



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**“Σχεδιασμός και ανάπτυξη εφαρμογής σε android
για την εκμάθηση της διατροφής
για παιδιά με διαβήτη τύπου Ι”**

**ΓΚΟΛΙΑ ΠΑΡΑΣΚΕΥΗ
ΑΕΜ: 302**

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Αγγελίδης Παντελής – Αναπληρωτής Καθηγητής

Κοζάνη, Μάρτιος 2015

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Παντελή Αγγελίδη για την υποστήριξη και καθοδήγηση που μου πρόσφερε κατά τη διάρκεια της υλοποίησης αυτής της εργασίας.

Ένα μεγάλο ευχαριστώ αξίζει στην οικογένεια μου, και ιδιαίτερα στους γονείς μου, που σταθήκαν δίπλα μου σε όλη τη διάρκεια της προετοιμασίας μου, με υπομονή και αμέριστη συμπαράσταση. Επίσης ευχαριστώ τους φίλους και τους γνωστούς που με τις ιδέες τους και τη βοήθειά τους μπόρεσε αυτή η διπλωματική να γίνει πραγματικότητα.

Τέλος, θα ήθελα να ευχαριστήσω τον ενδοκρινολόγο, κ. Σιδέρη Δελαρούδη, που με βοήθησε και με ενημέρωσε για το διαβήτη και τις ιδιαιτερότητες της συγκεκριμένης ασθένειας.

Περίληψη

Η παρούσα διπλωματική εργασία αποτελεί την αρχή της υλοποίησης ενός πολύ φιλόδοξου μεγαλύτερου project. Σκοπός ήταν να δημιουργηθεί ένα παιχνίδι το οποίο θα προορίζεται για την εκμάθηση της διατροφής των παιδιών με διαβήτη τύπου Ι. Αν και υπάρχουν αρκετές εφαρμογές παρατηρήθηκε ότι καμία δεν υπήρχε στα ελληνικά, οπότε και στόχος ήταν η δημιουργία μιας εφαρμογής που θα μπορέσει να μάθει στα παιδιά την επιτρεπόμενη διατροφή.

Τα προγράμματα που χρησιμοποιήθηκαν για την υλοποίηση αυτού του project, είναι κατά βάση δύο: το Eclipse Luna, με το οποίο δημιουργήθηκε όλο το παιχνίδι και το Autocad 2008, με το οποίο έγινε η δημιουργία και η επεξεργασία κάθε μιας εικόνας που έχει συμπεριληφθεί στην εφαρμογή. Για τον προγραμματισμό, χρησιμοποιήθηκε η αντικειμενοστραφής γλώσσα προγραμματισμού Java.

Τέλος, να αναφερθεί ότι η εφαρμογή σχεδιάστηκε για να παίζει σε οποιοδήποτε συσκευή που διαθέτει Android (κινητά και tablets), αφού πρώτα προσαρμόστηκε κατάλληλα στην ανάλυση κάθε εικόνας.

Περιεχόμενα

1. Παιχνιδοποίηση (Gamification.).....	1
1.1. Ορισμός και σκοπός.....	1
1.2. Το παιχνίδι στην υγειονομική περίθαλψη.....	2
1.3. Μηχανισμοί του Gamification.....	3
1.4. Συμβολή στη βελτίωση της υγείας.....	5
1.5. Mobile εφαρμογές βασισμένες στο Gamification.....	6
2. Διαβήτης.....	8
2.1. Διαβήτης Τύπου Ι.....	9
2.2. Διαβήτης και Διατροφή.....	11
2.2.1. Ισοδύναμες τροφές.....	13
2.3. Παιχνίδια - Εφαρμογές για τον παιδικό διαβήτη.....	14
3. Ενδυνάμωση των ασθενών (Patient Empowerment).....	19
3.1. Ενδυνάμωση σε άτομα με διαβήτη.....	20
3.2. Αυτοδιαχείριση (Self-Management).....	21
4. Android.....	23
4.1. Ιστορική Αναδρομή.....	24
4.2. Αρχιτεκτονική Android.....	25
4.3. Χαρακτηριστικά του Android.....	26
5. Προγράμματα που χρησιμοποιήθηκαν.....	27
5.1. Eclipse.....	27
5.1.1. Εισαγωγή.....	27
5.1.2. Android SDK.....	29
5.1.3. Το περιβάλλον του Eclipse Luna.....	30
5.2. Δημιουργία γραφικών.....	34
5.2.1. AutoCAD.....	34
5.2.2. Photoshop.....	40
6. Εισαγωγή στην Εφαρμογή.....	43
6.1. Περιβάλλον – Περιγραφή Εφαρμογής.....	43
6.1.1. Οθόνη Εισαγωγής.....	44
6.1.2. Αρχική Οθόνη.....	45

6.1.3. Παιδική Χαρά.....	46
6.1.4. Σχολείο.....	48
6.1.5. Σπίτι.....	50
6.1.5.1. Παιδικό Δωμάτιο.....	50
6.1.5.2. Γραφείο.....	52
6.1.5.3. Σαλόνι.....	53
6.1.5.4. Κουζίνα.....	55
6.2. Χαρακτήρες.....	58
7. Ανάλυση Κώδικα.....	59
7.1. Βασικά στοιχεία μιας Android εφαρμογής.....	59
7.2. Προγραμματισμός Παιχνιδιού.....	62
7.2.1. Προγραμματισμός Οθόνης Εισαγωγής.....	62
7.2.2. Προγραμματισμός Αρχικής Οθόνης.....	65
7.2.3. Προγραμματισμός Παιδικής Χαράς και Περιπτέρου.....	68
7.2.4. Προγραμματισμός Σχολείου και Πίνακα.....	78
7.2.5. Προγραμματισμός Σπιτιού.....	81
7.2.5.1. Προγραμματισμός Υπνοδωματίου.....	86
7.2.5.2. Προγραμματισμός Γραφείου.....	87
7.2.5.3. Προγραμματισμός Σαλονιού.....	90
7.2.5.4. Προγραμματισμός Κουζίνας.....	94
7.3. Προσαρμογή και ενσωμάτωση σε Android συσκευές.....	102
8. Μελλοντικές Βελτιώσεις.....	102
9. Βιβλιογραφία.....	103
10. Ιστογραφία.....	104

1. Παιχνιδοποίηση (Gamification)

Το περιβάλλον της υγείας έρχεται καθημερινά αντιμέτωπο με δεκάδες προκλήσεις που ωθούν τον κλάδο σε αναζήτηση νέων ευκαιριών για τη μείωση των δαπανών και την αύξηση της απόδοσης.

Στο χώρο της τεχνολογίας και των νεοφυών επιχειρήσεων υπάρχει μία λέξη η οποία ακούγεται σχεδόν σε κάθε συζήτηση. Η λέξη αυτή είναι η παιχνιδοποίηση (gamification). Είναι μια νέα διαδικασία μέσω της οποίας ενεργοποιούνται τα φυσικά ένστικτα του ανθρώπου –το παιχνίδι και η μάθηση– και η οποία συγκεκριμένα στο χώρο της υγείας βοηθά τους ασθενείς να γνωρίσουν καλύτερα την ασθένειά τους, να συμμετέχουν στην διαδικασία της περίθαλψής τους και να τηρούν σωστά τα θεραπευτικά σχήματα ή το πρόγραμμα παρακολούθησης της νόσου τους.

1.1. Ορισμός και σκοπός

Η παιχνιδοποίηση ή αλλιώς το gamification είναι η διαδικασία μιας συνηθισμένης καθημερινής δραστηριότητας προσθέτοντας σε αυτήν τους μηχανισμούς και τα χαρακτηριστικά που έχει ένα παιχνίδι, δηλαδή τις προτροπές, προκλήσεις, ανταμοιβές και βασικότερο απ'όλα τη διασκέδαση με στόχο την εφαρμογή τους στον πραγματικό κόσμο.

Το gamification ενσωματώνει τις αλλαγές της συμπεριφοράς με τεχνικές που χρησιμοποιούνται από σχεδιαστές παιχνιδιών ώστε να παρακινήσουν και να καταφέρουν ένα άτομο να ξεκινήσει να ασχολείται με το συγκεκριμένο παιχνίδι. Τα παιχνίδια αυτά παρέχουν συνήθως μια ποικιλία κινήτρων για να ενθαρρύνει τη συμμετοχή, όπως μέσα από την πρόοδο των επιπέδων, τη λήψη μηνυμάτων επιβράβευσης, ή την ανταμοιβή μέσω πόντων. Αυτή η ιδέα έχει μια ποικιλία εφαρμογών στα πλαίσια της υγειονομικής περίθαλψης, συμπεριλαμβανομένων της εκπαίδευσης των ασθενών, της τροποποίησης του τρόπου ζωής, της βελτιωμένης θεραπευτικής αντιμετώπισης και της συμμόρφωσης όσον αφορά τη φαρμακευτική αγωγή.

Τελικά, η έννοια του gamification στον τομέα της υγείας προσφέρει μια μεγάλη αύξηση της συμμετοχής των ασθενών και βελτιώνει κατά πολύ τον τρόπο ζωής τους (ενδυνάμωση), καθώς η εισαγωγή των ανταμοιβών μπορεί να κάνει την διαδικασία του αυτοέλεγχου και της αυτοδιαχείρισης πιο ευχάριστη δραστηριότητα.

1.2. Το παιχνίδι στην υγειονομική περίθαλψη

Η εταιρία Gamification Co σημειώνει ότι ήδη υπάρχουν παιχνίδια που μπορούν να βελτιώσουν τον τομέα της υγείας με διάφορους τρόπους και ενδεικτικά αναφέρεται σε ένα που αφορά την περίπτωση των σοβαρών εγκαυμάτων, τα οποία προκαλούν πόνο που δεν ανακουφίζεται εύκολα. Συγκεκριμένα, αναφέρει ότι το Πανεπιστήμιο της Ουάσινγκτον για να βοηθήσει τους ασθενείς εκείνους, ανέπτυξε ένα παιχνίδι εικονικής πραγματικότητας που βασίζεται στην υπερφόρτωση πληροφοριών, ενώ επιπλέον το ίδιο σε πειράματα αναδείχθηκε πιο αποτελεσματικό σε σύγκριση με την αναισθησία.

Παίζοντας το SnowWorld, όπως ονομάζεται, ο ασθενής χειρίζεται ένα χαρακτήρα που κινείται σε ένα αρκτικό περιβάλλον πετώντας μπάλες χιονιού σε πιγκουίνους και μαμούθ.

Η ιδέα του όλου επιχειρήματος βασίζεται στο ότι αν δεν υπάρχει κάτι στο οποίο μπορεί να επικεντρωθεί ο εγκέφαλος, θα το κάνει στον πόνο. Με την προσθήκη μιας πληθώρας ερεθισμάτων οι ασθενείς που λαμβάνουν τόσες πολλές πληροφορίες σε σύντομο χρονικό διάστημα, ενεργοποιούν τον τερματισμό της λειτουργίας των πόρων που προορίζονται για τον πόνο.

Την παιχνιδοποίηση της υγείας προσέγγισε και η Zamzee σχεδιάζοντας μια μικρή συσκευή για την αντιμετώπιση της παιδικής παχυσαρκίας. Η νεοφυής αυτή επιχείρηση δημιούργησε μια συσκευή που ο χρήστης μπορεί να φορέσει και η οποία με τη σειρά της καταγράφει την κινητικότητά του. Στη συνέχεια ο ίδιος κερδίζει πόντους (Zamz) από την εταιρία, τους οποίους μπορεί να εξαργυρώσει για να αποκτήσει δώρα, παρέχοντας τον έτσι ένα κίνητρο για σωματική εξάσκηση και δραστηριότητα.

Η φαρμακευτική βιομηχανία από την πλευρά της δεν έχει υιοθετήσει σε σημαντική έκταση την ιδέα αυτή, ωστόσο δύο εταιρίες έχουν ήδη κάνει τα πρώτα τους βήματα. Η Boehringer Ingelheim πέτυχε παρουσίασε το πρώτο της κοινωνικό παιχνίδι στο Facebook, με την ονομασία Syrum. Οι χρήστες παίζοντας το Syrum μπορούν να διαχειριστούν τη δική τους φαρμακευτική εταιρία και να αναπτύξουν φάρμακα για την αντιμετώπιση ασθενειών. Η Boehringer αναφέρει ότι το σχεδίασε για να προβάλει στην κοινωνία τις δυσκολίες που συναντούν οι φαρμακευτικές επιχειρήσεις και τις πολύπλοκες διαδικασίες γύρω από την έρευνα και την ανάπτυξη νέων φαρμάκων.

Νωρίτερα το 2012 κυκλοφόρησε μια εφαρμογή (application) για smartphones και η φαρμακευτική εταιρία Janssen για ασθενείς με Διαταραχή Ελλειμματικής Προσοχής – Υπερκινητικότητας (ADHD). Το Sorted, όπως ονομάζεται, βοηθά τους χρήστες να δημιουργήσουν, να ταξινομήσουν και να ιεραρχήσουν καθημερινές εργασίες, ενώ ακόμη

ενσωματώνει ένα στοιχείο βαθμολόγησης που προσδίδει στοιχεία παιχνιδιού για να τους προσφέρει κίνητρα.

1.3. Μηχανισμοί του Gamification

Όπως ήδη αναφέρθηκε, για να δοθεί στον ασθενή το κίνητρο να ασχοληθεί με μία εφαρμογή βασισμένη στο gamification, θα πρέπει να εμπεριέχονται σε αυτήν διάφοροι μηχανισμοί που ισχύουν στα παιχνίδια και μπορούν και ενθαρρύνουν τον παίκτη να συμμετέχει αλλά και να συνεχίσει.

Ένας από τους πιο κοινούς μηχανισμούς παιχνιδιού που χρησιμοποιείται και στο gamification είναι οι πόντοι. Στους χρήστες δίνονται πόντοι κάθε φορά που καταφέρνουν κάτι που το σύστημα προσπαθεί να τους ενθαρρύνει να το κάνουν. Με τους πόντους κρατείται το συνολικό σκορ με αποτέλεσμα να δημιουργείται στον χρήστη μια αίσθηση προόδου, κάτι το οποίο τον ενθαρρύνει και τον παροτρύνει να συνεχίσει το παιχνίδι. Ένας ακόμα μηχανισμός είναι τα badges, τα οποία είναι μια οπτική αναπαράσταση ενός συγκεκριμένου επιτεύγματος εντός του συστήματος του παιχνιδιού. Τέλος, η ύπαρξη πινάκων κατάταξης καλύτερων παικτών επιτρέπει στους χρήστες να δουν την πρόοδο τους σε σχέση με κάποιον άλλον. Οι πίνακες αυτοί έχουν τη δυνατότητα να παρέχουν απίστευτα κίνητρα, τα οποία κινητοποιούν τον χρήστη ώστε να επιτεχθεί ο στόχος του, αλλά ταυτόχρονα μπορεί να είναι και αποθαρρυντικοί, με την έννοια ότι οι χρήστες που δεν έχουν καλή απόδοση και δεν βρίσκονται υψηλά στην κατάταξη να σταματήσουν τη χρήση του παιχνιδιού αυτού.

Οι παραπάνω μηχανισμοί, θα πρέπει να σημειωθεί, ότι βρίσκονται στα περισσότερα, αν όχι σε όλα, τα συστήματα παιχνιδιών. Ειδικότερα, οι πόντοι είναι ένα από τους ευρέως χρησιμοποιούμενους μηχανισμούς. Είναι ισχυροί, ενώ παρέχουν άμεση ανατροφοδότηση στο χρήστη αν η ενέργεια που μόλις έκανε ήταν “ορθή”. Πέντε συστήματα έχουν εντοπιστεί, τα οποία βασίζονται στους πόντους και μπορούν να αξιοποιηθούν από το gamification.

Αρχικά είναι οι πόντοι εμπειρίας (Experience Points – XP) που εξατομικεύουν απλώς το βαθμό και την απόδοση του παίκτη. Οι χρήστες θα πρέπει να εκτελέσουν κάποιες συγκεκριμένες ενέργειες ώστε να ανταμειφθεί με XP πόντους. Τέλος, οι πόντοι εμπειρίας πότε δεν μειώνονται και συνήθως δεν υπάρχει κάποιο όριο.

Οι εξαγοράσιμοι πόντοι είναι πόντοι που μπορούν να ανταλλάσσονται είτε για εξωτερικές ανταμοιβές (δώρα, λεφτά), είτε για οφέλη εντός του συστήματος του παιχνιδιού.

Η παρουσία τους δημιουργεί μια εικονική οικονομία και στο τέλος αυτοί οι πόντοι θα έχουν αξία για τους χρήστες.

Ένα σπάνιο σύστημα πόντων στα πλαίσια του gamification είναι οι πόντοι ικανότητας. Οι συγκεκριμένη μορφή πόντων μπορεί να έχει αρκετές ομοιότητες με τους πόντους XP αλλά είναι μια διαφορετική οντότητα. Οι πόντοι ικανότητας μπορούν να κερδίζονται μόνο για πολύ συγκεκριμένες ενέργειες και δράσεις.

Επιπλέον, υπάρχουν και οι πόντοι κάρμα, που σε γενικές γραμμές είναι πόντοι που δεν έχουν κάποιο νόημα. Οι πόντοι αυτοί δημιουργούν μια συγκεκριμένη πορεία συμπεριφοράς με σκοπό την κατεύθυνση προς ορισμένες δραστηριότητες. Τα περισσότερα gamification συστήματα χρησιμοποιούν πόντους κάρμα αντί για εμπειρίας. Το καλύτερο παράδειγμα των κάρμα πόντων προέρχεται από την ιστοσελίδα reddit, όπου οι χρήστες αποκτούν “κάρμα” για κάθε ψήφο που παίρνουν για αυτά που έχουν ανεβάσει.

Τέλος, οι πόντοι φήμης είναι το πιο πολύπλοκο σύστημα, όπου φαίνεται η αξιοπιστία του χρήστη και αναπτύσσεται εμπιστοσύνη μεταξύ χρηστών. Ένα παράδειγμα είναι οι πόντοι φήμης που υπάρχει στο eBay, οι οποίοι δείχνουν την αξιοπιστία των πωλητών όσον αφορά την αποστολή και την ποιότητα των προϊόντων.

Υπάρχουν, όμως, και άλλοι μηχανισμοί του gamification που μπορούν να χρησιμοποιηθούν, δηλαδή πάλι κάποιο σύστημα με πόντους το οποίο όμως θα προάγει ένα διαφορετικό στόχο, όπως την προβολή κατάστασης και πρόοδου, με σκοπό να δημιουργηθούν προκλήσεις ή συνεργασίες μεταξύ χρηστών. Οι πιο κοινοί μηχανισμοί που μπορούν να χρησιμοποιηθούν μπορεί να είναι τα επίπεδα, τα οποία είναι μια περαιτέρω ένδειξη της πρόοδου του χρήστη μέσα σε ένα παιχνίδι και γενικά έχει δύο έννοιες. Η πρώτη έννοια δηλώνει την κατάσταση και τη γνώση του χρήστη για το σύστημα στο οποίο παίζει, ενώ η δεύτερη υποδεικνύει τη θέση ενός χρήστη μέσα σε ένα σύστημα.

Ένας ακόμα πολύ ενδιαφέρον μηχανισμός είναι οι προκλήσεις και τα quests. Οι προκλήσεις είναι ένα παζλ ή εργασίες που απαιτούν προσπάθεια για να λυθούν. Μπορούν να δώσουν στον παίκτη την κατεύθυνση που πρέπει να πάρει ώστε να μπορεί να “κινηθεί” μέσα στον κόσμο του παιχνιδιού. Συνήθως, οι προκλήσεις και τα quests βασίζονται σε συστήματα πόντων και επικεντρώνονται στην παροχή κινήτρων στους χρήστες ώστε να μπορούν και να έχουν την θέληση να ολοκληρώσουν ακόμη πιο δύσκολα επίπεδα και καθήκοντα.

Επιπροσθέτως, υπάρχει και ο ανταγωνισμός που επιτρέπει στους χρήστες να ανταγωνίζονται ο ένας τον άλλον, ώστε να προκύψει ένας νικητής και ένας ηττημένος. Βασικά η ύπαρξη των πινάκων κατάταξης μπορεί εύκολα να δημιουργήσει μια μορφή ανταγωνισμού. Ωστόσο, οι ανταγωνισμοί μεταξύ των χρηστών και των ομάδων καθορίζουν

ποιος χρήστης έχει την δυνατότητα να ολοκληρώσει ορισμένες εργασίες πιο γρήγορα και καλύτερα από τους υπόλοιπους. Αντίθετη έννοια είναι αυτή της συνεργασίας, όπου επιτρέπεται στους χρήστες να δουλέψουν και να συνεργαστούν μαζί για την επίτευξη ορισμένων καθηκόντων.

Τέλος, ο μηχανισμός της αφήγησης είναι μια σταθερή και συνεχή ιστορία. Σε επίπεδο gamification όμως είναι ένα κομμάτι παιχνιδιού που παραλείπουν, και ως χρησιμοποιείται στα κοινά παιχνίδια όπου είναι ένα από τα πιο ισχυρά εργαλεία για να προσελκύσει τους χρήστες να συνεχίσουν ώστε να ολοκληρώσουν ένα παιχνίδι.

1.4. Συμβολή στην υγεία

Ένας αυξανόμενος αριθμός κλινικών μελετών αναφέρουν τα οφέλη του gamification στη βελτίωση της υγείας. Το gamification παίζει σημαντικό ρόλο στην αλλαγή του τρόπου ζωής των ανθρώπων, διότι παρέχει ένα μακροπρόθεσμο στόχο καθώς επίσης και αλλαγή στην συμπεριφορά απέναντι στο πρόβλημα τους και έτσι μπορεί εύκολα να βελτιώσει σε μεγάλο βαθμό την περίθαλψη των ασθενών.

Το gamification μπορεί να βοηθήσει τον ασθενή να κατανοήσει και να διαχειριστεί τα συμπτώματα, να μάθει να φροντίζει τον εαυτό του ενώ ταυτόχρονα προσφέρει μια διασκεδαστική και συνεργατική προσέγγιση για την κατανόηση της κατάστασης. Επιπλέον του προσφέρει διάφορες στρατηγικές για τη περίθαλψη με βάση τη καθημερινή του ρουτίνα, έτσι ώστε να μπορεί να συνεχίσει ο ασθενής τη ζωή του χωρίς η πάθηση του να τον επηρεάζει, έτσι έχουμε και βελτίωση της συναισθηματικής και ψυχολογικής του κατάστασης αποφεύγοντας το φαινόμενο της χαμηλής αυτοεκτίμησης και διάθεσης του ασθενή σε σχέση με τους γύρω του. Έτσι, το gamification μπορεί να κάνει ένα άτομο να διαχειρίζεται σωστά την ασθένεια του με άμεσα αποτελέσματα χωρίς να αυτή να τον επηρεάζει αρνητικά.

Ένα παράδειγμα της προσφοράς του gamification στη βελτίωση της υγείας είναι η μελέτη που διεξήχθη το 2012 στο παιδικό νοσοκομείο του Τορόντο, όπου 20 έφηβοι με διαβήτη τύπου I δοκίμασαν μια εφαρμογή σε IOS που είχε σχεδιαστεί για να τους ενθαρρύνει να παρακολουθήσουν καλύτερα τα επίπεδα γλυκόζης στο αίμα τους. Οι μηχανισμοί του gamification που παρέχονταν ήταν πόντοι για κάθε εξέταση της γλυκόζης του αίματος, και bonus πόντοι για τις μετρήσεις (3 ανά ημέρα) και για κάθε 200 πόντους ανταμοίβονταν με αγορές στο App Store στο iTunes. Τα αποτελέσματα ήταν θετικά, δείχνοντας ότι η μέση συχνότητα ημερήσιων μετρήσεων αυξήθηκαν κατά 50% (Cafazzo,

Casselman, Hamming, Katzman & Palmert, 2012).

Ακόμα ένα παράδειγμα είναι και αυτό της εταιρίας CyberDoctor που ιδρύθηκε στο 2009 με σκοπό τη σωστή καθοδήγηση των ασθενών. Η CyberDoctor ανακοίνωσε έπειτα από μια δοκιμή που περιελάμβανε 100 ασθενείς με σακχαρώδη διαβήτη, τη βελτίωση του τρόπου ζωής αυτών ως προς την φαρμακευτική αγωγή, τη δίαιτα και την άσκηση (CyberDoctor Press Release 10/02/13). Συγκεκριμένα, η τήρηση της θεραπείας μέσω φαρμάκων αυξήθηκε από το 58% στο 95%, της διατροφής κατά 24%, της άσκησης κατά 14%, ενώ τέλος σημειώθηκε μείωση της HbA1c (γλυκοζυλιωμένης αιμοσφαιρίνης) από 10,7% σε 9,7%.

1.5. Mobile εφαρμογές βασισμένες στο Gamification

Οι εφαρμογές που είναι βασισμένες στο gamification εκμεταλλεύονται την φυσική παρόρμηση του ανθρώπου να ανταγωνιστεί και να επιτύχει. Αυτές οι εφαρμογές συνήθως χρησιμοποιούν διάφορους μηχανισμούς παιχνιδιών, που θα αναφερθούν αναλυτικότερα παρακάτω, ώστε να παρακινήσουν τους χρήστες και να τους ενθαρρύνουν προς μια πιο υγιή συμπεριφορά και πιο θετικά αποτελέσματα χρησιμοποιώντας μια προσέγγιση με βάση το παιχνίδι. Οι εφαρμογές, λοιπόν, που σχετίζονται με την υγειονομική περίθαλψη γίνονται όλο και πιο δημοφιλείς ειδικά στην ηλικιακή ομάδα των 18 με 25, οι οποίοι και πιο εξοικειωμένοι με την τεχνολογία.

Οι εφαρμογές υγείας έχουν σκοπό να παρακολουθεί μια συγκεκριμένη πτυχή της κατάστασης της υγείας του χρήστη, όπως τον ύπνο, τη διατροφή ή τη διαχείριση βάρους. Χρησιμοποιώντας τα εργαλεία και τις τεχνικές που ισχύουν σε ένα παιχνίδι, ενθαρρύνουν τον χρήστη να κάνει μικρές βελτιώσεις ως προς την παρακολούθηση διαφόρων μετρήσεων και να επιτύχουν υγιή αποτελέσματα. Στην συγκεκριμένη περίπτωση των χρόνιων νόσων όπως είναι ο διαβήτης, ο χρήστης μπορεί να παρακολουθεί την κατάσταση της υγείας του, τα επίπεδα σακχάρου στο αίμα, το σωματικό του βάρος, την κατανάλωση τροφίμων καθώς και να οργανώσουν τα γεύματα τους με την εύρεση κατάλληλων συνταγών για διαβητικούς.

Επίσης, οι ψηφιακές εφαρμογές υγείας μπορούν να βοηθήσουν στην πρόληψη ασθενειών, στην αυτο-διαχείριση και την τήρηση της θεραπείας του ασθενή. Για τη βοήθεια και την παρακίνηση των ανθρώπων που δυσκολεύονται να τηρήσουν το πρόγραμμα λήψης της φαρμακευτικής αγωγής και ιδιαίτερα στην περίπτωση των χρόνιων παθήσεων όπως ο διαβήτης που απαιτεί συνεχή παρακολούθηση, οι εφαρμογές αυτές χρησιμοποιούν “παιχνίδια” που ανταμοίβουν έτσι ώστε να ενθαρρύνεται η πρόσληψη των φαρμάκων. Για

αυτό το λόγο, κιάλας, οι γιατροί συνιστούν αυτές τις εφαρμογές που είναι πιο στενά συνδεδεμένες με τον ασθενή.

Τέλος, θα πρέπει να σημειωθεί ότι οι εφαρμογές στοχεύουν κυρίως στη βελτιώση της υγείας ως προς τη γνώση, την συμπεριφορά και τα συναισθήματα των χρηστών μέσα από παιχνίδια έτσι ώστε να μπορούν να διαχειρίζονται μόνοι τους το πρόβλημα υγείας με τον πιο ευχάριστο τρόπο.

Σχετικά με τις εφαρμογές πάνω στο διαβήτη, υπάρχουν τρεις βασικές περιοχές που καλύπτουν, έτσι ώστε να μπορεί να γίνει καλύτερη διαχείριση της νόσου από τους ασθενής. Οι περιοχές αυτές είναι η καταγραφή και έλεγχος της γλυκόζης, η καταγραφή των φαρμάκων και η καταγραφή της διατροφής.

Όσον αφορά την καταγραφή και τον έλεγχο της γλυκόζης του αίματος, όλες οι εφαρμογές πρέπει να έχουν μια μέθοδο για την παρακολούθηση των τιμών της γλυκόζης. Τυπικά, οι εφαρμογές αυτές επιτρέπουν στον χρήστη να προσδιορίσει την ημερομηνία και την ώρα της μέτρησης και στη συνέχεια να επιλέξει αν η μέτρηση αυτή πραγματοποιήθηκε πριν ή μετά από ένα γεύμα και φυσικά αν προηγήθηκε ή όχι κάποια φυσική δραστηριότητα. Μια ακόμη λειτουργία θα μπορούσε να ήταν η δυνατότητα του κάθε χρήστη να μπορεί να εισάγει τα επίπεδα γλυκόζης που θεωρούν οι ίδιοι ότι είναι πιθανόν να παρουσιάσουν κάποιο σύμπτωμα υπογλυκαιμίας ή υπεργλυκαιμίας. Επιπρόσθετες λειτουργίες θα μπορούσαν να ήταν η εισαγωγή του στόχου που έχει ο χρήστης ως προς τα επίπεδα γλυκόζης του και η προβολή γραφικών παραστάσεων κατά τη διάρκεια μιας ορισμένης χρονικής περιόδου.

Ένα ακόμα χαρακτηριστικό μερικών εφαρμογών για το διαβήτη είναι η ικανότητα να καταγράφονται τα φάρμακα που έλαβε ή πρέπει να λάβει ο ασθενής. Ο ασθενής θα πρέπει να εισάγει τότε χρειάστηκε ενέσεις ινσουλίνης ή οποιοδήποτε άλλο φάρμακο.

Οι ασθενείς με διαβήτη θα πρέπει να δώσουν μεγάλη βάση στην διατροφή τους και κατά πόσο είναι ισορροπημένη έτσι ώστε να επιτύχουν τον επαρκή έλεγχο της γλυκόζης στο αίμα τους. Κάποιες εφαρμογές παρέχουν εύκολη καταγραφή των θερμίδων, των υδατανθράκων και άλλων θρεπτικών συστατικών που έχουν καταναλώσει σε ένα γεύμα τους, ενώ άλλες έχουν μια βάση δεδομένων που είναι καταγεγραμμένα όλα τα επιτρεπόμενα φαγητά που μπορεί να καταναλώσει ο ασθενής σε οποιαδήποτε ώρα της ημέρας. Βέβαια, οι περισσότερες εφαρμογές δίνουν τη δυνατότητα στο χρήστη να εισάγει τα φαγητά που κατανάλωσε, καθώς και τα γραμμάρια υδατανθράκων που περιείχε το γεύμα του.

Άλλες δυνατότητες που μπορούν να παρέχουν αυτές οι εφαρμογές είναι η εξαγωγή των καταγραφών του ασθενή ώστε να μπορεί να στείλει τα δεδομένα στο γιατρό του, η

εισαγωγή βάρους ή και της πίεσης του. Γενικά, οι εφαρμογές που απευθύνονται σε χρήστες με διαβήτη μπορούν εύκολα να καταγράφουν και να διαχειρίζονται οτιδήποτε σχετίζεται με την αντιμετώπιση και τον έλεγχο της συγκεκριμένης νόσου.

2. Διαβήτης

Ο Διαβήτης κατατάσσεται στα νοσήματα διαταραχής του μεταβολισμού. Ο μεταβολισμός είναι ο μηχανισμός με τον οποίο το σώμα μας χρησιμοποιεί τις τροφές για να δώσει ενέργεια στον οργανισμό μας. Οι περισσότερες τροφές που τρώμε διασπώνται κυρίως σε γλυκόζη. Η γλυκόζη είναι μια μορφή «ζάχαρης» στο αίμα μας και αποτελεί την κύρια πηγή «καυσίμου» για το σώμα μας.

Ο άνθρωπος που πάσχει από Σακχαρώδη Διαβήτη έχει αρκετά αυξημένη ποσότητα γλυκόζης στο αίμα του (υπεργλυκαιμία). Αυτό συμβαίνει γιατί είτε το πάγκρεας δεν παράγει αρκετή ή καθόλου ινσουλίνη, ή γιατί τα κύτταρα δεν ανταποκρίνονται επαρκώς στην ινσουλίνη που παράγεται από το πάγκρεας. Ως αποτέλεσμα παρατηρείται αυξημένη κυκλοφορία γλυκόζης στο αίμα. Η παραπάνω αυτή ποσότητα γλυκόζης αποβάλλεται από το σώμα μέσω της ούρησης. Έτσι, αν το αίμα έχει επαρκή ποσότητα γλυκόζης, τα κύτταρα δεν μπορούν να την απορροφήσουν για να ικανοποιήσουν τις ανάγκες του σώματος σε ενέργεια.

Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας 171 εκατομμύρια άνθρωποι ή αλλιώς το 2,8% του παγκόσμιου πληθυσμού παρουσιάζει Σακχαρώδη Διαβήτη και υπολογίζεται πως μέχρι το 2030 το ποσοστό αυτό θα έχει σχεδόν διπλασιαστεί.

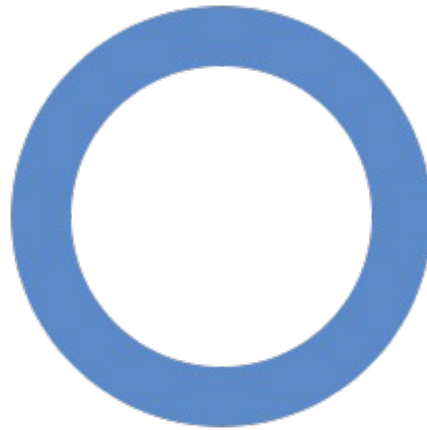
Ο Σακχαρώδης Διαβήτης συχνά δεν εμφανίζει κανένα σύμπτωμα. Εάν εμφανίζονται, τα συμπτώματα με τα οποία μπορεί να εκδηλωθεί περιλαμβάνουν την πολυδιψία (συχνή ή/και έντονη δίψα), την πολουρία/συχνουρία (συχνή ούρηση), την πολυφαγία και το αίσθημα πείνας, την κόπωση, τη θολή όραση, τη ναυτία, τον έμετο, την απώλεια σωματικού βάρους (παρά την πολυφαγία) και άλλα.

