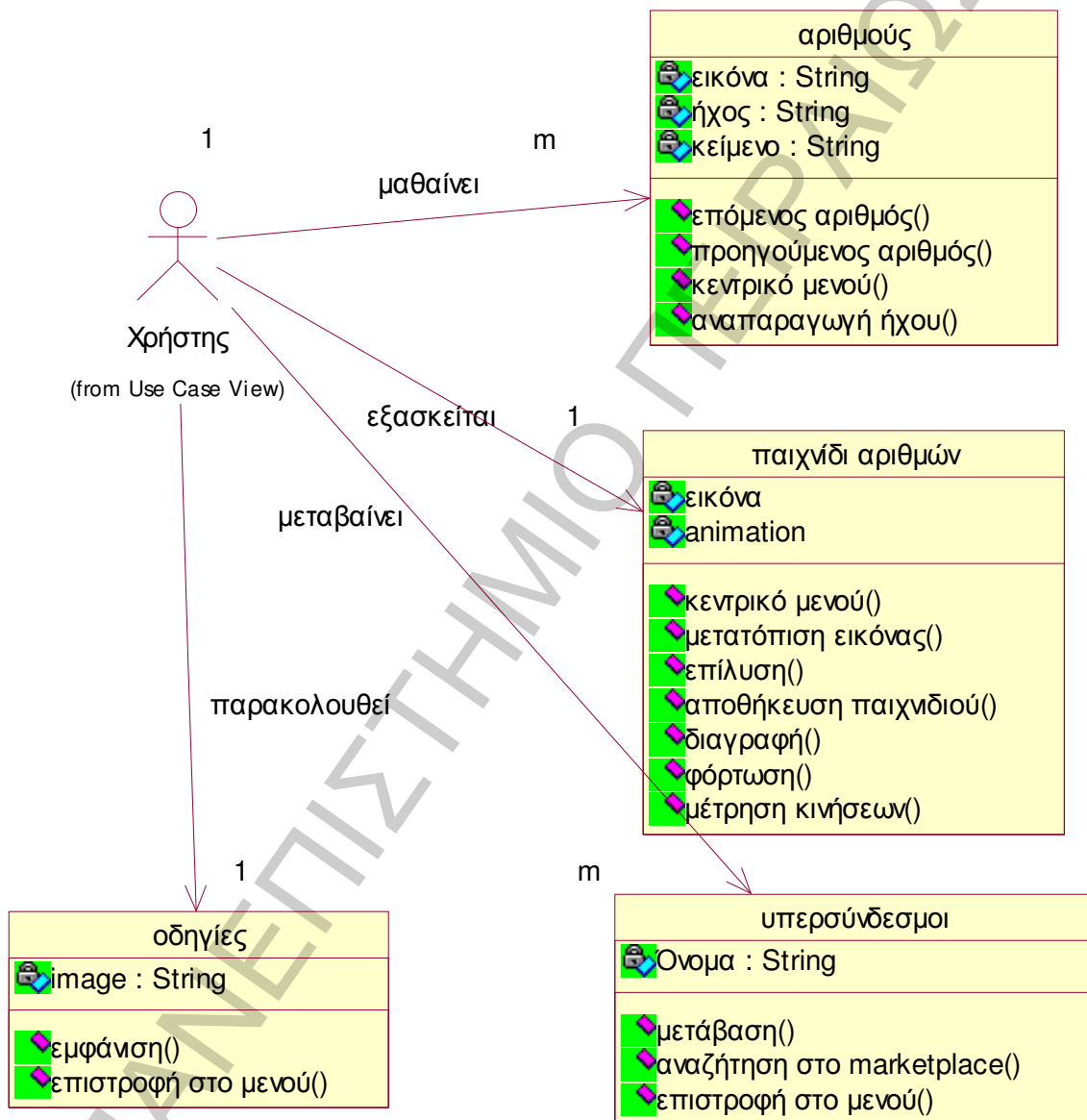
εικόνα 4.2.4-1

- **Διαγράμματα συστατικών** (εικόνα 4.2.4-2 )



εικόνα 4.2.4-2

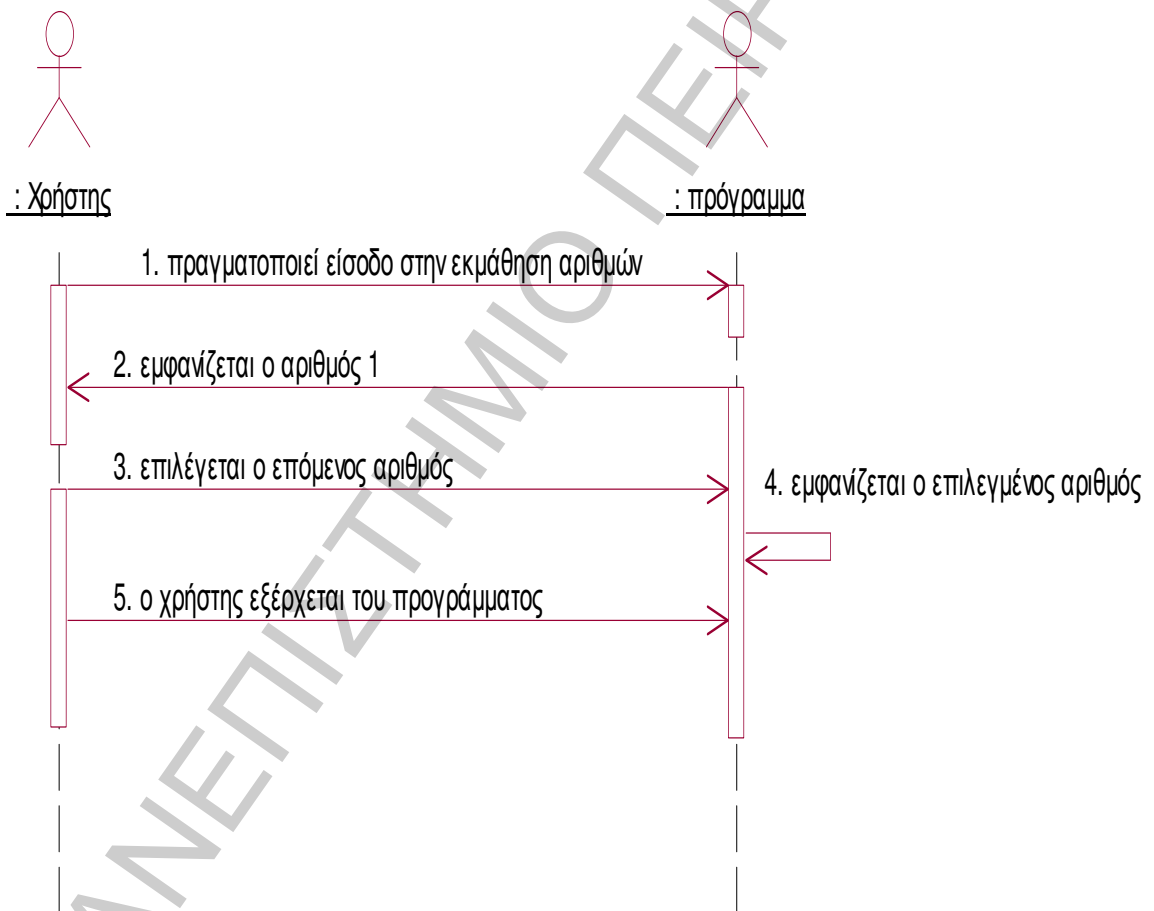
Διάγραμμα τάξεων αναλυτικό ( εικόνα 4.2.4-3 )



εικόνα 4.2.4-3

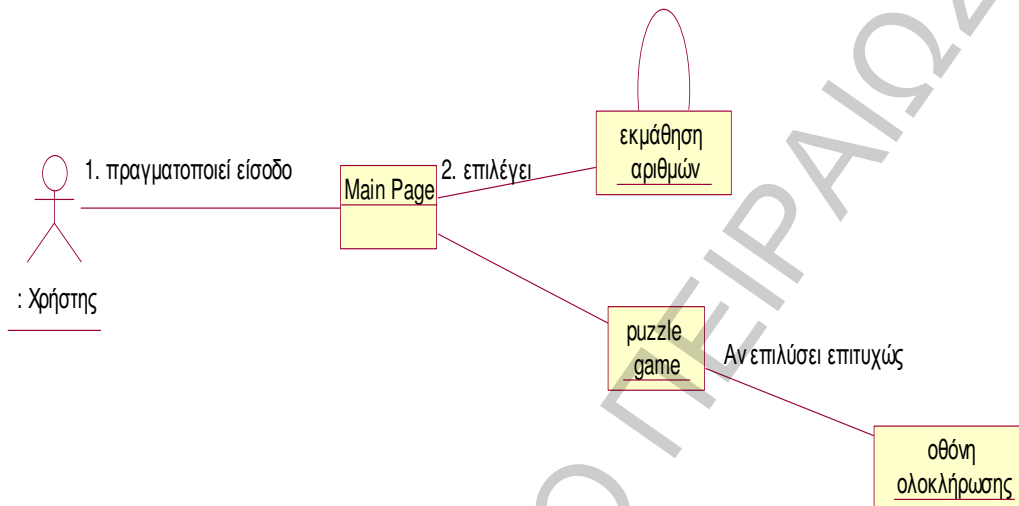
## Μοντελοποίηση δυναμικών χαρακτηριστικών του συστήματος :

Διάγραμμα ακολουθίας ( εικόνα 4.2.4-4 ) :



εικόνα 4.2.4-4

Διάγραμμα συνεργασίας ( εικόνα 4.2.4-5 ) :



εικόνα 4.2.4-5

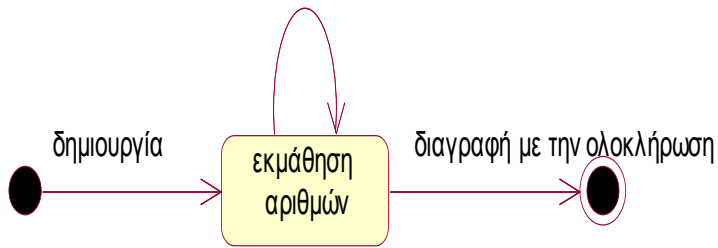
## Μοντελοποίηση συμπεριφοράς συγκεκριμένων αντικειμένων

Διαγράμματα κατάστασης ( εικόνα 4.2.4-6 )

Στην UML τα διαγράμματα κατάστασης χρησιμοποιούνται κυρίως για την μοντελοποίηση λειτουργιών που καθοδηγούνται από συμβάντα (αναδραστικές λειτουργίες). Ουσιαστικά τα διαγράμματα κατάστασης περιγράφουν όλες τις πιθανές **καταστάσεις (states)** στις οποίες μπορεί να βρεθούν τα αντικείμενα μιας τάξης και τον τρόπο που αλλάζει η κατάσταση αυτών των αντικειμένων σαν ανάδραση σε συμβάντα που προκαλούνται από άλλα αντικείμενα και φτάνουν στα αντικείμενα της τάξης την δυναμική συμπεριφορά της οποίας και περιγράφουμε.



Όσον αφορά το αντικείμενο εκμάθηση αριθμών :



εικόνα 4.2.4-6

Όσον αφορά το αντικείμενο puzzle game :

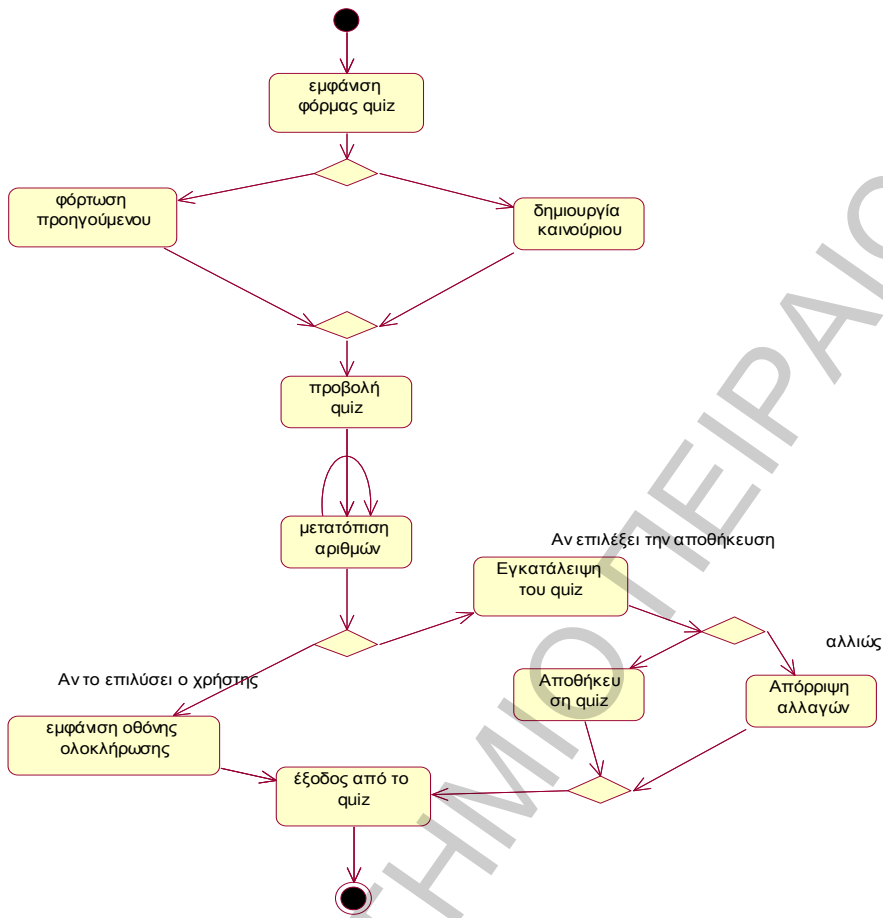


εικόνα 4.2.4-6

### **Μοντελοποίηση βημάτων εκτέλεσης μιας διαδικασίας**

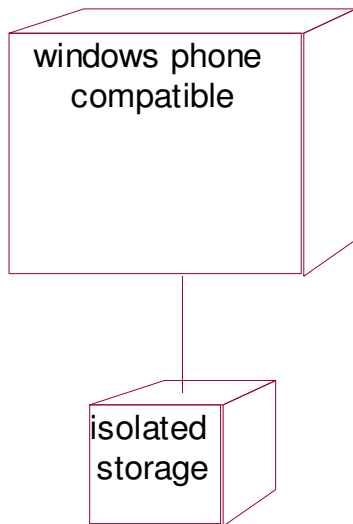
Διαγράμματα δραστηριοτήτων (εικόνα 4.2.4-7)

Για τη διαδικασία της επίλυσης του quiz :



εικόνα 4.2.4-7

Τέλος, τα παραταξιακά διαγράμματα ή αλλιώς διαγράμματα διανομής (εικόνα 4.2.4-8) αντικατοπτρίζουν μια στατική όψη της εν ενεργεία ρύθμισης των κόμβων και όλων των συστατικών που τρέχουν σε αυτούς τους κόμβους. Τα διαγράμματα διανομής δηλαδή αναπαριστούν το hardware που τρέχει στο σύστημα, και τι software είναι εγκατεστημένο σε αυτό. [6]



λογισμικό : windows phone 7+

εικόνα 4.2.4-8

## 4.3 Ανάπτυξη Εφαρμογής

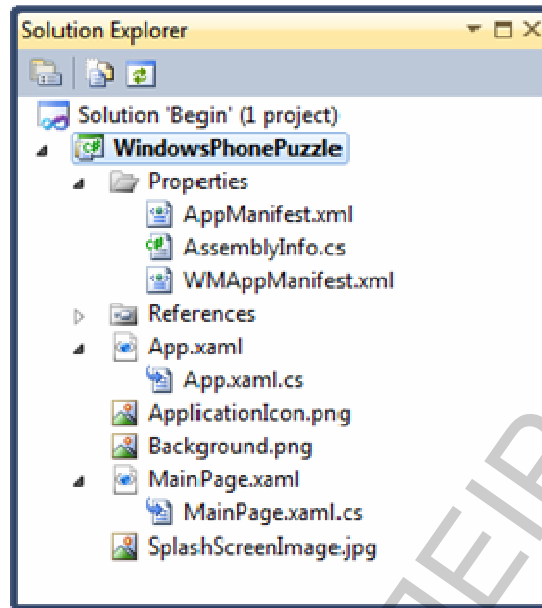
### 4.3.1 Γενικά

Για την δημιουργία της εφαρμογής έχει χρησιμοποιηθεί το **Visual Studio 2010 ultimate** σε συνδυασμό με το **Windows Phone SDK** καθώς και το **Windows Phone Emulator** . Η Microsoft παρέχει και μια έκδοση των παραπάνω την **Express** η οποία είναι μη χρεώσιμη και παρέχει σχεδόν όλα τα απαραίτητα εργαλεία για την ανάπτυξη εφαρμογών και παιχνιδιών.

Παρακάτω παρουσιάζονται αναλυτικά τα βήματα για την ανάπτυξη της εκπαιδευτικής εφαρμογής "**Αριθμάκια Παζλάκια**" τόσο στο θέμα της σχεδίασης του user interface όσο και του κώδικα.

Αρχικά ανοίγουμε το Visual Studio . Στο μενού **File** , επιλέγουμε **New Project** . Θα εμφανιστεί ένα νέο με όλες τις επιλογές που δίνονται αναλόγως του τι είδους προγράμματος θέλουμε να φτιάξουμε για τα Windows Phone . Στην προκειμένη περίπτωση επιλέγουμε **Silverlight for Windows Phone**. Ορίζουμε το όνομα της εφαρμογής "**WindowsPhonePuzzle**" και επιλέγουμε το φάκελο στον οποίο θέλουμε να γίνει η αποθήκευση. Αλλάζουμε το όνομα του solution σε **Begin** και επιλέγουμε **OK**.

Στον **Solution Explorer** (εικόνα 4.3.1-1) , μπορούμε να δούμε την δομή που δημιουργείται από των Windows Phone Application. Το κάθε solution του Visual Studio είναι ένα δοχείο "για συσχετιζόμενα project" στην δική μας περίπτωση , περιέχει μόνο ένα project με όνομα **WindowsPhonePuzzle**.



Εικόνα 4.3.1-1

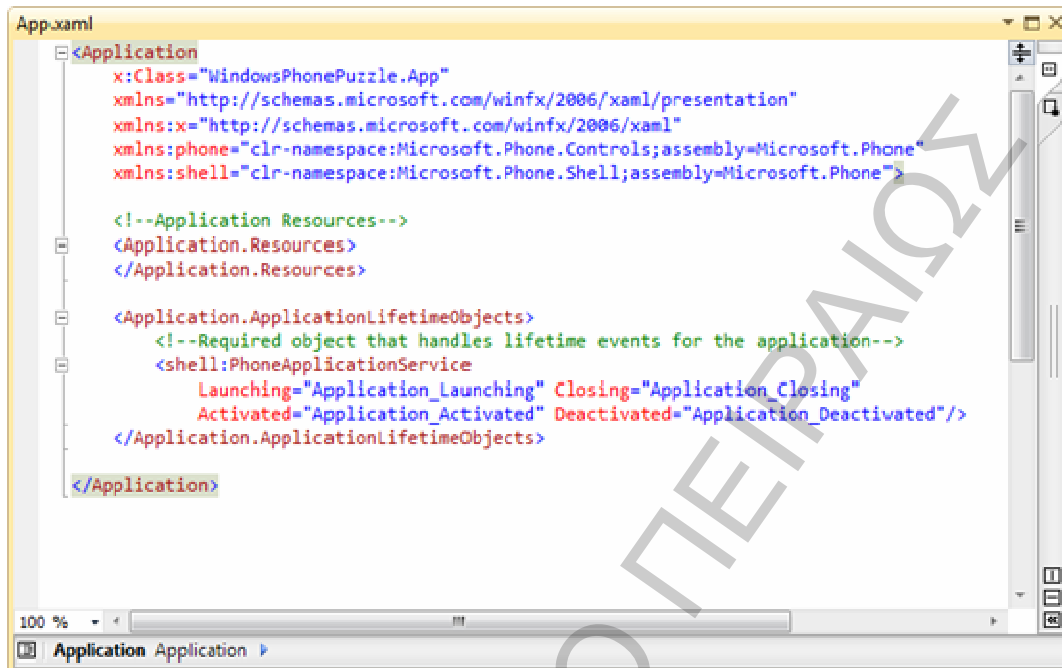
Το **WindowsPhonePuzzle** project περιέχει τα παρακάτω αντικείμενα :

<u>Αντικείμενα</u>	<u>Περιγραφή</u>
App.xaml / App.xaml.cs	Ορίζει το εισαγωγικό σημείο της εφαρμογής, αρχικοποιεί τους πόρους της εφαρμογής και εμφανίζει το περιβάλλον εργασίας χρήστη της εφαρμογής.
MainPage.xaml / MainPage.xaml.cs	Ορίζει μια σελίδα με το περιβάλλον χρήσης της εφαρμογής (μπορούμε να έχουμε πολλές σελίδες).
ApplicationIcon.png	Ένα αρχείο εικόνας το οποίο περιέχει την εικόνα η οποία αντιπροσωπεύει την εφαρμογή στην λίστα εφαρμογών.
Background.png	Ένα αρχείο εικόνας το οποίο περιέχει την εικόνα που εμφανίζεται στην αρχική οθόνη.
SplashScreenImage.jpg	Αυτή είναι η πρώτη εικόνα η οποία θα εμφανιστεί όταν η εφαρμογή φορτώνει. Η εικόνα δίνει στο χρήστη άμεση ενημέρωση ότι η εφαρμογή φορτώνει και θα παραμείνει εμφανή μέχρι να ολοκληρωθεί η πλοήγηση στην πρώτη σελίδα. Μπορούμε να βάλουμε η συγκεκριμένη εικόνα να

	είναι ίδια με την πρώτη σελίδα για να δώσουμε στο χρήστη την εντύπωση ότι η εφαρμογή φορτώνει γρήγορα.
Properties/AppManifest.xml	Το αρχείο δήλωσης της εφαρμογής που απαιτείται για να "παραχθεί" η εφαρμογή.
Properties/AssemblyInfo.cs	Περιέχει το όνομα και την έκδοση των μεταδεδομένων τα οποία εμπεριέχονται στη παραγόμενη συνδεσμολογία.
Properties/WMApManifest.xml	Αρχείο δήλωσης το οποίο περιέχει συγκεκριμένα μεταδεδομένα που σχετίζονται με την Windows Phone Silverlight εφαρμογή, περιλαμβάνει συγκεκριμένα χαρακτηριστικά τα οποία είναι διαθέσιμα μόνο στο Silverlight για WP.
References folder	Λίστα βιβλιοθηκών οι οποίες παρέχουν υπηρεσίες και λειτουργίες , οι οποίες απαιτούνται για να λειτουργήσει η εφαρμογή.

Αρχικά θα αναφερθούμε λίγο στο αντικείμενο **App.xaml** κάνουμε δεξί κλικ στο **solution explorer** και επιλέγουμε **View designer** . Παρατηρούμε ότι το συγκεκριμένο περιέχει κώδικα XAML με ένα ριζικό στοιχείο **Application** και μέσα σε αυτό ένα πεδίο το **Application.Resources**. Εκεί μπορούμε να ορίσουμε τους πόρους της εφαρμογής όπως τα χρώματα , βούρτσες και οπτικά αντικείμενα που χρησιμοποιούνται από την εφαρμογή .

Ο κώδικας XAML (εικόνα 4.3.1-2) επίσης αρχικοποιεί και την ιδιότητα **ApplicationLifetimeObjects** της εφαρμογής για να δημιουργήσει ένα αντικείμενο **PhoneApplicationService**. Η κλάση **PhoneApplicationService** παρέχει πρόσβαση σε διάφορες πτυχές του κύκλου ζωής της εφαρμογής. Αυτό περιλαμβάνει διαχείριση της εφαρμογής σε κατάσταση ηρεμίας (idle) καθώς και όταν ενεργοποιείται (active) ή όταν απενεργοποιείται (inactive) .[1]



Εικόνα 4.3.1-2

Το αρχείο **App.xaml** , μαζί με το αρχείο **App.xaml.cs** ορίζουν ένα instance της κλάσης Application. Αυτό συμπυκνώνει μια εφαρμογή τύπου Silverlight for Windows Phone και ορίζει το σημείο έναρξής της.