Υπάρχουν τρεις κύριοι τύποι Διαβήτη:

- Διαβήτης Τύπου 1 – Δεν παράγεται καθόλου ινσουλίνη
- Διαβήτης Τύπου 2 – Δεν παράγεται αρκετή ινσουλίνη ή η παραγόμενη ινσουλίνη δεν ανταποκρίνεται στο σκοπό της.
- Διαβήτης της Κήσης – Πολλές γυναίκες αναπτύσσουν διαβήτη κατά τη διάρκεια της κήσης.

Ο Διαβήτης Τύπου 1 & 2 είναι χρόνιες καταστάσεις. Ο Διαβήτης της κύησης συνήθως είναι παροδικός και διορθώνεται μετά τη γέννηση του παιδιού.

Η συγκεκριμένη διπλωματική εργασία αναπτύχθηκε στα πλαίσια του διαβήτη τύπου I.



Εικόνα 1: Ο μπλε κύκλος είναι το σύμβολο του διαβήτη

2.1 Διαβήτης Τύπου I

Ο διαβήτης τύπου 1 λέγεται και ινσουλινοεξαρτώμενος ή νεανικός διαβήτης. Ο χαρακτηρισμός του ως Νεανικός διαβήτης, οφείλεται στο γεγονός ότι κατά κανόνα εμφανίζεται σε άτομα νεώτερης ηλικίας, χωρίς βέβαια να αποκλείεται και η εμφάνιση αυτής της μορφής σε άτομα μεγαλύτερης ηλικίας.

Η ονομασία "ινσουλινοεξαρτώμενος" χαρακτηρίζει τη διαταραχή και τη μορφή θεραπείας που απαιτείται. Στο διαβήτη τύπου 1 το πάγκρεας παράγει ελάχιστη ή καθόλου ινσουλίνη. Επομένως, το άτομο βρίσκεται σε απόλυτη εξάρτηση από τη χορήγηση ινσουλίνης. Η μορφή αυτή διαβήτη θεωρείται ανοσολογικό νόσημα. Με απλά λόγια ο οργανισμός αναγνωρίζει τα ίδια του τα β-κύτταρα στο πάγκρεας, τα οποία συνθέτουν την ινσουλίνη και την ελευθερώνουν στο αίμα, σαν "ξένα" και κινητοποιεί μία διαδικασία καταστροφής. Έχει βρεθεί ότι σε ποσοστό 90% ευθύνεται το ανοσοποιητικό μας σύστημα το οποίο, από γενετικό λάθος, στρέφεται εναντίων των β κυττάρων και τα καταστρέφει. Στο υπόλοιπο 10% τα αίτια δεν είναι γνωστά και χαρακτηρίζεται ως ιδιοπαθής.

Υπολογίζεται ότι στη χώρα μας περίπου 10 στα 100.000 παιδιά εμφανίζουν σακχαρώδη διαβήτη κάθε χρόνο. Η ηλικία εμφάνισης ποικίλει. Συχνότερα παρουσιάζεται

σε ηλικίες από 10 έως 14 ετών, αν και σε ορισμένες περιπτώσεις εκδηλώνεται σε μεσαίες και μεγάλες ηλικίες. Τα τελευταία χρόνια, όμως, έχει παρατηρηθεί ότι ο νεανικός διαβήτης προσβάλλει ακόμα και μωρά.

Ο τύπος 1 δεν μπορεί να προληφθεί. Η πλειονότητα των ατόμων είναι κανονικού βάρους και δεν παρουσιάζουν κανένα σύμπτωμα μέχρι την εκδήλωση του Διαβήτη. Ούτε η άσκηση αλλά ούτε και η διατροφή μπορούν να το αποτρέψουν. Μέχρι στιγμής παρ' όλες τις προσπάθειες δεν έχει βρεθεί τρόπος πρόβλεψης ή καθυστέρησης εκδήλωσης του Διαβήτη τύπου 1.

Τα συμπτώματα του διαβήτη τύπου 1 συνήθως εμφανίζονται και εξελίσσονται σε μικρό χρονικό διάστημα και τις περισσότερες φορές είναι θορυβώδη. Τα κλασικά συμπτώματα του διαβήτη τύπου 1 περιλαμβάνουν πολυδιψία, πολυουρία, πολυφαγία, απώλεια βάρους, αδυναμία και αφυδάτωση.

Ο θεμελιώδης κανόνας για την αντιμετώπιση του σακχαρώδη διαβήτη είναι η γνώση και η κατανόησή του από τον ασθενή. Οι θεραπευτικές παρεμβάσεις αφορούν πολλούς τομείς. Για το λόγο αυτό απαιτείται λεπτομερής ενημέρωση από το γιατρό, ορθή εκπαίδευση του ασθενούς και καλή συνεργασία. Σωστή αντιμετώπιση του σακχαρώδη διαβήτη σημαίνει ρύθμιση των τιμών της γλυκόζης αλλά και πρόληψη των επιπλοκών του. Επιπλέον, μεγάλη προσπάθεια γίνεται ώστε ο ασθενής να μπορεί να χρησιμοποιεί εργαλεία για να ρυθμίζει μόνος του τα επίπεδα της γλυκόζης.

Οι ασθενείς με σακχαρώδη διαβήτη τύπου 1 πρέπει απαραίτητα να λαμβάνουν ινσουλίνη για να αναπληρώσουν όση δεν παράγεται. Η χορήγηση της ινσουλίνης γίνεται με ενέσεις κάτω από το δέρμα σε καθορισμένες δόσεις και ώρες. Απαραίτητη προϋπόθεση παράλληλα με την χρήση ινσουλίνης είναι τα συχνά και συνεπή γεύματα, ώστε να επιτυγχάνονται οι επιθυμητές τιμές σακχάρου χωρίς επεισόδια υπογλυκαιμίας. Για ειδικές περιπτώσεις υπάρχουν και εναλλακτικοί τρόποι χορήγησης της ινσουλίνης, όπως με αντλία ή με εισπνοές.



Εικόνα 2: Συσκευές χορήγησης ινσουλίνης

Ο θεραπευτικός στόχος για όλες τις ηλικιακές ομάδες (από τη βρεφική μέχρι τη μετεφηβική ηλικία) πρέπει να εξατομικεύεται με τελικό σκοπό την πλησιέστερη προς το

φυσιολογικό γλυκαιμική ρύθμιση, την αποφυγή των σοβαρών ή των πολλαπλών ήπιων/μέτριων υπογλυκαιμιών και τη φυσιολογική αύξηση και ανάπτυξη των πασχόντων. Τα επιδιωκόμενα επίπεδα γλυκόζης και γλυκοζυλιωμένης αιμοσφαιρίνης (HbA1c), σύμφωνα με την Ελληνική Διαβητολογική Εταιρεία (ΕΔΕ) είναι τα παρακάτω, όπως φαίνονται και στο πίνακα.

	Παιδιά	Έφηβοι
HbA1c %	< 7.5%	≤ 7.0%
Γλυκόζη νηστείας ή προγευματική (mg/dl)	90-145	80-130
Μεταγευματική γλυκόζη (mg/dl)	90-180	80-140
Γλυκόζη πριν από την κατάκλιση (mg/dl)	120-180	120-150
Υπογλυκαιμίες	Αποφυγή των σοβαρών Ανεκτές λίγες ήπιες	Αποφυγή των σοβαρών Ανεκτές λίγες ήπιες

Πίνακας 1: Επίπεδα γλυκόζης και γλυκοζυλιωμένης αιμοσφαιρίνης (HbA1c)

2.2. Διαβήτης και Διατροφή

Πέραν όμως της χορήγησης ινσουλίνης, βασικό ρόλο κατέχουν η σωστή διατροφή και η σωματική άσκηση. Το διαιτολόγιο μπορεί να περιλαμβάνει τροφές όπως πουλερικά, ψάρι, γάλα με χαμηλά λιπαρά, δημητριακά, ζυμαρικά, φρούτα και λαχανικά. Τρόφιμα με προσθετικά ζάχαρης, κόκκινο κρέας και αλμυρά, καλό είναι να καταναλώνονται κατά το ελάχιστο δυνατό. Γλυκαντικές ουσίες ειδικές για άτομα με σακχαρώδη διαβήτη κυκλοφορούν στο εμπόριο και μπορούν να χρησιμοποιηθούν σε λογική ποσότητα. Η βοήθεια του διαιτολόγου κρίνεται αναγκαία ώστε ο ασθενής να λάβει ένα διαιτολόγιο με μετρημένες ποσότητες και θερμίδες, προσαρμοσμένο στις προτιμήσεις του. Η καθημερινή άσκηση είναι απαραίτητη τόσο για την διατήρηση του κατάλληλου σωματικού βάρους, όσο και για την βελτίωση του μεταβολισμού.

- **Φρούτα και λαχανικά :**

Είναι πλούσια σε βιταμίνες και ιχνοστοιχεία καθώς και σε φυτικές ίνες. Τα φρούτα επίσης περιέχουν φυσική ζάχαρη που ονομάζεται φρουκτόζη. Κάποια από αυτά είναι αρκετά γλυκά όπως τα σταφύλια και το καρπούζι και πρέπει να τρώγονται σε περιορισμένες ποσότητες.

Αρκετά φρουτοποτά και χυμοί (νέктar) περιέχουν ζάχαρη και καλό είναι να αποφεύγονται. Να επιλέγονται οι χυμοί που αναγράφουν «χωρίς προσθήκη ζάχαρης» και ακόμα καλύτερα θα είναι να καταναλώνονται μαζί με κυρίως γεύματα

για μικρότερη αύξηση του σακχάρου στο αίμα.

Οι φυτικές ίνες που βρίσκονται στα φρούτα βοηθούν στον καλύτερο έλεγχο του σακχάρου και πρέπει να συμπεριλαμβάνονται στο καθημερινό διαιτολόγιο.

- **Λίπη και έλαια:**

Σε αυτή την κατηγορία ανήκουν τρόφιμα που έχουν υψηλή θερμιδική αξία. Λίπος συναντάται στο κρέας στο βούτυρο, στη μαργαρίνη, στο λάδι. Επίσης συναντάται στα παράγωγα του κρέατος, στο τυρί, στα επιδόρπια βασισμένα στη ζύμη (τυρόπιτες, λουκανικόπιτες, πίτσες, κέικ, μπισκότα, πατατάκια, γαριδάκια κ.α).

Στην χώρα μας καταναλώνονται τα τελευταία χρόνια όλο και περισσότερα λιπαρά. Καλό θα είναι λοιπόν από μικρή ηλικία τα παιδιά να μαθαίνουν να επιλέγουν τρόφιμα χαμηλά σε λιπαρά και οι γονείς να επιλέγουν του σωστούς τρόπους μαγειρέματος. Ψήσιμο στο φούρνο, στο grill, στον ατμό, βράσιμο είναι χαρακτηριστικά παραδείγματα σωστού τρόπου μαγειρέματος.

- **Γάλα και τα παράγωγά του (τυρί, γιαούρτι):**

Τρόφιμα πλούσια σε ασβέστιο. Το ασβέστιο είναι ένα πολύ σημαντικό ιχνοστοιχείο για τη σωστή ανάπτυξη των οστών και των δοντιών.

Γιαούρτι, τυρί σε σάντουιτς και δημητριακά σε γάλα αποτελούν εξαιρετικά επιδόρπια και γευματίδια. Θα πρέπει όμως να τονιστεί ότι αρκετά γιαούρτια που κυκλοφορούν στην αγορά για παιδιά περιέχουν αρκετές ποσότητες επιπρόσθετης ζάχαρης. Οπότε θα πρέπει να επιλέγονται αυτά στα οποία αναγράφεται η ένδειξη «χωρίς ζάχαρη». Επιπλέον, μεγάλη προσοχή πρέπει να δοθεί και στα “light” προϊόντα, τα οποία μπορεί να είναι χαμηλά σε λιπαρά αλλά δεν σημαίνει ότι δεν έχουν ζάχαρη.

- **Κρέας, ψάρι, αυγά, τυρί, όσπρια:**

Τα τρόφιμα αυτά είναι πλούσια σε πρωτεΐνες. Οι πρωτεΐνες είναι δομικά συστατικά που χρειάζεται το σώμα. Από τα παραπάνω προϊόντα καλό θα είναι να επιλέγονται αυτά με τα λιγότερα λιπαρά. Τρόφιμα σαν και αυτά πρέπει να περιλαμβάνονται δύο φορές στο καθημερινό διαιτολόγιο (μεσημέρι, βράδυ).

- **Φυτικές ίνες:**

Οι φυτικές ίνες βρίσκονται κατά κανόνα στις αμυλώδεις τροφές. Τρόφιμα με υψηλή περιεκτικότητα σε φυτικές ίνες βοηθούν στην ανάπτυξη της αίσθησης του κορεσμού, βελτιώνουν πιθανά προβλήματα στο πεπτικό σύστημα και προκαλούν μια σταδιακή και ομαλή αύξηση των επιπέδων σακχάρου στο αίμα.

Ωστόσο, θα πρέπει να δοθεί ιδιαίτερη προσοχή στην κατανάλωση των φυτικών ινών ανάλογα με την ηλικία του παιδιού. Συγκεκριμένα, σε παιδιά κάτω των 5 ετών, στα οποία δεν επιτρέπεται η κατανάλωση μεγάλων ποσοτήτων. Στις ηλικίες από 5

μέχρι 11, μπορεί να γίνει κατανάλωση λίγο μεγαλύτερων ποσοτήτων. Τέλος, για παιδιά άνω των 11 πρέπει να επιλέγονται τροφές με υψηλή περιεκτικότητα σε φυτικές ίνες.

Η τακτική μέτρηση του σακχάρου αίματος στο σπίτι με ειδικούς γλυκομετρητές βοηθά στην ρύθμιση των δόσεων της ινσουλίνης, ώστε να επιτυγχάνουμε τον επιθυμητό στόχο. Συνήθως, δύο με τρεις μετρήσεις την ημέρα είναι αρκετές. Έκτακτες μετρήσεις μπορούν να γίνουν εφόσον παρουσιαστούν συμπτώματα που θυμίζουν υπογλυκαιμία.



Εικόνα 3: Γλυκοζόμετρο

2.2.1. Ισοδύναμες τροφές

Ωστόσο, για την διευκόλυνση των ατόμων με διαβήτη στο σωστό υπολογισμό της δόσης της ινσουλίνης ή της αντιδιαβητικής αγωγής υπάρχει μια μέθοδος, η μέτρηση των ισοδύναμων υδατανθράκων της τροφής, που επιτρέπει να υπολογιστεί η απαιτούμενη αυτή δόση για την κάλυψη των αναγκών του γεύματος με μεγαλύτερη ακρίβεια, ώστε να μειωθεί η διακύμανση του μεταγευματικού σακχάρου, περιορίζοντας τη συχνότητα των υπεργλυκαιμικών ή των υπογλυκαιμικών επεισοδίων με τελικό αποτέλεσμα τη βελτίωση του γλυκαιμικού ελέγχου. Η μέθοδος αυτή επιτρέπει μεγαλύτερη ελευθερία ως προς την επιλογή του είδους, της ποσότητας της τροφής και του χρόνου του γεύματος, παρέχοντας μεγαλύτερη ευελιξία στο καθημερινό πρόγραμμα των ατόμων με σακχαρώδη διαβήτη.

Επιπρόσθετα, η γνώση των ισοδύναμων, αλλά και των θερμίδων που έχουν τα διάφορα τρόφιμα βοηθά στη διατήρηση της ποιότητας ζωής τους.

Τα παιδιά, όπως είναι λογικό, δεν είναι σε θέση να υπολογίζουν ισοδύναμες τροφές και γλυκαιμικούς δείκτες. Παρόλα αυτά, όμως, μπορούν μέσα από το παιχνίδι να καταλάβουν και να μάθουν την επιτρεπόμενη διατροφή.

2.3. Παιχνίδια – Εφαρμογές για τον παιδικό διαβήτη

Η κατανόηση και η διαχείριση της ασθένειας του διαβήτη από τα παιδιά είναι αρκετά δύσκολη. Μια στρατηγική για την προσέγγιση των παιδιών ώστε να ασχοληθούν με τον διαβήτη είναι οι εφαρμογές, οι οποίες μέσα από απλά παιχνίδια και ελκυστικά γραφικά, προσπαθούν να εξηγήσουν με τον πιο απλό τρόπο τον διαβήτη, χωρίς όμως να πουν εξ' ολοκλήρου όλη την αλήθεια λόγω του μικρού της ηλικίας τους.

mySugr Junior

Πρόκειται για μια εφαρμογή ημερολογίου για το διαβήτη ειδικά κατασκευασμένο για τα παιδιά. Η εφαρμογή αυτή βοηθάει τα παιδιά να διαχειρίζονται καλύτερα το διαβήτη τους με έναν πιο διασκεδαστικό και παιχνιδιάρικο τρόπο, ενώ βοηθάει και τους γονείς να αισθάνονται πιο σίγουροι για τα παιδιά τους καθώς τους δίνει τη δυνατότητα να ελέγχουν τη θεραπεία του παιδιού τους.



Εικόνα 4.1: mySugr Junior

Είναι γνωστό ότι η θεραπεία και διαχείριση του διαβήτη είναι πολύ περίπλοκη και ενοχλητική κυρίως για τα παιδιά που αρκετά συχνά δεν ξέρουν να χειρίζονται όλη τη

διαδικασία που απαιτεί ο διαβήτης, δηλαδή να πρέπει να αξιολογούν τα επίπεδα γλυκόζης τους, να υπολογίζουν πάντα τις ποσότητες των υδατανθράκων που λαμβάνουν μέσω του φαγητού και να γνωρίζουν την ποσότητα της ινσουλίνης που πρέπει να πάρουν.

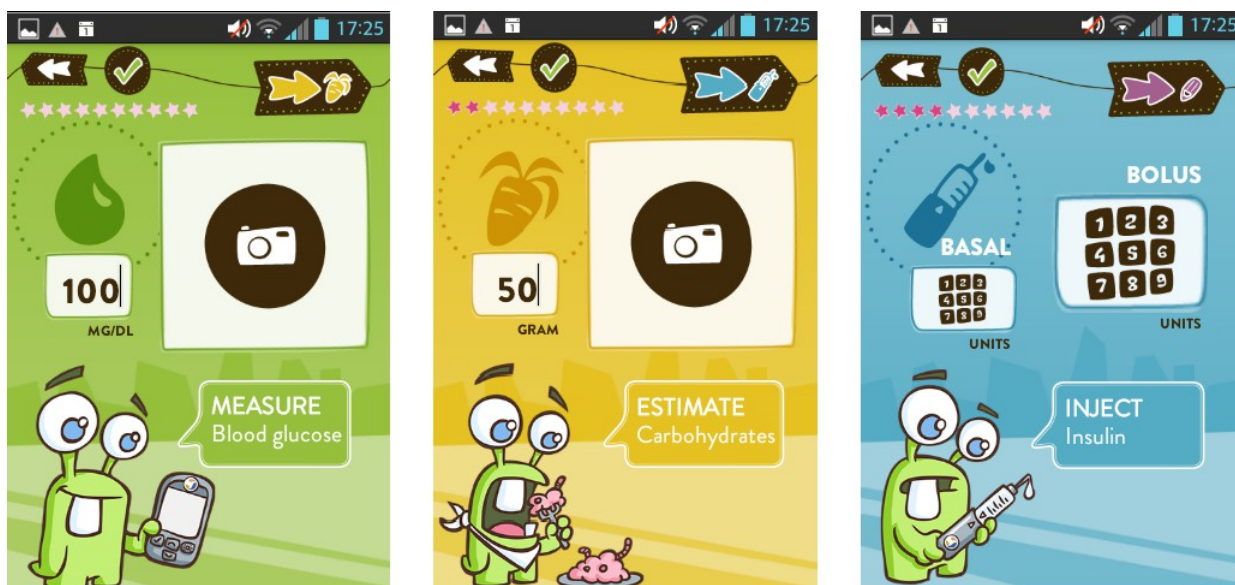
Για τον παραπάνω λόγο αναπτύχθηκε η εφαρμογή MySugr Junior για να καταστήσει ευκολότερη τη διαχείριση του διαβήτη από τα παιδιά. Επιπλέον επιτρέπει στους γονείς να ελέγχουν τη θεραπεία του παιδιού τους ακόμα και όταν αυτό δεν βρίσκεται στο σπίτι ή μαζί με τους γονείς του. Η εφαρμογή, λοιπόν, μοιάζει με παιχνίδι όπου τα παιδιά με κάθε είσοδο που κάνουν παίρνουν κάποιους πόντους. Ο στόχος είναι το παιδί να μαζέψει έναν συγκεκριμένο αριθμό πόντων καθημερινά, κάτι το οποίο ενθαρρύνει τα παιδιά να ασχοληθούν με το διαβήτη τους τακτικά.

Ο κεντρικός ήρωας είναι ένα μικρό τέρας το οποίο συνοδεύει τα παιδιά μέσα στην εφαρμογή και τους ενθαρρύνει ανάλογα με τις καταχωρήσεις που κάνουν. Το παιδί μπορεί να εισάγει δεδομένα, όπως είναι τα επίπεδα γλυκόζης στο αίμα, τα τρόφιμα, την ινσουλίνη ή και να τραβήξουν φωτογραφίες των γευμάτων τους καθώς και να ζητήσουν βοήθεια όταν οι γονείς τους δεν είναι μαζί τους. Όλες οι καταχωρήσεις που κάνει το παιδί στέλνονται με e-mail ή απλό μήνυμα στο κινητό των γονιών. Με τον τρόπο αυτό, το παιδί μπορεί να ζητήσει πληροφορίες σχετικά με τον υπολογισμό των υδατανθράκων ή τη δόση της ινσουλίνης που πρέπει να πάρει.

Γενικά, η εφαρμογή mySugr Junior δίνει τη δυνατότητα όχι μόνο στα παιδιά να είναι πιο ανεξάρτητα, αλλά τους βοηθάει να συμβιβαστούν με το διαβήτη και την ιδιαιτερότητα που τον χαρακτηρίζει καθώς και να μάθουν να ασχολούνται με τη θεραπεία τους.

Συγκεκριμένα, τα χαρακτηριστικά της εφαρμογής είναι:

- Εισαγωγή των επιπέδων γλυκόζης στο αίμα, υδατανθράκων και των μονάδων ινσουλίνης με έναν απλό και εύκολο τρόπο.
- Λήψη φωτογραφιών.
- Αποστολή δεδομένων και φωτογραφιών μέσω e-mail.
- Ενημέρωση σχετικά με διάφορα θέματα του διαβήτη.
- Συλλογή πόντων για την “εξημέρωση” του τέρατος του διαβήτη.



Εικόνα 4.2: mySugr Junior

Τέλος, χαρακτηρίζεται από την ευκολία της χρήση αλλά και τη δυνατότητα αναφοράς στους γονείς, η οποία είναι η ιδανικότερη επιλογή τόσο για τους γονείς όσο και για τα παιδιά, οπότε θεωρείται μια από τις καλύτερες εφαρμογές για τα παιδιά και την εκμάθηση τους όσον αφορά τη σωστή διαχείριση του διαβήτη.

Carb Counting with Lenny

Η μέτρηση των υδατανθράκων αποτελεί βασική ικανότητα για τη διαχείριση του διαβήτη. Η καταμέτρηση των υδατανθράκων επιτρέπει στα παιδιά με διαβήτη να καταναλώνουν τροφές ακριβώς όπως και τα άλλα παιδιά και να αυξάνει την αίσθηση του ελέγχου και την αυτοπεποίθηση στη διαχείριση του διαβήτη, αφού πλέον δεν θα χρειάζεται οι γονείς να προγραμματίζουν τα γεύματά τους.

Για αυτόν τον σκοπό αναπτύχθηκε μια εφαρμογή από την Medtronic, όπου ο Lenny, ένα λιοντάρι με διαβήτη τύπου I, επικεντρώνεται γύρω από τη διδασκαλία των παιδιών ώστε να μπορούν να υπολογίζουν με ακρίβεια τους υδατάνθρακες που καταναλώνουν μέσα από τα γεύματά τους. Διαθέτει έναν οδηγό τροφίμων, στον οποίο τα παιδιά μαθαίνουν για τις τιμές υδατανθράκων πολλών ειδών τροφίμων σε όλες τις βασικές ομάδες τροφίμων, ενώ υπάρχει και η επιλογή της προσαρμογής αυτού του οδηγού όπου τα παιδιά θα προσθέτουν τα δικά τους αγαπημένα φαγητά. Επιπλέον η εφαρμογή έχει τέσσερα παιχνίδια και είναι τα εξής:

- Υδατάνθρακες ή όχι Υδατάνθρακες
- Συγκρίνετε τους Υδατάνθρακες

- Μαντέψτε τους Υδατάνθρακες
- Δημιουργία ολοκληρωμένου γεύματος.

Τα παραπάνω παιχνίδια σχετίζονται πιο πολύ με την εκμάθηση και όχι τόσο την εκπαίδευση μέρα με την μέρα, όπως είναι η σωστή διαχείριση που προτείνει η εφαρμογή mySugr. Παρόλα αυτά, προσφέρονται πολλές πληροφορίες για τη διδασκαλία των παιδιών ώστε να γνωρίσουν τι είναι ακριβώς οι υδατάνθρακες, αλλά και ποιες είναι οι πιθανές τροφές των υδατανθράκων σε πολλά κοινά τρόφιμα.

Συνοψίζοντας, τα βασικά χαρακτηριστικά της συγκεκριμένης εφαρμογής είναι η εκμάθηση των τιμών των υδατανθράκων, η δοκιμή των γνώσεων για τους υδατάνθρακες με διασκεδαστικά και διαδραστικά παιχνίδια όπου θα μπορούν τα παιδιά να ανταγωνίζονται άλλους παίκτες ανάλογα με το σκορ που θα έχουν.



Εικόνα 5: Carb Counting with Lenny

Monster Manor

Το Monster Manor είναι ένας συνδυασμός παιχνιδιού και καταγραφής της γλυκόζης για παιδιά με διαβήτη τύπου Ι. Το σκεπτικό είναι απλό. Τα παιδιά εισάγουν τη γλυκόζη τους καθώς και άλλα στατιστικά, και ανταμείβονται με "Πινιάτες". Αυτές οι "Πινιάτες" περιέχουν διασκεδαστικά δωράκια, τα οποία ονομάζονται "Beanz", και επιτρέπουν στο παιδί να μεγαλώνουν και να διαχειρίζονται τα τέρατα τους, ενώ με το χρυσό που συλλέγει μπορεί να

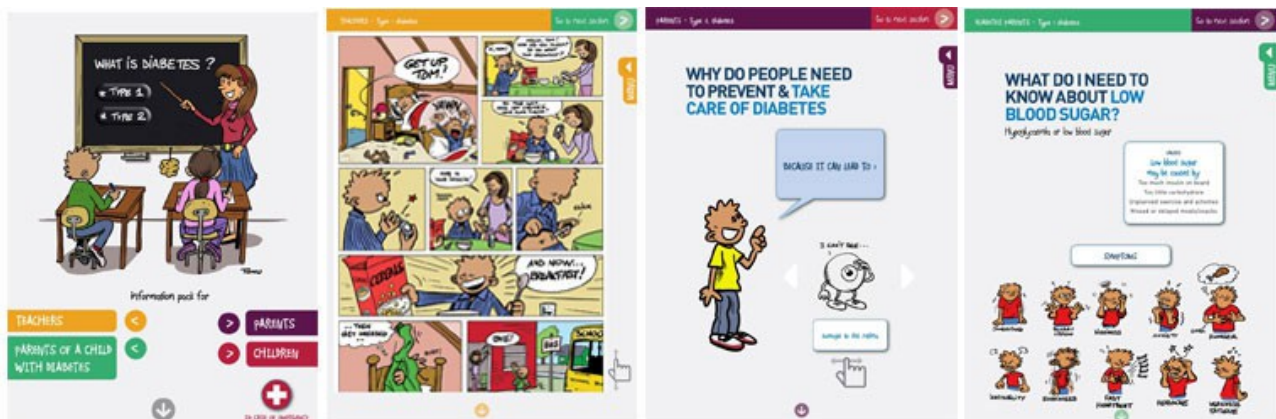
αγοράσει κατοικίδια ζώακια, αλλά και αξεσουάρ για τέρατα του.



Εικόνα 6: Monster Manor

Kids & Diabetes in Schools

Η εφαρμογή Kids & Diabetes in Schools είναι διαθέσιμη για κινητά που διαθέτουν λειτουργικό iOS. Πρόκειται για εφαρμογή, η οποία παρέχει πληροφορίες σχετικά με την πρόληψη και τη διαχείριση του διαβήτη στα σχολεία. Χωρίζεται σε 4 κεφάλαια και απευθύνεται στο προσωπικό του σχολείου, στους γονείς των παιδιών με διαβήτη, καθώς και στα παιδιά. Αυτό το πακέτο παράχθηκε από τη Διεθνή Ομοσπονδία για το Διαβήτη και των εταίρων.



Εικόνα 7: Kids & Diabetes in Schools

3. Ενδυνάμωση των ασθενών(Patient Empowerment)

Η ενδυνάμωση των ασθενών είναι η παροχή στους ασθενείς να λαμβάνουν ενεργό ρόλο στις αποφάσεις ως προς την δική τους υγειονομική περίθαλψη, δηλαδή διαχειρίζεται τη νόσο του και εκπαιδεύονται ώστε να μάθουν περισσότερες πληροφορίες για την ασθένεια του, την κατάσταση υγείας του ή και τις επιλογές θεραπείας. Ο ασθενής αναλαμβάνει την ευθύνη για τη φροντίδα του με την συχνή επικοινωνία με το γιατρό του και με άλλους πάροχους υγείας, την εξασφάλιση της ασφάλειας του, τη συλλογή στοιχείων για τη κατάσταση της υγείας του αλλά και σωστή λήψη αποφάσεων.

Μέχρι και τα τελευταία χρόνια, η σκέψη ότι ο ασθενής θα συμμετείχε πλήρως στη διαχείριση της υγειονομικής περίθαλψης του, ήταν αδιανόητη. Σήμερα, πολλοί ασθενείς έχουν συνειδητοποιήσει ότι αυτό έχει ζωτική σημασία για τη διατήρηση της υγείας τους αλλά και την αντιμετώπιση ιατρικών προβλημάτων ή προκλήσεων.

Ένας ασθενής που είναι πλέον αρμόδιος για την κατάσταση της υγείας του, θα πρέπει να πάρει την ευθύνη αυτής συνειδητοποιώντας ότι γνωρίζει το σώμα του και τις ανάγκες του, καλύτερα από οποιονδήποτε άλλον, κάτι το οποίο θα βοηθήσει στην σωστή λήψη αποφάσεων σχετικά με τη θεραπεία του. Επιπλέον, θα πρέπει ο ασθενής να θέσει στόχους. Κατανοώντας ότι ο οργανισμός δεν αντιδρά πάντα με τον τρόπο που αναμένεται, θα πρέπει ο ασθενής να θέσει κάποιους στόχους για τη θεραπεία του και να προσπαθήσει να φτάσει αυτό τον στόχο με σκληρή δουλειά. Σε ορισμένες περιπτώσεις, ο ασθενής μπορεί να έχει ως στόχο να θεραπευτεί τελείως, ή απλά να επιθυμεί να μπορεί να διαχειριστεί καλύτερα την πάθηση του ή και ακόμα να μάθει να αντιμετωπίζει κάποιο ιατρικό πρόβλημα που μπορεί να του προκύψει. Θα πρέπει, επίσης, να συνεργάζεται με το γιατρό του αλλά και με άλλους παρόχους υγείας ακόμα και με άλλους ασθενείς ώστε να τον βοηθήσει στις διάφορες αποφάσεις που θα κληθεί να πάρει σχετικά με τη θεραπεία του. Καθώς και να αναζητήσει περαιτέρω πληροφορίες για την πάθηση του, τα συμπτώματα, εναλλακτικές θεραπείες και να συμμετάσχει σε συζητήσεις με άλλους ασθενείς αλλά κυρίως με το γιατρό του. Με αυτό τον τρόπο, ο ασθενής θα νιώθει πιο καλά προετοιμασμένος για οποιοδήποτε πρόβλημα στην υγεία και αν προκύψει και θα καταλήγει πιο εύκολα σε αποφάσεις που έχουν να κάνουν με την υγεία του.

Συνοψίζοντας, καταλήγουμε στο γεγονός ότι όλο και περισσότεροι ασθενείς συνειδητοποιούν ότι μπορούν να βελτιώσουν την κατάσταση της υγείας τους, αναλαμβάνοντας ευθύνη για τις αποφάσεις της προσωπικής τους υγειονομικής περίθαλψης πάντα σε συνεργασία με τους γιατρούς που τους έχουν αναλάβει και έτσι μπορούν να συμμετέχουν πλήρως σε αυτή τη διαδικασία.

3.1. Ενδυνάμωση σε άτομα με διαβήτη

Σύμφωνα με τους Meeto & Gorau (2005), υπάρχουν δύο γενικοί στόχοι για τους ανθρώπους με διαβήτη. Πρώτον, η συνεχής βελτίωση εμπειρίας ως προς την υγεία και κανονικοποίηση της ζωής τόσο σε ποσότητα όσο και σε ποιότητα, ενώ τέλος η πρόληψη και θεραπεία του διαβήτη αλλά και των επιπλοκών του μέσω της εντατικοποίησης της ερευνητικής προσπάθειας. Οι δύο αυτοί στόχοι απαιτούν την ενεργό συμμετοχή των ασθενών με διαβήτη, το οποίο επιτυγχάνεται μέσω της ενδυνάμωσης που δίνεται στους ασθενείς.

Η φιλοσοφία της ενδυνάμωσης βασίζεται σε τρία βασικά στοιχεία της φροντίδας χρόνιων νόσων: τις επιλογές, τον έλεγχο και τις συνέπειες. Οι επιλογές που κάνουν οι ασθενείς κάθε μέρα έχουν αντίκτυπο στους στόχους που έχουν θέσει.

Ο προσδιορισμός των στόχων είναι μια αποτελεσματική στρατηγική της ενδυνάμωσης, καθώς οι ασθενείς συμμετέχοντας στην επιλογή των στόχων αισθάνονται πιο ενεργοί απέναντι στη διαχείριση της δικής του υγείας, Ο προσδιορισμός των στόχων περιλαμβάνει μια διαδικασία πέντε βημάτων (Funnel & Anderson, 2004) έτσι ώστε να βοηθήσει τους ασθενείς να:

- Ορίσουν το πρόβλημα τους,
- Προσδιορίσουν τις πεποιθήσεις, τις σκέψεις και τα συναισθήματα τους που μπορεί να τους παρεμπόδιζαν στις προσπάθειες τους,
- Προσδιορίσουν τους μακροπρόθεσμους στόχους με τους οποίους μπορούν να λειτουργήσουν καλύτερα οι ασθενείς,
- Δεσμευτούν ότι θα αλλάξουν συμπεριφορά, κάτι που θα τους βοηθήσει να επιτύχουν τους στόχους τους, και τέλος,
- Αξιολογήσουν τις προσπάθειές τους και να προσδιορίσουν τις γνώσεις που παρέλαβαν μέσω από αυτή τη διαδικασία.

Η διαδικασία, λοιπόν, της ενδυνάμωσης επιτρέπει στους ασθενείς με διαβήτη να αναλάβουν την ευθύνη των επιλογών τους και την καθημερινή φροντίδα του διαβήτη. Η πρόκληση, σύμφωνα με την Funnel (2004) είναι να αναγνωριστεί ότι με τον καθορισμό των στόχων οι ασθενείς θα επιτύχουν καλύτερα μακροπρόθεσμα αποτελέσματα προϋποθέτοντας, βέβαια, και την κατάλληλη εκπαίδευση.

Έτσι στα πλαίσια της εκπαίδευσης του διαβήτη, η έννοια της ενδυνάμωσης βασίζεται σε αρκετές υποθέσεις, όπως:

- Η φροντίδα του διαβήτη παρέχεται από το άτομο το οποίο ζει με αυτή την κατάσταση σε καθημερινή βάση.
- Ο διαβήτης επηρεάζει τις συναισθηματικές, πνευματικές, κοινωνικές, σωματικές και γνωστικές πτυχές της ζωής του ατόμου.
- Άτομα με διαβήτη γνωρίζουν τόσο τις επιβαρύνσεις όσο και τα οφέλη του διαβήτη καθώς και τις επιλογές όσον αφορά την προσωπική φροντίδας.
- Τέλος, τα άτομα που έχουν διαγνωστεί με διαβήτη χρειάζονται πληροφορίες για την κατάσταση ώστε να μπορούν οι ίδιοι να κάνουν συνειδητές επιλογές.

Τελικά, η φιλοσοφία της ενδυνάμωσης αντιπροσωπεύει μια αλλαγή στη αντιμετώπιση ασθενειών τόσο από τους ασθενείς όσο και από τους επαγγελματίες υγείας. Για το διαβήτη υπάρχει ακόμα μια ανησυχία κατά πόσο οι άνθρωποι αυτοί θα μπορέσουν να εκπαιδευτούν σωστά ώστε να φροντίζουν τον εαυτό τους, αλλά με την καλύτερη ενημέρωση και την παροχή κατάλληλων πόρων, το επίπεδο της περίθαλψης θα ανέβει. Με τη διάδοση της ενδυνάμωσης των ατόμων με διαβήτη θα διευκολυνθούν τόσο οι ασθενείς όσο και οι οικογένειες τους, οπότε αξίζει τον χρόνο και την ενέργεια που απαιτείται για την επίτευξη αυτού το στόχου.

3.2. Αυτοδιαχείριση (Self-Management)

Η αυτοδιαχείριση (self-management) συνδέεται άμεσα με την ενδυνάμωση των ασθενών (patient empowerment). Συγκεκριμένα η αυτοδιαχείριση είναι η δυνατότητα του ασθενή να διαχειρίζεται μια χρόνια κατάσταση και τις επιπτώσεις της στην σωματική του υγεία σε καθημερινή βάση. Η ενημέρωση και η απόκτηση γνώσεων του ασθενή για αυτοέλεγχο της νόσου αποτελεί τον παράγοντα – κλειδί για αλλαγή του τρόπου αντίληψης της νόσου, για τη σωστή μεταβολική ρύθμιση και τέλος για τη βελτίωση της ποιότητας ζωής. Στην περίπτωση του διαβήτη, ο ασθενής είναι υπεύθυνος να διαχειρίζεται την νόσο του, όπως δηλαδή να λαμβάνει την φαρμακευτική του αγωγή, να μετράει τα επίπεδα σακχάρου στο αίμα και αναλόγως να προσαρμόζει τα φάρμακα, το διαιτολόγιο αλλά και την σωματική του δραστηριότητα. Όλα τα παραπάνω θα πρέπει να ενσωματωθούν στην καθημερινότητα του ασθενή, με την υποστήριξη των επαγγελματιών υγείας, της οικογένειας και των φίλων.

Συνήθως, ασθενείς με διαβήτη έχουν επαφή με γιατρό για λίγες ώρες ανά έτος, ενώ το υπόλοιπο του χρόνου καλούνται να διαχειρίζονται τα προβλήματα και τις ερωτήσεις

σχετικά με το διαβήτη μόνοι τους. Οπότε, η σωστή διαχείριση μπορεί να συμβάλλει στη μείωση εμφάνισης επικίνδυνων επιπλοκών και κατ'επέκταση στη μείωση εισαγωγής σε νοσοκομεία, καθώς και στη βελτίωση της ποιότητας ζωής του ασθενή με διαβήτη. Βέβαια, μια επιτυχημένη αυτοδιαχείριση απαιτεί γνώσεις ως προς την συγκεκριμένη νόσο αλλά και εκπαίδευση από τους επαγγελματίες υγείας.

Η εκπαίδευση για αυτοδιαχείριση στο σακχαρώδη διαβήτη έχει ως κέντρο τον πάσχοντα από διαβήτη και προσβλέπει στην εκπαίδευση του ιδίου αλλά και του περιβάλλοντός του με στόχο την απόκτηση γνώσεων και δεξιοτήτων που θα συμβάλλουν στον καλύτερο δυνατό έλεγχο της νόσου, την πρόληψη των επιπλοκών και τη βελτίωση της ποιότητας ζωής του. Πρόκειται για μια συνεχή δια βίου διαδικασία, η οποία πρέπει να συνοδεύεται και από ψυχοκοινωνική υποστήριξη του πάσχοντος. Η εκπαίδευση- ενημέρωση των ασθενών είναι ευρέως αποδεκτή, ως απόλυτη προϋπόθεση της ρύθμισης του σακχαρώδη διαβήτη και ολοένα και περισσότερες έρευνες συνηγορούν, ότι χωρίς αυτή, δεν είναι δυνατό να εξασφαλισθεί αποτελεσματικός έλεγχος.

Η εκπαίδευση αυτή, λοιπόν, αποσκοπεί στο να καταστήσει τους ασθενείς ικανούς για ενεργό συμμετοχή στη λήψη αποφάσεων και χάραξη θεραπευτικής στρατηγικής μαζί με τον αρμόδιο επαγγελματία υγείας. Η συμμετοχή του ασθενή είναι απαραίτητη για την επιτυχή αντιμετώπιση της νόσου. Έτσι θα πρέπει να εξατομικεύεται λαμβανόμενων υπόψη των αναγκών, υποκειμενικών και αντικειμενικών, των επιθυμιών, των πεποιθήσεων, αλλά και δυνατοτήτων του κάθε ασθενή.

Θα πρέπει να γίνεται περιγραφή της εξελεγκτικής διαδικασίας του διαβήτη και να τονίζεται η σημασία της ρύθμισης του σακχάρου, αλλά και άλλων παραμέτρων, όπως της αρτηριακής πίεσης και της χοληστερίνης, έστω και αν ο ασθενής δεν έχει υποκειμενικά ενοχλήματα από τη διαταραχή των στοιχείων αυτών.

Επιπλέον, είναι απαραίτητη η ενσωμάτωση της διατροφικής διαχείρισης και η άσκηση στον τρόπο ζωής, καθώς και η περιγραφή των θεραπευτικών επιλογών και έμφαση στην ανάγκη τήρησης της επιλεγμένης φαρμακευτικής αγωγής. Τέλος, κρίνεται απαραίτητη η εκπαίδευση στη λήψη αποφάσεων για τροποποίηση της θεραπευτικής αγωγής με βάση τα αποτελέσματα της αυτοδιαχείρισης.

Παρότι, η εκπαίδευση αυτή καθεαυτή δεν αποτελεί θεραπεία, εντούτοις κανένας ασθενής με διαβήτη δεν μπορεί να αντιμετωπισθεί επαρκώς, εάν δεν έχει διδαχθεί τις βασικές αρχές διατροφικής αγωγής, πειθαρχίας, τεχνικής χορήγησης υποδόριας ένεσης ινσουλίνης, ελέγχου των επιπέδων σακχάρου στο αίμα, σωματικής άσκησης, φροντίδας κάτω άκρων, και άλλων απαραίτητων παραμέτρων. Επιπροσθέτως, η συνεχής εκπαίδευση των ασθενών είναι αυτή που μακροπρόθεσμα εξασφαλίζει τη διατήρηση του

ευεργετικού αποτελέσματος της αυτο-διαχείρισης.

Τα πλεονεκτήματα του αυτοέλεγχου και κατ'επέκταση της αυτοδιαχείρισης μπορούν να συνοψιστούν ως εξής. Αρχικά, δίνεται η δυνατότητα να επιτευχθεί η καλή ρύθμιση του σακχάρου, τόσο στην καθημερινή ζωή του διαβητικού όσο και κατά τη διάρκεια γεγονότων όπως λοίμωξη, εγχείρηση κ.α., μέσω της συλλογής και της αξιολόγησης των στοιχείων. Ο ασθενής μπορεί ευκολότερα να κατανοήσει την νόσο και να συμμετέχει ενεργά σε αυτή με αποτέλεσμα την καλύτερη κοινωνική και επαγγελματική προσαρμογή του ασθενή. Επιπλέον, μειώνονται σημαντικά οι εισαγωγές στα νοσοκομεία, αλλά και η συχνότητα και η βαρύτητα των επιπλοκών του σακχαρώδη διαβήτη.

Συμπερασματικά, λοιπόν, μπορεί να ειπωθεί ότι η παρακολούθηση, ο αυτοέλεγχος και κυρίως η αυτο-διαχείριση είναι ένας τομέας με πολύ μεγάλη σημασία, που συμβάλλει στην καλύτερη ποιότητα ζωής του διαβητικού ασθενή, εφόσον, βέβαια, χρησιμοποιηθεί σωστά.

4. Android

Το Android είναι λειτουργικό σύστημα για κινητές συσκευές, το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Handset Alliance. Επιτρέπει στους προγραμματιστές να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα smartphones και τα tablets, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, συνηθισμένους Η/Υ (π.χ. το HP Slate 21) και σε άλλες ηλεκτρονικές συσκευές.



Εικόνα 8: Android

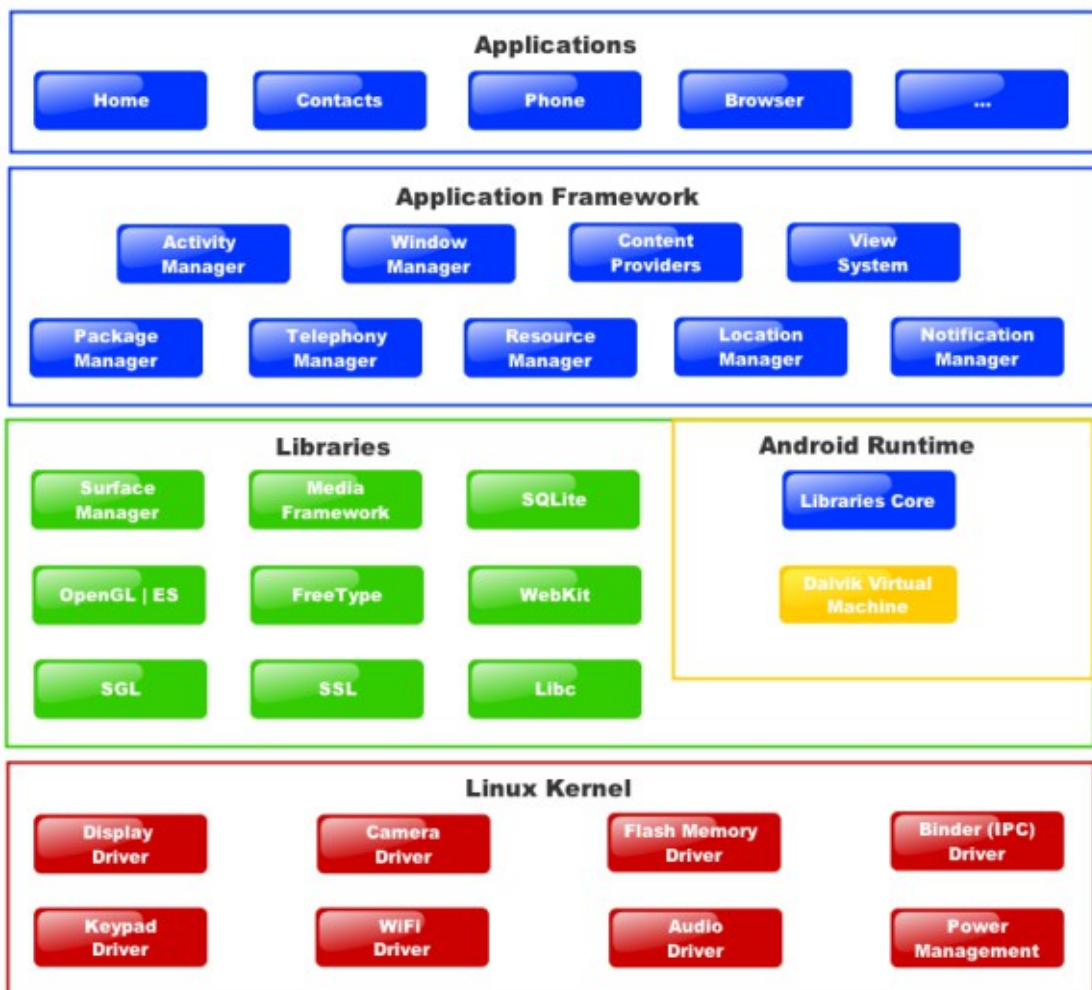
4.1. Ιστορική αναδρομή

Η Google, έχοντας εντοπίσει αυξημένη χρήση του internet και αναζητήσεων στον παγκόσμιο ιστό μέσω κινητών συσκευών (mobile devices) εξαγοράζει το 2005 την Android Inc με σκοπό την ανάπτυξη μιας πλατφόρμας για τέτοιου είδους συσκευές. Περίπου την ίδια περίοδο, η Apple παρουσιάζει το IOS (2007) το οποίο κάνει χρήση κάποιων επαναστατικών καινοτομιών όπως η υποστήριξη multitouch και η ανοιχτή παγκόσμια αγορά εφαρμογών. Το Android γρήγορα προσαρμόστηκε ώστε να υιοθετήσει και να παρέχει επίσης τις προαναφερθείσες δυνατότητες αν και κατά κοινή ομολογία οι πρώτες εκδόσεις του υπολείπονταν του IOS όσον αφορά τις υποστηριζόμενες λειτουργίες και τη συνολική εμπειρία που πρόσφερε στους χρήστες. Το 2007, λοιπόν, τη χρονιά λανσαρίσματος του IOS, δημιουργείται ένας οργανισμός που αποτελείται από μεγάλο αριθμό εταιρειών τηλεπικοινωνιακού εξοπλισμού καθώς και εταιρείες πληροφορικής όπως η Google, η T-Mobile, η Motorola, η Samsung, η Sony Ericsson, η Intel, η Vodafone, η Toshiba κ.α. με όνομα Open Handset Alliance και σκοπό την έρευνα και την ανάπτυξη τεχνολογιών για την παραγωγή συσκευών που θα διευκολύνουν τόσο τους παρόχους κινητής τηλεφωνίας όσο και τους κατασκευαστές κινητών τηλεφώνων αλλά και τους προγραμματιστές εφαρμογών. Τα μέλη της συμμαχίας δεσμεύτηκαν να παρέχουν τις τεχνολογίες αυτές βάσει του μοντέλου ανοιχτού πηγαίου κώδικα Apache. Η πρώτη 'early look' έκδοση του Android SDK δημοσιεύτηκε το Νοέμβριο του 2007, ενώ το πρώτο smartphone που έκανε χρήση λειτουργικού Android ήταν το G1 της T-Mobile. Οι συσκευές Android άρχισαν να διαδίδονται με γρήγορο ρυθμό κυρίως λόγω της δυνατότητας της πλατφόρμας να εκμεταλλεύεται το μοντέλο cloud computing αλλά και της έμφυτης υποστήριξης για συνεργασία με μία σχεσιακή βάση δεδομένων (SQLite). Ακολούθησαν αρκετές αναβαθμισμένες εκδόσεις του Android, κάθε μία προσθέτοντας νέα χαρακτηριστικά και λειτουργίες. Για κάποιον άγνωστο λόγο κάθε έκδοση του Android φέρει και μία κωδική ονομασία ενός γλυκού εδέσματος (1.5 - Cupcake, 1.6 - Donut, 2.0 Eclair κ.α.), ενώ από τις εκδόσεις αυτές εκείνη που εισήγαγε την υποστήριξη πιο ανεπτυγμένων λειτουργιών ήταν σίγουρα η 2.0 στην οποία ενσωματώθηκε η υποστήριξη multitouch, HTML 5, text-to-speech και η δυνατότητα πιο προχωρημένων αναζητήσεων.

Η πιο πρόσφατη έκδοση του android που έχει κυκλοφορήσει είναι η 4.4, με το όνομα KitKat, που κυκλοφόρησε στις 31 Οκτωβρίου 2013, ενώ τελευταία ανακοινώθηκε ότι θα κυκλοφορήσει το νέο android με το όνομα Lollipop (5.0).

4.2. Αρχιτεκτονική του Android

Όπως φαίνεται στο σχήμα 1, στον πυρήνα της πλατφόρμας Android βρίσκεται ένα Linux kernel το οποίο είναι υπεύθυνο για τη διαχείριση των device drivers, τον έλεγχο πρόσβασης στους πόρους του συστήματος, τη διαχείριση μνήμης και τις λοιπές υπηρεσίες που παρέχει ένα λειτουργικό σύστημα. Στους device drivers συγκαταλέγονται αυτοί της οθόνης, του WiFi, της κάμερας, του ήχου κ.α. Ένα επίπεδο επάνω βρίσκονται οι native βιβλιοθήκες του συστήματος που είναι γραμμένες σε C++ και περιλαμβάνουν το OpenGL, την SQLite, την Media library κ.α. Οι εφαρμογές που τρέχουν στο κινητό μπορούν να έχουν πρόσβαση στις βιβλιοθήκες αυτές μέσω της Dalvik JVM. Όπως έχει ήδη αναφερθεί, οι εφαρμογές Android είναι γραμμένες σε Java και άρα για να τρέξουν χρειάζονται το αντίστοιχο περιβάλλον. Για να εκτελέσουμε λοιπόν μία εφαρμογή σε ένα PC είναι απαραίτητο να είναι εγκατεστημένο το κατάλληλο JRE (Java Runtime Environment), για τις εφαρμογές Android τον ρόλο του JRE παίζει η Dalvik VM.



Σχήμα 1: Αρχιτεκτονική Android

Δεδομένης της σαφώς πιο περιορισμένης επεξεργαστικής ισχύος καθώς και της διαθέσιμης μνήμης που έχουν οι κινητές συσκευές σε σχέση με τους υπολογιστές, η συγκεκριμένη VM είναι βελτιστοποιημένη να χρησιμοποιεί μικρότερα σε μέγεθος αρχεία ενδιάμεσου κώδικα, τα οποία σε αντίθεση με αυτά της Java SE έχουν κατάληξη .dex αντί για .class. Επίσης, σύμφωνα με την Google, διαθέτει και πιο καλά γραμμένο garbage collector. Αν και υπάρχει έκδοση της Java για κινητά τηλέφωνα, η Java ME, η Google έκρινε σωστό να χρησιμοποιήσει τη δική της υλοποίηση και έτσι γεννήθηκε η Dalvik. Κάθε εφαρμογή λοιπόν που γράφετε και εκτελείτε στο Android κινητό σας, χρησιμοποιεί τη Dalvik και τρέχει σε ξεχωριστό instance του VM. Στο αμέσως επόμενο επίπεδο βρίσκεται το Android SDK που περιέχει όλες τις απαραίτητες βιβλιοθήκες για τη συγγραφή εφαρμογών. Ο κώδικας που θα γράφετε για την εκτέλεση κάποιας λειτουργίας, π.χ. για την πραγματοποίηση μιας κλήσης, την αποστολή ενός SMS, τον εντοπισμό της τρέχουσας θέσης κλπ. στην ουσία θα καλεί κάποια από τις μεθόδους που παρέχονται από το συγκεκριμένο SDK. Έτσι λοιπόν, οι εφαρμογές βρίσκονται στην κορυφή του application stack που στο σχήμα 1 ονομάζεται Application Layer.

4.3. Χαρακτηριστικά του Android

- **Συνδεσιμότητα:**

Το Android υποστηρίζει διάφορες τεχνολογίες ανάμεσα στις οποίες είναι GSM/EDGE, Bluetooth, Wi-Fi, WiMAX και LTE.

- **Μηνύματα:**

Υποστηρίζει τις κλασσικές μορφές SMS και MMS καθώς επίσης και το Android Cloud To Device Messaging (C2DM), που είναι μια υπηρεσία προώθησης ειδοποιήσεων που βοηθά τους προγραμματιστές να στέλνουν δεδομένα από τον server στις εφαρμογές τους.

- **Φυλλομετρητής (browser):**

Ο browser που είναι διαθέσιμος στο Android βασίζεται στην διάταξη ανοικτού κώδικα WebKit, που είναι μια μηχανή που ενδυναμώνει διάφορους browsers.

- **Υποστήριξη Java:**

Ενώ οι περισσότερες εφαρμογές είναι γραμμένες σε Java, δεν υπάρχει Java virtual machine και δεν εκτελείται Java byte code. Οι κλάσεις στην Java συντάσσονται σε εκτελέσιμα τα οποία τρέχουν πάνω στο Dalvik, ένα virtual machine ειδικά για το Android που είναι βελτιστοποιημένο για συσκευές που τροφοδοτούνται από

μπαταρία με περιορισμένη μνήμη και υπολογιστική ισχύ.

- **Αποθήκευση:**

Χρησιμοποιείται το SQLite, μια μικρή σε απαιτήσεις σχεσιακά βάση δεδομένων για την αποθήκευση δεδομένων

Επιπλέον υπάρχει ένα ευρύ φάσμα από χαρακτηριστικά όπως το Multitasking, κλήσεις μέσω IP, εξωτερική κάρτα μνήμης κ.α.

5. Προγράμματα που χρησιμοποιήθηκαν

Για την εκπόνηση της διπλωματικής εργασίας χρησιμοποιήθηκαν 2 προγράμματα, το Eclipse Luna για τον προγραμματισμό της εφαρμογής, το Autocad 2008 για τον σχεδιασμό των γραφικών της εφαρμογής και το Adobe Photoshop για την επεξεργασία κάποιων γραφικών.

5.1. Eclipse

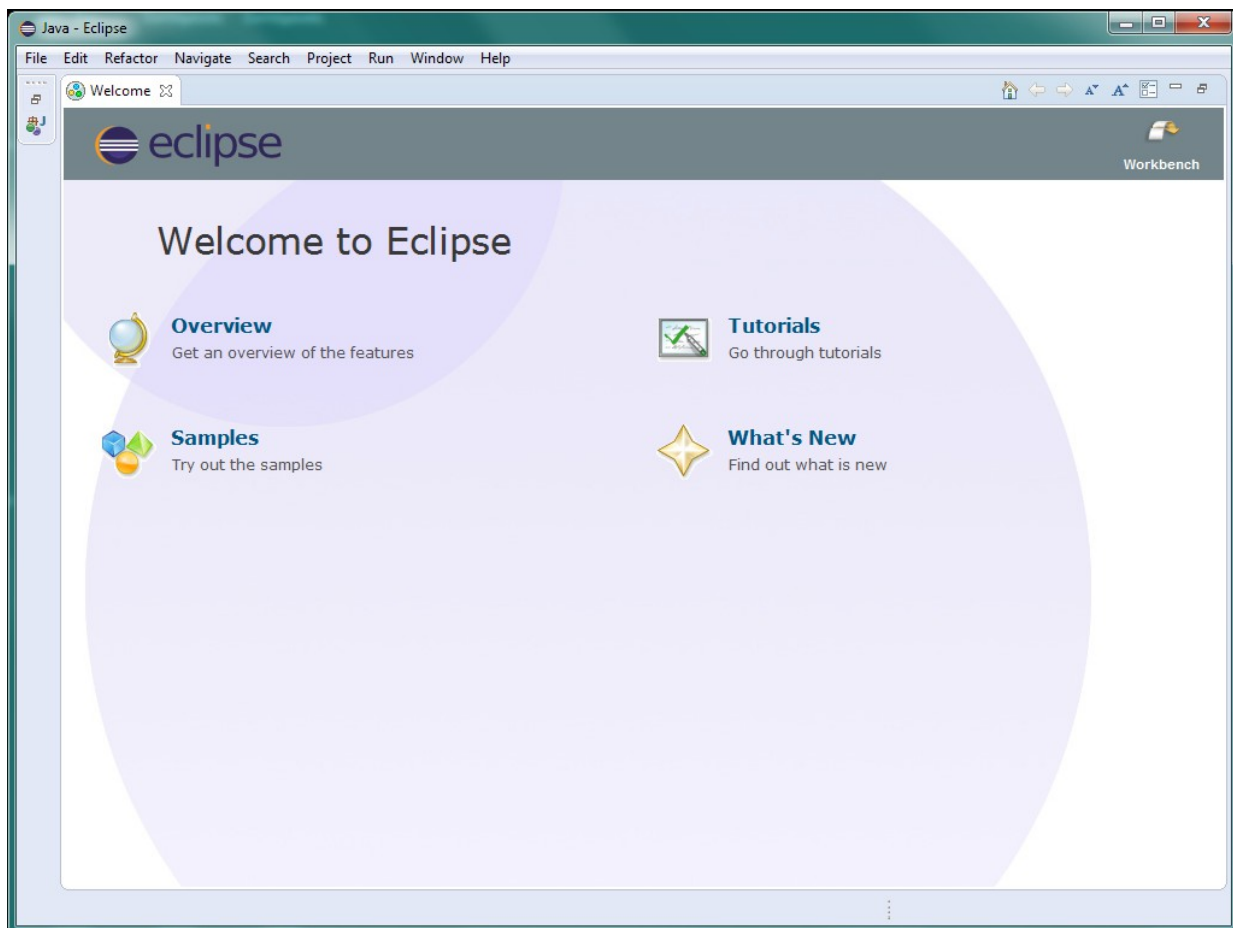
Το Eclipse αποτελεί ένα SDK (Software Development Kit), δηλαδή ένα ολοκληρωμένο περιβάλλον μέσα από το οποίο μπορούμε να γράψουμε και να εκτελέσουμε κώδικα. Το περιβάλλον αυτό καθώς και όλα τα υπόλοιπα προγράμματα που χρειάζονται για να εκτελέσουμε κώδικα σε Java ή/και C/C++ είναι ελεύθερης διανομής (freeware) και ανοικτού κώδικα (open source). Επιπλέον αποτελεί το ανερχόμενο περιβάλλον ανάπτυξης κώδικα, καθώς χρησιμοποιείται από ολοένα και περισσότερους χρήστες αλλά και εταιρείες.

5.1.1. Εισαγωγή

Το Eclipse ξεκίνησε ως ένα ανοιχτού κώδικα εργαλείο από την IBM, το οποίο όμως απέκτησε πραγματικούς “οπαδούς” μετά την δημιουργία του Eclipse Foundation το 2004. Η επιτυχία οφείλεται στο λιτό και απλό του περιβάλλον το οποίο είναι φιλικό ακόμα και στον αρχάριο προγραμματιστή αλλά κυρίως επειδή υποστηρίζει πολλές γλώσσες προγραμματισμού, από Java μέχρι C, C++, Javascript, PHP δίνοντας έτσι στο προγραμματιστή το ίδιο περιβάλλον εργασίας για τελείως διαφορετικούς κόσμους.



Εικόνα 9.1: Eclipse Luna



Εικόνα 9.2: Eclipse

Διάφορες εκδόσεις του Eclipse έχει το όνομα από διάφορα ουράνια σώματα και ειδικότερα πλανήτες ή φυσικού δορυφόρους των πλανητών. Παρακάτω φαίνονται όλες οι εκδόσεις του Eclipse και η ημερομηνία κυκλοφορίας.

Όνομα έκδοσης	Ημερομηνία Κυκλοφορίας	Έκδοση Πλατφόρμας
Austin	21 Ιουνίου 2004	3
Bravo	28 Ιουνίου 2005	3,1
Callisto	30 Ιουνίου 2006	3,2
Europa	29 Ιουνίου 2007	3,3
Galileo	24 Ιουνίου 2009	3,4
Ganymede	25 Ιουνίου 2008	3,5
Helios	23 Ιουνίου 2010	3,6
Indigo	22 Ιουνίου 2011	3,7
Juno	27 Ιουνίου 2012	3,8 και 4,2
Kepler	26 Ιουνίου 2013	4,3
Luna	25 Ιουνίου 2014	4,4
Mars	Προγραμματίστηκε για τις 24 Ιουνίου 2015	4,5

Πίνακας 2: Εκδόσεις Android

Ουσιαστικά, όπως αναφέρθηκε παραπάνω, η εφαρμογή που κατασκευάστηκε στα πλαίσια της παρούσας διπλωματικής έγινε με την χρήση του Eclipse και το plug-in που παρέχεται για τις Android εφαρμογές.

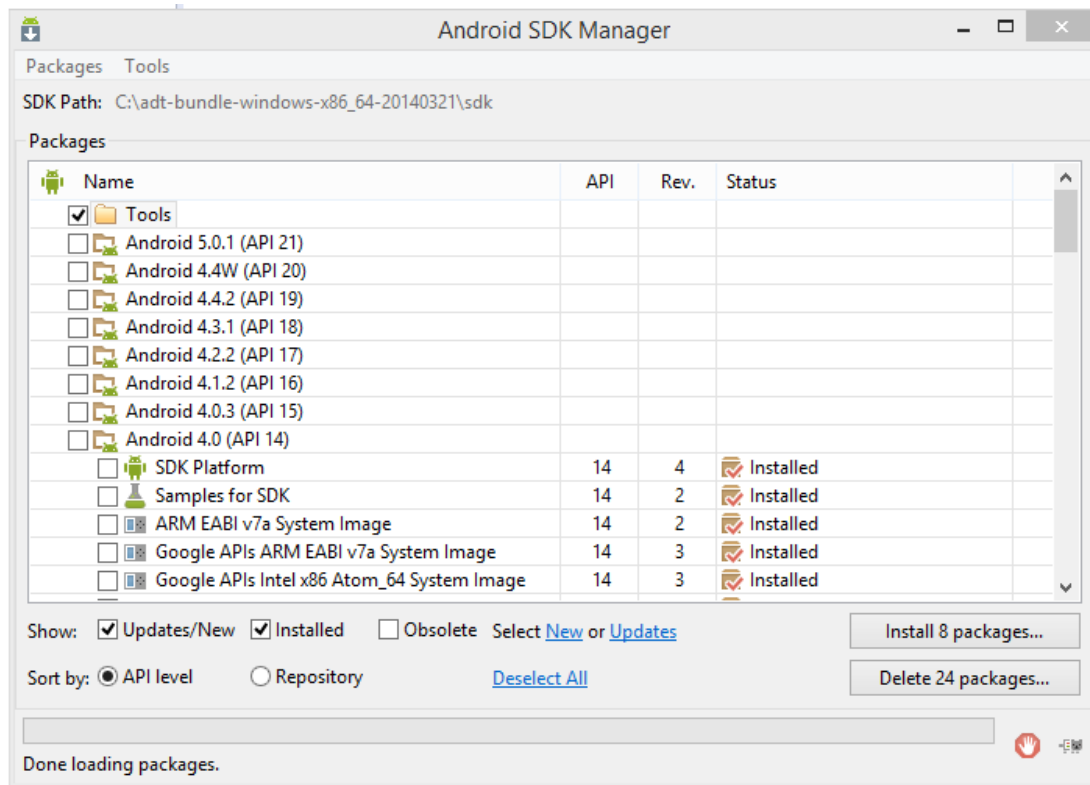
5.1.2. Android SDK

Το βασικότερο εργαλείο αποτελεί το Android SDK (Software Development Kit), το οποίο ουσιαστικά παρέχει τα επιπλέον εργαλεία ώστε να γίνει εφικτή η δημιουργία κώδικα για την κατασκευή μιας εφαρμογής Android. Η σύνδεση του Android SDK με το γραφικό περιβάλλον (Eclipse) γίνεται μέσω μιας επέκτασης (Android Development Tools ή ADT Plug-in) που γίνεται κατά την εγκατάσταση στο Eclipse.

Το Android SDK είναι ουσιαστικά το επίσημο εργαλείο της Google για τη δημιουργία εφαρμογών σε Android, το οποίο περιέχει όλες τις απαραίτητες βιβλιοθήκες από κάθε API έκδοση του Android.

Πρόκειται για το βασικό εργαλείο για τη δημιουργία εφαρμογών, όπως αναφέρθηκε πιο πάνω, την κατασκευή custom ROM, τη μεταγλώττιση του πυρήνα για την Android συσκευή, αλλά και για γενικό hacking.