Για να δούμε το αρχείο **App.xaml.cs** (εικόνα 4.3.1-3) κάνουμε δεξί κλικ πάνω του και επιλέγουμε **View Code**. Η ιδιότητα Root Frame στην κλάση application αναγνωρίζει την σελίδα έναρξης της εφαρμογής. Όλες οι εφαρμογές για windows phone έχουν ένα στοιχείο του οποίο ο τύπος δεδομένων είναι **PhoneApplicationFrame**. Αυτό το πλαίσιο (**Frame**) περιλαμβάνει ένα ή περισσότερα **PhoneApplicationPage** στοιχεία , τα οποία παρουσιάζουν το περιεχόμενο της εφαρμογής. Επίσης διαχειρίζεται και τις μεταβάσεις μεταξύ των σελίδων.

```
namespace WindowsPhonePuzzle
{
    public partial class App : Application
    {
        // Easy access to the root frame
        public PhoneApplicationFrame RootFrame { get; private set; }

        // Constructor
        public App(...)

        // Code to execute when the application is launching (eg, from Start) ...
        private void Application_Launching(object sender, LaunchingEventArgs e)
        {
        }

        // Code to execute when the application is activated (brought to foreground)
        // This code will not execute when the application is first launched
        private void Application_Activated(object sender, ActivatedEventArgs e)
        {
        }

        // Code to execute when the application is deactivated (sent to background)
        // This code will not execute when the application is closing
        private void Application_Deactivated(object sender, DeactivatedEventArgs e)
        {
        }

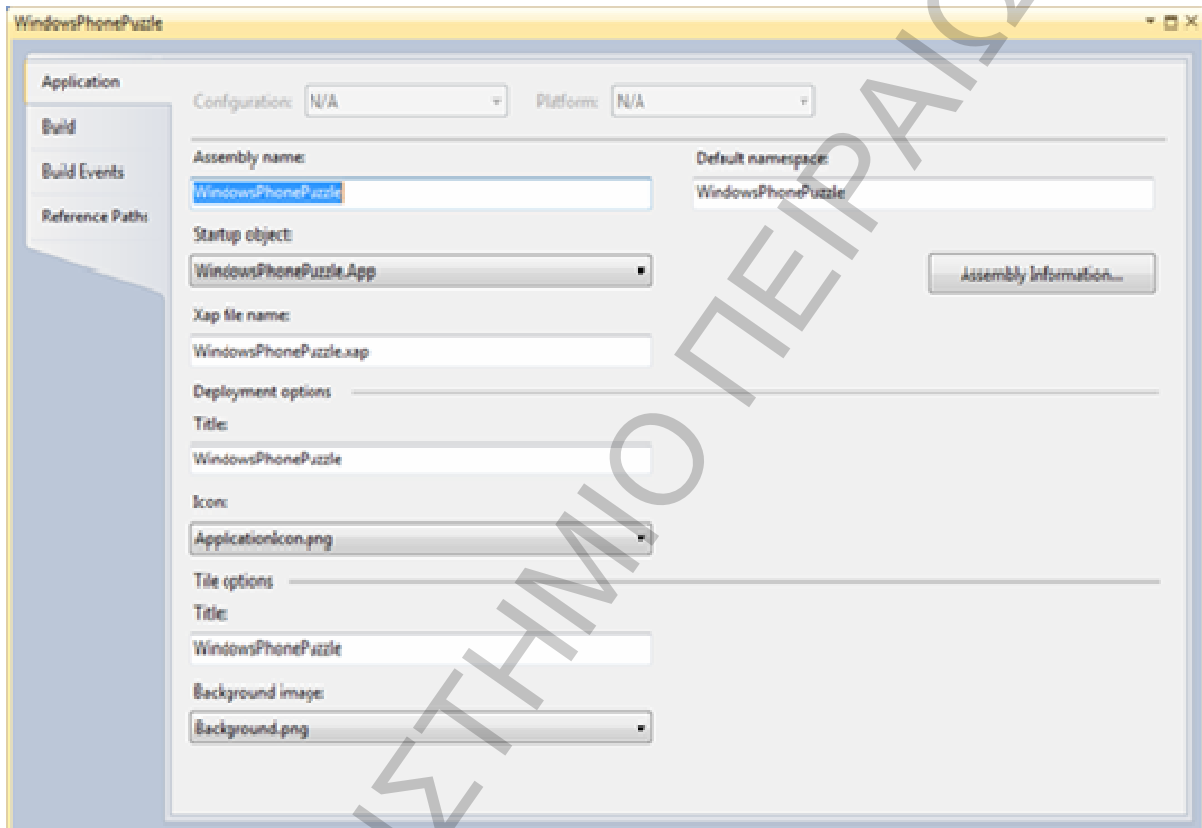
        // Code to execute when the application is closing (eg, user hit Back)
        // This code will not execute when the application is deactivated
        private void Application_Closing(object sender, ClosingEventArgs e)
        {
        }
    }
}
```

Εικόνα 4.3.1-3

Μια εφαρμογή για windows phone εκμεταλλεύεται υπηρεσίες από άλλες πλατφόρμες καθώς και από βιβλιοθήκες. Για να χρησιμοποιήσουμε αυτή την λειτουργία , η εφαρμογή θα χρειαστεί να κάνει παραπομπή στα κατάλληλα **assemblies** τα οποία περιέχουν αυτές τις παροχές. Για να προβάσουμε τα assemblies που χρησιμοποιούνται από το project μας , απλώς επεκτείνουμε τον κόμβο **References** στον **Solution Explorer** και βλέπουμε από την λίστα που εμφανίζεται. Εξ αρχής περιέχει τυπικά Silverlight assemblies καθώς και assemblies σχετικά με την Windows Phone platform. Υπάρχει φυσικά δυνατότητα να προσθέσουμε και επιπλέον κάτι το οποίο θα το δείξουμε στην συνέχεια καθώς θα μας χρειαστεί και στο project που αναπτύσσουμε.

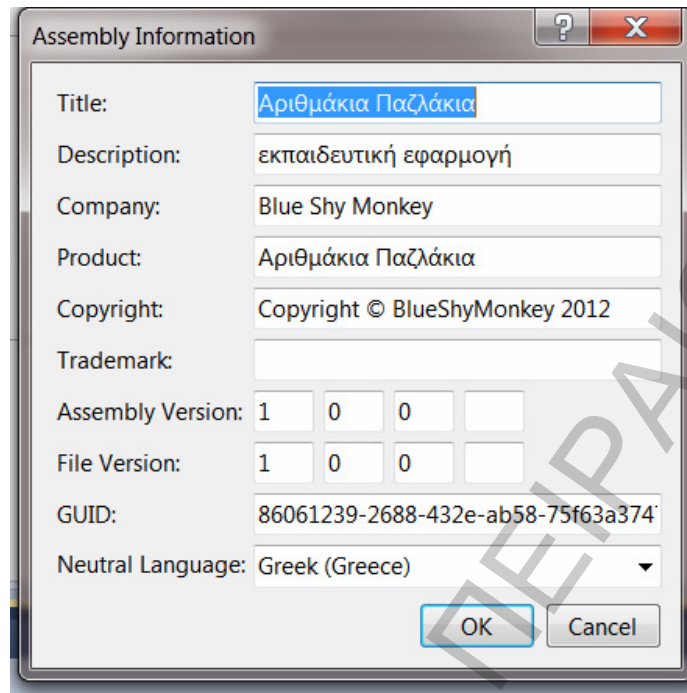


Το παράθυρο **Properties του project** είναι ο μόνος τρόπος για να επεξεργαστούμε το αρχείο **WP manifest**, όπως φαίνεται στην Εικόνα 4.3.1-4 . Για να ανοίξουμε το παράθυρο κάνουμε δεξί κλικ στο **WindowsPhonePuzzle** project στο πεδίο του **Solution Explorer** και επιλέγουμε **Properties**.



Εικόνα 4.3.1-4

Πατώντας στο πλήκτρο **Assembly Information** εμφανίζεται ένα νέο παράθυρο όπου μπορούμε να δηλώσουμε κάποια στοιχεία της εφαρμογής όπως το όνομα , ο εκδότης , ο τύπος της εφαρμογής κτλ. ( εικόνα 4.3.1-5 ) .Μεγάλη προσοχή χρειάζεται στο πεδίο `neutral language` όπου δηλώνεται και η κύρια γλώσσα της εφαρμογής στην προκείμενη περίπτωση τα ελληνικά μπορεί να αλλαχθεί ανά πάσα στιγμή, αλλά καλό θα είναι να το ορίζουμε από την αρχή για να μην το ξεχάσουμε και ανεβάσουμε την εφαρμογή με λάθος γλώσσα στο Marketplace , το οποίο θα έχει ως αποτέλεσμα να μην πάρει την απαραίτητη έγκριση (θα αναφερθούμε εκτενέστερα στο Marketplace και με την διαδικασία ανεβάσματος της εφαρμογής μας ).

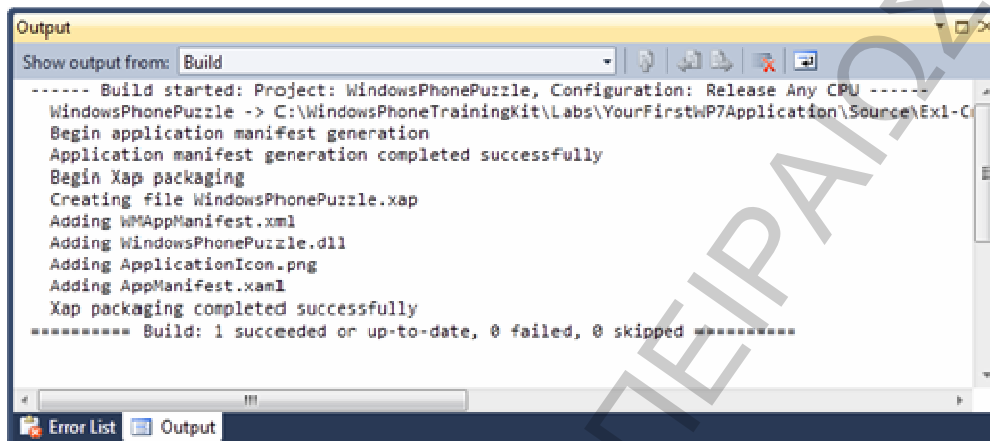


Εικόνα 4.3.1-5

### 4.3.2 Δοκιμή της εφαρμογής στον Emulator

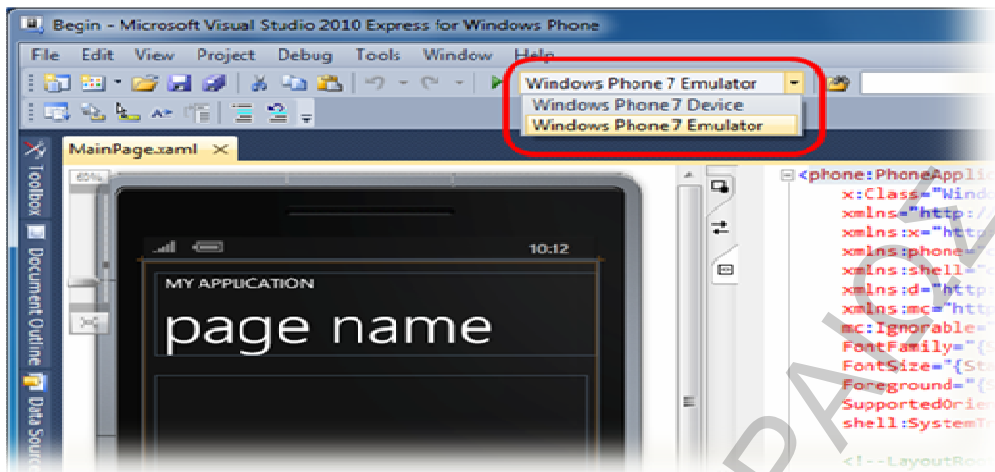
Μέχρι στιγμής η εφαρμογή μας δεν κάνει κάτι το ιδιαίτερο, αλλά είναι έτοιμη για την πρώτη δοκιμαστική της λειτουργία. Πριν δημιουργήσουμε την εφαρμογή θα κάνουμε build την εφαρμογή στην τρέχουσα κατάσταση της, θα την τρέξουμε στον emulator των Windows Phone και θα την εκτελέσουμε για να κατανοήσουμε τον κύκλο ανάπτυξης της.

1. Από το μενού **Debug**, εστιάζουμε στην επιλογή **Windows** και επιλέγουμε **Output**, για να ανοίξει το παράθυρο **Output** (εικόνα 4.3.2-1).
2. Επιλέγουμε **build solution** από το μενού **build**.
3. Παρατηρούμε ότι στο παράθυρο **Output** παράγονται διάφορα μηνύματα κατά το "χτίσιμο" της εφαρμογής συμπεριλαμβανομένου και του τελικού μηνύματος με το αποτέλεσμα.



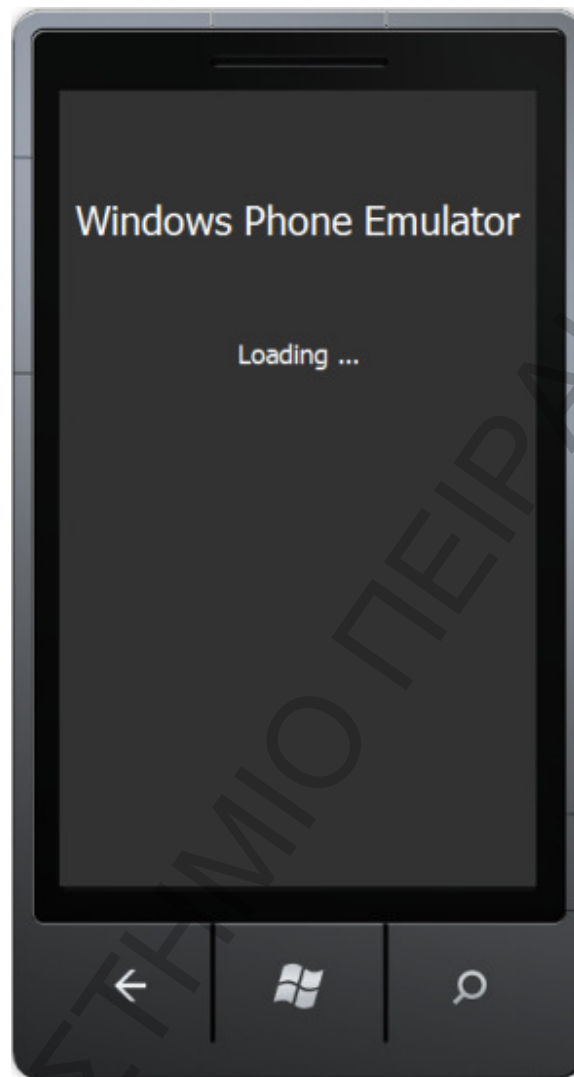
Εικόνα 4.3.2-1

4. Θα πρέπει να παρατηρήσετε ότι δεν υπάρχουν λάθη σε αυτή την φάση , αν υπήρχαν πάντως θα εμφανίζονταν σε αυτό το παράθυρο . Για να αντιμετωπίσουμε τυχόν λάθη που μπορεί να προκύψουν μπορούμε να τα διαχειριστούμε από το παράθυρο **Error List**.
5. Επόμενο βήμα είναι να τρέξουμε την εφαρμογή στο Windows Phone Emulator. Για να το κάνουμε αυτό επιλέγουμε στο πεδίο **Select Target** το οποίο βρίσκεται δίπλα στο πλήκτρο **Start Debugging** από το drop down menu που θα εμφανιστεί Windows Phone 7 Emulator (εικόνα 4.3.2-2).



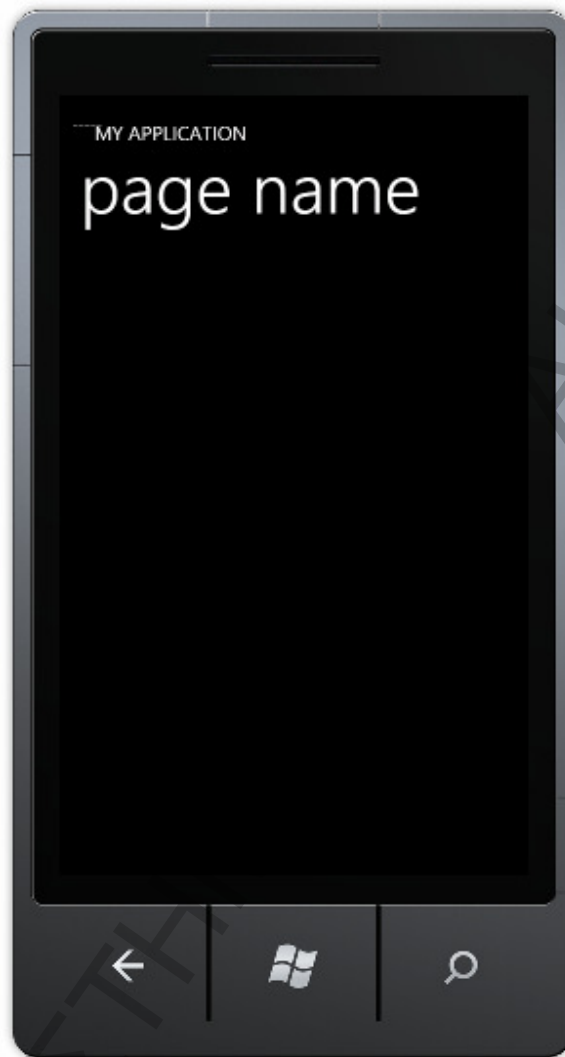
Εικόνα 4.3.2-2

6. Πατάμε το F5 για να τρέξουμε την εφαρμογή στο Windows Phone Emulator . Παρατηρούμε ότι εμφανίζεται ένα παράθυρο εξομοιωτή συσκευής και παρουσιάζεται μία μικρή παύση μέχρι το Visual Studio να εξομοιώσει την εφαρμογή (εικόνα 4.3.2-3).



Εικόνα 4.3.2-3

7. Όταν είναι έτοιμο ο εξομοιωτής δείχνει την αρχική εικόνα και μετά από λίγο η εφαρμογή εμφανίζεται στο παράθυρο του εξομοιωτή (εικόνα 4.3.2-4).



Εικόνα 4.3.2-4

8. Για να σταματήσουμε το τρέξιμο του εξομοιωτή απλώς πατάμε Shift + F5 ή το πλήκτρο stop από την μπάρα εργαλείων . Για να γλυτώσουμε χρόνο καθώς για να εκκινήσει ο εξομοιωτής θέλει κάποια ώρα καλό είναι να μην τον "τερματίζουμε" από το πλήκτρο "X" αλλά απλώς να τον σταματάμε με την παραπάνω διαδικασία .

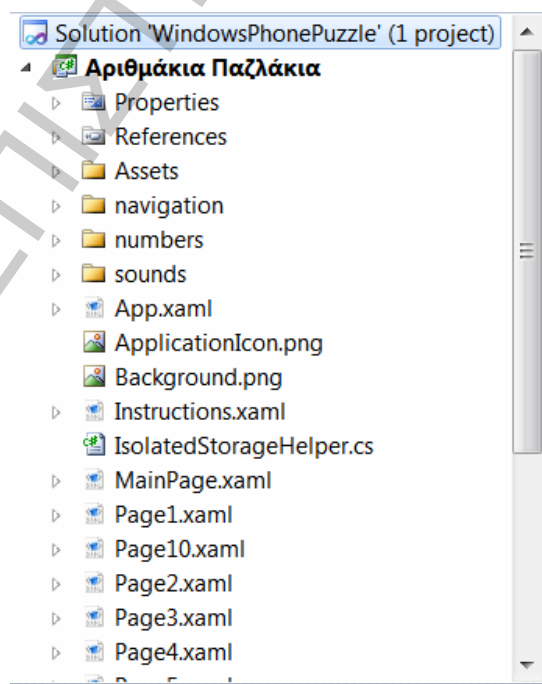
Έχοντας αναφέρει τα σημαντικότερα στοιχεία των εφαρμογών που τρέχουν σε Windows Phone και για την χρήση του Visual Studio για το πώς μπορούμε να τρέξουμε μια

εφαρμογή , στην επόμενη ενότητα θα δούμε τα βασικότερα βήματα για την ανάπτυξη της εφαρμογής μας . [1]

### 4.3.3 Δημιουργώντας την "Κύρια Σελίδα"

Καταρχήν πριν ξεκινήσουμε την δημιουργία της πρώτης σελίδας θα προσθέσουμε τέσσερις φάκελους στο project μας και συγκεκριμένα τους **Assets** , **Navigation** , **Numbers** , **Sounds** στους οποίους θα περιέχονται τα multimedia αρχεία (εικόνα 4.3.3-1) που θα χρησιμοποιήσει η εφαρμογή μας και συγκεκριμένα :

- Assets** : Περιλαμβάνει όλα τα αρχεία εικόνων που θα χρησιμοποιηθούν για background, και διάφορες εντολές όπως αποθήκευση κτλ.
- Navigation** : Περιλαμβάνει τις εικόνες οι οποίες θα χρησιμοποιηθούν για την περιήγηση μεταξύ των σελίδων.
- Numbers** : Περιλαμβάνει τις εικόνες οι οποίες χρησιμοποιούνται για την προβολή των αριθμών.
- Sounds** : Περιλαμβάνει όλα τα ηχητικά αρχεία.



Εικόνα 4.3.3-1

Για να προσθέσουμε τους παραπάνω φακέλους κάνουμε δεξί κλικ πάνω στο όνομα του project μας στο **Solution Explorer** και από τις επιλογές που εμφανίζονται πηγαίνουμε στην **Add** στην συνέχεια αν θέλουμε να προσθέσουμε φάκελο επιλέγουμε το **New Folder**, αλλιώς για κάποια εικόνα ή ήχο το **existing item**.

Η "**Κύρια Σελίδα**" ( **Main Page** ) της εφαρμογής μας αποτελείται βασικά από τέσσερα **κουμπιά (buttons)** συγκεκριμένα τα "**Μαθήματα**" , "**Παιχνίδι**" , "**Οδηγίες**" και "**BlueShyMonkey Apps**" τα οποία θα έχουν τις ακόλουθες λειτουργίες :

<u>Μαθήματα</u> :	Μας μεταφέρει στην πρώτη σελίδα όπου ξεκινάνε οι αριθμοί και συγκεκριμένα του αριθμού ένα.
<u>Παιχνίδι</u> :	Μας μεταφέρει στην σελίδα του εκπαιδευτικού παιχνιδιού.
<u>Οδηγίες</u> :	Μας μεταφέρει στην σελίδα με τις οδηγίες χρήσης της εφαρμογής μας.
<u>BlueShyMonkey Apps</u> :	Μας μεταφέρει στην ενότητα του Marketplace όπου είναι συγκεντρωμένες όλες οι εφαρμογές που έχουμε ανεβάσει.

Στην παρακάτω εικόνα εμφανίζεται η "**Κύρια Σελίδα**" (Main Page.xaml εικόνα 4.3.3-2) της εφαρμογής μας



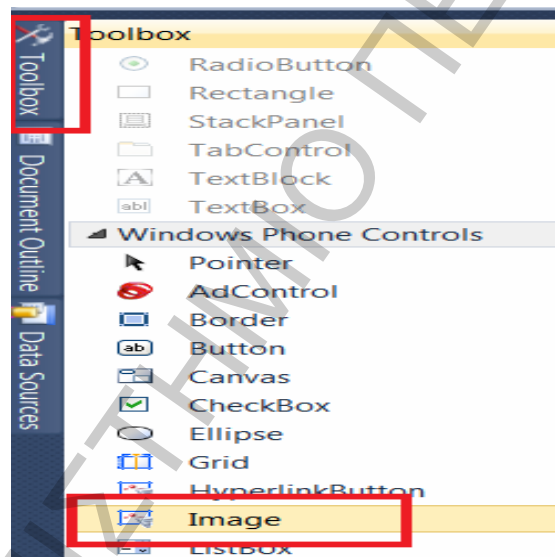


Εικόνα 4.3.3-2

Όπως είπαμε η εφαρμογή μας αποτελείται από τέσσερα **κουμπιά (buttons)** και μία **εικόνα (image)** η οποία καλύπτει όλο το background .