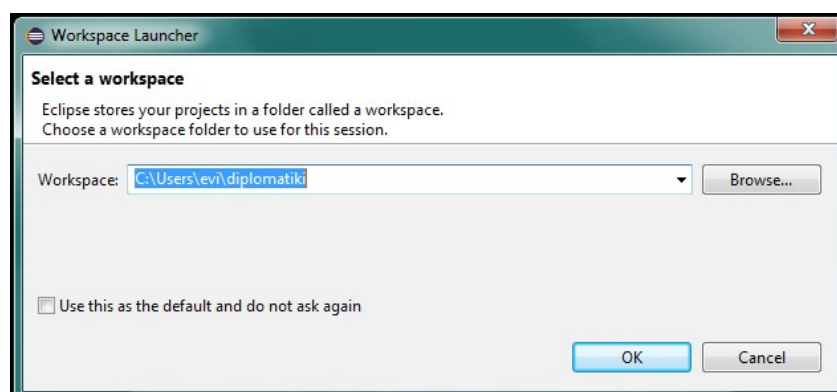
Μέσω του Android SDK χρησιμοποιούνται και εργαλεία, όπως το ADB, για τη μεταφορά αρχείων σε χώρους που κανονικά δεν επιτρέπεται, αλλά κυρίως για την κατασκευή προγραμμάτων για το Android.



Εικόνα 9: Android SDK Manager

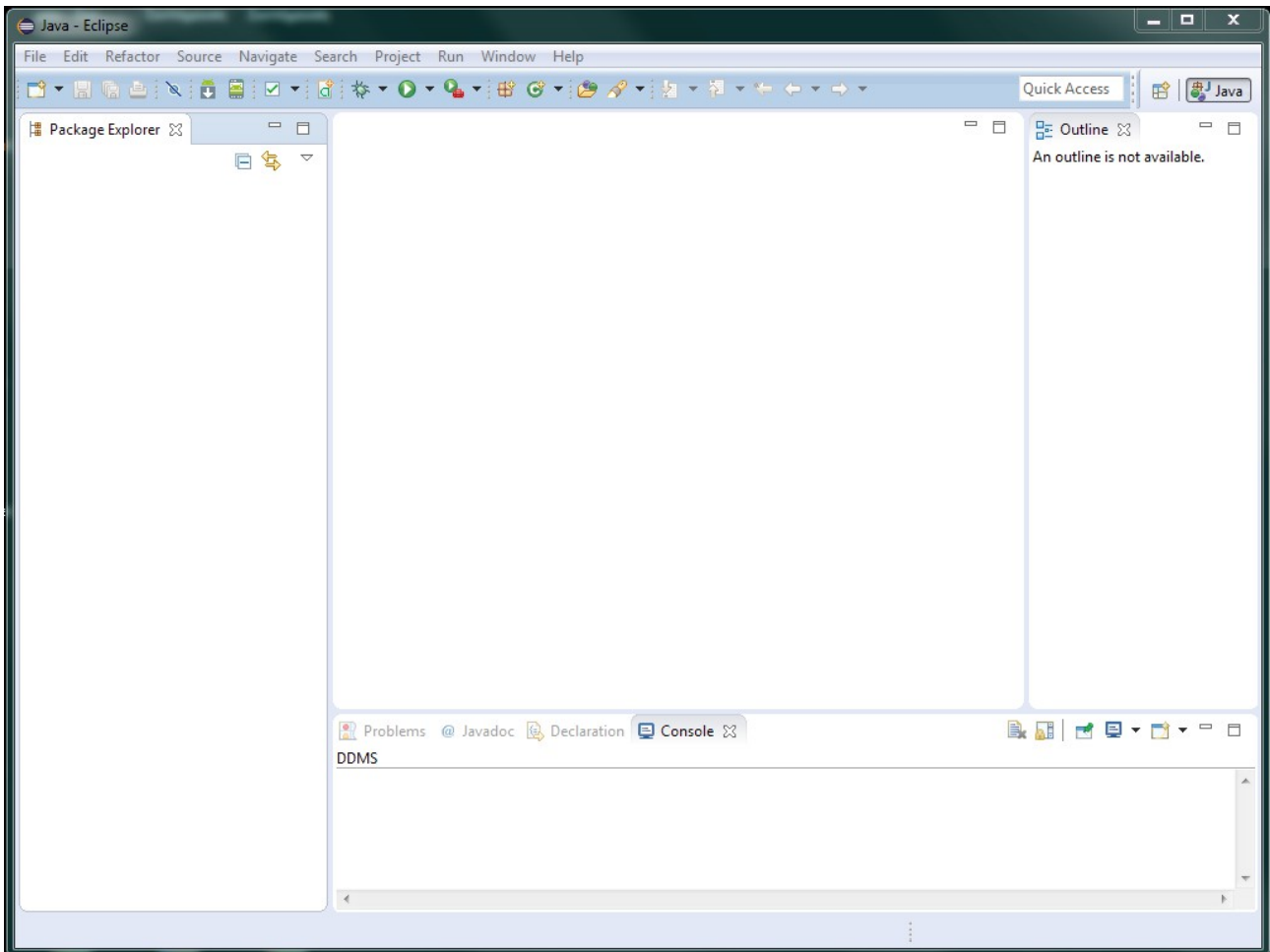
5.1.3. Το περιβάλλον του Eclipse Luna

Κατά την έναρξη του eclipse ζητείται η επιλογή ενός χώρου εργασίας, όπως φαίνεται παρακάτω.



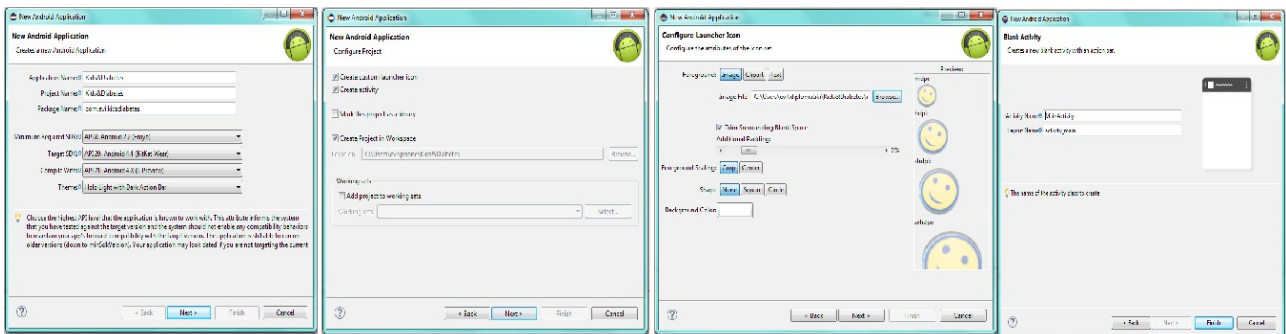
Εικόνα 10.1: Επιλογή workspace

Με την εισαγωγή του χώρου εργασίας, διακρίνεται στη συνέχεια το workbench, το οποίο πρόκειται για το περιβάλλον της επιφάνεια εργασίας για την ανάπτυξη της εφαρμογής. Το workbench είναι μοιρασμένο σε υποκατηγορίες για τον καλύτερο έλεγχο και τη διαχείριση του κώδικα.



Εικόνα 10.1: Workbench

Για τη δημιουργία ενός Android Project αρκεί η επιλογή του από την κατηγορία File. Στη συνέχεια ακολουθεί όλη η διαδικασία δήλωσης του ονόματος της εφαρμογής καθώς και του package το οποίο θα χρησιμοποιηθεί για την ομαδοποίηση των κλάσεων. Επιπλέον, χρειάζεται οι δηλώσεις του API, του εικονιδίου της εφαρμογής, αλλά και τα ονόματα της αρχικής κλάσης και του xml αρχείο, το οποίο καθορίζει το layout της κλάσης.



Εικόνα 10.3: Διαδικασία δημιουργίας Android project

Με αυτόν τον τρόπο το workbench είναι έτοιμο έτσι ώστε να αρχίσει η δημιουργία

της android εφαρμογής, όπου στα αριστερά υπάρχει το Package Explorer που περιέχει όλους τους καταλόγους που απαιτεί το project, όπως ο src που περιέχει όλες τις κλάσεις μέσα πάντα σε κάποιο πακέτο. Ένας ακόμα κατάλογος είναι ο res που περιέχει όλους τους πόρους που χρειάζεται η εφαρμογή. Αυτοί οι πόροι μπορεί να είναι κείμενο, εικόνες, ήχος. Σ'αυτόν τον κατάλογο δημιουργούνται και αποθηκεύονται όλοι οι πόροι ανάλογα με τον τύπο τους, για παράδειγμα οι εικόνες αποθηκεύονται στον /res/drawable και μπορεί να έχει κατάληξη -hdpi, -mdpi, -ldpi, -xhdpi και -xxhdpi. Τέλος υπάρχουν και τα xml αρχεία, τα οποία καθορίζουν την εμφάνιση των κλάσεων της εφαρμογής και βρίσκονται στον κατάλογο /res/layout.

Όλοι οι πόροι μεταγλωττίζονται από τον resource compiler, ο οποίος τους συμπιέζει και του "πακετάρει" δημιουργώντας μια κλάση με το όνομα R, η οποία περιέχεται στον κατάλογο gen. Η κλάση R περιέχει όλα τα αναγνωριστικά που χρησιμοποιούνται έτσι ώστε να επιτρέπεται η αναφορά σε αυτούς τους πόρους μέσα από το πρόγραμμα.

Στον κατάλογο gen βρίσκεται η κλάση R, όπως αναφέρθηκε παραπάνω, και ότι συμβαίνει με τον κατάλογο /src έτσι και αυτή η κλάση βρίσκεται κάτω από ένα πακέτο που έχει οριστεί. Είναι σημαντικό να σημειωθεί ότι η κλάση R διαχειρίζεται αυτόματα από το Android plug-in του Eclipse. Κάθε φορά που εισάγεται ένα αρχείο οπουδήποτε στον κατάλογο /res το plug-in παρατηρεί την αλλαγή και προσθέτει τα κατάλληλα αναγνωριστικά id των πόρων στο αρχείο R.java.

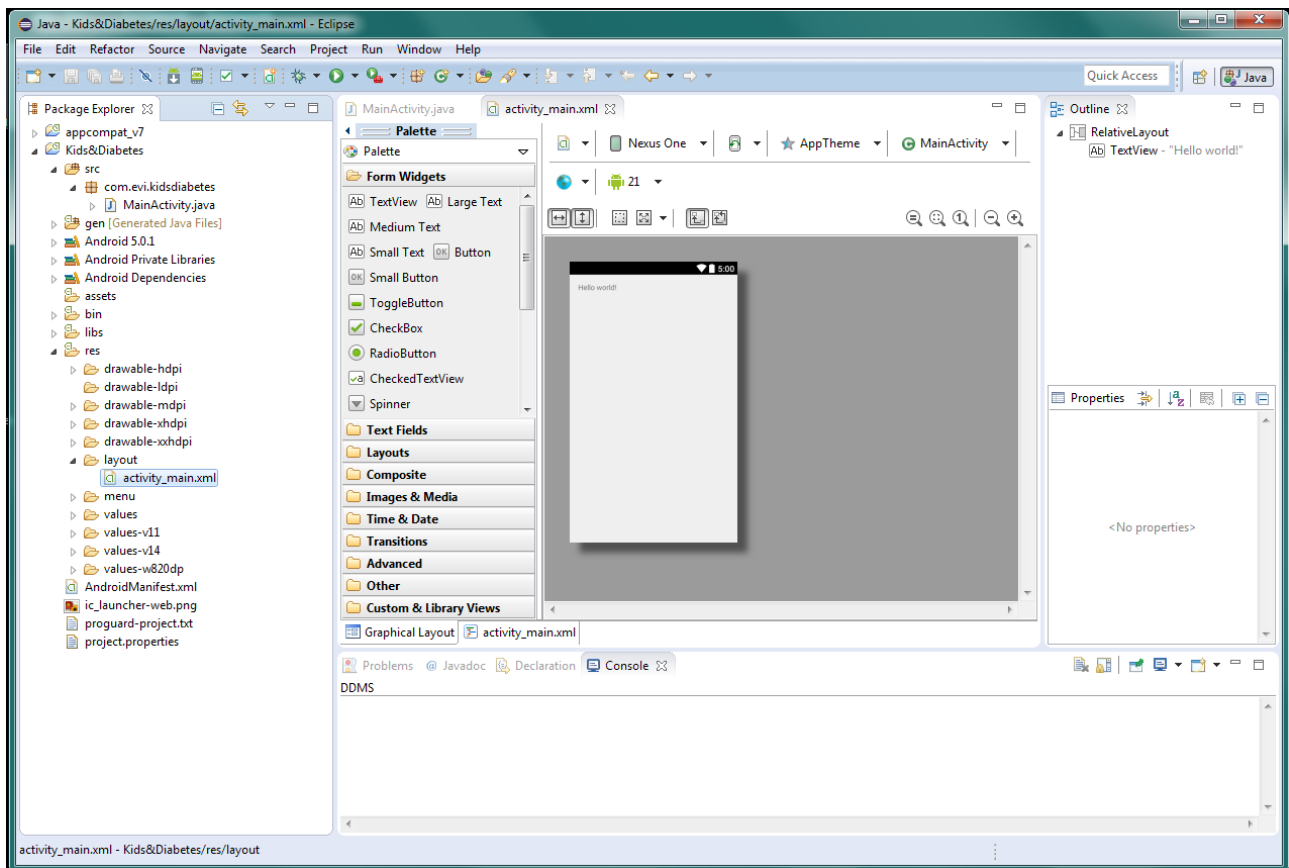
Κάθε εφαρμογή πρέπει να έχει και ένα αρχείο Manifest.xml. Στο αρχείο αυτό παρουσιάζονται όλες οι απαραίτητες πληροφορίες, τις οποίες χρειάζεται το σύστημα πριν μπορέσει να εκτελέσει οποιοδήποτε κώδικα της εφαρμογής. Μεταξύ άλλων το AndroidManifest εκτελεί τα ακόλουθα:

- Ονομάζει το Java πακέτο της εφαρμογής, του οποίου το όνομα λειτουργεί ως μοναδικό αναγνωριστικό id για την εφαρμογή.
- Περιγράφει τις συνιστώσες της εφαρμογής, για παράδειγμα τα activities, services (υπηρεσίες), content providers (παρόχους περιεχομένου), κτλ.
- Ονομάζει τις κλάσεις που εφαρμόζουν καθεμιά από τις συνιστώσες και δημοσιεύει τις ικανότητές τους, για παράδειγμα ποια intents μπορούν να χειριστούν.
- Οι δηλώσεις αυτές αφήνουν το σύστημα Android να γνωρίζει ποιες είναι οι συνιστώσες και υπό ποιες συνθήκες να ενεργοποιούνται.
- Δηλώνει ποια δικαιώματα η εφαρμογή πρέπει να έχει προκειμένου να γίνει δυνατή η πρόσβαση σε προστατευμένες περιοχές του API και μπορεί να αλληλεπιδρά με στοιχεία της εφαρμογής.
- Δηλώνει τα δικαιώματα που οι άλλοι οφείλουν να έχουν προκειμένου να

αλληλεπιδράσουν με στοιχεία της εφαρμογής.

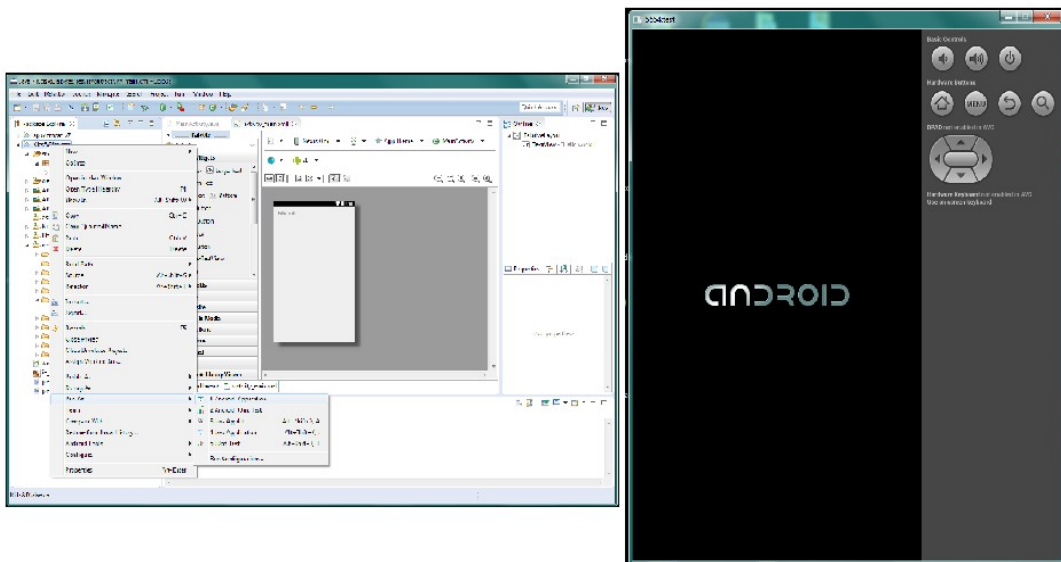
- Δηλώνει το ελάχιστο επίπεδο του Android API που απαιτεί η εφαρμογή.

Παρακάτω να αναλυθεί περαιτέρω το Manifest.xml της παρούσας διπλωματικής.



Εικόνα 10.4: Workbench του Android Project

Τέλος, αφού ολοκληρωθεί ο προγραμματισμός της εφαρμογής, υπάρχει η δυνατότητα ο προγραμματιστής να μπορέσει να δει τα αποτελέσματα του κώδικα του, μέσω emulator, χωρίς να χρειάζεται να γίνεται εγκατάσταση σε κινητή συσκευή, το οποίο θα είναι χρονοβόρο αν χρειάζεται οποιαδήποτε αλλαγή ή έλεγχος και διόρθωση του κώδικα. Ουσιαστικά, ο android emulator, Android Virtual Device (AVD), είναι μια εικονική συσκευή, όπου προσομοιώνεται μια συσκευή android, της οποίας γίνονται οι απαραίτητες ρυθμίσεις όσον αφορά την ανάλυση της συσκευής, την χωρητικότητα, το API που θα υποστηρίζει κ.α..



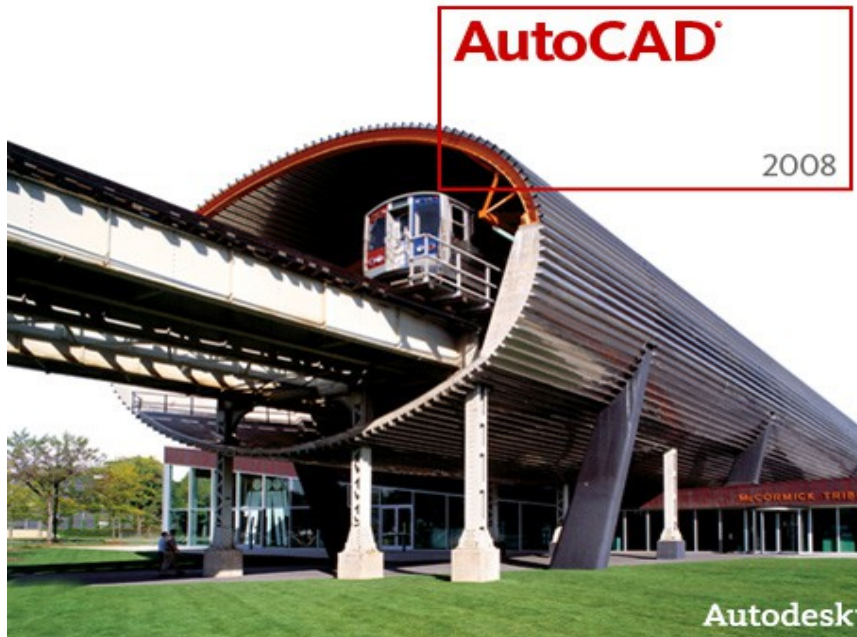
Εικόνα 10.5: Run as Android Application

5.2. Δημιουργία γραφικών

Για τη δημιουργία των γραφικών της εφαρμογής χρησιμοποιήθηκε το AutoCAD για όλο τον σχεδιασμό και το Photoshop για την επεξεργασία μερικών σκίτσων που δημιουργήθηκαν στο AutoCAD.

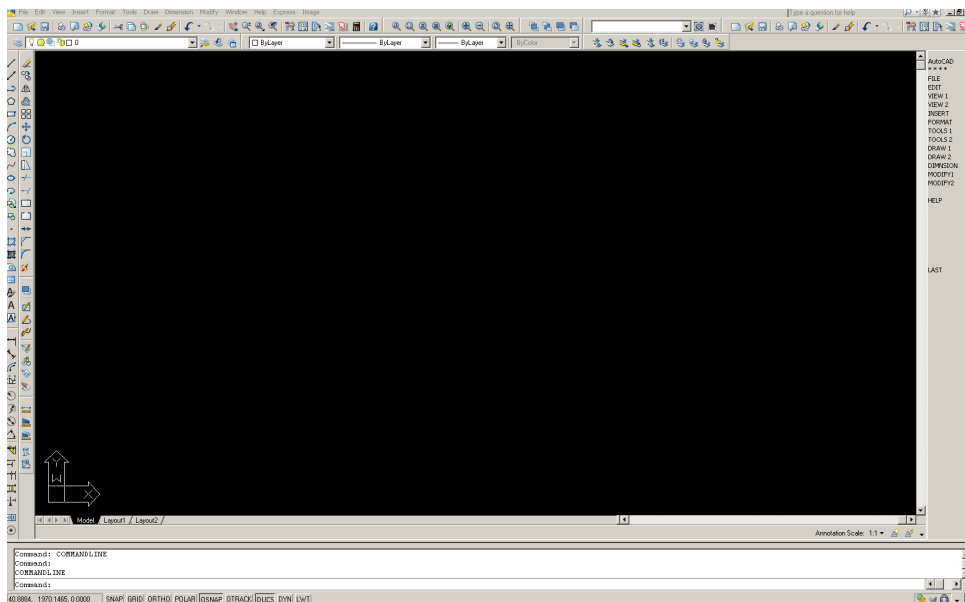
5.2.1. Autocad

Τα γραφικά της εφαρμογής δημιουργήθηκαν στο πρόγραμμα Autocad. Το AutoCAD είναι το πιο διαδεδομένο πρόγραμμα CAD σε παγκόσμιο επίπεδο που απευθύνεται σε Αρχιτέκτονες, πολιτικούς μηχανικούς, διακοσμητές, σχεδιαστές, κ.λ.π. Διαθέτει εργαλεία και δυνατότητες για σχεδιασμό σε δύο ή τρεις διαστάσεις. Επιπλέον, ένα από τα βασικά του πλεονεκτήματα και η προτίμηση του συγκεκριμένου προγράμματος είναι η εύκολη σχεδίαση και επεξεργασία, αλλά και η παλαιότερη γνώση χρήσης του συγκεκριμένου προγράμματος.



Εικόνα 11.1: AutoCAD

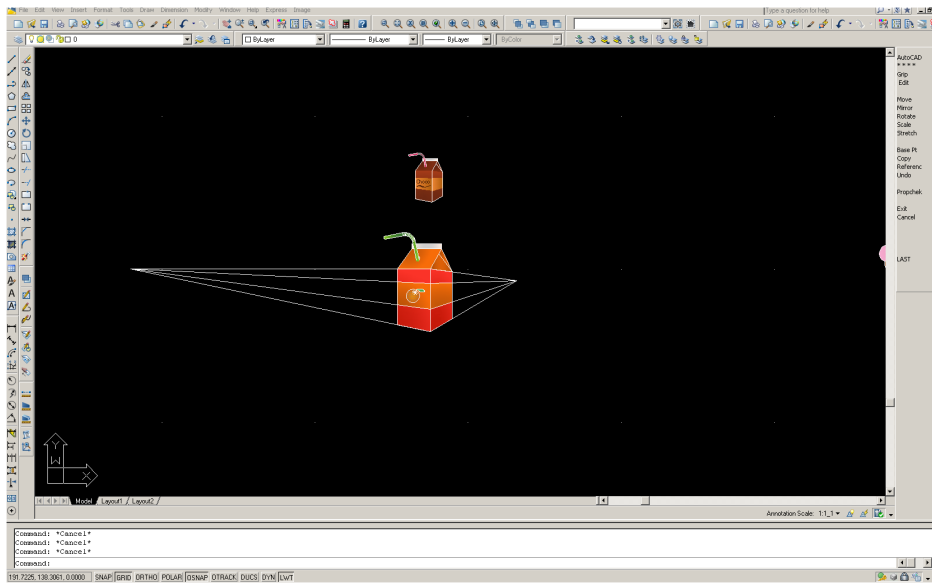
Τα αρχεία που εξάγει το AutoCAD είναι με κατάληξη .dwg. Το περιβάλλον του AutoCad φαίνεται παρακάτω (Εικόνα 11.2), το οποίο ουσιαστικά πρόκειται για μια επιφάνεια εργασίας η οποία συνδυάζει τις διάφορες επιλογές για την χάραξη γραμμών και σχεδίων, αλλά και ένα παράθυρο στο οποίο μπορούν να γίνουν οι εισαγωγές των εντολών ώστε να υπάρχει το επιθυμητό αποτέλεσμα.



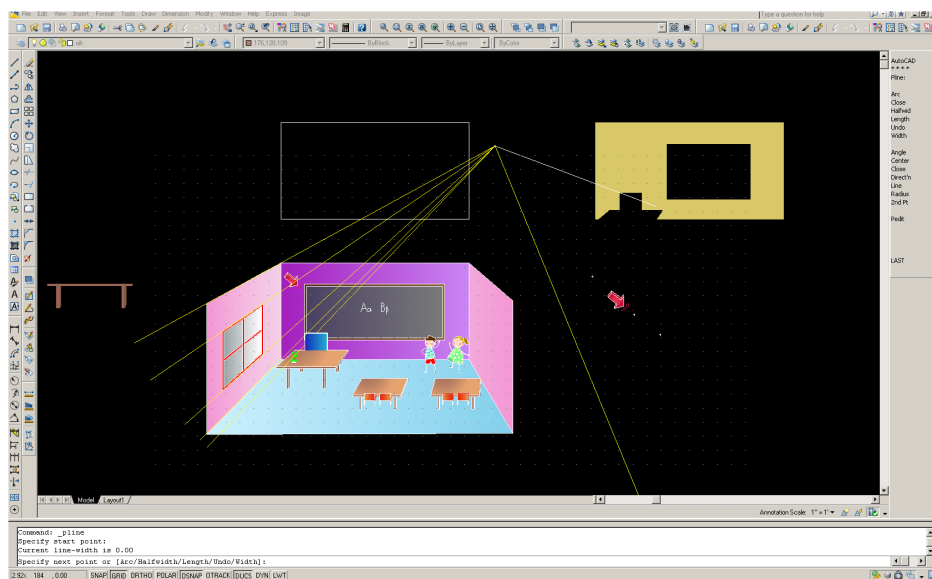
Εικόνα 11.2: Περιβάλλον AutoCAD

Σ'αυτό το σημείο θα πρέπει να τονιστεί ότι προτιμήθηκε η σχεδίαση δισδιάστατων σχεδίων και όχι τρισδιάστατων (3D), διότι στα 2D σχέδια η οποιαδήποτε αλλαγή γίνεται

πολύ πιο εύκολα καθώς επίσης τα τελικά αρχεία είναι πολύ μικρότερα σε μέγεθος απ'ότι τα 3D σχέδια. Παρόλο αυτά όμως, αν και δισδιάστατα, τα σχέδια σχεδιάστηκαν προοπτικά με σημεία φυγής (Εικόνες 11.3.1 και 11.3.2) ώστε να δίνει την εικόνα τρισδιάστατου.



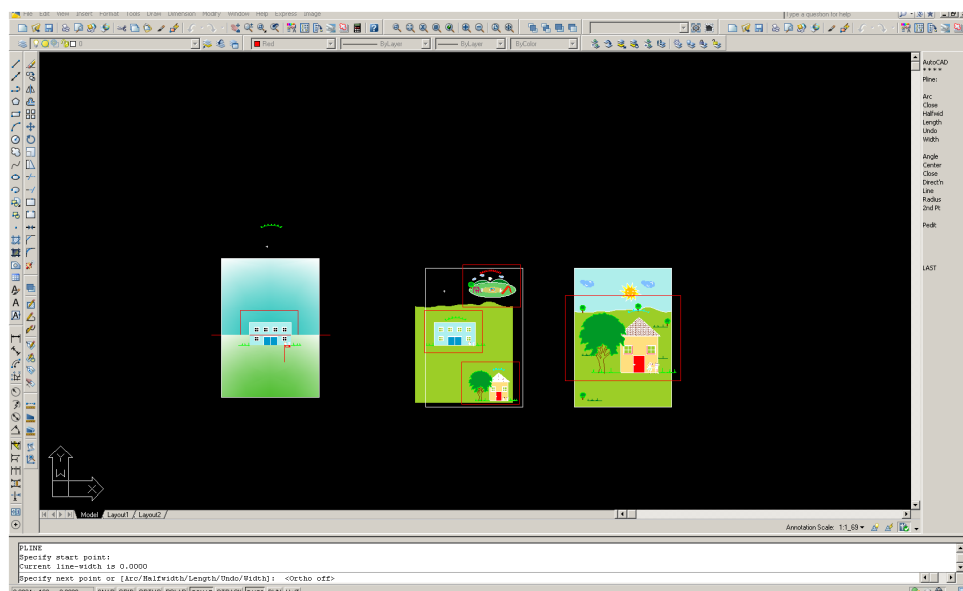
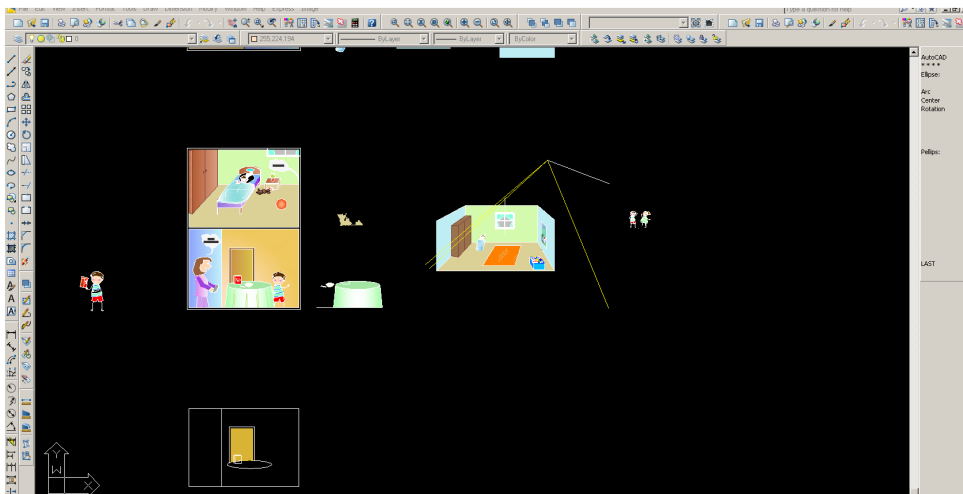
Εικόνα 11.3.1: Σχεδίασμος προοπτικά με σημεία φυγής



Εικόνα 11.3.2: Σχεδίασμος προοπτικά με σημεία φυγής

Από τις διαθέσιμες εντολές που παρέχει το AutoCAD χρησιμοποιήθηκε η πλειοψηφία τους ,καθώς και ο συνδυασμός αυτών. Παράδειγμα αυτών των εντολών είναι τα περιγράμματα των χαρακτήρων, των αντικειμένων και των διαφόρων εσωτερικών και εξωτερικών χώρων. Περισσότερη χρήση έγινε στις εντολές line(γραμμή), polyline (πολυγραμμή), arc (τόξο), ellipse έλλειψη , circle (κύκλος) κλπ. Περισσότερο βέβαια η

εντολή polyline γιατί είναι πιο επεξεργάσιμο στοιχείο από ότι η απλή γραμμή. Η εντολή line έχει τον περιορισμό ότι πρέπει να γίνει μετατροπή σε polyline έτσι ώστε να μπορέσει να πραγματοποιηθεί οποιαδήποτε επεξεργασία.



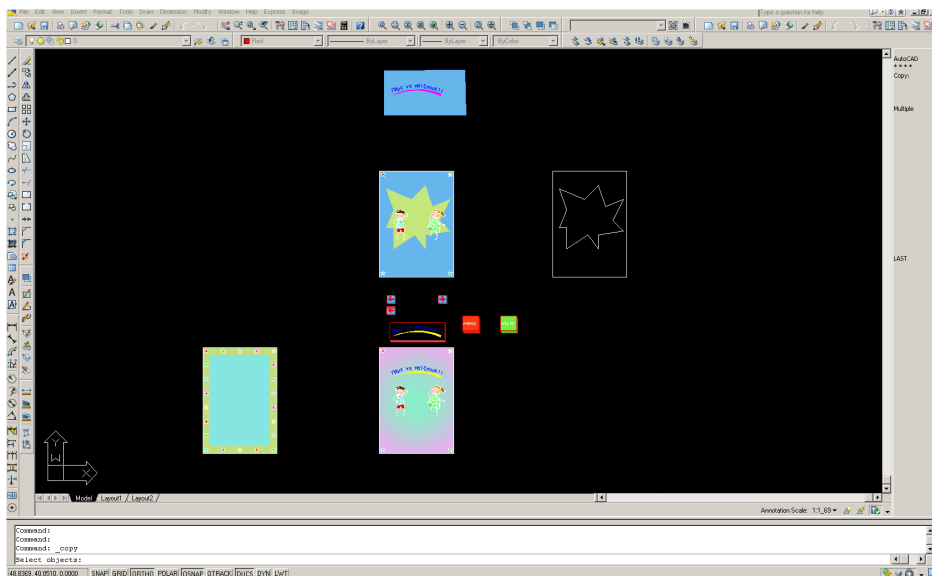
Εικόνα 11.5.2: Ενσωμάτωση πολλών σκίτσων σε ένα

Για την επεξεργασία αυτών των γραμμών, την σχεδιαστική διευκόλυνση π.χ. του πάχους της γραμμής, της ένωσης μεταξύ τους χρησιμοποιήθηκαν οι εντολές edit polyline (επεξεργασία πολυγραμμής), edit, join (ένωση), και spline (καμπύλη πολυγραμμή) γραμμή για να δοθεί καμπυλότητα σε γωνίες σε όλη την γραμμή, ενώ τέλος η εντολή edit spline για την επεξεργασία της.

Για την αναπαραγωγή σχημάτων, διόρθωση αυτών, αφαίρεσης, μετακίνησης κ.λ.π. από το modify menu (μενού για τροποποιήσεις) , οι εντολές copy (αντιγράψω), move (μετακίνηση), erase (διαγραφή), mirror για αντικατοπτρικό είδωλο, trim για “κόψιμο” γραμμών, extend για επέκταση γραμμών , offset για παράλληλη επανάληψη, fillet για

καμπύλωμα γωνιών, αλλά και πολλές ακόμη.

Μία ακόμη χρήσιμη εντολή το scale (κλίμακα), η οποία διευκόλυνε να αυξομειωθούν τα γραφικά για να έχουν την ίδια κλίμακα όταν γινόταν η εισαγωγή τους στο τελικό σχέδιο που έπρεπε να βρίσκονται. Αυτό συνέβη γιατί τα αρχεία χωρίστηκαν από την αρχή σε ομάδες έτσι ώστε να μη γίνει ένα μεγάλο αρχείο, βαρύ και δυσλειτουργικό. Για παράδειγμα, δημιουργήθηκε ξεχωριστό αρχείο για τη δημιουργία των χαρακτήρων και όλων των ανθρώπινων φιγούρων, διαφορετικό αρχείο για το σπίτι και τους εσωτερικούς χώρους (δωμάτιο, κουζίνα, γραφείο, κ.α.), ένα αρχείο dwg για τη δημιουργία των backgrounds (αρχική σελίδα, σχολείο, παιδική χαρά, κ.α.) και ένα άλλο για τον σχεδιασμό των τροφίμων ένα προς ένα. Τέλος, δημιουργήθηκε ένα αρχείο για τον σχεδιασμό μιας ιστορίας με την διαδικασία που ακολουθεί ένα παιδί με διαβήτη (παρακάτω θα δοθεί εξήγηση). Κάποια από αυτά, λοιπόν, έπρεπε να ενωθούν μεταξύ τους και με την χρήση της εντολής scale τα διάφορα σκίτσα μπορούσαν να μικρύνουν ή να μεγαλώσουν έτσι ώστε να ενσωματωθούν μεταξύ τους.

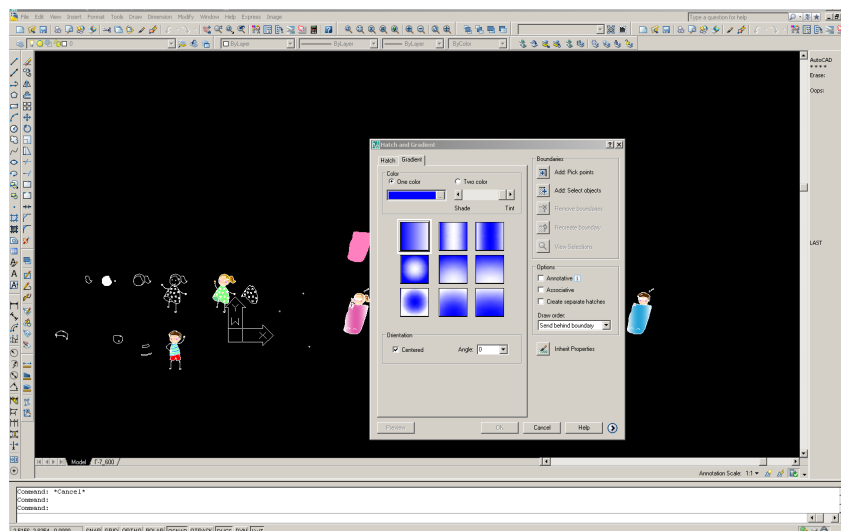


Εικόνα 11.5.1: Ενσωμάτωση πολλών σκίτσων σε ένα

Για να γίνει εισαγωγή ενός σκίτσου μέσα σε ένα άλλο μόλις ολοκληρωνόταν, γινόταν η χρήση της εντολής insert block (εισαγωγή), έτσι ώστε το κάθε αντικείμενο να είναι μία οντότητα και να μετακινείται, να επαναλαμβάνεται, να αυξομειώνεται εύκολα και για οποιαδήποτε αλλαγή γινόταν, να ενημερώνεται το τελικό σχέδιο άμεσα.

Για τον χρωματισμό όλων αυτών που σχεδιάστηκαν εφαρμόστηκε η εντολή hatch (γέμισμα με διαγράμμιση). Βέβαια η εντολή για να έχει αποτελέσματα πρέπει να ανιχνεύει κλειστά περιγράμματα. Έτσι λοιπόν η δυσκολία της σχεδίασης ήταν αυτή, κάποια από τα περιγράμματα έπρεπε να γίνουν πολλές φορές προσεκτικά για να εφαρμόζεται το hatch.

Συνήθως όταν χρησιμοποιείται η εντολή spline δεν ανιχνεύεται εύκολα το περίγραμμα και μπερδεύεται με άλλες γραμμές. Η λύση ήταν να γίνεται copy το προβληματικό περίγραμμα χωριστά, να γεμίζει με διαγράμμιση και μετά να μεταφέρεται η διαγράμμιση στην αρχική της θέση. Πολλές φορές όταν σχεδιάζονταν μακριά από το 0,0 η διαγράμμιση ήταν προβληματική με κενά.



Εικόνα 11.6: Εφαρμογή hatch

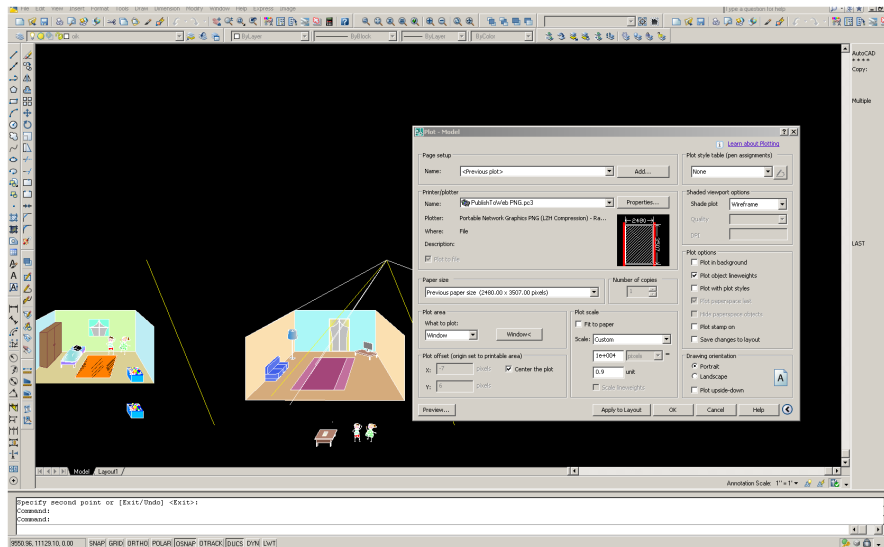
Δυστυχώς αυτά τα σχεδιαστικά προγράμματα έχουν τα αποκαλούμενα bugs, τα οποία δυσκολεύουν την σχεδίαση. Παρόλο αυτά όμως, είναι ένα πρόγραμμα που καλύπτει μεγάλο φάσμα για σχεδιαστικές απαιτήσεις.

Για να δοθεί η αίσθηση της σκίασης και του τρισδιάστατου εφαρμόστηκε το υπομενού Gradient της εντολής hatch. Όλες οι διορθώσεις και οι αλλαγές αυτών των διαγραμμίσεων, γεμισμάτων πραγματοποιήθηκαν με την εντολή Edit Hatch.

Το περίγραμμα των περισσότερων στοιχείων αφού σχεδιάστηκε μεταφέρθηκε στο layer (επίπεδο) defpoints ένα επίπεδο που είναι ορατό στην οθόνη αλλά δεν εκτυπώνεται και τα σχέδια είναι πιο θελκτικά στα μικρά παιδιά που θα είναι οι χρήστες της εφαρμογής.

Επιπλέον χρησιμοποιήθηκαν οι εντολές Dtext για απλό κείμενο και arc-aligned text των express tools του Autocad για γραφή κειμένου με καμπυλότητα.

Στο τέλος κάθε εικόνα λοιπόν, μόλις ετοιμαζόταν, έπρεπε να μετατραπεί σε αρχείο .png σε τρεις σχεδιαστικές κλίμακες για να βρεθεί το κατάλληλο σε μέγεθος για να μπορέσει να κουμπώσει στον κώδικα. Με την εντολή plot (σχεδίαση, εκτύπωση) δημιουργούνταν το αρχείο .png.



Εικόνα 11.7: Εξαγωγή αρχείου .png

5.2.2. Photoshop

Το Adobe Photoshop, ή απλά Photoshop είναι πρόγραμμα επεξεργασίας γραφικών που αναπτύχθηκε και δημοσιεύθηκε από την Adobe Systems και έχει ηγετική θέση στην σημερινή αγορά εμπορικών προγραμμάτων επεξεργασίας εικόνας. Τα βασικότερα χαρακτηριστικά αυτού του προγράμματος είναι ότι ο χρήστης έχει τη δυνατότητα να χρησιμοποιεί επίπεδα στην εικόνα που επεξεργάζεται. Αυτό σημαίνει πως μπορεί να προσδώσει σε κάθε αντικείμενο ξεχωριστές ιδιότητες και να το επεξεργάζεται ανεξάρτητα από τα υπόλοιπα.



Εικόνα 12.1: Photoshop CS6

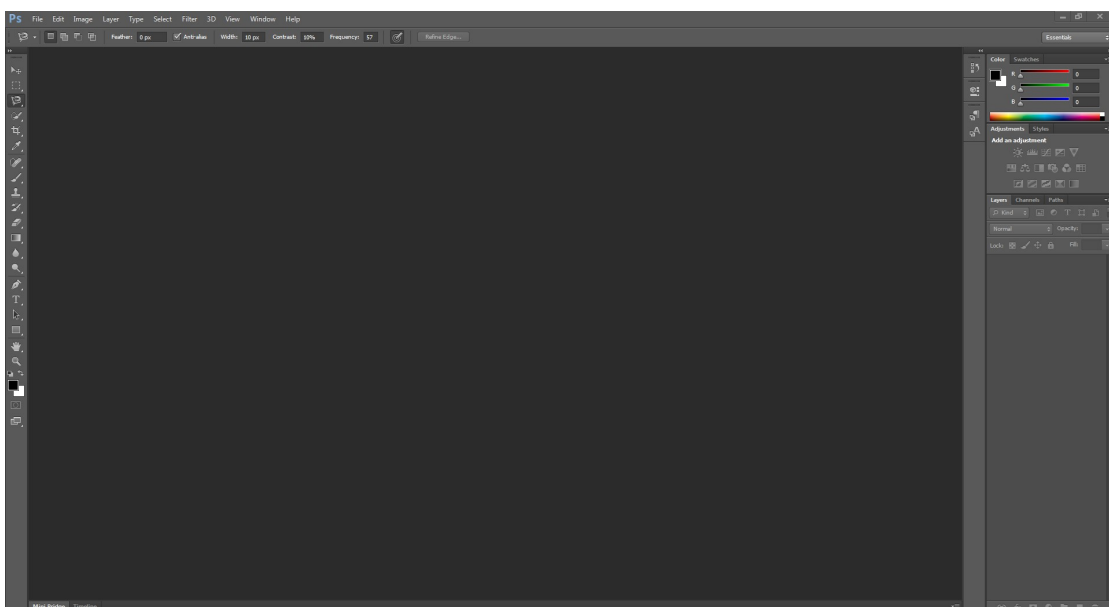
Υπάρχουν δύο τρόποι με τους οποίους μια εικόνα γίνεται κατανοητή από έναν Η/Υ: σαν ένα καρτεσιανό επίπεδο όπου η κάθε ψηφίδα (x,y) έχει κάποιο χρώμα, ή σαν ένα

σύνολο από μαθηματικές εξισώσεις που περιγράφουν επιφάνειες και γραμμές. Τα πρώτα λέγονται ψηφιογραφικά γραφικά (Bitmap / Raster graphics) ενώ τα δεύτερα διανυσματικά (Vector graphics).

Το Photoshop δίνει τη δυνατότητα δημιουργίας φωτορεαλιστικών εικόνων προσφέροντας μεγάλο φάσμα χρωμάτων, μεγάλο επίπεδο λεπτομέρειας και σκιάσεων, ειδικά εφέ, μεγάλη ποικιλία φίλτρων , δυνατότητα εξαγωγής (save as) των εικόνων επιλέγοντας από μία μεγάλη γκάμα τύπων μορφοποίησης, την ιδανική για την κάθε περίπτωση. Συγκεκριμένα οι ανάγκες ήταν δημιουργία εικόνων τύπου :

- .jpg: πρόκειται για μια διαδεδομένη μέθοδο συμπίεσης που χρησιμοποιεί πολλές τεχνικές
- .png: για παραγωγή εικόνων που περιέχουν διαφανείς περιοχές τις οποίες και διατηρούν
- .psd: μορφή αρχείου που διατηρεί όλα τα επίπεδα, μάσκες, κανάλια και τις επιλογές που έχουν δημιουργηθεί σε μία εικόνα.

Το Adobe Photoshop είναι το πιο δημοφιλές πρόγραμμα για τη δημιουργία και τη μετατροπή εικόνων στο web. Υποστηρίζει τα ακόλουθα χρωματικά μοντέλα : RGB, LAB, CMYK, Grayscale, Bitmap και Duotone. Ο χώρος εργασίας περιλαμβάνει μενού και μια ποικιλία εργαλείων και παλετών για την προβολή, επεξεργασία και προσθήκη στοιχείων στις εικόνες σας. Τελευταία έκδοση του προγράμματος είναι Photoshop CS6 που είναι εμπλουτισμένη με πολλά καινούρια χαρακτηριστικά και δυνατότητες. Η συντομογραφία «CS6» σημαίνει ότι το προϊόν έχει ενσωματωθεί στην πέμπτη έκδοση του λογισμικού πακέτου «Adobe Creative Suite».

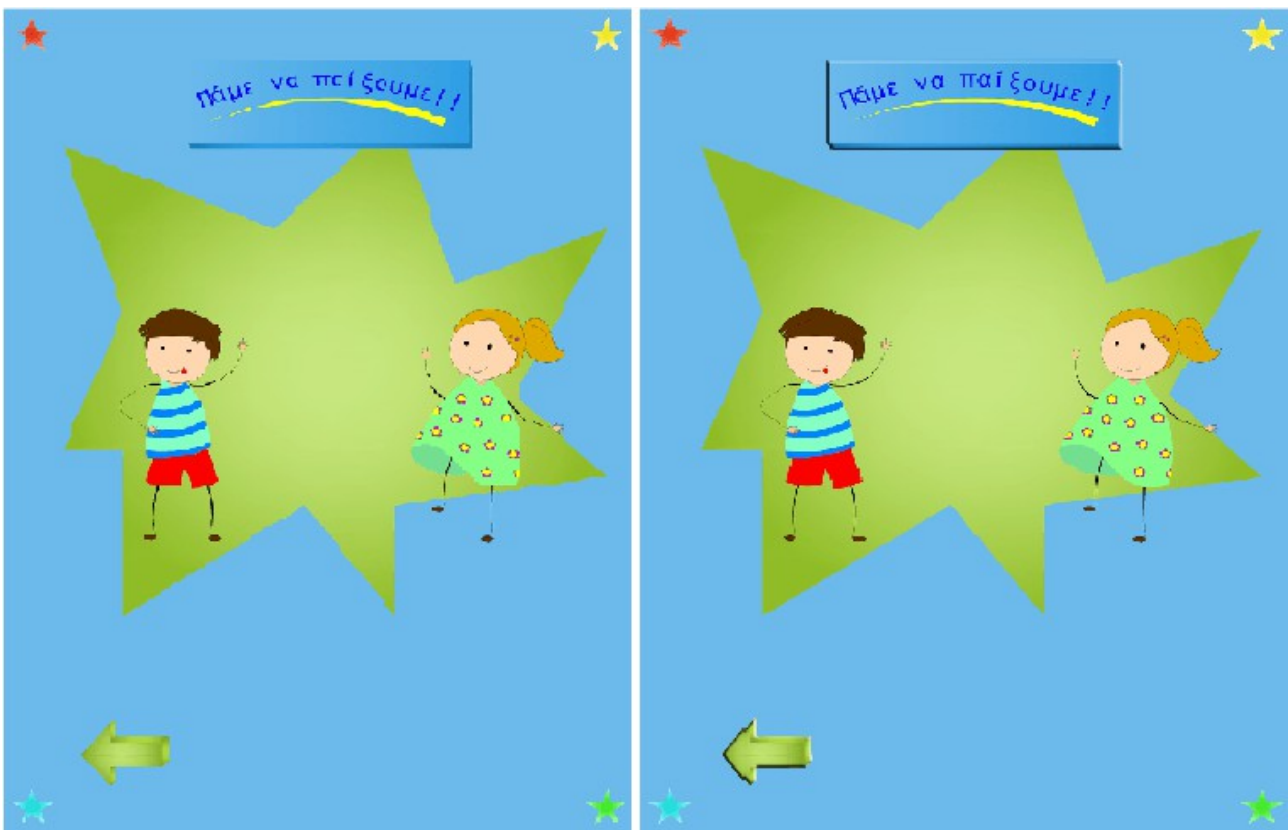


Εικόνα 12.2: Περιβάλλον του Photoshop Extended

Τα βασικά στοιχεία στο περιβάλλον εργασίας του Photoshop CS6 είναι τα εξής:

- **Μενού επιλογών :** Θεματικά οργανωμένοι κατάλογοι επιλογών.
- **Μπάρα εργαλείων (options bar):** Ρυθμίσεις που διαμορφώνουν τον τρόπο λειτουργίας του επιλεγμένου εργαλείου.
- **Εργαλειοθήκη (tools bar):** Η βασική συλλογή εργαλείων του Photoshop.
- **Παράθυρο εικόνας :** Το παράθυρο στο οποίο παρουσιάζεται η προς επεξεργασία εικόνα.
- **Παλέτες:** Θεματικά οργανωμένες ομάδες χαρακτηριστικών που σχετίζονται και συμπληρώνουν τη βασική εργαλειοθήκη.
- **Γραμμή κατάστασης (status bar):** Η γνωστή σε κάθε παράθυρο γραμμή, με συμπληρωματικές πληροφορίες σχετικά με το περιβάλλον εργασίας του Photoshop.

Το Photoshop χρησιμοποιήθηκε κυρίως για την εισαγωγή φίλτρων σε ορισμένα από τα γραφικά που δημιουργήθηκαν και αναφέρθηκαν πιο πάνω. Συγκεκριμένα χρησιμοποιήθηκε το Contour, έτσι ώστε κάποια από τα αντικείμενα να φαίνονται σαν κουμπί και με αυτόν τον τρόπο να είναι πιο εύκολο στο παιδί να το καταλάβει.



Εικόνα 12.3: Εισαγωγή Contour

Μια ακόμη επεξεργασία που υπέστησαν οι εικόνες ήταν και η αφαίρεση του άσπρου background στα τρόφιμα. Αρχικά επιλεγόταν το αντικείμενο με τη βοήθεια του magnetic lasso tool, το οποίο αναγνωρίζει αυτόματα τις γραμμές του αντικειμένου και μόλις φτάσει στο σημείο απ' όπου ξεκίνησε το selection, η επιλογή αυτή "κλείνει". Μ'αυτόν τον τρόπο, διαχωρίζεται το αντικείμενο από το background, το οποίο μετά εύκολα μπορεί να διαγραφεί.

6. Εισαγωγή στην εφαρμογή

Έπειτα από μεγάλη και χρονοβόρα έρευνα διαπιστώθηκε ότι δεν υπάρχει ελληνική εφαρμογή, η οποία να σχετίζεται με τον παιδικό διαβήτη. Για τον λόγο επιλέχθηκε για γίνει μια εφαρμογή που να αποσκοπεί στην εκμάθηση του διαβήτη μέσα από ευχάριστα γραφικά και απλά παιχνίδια. Η μεγαλύτερη δυσκολία, όμως, παρουσιάστηκε όταν έπρεπε να επιλεγούν ακριβώς τα παιχνίδια και γενικά ο τρόπος που μπορεί μια τέτοια εφαρμογή να προσεγγίσει τα παιδιά.

Υπήρχε η ιδέα της εισαγωγής ενός χαρακτήρα, ο οποίος θα λαμβάνει την τιμή της γλυκόζης που θα εισάγει το παιδί, με τη βοήθεια του γονέα, και αναλόγως τα αποτελέσματα θα προτείνει την χορήγηση ινσουλίνης. Η συγκεκριμένη ιδέα, όμως, απορρίφθηκε κατευθείαν, διότι κάθε παιδί λαμβάνει διαφορετική δοσολογία ινσουλίνης και με διαφορετική συχνότητα μέσα στην ημέρα, αλλά και όλες οι εφαρμογές που κυκλοφορούν στο Google Play Store και στο IOS store είναι δουλειά μιας ολόκληρης ομάδας από γιατρούς, παιδοψυχολόγους και τους αρμόδιους που θα αναπτύξουν μια τέτοια εφαρμογή. Οπότε, η επικρατέστερη ιδέα ήταν τελικά, ένα παιδί με διαβήτη μέσα από την εφαρμογή να μπορέσει να μάθει τη επιτρεπόμενη διατροφή, έτσι ώστε να μπορεί στην συνέχεια αλλά και με την πάροδο του χρόνου να είναι σε θέση να καταλάβει τις διατροφικές επιλογές του, καθώς και να μπορέσει από μόνο να διαχειριστεί το θέμα του παιδικού διαβήτη και της διατροφής που θα πρέπει να ακολουθήσει ώστε να έχει όσο το δυνατό μια φυσιολογική ζωή, χωρίς στερήσεις.

6.1. Περιβάλλον – Περιγραφή Εφαρμογής

Για τους παραπάνω λόγους, η επιλογή του περιβάλλοντος έπρεπε να γίνει πολύ προσεκτικά, όπως και – πράγματι- έγινε με την εισαγωγή ενός περιβάλλοντος όπου το

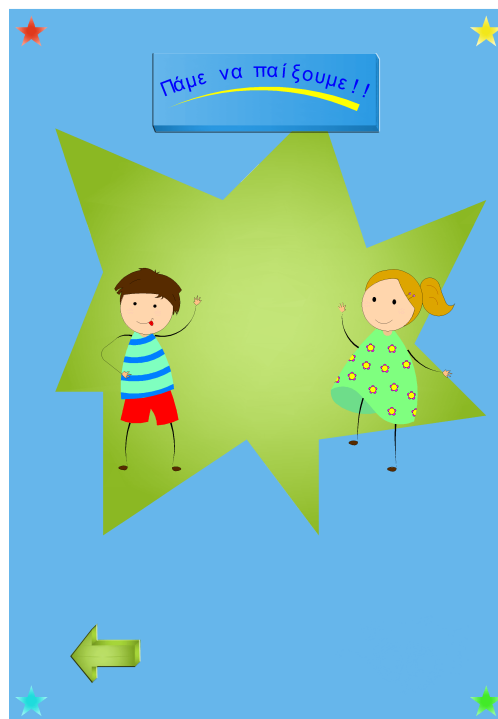
παιδί περνάει την καθημερινότητα του.

Το σπίτι, το σχολείο, αλλά και η παιδική χαρά αποτελούν την καταλληλότερη επιλογή, διότι με αυτόν τον τρόπο ένα παιδί μπορεί να μάθει να διαχειρίζεται όλες τις καταστάσεις κατά τη διάρκεια της ημέρας ανεξάρτητα από το που βρίσκεται.

Μέσα στο σπίτι υπάρχουν τα 4 βασικά δωμάτια, όπου το παιδί μπορεί να παίξει και να μάθει σχετικά με τη διατροφή του, νιώθοντας ότι βρίσκεται σε γνωστό και οικείο περιβάλλον όπου μπορεί εύκολα να εξοικειωθεί και να νιώσει άνετα.

Στο σχολείο μπορεί να ασχοληθεί με την εκμάθηση, λόγω του σχετικού περιβάλλοντος, ενώ στην παιδική χαρά γνωρίζει πως να ξεφύγει από μια δυσάρεστη κατάσταση την ώρα που παιχνιδιού.

6.1.1. Οθόνη Εισαγωγής



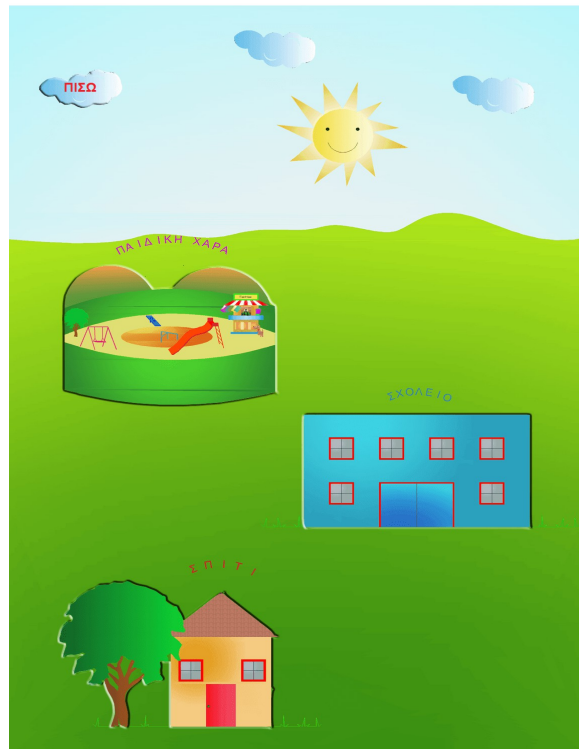
Εικόνα 13.1: Οθόνη Εισαγωγής

Κατά την έναρξη της εφαρμογής και ακριβώς πριν εμφανιστεί η κεντρική οθόνη, ο χρήστης βλέπει την οθόνη εισαγωγής. Σε αυτή την οθόνη υπάρχουν οι δύο παιδικοί χαρακτήρες, στους οποίους από πάνω έχει τοποθετηθεί ένα κουμπί που λέει “Πάμε να παίξουμε!!” και παροτρύνει το παιδί να το πατήσει ώστε να ξεκινήσει την εφαρμογή.

Επιπλέον, υπάρχει στο κάτω αριστερό μέρος ένα βελάκι σε πράσινο χρώμα που θα αναλυθεί παρακάτω και το οποίο το παιδί δεν χρειάζεται να πατήσει και γι'αυτόν το λόγο

βρίσκεται σε αυτή τη θέση.

6.1.2. Αρχική Οθόνη



Εικόνα 13.2: Αρχική Οθόνη

Η κεντρική οθόνη του παιχνιδιού είναι όπως φαίνεται στην παραπάνω εικόνα: ένας χώρος που αποτελείται από το σπίτι, το σχολείο και την παιδική χαρά. Σκοπός είναι το παιδί να διαλέξει ένα από τους τρεις χώρους ώστε να εισαχθεί σε αυτόν και να ανακαλύψει τι υπάρχει στο σπίτι ή στο σχολείο ή στην παιδική χαρά.

Ο περιβάλλον χώρος είναι με έντονα χρώματα και κατάλληλο «περίβλημα» για ένα κατά βάση παιδικό παιχνίδι. Τα έντονα χρώματα χρησιμοποιούνται κατά κύριο λόγο (και σε όλα τα σημεία της εφαρμογής), ως μέσο προσέλκυσης της παιδικής προσοχής. Όπως άλλωστε είναι ευρέως γνωστό, τα παιδάκια έχουν μια αξιοσημείωτη αδυναμία στα έντονα και ζωντανά χρώματα.

Η κεντρική οθόνη δημιουργήθηκε, έτσι ώστε να γίνεται ομαλά η μετάβαση στους επιμέρους χώρους της εφαρμογής, δίνοντας στο χρήστη λίγο χρόνο, να προσαρμοστεί πριν ξεκινήσει την προσπάθεια, να ανακαλύψει τις δοκιμασίες που έχουν δημιουργηθεί για αυτόν.

Σε αυτή λοιπόν την αρχική οθόνη, το παιδί παίρνει μία πολύ μικρή ιδέα του τι θα

επακολουθήσει. Αυτό που πρέπει να γίνει σε αυτό το σημείο είναι να πατήσει πάνω σε κάποιον από τους τρεις χώρους για να ξεκινήσει να παίζει. Κάθε χώρος αποτελείται από μία έως και περισσότερες δοκιμασίες – εκπαιδευτικά παιχνίδια. Στόχος ήταν, τα επιμέρους παιχνίδια που δημιουργηθήκαν να αντιστοιχιστούν στους χώρους που ανήκουν.

Παρακάτω, παραθέτονται οι χώροι με ανάλυση του περιεχομένου κάθε χώρου, καθώς και τα παιχνίδια που βρίσκονται μέσα σε αυτούς.

6.1.3. Παιδική Χαρά

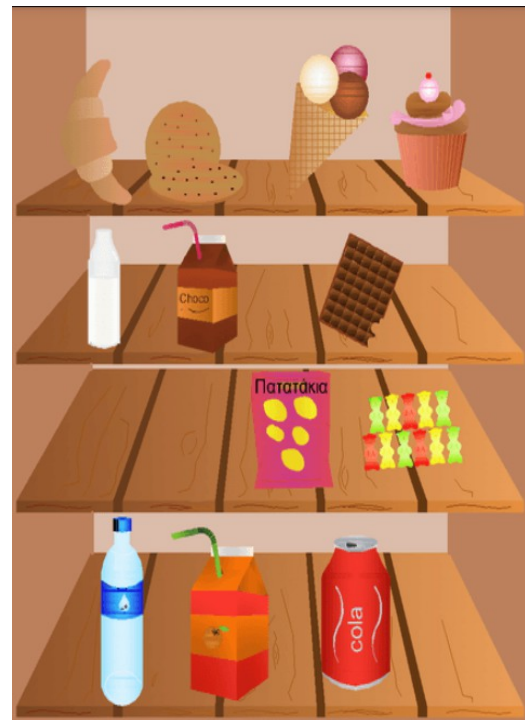


Εικόνα 13.3: Παιδική Χαρά

Μέσα στην παιδική χαρά δίνεται η επιλογή στο παιδί να διαλέξει το περίπτερο ή πατώντας το κάτω αριστερό κουμπί να μεταβεί ξανά στην αρχική οθόνη. Με την επιλογή του περιπτερου, γίνεται η μετάβαση σε ένα παιχνίδι, το οποίο ζητά από τον παίκτη να διαλέξει ένα σνακ από αυτά που είναι διαθέσιμα. Έτσι ένα παιδί με διαβήτη μπορεί να μάθει εύκολα να διαλέγει το κατάλληλο σνακ όταν βρίσκεται εκτός σπιτιού, π.χ. στην παιδική χαρά όταν μετά το παιχνίδι πεινάσει και θέλει να πάρει κάτι για να φάει ή να πιει.

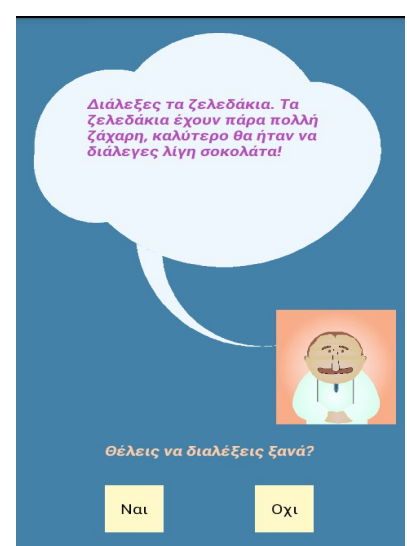


Εικόνα 13.4.1: Περίπτερο



Εικόνα 13.4.2: Ράφια περιπτέρου

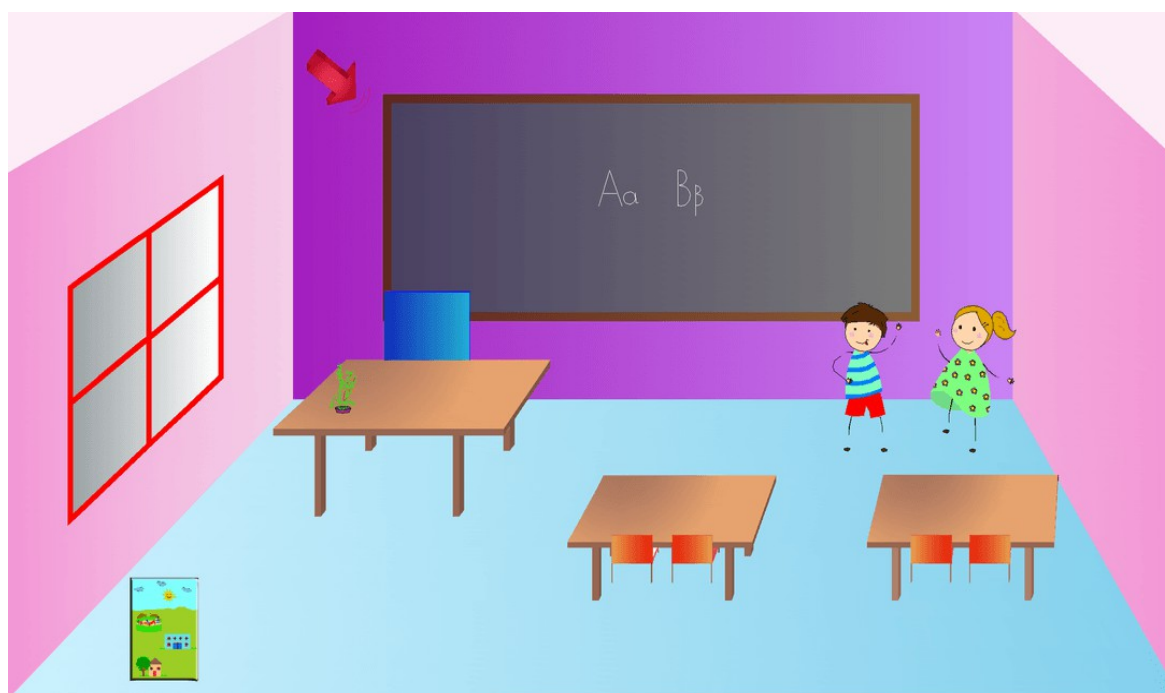
Επιλέγοντας κάποιο από τα σνακ που διαθέτει το περίπτερο, όπως φαίνεται στην Εικόνα 13.4.2, εμφανίζεται ένα μήνυμα, στο οποίο ο κύριος που δουλεύει στο περίπτερο ενημερώνει το παιδί για την επιλογή που έκανε αν δηλαδή αυτό που διάλεξε ήταν υγιεινό και κατάλληλο για τη διατροφή του ή όχι. Στην περίπτωση που το παιδί διαλέξει κάτι το οποίο δεν είναι και τόσο υγιεινό, ο περιπτεράς το παρατρύνει την επόμενη φορά που θα κλιθεί να διαλέξει, να προτιμήσει κάποιο σνακ που θα έχει λιγότερη ζάχαρη ή θα είναι πιο υγιεινό.