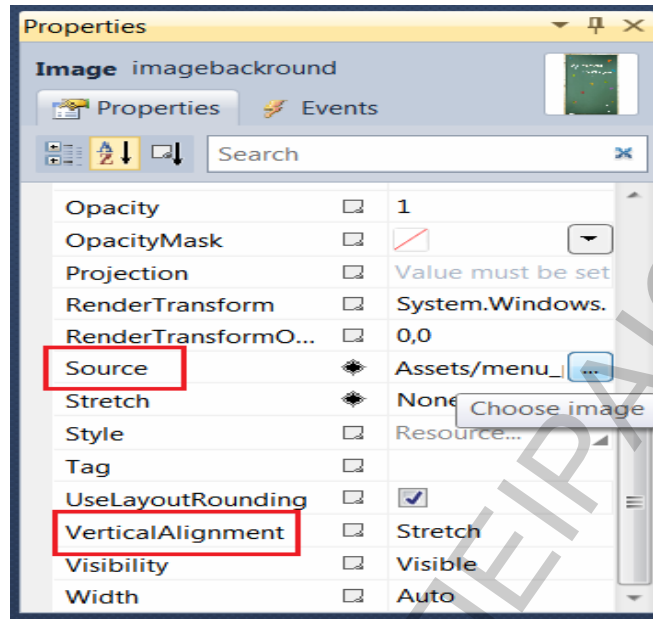
Αρχικά θα αναφερθούμε στο σχεδιαστικό κομμάτι της εφαρμογής. Θα χρησιμοποιήσουμε ένα **"grid"** όπου μέσα σε αυτό θα εμπεριέχονται τα τέσσερα κουμπιά και η εικόνα του background μας.

Για να προσθέσουμε την εικόνα του background από την αριστερή στήλη **toolbox** θα επιλέξουμε το εργαλείο **"Image"** (Εικόνα 4.3.3-3) και θα το σύρουμε μέσα στην οθόνη σχεδίασης.



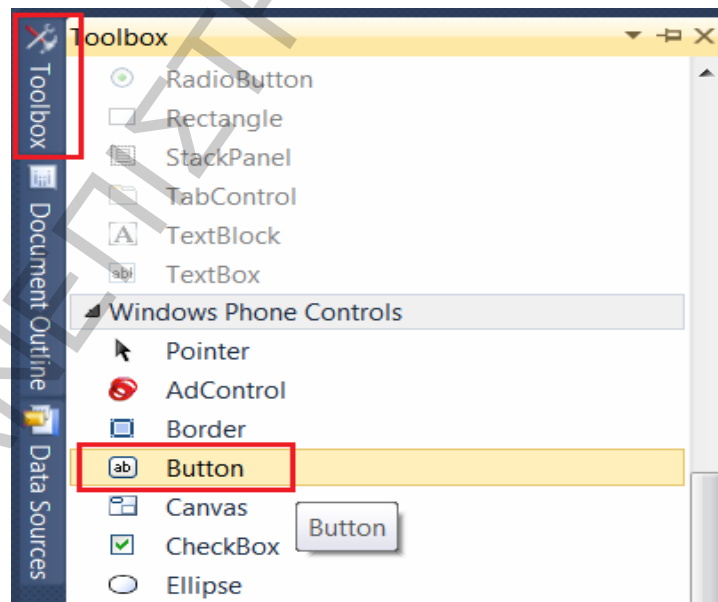
Εικόνα 4.3.3-3

Στην συνέχεια έχοντας επιλεγμένη την εικόνα πάμε κάτω δεξιά στο παράθυρο **"Properties"** (εικόνα 4.3.3-4) και ορίζουμε τις ιδιότητες της εικόνας μας όπως πχ οι διαστάσεις που θα έχει , η ευθυγράμμιση (alignment ), το όνομα κτλ. Από το πεδίο **Source** των ιδιοτήτων ορίζουμε και το μονοπάτι όπου βρίσκεται η εικόνα που θέλουμε να εμφανίζεται μέσα στο εργαλείο **Image** που τοποθετήσαμε.[3]



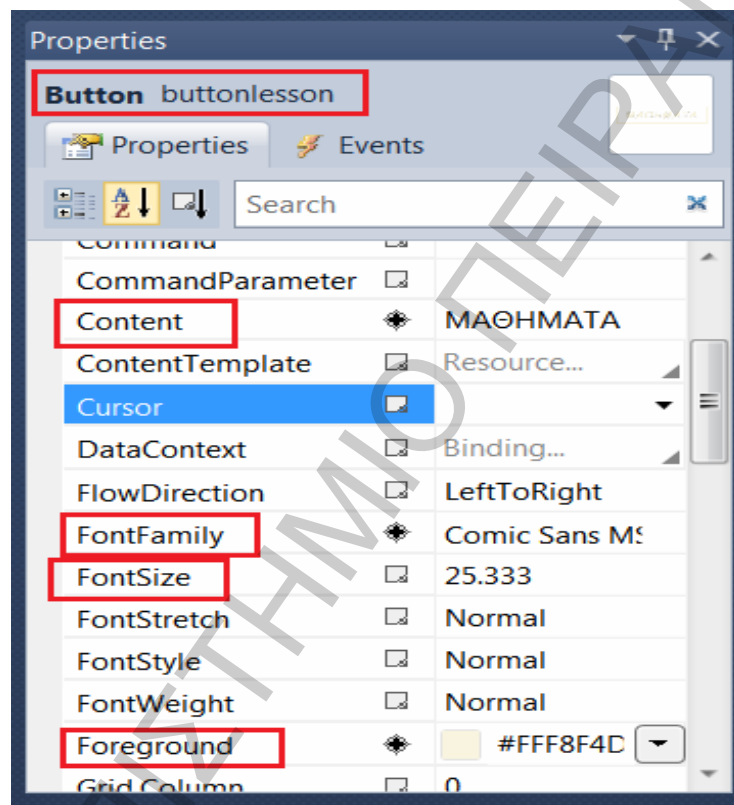
Εικόνα 4.3.3-4

Επόμενο στάδιο είναι να τοποθετήσουμε τα τέσσερα **buttons** , η διαδικασία είναι περίπου αντίστοιχη με την τοποθέτηση της εικόνας απλώς από το πεδίο **toolbox** επιλέγουμε **button** (Εικόνα 4.3.3-5) .



Εικόνα 4.3.3-5

Στην συνέχεια έχοντας επιλεγμένο το αντίστοιχο κουμπί πάμε και ορίζουμε τις ιδιότητες του μέσω του παραθύρου **Properties**. Από τις πιο σημαντικές είναι το όνομα του κουμπιού , το περιεχόμενο του ( **content** ) , το είδος της γραμματοσειράς (**FontFamily**) , το μέγεθος της γραμματοσειράς (**FontSize**) , το χρώμα (**Foreground**) κτλ. (εικόνα 4.3.3-6) .



Εικόνα 4.3.3-6

Την αντίστοιχη διαδικασία κάνουμε και για τα τέσσερα κουμπιά , θα μπορούσαμε αντί να χρησιμοποιήσουμε τα έτοιμα εργαλεία μέσω του **toolbox** να πάμε και να γράψουμε κατευθείαν μέσω κώδικα **XAML** .

Έχοντας ολοκληρώσει το σχεδιαστικό κομμάτι της "**Main Page**" το επόμενο βήμα είναι να ορίσουμε λειτουργίες στα κουμπιά (**buttons**) που έχουμε χρησιμοποιήσει. Κάνοντας διπλό κλικ στο **button "ΜΑΘΗΜΑΤΑ"** αυτόματα το **Visual Studio** του προσθέτει το event **click** το οποίο ενεργοποιείται πατώντας ο χρήστης το κουμπί εναλλακτικός τρόπος είναι μέσω του κώδικα **XAML** να βρούμε το **button** που θέλουμε και μέσα στις ιδιότητες του να γράψουμε

την **click**, εκεί μπορούμε να ορίσουμε και το όνομα που θα έχει το **event** μας . Με το διπλό κλικ το Visual Studio θα μας μεταφέρει στο "προγραμματιστικό" κομμάτι της σελίδας **MainPage.xaml.cs** όπου θα μας έχει έτοιμο το **event click** και εμείς απλώς θα πρέπει να πάμε να ορίσουμε τι θέλουμε να κάνει. Στην προκειμένη περίπτωση να μας μεταφέρει στην σελίδα που ξεκινάνε οι αριθμοί. Θα χρησιμοποιήσουμε την συνάρτηση **Navigation.Service** για το συγκεκριμένο σκοπό (εικόνα 4.3.3-7) .

```
private void buttonlesson_Click(object sender, RoutedEventArgs e)
{
    this.NavigationService.Navigate(new Uri("/Page1.xaml", UriKind.Relative));
}
```

Εικόνα 4.3.3-7

Ακριβώς την ίδια διαδικασία ακολουθούμε για τα **buttons "ΠΑΙΧΝΙΔΙ"** και **"ΟΔΗΓΙΕΣ"** καθώς και για το **"BlueShyMonkey Apps"** , όπου στο κομμάτι κώδικα χρησιμοποιούμε το παρακάτω (εικόνα 4.3.3-8)

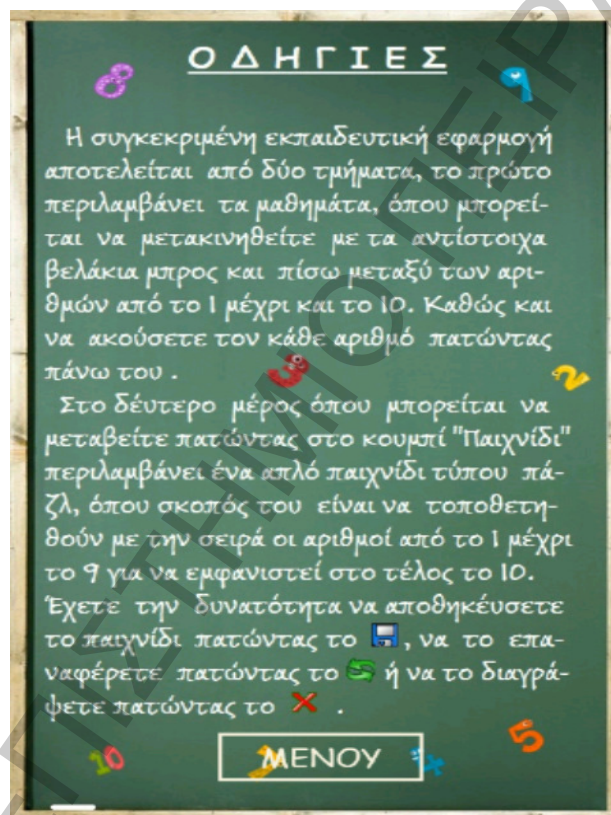
```
private void buttonApps_Click(object sender, RoutedEventArgs e)
{
    Microsoft.Phone.Tasks.MarketplaceSearchTask t = new Microsoft.Phone.Tasks.MarketplaceSearchTask();
    t.ContentType = Microsoft.Phone.Tasks.MarketplaceContentType.Applications;
    t.SearchTerms = "BlueShyMonkey";
    t.Show();
}
```

Εικόνα 4.3.3-8

Αυτό που κάνει είναι να δημιουργήσει ένα **Task** στο τηλέφωνο το οποίο είναι να συνδεθεί στο **Marketplace** και συγκεκριμένα στο κομμάτι των εφαρμογών (**Applications**) και να ψάξει με τον όρο (**term**) που θέλουμε εμείς , εδώ **BlueShyMonkey**. [3]

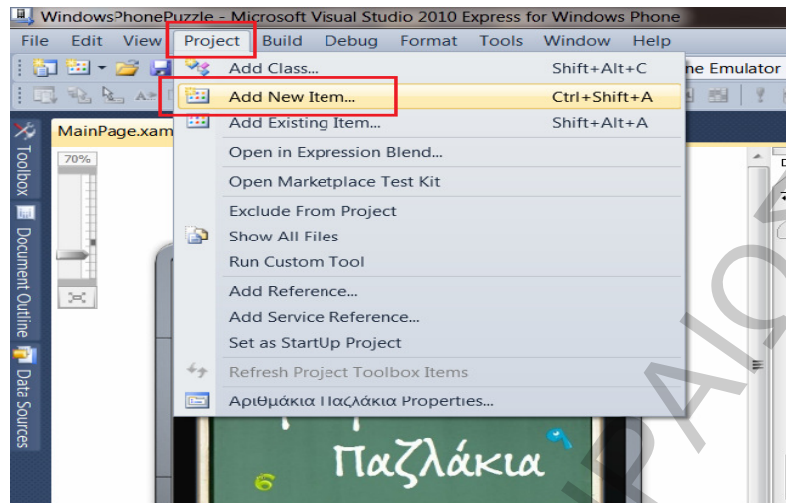
### 4.3.4 Δημιουργώντας την "Σελίδα Οδηγιών"

Στην συγκεκριμένη ενότητα θα αναφερθούμε στην δημιουργία της σελίδας όπου παρέχονται στο χρήστη οι οδηγίες χρήσης της εφαρμογής **Instructions.xaml** (εικόνα 4.3.4-1).



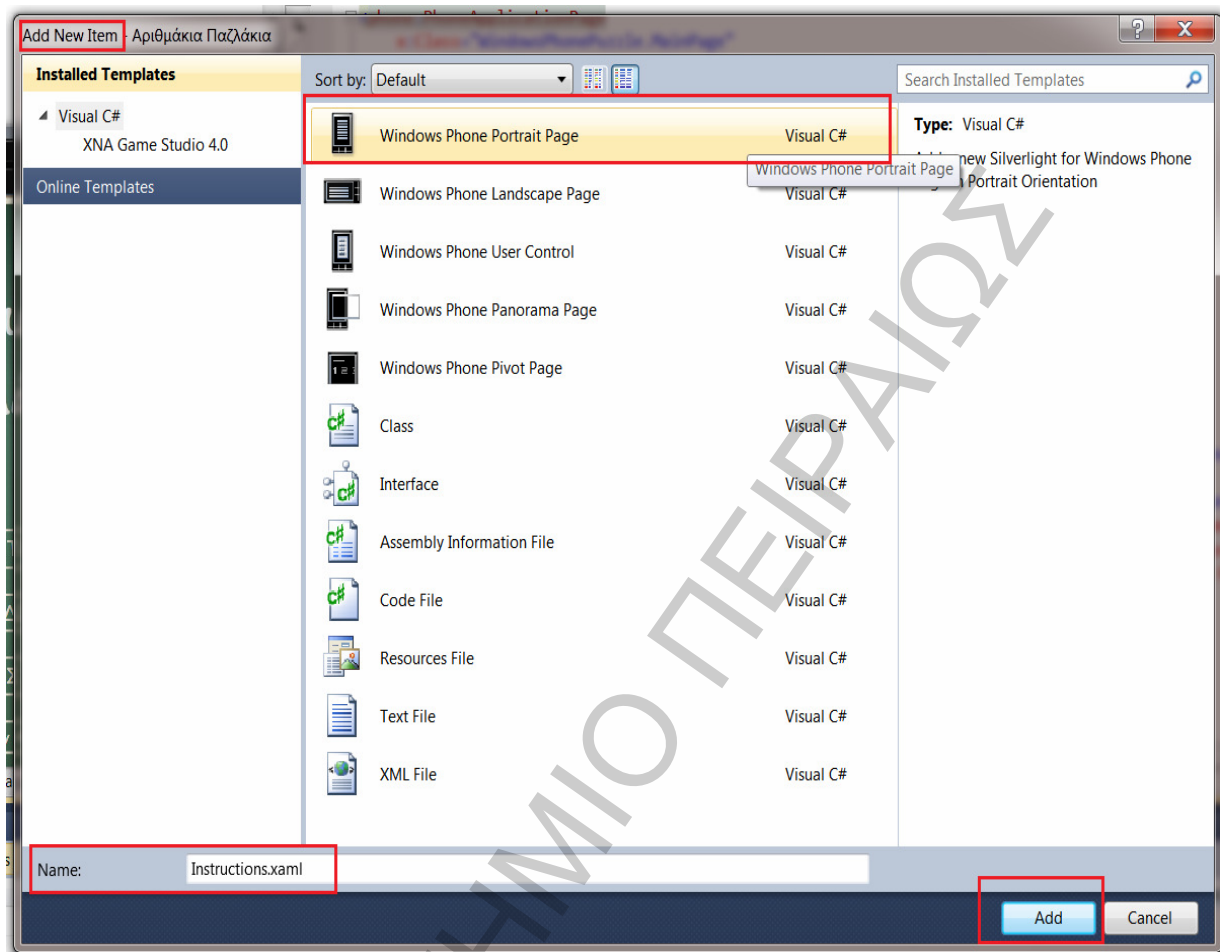
Εικόνα 4.3.4-1

Καταρχήν θα πρέπει να προσθέσουμε μια νέα σελίδα στο Project μας. Για να γίνει αυτό πάμε στις επιλογές των μενού πάνω και επιλέγουμε **Project -> Add new Item** (εικόνα 4.3.4-2)



Εικόνα 4.3.4-2

Στην συνέχεια από το καινούριο παράθυρο που θα εμφανιστεί διαλέγουμε το είδος του αντικειμένου (**Item**) στην προκειμένη περίπτωση **Windows Phone Portrait Page**, το όνομα της σελίδας **Instructions.xaml** και πατάμε **Add** (εικόνα 4.3.4-3) .



Εικόνα 4.3.4-3

Για την ανάπτυξη της σελίδας οδηγιών (**Instructions.xaml**) δεν χρησιμοποιούμε τίποτα διαφορετικά από όσα αναφέρθηκαν στην προηγούμενη ενότητα . Η σελίδα μας αποτελείται από μία εικόνα (**Image**) η οποία είναι το background και ένα κουμπί (**button**) το οποίο έχει ένα event που ορίζουμε να μας μεταφέρει στην κύρια σελίδα (**MainPage.xaml**) . Συνεπώς κάνουμε αντίστοιχες ενέργειες με πριν. [3]

### 4.3.5 Δημιουργώντας την "Πρώτη Σελίδα Αριθμών"

Στην συγκεκριμένη ενότητα θα δημιουργήσουμε την πρώτη από την σελίδα των Αριθμών **Page1.xaml** ( εικόνα 4.3.5-1 ) Αντίστοιχα δημιουργούνται και οι υπόλοιπες . Αυτό που θα κάνουν οι συγκεκριμένες σελίδες είναι να εμφανίζουν μια εικόνα με τον αντίστοιχο αριθμό

καθώς και πώς γράφεται ως λέξη . Επιπλέον θα δίνεται η δυνατότητα στο χρήστη να περιηγηθεί εμπρός και πίσω μεταξύ των αριθμών να ξαναπάει στο κεντρικό μενού επιλογών και να ακούσει τον κάθε αριθμό πατώντας πάνω στην εικόνα του ή στην λέξη του.

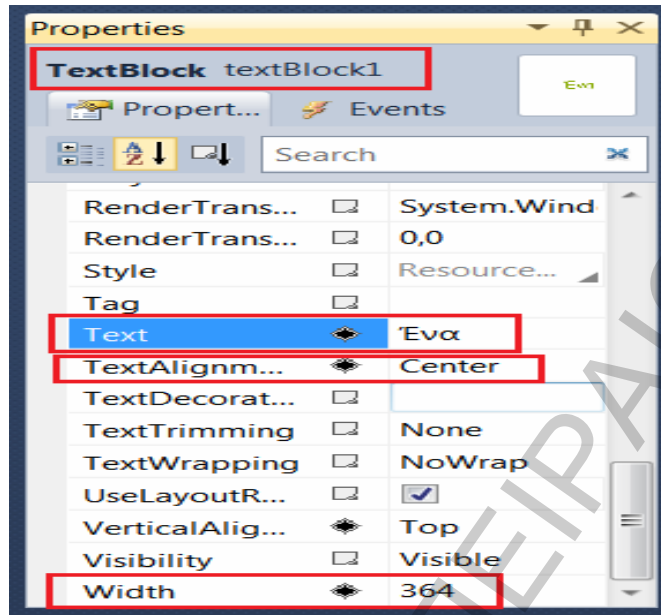


Εικόνα 4.3.5-1

Η διαδικασία που θα ακολουθήσουμε είναι η ακόλουθη, θα δώσουμε έμφαση σε εργαλεία που χρησιμοποιούμε για πρώτη φορά ενώ στα υπόλοιπα που ήδη έχουμε δει θα αναφερθούμε περιληπτικά. Καταρχήν θα δημιουργήσουμε μια νέα σελίδα την οποία θα ονομάσουμε **Page1.xaml** από το **Project** -> **Add new Item** αντίστοιχα δημιουργούμε και άλλες για τους υπόλοιπους αριθμούς. Τα εργαλεία που θα χρησιμοποιήσουμε σε κάθε σελίδα είναι τα εξής : τέσσερις εικόνες (**Images**) και ένα **textblock** . Έχουμε χρησιμοποιήσει μια εικόνα για κάθε βέλος στις οποίες θα ορίσουμε ένα event για να μας μεταφέρουν στην επόμενη και προηγούμενη σελίδα αντίστοιχα . Μια για την μεταφορά μας στην αρχική σελίδα με το μενού επιλογών (**MainPage.xaml**) και τέλος την εικόνα όπου εμφανίζεται ο

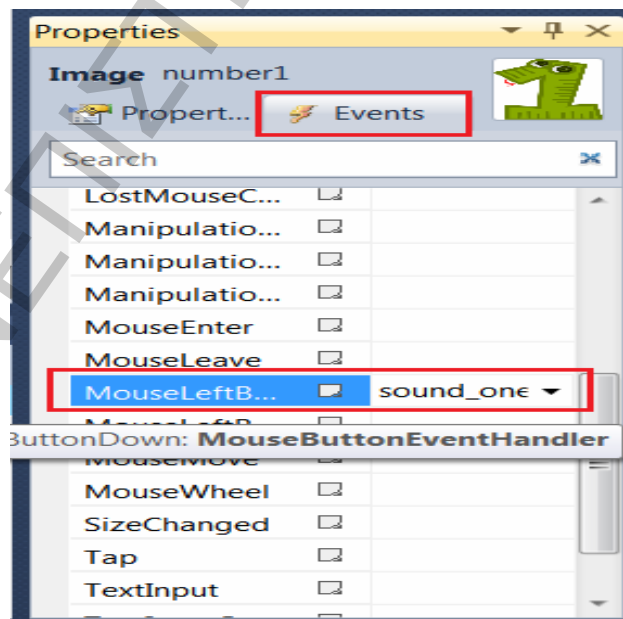






Εικόνα 4.3.5-3

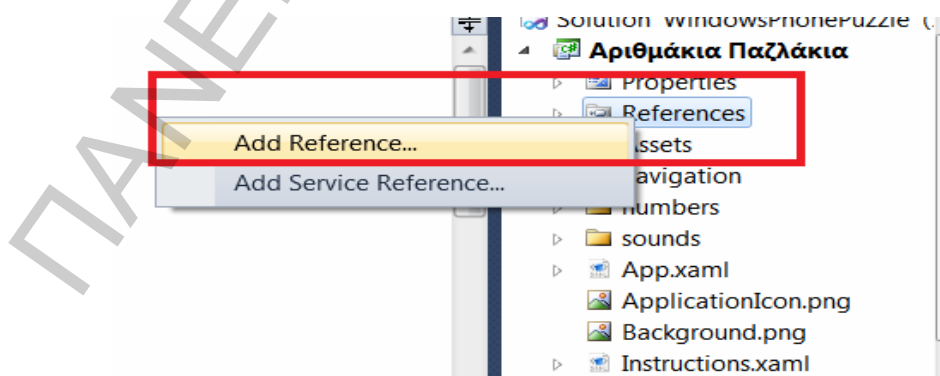
Στην συνέχεια θα ορίσουμε events στις εικόνες (**Images**) μας και στο **textblock** η διαδικασία είναι κοινή. Έχοντας επιλεγμένο το αντικείμενο που θέλουμε για παράδειγμα την εικόνα (**Image**) του αριθμού, πάμε στο παράθυρο **Properties** και επιλέγουμε το πεδίο **Events** και βρίσκουμε το event **MouseLeftButtonDown** ( Εικόνα 4.3.5-4 )



Εικόνα 4.3.5-4

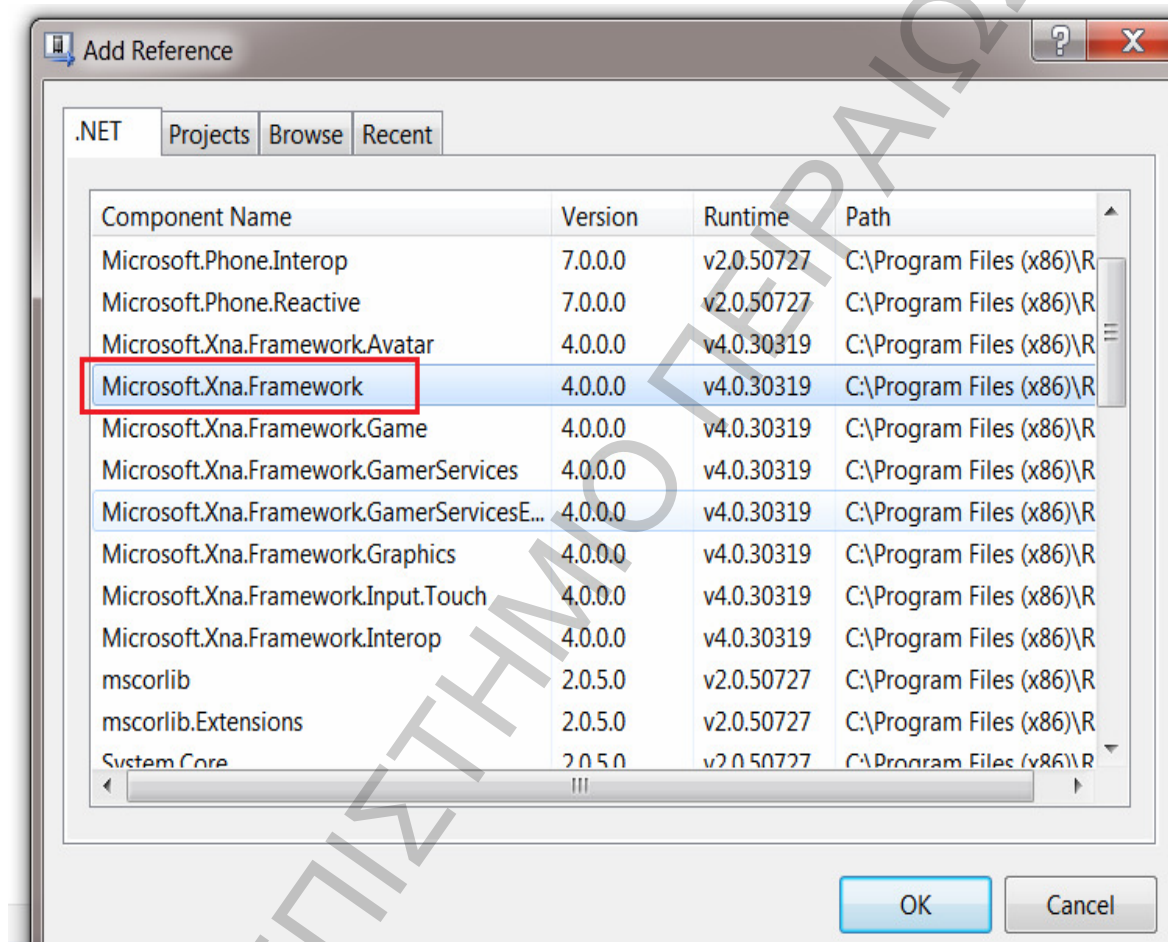
Το συγκεκριμένο **event** αυτό που κάνει είναι κάτι αντίστοιχο με το click στα buttons δηλαδή με το που πατήσει ο χρήστης στην εικόνα τρέχει την εντολή που του έχουμε ορίσει εμείς . Αφού έχουμε βρει το **event** που αναφέραμε πάμε και ορίζουμε ένα όνομα για το συγκεκριμένο **sound\_one** , κάνοντας το συγκεκριμένο το Visual Studio θα μας μεταφέρει στο "προγραμματιστικό" κομμάτι της σελίδας **Page1.xaml.cs** . Όπως θα παρατηρήσουμε έχει ετοιμάσει το αντίστοιχο event και το μόνο που έχουμε να κάνουμε εμείς είναι να ορίσουμε το τις ενέργειες θα γίνονται όταν ο χρήστης πατάει την κάθε εικόνα ή textblock . Στα βέλη και στο μενού χρησιμοποιούμε την συνάρτηση **navigationservice** όπως αναφέραμε και σε προηγούμενη ενότητα κάνοντας τις απαραίτητες αλλαγές ανάλογα με το σε ποία σελίδα θέλουμε να πλοηγηθούμε. Στην εικόνα με τον αριθμό και το textblock θα πρέπει να ορίσουμε να αναπαράγεται το ηχητικό αρχείο. Εδώ έχουμε δύο επιλογές είτε χρησιμοποιώντας το **mediaElement** το οποίο αναπαράγει αρχεία multimedia όπως video και μουσικά αρχεία είτε το **soundEffect** από την πλατφόρμα XNA το οποίο αναπαράγει ηχητικά εφέ . Γενικά χρησιμοποιούμε το **mediaElement** όταν έχουμε να αναπαράγουμε αρχεία μεγάλους μεγέθους όπως τραγούδια και video clip και το **soundEffect** για αρχεία μικρότερου μήκους όπως ηχητικά εφέ . Το μεγάλο πλεονέκτημα του **soundEffect** είναι ότι τα αρχεία αναπαράγονται ως ηχητικά εφέ το οποίο σημαίνει ότι ο χρήστης μπορεί να τα ακούσει παράλληλα με την μουσική του χωρίς να χρειαστεί να την διακόψει για να ακούσει το συγκεκριμένο κάτι το οποίο δεν ισχύει με το **mediaElement**. Από την άλλη υστερούν στην δυνατότητα ελέγχου του ηχητικού αρχείου σε σχέση με το **mediaElement** και καταλαμβάνουν περισσότερο χώρο καθώς το ηχητικό αρχείο πρέπει να είναι σε ασυμπίεστη μορφή τύπου **wav** .

Για να μπορέσουμε να χρησιμοποιήσουμε το **soundEffect** το οποίο ανήκει στην πλατφόρμα **XNA** θα πρέπει πρώτα να κάνουμε παραπομπή (**reference**) στην αντίστοιχη βιβλιοθήκη καθώς όπως είπαμε και στην αρχή η εφαρμογή που έχουμε ξεκινήσει να δημιουργήσουμε είναι **Silverlight** . Για να γίνει αυτό θα πρέπει να κάνουμε δεξί κλικ στο πεδίο **References** στο **Solution Explorer** και επιλέγουμε **Add Reference** (εικόνα 4.3.5-5 )



Εικόνα 4.3.5-5

Στο παράθυρο που θα εμφανιστεί από την καρτέλα **.NET** επιλέγουμε **Microsoft.Xna.Framework** ( εικόνα 4.3.5-6 )



Εικόνα 4.3.5-6

Στην συνέχεια για να μπορεί η εφαρμογή μας να κάνει χρήση των συγκεκριμένων θα χρειαστεί στα **using statements** να προσθέσουμε και τα **using Microsoft.Xna.Framework;** και **using Microsoft.Xna.Framework.Audio ;** ( εικόνα 4.3.5-7 )

```
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Audio;
using System.Windows.Resources;
using System.Windows.Threading;

namespace WindowsPhonePuzzle
```

Εικόνα 4.3.5-7

Στην συνέχεια δηλώνουμε τις μεταβλητές που θα αντιστοιχούν στους ήχους μας (εικόνα 4.3.5-8)

```
namespace WindowsPhonePuzzle
{
    public partial class Page1 : PhoneApplicationPage
    {
        private SoundEffect soundOne;
        private SoundEffect soundBack;
        private SoundEffect soundNext;
        public Page1()
        {
            InitializeComponent();
        }
    }
}
```

Εικόνα 4.3.5-8

Φορτώνουμε του ήχους μας με την χρήση της συνάρτησης **LoadSound** στην σελίδα μας (εικόνα 4.3.5-9)

```
public Page1()
{
    InitializeComponent();
    LoadSound("sounds/1.wav", out soundOne);
    LoadSound("sounds/back.wav", out soundBack);
    LoadSound("sounds/next.wav", out soundNext);
}
```

Εικόνα 4.3.5-9

Στην συνέχεια θα πρέπει να χρησιμοποιήσουμε την κλάση `DispatcherTimer` για να προσομοιώσουμε τον κύκλο παιχνιδιού (**game loop**) που υπάρχει στην πλατφόρμα του **XNA** (εικόνα 4.3.5-10)

```
// Timer to simulate the XNA game loop (SoundEffect classes are from the XNA Framework)

DispatcherTimer XnaDispatchTimer = new DispatcherTimer();
XnaDispatchTimer.Interval = TimeSpan.FromMilliseconds(50);

// Call FrameworkDispatcher.Update to update the XNA Framework internals.
XnaDispatchTimer.Tick += delegate { try { FrameworkDispatcher.Update(); } catch { } };

// Start the DispatchTimer running.
XnaDispatchTimer.Start();
}
```

Εικόνα 4.3.5-10

Επόμενο βήμα είναι να ορίσουμε την **LoadSound** μέσω της οποίας φορτώνετε το ηχητικό αρχείο χρησιμοποιούμε και την μέθοδο **try - catch** έτσι ώστε αν προκύψει κάποιο πρόβλημα στην φόρτωση του ηχητικού εφέ να βγάλει ενημερωτικό μήνυμα στο χρήστη (εικόνα 4.3.5-11)

```
private void LoadSound(String SoundFilePath, out SoundEffect Sound)
{
    // For error checking, assume we'll fail to load the file.
    Sound = null;

    try
    {
        // Holds informations about a file stream.
        StreamResourceInfo SoundFileInfo = App.GetResourceStream(new Uri(SoundFilePath,
            UriKind.Relative));

        // Create the SoundEffect from the Stream
        Sound = SoundEffect.FromStream(SoundFileInfo.Stream);
    }
    catch (NullReferenceException)
    {
        // Display an error message
        MessageBox.Show("Δεν μπόρεσε να φορτώσει τον ήχο " + SoundFilePath);
    }
}
```

Εικόνα 4.3.5-11

Τέλος πάμε και ορίζουμε στο **event** που δημιουργήσαμε αρχικά το **sound\_one** τι ακριβώς θα γίνεται όταν ο χρήστης πατάει την εικόνα δηλαδή να αναπαράγεται το αντίστοιχο ηχητικό αρχείο και στην προκειμένη περίπτωση κάνουμε χρήση των **try - catch** με εμφάνιση πάλι ειδοποιητικού μηνύματος αν τυχόν προκύψει κάτι στην αναπαραγωγή αυτή την φορά του αρχείου (εικόνα 4.3.5-12).

```
private void sound_one(object sender, MouseButtonEventArgs e)
{
    try
    {
        soundOne.Play();
    }

    catch (NullReferenceException)
    {
        MessageBox.Show("Πρόβλημα με το αρχείο Ένα.");
    }
}
```

εικόνα 4.3.5-12

Συνεπώς έχουμε ολοκληρώσει και την δημιουργία της συγκεκριμένης σελίδας αντίστοιχα βήματα κάνουμε για να ορίσουμε την αναπαραγωγή ηχητικών εφέ και στις εικόνες με τα βέλη πλοήγησης . Για την δημιουργία των επόμενων σελίδων αριθμών τα βήματα είναι αντίστοιχα δημιουργώντας νέες σελίδες **Page2.xaml** , **Page3.xaml** κτλ και ακλουθώντας πάλι τα ίδια βήματα για την διαμόρφωση του user interface και τον ορισμό των λειτουργιών . [3]

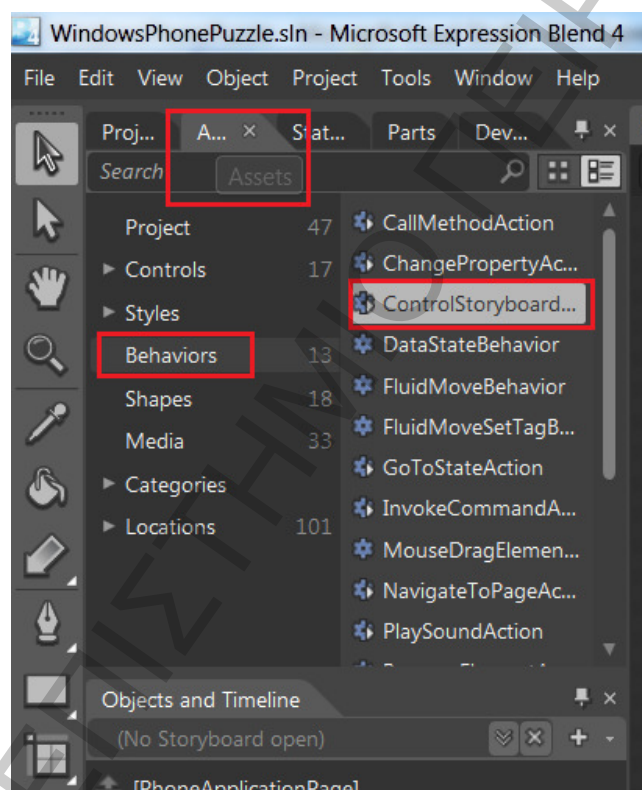
#### 4.3.6 Δημιουργώντας τα εφέ κίνησης

Στην συγκεκριμένη ενότητα θα δούμε πως μπορούμε να προσθέσουμε στις εικόνες που αναπαριστάνουν τους αριθμούς και στα textblock κάποια εφέ κίνησης. Για την εισαγωγή των εφέ μπορεί να γίνει με δύο τρόπους είτε μέσω του **Visual Studio** και την χρήση κώδικα **XAML**,



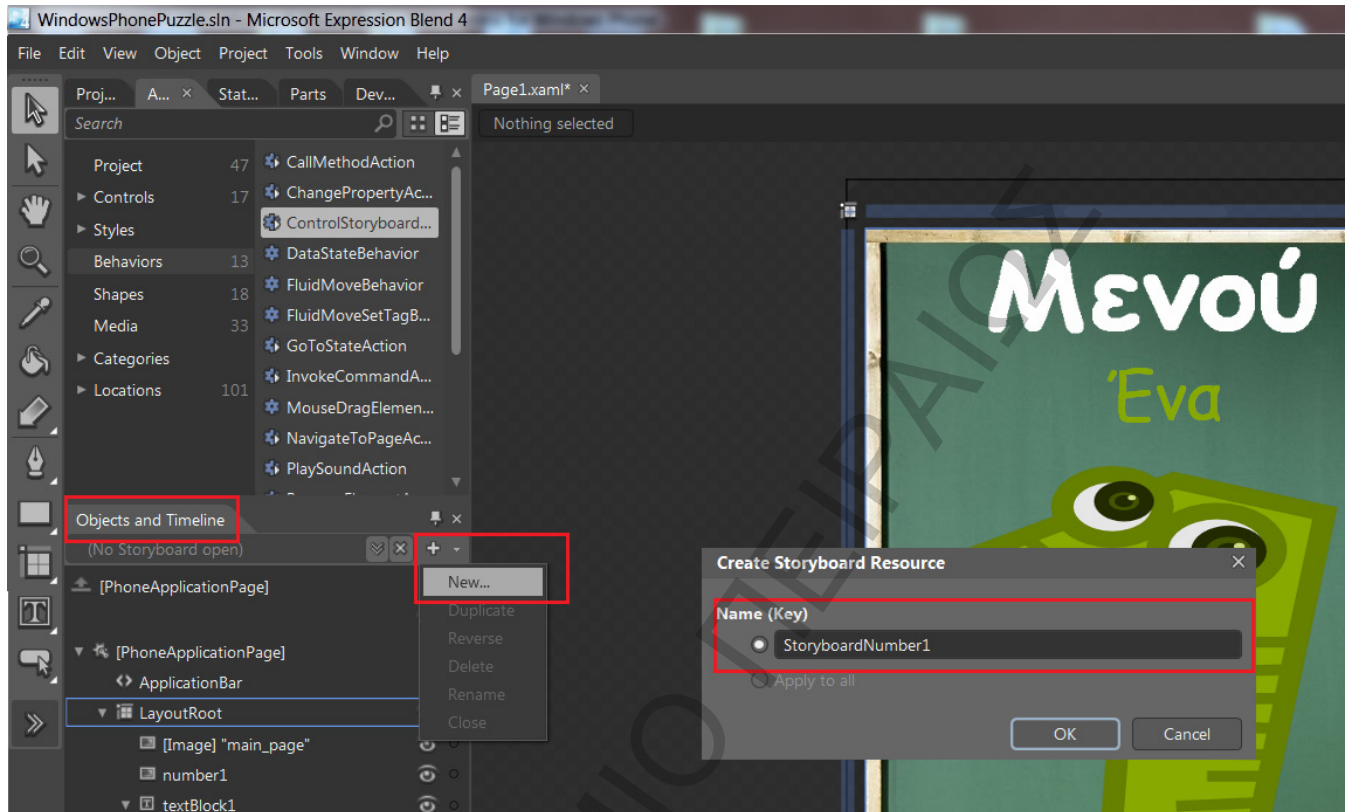
είτε μέσω του **Expression Blend 4 Express** όπου δεν χρειάζεται καθόλου χρήση κώδικα καθώς όλα γίνονται σε σχεδιαστικό περιβάλλον. Εμείς στην συγκεκριμένη ενότητα θα δείξουμε πως μπορεί να γίνει αυτό με την χρήση του δεύτερου τρόπου δηλαδή με το **Expression Blend Express** το οποίο διατίθεται δωρεάν από την **Microsoft**.

Όπως έχουμε ανοιχτό το project μας στο Visual Studio , κάνουμε δεξί κλικ στο όνομα του και πατάμε την επιλογή **Open in Expression Blend**. Διαλέγουμε την εικόνα στην οποία θέλουμε να δημιουργήσουμε το εφέ και από το πεδίο **Assets** -> **Behaviors** σύρουμε την επιλογή **ControlStoryboardAction**(εικόνα 4.3.6-1 ) στην εικόνα ή το textblock μας.



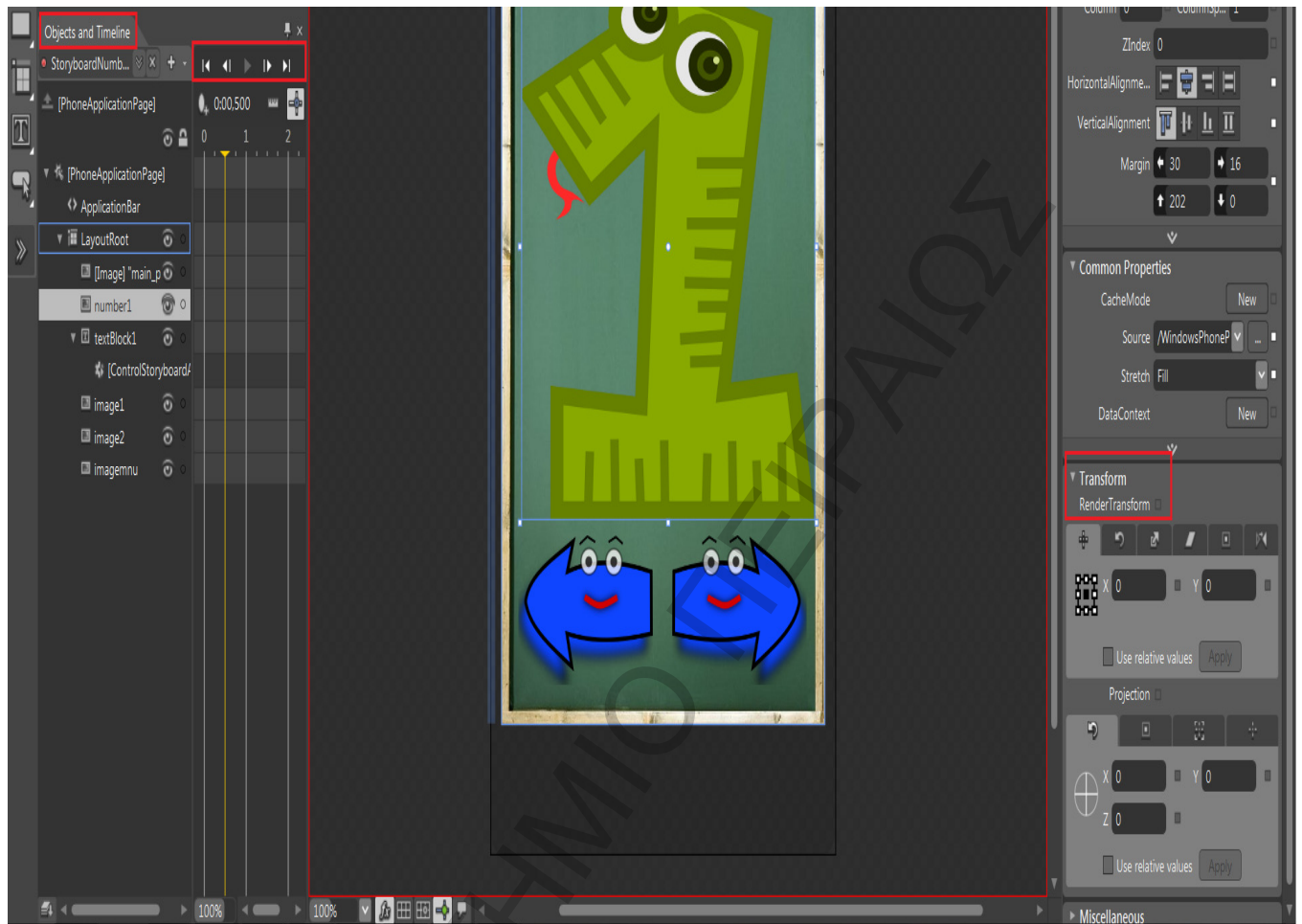
Εικόνα 4.3.6-1

Στην συνέχεια από το πεδίο **Objects and Timeline** επιλέγουμε την επιλογή **New** για να δημιουργήσουμε ένα καινούριο StoryBoard και στο πεδίο που μας βγάζει ορίζουμε το όνομα του (εικόνα 4.3.6-2).