Εικόνες 13.4.3: Επιλογή

Αφού εμφανιστεί, λοιπόν, το αντίστοιχο μήνυμα, το παιδί μπορεί να βγεί από το παιχνίδι ή να ξαναπροσπαθήσει επιλέγοντας κάποιο άλλο σνακ, στο οποίο θα ακολουθηθεί η παραπάνω διαδικασία. Έτσι μέσα από αυτή τη δραστηριότητα μπορεί εύκολα να μάθει τι επιτρέπεται να τρώει και τι όχι, χωρίς όμως να στερείται τα γλυκά, π.χ. επιλέγοντας τη σοκολάτα εμφανίζεται το μήνυμα ότι μπορεί να φάει μερικά κομμάτια, χωρίς να νιώθει μειονεκτικά σε σχέση με τα υπόλοιπα παιδιά.

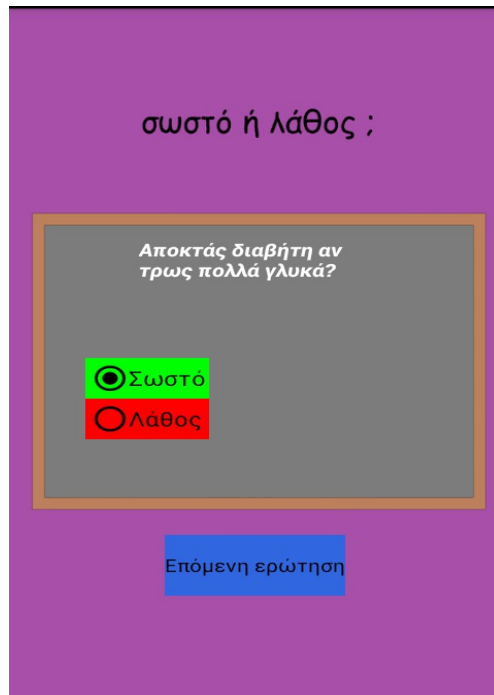
6.1.4. Σχολείο



Εικόνα 13.5.1: Σχολείο

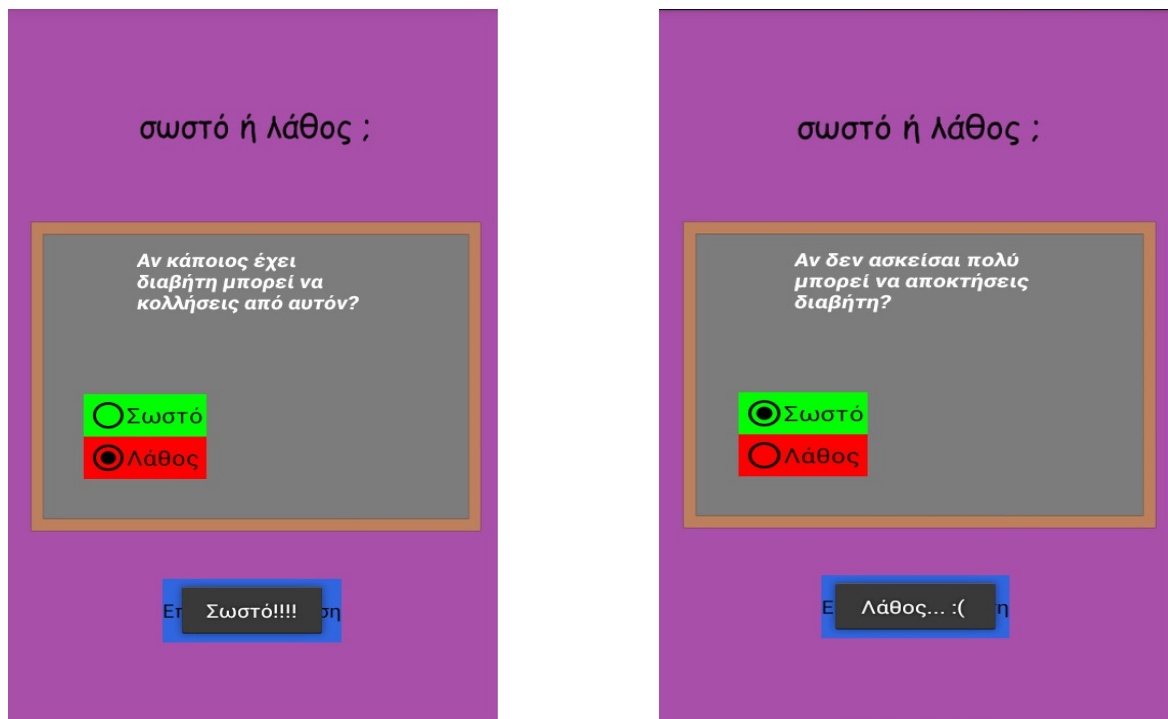
Στο σχολείο, ένα παιδί περνάει την περισσότερη ώρα της ημέρας και για αυτό τον λόγο επιλέχθηκε ο συγκεκριμένος χώρος για τη δημιουργία μιας πιο θεωρητικής δραστηριότητας, η οποία εμφανίζεται πατώντας επάνω στον πίνακα ή, όπως και στην παιδική χαρά, υπάρχει η δυνατότητα επιστροφής στην αρχική οθόνη με το κάτω αριστερά κουμπί.

Όσον αφορά τη δραστηριότητα του πίνακα, πρόκειται για ένα κουίζ όπου το παιδί θα πρέπει να απαντήσει σε 4 ερωτήσεις σωστού – λάθους ώστε να μπορέσει να μάθει κάποιες παραπάνω λεπτομέρειες σχετικά με τον παιδικό διαβήτη, χωρίς όμως να μάθει την πλήρη αλήθεια, διότι ένα παιδί με διαβήτη τύπου I βρίσκεται σε πολύ ευαίσθητη ηλικία και θα ήταν ψυχοφθόρο να γνωρίζει τόσο νωρίς όλη την αλήθεια γύρω από την ασθένεια του.



Εικόνα 13.5.2: Quiz

Κάθε φορά που ο χρήστης διαλέξει μία από τις επιλογές, σωστό ή λάθος, και πατώντας το κουμπί “Επόμενη ερώτηση”, εμφανίζεται ένα pop up μήνυμα λέγοντας αν η επιλογή ήταν σωστή ή λάθος.



Εικόνα 13.5.3: Εμφάνιση μηνύματος

Η επιλογή του σχολείου για να τοποθετηθεί το συγκεκριμένο παιχνίδι έγινε, διότι τα παιδιά συνδέουν τις περισσότερες φορές κάτι θεωρητικό με το σχολείο, οπότε η ιδέα ήταν

να μάθουν τι ισχύει σχετικά με τον διαβήτη, αλλά και να λύσουν διάφορες απορίες που τυχόν να έχουν μέσω αυτής της απλής δραστηριότητας.

6.1.5. Σπίτι

Με την εισαγωγή στο σπίτι διακρίνονται τέσσερα δωμάτια, τα οποία εμφανίζονται ανοικτά. Σκοπός είναι να μπορέσει το παιδί να είναι σε θέση να γνωρίζει τους χώρους στους οποίους μπορεί να περιηγηθεί.



Εικόνα 13.6: Πρόσοψη σπιτιού

Φυσικά και σε αυτή την περίπτωση υπάρχει η επιλογή το παιδί να επιστρέψει πίσω στην αρχική οθόνη.

Μπορεί εύκολα να διακρίνει τα τέσσερα δωμάτια και πατώντας στο αντίστοιχο δωμάτιο για να ξεκινήσει να παίζει. Το κάθε δωμάτιο περιλαμβάνει μία δραστηριότητα, εκτός του σαλονιού, όπου απαιτείται η βοήθεια του γονιού.

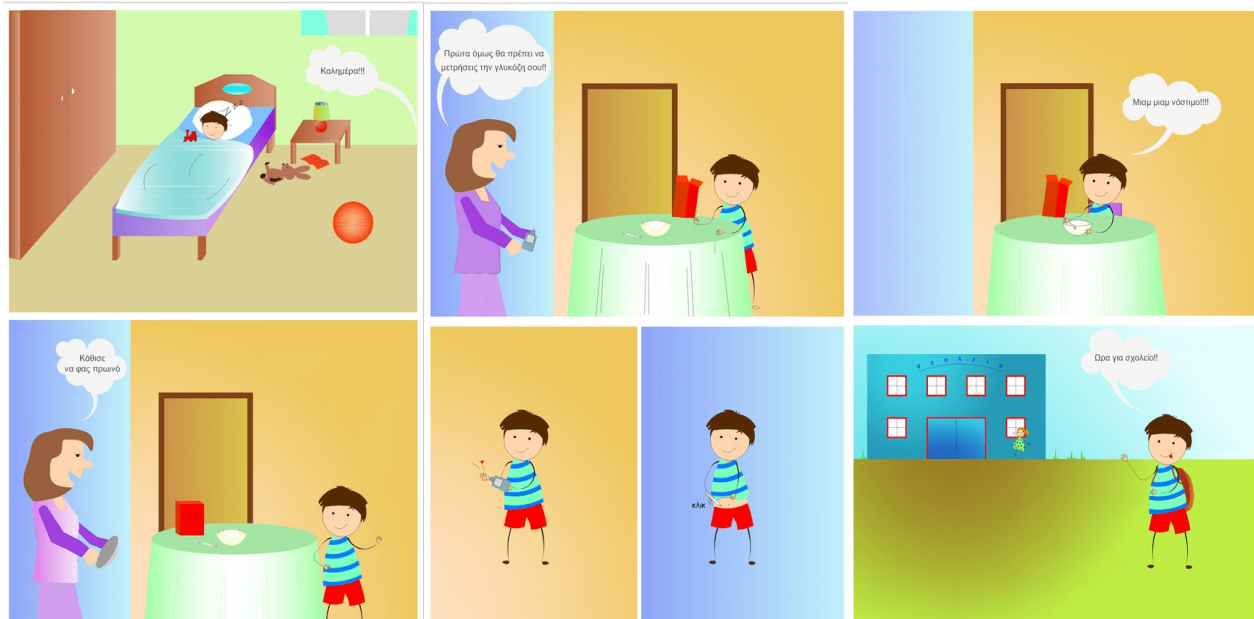
6.1.5.1. Παιδικό Δωμάτιο



Εικόνα 13.7.1: Παιδικό Δωμάτιο

Το παιδικό δωμάτιο αποτελεί ίσως τον πιο οικείο χώρο του σπιτιού για τον χρήστη που παίζει. Είναι ένα δωμάτιο γεμάτο χρώμα και παιχνίδια. Στο συγκεκριμένο δωμάτιο έχει τοποθετηθεί μία δραστηριότητα, η οποία μπορεί να ξεκινήσει πατώντας πάνω στο παιδικό κρεβάτι. Επίσης και σε αυτόν τον χώρο υπάρχει ένα κουμπί, με το οποίο το παιδί μπορεί να μεταβεί αυτή τη φορά στη σελίδα όπου φαίνεται το σπίτι με τα τέσσερα δωμάτια.

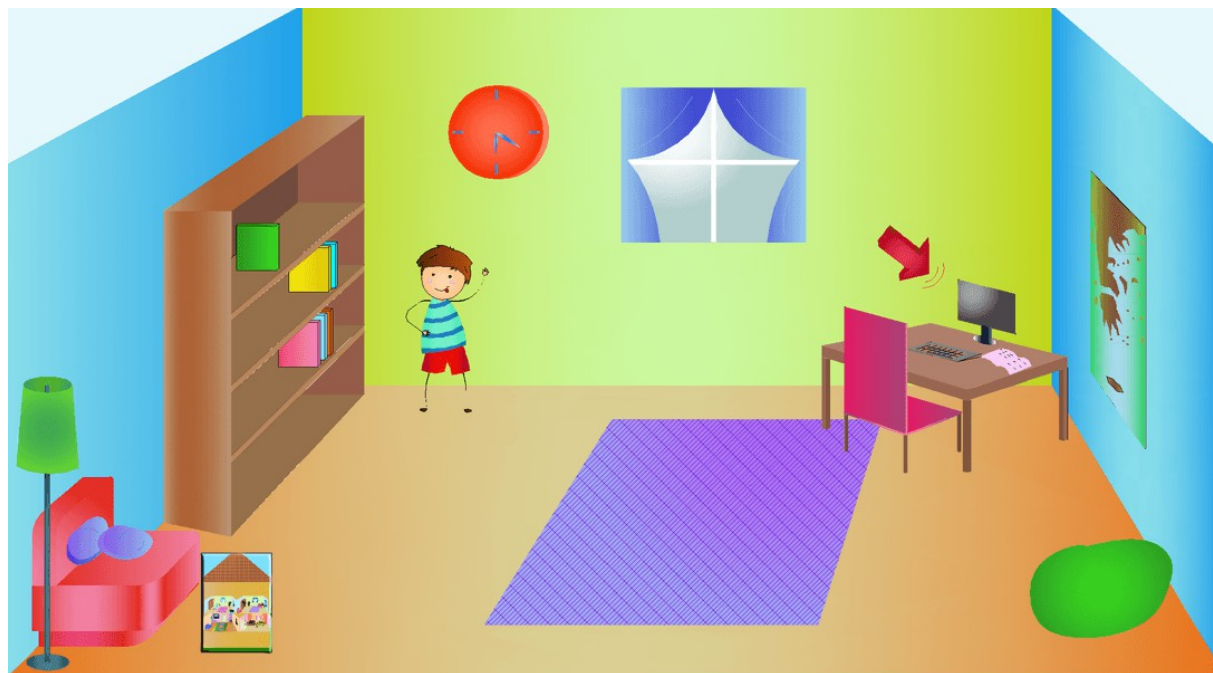
Σχετικά με τη δραστηριότητα που υπάρχει στο κρεβάτι, πρόκειται για ένα comics, το οποίο αφηγείται την πρωινή ρουτίνα ενός παιδιού με διαβήτη τύπου Ι. Συγκεκριμένα, πώς ένα διαβητικό παιδί πριν ετοιμαστεί για το σχολείο και φάει το πρωινό του, μαζί με την μητέρα του ελέγχει τη γλυκόζη του αίματος του και του χορηγείται ινσουλίνη.



Εικόνα 13.7.2: Comics

Σ'αυτό το σημείο θα πρέπει να τονιστεί ότι είναι και η μοναδική φορά που γίνεται αναφορά της χορήγησης ινσουλίνης μέσα στην εφαρμογή.

6.5.1.2. Γραφείο



Εικόνα 13.8.1: Γραφείο

Το γραφείο, ουσιαστικά, είναι ο χώρος του σπιτιού ο οποίος περιέχει τον ηλεκτρονικό υπολογιστή και μπορεί να θεωρηθεί από το παιδί ότι είναι και ο χώρος στον οποίο μπορεί να διαβάσει τα μαθήματα του ή να ασχοληθεί με τον υπολογιστή.

Το παιχνίδι αυτού του δωματίου βρίσκεται, όπως είναι λογικό, στον ηλεκτρονικό υπολογιστή και αυτό διότι τα περισσότερα παιδιά λατρεύουν να ασχολούνται ή να παίζουν σε υπολογιστές. Έτσι το παιδί μπορεί να παίξει ένα παιχνίδι και συγκεκριμένα σύγκρισης δύο τροφίμων ώστε να βρει ποιο από τα δύο είναι αυτό με τους περισσότερους υδατάνθρακες.



Εικόνα 13.8.2: Παιχνίδι σύγκρισης

Όταν ο παίκτης κάνει επιλογή του, εμφανίζεται το κατάλληλο μήνυμα. Σε περίπτωση σωστής επιλογής εμφανίζεται μήνυμα που τον ενημερώνει ότι διάλεξε το σωστό, ενώ στην λάθος επιλογή εμφανίζεται μήνυμα που τον ενημερώνει ότι η επιλογή που έκανε ήταν λάθος, αλλά και τον λόγο της λάθος επιλογής, π.χ. το hotdog έχει περισσότερους υδατάνθρακες από το αυγό, γιατί έχει το ψωμάκι.

Γενικά στόχος του παιχνιδιού, είναι η εκμάθηση του παιδιού σχετικά με τις τροφές, έτσι ώστε να είναι σε θέση να κρίνει από μόνο του, ποια τροφή να προτιμήσει.

6.5.1.3. Σαλόνι

Το σαλόνι δεν περιλαμβάνει κάποιο παιχνίδι ή δραστηριότητα. Το παιδί μαζί με τη μαμά ή τον μπαμπά μπορεί να επιλέξει το τραπεζάκι που υπάρχει στη μέση του σαλονιού και να μεταβεί σε μια φόρμα, στην οποία μπορεί να γίνει η εισαγωγή της γλυκόζης που έχει μόλις μετρηθεί από το γλυκοζόμετρο.

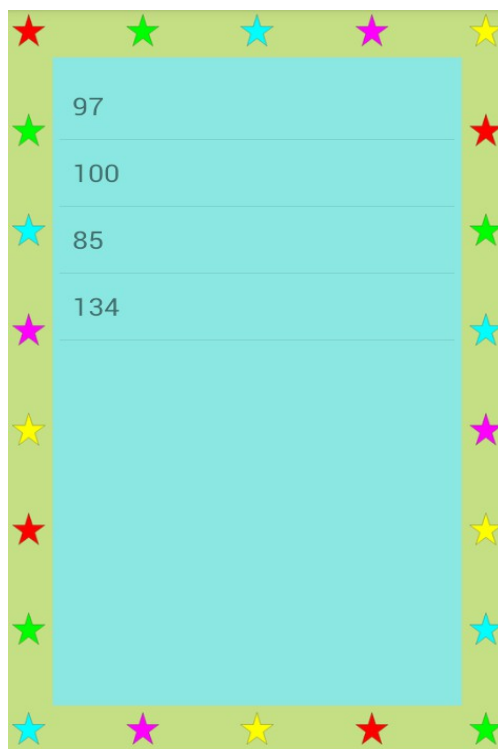


Εικόνα 13.9.1: Σαλόνι

Επιλέγοντας, λοιπόν, το γλυκοζόμετρο εμφανίζεται η φόρμα εισαγωγής στην γλυκόζη, στην οποία ο γονιός μόνος του ή και το παιδί μαζί μπορούν να αποθηκεύσουν την τιμή της. Η τιμή, αποθηκεύεται κατευθείαν σε μια βάση δεδομένων SQLite, η οποία περιέχει τρία πεδία, το id της καταγραφής, την τιμή που εισήχθη από τη φόρμα, καθώς και την ημερομηνία και ώρα της εισαγωγής. Το τελευταίο εισάγεται αυτόματα, ανάλογα με την ρυθμισμένη ώρα που έχει η κινητή συσκευή.

Εικόνα 13.9.2: Εισαγωγή γλυκόζης

Όπως αναφέρθηκε πιο πάνω, στην οθόνη εισαγωγής υπάρχει ένα πράσινο βέλος, αν επιλεγθεί τότε μπορεί ο χρήστης να μεταβεί σε μια νέα σελίδα, η οποία παρουσιάζει όλες τις τιμές της γλυκόζης που έχουν εισαχθεί. Ο λόγος που δεν βρίσκεται η συγκεκριμένη επιλογή κάπου μέσα στο παιχνίδι είναι επειδή ο γιατρός ή ο γονιός που θα θέλει να κοιτάξει το ιστορικό των τιμών δεν χρειάζεται να μπλέκει μέσα στο παιχνίδι.



Εικόνα 13.9.3: Ιστορικό γλυκόζης

6.5.1.4. Κουζίνα

Τέλος, μπαίνοντας στην κουζίνα το παιδί έχει δύο επιλογές, να πατήσει την φρουτιέρα ή το ψυγείο. Και οι δύο επιλογές συνδέονται ως προς την λογική τους, δηλαδή η εκμάθηση όλων των τροφίμων και των υδατανθράκων τους, κάτι το οποίο μπορεί να τους φανεί χρήσιμο και για το παιχνίδι στο ηλεκτρονικό υπολογιστή που υπάρχει στο γραφείο.



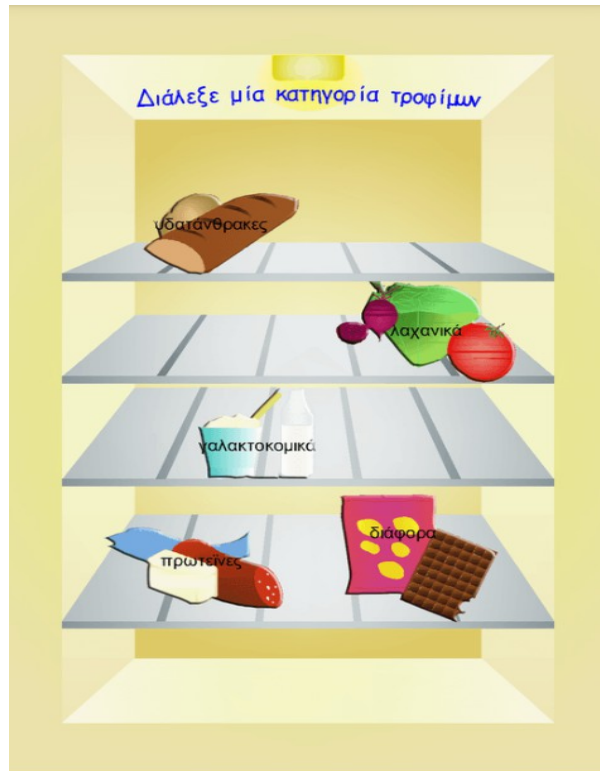
Εικόνα 13.10.1: Κουζίνα

Η φρουτιέρα περιέχει μεγάλο πλήθος φρούτων σε μια στήλη, όπου απεικονίζεται καθένα χωριστά και δίπλα ακριβώς αναγράφονται οι υδατάνθρακες που αντιστοιχούν στο φρούτο.

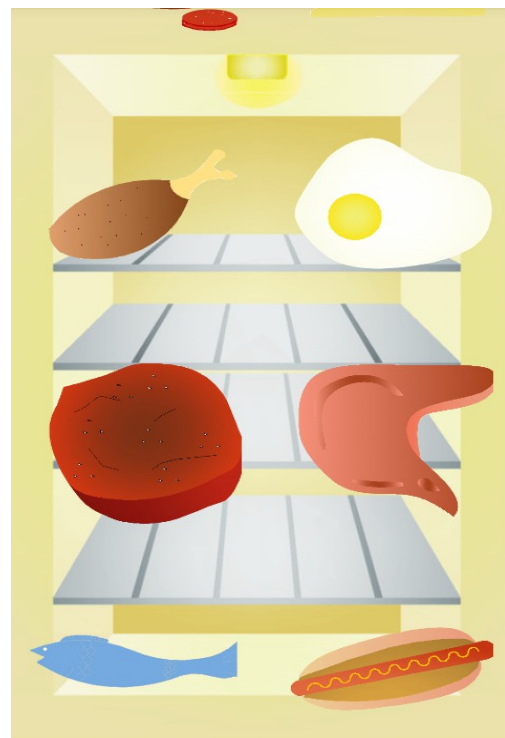
	1 Αχλάδι = 27,5γρ. υδατάνθρακες
	1 Βερύκοκο = 11γρ. υδατάνθρακες
	Μισό φλιτζάνι Φράουλες = 6,5γρ. υδατάνθρακες
	Μισό φλιτζάνι Καρπούζι = 5,5γρ. υδατάνθρακες
	Μισό φλιτζάνι Κεράσια = 11γρ.

Εικόνα 13.10.2: Περιεχόμενο φρουτιέρας

Από την άλλη, στο ψυγείο ομαδοποιεί τα τρόφιμα στις βασικές κατηγορίες, υδατάνθρακες, λαχανικά, γαλακτομικά, πρωτεΐνες και διάφορα. Στόχος αυτής της δραστηριότητας πέρα της εκμάθησης της ποσότητας των υδατανθράκων σε κάθε τρόφιμο είναι και η δυνατότητα του παιδιού να γνωρίσει τις κατηγορίες και τα τρόφιμα συγκαταλέγονται σε αυτές.



Εικόνα 13.10.3: Ψυγείο



Εικόνα 13.10.4 : Περιεχόμενο κατηγοριών

6.2. Χαρακτήρες

Το βασικό κριτήριο για την επιλογή των χαρακτήρων ήταν το παιδί να μπορεί να έχει κάποιον σαν συντροφιά μέσα στην εφαρμογή, καθώς επίσης και να μπορέσει να ταυτιστεί μαζί τους, μιας και στη δραστηριότητα του υπνοδωματίου, το αγοράκι φαίνεται ότι υποβάλλεται στη διαδικασία μέτρησης της γλυκόζης και χορήγησης ινσουλίνης. Ένας άλλος λόγος της ύπαρξης των δύο μικρών παιδιών είναι και η παρότρυνση του χρήστη να μπει στην εφαρμογή και να παίξει.



Εικόνα 14.1: Παιδικό Χαρακτήρες

Ένας ακόμα χαρακτήρας που χρησιμοποιήθηκε ήταν αυτός στο περίπτερο. Όταν ξεκίνησε η ιδέα της δημιουργίας του συγκεκριμένου παιχνιδιού, κρίθηκε αναγκαίο να υπάρχει αυτός ο κύριος ώστε η όλη διαδικασία να φαίνεται πιο αληθοφανής και ένα παιδί να μπορέσει στην πραγματικότητα αν βρεθεί σε μια τέτοια κατάσταση να είναι σε θέση να διαλέξει εξίσου κάτι από αυτά που έμαθε μέσα από το παιχνίδι αυτό.



Εικόνα 14.2: Περιπτεράς

7. Ανάλυση Κώδικα

Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, ο προγραμματισμός της εφαρμογής πραγματοποιήθηκε με το Eclipse Luna. Πριν όμως την ανάλυση της κάθε κλάσης ξεχωριστά, σημαντικό είναι να επισημανθούν κάποια από τα βασικά στοιχεία μιας εφαρμογής Android.

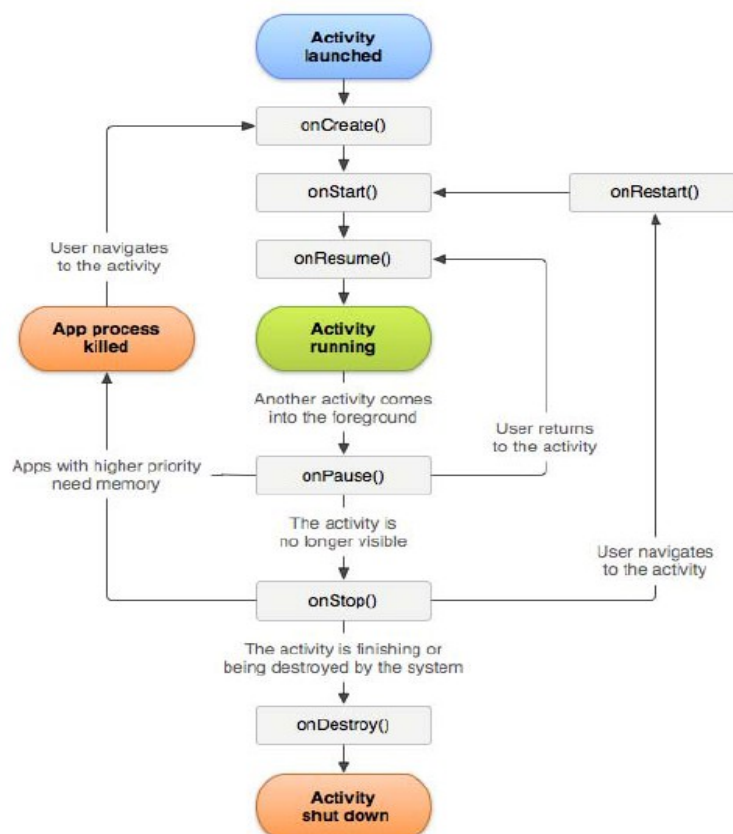
7.1. Βασικά στοιχεία μιας Android εφαρμογής

Οι δραστηριότητες (activities) αποτελούν το βασικότερο στοιχείο οποιασδήποτε εφαρμογής. Κάθε activity αποτελεί μια διαφορετική οθόνη (παράθυρο) της εφαρμογής, με την οποία ο χρήστης αλληλεπιδρά. Σε αυτήν δηλαδή φορτώνεται το γραφικό περιβάλλον το οποίο βλέπει τελικά το χρήστης. Η διεπαφή χρήστη (user interface) της εφαρμογής εμφανίζεται σε μια συσκευή μέσω μιας activity, συνήθως με μια activity που δημιουργήθηκε για κάθε μοναδική οθόνη. Η πρώτη οθόνη που βλέπει ο χρήστης ονομάζεται αρχική (main) activity. Εσωτερικά υπάρχει μια στοίβα των δραστηριοτήτων, κατά τη μετακίνηση από τη μία οθόνη στην άλλη. Συγκεκριμένα, η επόμενη activity που είναι να εμφανιστεί στέλνεται στην κορυφή της στοίβας - με άλλα λόγια, η activity στην κορυφή της στοίβας είναι αυτή που είναι ορατή στην οθόνη. Οι activities βγαίνουν από τη στοίβα πατώντας το πλήκτρο

επιστροφής, ενέργεια η οποία συνεχίζει την προηγούμενη activity. Δηλαδή υπάρχει μια μορφή στοίβας LIFO (Last In First Out). Πατώντας το πλήκτρο επιστροφής στην main activity τότε η εφαρμογή τερματίζει.

Κάθε activity πρέπει να υλοποιεί την μέθοδο `onCreate()`, η οποία καλείται την στιγμή που ξεκινά η activity. Η μέθοδος αυτή καλείται μόνο όταν τρέχει μια νέα activity είτε από ένα κουμπί είτε από κάποια επιλογή, δηλαδή δεν καλείται με την επιστροφή σε αυτή. Επίσης, μέσα σε αυτήν την μέθοδο πρέπει να δημιουργηθεί και το user interface το οποίο δηλώνεται με την εντολή `setContentView()`.

Για να ξεκινήσει μια activity μέσα από μια άλλη χρησιμοποιείται η εντολή `Intent intent = new Intent (this, ActivityB.class); startActivity(intent);`. Ενώ για τον τερματισμό μιας activity, αρκεί η εντολή `finish()`. Σε γενικές γραμμές, το Dalvik διαχειρίζεται τον κύκλο ζωής της κάθε activity οπότε δε χρειάζεται να τις τερματίσει ο προγραμματιστής. Ο κύκλος ζωής καθώς και οι αντίστοιχες μέθοδοι φαίνονται αναλυτικά στην παρακάτω εικόνα.



Εικόνα 15: Κύκλο ζωής μιας activity

Ένα ακόμα βασικό στοιχείο είναι και οι πάροχοι περιεχομένου (content providers), οι οποίοι διαχειρίζονται αποθηκευτικούς χώρους για δεδομένα, οι οποίοι είναι προσπελάσιμοι από οποιαδήποτε εφαρμογή. Αποτελούν τον μοναδικό τρόπο για αποθήκευση δεδομένων,

στα οποία θέλουμε πρόσβαση και από άλλες εφαρμογές. Κλασικό παράδειγμα είναι η εφαρμογή για την αναπαραγωγή μουσικής η οποία αποθηκεύει τα αρχεία σε συγκεκριμένο χώρο ώστε αυτά να είναι προσβάσιμα από οποιαδήποτε άλλη εφαρμογή.

Σε αυτό το πλαίσιο υπάρχει και η ενσωματωμένη βάση δεδομένων στο λειτουργικό σύστημα Android, η SQLite Database, στην οποία μπορεί να αποθηκεύσουν και να διαβάσουν δεδομένα οι content providers.

Η διεπαφή χρήστη (user interface) έχει τεράστια σημασία για κάθε εφαρμογή. Αποτελεί την τελική εικόνα που βλέπει ο χρήστης, το γραφικό περιβάλλον στο οποίο θα περιηγηθεί και ενεργοποιεί όλες τις λειτουργίες της εφαρμογής. Το user interface και η λειτουργικότητα της εφαρμογής αποτελούν αλληλένδετα στοιχεία και χωρίς το ένα δεν μπορεί να υπάρξει το άλλο. Πολλές φορές είναι πιο δύσκολο για τον προγραμματιστή να κατασκευάσει ένα εύχρηστο και όμορφο περιβάλλον εργασίας παρά η ίδια η εφαρμογή. Καθίσταται σαφές λοιπόν, ότι απαιτεί μεγάλη προσπάθεια και προσοχή η δημιουργία ενός γραφικού περιβάλλοντος που θα προσελκύει τους χρήστες και θα τους ωθεί να χρησιμοποιούν μια συγκεκριμένη εφαρμογή έναντι μιας άλλης με τις ίδιες λειτουργίες.

Τέλος, σε κάθε οθόνη της εφαρμογής πρωταρχικό στοιχείο του γραφικού περιβάλλοντος αποτελεί η διάταξη των γραφικών στοιχείων (layout). Το layout περιλαμβάνει όλα τα γραφικά στοιχεία της οθόνης τα οποία μπορεί να είναι διατεταγμένα σε επιμέρους layouts.