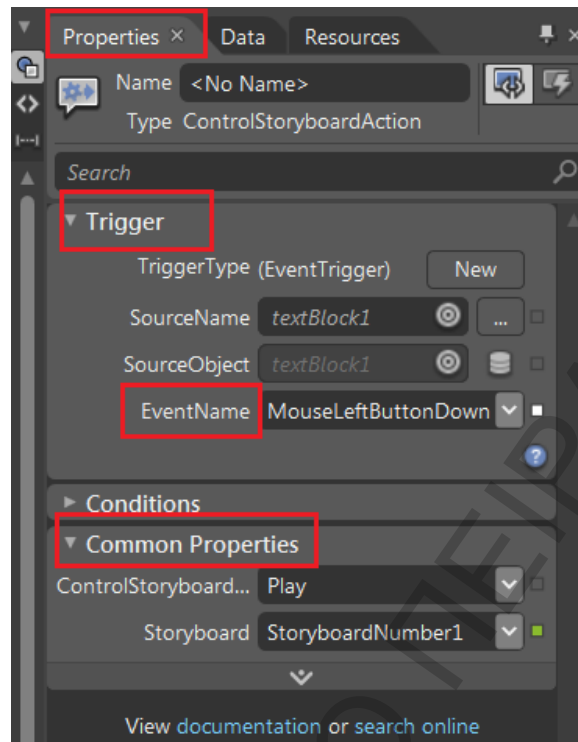
Εικόνα 4.3.6-2

Στην συνέχεια από την Timeline που θα εμφανιστεί στα αριστερά του **Expression Blend** διαλέγουμε την χρονική στιγμή που θέλουμε και από την δεξιά στήλη, μπορούμε να αλλάξουμε από το πεδίο **Transform** τις διάφορες επιλογές που μας δίνονται (εικόνα 4.3.6-3). Κάνοντας το ίδιο και για άλλες χρονικές στιγμές δημιουργούμε ουσιαστικά τα διάφορα εφέ κίνησης. Ανά πάσα στιγμή έχουμε την δυνατότητα μέσω των επιλογών στην Timeline να αναπαράγουμε το εφέ μας καθώς και να μεταβούμε στις διάφορες χρονικές στιγμές που έχουμε ορίσει. [4]



Εικόνα 4.3.6-3

Αφού έχουμε δημιουργήσει το εφέ μας το επόμενο και τελικό στάδιο είναι να ορίσουμε να αναπαράγεται όταν ο χρήστης πατάει την εικόνα ή το textblock. Αφού κλείσουμε το StoryBoard . Προσέχουμε να έχουμε στην στήλη **Objects and Timeline** και επιλέγουμε το **ControlStoryboardAction** της εικόνας ή textblock που μας ενδιαφέρει. Στην δεξιά στήλη **Properties** και συγκεκριμένα στο πεδίο **Trigger** θα πρέπει να ορίσουμε στο **EventName** βάσει ποιού Event θα αναπαράγεται το εφέ μας στην προκειμένη περίπτωση όταν το πατήσει ο χρήστης οπότε διαλέγουμε **MouseLeftButtonDown**. Τέλος στα common Properties πρέπει να προσέξουμε στο πεδίο **ControlStoryboardOption** να έχουμε επιλεγμένη την επιλογή **Play** και στο πεδίο **Storyboard** να βάλουμε το όνομα του storyboard που δημιουργήσαμε στην προκειμένη περίπτωση **StoryboardNumberOne** (εικόνα 4.3.6-4) . [4]

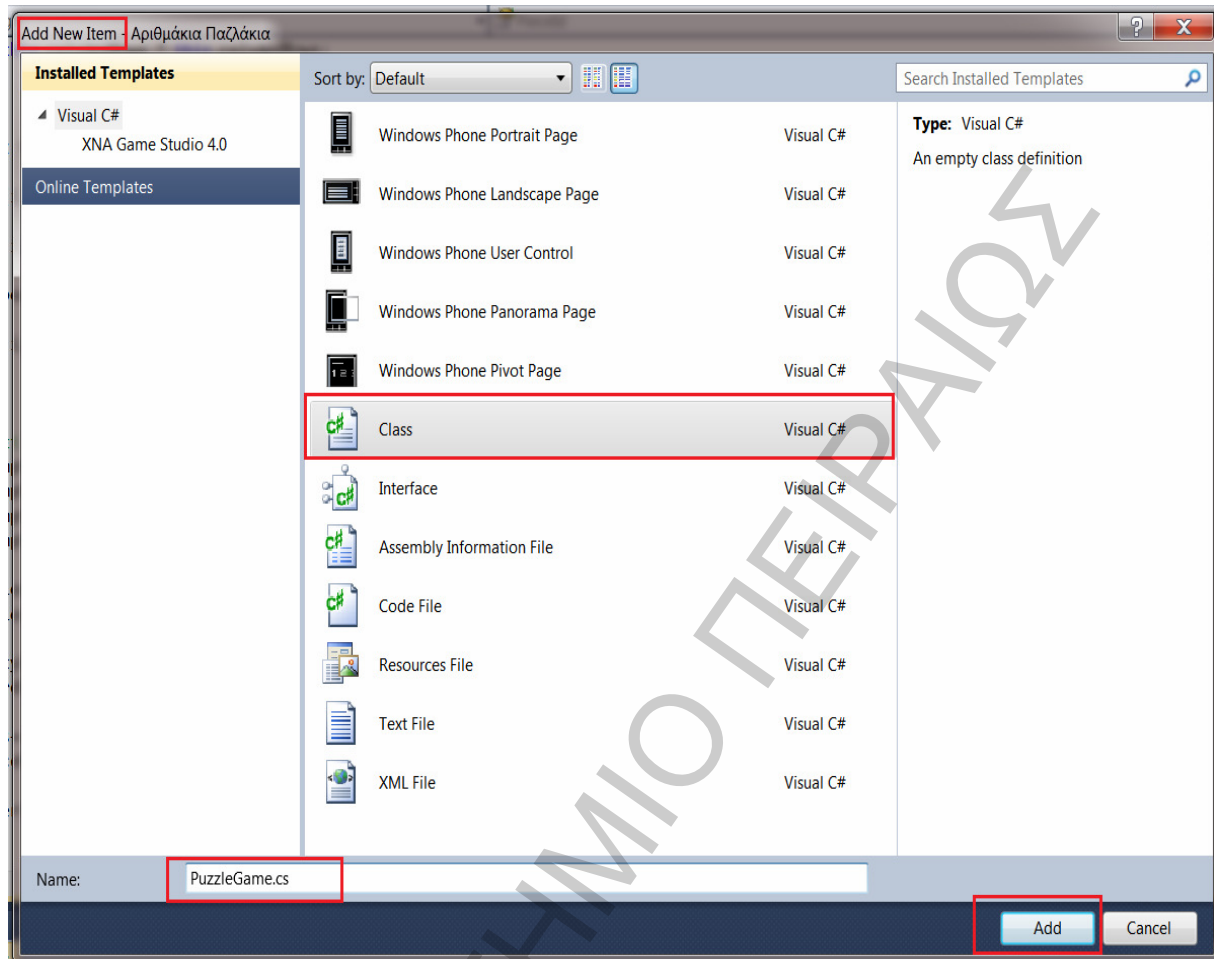


Εικόνα 4.3.6-4

### 4.3.7 Δημιουργώντας τις απαραίτητες κλάσεις

Πριν ασχοληθούμε με την διαμόρφωση της σελίδας θα χρειαστεί να προσθέσουμε δυο κλάσεις στο project μας την **IsolatedStorageHelper.cs** η οποία κάνει χρήση του Isolated storage για την αποθήκευση δεδομένων στην μνήμη του τηλεφώνου και την **PuzzleGame.cs** η οποία χρησιμεύει για την δημιουργία της λογικής του puzzle .

Για να προσθέσουμε τις δύο κλάσεις θα κάνουμε μια διαδικασία αντίστοιχη με την προσθήκη σελίδων (Pages) . Πατάμε στο menu **Project -> Add Class** ή από το menu **Project -> Add New Item -> Class** και στο παράθυρο που θα εμφανιστεί δίνουμε το όνομα της κλάσης μας (εικόνα 4.3.7-1) .



Εικόνα 4.3.7-1

Προσθέτουμε την πρώτη κλάση μας την **PuzzleGame.cs** η οποία αυτό που κάνει είναι να αρχικοποιεί το ταμπλό του παιχνιδιού μας καθώς και να κάνει έλεγχο των κινήσεων των κομματιών του puzzle . Παρακάτω παρουσιάζεται αναλυτικά ο κώδικας της συγκεκριμένης κλάσης (εικόνα 4.3.7-2).

```
using System;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Ink;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;

namespace WindowsPhonePuzzle
{
    public class PieceUpdatedEventArgs : EventArgs
    {
        public int PieceId { get; set; }
        public Point NewPosition { get; set; }
    }

    public class GameOverEventArgs : EventArgs
    {
        public int TotalMoves { get; set; }
    }

    public class PuzzleState
    {
        public bool IsPlaying { get; set; }
        public int ColsAndRows { get; set; }
        public int[] Board { get; set; }
        public int TotalMoves { get; set; }
    }
}
```

εικόνα 4.3.7-2

Ορίζουμε παρακάτω την `PuzzleGame` (εικόνα 4.3.7-3)

```
public class PuzzleGame
{
    private int colsAndRows;
    private int[] board;
    private int totalMoves;
    private bool isPlaying;

    public int ColsAndRows { get { return this.colsAndRows; } }
    public int TotalMoves { get { return this.totalMoves; } }
    public bool IsPlaying { get { return this.isPlaying; } }
    public int[] BoardPieces { get { return this.board; } }

    public PuzzleGame(int colsAndRows)
    {
        if (colsAndRows < 2)
        {
            throw new ArgumentOutOfRangeException("colsAndRows");
        }

        this.colsAndRows = colsAndRows;
        int totalPieces = colsAndRows * colsAndRows;
        this.board = new int[totalPieces];
        this.Reset();
    }

    public EventHandler GameStarted;
    public EventHandler<PieceUpdatedEventArgs> PieceUpdated;
    public EventHandler<GameOverEventArgs> GameOver;

    public void Reset()
    {
        this.isPlaying = false;
        int totalPieces = colsAndRows * colsAndRows;
        this.board = new int[totalPieces];
        this.Reset();
    }

    public EventHandler GameStarted;
    public EventHandler<PieceUpdatedEventArgs> PieceUpdated;
    public EventHandler<GameOverEventArgs> GameOver;

    public void Reset()
    {
        this.isPlaying = false;
        int totalPieces = colsAndRows * colsAndRows;
        for (int n = 0; n < totalPieces - 1; n++)
        {
            this.board[n] = n;
            int nx = n / ColsAndRows;
            int ny = n % ColsAndRows;
            this.InvokePieceUpdated(n, new Point(nx, ny));
        }
    }
}
```

εικόνα 4.3.7-3α

```
public void NewGame()
{
    int totalPieces = colsAndRows * colsAndRows;
    this.totalMoves = 0;

    // αρχικοποίηση ταμπλώ
    for (int n = 0; n < totalPieces - 1; n++)
    {
        this.board[n] = n;
    }

    Random rand = new Random(System.DateTime.Now.Second);
    for (int n = 0; n < 100; n++)
    {
        int n1 = rand.Next(totalPieces - 1);
        int n2 = rand.Next(totalPieces - 1);
        if (n1 != n2)
        {
            int tmp = this.board[n1];
            this.board[n1] = this.board[n2];
            this.board[n2] = tmp;
        }
    }

    this.board[totalPieces - 1] = -1;

    for (int n = 0; n < totalPieces - 1; n++)
    {
        int nx = n / colsAndRows;
        int ny = n % colsAndRows;
        if (this.board[n] >= 0)
        {
            this.InvokePieceUpdated(this.board[n], new Point(nx, ny));
        }
    }
}
```

εικόνα 4.3.7-3β



```
    }  
  }  
  
  this.isPlaying = true;  
  if (this.GameStarted != null)  
  {  
    this.GameStarted(this, null);  
  }  
}  
  
// (0 = καθόλου κίνηση )  
//   | 1 |  
// ---+---+---  
// 4 | 0 | 2  
// ---+---+---  
//   | 3 |  
public int CanMovePiece(int pieceId)  
{  
  if (!this.isPlaying)  
  {  
    return 0;  
  }  
  
  int totalPieces = this.colsAndRows * this.colsAndRows;  
  int boardLoc = -1;  
  int emptyLoc = -1;  
  
  for (int i = 0; i < totalPieces; i++)  
  {  
    if (this.board[i] == pieceId)  
    {  
      boardLoc = i;  
    }  
    else if (this.board[i] == -1)
```

εικόνα 4.3.7-3γ

```
        {
            emptyLoc = i;
        }
    }

    if ((boardLoc == emptyLoc + 1) ||
        (boardLoc == emptyLoc - 1) ||
        (boardLoc == emptyLoc + ColsAndRows) ||
        (boardLoc == emptyLoc - ColsAndRows))
    {
        if (boardLoc + 1 == emptyLoc)
        {
            return 3;
        }
        else if (boardLoc - 1 == emptyLoc)
        {
            return 1;
        }
        else if (boardLoc - this.ColsAndRows == emptyLoc)
        {
            return 4;
        }
        else if (boardLoc + this.ColsAndRows == emptyLoc)
        {
            return 2;
        }
    }

    return 0;
}

public bool MovePiece(int pieceId)
{
    if (!this.isPlaying)
```

εικόνα 4.3.7-3δ

```
{
    return false;
}

int totalPieces = this.colsAndRows * this.colsAndRows;
int boardLoc = -1;
int emptyLoc = -1;

for (int i = 0; i < totalPieces; i++)
{
    if (this.board[i] == pieceId)
    {
        boardLoc = i;
    }
    else if (this.board[i] == -1)
    {
        emptyLoc = i;
    }
}

// έλεγχος αν μπορεί να γίνει κίνηση
if ((boardLoc == emptyLoc + 1) ||
    (boardLoc == emptyLoc - 1) ||
    (boardLoc == emptyLoc + ColsAndRows) ||
    (boardLoc == emptyLoc - ColsAndRows))
{
    int nx = emptyLoc / ColsAndRows;
    int ny = emptyLoc % ColsAndRows;

    this.board[emptyLoc] = pieceId;
    this.board[boardLoc] = -1;

    this.totalMoves++;
    this.InvokePieceUpdated(pieceId, new Point(nx, ny));
}
```

εικόνα 4.3.7-3ε

```
        this.CheckWinner();
        return true;
    }

    return false;
}

public void CheckWinner()
{
    bool completed = true;
    int totalPieces = colsAndRows * colsAndRows;
    for (int n = 0; n < totalPieces - 1; n++)
    {
        if (n != this.board[n])
        {
            completed = false;
            break;
        }
    }

    if (completed)
    {
        if (this.GameOver != null)
        {
            this.GameOver(this, new GameOverEventArgs { TotalMoves = this.totalMoves });
        }
    }
}

public PuzzleState GetState()
{
    return new PuzzleState
    {
```

εικόνα 4.3.7-3στ



Επόμενο βήμα είναι η προσθήκη της δεύτερης κλάσης **IsolatedStorageHelper.cs** με αντίστοιχη διαδικασία. Παρακάτω παρουσιάζεται ο αντίστοιχος κώδικας (εικόνα 4.3.7-4) . [5]

```
using System.IO;
using System.IO.IsolatedStorage;
using System.Runtime.Serialization.Json;
using System.Text;

namespace WindowsPhonePuzzle
{
    public static class IsolatedStorageHelper
    {
        public static T GetObject<T>(string key)
        {
            if (IsolatedStorageSettings.ApplicationSettings.Contains(key))
            {
                string serializedObject = IsolatedStorageSettings.ApplicationSettings[key].ToString();
                return Deserialize<T>(serializedObject);
            }

            return default(T);
        }

        public static void SaveObject<T>(string key, T objectToSave)
        {
            string serializedObject = Serialize(objectToSave);
            IsolatedStorageSettings.ApplicationSettings[key] = serializedObject;
        }

        public static void DeleteObject(string key)
        {
            IsolatedStorageSettings.ApplicationSettings.Remove(key);
        }

        private static string Serialize(object objectToSerialize)
        {
            using (MemoryStream ms = new MemoryStream())
            {
                DataContractJsonSerializer serializer = new DataContractJsonSerializer(objectToSerialize.GetType());
                serializer.WriteObject(ms, objectToSerialize);
                ms.Position = 0;

                using (StreamReader reader = new StreamReader(ms))
                {
                    return reader.ReadToEnd();
                }
            }
        }

        private static T Deserialize<T>(string jsonString)
        {
            using (MemoryStream ms = new MemoryStream(Encoding.Unicode.GetBytes(jsonString)))
            {
                DataContractJsonSerializer serializer = new DataContractJsonSerializer(typeof(T));
                return (T)serializer.ReadObject(ms);
            }
        }
    }
}
```

εικόνα 4.3.7-4

### 4.3.8 Δημιουργώντας την σελίδα για το "πλακάκι"

Στην συγκεκριμένη ενότητα θα αναφερθούμε στην δημιουργία της σελίδας για το πλακάκι (Tile), το οποίο χρησιμοποιούμε στην σελίδα του puzzle ( **PuzzlePage.xaml** ) για τον "τεμαχισμό" της εικόνας και για την κίνηση των κομματιών στο παιχνίδι. Παρακάτω παρουσιάζεται ο κώδικας για το σχεδιασμό του πλακιδίου (εικόνα 4.3.8-1).

```

<UserControl
  xmlns:local="clr-namespace:WindowsPhonePuzzle;assembly=WindowsPhonePuzzle"
  x:Class="WindowsPhonePuzzle.PuzzleTile"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  d:DesignHeight="135" d:DesignWidth="135">
  <UserControl.Resources>
    <Storyboard x:Key="ShowLabel" BeginTime="00:00:01">
      <DoubleAnimation Storyboard.TargetName="Label" Storyboard.TargetProperty="Opacity" To="0.5" Duration="00:00:00.25" />
    </Storyboard>
  </UserControl.Resources>

  <Grid Margin="0,0,1,1" >
    <Grid>
      <Image Source="Assets/Shadow.png" Margin="-2,-2,-4,-4" Stretch="Fill" />
      <Canvas Background="Transparent" Margin="0" >
        <Image x:Name="TileImage" />
      </Canvas>

      <Border CornerRadius="3">
        <Border.Background>
          <LinearGradientBrush StartPoint="0,0" EndPoint="0,3" MappingMode="Absolute">
            <GradientStop Color="#C0FFFFFF" Offset="0" />
            <GradientStop Color="#00FFFFFF" Offset="1" />
          </LinearGradientBrush>
        </Border.Background>
      </Border>

      <Border CornerRadius="3">
        <Border.Background>
          <LinearGradientBrush StartPoint="0,0" EndPoint="3,0" MappingMode="Absolute">
            <GradientStop Color="#A0FFFFFF" Offset="0" />
            <GradientStop Color="#00FFFFFF" Offset="1" />
          </LinearGradientBrush>
        </Border.Background>
      </Border>

      <Border CornerRadius="0,3,3,0" Width="3" HorizontalAlignment="Right" BorderBrush="#482408" BorderThickness="0,0,1,0">
        <Border.Background>
          <LinearGradientBrush StartPoint="0,0" EndPoint="1,0">
            <GradientStop Color="#00482408" Offset="0" />
            <GradientStop Color="#80482408" Offset="1" />
          </LinearGradientBrush>
        </Border.Background>
      </Border>

      <Border CornerRadius="0,0,3,3" Height="6" VerticalAlignment="Bottom" BorderBrush="#482408" BorderThickness="0,0,0,1" >
        <Border.Background>
          <LinearGradientBrush StartPoint="0,0" EndPoint="0,1">
            <GradientStop Color="#00482408" Offset="0" />
            <GradientStop Color="#30482408" Offset="0.5" />
            <GradientStop Color="#A0482408" Offset="1" />
          </LinearGradientBrush>
        </Border.Background>
      </Border>

      <Grid x:Name="Label" VerticalAlignment="Center" HorizontalAlignment="Center" Opacity="0.0" IsHitTestVisible="False">
        <TextBlock x:Name="Label1" FontFamily="Segoe WP Semilight" FontSize="60" Foreground="#ffebab" TextOptions.TextHintingMode="Animated" Margin="0,1,0,-1" />
        <TextBlock x:Name="Label2" FontFamily="Segoe WP Semilight" FontSize="60" Foreground="#332012" TextOptions.TextHintingMode="Animated" />
      </Grid>
    </Grid>
  </Grid>
</UserControl>

```

Εικόνα 4.3.8-1

και στην συνέχεια ο κώδικας για την **PuzzleTile.xaml.cs** (εικόνα 4.3.8-2)

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;

namespace WindowsPhonePuzzle
{
    public partial class PuzzleTile : UserControl
    {
        public PuzzleTile()
        {
            InitializeComponent();
        }

        const double imageMargin = 2;

        public ImageSource Source
        {
            get { return this.TileImage.Source; }
            set
            {
                this.TileImage.Source = value;

                this.TileImage.Height = ImageSize;
                this.TileImage.Width = ImageSize;

                double pieceSize = PieceSize - imageMargin;

                this.TileImage.Stretch = Stretch.UniformToFill;
                RectangleGeometry r = new RectangleGeometry();
                r.Rect = new Rect((XLocation * pieceSize), (YLocation * pieceSize), pieceSize, pieceSize);
                r.RadiusX = 3;
                r.RadiusY = 3;
                this.TileImage.Clip = r;
                this.TileImage.SetValue(Canvas.TopProperty, Convert.ToDouble(YLocation * pieceSize * -1));
                this.TileImage.SetValue(Canvas.LeftProperty, Convert.ToDouble(XLocation * pieceSize * -1));

                this.Label1.Text = LabelNumber.ToString();
                this.Label2.Text = LabelNumber.ToString();

                Label.Opacity = 0;
            }
        }

        public void ShowNumberLabel()
        {
            Storyboard sb = this.Resources["ShowLabel"] as Storyboard;
            if (sb != null) sb.Begin();
        }

        public void HideNumberLabel()
        {
            Label.Opacity = 0;
        }

        public int LabelNumber { get; set; }
        public int XLocation { get; set; }
        public int YLocation { get; set; }
        public double PieceSize { get; set; }
        public double ImageSize { get; set; }
    }
}

```

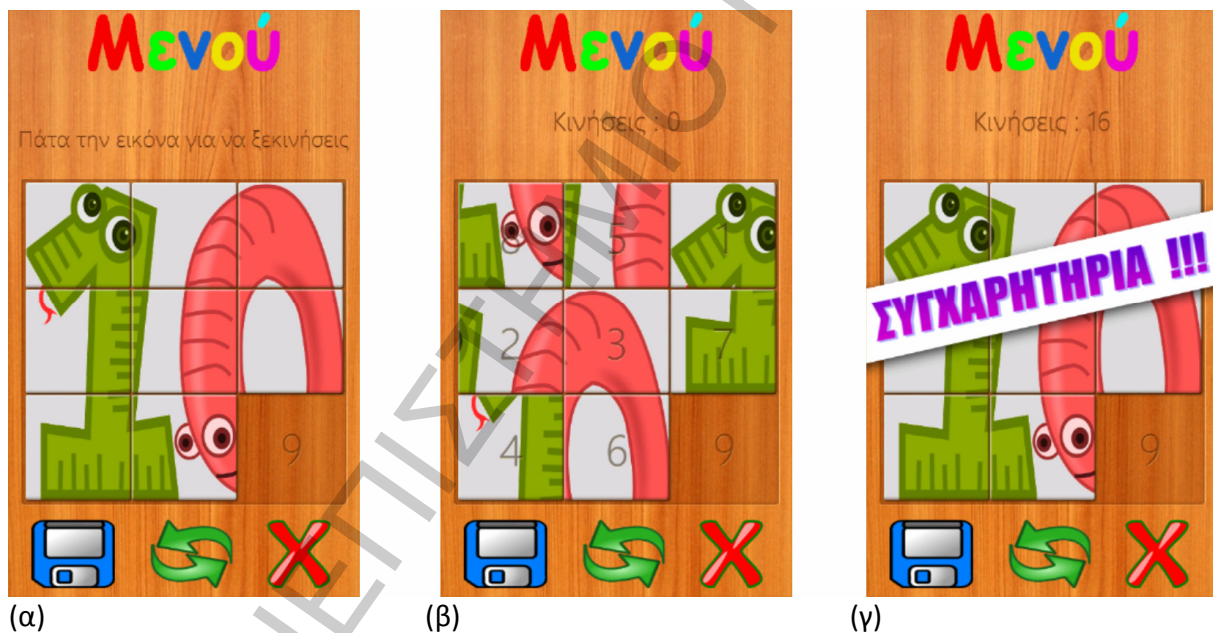
εικόνα 4.3.8-2



### 4.3.9 Δημιουργώντας την σελίδα του "Παιχνιδιού"

Επόμενο βήμα είναι η δημιουργία της σελίδας του παιχνιδιού **PuzzlePage.xaml** . Στην συγκεκριμένη σελίδα λαμβάνει μέρος όλη η δράση του εκπαιδευτικού μας παιχνιδιού , όπου σκοπός του είναι ο χρήστης να βάλει σωστά του αριθμούς από το 1 - 9 για να εμφανιστεί στην συνέχεια ο αριθμός δέκα .

Η σελίδα μας θα αποτελείται από εργαλεία που έχουμε δει μέχρι τώρα αλλά και κάποια καινούρια τα οποία αυτά θα εξηγήσουμε κυρίως. Στο πάνω μέρος υπάρχει μία εικόνα η οποία μας μεταφέρει στο κυρίως μενού **MainPage.xaml** ακριβώς από κάτω ένα **textblock** που αναφέρει στο χρήστη να πατήσει πάνω στην εικόνα για να ξεκινήσει το παιχνίδι (α) και όταν την πατήσει εμφανίζει τις κινήσεις που έχει κάνει μέχρι τώρα για να λύσει το ruzzle (β) . Με το που ολοκληρώσει ο χρήστης το ruzzle εμφανίζεται ένα εφέ όπου στριφογυρίζει την εικόνα και εμφανίζει ένα banner με την ένδειξη "ΣΥΓΧΑΡΗΤΗΡΙΑ" (γ) ( εικόνα 4.3.9-1 )



Εικόνα 4.3.9-1

Θα κάνουμε χρήση ενός νέου εργαλείου του **Canvas** το οποίο είναι ένα εργαλείο το οποίο θα περιέχει την εικόνα του ruzzle η εισαγωγή του γίνεται με αντίστοιχη διαδικασία **Toolbox -> Canvas** . Ένα ακόμα νέο στοιχείο είναι η χρήση στυλ (**style**) κάτι αντίστοιχο με την HTML οπού αντί να ορίζουμε την ιδιότητα ξεχωριστά σε κάθε εργαλείο ορίζουμε ένα αρχικό στυλ και το χρησιμοποιούμε στην συνέχεια στα αντίστοιχα εργαλεία εδώ θα το δείξουμε με το **textblock** .

Στην ακόλουθη εικόνα 4.3.9-2 φαίνεται πως ορίζουμε το **style** στα δύο textblock που χρησιμοποιούμε για να πετύχουμε το εφέ με τα σκαλισμένα γράμματα. Έχει δύο βασικές ενότητες το **Key** όπου ουσιαστικά είναι το όνομα του και την Target Type όπου είναι ο τύπος του εργαλείου στο οποίο απευθύνεται εδώ **textblock**.