Υπάρχουν τέσσερα είδη διάταξης η γραμμική διάταξη (Linear Layout), η σχετική διάταξη (Relative Layout), η διάταξη πλαισίου (Frame Layout) και η διάταξη πίνακα (Table Layout). Υπάρχει όμως και ένα άλλο είδος διάταξης, το Absolute Layout, το οποίο έχει αποφασιστεί να μην χρησιμοποιείται πλέον γιατί ορίζει την διάταξη των αντικείμενων με βάση την απόλυτη θέση τους σε σχέση με την οθόνη, η οποία όμως διαφέρει ανάλογα με την συσκευή που έχει ο χρήστης. Το Linear Layout αποτελεί την διάταξη στοιχείων σε οριζόντια ή κατακόρυφη σειρά. Το Relative Layout δίνει στον προγραμματιστή μια μεγαλύτερη ευχέρεια και ελευθερία με την έννοια ότι μπορεί να εμφανίσει οποιοδήποτε αντικείμενο σε όποια θέση θέλει, δηλαδή στην αρχή, στη μέση και στο τέλος της οθόνης, ακόμα και να ορίσει την θέση σε σχέση με κάποιο άλλο αντικείμενο. Το Frame Layout αποτελεί την πιο απλή διάταξη στοιχείων, καθώς δημιουργεί ένα κενό χώρο όπου μπορεί ο προγραμματιστής να βάλει ένα αντικείμενο, για παράδειγμα μια εικόνα, και αυτό το στοιχείο θα έχει τον ρόλο της ρίζας (root) όπου όλα τα υπόλοιπα θα διατάσσονται σύμφωνα με αυτό. Τέλος, το Table Layout, είναι η διάταξη πίνακα, μπορεί δηλαδή να διατάσσει τα παιδιά του (children) σε σειρές και στήλες. Όλα τα παραπάνω μπορούν να πάρουν και διάφορες παραμέτρους όπως το μέγεθος, οριζόντια η κατακόρυφη διάταξη,

βαρύτητα (gravity) και άλλα.

7.2. Προγραμματισμός Παιχνιδιού

Κατά τη δημιουργία του συγκεκριμένου project διαπιστώθηκε ότι είναι αρκετά μεγάλο και περίπλοκο, οπότε σημαντικό ρόλο έπαιξε η σωστή οργάνωση. Μέγαλο πρόβλημα υπήρχε στην πληθώρα διαφορετικών χώρων και των επιμέρους παιχνιδιών. Οπότε, για να μην υπάρξει μεγάλο μπέρδεμα στον κώδικα με την δημιουργία μεγάλων και χαοτικών κλάσεων, έσπασαν σε μικρότερες, έτσι ώστε η διαχείριση του κώδικα αλλά και οι διάφορες τροποποιήσεις να γίνονται πιο εύκολα.

7.2.1. Προγραμματισμός Οθόνης Εισαγωγής

Η οθόνη εισαγωγής της εφαρμογής , όπως αναφέρθηκε παραπάνω, αποτελείται απο δύο κουμπιά. Αυτό της εισαγωγής μέσα στο παιχνίδι (*btnGame*) και αυτό για την προβολή του ιστορικού καταγραφής της γλυκόζης (*viewLog*). Η κλάση στην οποία γίνεται η δήλωση αυτών των κουμπιών είναι η *MainActivity.java*, η οποία φαίνεται παρακάτω.

```
package com.evi.kidsdiabetes;

import android.support.v7.app.ActionBarActivity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;

public class MainActivity extends ActionBarActivity {

    Button btnGame;
    Button viewLog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);

        btnGame = (Button) findViewById(R.id.btn_game);
```

```

btnGame.setVisibility(View.VISIBLE);
btnGame.setBackgroundColor(Color.TRANSPARENT);

btnGame.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(MainActivity.this, Game.class);
        startActivity(i);
    } });

viewLog = (Button) findViewById(R.id.btn_vl);
viewLog.setVisibility(View.VISIBLE);
viewLog.setBackgroundColor(Color.TRANSPARENT);

viewLog.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(MainActivity.this,ViewLog.class);
        startActivity(i);
    } });
}

public Bitmap getResizedBitmap(Bitmap image, int maxSize) {
    int width = image.getWidth();
    int height = image.getHeight();

    float bitmapRatio = (float)width / (float) height;
    if (bitmapRatio > 0) {
        width = maxSize;
        height = (int) (width / bitmapRatio);
    } else {
        height = maxSize;
        width = (int) (height * bitmapRatio);
    }
    return Bitmap.createScaledBitmap(image, width, height, true);
}

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

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

```

Αρχικά γίνεται η δήλωση των δύο κουμπιών, τα οποία μετατρέπονται σε αόρατα με `.setVisibility(View.VISIBLE);` και `.setBackgroundColor(Color.TRANSPARENT);`, ενώ στη

συνέχεια καλώντας την `.setOnClickListener(new View.OnClickListener())` και κατ'επέκταση την μέθοδο `onClick(View v)` γίνεται `intent` και μεταβαίνει ο χρήστης ή στην αρχική οθόνη του παιχνιδιού είτε στο ιστορικό της γλυκόζης.

Θα πρέπει να σημειωθεί ότι επιλέχθηκε η δημιουργία μια ολοκληρωμένης εικόνας `background` και έπειτα να προστεθούν από πάνω τα κουμπιά στις κατάλληλες θέσεις, αλλά αόρατα. Με αυτόν τον τρόπο αποφεύχθηκε η περίπτωση τα κουμπιά να μπερδευτούν και να μπουν σε λανθασμένες θέσεις, διότι ο καθορισμός των θέσεων των στοιχείων και ιδιαίτερα των κουμπιών είναι αρκετά δύσκολο με την χρήση του `layout`.

Η μέθοδος `public Bitmap getResizedBitmap(Bitmap image, int maxSize)` ουσιαστικά ρυθμίζει κατάλληλα τις διαστάσεις των εικονών που περνιούνται μέσω του `xml` αρχείο ώστε να ταιριάζει κατάλληλα στις ίντσες της κινητής συσκευής.

Για τη διαμόρφωση της σελίδας, αντίστοιχα, δημιουργήθηκε το `activity_main.xml`, στο οποίο δηλώθηκαν τα κουμπιά καθώς και το `background` της οθόνης εισαγωγής. Ο κώδικας του `layout` φαίνεται παρακάτω.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="@drawable/background_1"
    >

    <Button
        android:id="@+id/btn_vl"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_marginBottom="50dp"
        android:layout_marginLeft="95dp" />

    <Button
        android:id="@+id/btn_game"
        android:layout_width="350dp"
        android:layout_height="140dp"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="32dp"
        android:layout_marginTop="62dp"
        android:layout_toRightOf="@+id/btn_vl" />

</RelativeLayout>
```

Η ιδιότητα `alignParent` κάνει το στοιχείο, σ'αυτήν την περίπτωση το κουμπί (`button`), να ταιριάζει στην επιλεγμένη πλευρά, για παράδειγμα η εντολή

`android:layout_alignParentBottom="true"` κάνει το κάτω άκρο του στοιχείου να ταιριάζει στο κάτω άκρο της σελίδας. Φυσικά κάτω από αυτή την ιδιότητα δηλώνεται πάντα και το περιθώριο που υπάρχει από το στοιχείο και την επιλεγμένη πλευρά, με την εντολή `margin` .

7.2.2. Προγραμματισμός Αρχικής Οθόνης

Με τον ίδιο τρόπο έγινε και ο προγραμματισμός της αρχικής οθόνης, η οποία αποτελείται από τέσσερα κουμπιά, τρία από τα οποία είναι για την μετάβαση του αντίστοιχου χώρου και ένα κουμπί το οποίο μπορεί ο χρήστης να επιστρέψει στην οθόνη εισαγωγής. Ο κώδικας της κλάσης `Game.java` φαίνεται παρακάτω.

```
package com.evi.kidsdiabetes;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Game extends Activity{

    Button btnHouse;
    Button btnSchool;
    Button btnPlayground;
    Button btnBack;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.game);

        ///////////-----HOUSE-----//////////

        btnHouse = (Button) findViewById(R.id.button1);
        btnHouse.setVisibility(View.VISIBLE);
        btnHouse.setBackgroundColor(Color.TRANSPARENT);

        btnHouse.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent i = new Intent(Game.this, House.class);
                startActivity(i);
            } });

        ///////////-----SCHOOL-----//////////

        btnSchool = (Button) findViewById(R.id.button2);
        btnSchool.setVisibility(View.VISIBLE);
        btnSchool.setBackgroundColor(Color.TRANSPARENT);
```

```

btnSchool.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(Game.this, School.class);
        startActivity(i);
    } });

//////////-----PLAYGROUND-----//////////

btnPlayground = (Button) findViewById(R.id.button3);
btnPlayground.setVisibility(View.VISIBLE);
btnPlayground.setBackgroundColor(Color.TRANSPARENT);

btnPlayground.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(Game.this, Playground.class);
        startActivity(i);
    } });

//////////-----BACK-----//////////

btnBack = (Button) findViewById(R.id.back);
btnBack.setVisibility(View.VISIBLE);
btnBack.setBackgroundColor(Color.TRANSPARENT);

btnBack.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(Game.this, MainActivity.class);
        startActivity(i);
    } });

}

public Bitmap getResizedBitmap(Bitmap image, int maxSize) {
    int width = image.getWidth();
    int height = image.getHeight();

    float bitmapRatio = (float)width / (float) height;
    if (bitmapRatio > 0) {
        width = maxSize;
        height = (int) (width / bitmapRatio);
    } else {
        height = maxSize;
        width = (int) (height * bitmapRatio);
    }
    return Bitmap.createScaledBitmap(image, width, height, true);
}
}

```

Ενώ το layout αυτής της κλάσης είναι ακριβώς δομημένο όπως και αυτό της οθόνης εισαγωγής.


```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="@drawable/background_2"
    tools:context="com.example.kidsdiabetes.Game">

    <Button
        android:id="@+id/button2"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="300dp"
        android:layout_height="230dp"
        android:layout_above="@+id/button1"
        android:layout_alignParentRight="true"
        android:layout_marginBottom="55dp"
        android:layout_marginRight="75dp" />

    <Button
        android:id="@+id/button1"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="300dp"
        android:layout_height="230dp"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@+id/button2" />

    <Button
        android:id="@+id/button3"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="300dp"
        android:layout_height="250dp"
        android:layout_alignBottom="@+id/button2"
        android:layout_alignRight="@+id/button1"
        android:layout_marginBottom="180dp"
        android:layout_marginRight="36dp" />

    <Button
        android:id="@+id/back"
        android:layout_width="150dp"
        android:layout_height="100dp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="40dp"
        android:text="ΠΙΣΩ"
        android:textColor="#FF0000" />

</RelativeLayout>

```

Στο κάθε κουμπί έχει οριστεί το μήκος και το πλάτος, ενώ στο κουμπί με id back έχει οριστεί το κείμενο “ΠΙΣΩ” με χρώμα κειμένου το “#FF0000”, που αντιστοιχεί σε μια απόχρωση του κόκκινου.

7.2.3. Προγραμματισμός Παιδικής Χαράς και Περιπτέρου

Ο χρήστης, όπως αναφέρθηκε παραπάνω, μπορεί να επιλέξει είτε να μεταβεί στο περίπτερο είτε να επιστρέψει πίσω στην αρχική οθόνη. Η κλάση που περιγράφει αυτή τη διαδικασία είναι η Playground.java και το αντίστοιχο xml αρχείο το playground.xml.

```
package com.evi.kidsdiabetes;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Playground extends Activity{

    Button btnKiosk, btnBack;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.Playground);

        //-----KIOSK-----//

        btnKiosk = (Button) findViewById(R.id.kiosk);
        btnKiosk.setVisibility(View.VISIBLE);
        btnKiosk.setBackgroundColor(Color.TRANSPARENT);

        btnKiosk.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent i = new Intent(Playground.this, Kiosk.class);
                startActivity(i);
            }
        });

        //-----BACK-----//

        btnBack = (Button) findViewById(R.id.back);
        btnBack.setVisibility(View.VISIBLE);
        btnBack.setBackgroundColor(Color.TRANSPARENT);

        btnBack.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent i = new Intent(Playground.this, Game.class);
                startActivity(i);
            }
        });
    }

    public Bitmap getResizedBitmap(Bitmap image, int maxSize) {
        int width = image.getWidth();
        int height = image.getHeight();

        float bitmapRatio = (float)width / (float) height;
```

```

        if (bitmapRatio > 0) {
            width = maxSize;
            height = (int) (width / bitmapRatio);
        } else {
            height = maxSize;
            width = (int) (height * bitmapRatio);
        }
        return Bitmap.createScaledBitmap(image, width, height, true);
    }
}

```

Το layout της οθόνης αυτής είναι το εξής:

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/paidikh"
    tools:context="com.example.kidsdiabetes.PPlayground">

    <Button
        android:id="@+id/kiosk"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="200dp"
        android:layout_height="50dp"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true" />

    <Button
        android:id="@+id/back"
        android:layout_width="150dp"
        android:layout_height="20dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        />

</RelativeLayout>

```

Αφού το παιδί βρεθεί στο περίπτερο εμφανίζεται ένα μήνυμα πάνω από αυτό και ένα κουμπί το οποίο το παραπέμπει να μπει στο παιχνίδι. Παρακάτω διακρίνονται η κλάση Kiosk.java και το αντίστοιχο layout kiosk.xml.

```

package com.evi.kidsdiabetes;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class Kiosk extends Activity{

    Button btnSnack;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.kiosk);
    }
}

```

```

        btnSnack = (Button) findViewById(R.id.go);

        btnSnack.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent i = new Intent(Kiosk.this, Snack.class);
                startActivity(i);
            }
        });
    }
}
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/periptero"
    tools:context="com.example.kidsdiabetes.Kiosk">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="25dp"
        android:text="Πείνασες? Διάλεξε ένα σνακ!"
        android:textStyle="bold|italic"
        android:textColor="#F75D59"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <Button
        android:id="@+id/go"
        android:layout_width="150dp"
        android:layout_height="80dp"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="81dp"
        android:background="#01DFD7"
        android:text="Πάμε!! :)" />

</RelativeLayout>

```

Το κείμενο που αναγράφεται δηλώνεται από το πεδίο TextView, στο οποίο ορίζεται το κείμενο, το χρώμα και το στυλ του κειμένου αυτού καθώς και τις διαστάσεις που θα καταλαμβάνει μέσα στην οθόνη, όπου σε αυτή την περίπτωση είναι *wrap_content*, δηλαδή το μέγεθος του αντικειμένου textview θα είναι τόσο όσο χωράει στη σελίδα, λαμβάνοντας υπόψιν το περιεχόμενο του.

Στο περίπτερο, εμφανίζονται τέσσερα ράφια στα οποία πάνω βρίσκονται οι δώδεκα επιλογές. Στην κλάση Snack.java ορίζονται δώδεκα κουμπιά, ένα για το κάθε προϊόν που είναι και αόρατα, αλλά και ένα πίνακας με το όνομα message, ο οποίος περιέχει δώδεκα μηνύματα, ένα για κάθε επιλογή, τα οποία μεταφέρονται (i.putExtra("Msg", message[0])); στην κλάση SnackMessage.java, πατώντας πάνω στο αντίστοιχο σνακ.

```
package com.evi.kidsdiabetes;
```

```
import android.app.Activity;  
import android.content.Intent;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;
```

```
public class Snack extends Activity{
```

```
    Button btnCola;  
    Button btnCocoa;  
    Button btnMilk;  
    Button btnCroissant;  
    Button btnWater;  
    Button btnIceCream;  
    Button btnChips;  
    Button btnOrangejuice;  
    Button btnChocolate;  
    Button btnJellies;  
    Button btnCake;  
    Button btnCookie;
```

```
    String message[] = { "Διάλεξες το αναψυκτικό. Την επόμενη φορά είναι καλύτερα να  
        διαλέξεις κάτι με λιγότερη ζάχαρη, όπως χυμό ή νερό",  
        "Διάλεξες την πορτοκαλάδα. Είναι μια πολύ καλή επιλογή!",  
        "Διάλεξες το γάλα. Πολύ καλή επιλογή!",  
        "Διάλεξες το σοκολατούχο γάλα. Θα ήταν καλύτερο την επόμενη  
        φορά να πάρεις το απλό γάλα!",  
        "Διάλεξες το νερό. Πολύ καλή επιλογή! Πρέπει να θυμάσαι ότι  
        είναι απαραίτητο να πίνουμε πολύ νερό μέσα στην ημέρα!",  
        "Διάλεξες το κρουασάν. Είναι μια καλή επιλογή, αλλά την  
        επόμενη φορά διάλεξε κάτι με λιγότερη ζάχαρη.",  
        "Διάλεξες το παγωτό. Μια φορά στο τόσο είναι καλό να τρώμε  
        και ένα παγωτάκι!",  
        "Διάλεξες τη σοκολάτα. Λίγα κομμάτια σοκολάτας είναι ότι  
        πρέπει!",  
        "Διάλεξες το πατατάκια. Την επόμενη φορά διάλεξε κάτι πιο  
        υγιεινό!",  
        "Διάλεξες τα ζελεδάκια. Τα ζελεδάκια έχουν πάρα πολλή ζάχαρη,  
        καλύτερο θα ήταν να διάλεγες λίγη σοκολάτα!",  
        "Διάλεξες το cupcake. Ένα μικρό cupcake είναι μια καλή λύση  
        για γλυκό!",  
        "Διάλεξες τα μπισκότα. Ένα με δύο μπισκότα είναι ότι πρέπει."  
    };
```

```
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.snack);
```

```
        btnCola = (Button) findViewById(R.id.button1);  
        btnCola.setVisibility(View.VISIBLE);  
        btnCola.setBackgroundColor(Color.TRANSPARENT);
```

```
        btnCola.setOnClickListener(new View.OnClickListener() {
```

```

@Override
public void onClick(View v) {
    Intent i = new Intent(getBaseContext(), Snackbar.class);
    i.putExtra("Msg", message[0]);
    startActivity(i);
} });

////////////////////////////////////

btnOrangejuice = (Button) findViewById(R.id.button6);
btnOrangejuice.setVisibility(View.VISIBLE);
btnOrangejuice.setBackgroundColor(Color.TRANSPARENT);

btnOrangejuice.setOnClickListener(new View.OnClickListener() {

@Override
public void onClick(View v) {
    Intent i = new Intent(getBaseContext(), Snackbar.class);
    i.putExtra("Msg", message[1]);
    startActivity(i);
} });

////////////////////////////////////

btnMilk = (Button) findViewById(R.id.button2);
btnMilk.setVisibility(View.VISIBLE);
btnMilk.setBackgroundColor(Color.TRANSPARENT);

btnMilk.setOnClickListener(new View.OnClickListener() {

@Override
public void onClick(View v) {
    Intent i = new Intent(getBaseContext(), Snackbar.class);
    i.putExtra("Msg", message[2]);
    startActivity(i);
} });

////////////////////////////////////

btnCocoa = (Button) findViewById(R.id.button7);
btnCocoa.setVisibility(View.VISIBLE);
btnCocoa.setBackgroundColor(Color.TRANSPARENT);

btnCocoa.setOnClickListener(new View.OnClickListener() {

@Override
public void onClick(View v) {
    Intent i = new Intent(getBaseContext(), Snackbar.class);
    i.putExtra("Msg", message[3]);
    startActivity(i);
} });

////////////////////////////////////

btnWater = (Button) findViewById(R.id.button3);
btnWater.setVisibility(View.VISIBLE);
btnWater.setBackgroundColor(Color.TRANSPARENT);

btnWater.setOnClickListener(new View.OnClickListener() {

@Override

```

```

        public void onClick(View v) {
            Intent i = new Intent(getApplicationContext(), Snackbar.class);
            i.putExtra("Msg", message[4]);
            startActivity(i);
        } });

////////////////////////////////////

btnCroissant = (Button) findViewById(R.id.button8);
btnCroissant.setVisibility(View.VISIBLE);
btnCroissant.setBackgroundColor(Color.TRANSPARENT);

btnCroissant.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(getApplicationContext(), Snackbar.class);
        i.putExtra("Msg", message[5]);
        startActivity(i);
    } });

////////////////////////////////////

btnIceCream = (Button) findViewById(R.id.button4);
btnIceCream.setVisibility(View.VISIBLE);
btnIceCream.setBackgroundColor(Color.TRANSPARENT);

btnIceCream.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(getApplicationContext(), Snackbar.class);
        i.putExtra("Msg", message[6]);
        startActivity(i);
    } });

////////////////////////////////////

btnChocolate = (Button) findViewById(R.id.button9);
btnChocolate.setVisibility(View.VISIBLE);
btnChocolate.setBackgroundColor(Color.TRANSPARENT);

btnChocolate.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(getApplicationContext(), Snackbar.class);
        i.putExtra("Msg", message[7]);
        startActivity(i);
    } });

////////////////////////////////////

btnChips = (Button) findViewById(R.id.button5);
btnChips.setVisibility(View.VISIBLE);
btnChips.setBackgroundColor(Color.TRANSPARENT);

btnChips.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(getApplicationContext(), Snackbar.class);

```

```

        i.putExtra("Msg", message[8]);
        startActivity(i);
    } });

////////////////////////////////////

btnJellies = (Button) findViewById(R.id.button10);
btnJellies.setVisibility(View.VISIBLE);
btnJellies.setBackgroundColor(Color.TRANSPARENT);

btnJellies.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(getBaseContext(), SnackbarMessage.class);
        i.putExtra("Msg", message[9]);
        startActivity(i);
    } });

////////////////////////////////////

btnCake= (Button) findViewById(R.id.button11);
btnCake.setVisibility(View.VISIBLE);
btnCake.setBackgroundColor(Color.TRANSPARENT);

btnCake.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(getBaseContext(), SnackbarMessage.class);
        i.putExtra("Msg", message[10]);
        startActivity(i);
    } });

////////////////////////////////////

btnCookie = (Button) findViewById(R.id.button12);
btnCookie.setVisibility(View.VISIBLE);
btnCookie.setBackgroundColor(Color.TRANSPARENT);

btnCookie.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(getBaseContext(), SnackbarMessage.class);
        i.putExtra("Msg", message[11]);
        startActivity(i);
    } });
}
}

```

Το layout snack.xml είναι ουσιαστικά πάλι ένα xml αρχείο που περιέχει δώδεκα κουμπιά με τις αντίστοιχες ιδιότητες τους.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

```



```
android:background="@drawable/rafia" >
```

```
<Button
    android:id="@+id/button1"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="150dp"
    android:layout_height="250dp"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_marginBottom="20dp"
    android:layout_marginRight="200dp"
    />

<Button
    android:id="@+id/button6"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="150dp"
    android:layout_height="250dp"
    android:layout_alignBaseline="@+id/button1"
    android:layout_alignBottom="@+id/button1"
    android:layout_marginRight="50dp"
    android:layout_toLeftOf="@+id/button1" />

<Button
    android:id="@+id/button3"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="150dp"
    android:layout_height="250dp"
    android:layout_alignBaseline="@+id/button6"
    android:layout_alignBottom="@+id/button6"
    android:layout_marginRight="14dp"
    android:layout_toLeftOf="@+id/button6" />

<Button
    android:id="@+id/button10"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="150dp"
    android:layout_height="130dp"
    android:layout_above="@+id/button1"
    android:layout_alignLeft="@+id/button1"
    android:layout_marginBottom="70dp"
    android:layout_marginLeft="85dp" />

<Button
    android:id="@+id/button5"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="150dp"
    android:layout_height="200dp"
    android:layout_alignBaseline="@+id/button10"
    android:layout_alignBottom="@+id/button10"
    android:layout_alignRight="@+id/button1"
    android:layout_marginRight="90dp" />

<Button
    android:id="@+id/button2"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="130dp"
    android:layout_height="200dp"
    android:layout_alignBaseline="@+id/button7"
    android:layout_alignBottom="@+id/button7"
    android:layout_alignLeft="@+id/button3" />
```

```

<Button
    android:id="@+id/button9"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="150dp"
    android:layout_height="180dp"
    android:layout_alignLeft="@+id/button1"
    android:layout_centerVertical="true" />

<Button
    android:id="@+id/button7"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="130dp"
    android:layout_height="200dp"
    android:layout_alignTop="@+id/button9"
    android:layout_toLeftOf="@+id/button5" />

<Button
    android:id="@+id/button8"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="150dp"
    android:layout_height="200dp"
    android:layout_alignParentTop="true"
    android:layout_alignRight="@+id/button2"
    android:layout_marginTop="100dp" />

<Button
    android:id="@+id/button12"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="150dp"
    android:layout_height="200dp"
    android:layout_alignLeft="@+id/button7"
    android:layout_alignTop="@+id/button8"
    android:layout_marginTop="42dp" />

<Button
    android:id="@+id/button11"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="130dp"
    android:layout_height="200dp"
    android:layout_alignBaseline="@+id/button4"
    android:layout_alignBottom="@+id/button4"
    android:layout_toRightOf="@+id/button9" />

<Button
    android:id="@+id/button4"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="180dp"
    android:layout_height="250dp"
    android:layout_alignRight="@+id/button9"
    android:layout_alignTop="@+id/button8" />

</RelativeLayout>

```

Αφού λοιπόν πατηθεί ένα από τα σνακ και ο χρήστης μεταβεί στην `SnackbarMessage.java`, βλέπει τον κύριο που εργάζεται στο περίπτερο να ενημερώνει τον χρήστη με την επιλογή του καθώς και αν αυτή η επιλογή είναι υγιεινή ή όχι.

```

package com.evi.kidsdiabetes;

import android.app.Activity;

```

```

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class SnackMessage extends Activity{

    Button btnYes;
    Button btnNo;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.snackmessage);

        Bundle extra = getIntent().getExtras();
        String Message = extra.getString("Msg");

        TextView tv = (TextView)findViewById(R.id.textView1);
        tv.setText(Message);

        btnYes = (Button) findViewById(R.id.yes);

        btnYes.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent i = new Intent(SnackMessage.this, Snack.class);
                startActivity(i);
            }
        });

        btnNo = (Button) findViewById(R.id.no);

        btnNo.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent i = new Intent(SnackMessage.this, Playground.class);
                startActivity(i);
            }
        });
    }
}

```

Στο xml αρχείο snackmessage.xml, τώρα, ορίζεται ένα πεδίο στο οποίο εμφανίζεται το μήνυμα και ένα ακόμα όπου εμφανίζεται η ερώτηση αν θέλει να ξαναδιαλέξει κάποιο νέο σνακ ή όχι. Οπώς είναι επόμενο, κάτω από την ερώτηση, υπάρχουν αντίστοιχα δύο κουμπιά ΝΑΙ – ΟΧΙ και αναλόγως την επιλογή είτε ξεκινάει πάλι η κλάση SnackMessage ή την Playground.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/message">

    <TextView
        android:id="@+id/textView1"

```

```
android:layout_width="400dp"
android:layout_height="245dp"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:layout_marginTop="170dp"
android:textColor="#B048B5"
android:textStyle="bold|italic" />
```

<Button

```
android:id="@+id/yes"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/no"
android:layout_alignBottom="@+id/no"
android:layout_alignLeft="@+id/textView2"
android:background="#FFF8C6"
android:text="Ναι" />
```

<Button

```
android:id="@+id/no"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_alignRight="@+id/textView2"
android:layout_marginBottom="75dp"
android:background="#FFF8C6"
android:text="Όχι" />
```

<TextView

```
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_above="@+id/yes"
android:layout_centerHorizontal="true"
android:layout_marginBottom="82dp"
android:text="Θέλεις να διαλέξεις ξανά?"
android:textAppearance="?android:attr/textAppearanceLarge"
android:textColor="#FFCBA4"
android:textStyle="bold|italic" />
```

</RelativeLayout>

7.2.4. Προγραμματισμός Σχολείου και Πίνακα

Το σχολείο έχει ακριβώς την ίδια δομή με την παιδική χαρά (θεωρείται πλεονασμός να τοποθετηθεί ο κώδικα από τον συγκεκριμένο χώρο). Το παιδί πατώντας πάνω στον πίνακα μπορεί να μεταβεί στο παιχνίδι/κουίζ.

Το κουίζ αποτελείται από ένα πεδίο που εμφανίζεται η ερώτηση καθώς και ένα RadioGroup με τις δύο επιλογές σωστού ή λάθους.

```
package com.evi.kidsdiabetes;
```

```
import android.app.Activity;
```

```
import android.content.Intent;
```

```

import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;

public class Board extends Activity{

    TextView tv;
    Button btnNext;
    RadioGroup rg;
    RadioButton buttonCorrect, buttonWrong;

    String questions[]={ "Αποκτάς διαβήτη αν τρως πολλά γλυκά?",
        "Αν κάποιος έχει διαβήτη μπορεί να κολλήσεις από αυτόν?",
        "Αν δεν ασκείσαι πολύ μπορεί να αποκτήσεις διαβήτη?",
        "Έχεις διαβήτη, επειδή η γιαγιά σου ή ο παππούς σου έχει
        διαβήτη τύπου 2?"};

    String ans[]={ " Λάθος ", " Λάθος ", " Λάθος ", " Λάθος "};
    String opt[]={ " Σωστό ", " Λάθος ", " Σωστό ", " Λάθος ",
        " Σωστό ", " Λάθος ", " Σωστό ", " Λάθος "};

    int flag=0;
    public static int correct,wrong;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.board);

        tv=(TextView)findViewById(R.id.question);
        btnNext=(Button)findViewById(R.id.next);
        rg = (RadioGroup)findViewById(R.id.radioGroup1);
        buttonCorrect = (RadioButton)findViewById(R.id.sosto);
        buttonWrong = (RadioButton)findViewById(R.id.lathos);

        tv.setText(questions[flag]);
        buttonCorrect.setText(opt[0]);
        buttonWrong.setText(opt[1]);

        btnNext.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                RadioButton uans =
(RadioButton)findViewById(rg.getCheckedRadioButtonId());
                String ansText=uans.getText().toString();

                if(ansText.equalsIgnoreCase(ans[flag])) {
                    Toast.makeText(getApplicationContext(),
"Σωστό!!!!",Toast.LENGTH_SHORT).show();
                }
                else{
                    Toast.makeText(getApplicationContext(), "Λάθος... :(
",Toast.LENGTH_SHORT).show();
                }
                flag++;
                if(flag<questions.length){

```

```

        tv.setText(questions[flag]);
        buttonCorrect.setText(opt[flag*2]);
        buttonWrong.setText(opt[(flag*2)+1]);
    }
    else{
        Intent i = new Intent(getApplicationContext(), School.class);
        startActivity(i); }
    } });
} }

```

Ουσιαστικά ελέγχεται αν η επιλογή από το RadioGroup είναι ίση με την αντίστοιχη σωστή απάντηση της ερώτησης από τον πίνακα ans[] πατώντας το κουμπί btnNext. Στην περίπτωση που η παραπάνω συνθήκη αληθεύει , εμφανίζεται το μήνυμα “Σωστό!!!” με Toast, διαφορετικά “Λάθος... :(”. Στην συνέχεια αυξάνεται ένας μετρητής (flag++) και γίνεται ο έλεγχος αν η flag είναι μικρότερη από το μήκος του πίνακα questions[], αν είναι αληθής η συνθήκη τότε στη μεταβλητή tv εισάγεται η νέα ερώτηση από τον πίνακα questions και στα radioButton, buttonCorrect και buttonWrong, οι δύο επιλογές από τον opt[].

Στο layout board.xml δηλώνονται τα απαραίτητα πεδία, δηλαδή αυτό στο οποίο εμφανίζεται η κάθε ερώτηση και το radioGroup με τις δύο επιλογές καθώς και το btnNext για την φόρτωση της επόμενης ερώτησης.

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/pinakas" >

    <TextView
        android:id="@+id/question"
        android:layout_width="350dp"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="350dp"
        android:text="TextView"
        android:textStyle="bold/italic"
        android:textColor="#FFFFFF"
        android:textAppearance="?android:attr/textAppearanceLarge"/>

    <RadioGroup
        android:id="@+id/radioGroup1"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/question"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="60dp" >

        <RadioButton
            android:id="@+id/sosto"
            android:layout_width="150dp"

```

```

        android:layout_height="50dp"
        android:checked="true"
        android:text="Σωστό "
        android:background="#00FF00" />

<RadioButton
    android:id="@+id/Lathos"
    android:layout_width="150dp"
    android:layout_height="50dp"
    android:text="Λάθος "
    android:background="#FF0000" />

</RadioGroup>

<Button
    android:id="@+id/next"
    android:layout_width="200dp"
    android:layout_height="80dp"
    android:layout_below="@+id/radioGroup1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="150dp"
    android:text="Επόμενη ερώτηση"
    android:background="#2B60DE" />

</RelativeLayout>

```

7.2.5. Προγραμματισμός σπιτιού

Όσον αφορά το σπίτι, αποτελείται από τους τέσσερις χώρους – δωμάτια και το κουμπί επιστροφής στην αρχική οθόνη. Ο προγραμματισμός της συγκεκριμένης διάταξης είναι απλή. Περιέχει τις δηλώσεις των κουμπιών αλλά και τα αντίστοιχα intent με τα οποία ο χρήστης μπορεί να μεταβεί στο δωμάτιο που έχει επιλέξει. Η κλάση House.java και το xml αρχείο αυτής της κλάσης house.xml φαίνονται παρακάτω.