```
<Style x:Key="EtchedTextBack" TargetType="TextBlock" >
  <Setter Property="FontFamily" Value="Segoe WP Light" />
  <Setter Property="Foreground" Value="#F0210B01" />
  <Setter Property="FontSize" Value="30" />
  <Setter Property="Margin" Value="0,-1,0,1" />
</Style>

<Style x:Key="EtchedTextFront" TargetType="TextBlock" >
  <Setter Property="FontFamily" Value="Segoe WP Light" />
  <Setter Property="Foreground" Value="#D0b65c20" />
  <Setter Property="FontSize" Value="30" />
  <Setter Property="Margin" Value="0" />
</Style>
```

Εικόνα 4.3.9-2

Για να τα χρησιμοποιήσουμε απλά μέσα στις ιδιότητες του textblock στο πεδίο style δίνουμε το αντίστοιχο όνομα (εικόνα 4.3.9-3)

```
<!-- κείμενο με οδηγίες -->
<Grid Margin="0,50,0,0" HorizontalAlignment="Center" VerticalAlignment="Center" >
  <TextBlock x:Name="TapToContinueTextBlock" Style="{StaticResource EtchedTextBack}" Text="Πάτα την εικόνα για να ξεκινήσεις" TextWrapping="Wrap" />
  <TextBlock x:Name="TapToContinueTextBlockFront" Style="{StaticResource EtchedTextFront}" Text="Πάτα την εικόνα για να ξεκινήσεις" TextWrapping="Wrap" />
</Grid>
```

Εικόνα 4.3.9-3

Με αντίστοιχες διαδικασίες που έχουμε αναφέρει και παραπάνω εισάγουμε τα εργαλεία με τις εικόνες. Τέλος θα αναφερθούμε στο κινούμενο banner με το μήνυμα που εμφανίζει με το που ο χρήστης λύσει το ruzzle (εικόνα 4.3.9-4).

```

Storyboard x:Name="WinTransition">
  <DoubleAnimationUsingKeyFrames BeginTime="00:00:00" Storyboard.TargetName="border" Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[0].(ScaleTransform.ScaleX)">
    <EasingDoubleKeyFrame KeyTime="00:00:00" Value="1"/>
    <EasingDoubleKeyFrame KeyTime="00:00:00.7000000" Value="-1">
      <EasingDoubleKeyFrame.EasingFunction>
        <CubicEase EasingMode="EaseInOut"/>
      </EasingDoubleKeyFrame.EasingFunction>
    </EasingDoubleKeyFrame>
    <EasingDoubleKeyFrame KeyTime="00:00:01.7000000" Value="1">
      <EasingDoubleKeyFrame.EasingFunction>
        <CubicEase EasingMode="EaseInOut"/>
      </EasingDoubleKeyFrame.EasingFunction>
    </EasingDoubleKeyFrame>
  </DoubleAnimationUsingKeyFrames>
  <DoubleAnimationUsingKeyFrames BeginTime="00:00:00" Storyboard.TargetName="CongratsBorder" Storyboard.TargetProperty="(UIElement.RenderTransform).(TransformGroup.Children)[3].(TranslateTransform.Y)">
    <EasingDoubleKeyFrame KeyTime="00:00:00" Value="-300"/>
    <EasingDoubleKeyFrame KeyTime="00:00:01" Value="0">
      <EasingDoubleKeyFrame.EasingFunction>
        <CubicEase EasingMode="EaseInOut"/>
      </EasingDoubleKeyFrame.EasingFunction>
    </EasingDoubleKeyFrame>
  </DoubleAnimationUsingKeyFrames>
  <DoubleAnimationUsingKeyFrames BeginTime="00:00:00" Storyboard.TargetName="CongratsBorder" Storyboard.TargetProperty="(UIElement.Opacity)">
    <EasingDoubleKeyFrame KeyTime="00:00:00" Value="0"/>
    <EasingDoubleKeyFrame KeyTime="00:00:01" Value="1"/>
  </DoubleAnimationUsingKeyFrames>
</Storyboard>
Storyboard x:Name="ResetWinTransition">
  <DoubleAnimationUsingKeyFrames BeginTime="00:00:00" Storyboard.TargetName="CongratsBorder" Storyboard.TargetProperty="(UIElement.Opacity)" Duration="00:00:00.0010000">
    <SplineDoubleKeyFrame KeyTime="00:00:00" Value="0"/>
  </DoubleAnimationUsingKeyFrames>
  <DoubleAnimationUsingKeyFrames BeginTime="00:00:00" Duration="00:00:00.0010000" Storyboard.TargetName="GameContainer" Storyboard.TargetProperty="(UIElement.Opacity)">
    <EasingDoubleKeyFrame KeyTime="00:00:00" Value="1"/>
  </DoubleAnimationUsingKeyFrames>
</Storyboard>

```

εικόνα 4.3.9-4

Στην συνέχεια θα αναφερθούμε στο αρχείο **PuzzlePage.xaml.cs** και στις σημαντικότερες από τις ενέργειες που εκτελεί. Αρχικά δηλώνουμε τις μεταβλητές που θα χρησιμοποιήσουμε (εικόνα 4.3.9-5 )

```
using System;
using System.IO;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Media.Imaging;
using System.Windows.Resources;
using Microsoft.Phone.Controls;

namespace WindowsPhonePuzzle
{
    public partial class PuzzlePage : PhoneApplicationPage
    {
        private const double DoubleTapSpeed = 500;
        private const int ImageSize = 435;
        private PuzzleGame game;
        private Canvas[] puzzlePieces;
        private Stream imageStream;

        private long lastTapTicks;
        private int movingPieceId = -1;
        private int movingPieceDirection;
        private double movingPieceStartingPosition;
```

Εικόνα 4.3.9-5

Παρακάτω βλέπουμε πως προσδιορίζονται οι διάφορες καταστάσεις και τις ενέργειες γίνονται όταν ξεκινάει ένα καινούριο παιχνίδι, τερματίζεται ή ανανεώνεται (εικόνα 4.3.9-6).

```

public PuzzlePage()
{
    InitializeComponent();

    SupportedOrientations = SupportedPageOrientation.Portrait | SupportedPageOrientation.Landscape;

    // Παχνίδι παζλ
    this.game = new PuzzleGame(3);

    this.game.GameStarted += delegate
    {
        this.ResetWinTransition.Begin();
        this.StatusPanel.Visibility = Visibility.Visible;
        this.TapToContinueTextBlock.Opacity = 0;
        this.TapToContinueTextBlockFront.Opacity = 0;
        this.TotalMovesTextBlock.Text = this.game.TotalMoves.ToString();
    };

    this.game.GameOver += delegate
    {
        this.WinTransition.Begin();
        this.StatusPanel.Visibility = Visibility.Visible;
        this.TotalMovesTextBlock.Text = this.game.TotalMoves.ToString();

        foreach (Canvas c in this.puzzlePieces)
        {
            PuzzleTile tile = c.Children[0] as PuzzleTile;
            tile.HideNumberLabel();
        }
    };

    this.game.PieceUpdated += delegate(object sender, PieceUpdatedEventArgs args)
    {
        int pieceSize = ImageSize / this.game.ColsAndRows;
        this.AnimatePiece(this.puzzlePieces[args.PieceId], Canvas.LeftProperty, (int)args.NewPosition.X * pieceSize);
        this.AnimatePiece(this.puzzlePieces[args.PieceId], Canvas.TopProperty, (int)args.NewPosition.Y * pieceSize);
        this.TotalMovesTextBlock.Text = this.game.TotalMoves.ToString();
    };

    this.InitBoard();
}

```

Εικόνα 4.3.9-6

Παρακάτω παρουσιάζεται ο ορισμός των διάφορων κινήσεων που γίνονται μέσω της χρήσης του **Manipulation** . Παρακάτω βλέπουμε τον κώδικα για όταν ξεκινάμε (εικόνα 4.3.9-7 )

```

private void PhoneApplicationPage_ManipulationStarted(object sender, ManipulationStartedEventArgs e)
{
    if (this.game.IsPlaying)
    {
        // μετακίνηση στο δέντρο για να βρούμε το κομμάτι του παζλ

        FrameworkElement parent = VisualTreeHelper.GetParent(e.ManipulationContainer) as FrameworkElement;
        while (parent as PuzzleTile == null)
        {
            if (parent == null) return;
            parent = VisualTreeHelper.GetParent(parent) as FrameworkElement;
        }

        if (parent.GetValue(FrameworkElement.NameProperty).ToString().StartsWith("PuzzleImage_"))
        {
            int pieceIx = Convert.ToInt32(parent.GetValue(FrameworkElement.NameProperty).ToString().Substring(12));
            Canvas piece = this.FindName("PuzzlePiece_" + pieceIx) as Canvas;
            if (piece != null)
            {
                int totalPieces = this.game.ColsAndRows * this.game.ColsAndRows;
                for (int i = 0; i < totalPieces; i++)
                {
                    if (piece == this.puzzlePieces[i] && this.game.CanMovePiece(i) > 0)
                    {
                        int direction = this.game.CanMovePiece(i);
                        DependencyProperty axisProperty = (direction % 2 == 0) ? Canvas.LeftProperty : Canvas.TopProperty;
                        this.movingPieceDirection = direction;
                        this.movingPieceStartingPosition = Convert.ToDouble(piece.GetValue(axisProperty));
                        this.movingPieceId = i;
                        break;
                    }
                }
            }
        }
    }
}

```

Εικόνα 4.3.9-7

Στην συνέχεια κατά την διάρκεια της κίνησης (εικόνα 4.3.9-8 )

```

private void PhoneApplicationPage_ManipulationDelta(object sender, ManipulationDeltaEventArgs e)
{
    if (this.movingPieceId > -1)
    {
        int pieceSize = ImageSize / this.game.ColsAndRows;
        Canvas movingPiece = this.puzzlePieces[this.movingPieceId];

        // επικύρωση κίνησης
        DependencyProperty axisProperty;
        double normalizedValue;

        if (this.movingPieceDirection % 2 == 0)
        {
            axisProperty = Canvas.LeftProperty;
            normalizedValue = e.CumulativeManipulation.Translation.X;
        }
        else
        {
            axisProperty = Canvas.TopProperty;
            normalizedValue = e.CumulativeManipulation.Translation.Y;
        }

        // ορίζουμε περιορισμούς
        // (πάνω ή αριστερά )
        if (this.movingPieceDirection == 1 || this.movingPieceDirection == 4)
        {
            if (normalizedValue < -pieceSize)
            {
                normalizedValue = -pieceSize;
            }
            else if (normalizedValue > 0)
            {
                normalizedValue = 0;
            }
        }
        // (κάτω ή δεξιά )
        else if (this.movingPieceDirection == 3 || this.movingPieceDirection == 2)
        {
            if (normalizedValue > pieceSize)
            {
                normalizedValue = pieceSize;
            }
            else if (normalizedValue < 0)
            {
                normalizedValue = 0;
            }
        }

        // ορισμός θέσης
        movingPiece.SetValue(axisProperty, normalizedValue + this.movingPieceStartingPosition);
    }
}

```

Εικόνα 4.3.9-8

και τέλος όταν έχει ολοκληρωθεί η διαδικασία (εικόνα 4.3.9-9) .

```

private void PhoneApplicationPage_ManipulationCompleted(object sender, ManipulationCompletedEventArgs e)
{
    if (this.movingPieceId > -1)
    {
        int pieceSize = ImageSize / this.game.ColsAndRows;
        Canvas piece = this.puzzlePieces[this.movingPieceId];

        // έλεγχος για διπλό πάτημα
        if (TimeSpan.FromTicks(DateTime.Now.Ticks - this.lastTapTicks).TotalMilliseconds < DoubleTapSpeed)
        {
            // εξαναγκασμός κίνησης
            this.game.MovePiece(this.movingPieceId);
            this.lastTapTicks = int.MinValue;
        }
        else
        {
            // υπολογισμός απόστασης που διανήθηκε
            DependencyProperty axisProperty = (this.movingPieceDirection % 2 == 0) ? Canvas.LeftProperty : Canvas.TopProperty;
            double minRequiredDisplacement = pieceSize / 3;
            double diff = Math.Abs(Convert.ToDouble(piece.GetValue(axisProperty)) - this.movingPieceStartingPosition);

            // μετακινήθηκε ως την μέση ?
            if (diff > minRequiredDisplacement)
            {
                // μετακίνηση κομματιού
                this.game.MovePiece(this.movingPieceId);
            }
            else
            {
                // αποθήκευση κομματιού
                this.AnimatePiece(piece, axisProperty, this.movingPieceStartingPosition);
            }
        }

        this.movingPieceId = -1;
        this.movingPieceStartingPosition = 0;
        this.movingPieceDirection = 0;
        this.lastTapTicks = DateTime.Now.Ticks;
    }
}

```

Εικόνα 4.3.9-9.

Ένα ακόμα σημείο που θα πρέπει να κοιτάξουμε είναι ο ορισμός των event που γίνονται στις εικόνες για τις διαδικασίες της **αποθήκευσης** , **φόρτωσης** και **διαγραφής** που παρουσιάζεται στην ακόλουθη εικόνα (εικόνα 4.3.9-10 )



```
private void Save_Game(object sender, MouseButtonEventArgs e)
{
    // αποθηκευση game state
    PuzzleState gameState = this.game.GetState();
    IsolatedStorageHelper.SaveObject("PuzzleState", gameState);
}

private void Load_Game(object sender, MouseButtonEventArgs e)
{
    var gameState = IsolatedStorageHelper.GetObject<PuzzleState>("PuzzleState");
    if (gameState == null)
    {
        MessageBox.Show("Δεν έχεις αποθηκεύσει κάποιο παιχνίδι !!", "Oops!", MessageBoxButton.OK);
    }
    else
    {
        // ορισμος game state
        this.game.SetState(gameState);
    }
}

private void Game_Delete(object sender, MouseButtonEventArgs e)
{
    // αφαίρεση state and image
    IsolatedStorageHelper.DeleteObject("PuzzleState");
}
```

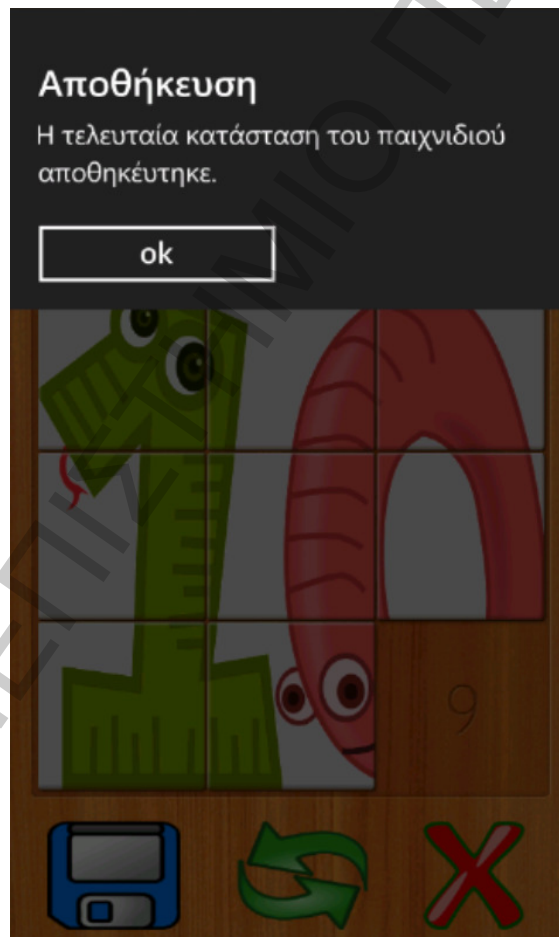
Εικόνα 4.3.9-10

Για να γνωρίζει ο χρήστης ότι έχοντας πατήσει το κουμπί έχει γίνει η ενέργεια που το αντιπροσωπεύει μπορούμε να βάλουμε να εμφανίζονται ενημερωτικά μηνύματα. Αυτό μπορούμε να το κάνουμε με την χρήση της συνάρτησης `MessageBox.Show` (εικόνα 4.3.9-11) .

```
private void Save_Game(object sender, MouseButtonEventArgs e)
{
    // αποθήκευση game state
    MessageBox.Show("Η τελευταία κατάσταση του παιχνιδιού αποθηκεύτηκε.", "Αποθήκευση", MessageBoxButton.OK);
    PuzzleState gameState = this.game.GetState();
    IsolatedStorageHelper.SaveObject("PuzzleState", gameState);
}
```

Εικόνα 4.3.9-11

Στην οποία μπορούμε να περάσουμε ως παράμετρο το μήνυμα που θέλουμε να εμφανίζεται στην οθόνη με το που ο χρήστης πατήσει το κουμπί. Στην προκειμένη περίπτωση ότι έχει αποθηκευτεί η κατάσταση του παιχνιδιού (εικόνα 4.3.9-12). [3]



Εικόνα 4.3.9-12

### 4.3.10 Πλοήγηση

Πριν κλείσουμε θα πρέπει να κάνουμε δύο αλλαγές στην εφαρμογή μας. Όπως αναφέραμε και προηγουμένως όλα τα Windows Phone κινητά έχουν ως προαπαιτούμενο στο hardware να υπάρχει και το κουμπί back button. Το συγκεκριμένο κουμπί όπως αναφέρει και το όνομα του αυτό που κάνει πατώντας το είναι να μας επιστρέφει στην προηγούμενη σελίδα της εφαρμογής ή γενικά του λειτουργικού. Αυτό που κάνει το λειτουργικό είναι κρατάει μια "στοίβα" τύπου **LIFO** με τις σελίδες που έχει περάσει ο χρήστης και όταν πατήσει το back button την επισκέπτεται αλλά με την αντιστροφή σειρά μέχρι να φτάσει στην αρχική σελίδα όπου εκεί πατώντας το back button τερματίζεται η εφαρμογή. Οπότε στη δική μας εφαρμογή όταν ο χρήστης έχει πάει από την MainPage.xaml διαδοχικά στις Page1.xaml -> Page2.xaml -> Page3.xaml πατώντας το back button θα τον οδηγήσει αντίστροφα στις Page3.xaml -> Page2.xaml -> Page1.xaml -> MainPage.xaml για να το αποφύγουμε αυτό μπορούμε σε όλες τις σελίδες με τους αριθμούς Page1.xaml , Page2.xaml κτλ , καθώς και στην Instruction.xaml και PuzzlePage.xaml τον παρακάτω κώδικα. (Εικόνα 4.3.10-1)

```
protected override void OnBackKeyPress(System.ComponentModel.CancelEventArgs e)
{
    this.NavigationService.Navigate(new Uri("/MainPage.xaml", UriKind.Relative));
}
```

Εικόνα 4.3.10-1

Με τον συγκεκριμένο κώδικα παρακάμπτουμε την προκαθορισμένη λειτουργία του back button και όταν το πατάει ο χρήστης τον οδηγεί στην MainPage.xaml και στις άλλες σελίδες με τους αριθμούς μπορεί να πλοηγηθεί με τα κουμπιά-βέλη που έχουμε φτιάξει εμείς.

Το δεύτερο πράγμα που θα πρέπει να κάνουμε είναι τι θα γίνεται όταν ο χρήστης φτάσει στην MainPage.xaml. Όπως είναι τώρα ο κώδικας μας αν ο χρήστης πατήσει το back button χωρίς να έχει επισκεφτεί κάποια άλλη σελίδα της εφαρμογής η στοίβα με τις σελίδες που κρατάει το λειτουργικό είναι άδεια οπότε όπως είπαμε και παραπάνω θα τερματιστεί η εφαρμογή. Αν όμως από την MainPage.xaml έχει πάει στην Page6.xaml και από εκεί είτε με το πλήκτρο "μενού" είτε με το back button στην MainPage.xaml η στοίβα μας θα έχει την παραπάνω σελίδα οπότε ο χρήστης πατώντας το back button δεν θα έχει ως αποτέλεσμα να τερματιστεί η εφαρμογή μας αλλά να πλοηγηθεί στην προηγούμενη σελίδα της στοίβας κάτι το οποίο μπορεί

να μην είναι επιθυμητό αν ο χρήστης έχει επισκεφτεί πολλές σελίδες. Αυτό μπορούμε να το ξεπεράσουμε χρησιμοποιώντας τον παρακάτω κώδικα στην MainPage.xaml (εικόνα 4.3.10-2 )

```
protected override void OnBackKeyPress(System.ComponentModel.CancelEventArgs e)
{
    while (NavigationService.CanGoBack)
        NavigationService.RemoveBackEntry();
}
```

Εικόνα 4.3.10-2

Ο συγκεκριμένος κώδικας κάνει το εξής με το που πατήσει ο χρήστης το back button "αδειάζει" όλες τις καταχωρήσεις που έχουν γίνει στην στοίβα οπότε αυτό έχει ως αποτέλεσμα εφόσον δεν υπάρχει άλλη σελίδα να τερματιστεί η εφαρμογή μας.

Συνεπώς έχουμε ολοκληρώσει την εφαρμογή μας το επόμενο βήμα είναι να την ανεβάσουμε στο **Marketplace** των **Windows Phone** που μας παρέχει η **Microsoft** για να την διαθέσουμε στους χρήστες του λειτουργικού . Γενικά οι εφαρμογές μπορούν να είναι είτε χρεώσιμες είτε δωρεάν στις οποίες όμως έχουμε την δυνατότητα να προβάλλουμε διαφημιστικά μηνύματα διαφόρων προϊόντων με αποτέλεσμα να έχουμε και κάποιο έσοδο .[2]

## 4.4 Διαφημίσεις

### 4.4.1 Γενικά

Όπως αναφέραμε και προηγουμένως μπορούμε να έχουμε κάποια έσοδα από την εφαρμογή παρόλο που την παρέχουμε δωρεάν τοποθετώντας σε συγκεκριμένα σημεία της, που ορίζουμε εμείς διαφημιστικά μηνύματα . Τα διαφημιστικά μηνύματα μπορούν να έχουν συγκεκριμένα θέματα που ορίζουμε εμείς αλλά χωρίς την δυνατότητα να επιλέξουμε συγκεκριμένες εταιρείες, για παράδειγμα μπορούμε να επιλέξουμε προϊόντα πληροφορικής χωρίς να μπορούμε να επιλέξουμε ποιες εταιρείες θέλουμε να εμφανίζονται παρά μόνο όσες θέλουμε να μην εμφανίζονται μόνο . [12]

Το πρώτο πράγμα που θα πρέπει να κάνουμε είναι να κατεβάσουμε και να εγκαταστήσουμε το **Microsoft Advertising SDK for Windows Phone** το οποίο μας παρέχει τα εργαλεία που χρειαζόμαστε για να μπορέσουμε να εισάγουμε τις διαφημίσεις . Το συγκεκριμένο διατίθεται δωρεάν από την Microsoft και μπορούμε να το κατεβάσουμε από εδώ <http://www.microsoft.com/en-us/download/details.aspx?id=8729> .

Επόμενο βήμα είναι να γραφτούμε σε μια υπηρεσία που παρέχει διαφημίσεις για **Windows Phone** , εμείς έχουμε χρησιμοποιήσει την **PubCenter** της **Microsoft** . Θα πρέπει να δημιουργήσουμε έναν νέο λογαριασμό αρχικά στην υπηρεσία , εκτός αν είμαστε ήδη κάτοχοι κάποιου **windows live ID** , οπότε μπορούμε να κάνουμε χρήση κατευθείαν του συγκεκριμένου χωρίς κάποια εγγραφή. [12]

## 4.4.2 Δημιουργία διαφημιστικής μονάδας

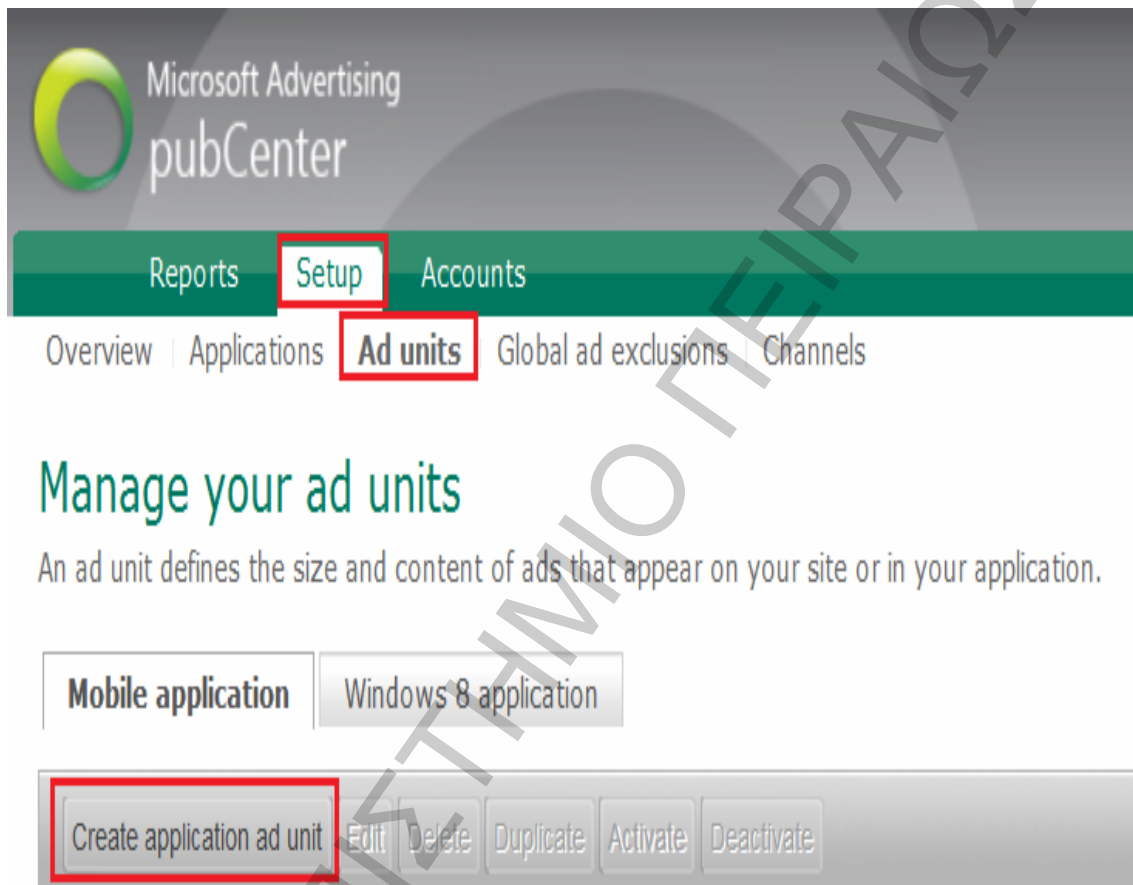
Εφόσον έχουμε εγγραφεί και συνδεθεί θα πρέπει να εγγράψουμε και την εφαρμογή μας αυτό γίνεται επιλέγοντας **Setup** -> **Applications** -> **Register Application** (εικόνα 4.4.2-1 ) με το που ολοκληρωθεί η εγγραφή θα μας βγάλει ένα επιβεβαιωτικό μήνυμα που θα αναφέρει και τον μοναδικό κωδικό της εφαρμογής μας .



Εικόνα 4.4.2-1

Σε κάθε διαφήμιση μπορούμε να έχουμε από μία έως περισσότερα διαφημιστικές μονάδες, όπου σε κάθε μονάδα ορίζουμε το μέγεθος και το είδος των διαφημιστικών μηνυμάτων που θέλουμε να εμφανίζει . Ακολουθώντας μια παρόμοια διαδικασία με παραπάνω δημιουργούμε

για μια διαφημιστική μονάδα επιλέγοντας **Setup -> Ad Units -> Create Application Ad Unit** (εικόνα 4.4.2-2)



Εικόνα 4.4.2-2

Στο νέο παράθυρο που θα εμφανιστεί ορίζουμε το **όνομα** της , το **μέγεθος** του διαφημιστικού παραθύρου ( υποστηρίζονται δύο μεγέθη ) , την **κατηγορία και υποκατηγορία** της διαφήμισης καθώς και κάποια **site** που θέλουμε να αποκλείσουμε ( κυρίως ανταγωνιστών ) από το να προβάλλουν διαφημίσεις στην μονάδα μας (εικόνα 4.4.2-3 ). Ολοκληρώνοντας την διαδικασία θα μας ενημερώσει και με το κωδικό αριθμό της διαφημιστικής μας μονάδας .

**pubCenter**

Reports | **Setup** | Accounts

Overview | Applications | **Ad units** | Global ad exclusions | Channels

### Create a mobile application ad unit

A mobile application ad unit defines the size and types of ads that appear in your Windows Phone 7 application.

**Ad unit name:**

**Ad unit size:**   
 480x80 XX-Large banner   
 480x80 XX-Large banner   
 300x50 X-Large banner

**Note:** The ad unit size that you have selected is shown in a sample location of a device. You define the location of your ad unit in your application code.

**Ad categories:**

Select up to three categories from the tier 1 and tier 2 categories. Click a tier 1 name to see the tier 2 categories within it. These categories set the general guidelines for what types of ads appear in your ad unit. Some ads might apply to more than one category or otherwise appear to be out of place.

Tier 1 categories	Tier 2 categories	Selected categories
<input type="checkbox"/> <b>Arts &amp; Entertainment (7)</b> <input type="checkbox"/> Automotive (23) <input type="checkbox"/> Business (12) <input type="checkbox"/> Careers (11) <input type="checkbox"/> Education (15) <input type="checkbox"/> Family & Parenting (9) <input type="checkbox"/> Food & Drink (18) <input type="checkbox"/> Health & Fitness (45) <input type="checkbox"/> Hobbies & Interests (31)	<input type="checkbox"/> Books & Literature <input type="checkbox"/> Celebrity Fan/Gossip <input type="checkbox"/> Fine Art <input type="checkbox"/> Humor <input type="checkbox"/> Movies <input type="checkbox"/> Music <input type="checkbox"/> Television	

**Excluded URLs:**

You can exclude ads that link to webpages on specific URLs. For example, you might want to exclude ads that lead to your competitors' sites. If you add a top-level domain all webpages under that domain will be excluded.

**Examples:**

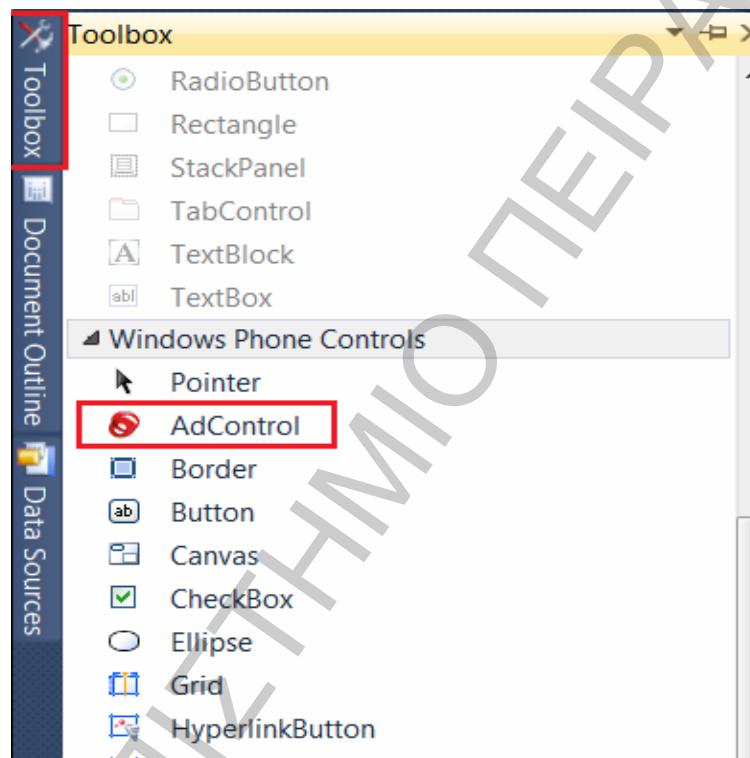
```
www.contoso.com
http://www.contoso.com
http://www.contoso.com/area3
http://subdomain.contoso.com/Area12
```

250 excluded URLs maximum, one per line

Εικόνα 4.4.2-3



Το μόνο που απομένει είναι να συνδέσουμε την εφαρμογή με την διαφήμιση . Για να γίνει αυτό θα χρησιμοποιήσουμε το εργαλείο **AdControl** το οποίο είναι διαθέσιμο από την στιγμή που εγκαταστήσουμε το **Microsoft Advertising SDK for Windows Phone** . Αφού έχουμε διαλέξει την σελίδα και το μέρος που θέλουμε να εμφανίζεται το διαφημιστικό μας μήνυμα κάνουμε drag and drop το εργαλείο **AdControl** από το **Toolbox** (εικόνα 4.4.2-4 ) .



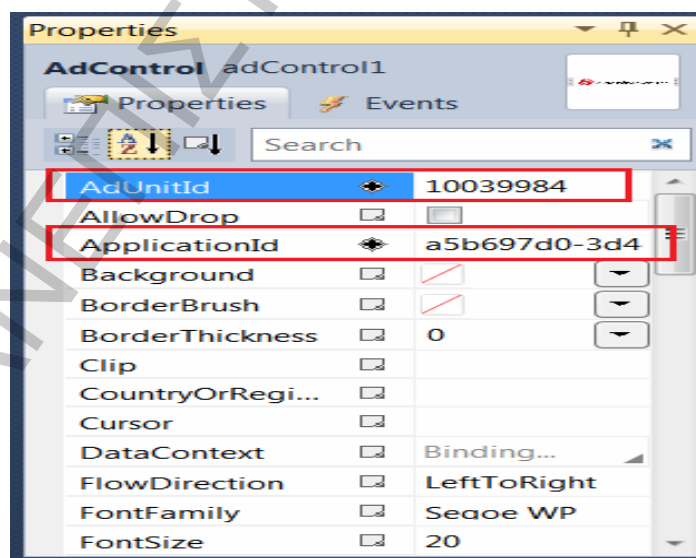
Εικόνα 4.4.2-4

οπότε πλέον στην εφαρμογή μας θα εμφανίζεται ένα πεδίο που θα αναφέρει **Microsoft Advertising** (εικόνα 4.4.2-5 )



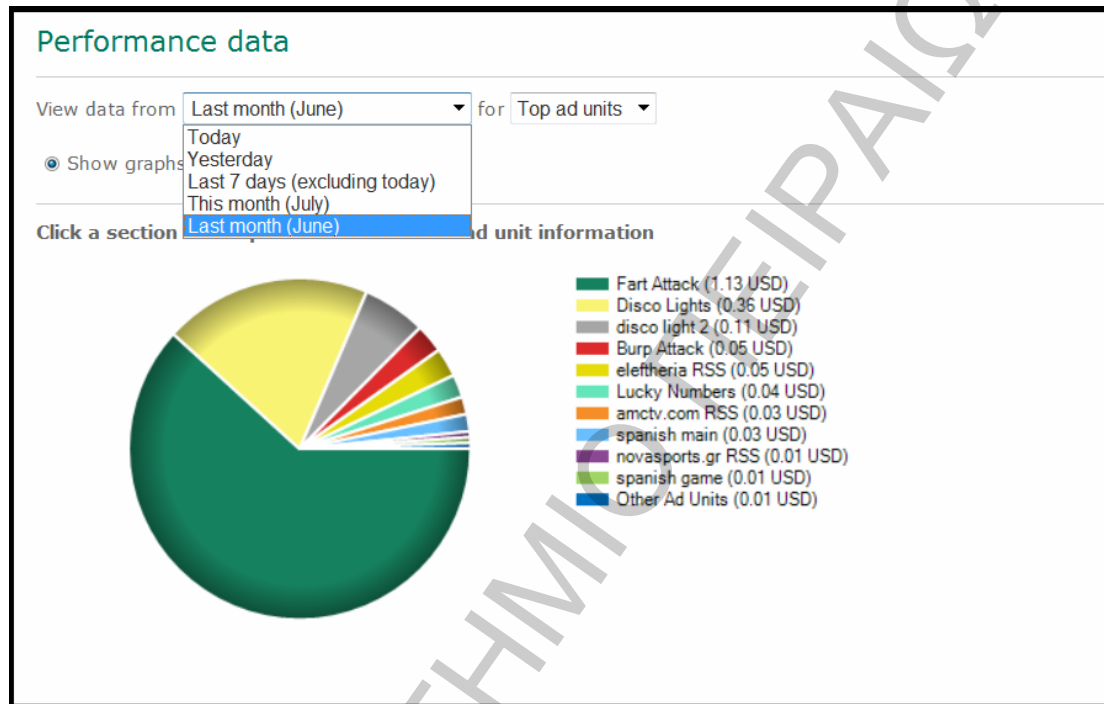
Εικόνα 4.4.2-5

Όπου έχοντας το επιλεγμένο μπορούμε να ορίσουμε τις ιδιότητες του από το παράθυρο **Properties** μεγάλη προσοχή πρέπει να δώσουμε στα πεδία **ApplicationId** και **AdUnitId** όπου θα πρέπει να ορίσουμε σωστά τις αντίστοιχες συνιστώσες που έχουμε λάβει από το **PubCenter** καθώς και στα πεδία **Width** και **Height** για να ορίσουμε τα σωστά μεγέθη για να μπορεί να προβάλλεται το διαφημιστικό μήνυμα χωρίς προβλήματα (εικόνα 4.4.2-6) .



Εικόνα 4.4.2-6

Τέλος από το πεδίο **reports** μπορούμε να δημιουργήσουμε διάφορες αναφορές σχετικά με τις διαφημιστικές μονάδες που έχουμε δημιουργήσει και από το **Overview** να έχουμε μια αναλυτική εικόνα των εσόδων σε διάφορα χρονικά πλαίσια και με διαγράμματα ή πίνακες (εικόνα 4.4.2-7) . [12]



Name	Impressions	Estimated Revenue	eCPM
Account totals		3,517	1.83
Fart Attack		2,246	1.13
Disco Lights		597	0.36
disco light 2		360	0.11
Burp Attack		139	0.05
eleftheria RSS		6	0.05
Lucky Numbers		97	0.04
amctv.com RSS		12	0.03
spanish main		23	0.03
novasports.gr RSS		17	0.01
spanish game		6	0.01

Εικόνα 4.4.2-7

## 4.5 Marketplace

### 4.5.1 Εγγραφή στο Marketplace

Όπως αναφέραμε η Microsoft μας παρέχει έτοιμη πλατφόρμα για να μπορέσουμε να προωθήσουμε τις εφαρμογές μας . Στην συγκεκριμένη ενότητα θα δούμε τα απαραίτητα βήματα για εγγραφή στο marketplace, ρύθμιση του τραπεζικού λογαριασμού για paid εφαρμογές αλλά και τα βήματα που πρέπει να κάνουμε για να ανεβάσουμε την εφαρμογή μας στο marketplace. [10]

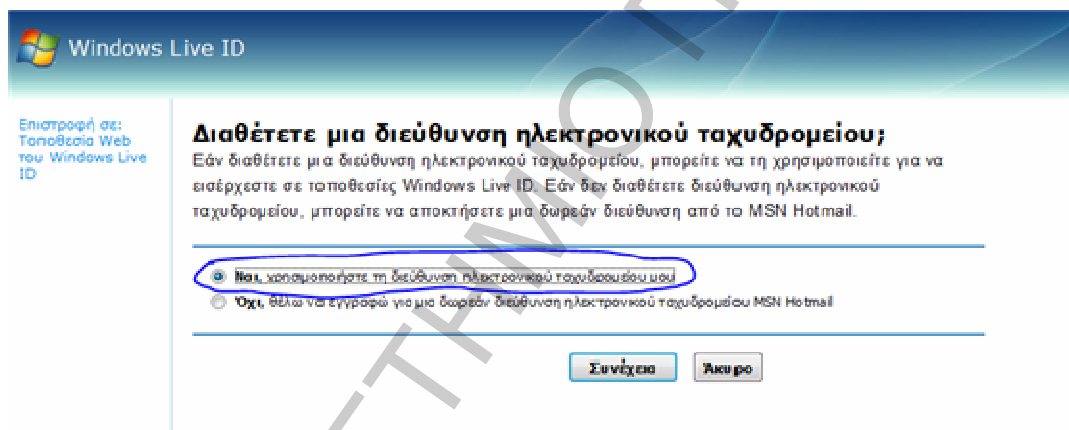
#### Βήμα 1 - Create Live ID

Κάθε εκπαιδευτικό ίδρυμα έχει για τους φοιτητές του ένα μοναδικό αναγνωριστικό με το οποίο γίνεται και η πιστοποίηση τους για την εγγραφή στο App Hub και στο δωρεάν λογισμικό από το [www.dreamspark.com](http://www.dreamspark.com) . Τα περισσότερα ΑΤΕΙ χρησιμοποιούν το σύστημα "dionysos", το email που δηλώσατε στο σύστημα αυτό είναι το μοναδικό αναγνωριστικό σας. Τα εκπαιδευτικά ιδρύματα που δεν χρησιμοποιούν το "dionysos" δίνουν στους φοιτητές λογαριασμό email με τον τομέα τους ως αναγνωριστικό. Αν το email σας είναι [someone@hotmail/windowslive/live.com/gr](mailto:someone@hotmail/windowslive/live.com/gr) κλπ τότε είναι ήδη Live ID οπότε προσπεράστε αυτό το βήμα. Αν έχετε email που δεν είναι live id θα πρέπει να το ρυθμίσετε ως εξής.

- Πηγαίνετε στην διεύθυνση [www.passport.net](http://www.passport.net) .
- Πατάτε "εγγραφή" πάνω αριστερά (εικόνα 4.5.1-1) .
- Πατάτε "Ναι χρησιμοποιείτε την διεύθυνση ηλεκτρονικού ταχυδρομείου μου" (εικόνα 4.5.1-2) .
- Συνεχίστε βάζοντας όποια στοιχεία σας ζητηθούν.



Εικόνα 4.5.1-1



Εικόνα 4.5.1-2

## Βήμα 2 - Register in App Hub

- Πηγαίνετε στο **create.msdn.com** .
- Κάνετε sign in με το Live ID σας.
- Θα εμφανιστεί μία φόρμα εγγραφής. Εκεί επιλέγετε χώρα, στο account type επιλέγεται "**Student**" ώστε να μην πληρώσετε τα 99\$ για την εγγραφή και κάνετε check στους όρους που υπάρχουν στο τέλος (εικόνα 4.5.1-3) .

## Account Registration

The screenshot shows the 'Account Registration' process. On the left, a navigation menu includes 'choose account' (selected), 'account details', 'my profile', 'payment', and 'confirmation'. The main content area is titled 'account type' and contains the following fields:

- Country of residence or business:** A dropdown menu labeled 'Select a country/region'.
- Account type:** Three radio button options:
  - Company:** Select if you're registering as a business.
  - Individual:** Select if you're registering as an individual developer.
  - Student:** Select if you are registering as a student. *Notes:* requires successful verification of your student status through [DreamSpark](#). This option is highlighted with a blue circle.
- Terms of Use for App Hub:** A checkbox labeled 'By checking this box, I agree to be bound by the following legal Terms of Use: App Hub Terms of Use, Xbox LIVE Terms of Use, Windows Phone Marketplace Application Provider Agreement, DreamSpark Addendum (if you are a student licensed under the DreamSpark program), Windows Phone Marketplace Certification Vendor Agreement, and App Hub Submission Agreement for Xbox 360. Further, if accepting on behalf of a company, then I represent that I am authorized to act on my company's behalf.'

At the bottom right, there are 'Cancel' and 'I Accept' buttons.

Εικόνα 4.5.1-3

- Το σύστημα θα σας ειδοποιήσει ότι η πιστοποίησή σας μέσω **dreamspark** απέτυχε αν δεν το έχετε χρησιμοποιήσει ποτέ στο παρελθόν. Εκεί πατήστε στο link "**Visit Dreaspark now to get verified**" (εικόνα 4.5.1-4) .

The screenshot shows the 'APP HUB' logo and the text 'DEVELOP FOR WINDOWS PHONE & XBOX 360'. Below it, the page is titled 'account creation'. On the left, a navigation menu includes 'choose account' (selected), 'account details', 'my profile', 'payment', and 'confirmation'. The main content area displays an error message:

**we weren't able to validate your dreamspark account.**

Unfortunately, we weren't able to verify that you're a student with a DreamSpark account. This could be caused by two reasons:

- You haven't completed the DreamSpark verification process. If not, getting verified is simple and won't take more than a few minutes. [Visit DreamSpark now to get verified.](#)
- You aren't signed in with the same Windows Live ID that you used to verify your student status on DreamSpark. Try signing in again with the same Live ID that you used to set up your DreamSpark account.

After you've successfully verified your student status on DreamSpark or signed in with the correct Live ID, try registering here again.  
If you're not a student, you'll need to register as an individual or a business to continue. [Go back to registration.](#)

Εικόνα 4.5.1-4

- Μετά το Βήμα 3 επιστρέψτε εδώ. Στα επόμενα βήματα θα σας ζητηθούν κάποιες πληροφορίες όπως το όνομά σας κλπ. Γράψτε αληθινά στοιχεία με λατινικούς χαρακτήρες. Στο πεδίο του ονόματος μην βάλετε ψευδώνυμο πιο μετά θα σας ζητηθεί να επιλέξετε ψευδώνυμο για τον λογαριασμό σας.

### Βήμα 3 - Getting verified through Dreamspark

- Επιλέξτε χώρα .
- Επιλέξτε "**Verify as a Student**".
- Get verified through my school .
- Βρείτε το εκπαιδευτικό ίδρυμα σας .
- Εισάγετε το μοναδικό σας αναγνωριστικό πχ **username/email** και **password**
- Αν όλα πήγαν καλά θα εμφανιστεί μήνυμα επιβεβαίωσης.

### Βήμα 4 - Setting Up Bank Account

- Αφού τελειώσετε με την εγγραφή πηγαίνετε στο **create.msdn.com** και κάντε **sign in**.
- Πάνω δεξιά κάντε κλικ στο όνομά σας.
- Στην σελίδα που θα εμφανιστεί κάντε κλικ στην καρτέλα "**payout setup**".
- Θα εμφανιστεί η ακόλουθη φόρμα (εικόνα 4.5.1-5) .

payout setup

Bank account holder name:

Bank name:

Branch name:

Optional

Address:

City:

Country/Region:

BIC / SWIFT:

IBAN:

VAT identification number:

Optional

**Certification Instructions:**  
You must advise us if you have been notified by the IRS that you are currently subject to back up withholding.

Εικόνα 4.5.1-5

- Πάρτε το βιβλιάριο του λογαριασμού σας και συμπληρώστε τα στοιχεία όπως αναγράφονται.

#### Βήμα 5 - Sending W-8 Form

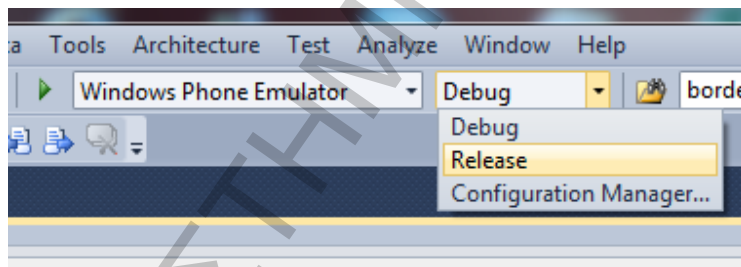
- Κατεβάσετε το pdf από εδώ [www.irs.gov/pub/irs-pdf/fw8ben.pdf](http://www.irs.gov/pub/irs-pdf/fw8ben.pdf)
- Συμπληρώνετε τα πεδία 1,2,3 και 4 από το Part I. Στο 1 βάζετε το όνομά σας όπως το γράψατε κατά την εγγραφή σας στο app hub. Στο 2 γράφετε "N/A". Στο 3 επιλέγετε "Individual" και στο 4 διεύθυνση και χώρα.
- Εκτυπώνεται το συμπληρωμένο έγγραφο και στο τέλος βάζετε ημερομηνία και υπογραφή "ΧΕΙΡΟΓΡΑΦΑ".
- Σκανάρετε το έγγραφο και στείλτε το με email στην διεύθυνση [mplcomm@microsoft.com](mailto:mplcomm@microsoft.com)



## 4.5.2 Ανέβασμα Εφαρμογής

### Βήμα 1 - Testing your app[10]

- Ανοίξτε το app στο **visual studio**
- Αλλάξτε το **Debug** προφίλ σε **Release** και πατήστε **Build Solution** (εικόνα 4.5.2-1) .
- Πατήστε **Project -> Open Marketplace Test Kit**.
- Προσθέστε τις εικόνες του app σας και στην καρτέλα "**Automated Tests**" πατήστε "**Run Tests**".
- Διορθώστε τυχόν λάθη.



Εικόνα 4.5.2-1

### Βήμα 2 - Uploading your app

- Πηγαίνετε στο **create.msdn.com** και κάντε **sign in**.
- Πατήστε "**Submit for Windows Phone**".
- Εμφανίζεται η παρακάτω φόρμα (εικόνα 4.5.2-2) .
- Στο πεδίο "**App info**" συμπληρώνουμε το **όνομα** της εφαρμογής , το **κόστος** και άλλες σχετικές πληροφορίες.

- Στο πεδίο **"Upload and describe XAP file"**, βάζουμε την **περιγραφή** , **λέξεις-κλειδια** για αναζήτηση της εφαρμογής μας στο marketplace καθώς και **screenshots** από την εφαρμογή μας.
- Τέλος στα πεδία **"add in-App advertising"** και **"market selection and custom pricing"** μπορούμε να ορίσουμε να εμφανίζονται διαφημίσεις στην εφαρμογή μας καθώς και σε ποια market θα είναι διαθέσιμη.

Windows Phone | Dev Center

Design Develop Publish Community **Dashboard**

## Submit app

You've spent hours developing and designing your app, and now it's time for the rest of the world to experience your masterpiece. In just two steps we'll gather the information we need to successfully launch your app in the Windows Phone Store. [Learn more](#) about the steps for successfully submitting your app.

### Required

- 1** **App info**  
Give your app an alias, price it, and enter other relevant info
- 2** **Upload and describe your XAP(s)**  
For each XAP in your app, this is where you'll enter descriptions and upload screenshots that will showcase your app in the Store.

### Optional

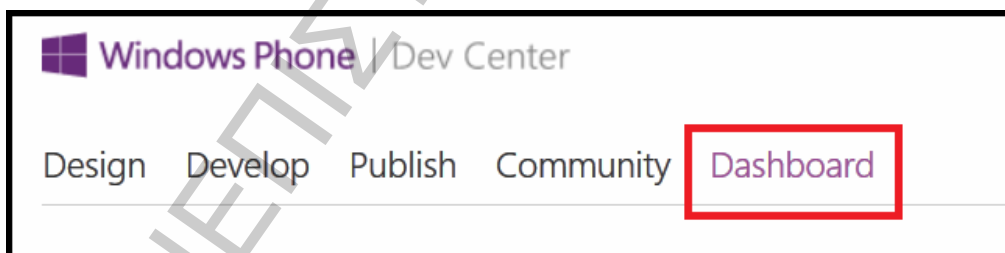
- Add in-app advertising**  
Getting paid through ads? It's all here.
- Market selection and custom pricing**  
For apps, you have the option to define different pricing and availability for different countries/regions.

Εικόνα 4.5.2-2

### 4.5.3 Έλεγχος εφαρμογών ( Dashboard )

Έχοντας ανεβάσει την εφαρμογή μας στο Marketplace δεν σημαίνει ότι έχουμε τελειώσει, θα πρέπει πάντα να λαμβάνουμε υπόψη μας τα σχόλια και κριτικές που μας κάνουν οι χρήστες του Marketplace έτσι ώστε αν υπάρχει κάποιο πρόβλημα που δεν έχουμε προσέξει ( bug ) να είμαστε σε θέση να ανεβάσουμε αναβαθμισμένες εκδόσεις της εφαρμογής μας.

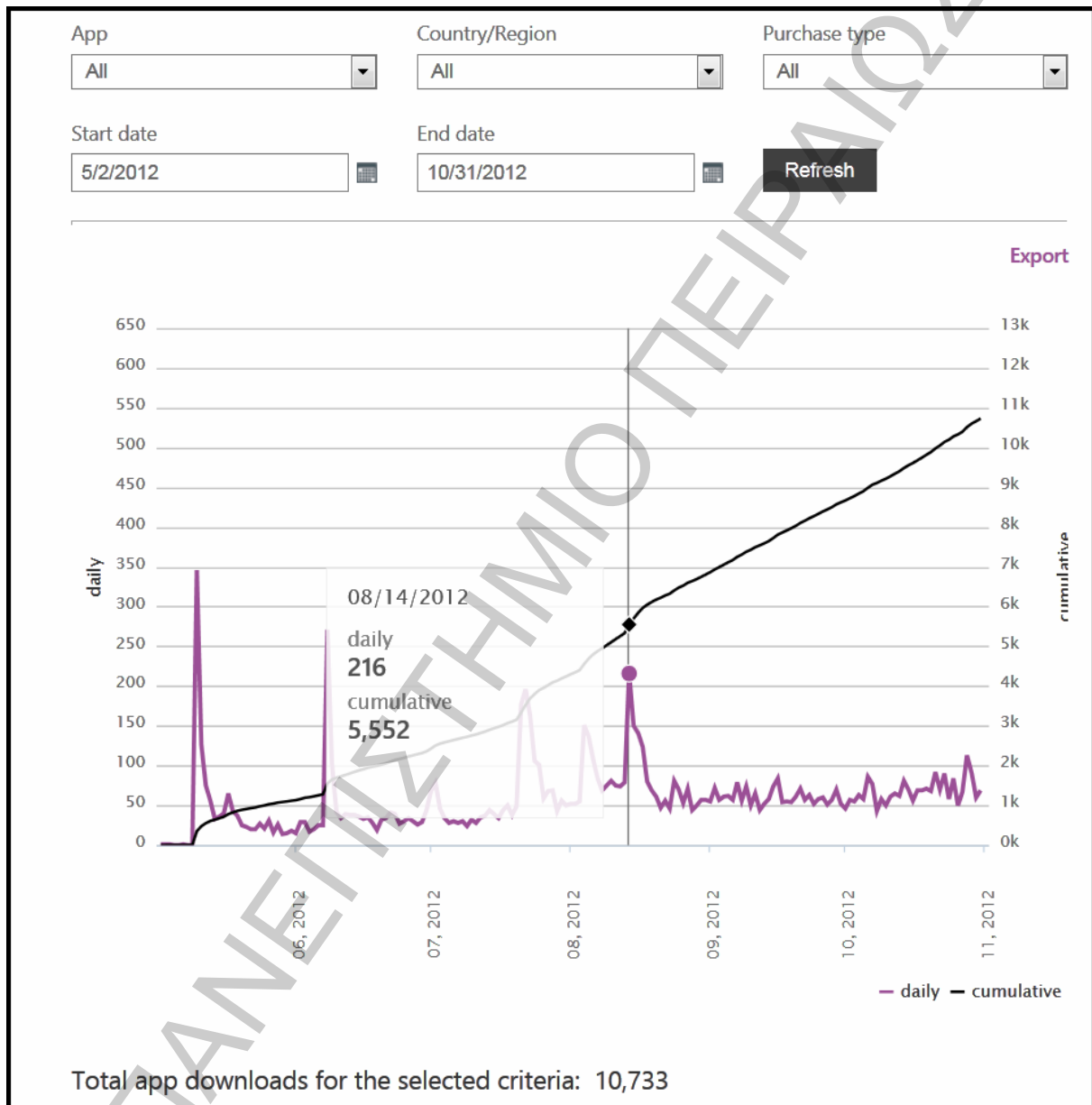
Η Microsoft μας παρέχει πλήρη έλεγχο των εφαρμογών μας μέσα από το **MSDN**. Αφού έχουμε κάνει log in με τα στοιχεία μας στο <http://msdn.microsoft.com/> επιλέγουμε το πεδίο που εμφανίζεται για τα Windows Phone και στη συνέχεια το Dashboard (εικόνα 7.3-1).



εικόνα 4.5.3-1

Από εκεί μπορούμε να λάβουμε αρκετά στοιχεία σχετικά με τις εφαρμογές συγκεκριμένα στο πεδίο "**My money**" μπορούμε να δούμε τα λεφτά που έχουμε αποκτήσει από τις εφαρμογές που είναι επί πληρωμή, στο πεδίο "**In-app purchases**" το πόσες εφαρμογές μας έχουν

αγοράσει και τέλος στο πεδίο "**Reports**" μπορούμε να δούμε τα συνολικά μας **downloads** (εικόνα 4.5.3-2) ή για κάθε εφαρμογή ξεχωριστά (εικόνα 4.5.3-3) καθώς και τα **crashes** που μπορεί να έχουμε .



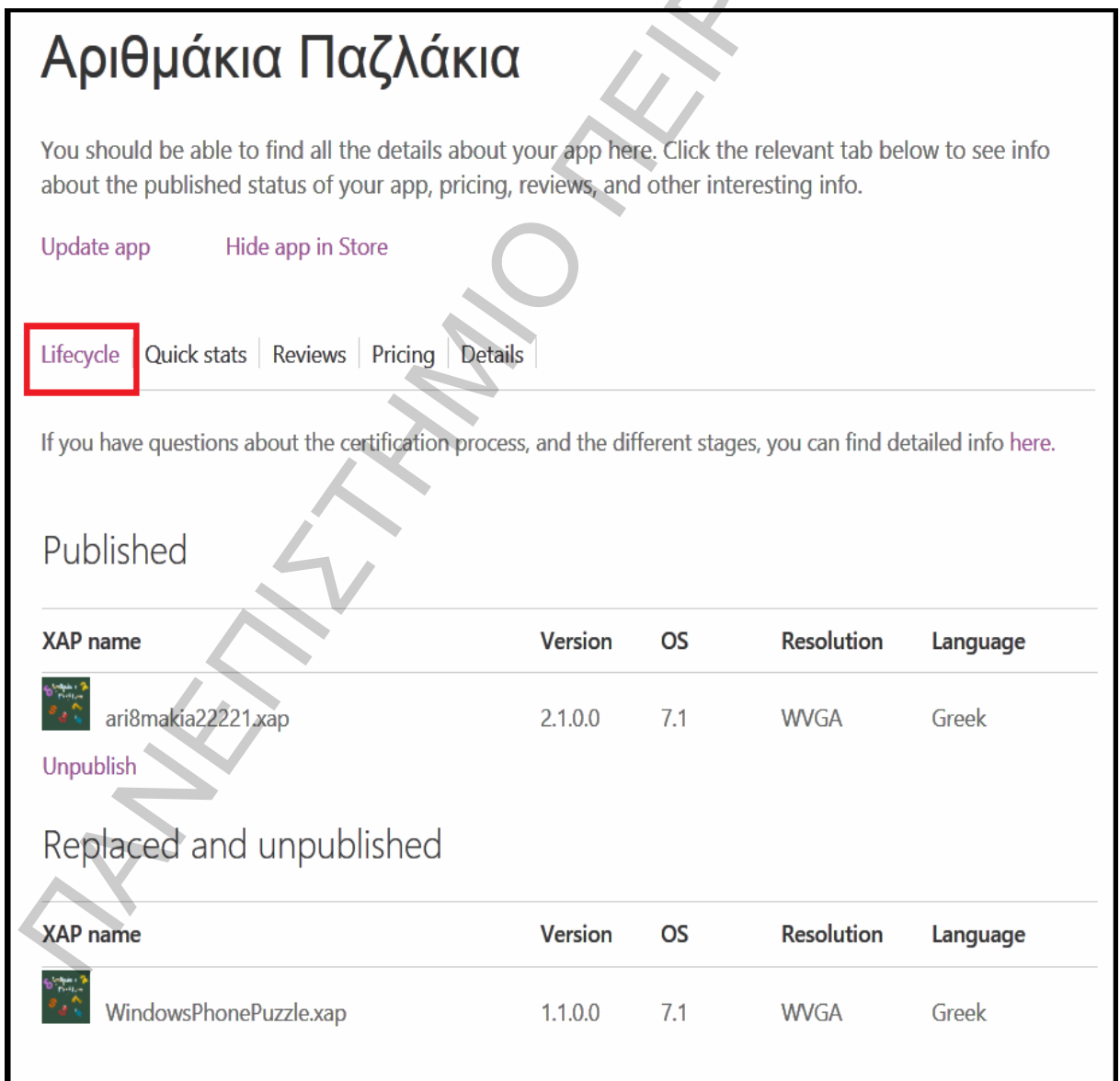
εικόνα 4.5.3-2

App	Total downloads
Disco Lights	4,179
Charlize Theron	918
Moral Stories for Kids	899
novasports.gr RSS	850
Fart Attack	783
Números y Rompecabezas	488
myphone.gr RSS	461
insomnia.gr RSS	451
amctv.com RSS	230
Lucky Numbers	202
Αισώπου Μύθοι	193
numero e puzzle	143
adslgr.com RSS	142
Red Planet RSS	132
Numbers & Puzzles	123
Burp Attack	106
Vasilis Papakonstantinou	102
Αριθμάρια Παζλάκια	98
e-pcmag.gr RSS	88
sport.gr RSS	46
trypes	29

εικόνα 4.5.3-3

Τέλος επιλέγοντας την κάθε εφαρμογή ξεχωριστά μας εμφανίζονται πέντε πεδία όπου από το καθένα μπορούμε να αντλήσουμε χρήσιμες πληροφορίες. Συγκεκριμένα τα πεδία είναι τα ακόλουθα.

- **Lifecycle** : Εδώ μπορούμε να δούμε τον κύκλο ζωής της εφαρμογής μας δηλαδή ποια έκδοση είναι ενεργή καθώς και να "κατεβάσουμε" ή να "αναβαθμίσουμε" μια εφαρμογή (εικόνα 4.5.3-4)



**Αριθμάκια Παζλάκια**

You should be able to find all the details about your app here. Click the relevant tab below to see info about the published status of your app, pricing, reviews, and other interesting info.

[Update app](#)   [Hide app in Store](#)

**Lifecycle** | [Quick stats](#) | [Reviews](#) | [Pricing](#) | [Details](#)

If you have questions about the certification process, and the different stages, you can find detailed info [here](#).

### Published

XAP name	Version	OS	Resolution	Language
 ari8makia22221.xap	2.1.0.0	7.1	WVGA	Greek

[Unpublish](#)

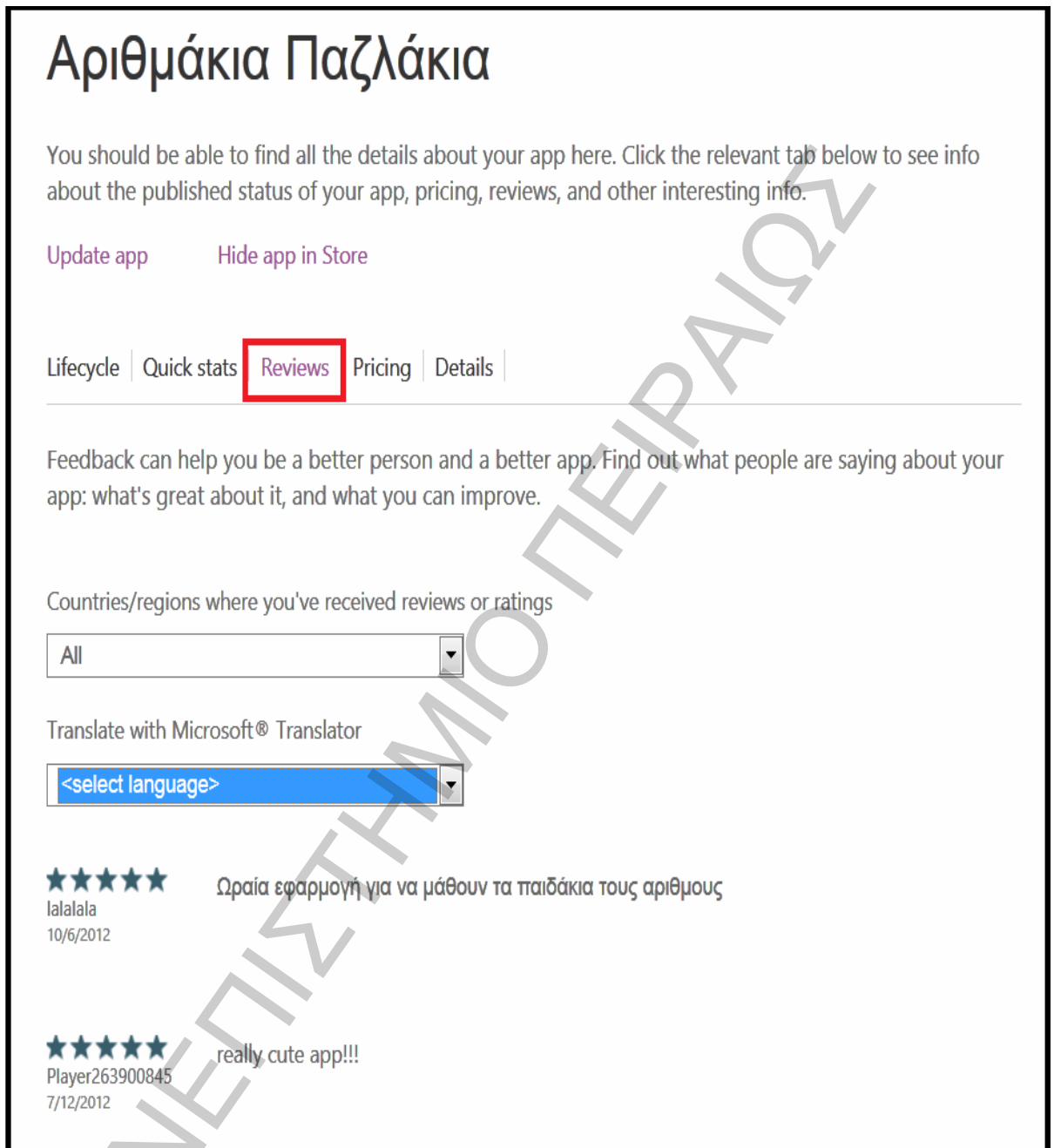
### Replaced and unpublished

XAP name	Version	OS	Resolution	Language
 WindowsPhonePuzzle.xap	1.1.0.0	7.1	WVGA	Greek

## Εικόνα 4.5.3-4

- Quick stats : Εδώ εμφανίζονται μέσω διαγραμμάτων τα **downloads** και τα **crashes** της εφαρμογής μας.
- Reviews : Περιλαμβάνονται τα review των εφαρμογών από διάφορους χρήστες τόσο συνολικά όσο και σε κάθε χώρα ξεχωριστά (εικόνα 4.5.3-5) . Τέλος μας δίνεται η δυνατότητα να τα μεταφράσουμε με τον Microsoft Translator.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΝ



εικόνα 4.5.3-5

- **Pricing** : Μας εμφανίζει το κόστος της εφαρμογής μας στα marketplace ανά τον κόσμο.
- **Details** : Εμφανίζονται αναλυτικά όλες οι πληροφορίες της εφαρμογής δηλαδή από την περιγραφή που έχουμε δώσει στην εφαρμογή μας , τις λέξεις-κλειδιά που έχουμε δηλώσει στην αναζήτηση της , την κατηγορία στην οποία ανήκει μέχρι και το μοναδικό της ID και το link της στο Marketplace.



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

## 5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Όπως αναφέραμε και νωρίτερα ο κόσμος της τεχνολογίας είναι ταχεία αναπτυσσόμενος και με πολλές αλλαγές. Πολλοί είναι κατά της χρήσης της τεχνολογίας σε διάφορους τομείς με ένα από αυτούς να είναι και η εκπαίδευση. Από την άλλη είναι αρκετοί οι οποίοι υποστηρίζουν ότι με την σωστή χρήση της έχει να δώσει πάρα πολλά και να αλλάξει τον τρόπο της παραδοσιακής εκπαίδευσης προς το καλύτερο. Δημιουργώντας την συγκεκριμένη εφαρμογή προσπαθήσαμε να δείξουμε όχι μόνο την χρησιμότητα της τεχνολογίας αλλά και την αναγκαιότητα της χρήσης της πλέον στην εκπαίδευση για να μπορέσει, όχι μόνο ουσιαστικά να εξελίξει τον τρόπο διδασκαλίας, αλλά κυρίως για να προσελκύσει την προσοχή των νέων.

Από την μέχρι τώρα πορεία της εφαρμογής μπορούμε να πούμε ότι ως προσπάθεια είναι αρκετά επιτυχημένη, καθώς σε ένα δύσκολο πεδίο όπως είναι αυτός της τεχνολογίας και ειδικά των κινητών τηλεφώνων με την χρησιμοποίησή τους από παιδιά έδειξε ότι αν υπάρχει

διάθεση και σωστή παροχή των πληροφοριών αλλά και χρησιμοποίηση του μέσου το ευρύ κοινό είναι έτοιμο να δοκιμάσει και να χρησιμοποιήσει το "διαφορετικό".

Μέχρι στιγμής η εφαρμογή μας αριθμεί πάνω από 2000 downloads μέσα σε σύντομο χρονικό διάστημα στις τέσσερις γλώσσες που είναι προς το παρόν διαθέσιμη (Ελληνικά , Αγγλικά , Ιταλικά , Ισπανικά). Ως αριθμός είναι αρκετά ικανοποιητικός βάσει και του ότι έχει αναπτυχθεί για ένα καινούριο λειτουργικό τα Windows Phone που ακόμα είναι στο ξεκίνημα τους και κατέχουν ουσιαστικά μόνο το 4% της σχετικής "πίτας" από τα mobile λειτουργικά.

Πέραν του αριθμού των download θετικό feedback λαμβάνουμε και από τις σχετικές αξιολογήσεις που γίνονται από τους ήδη χρήστες της εφαρμογής αλλά και από ανεξάρτητα site που ασχολούνται με review εφαρμογών για Windows Phone όπως είναι το wrapps.gr , στο οποίο έχουμε αποσπάσει μια αρκετά καλή κριτική.

Το πρώτο μήνυμα από την ανταπόκριση των χρηστών είναι θετικό , αλλά σίγουρα δεν φτάνει μόνο αυτό. Όπως αναφέραμε η τεχνολογία εξελίσσεται γρήγορα οπότε αν θέλουμε να διατηρήσουμε το ενδιαφέρον των χρηστών αντίστοιχα θα πρέπει να εξελίσσουμε και εμείς την εφαρμογή μας. Ως εφαρμογή σίγουρα έχουμε δημιουργήσει ένα αρκετά ισχυρό κορμό ο οποίος με διάφορες επεκτάσεις μπορεί να αναβαθμίσει την εφαρμογή μας και να προσφέρει και νέα επίπεδα εμπειρίας στο τελικό χρήστη. Από τις πιο εύκολες επεκτάσεις θα ήταν η χρησιμοποίηση μεγαλύτερου εύρους αριθμών αλλά και η χρησιμοποίηση των βασικών αριθμητικών πράξεων ( πρόσθεση , αφαίρεση , πολλαπλασιασμός , διαίρεση ) όσο αφορά το κομμάτι της εκμάθησης . Σχετικά με το ψυχαγωγικό κομμάτι θα μπορούσαμε να χρησιμοποιήσουμε περισσότερους αριθμούς οι οποίοι θα είναι διαθέσιμοι για την επίλυση στο πάζλ , καθώς και νέου τύπου παιχνίδια όπως να ενώσει ο χρήστης τις κουκίδες με σκοπό να σχηματιστεί ο αντίστοιχος αριθμός .

Σίγουρα εκτός από τις παραπάνω επεκτάσεις η συγκεκριμένη εφαρμογή μπορεί να δοθεί ως έναυσμα για την δημιουργία μιας σειράς σε αντίστοιχο τόνο εκπαιδευτικών εφαρμογών καλύπτοντας και άλλους τομείς , όπως η εκμάθηση της αλφαβήτου, των ζώων, των χρωμάτων κτλ.

## - Βιβλιογραφία

- "Beginning Windows Phone 7 Application Development" Nick Levenski, Karli Watson, Robert Fonseca-Ensor [1]

- "101 Windows Phone Apps : Volume 1 Developing Apps 1 -50" Adam Nathan [2]
- "Programming Windows Phone 7" Charles Petzold [3]
- "Microsoft Expression Blend 4 Step by Step" Elena Konsiska and Cris Leeds [4]
- "Introducing Visual C# 2010" Adam Freeman [5]
- "Σημειώσεις διδασκαλίας Τεχνολογία Λογισμικού για τη γλώσσα μοντελοποίησης UML" Μαρία Βίρβου [6]
- "Νέες Τεχνολογίες στην Εκπαίδευση" Τ. Παγγέ, Μ. Κυριαζή [7]
- msdn.microsoft.com [8]
- <http://windowsphonegeek.com> [9]
- <http://studentguru.gr> [10]
- <http://www.windowsphone.com> [11]
- <http://pubcenter.microsoft.com> [12]