```

package com.evi.kidsdiabetes;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class House extends Activity{

    Button btnBedroom;
    Button btnStudyroom;
    Button btnLivingroom;
    Button btnKitchen;
    Button btnBack;

    public void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.house);

//////////-----BEDROOM-----//////////

btnBedroom = (Button) findViewById(R.id.bedroom);
btnBedroom.setVisibility(View.VISIBLE);
btnBedroom.setBackgroundColor(Color.TRANSPARENT);

btnBedroom.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(House.this, Bedroom.class);
        startActivity(i);
    } });

//////////-----STUDYROOM-----//////////

btnStudyroom = (Button) findViewById(R.id.studyroom);
btnStudyroom.setVisibility(View.VISIBLE);
btnStudyroom.setBackgroundColor(Color.TRANSPARENT);

btnStudyroom.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(House.this, Studyroom.class);
        startActivity(i);
    } });

//////////-----LIVINGROOM-----//////////

btnLivingroom = (Button) findViewById(R.id.livingroom);
btnLivingroom.setVisibility(View.VISIBLE);
btnLivingroom.setBackgroundColor(Color.TRANSPARENT);

btnLivingroom.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(House.this, Livingroom.class);
        startActivity(i);
    } });

//////////-----KITCHEN-----//////////

btnKitchen = (Button) findViewById(R.id.kitchen);
btnKitchen.setVisibility(View.VISIBLE);
btnKitchen.setBackgroundColor(Color.TRANSPARENT);

btnKitchen.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent i = new Intent(House.this, Kitchen.class);
        startActivity(i);
    } });

```



```

        } });

        btnBack = (Button) findViewById(R.id.back);
        btnBack.setVisibility(View.VISIBLE);
        btnBack.setBackgroundColor(Color.TRANSPARENT);

        btnBack.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent i = new Intent(House.this, Game.class);
                startActivity(i);
            } });
    }

    public Bitmap getResizedBitmap(Bitmap image, int maxSize) {
        int width = image.getWidth();
        int height = image.getHeight();

        float bitmapRatio = (float)width / (float) height;
        if (bitmapRatio > 0) {
            width = maxSize;
            height = (int) (width / bitmapRatio);
        } else {
            height = maxSize;
            width = (int) (height * bitmapRatio);
        }
        return Bitmap.createScaledBitmap(image, width, height, true);
    }
}

```

Kai to layout.

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/spiti_anoikto"
    >

```

```

<Button
    android:id="@+id/kitchen"
    android:layout_width="380dp"
    android:layout_height="200dp"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_marginBottom="200dp"
    android:layout_marginRight="16dp" />

<Button
    android:id="@+id/Livingroom"
    android:layout_width="380dp"
    android:layout_height="200dp"
    android:layout_alignBaseline="@+id/kitchen"
    android:layout_alignBottom="@+id/kitchen"
    android:layout_toLeftOf="@+id/kitchen" />

```

```

<Button
    android:id="@+id/studyroom"
    android:layout_width="380dp"
    android:layout_height="240dp"
    android:layout_above="@+id/kitchen"
    android:layout_toRightOf="@+id/livingroom" />

```

```

<Button
    android:id="@+id/bedroom"
    android:layout_width="380dp"
    android:layout_height="240dp"
    android:layout_alignTop="@+id/studyroom"
    android:layout_toLeftOf="@+id/studyroom" />

```

```

<Button
    android:id="@+id/back"
    style="?android:attr/buttonStyleSmall"
    android:layout_width="120dp"
    android:layout_height="170dp"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
/>

```

```

</RelativeLayout>

```

7.2.5.1. Προγραμματισμός Υπνοδωματίου

Μέσα στο δωμάτιο και με την επιλογή του κρεβατιού, ο παίκτης μπορεί να εισαχθεί σε μια μικρή ιστοριούλα, της οποίας ο κώδικας αναλύεται παρακάτω.

```

package com.evi.kidsdiabetes;

import android.os.Bundle;
import android.app.Activity;
import android.graphics.Bitmap;
import android.view.MotionEvent;
import android.widget.ViewFlipper;

public class Breakfast extends Activity {

    private ViewFlipper viewFlipper;
    private float lastX;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.breakfast);
        viewFlipper = (ViewFlipper) findViewById(R.id.ViewFlipper1);
    }

    // Method to handle touch event like left to right swap and right to left swap
    public boolean onTouchEvent(MotionEvent touchevent)
    {
        switch (touchevent.getAction())
        {
            // when user first touches the screen to swap
            case MotionEvent.ACTION_DOWN:

```

```

    {
        lastX = touchevent.getX();
        break;
    }
    case MotionEvent.ACTION_UP:
    {
        float currentX = touchevent.getX();

        // if left to right swipe on screen
        if (lastX < currentX) {
            // If no more View/Child to flip
            if (viewFlipper.getDisplayedChild() == 0)
                break;

            // set the required Animation type to ViewFlipper
            // The Next screen will come in form Left and
            // current Screen will go OUT from Right
            viewFlipper.setInAnimation(this,R.anim.in_from_left);
            viewFlipper.setOutAnimation(this,R.anim.out_to_right);
            // Show the next Screen
            viewFlipper.showNext();
        }

        // if right to left swipe on screen
        if (lastX > currentX)
        {
            if (viewFlipper.getDisplayedChild() == 1)
                break;
            // set the required Animation type to ViewFlipper
            // The Next screen will come in form Right and
            // current Screen will go OUT from Left
            viewFlipper.setInAnimation(this,R.anim.in_from_right);
            viewFlipper.setOutAnimation(this,R.anim.out_to_left);
            // Show The Previous Screen
            viewFlipper.showPrevious();
        }
        break;
    }
}
return false;
}
}

```

```

public Bitmap getResizedBitmap(Bitmap image, int maxSize) {
    int width = image.getWidth();
    int height = image.getHeight();

    float bitmapRatio = (float)width / (float) height;
    if (bitmapRatio > 0) {
        width = maxSize;
        height = (int) (width / bitmapRatio);
    } else {
        height = maxSize;
        width = (int) (height * bitmapRatio);
    }
    return Bitmap.createScaledBitmap(image, width, height, true);
}
}

```

Σ'αυτή την κλάση υπάρχει η μέθοδος onTouchEvent, η οποία αναγνωρίζει πότε ο χρήστης αγγίζει την οθόνη, έτσι ώστε να γίνει η εναλλαγή της εικόνας με την βοήθεια του

viewFlipper το οποίο ορίζεται μέσω του layout. Αν τώρα η κίνηση γίνει προς τα αριστερά και υπάρχει εικόνα τότε η εικόνα αυτή αλλάζει, ενώ αν είναι προς τα δεξιά τότε αντίστοιχα μπορεί να γίνει η μετάβαση στην νέα εικόνα. Για την εναλλαγή των εικόνων δημιουργήθηκε μέσα στον κατάλογο /res ο φάκελος anim, ο οποίος περιέχει τα animations για την αλλαγή των εικόνων. Τα βασικά είναι τα in_from_right(ή left).xml και out_from_left(ή right).xml που παρουσιάζονται παρακάτω.

in_from_right.xml & out_from_left.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <translate
        android:fromXDelta="100%" android:toXDelta="0%"
        android:fromYDelta="0%" android:toYDelta="0%"
        android:duration="1400" />
</set>

<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <translate android:fromXDelta="0%" android:toXDelta="-100%"
        android:fromYDelta="0%" android:toYDelta="0%"
        android:duration="1400"/>
</set>
```

in_from_left.xml & out_from_right.xml

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <translate android:fromXDelta="-100%" android:toXDelta="0%"
        android:fromYDelta="0%" android:toYDelta="0%"
        android:duration="1400" />
</set>

<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">
    <translate android:fromXDelta="0%" android:toXDelta="100%"
        android:fromYDelta="0%" android:toYDelta="0%"
        android:duration="1400"/>
</set>
```

Τέλος, το layout του comics που δημιουργήθηκε για την εξιστόρηση της πρωϊνής ρουτίνας ενός διαβητικού παιδιού είναι το breakfast.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <ViewFlipper
        android:id="@+id/ViewFlipper1"
```

```

        android:layout_width="fill_parent"
        android:layout_height="fill_parent" >

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:orientation="vertical" >

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/comic1" >
    </ImageView>
</LinearLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:orientation="vertical" >

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/comic3">
    </ImageView>

</LinearLayout>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:orientation="vertical" >

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/comic2">
    </ImageView>

</LinearLayout>

</ViewFlipper>

</LinearLayout>

```

7.2.5.2. Προγραμματισμός Γραφείου

Το παιχνίδι που βρίσκεται μέσα στο γραφείο είναι αυτό στον υπολογιστή, όπου το παιδί καλείται να διαλέξει ανάμεσα σε δύο τρόφιμα αυτό με τους λιγότερους υδατάνθρακες.

Για να δηλωθούν οι εικόνες των τροφίμων στην κλάση Pcgame.java δημιουργήθηκε ο πίνακας `images[]` και αντίστοιχα τα μηνύματα για την επιλογή του τροφίμου είναι αποθηκευμένα στον πίνακα `message[]`. Τέλος, δηλώνονται δύο κουμπιά, στα οποία εισάγονται οι δύο εικόνες από τον `images[]` και αφού κληθεί η μέθοδος `onClick()` μέσω του

setOnClickListener() εμφανίζεται μήνυμα με το Toast. Αφού πατηθεί ένα από τα δύο κουμπιά και βγει το κατάλληλο μήνυμα, φορτώνεται το νέο ζεύγος εικόνων αν αληθεύει η συνθήκη `if(flag<(message.length/2))`.

```
package com.evi.kidsdiabetes;

import android.app.Activity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class PCgame extends Activity implements OnClickListener{

    final int[] images = { R.drawable.mpanana, R.drawable.gala,
        R.drawable.abga, R.drawable.hotdog,
        R.drawable.portokalada, R.drawable.kalampoki,
        R.drawable.burger, R.drawable.karoto,
        R.drawable.salami, R.drawable.patata};

    final String[] message = {"ΣΩΣΤΑ!!!!", "Λάθος. Η μπανάνα έχει περισσότερους
        υδατανθρακες από το γάλα!",
        "Λάθος. Το HotDog έχει περισσότερους
        υδατάνθρακες, εξαιτίας του ψωμιού!", "ΣΩΣΤΑ!!!!",
        "ΣΩΣΤΑ!!!!", "Λάθος. Ο χυμός έχει περισσότερους
        υδατανθρακες!",
        "ΣΩΣΤΑ!!!!", "Λάθος. Το burger έχει περισσότερους
        υδατανθρακες από το καρότο!",
        "Λάθος. Η πατάτα έχει περισσότερους
        υδατάνθρακες!", "ΣΩΣΤΑ!!!!"};

    int flag=0;

    Button food1,food2;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.comparison);

        food1=(Button)findViewById(R.id.button1);
        food2=(Button)findViewById(R.id.button2);

        food1.setBackgroundResource(images[0]);
        food2.setBackgroundResource(images[1]);

        food1.setOnClickListener(this); //call onClick()
        food2.setOnClickListener(this);
    }

    public void onClick(View v) {
```

```

        if (v == food1) {
            Toast.makeText(getApplicationContext(), message[flag*2],
Toast.LENGTH_SHORT).show();
        }
        if(v == food2){
            Toast.makeText(getApplicationContext(), message[(flag*2)+1],
Toast.LENGTH_SHORT).show();
        }
        flag++;
        if(flag<(message.length/2)){
            food1.setBackgroundResource(images[flag*2]);
            food2.setBackgroundResource(images[(flag*2)+1]);
            food1.setOnClickListener(this); //call onClick()
            food2.setOnClickListener(this);

        }else{
            Intent i = new Intent(getApplicationContext(), Studyroom.class);
            startActivity(i);}
    }
}

```

Το layout δεν παρουσιάζει κάποια δυσκολία, αφού γίνεται πάλι η δήλωση ενός TextView και των δύο Buttons στα οποία φορτώνονται οι εικόνες.

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/hy" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="600dp"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="183dp"
        android:text="Ποιο από τα δύο έχει τους περισσότερους υδατάνθρακες?"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="#FFFFFF" />

    <Button
        android:id="@+id/button1"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_alignLeft="@+id/textView1"
        android:layout_below="@+id/textView1"
        android:layout_marginLeft="26dp"
        android:layout_marginTop="201dp"
        android:onClick="myClick" />

    <Button
        android:id="@+id/button2"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_alignBaseline="@+id/button1"
        android:layout_alignBottom="@+id/button1"
        android:layout_alignRight="@+id/textView1"

```

```
    android:layout_marginRight="48dp"  
    android:onClick="myClick" />
```

```
</RelativeLayout>
```

7.2.5.3. Προγραμματισμός Σαλονιού

Όπως έχει αναφερθεί πιο πάνω στο σαλόνι υπάρχει η επιλογή της αποθήκευσης της γλυκόζης Glucose.java αν ο χρήστης πατήσει πάνω στο τραπεζάκι που υπάρχει στη μέση του χώρου.

```
package com.evi.kidsdiabetes;
```

```
import android.os.Bundle;  
import android.app.Activity;  
import android.view.Menu;  
import android.content.Context;  
import android.view.View;  
import android.view.inputmethod.InputMethodManager;  
import android.widget.AdapterView;  
import android.widget.AdapterView.OnItemClickListener;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;
```

```
public class Glucose extends Activity implements OnItemClickListener{
```

```
    Button btnAdd;
```

```
    EditText inputGlucose;
```

```
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.glucose);
```

```
        btnAdd = (Button) findViewById(R.id.btn_add);  
        inputGlucose = (EditText) findViewById(R.id.input_glucose);
```

```
        btnAdd.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View arg0) {
```

```
                String glucose = inputGlucose.getText().toString();
```

```
                if (glucose.trim().length() > 0)  
                {
```

```
                    DatabaseHandler db = new DatabaseHandler(getApplicationContext());  
                    db.insertGlucose(glucose);
```

```
                    // making input filed text to blank  
                    inputGlucose.setText("");
```

```
                    // Hiding the keyboard
```



```

        InputMethodManager imm = (InputMethodManager)
            getSystemService(Context.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(inputGlucose.getWindowToken(),0);

        //Toast
        Toast.makeText(getApplicationContext(), "Η τιμή προστέθηκε!",
            Toast.LENGTH_SHORT).show();
    }
    else {
        Toast.makeText(getApplicationContext(), "Παρακαλώ βάλε τιμή",
            Toast.LENGTH_SHORT).show(); }
    }
});
}

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

@Override
public void onItemClick(AdapterView<?> parent, View view, int position,
    long id) {
    // TODO Auto-generated method stub
}

@Override
public void onNothingSelected(AdapterView<?> parent) {
    // TODO Auto-generated method stub
}
}

```

Με την εισαγωγή της γλυκόζης, καλείται η κλάση DatabaseHandler.java, και αποθηκεύεται η τιμή στη βάση δεδομένων `insertGlucose.db` στο πεδίο `glucose` και αυτομάτως με τη μέθοδο `SimpleDateFormat` συμπληρώνεται στο πεδίο `date` η ημερομηνία και η ώρα.

```

package com.evi.kidsdiabetes;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

```

```

public class DatabaseHandler extends SQLiteOpenHelper {
    // Database Version
    private static final int DATABASE_VERSION = 2;

    // Database Name
    private static final String DATABASE_NAME = "insertGlucose.db";

    // Labels table name
    private static final String TABLE_GLUCOSE = "Glucose";

    // Labels Table Columns names
    private static final String KEY_ID = "id";
    private static final String KEY_NAME = "glucose";

    private static final String KEY_DATE = "datetime";

    public DatabaseHandler(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    // Creating Tables
    @Override
    public void onCreate(SQLiteDatabase db) {
        // Category table create query
        String CREATE_CATEGORIES_TABLE = "CREATE TABLE " + TABLE_GLUCOSE + "("
            + KEY_ID + " INTEGER PRIMARY KEY," + KEY_NAME + " TEXT," + KEY_DATE +
        ")";

        db.execSQL(CREATE_CATEGORIES_TABLE);

    }

    // Upgrading database
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // Drop older table if existed
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_GLUCOSE);

        // Create tables again
        onCreate(db);
    }

    //-----Inserting new glucose into table-----//

    public void insertGlucose(String glucose){
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();
        values.put(KEY_NAME, glucose);

        // set the format to sql date time
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        Date date = new Date();
        values.put("datetime", dateFormat.format(date));

        // Inserting Row
        db.insert(TABLE_GLUCOSE, null, values);
    }
}

```

```

        db.close(); // Closing database connection
    }

    //-----Get all data from database-----//
    public List<String> getAllValues(){
        List<String> value1 = new ArrayList<String>();
        List<String> value2 = new ArrayList<String>();

        // Select All Query
        String selectQuery = "SELECT * FROM " + TABLE_GLUCOSE ;

        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery(selectQuery, null);

        if (cursor.moveToFirst()) {
            do {
                value1.add(cursor.getString(1));
                value2.add(cursor.getString(2));
            } while (cursor.moveToNext());
        }

        // returning glucose
        return value1;
    }
}

```

Η διαδικασία αποθήκευσης αυτών των τιμών πραγματοποιείται με τη μέθοδο `insertGlucose` . Επιπλέον με την μέθοδο `onUpgrade` γίνεται η απαραίτητη ανανέωση της βάσης δεδομένων. Τέλος, η `getAllValues()` βοηθάει για την αποθήκευση όλων των στοιχείων σε μια στήλη ώστε να σταλούν στην κλάση `ViewLog.java` και να προβληθεί το ιστορικό της γλυκόζης με ένα απλό `ListView`, το οποίο δηλώνεται με το `view_log.xml`.

```

package com.evi.kidsdiabetes;

import java.util.List;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class ViewLog extends Activity implements OnItemClickListener{

    ListView listView;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.view_log);

        listView = (ListView)findViewById(R.id.listView);

        listView.setOnItemClickListener(this);

        loadListData(); }
}

```

```

private void loadListData() {
    DatabaseHandler db = new DatabaseHandler(getApplicationContext());
    List<String> glucose = db.getAllValues();

    // Creating adapter
    ArrayAdapter<String> dataAdapter = new
ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, glucose);

    // attaching data adapter
    listView.setAdapter(dataAdapter);
}

@Override
public void onNothingSelected(AdapterView<?> arg0) {
    // TODO Auto-generated method stub
}

@Override
public void onItemClick(AdapterView<?> parent, View view, int position,
    long id) {
    // TODO Auto-generated method stub
}
}
}

```

Κατά τον έλεγχο της λειτουργίας της βάσης δεδομένων SQLite χρειάστηκε να εγκατασταθεί ένα plugin στο Eclipse, το com.questoid.sqlitemanager_1.0.0.jar. Έτσι μπαίνοντας στο DDMS ακολουθώντας το data/data/com.evi.kidsdiabetes/databases/com.evi.kidsdiabetes εμφανίζεται όλος ο πίνακας με τις όλες τις καταγραφές που έχουν γίνει.

7.2.5.4. Προγραμματισμός Κουζίνα

Στην κουζίνα, πατώντας πάνω στην φρουτιέρα εμφανίζεται μια ListView με τα όλα τα φρούτα και από δίπλα πληροφορίες σχετικά με την περιεκτικότητα τους σε υδατάνθρακες. Αυτή η διαδικασία δημιουργήκε με την χρήση δύο κλάσεων, Fruits και FruitsAdapter.

Στην Fruits.java δηλώνεται ένας πίνακας με τις πληροφορίες του κάθε φρούτου και καλείται η κλάση FruitsAdapter.java.

```

package com.evi.kidsdiabetes;

import android.app.ListActivity;
import android.os.Bundle;
import android.widget.ListView;
import android.widget.Toast;
import android.view.View;

```

```

public class Fruits extends ListActivity {

    static final String[] FRUITS = new String[] { "1 Αχλάδι = 27,5γρ. υδατάνθρακες",
        "1 Βερύκοκο = 11γρ. υδατάνθρακες",
        "Μισό φλιτζάνι Φράουλες = 6,5γρ. υδατάνθρακες", "Μισό φλιτζάνι Καρπούζι =
        5,5γρ. υδατάνθρακες", "Μισό φλιτζάνι Κεράσια = 11γρ. υδατάνθρακες",
        "1 Μήλο = 25γρ. υδατάνθρακες", "1 Μπανάνα = 27γρ. υδατάνθρακες", "Μισό
        φλιτζάνι Πεπόνι = 8γρ. υδατάνθρακες", "1 Πορτοκάλι = 15,5γρ.
        υδατάνθρακες",
        "1 Ροδάκινο = 14,5 γρ. υδατάνθρακες", "Μισό φλιτζάνι Σταφύλια = 14γρ.
        υδατάνθρακες", "1 Ακτινίδιο = 11γρ. υδατάνθρακες"};

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setListAdapter(new FruitsAdapter(this, FRUITS));
    }

    @Override
    protected void onItemClick(ListView l, View v, int position, long id) {

        //get selected items
        String selectedValue = (String) getListAdapter().getItem(position);
        Toast.makeText(this, selectedValue, Toast.LENGTH_SHORT).show();
    }
}

```

Στην κλάση που φαίνεται παρακάτω, φορτώνεται η εικόνα στο σημείο που έχει δηλωθεί το ImageView και το κείμενο από την προηγούμενη κλάση στο πεδίο TextView.

```

package com.evi.kidsdiabetes;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

public class FruitsAdapter extends ArrayAdapter<String> {
    private final Context context;
    private final String[] values;

    public FruitsAdapter(Context context, String[] values) {
        super(context, R.layout.fruits, values);
        this.context = context;
        this.values = values;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = (LayoutInflater) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        View rowView = inflater.inflate(R.layout.fruits, parent, false);
    }
}

```

```

TextView textView = (TextView) rowView.findViewById(R.id.txt);
ImageView imageView = (ImageView) rowView.findViewById(R.id.fruits);
textView.setText(values[position]);

// Change icon based on name
String s = values[position];

System.out.println(s);

if (s.equals("1 Αχλάδι = 27,5γρ. υδατάνθρακες")) {
    imageView.setImageResource(R.drawable.axladi);
} else if (s.equals("1 Βερύκοκο = 11γρ. υδατάνθρακες")) {
    imageView.setImageResource(R.drawable.berikoko);
} else if (s.equals("Μισό φλιτζάνι Φράουλες = 6,5γρ. υδατάνθρακες")) {
    imageView.setImageResource(R.drawable.fraoyla);
} else if (s.equals("Μισό φλιτζάνι Καρπούζι = 5,5γρ. υδατάνθρακες")) {
    imageView.setImageResource(R.drawable.karpouzi);
} else if (s.equals("Μισό φλιτζάνι Κεράσια = 11γρ. υδατάνθρακες")) {
    imageView.setImageResource(R.drawable.kerasia);
} else if (s.equals("1 Μήλο = 25γρ. υδατάνθρακες")) {
    imageView.setImageResource(R.drawable.mhlo);
} else if (s.equals("1 Μπανάνα = 27γρ. υδατάνθρακες")) {
    imageView.setImageResource(R.drawable.mpanana);
} else if (s.equals("Μισό φλιτζάνι Πεπόνι = 8γρ. υδατάνθρακες")) {
    imageView.setImageResource(R.drawable.peponi);
} else if (s.equals("1 Ποτοκάλι = 15,5γρ. υδατάνθρακες")) {
    imageView.setImageResource(R.drawable.portokali);
} else if (s.equals("1 Ροδάκινο = 14,5 γρ. υδατάνθρακες")) {
    imageView.setImageResource(R.drawable.rodakino);
} else if (s.equals("Μισό φλιτζάνι Σταφύλια = 14γρ. υδατάνθρακες")) {
    imageView.setImageResource(R.drawable.stafylia);
} else {
    imageView.setImageResource(R.drawable.aktinidio);
}

return rowView;
}
}

```

Το layout της φρουτιέρας είναι ουσιαστικά μια λίστα που περιέχει εικόνα και κείμενο.

```

<?xml version="1.0" encoding="UTF-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FBB117"
    android:padding="5dp" >

<ImageView
    android:layout_height="200px"
    android:layout_width="200px"
    android:src="@drawable/axladi"
    android:layout_marginTop="100px"
    android:layout_marginRight="50px"
    android:layout_marginLeft="50px"
    android:id="@+id/fruits">
</ImageView>

<TextView
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"

```

```
android:id="@+id/txt"  
android:textSize="50px"  
android:text="@+id/Label">  
</TextView>
```

```
</LinearLayout>
```

Από την άλλη, μέσα στο ψυγείο υπάρχουν πέντε διακριτές κατηγορίες τροφίμων. Επιλέγοντας μια κατηγορία τροφίμου εμφανίζεται GridView αυτών που περιλαμβάνονται σε αυτήν. Με την επιλογή ενός συγκεκριμένου τροφίμου ο χρήστης μέσω ενός Toast ενημερώνεται για την ποσότητα των υδατανθράκων που περιέχεται σε αυτό. Στην κλάση FoodGuide.java δηλώνονται σε δύο πίνακες οι εικόνες και οι αντίστοιχες πληροφορίες τους, ενώ στην συνέχεια ανάλογα με την επιλεγμένη κατηγορία καλείται η κλάση FoodAdapter.

```
package com.evi.kidsdiabetes;
```

```
import android.os.Bundle;  
import android.app.Activity;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.GridView;  
import android.widget.Toast;
```

```
public class FoodGuide extends Activity {  
    GridView grid;
```

```
    String[] Carbohydrates = {  
        "1 φλιτζάνι Μακαρόνια = 45gr υδατάνθρακες",  
        "1 φλιτζάνι Ρύζι = 45gr υδατάνθρακες",  
        "1 φέτα Ψωμί = 15gr υδατάνθρακες",  
        "1 φέτα Ψωμί Ολικής = 15gr υδατάνθρακες",  
        "1 Κουλούρι Θεσσαλονίκης = 47,5gr υδατάνθρακες",  
        "1 βραστή Πατάτα = 19.8gr υδατάνθρακες"} ;
```

```
    int[] imageCarbohydrates = {  
        R.drawable.makaronia,  
        R.drawable.ryzi,  
        R.drawable.psomi,  
        R.drawable.psomi_olikhs,  
        R.drawable.koyloyria,  
        R.drawable.patata};
```

```
    String[] Vegetables = {  
        "Αγγούρι = 3,6gr υδατάνθρακες",  
        "Ντομάτα = 4gr υδατάνθρακες",  
        "Φασολάκια = 5gr υδατάνθρακες",  
        "μισό Καλαμπόκι = 15gr υδατάνθρακες",  
        "Καρότο = 5gr υδατάνθρακες",  
        "Μαρουλι = 4.6gr υδατάνθρακες",  
        "Πιπεριά = 4.6gr υδατάνθρακες",  
        "Παντζάρι = 7.2gr υδατάνθρακες",  
        "Μπρόκολο = 1.6gr υδατάνθρακες",  
        "1 πιάτο Φακές = 13,7gr υδατάνθρακες",
```

```

        "1 πιάτο Αρακά = 12,9gr υδατάνθρακες"} ;
int[] imageVegetables = {
    R.drawable.aggoyri,
    R.drawable.ntomata,
    R.drawable.fasolakia,
    R.drawable.kalampoki,
    R.drawable.karoto,
    R.drawable.maroyli,
    R.drawable.piperia,
    R.drawable.pantzari,
    R.drawable.mprokolo,
    R.drawable.fakes,
    R.drawable.arakas};

String[] Dairy = {
    "Γάλα = 12gr υδατάνθρακες",
    "Σοκολατούχο Γάλα = 25gr υδατάνθρακες",
    "Γιαουρτι = 30gr υδατάνθρακες"} ;
int[] imageDairy = {
    R.drawable.gala,
    R.drawable.kakao,
    R.drawable.giaoyrti };

String[] Protein = {
    "Σαλάμι = 0gr υδατάνθρακες",
    "Τυρί = 0gr υδατάνθρακες",
    "Κοτόπουλο = 0gr υδατάνθρακες",
    "Αυγό = 0gr υδατάνθρακες",
    "Μπιφτέκι = 0gr υδατάνθρακες",
    "Μπριζόλα = 0gr υδατάνθρακες",
    "Ψάρι = 0gr υδατάνθρακες",
    "Hot-Dog = 30gr υδατάνθρακες",
    "Burger = 30gr υδατάνθρακες"} ;
int[] imageProtein = {
    R.drawable.salami,
    R.drawable.tyri,
    R.drawable.kotopoulo,
    R.drawable.abga,
    R.drawable.mpifteki,
    R.drawable.mprizola,
    R.drawable.psari,
    R.drawable.hotdog,
    R.drawable.burger};

String[] Other = {
    "1 κουτάκι Αναψυκτικό = 40gr υδατάνθρακες",
    "1 μικρό Κεκάκι = 30gr υδατάνθρακες",
    "1 κουταλάκι Μαρμελάδα = 15gr υδατάνθρακες",
    "1 κουταλάκι Πραλίνα = 15gr υδατάνθρακες",
    "2 μικρά Μπισκότα = 15gr υδατάνθρακες",
    "1 μπάλα Παγωτό = 15gr υδατάνθρακες",
    "2 μικρά κομμάτια Πίτσα = 30gr υδατάνθρακες",
    "1 μικρό σακουλάκι Πατατάκια = 50gr υδατάνθρακες",
    "100 γραμμάρια Σοκολάτα = 60gr υδατάνθρακες",
    "1 Κρουασάν = 30gr υδατάνθρακες"} ;
int[] imageOther = {
    R.drawable.colas,
    R.drawable.keik,
    R.drawable.marmelada,
    R.drawable.merenta,
    R.drawable.mpiskota,
    R.drawable.pagoto,

```



```
R.drawable.pitsa,  
R.drawable.patatakia,  
R.drawable.sokolata,  
R.drawable.kroyasan};
```

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.foodguide);  
  
    Bundle extra = getIntent().getExtras();  
    String Category = extra.getString("Choice");  
  
    if (Category.equals("Υδατάνθρακες")) {  
  
        FoodAdapter adapter = new FoodAdapter(FoodGuide.this, Carbohydrates,  
imageCarbohydrates);  
  
        grid=(GridView)findViewById(R.id.gridView1);  
        grid.setAdapter(adapter);  
  
        grid.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
            @Override  
            public void onItemClick(AdapterView<?> parent, View view, int position,  
long id) {  
                Toast.makeText(FoodGuide.this, Carbohydrates[+ position],  
Toast.LENGTH_SHORT).show();  
            }  
        });  
    }else if(Category.equals("Λαχανικά")) {  
  
        FoodAdapter adapter = new FoodAdapter(FoodGuide.this, Vegetables, imageVegetables);  
  
        grid=(GridView)findViewById(R.id.gridView1);  
        grid.setAdapter(adapter);  
  
        grid.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
            @Override  
            public void onItemClick(AdapterView<?> parent, View view, int  
position, long id) {  
                Toast.makeText(FoodGuide.this, Vegetables[+ position],  
Toast.LENGTH_SHORT).show();  
            }  
        });  
    }else if(Category.equals("Γαλακτοκομικά")){  
  
        FoodAdapter adapter = new FoodAdapter(FoodGuide.this, Dairy, imageDairy);  
  
        grid=(GridView)findViewById(R.id.gridView1);  
        grid.setAdapter(adapter);  
  
        grid.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
            @Override  
            public void onItemClick(AdapterView<?> parent, View view, int  
position, long id) {  
                Toast.makeText(FoodGuide.this, Dairy[+ position],  
Toast.LENGTH_SHORT).show();  
            }  
        });  
    }else if(Category.equals("Κρέας")){
```

```

FoodAdapter adapter = new FoodAdapter(FoodGuide.this, Protein, imageProtein);

grid=(GridView)findViewById(R.id.gridView1);
grid.setAdapter(adapter);

    grid.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
            Toast.makeText(FoodGuide.this, Protein[+ position],
Toast.LENGTH_SHORT).show();
        }
    });
}else if(Category.equals("Διάφορα")){

    FoodAdapter adapter = new FoodAdapter(FoodGuide.this, Other, imageOther);

    grid=(GridView)findViewById(R.id.gridView1);
    grid.setAdapter(adapter);

        grid.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
                Toast.makeText(FoodGuide.this, Other[+ position],
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
}
}

```

Τέλος στην FoodAdapter τοποθετούνται στις αντίστοιχες θέσεις οι εικόνες.

```

package com.evi.kidsdiabetes;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;

public class FoodAdapter extends BaseAdapter {
    private Context mContext;
    private final String[] food;
    private final int[] Imageid;

    public FoodAdapter(Context c,String[] food,int[] Imageid ) {
        mContext = c;
        this.Imageid = Imageid;
        this.food = food;
    }

    @Override
    public int getCount() {
        // TODO Auto-generated method stub
        return food.length;
    }
}

```

```

}

@Override
public Object getItem(int position) {
    // TODO Auto-generated method stub
    return null;
}

@Override
public long getItemId(int position) {
    // TODO Auto-generated method stub
    return 0;
}

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    // TODO Auto-generated method stub
    //View grid;
    LayoutInflater inflater = (LayoutInflater) mContext
        .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    View grid = inflater.inflate(R.layout.food, parent, false);
    if (convertView == null) {
        grid = new View(mContext);
        grid = inflater.inflate(R.layout.food, null);

        } else {
        grid = (View) convertView;
        }
    ImageView imageView = (ImageView)grid.findViewById(R.id.grid_image);
    imageView.setImageResource(Imageid[position]);
    return grid;
}
}

```

Για την προβολή του GridView χρειάστηκα δύο layouts, το foodguide.xml για τον ορισμό μόνο του gridView και το food.xml για τις εικόνες και το textview.

```

<?xml version="1.0" encoding="utf-8"?>

<GridView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/gridView1"
    android:numColumns="2"
    android:gravity="center"
    android:columnWidth="350dp"
    android:stretchMode="columnWidth"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/psugeio1">

</GridView>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="10dp" >

<ImageView
    android:id="@+id/grid_image"

```

```
        android:layout_width="350dp"  
        android:layout_height="350dp">  
    </ImageView>  
  
</LinearLayout>
```

7.3. Προσαρμογή και ενσωμάτωση σε Android συσκευές

Χρησιμοποιήθηκαν τρεις διαφορετικές κινητές συσκευές για τον έλεγχο της εφαρμογής:

- ✓ Sony Xperia U (Android 4.0.4, 3.5", Dual-core 1 GHz Cortex-A9, 4GB RAM)
- ✓ Turbo-X Tablet Rainbow (Android 4.2, 9.7", Quad Core Rockchip RK3188 1.6 GHz, 2GB RAM)
- ✓ Samsung Galaxy Note 10.1, 2014 Edition (Android 4.3, 10.1", Quad-core 1.9 GHz Cortex-A15 & Quad-core 1.3 GHz Cortex-A7, 16GB RAM)

Λόγω της διαφοράς των εκδόσεων του λειτουργικού συστήματος (Android) που διαθέτει η κάθε κινητή συσκευή, δεν χρησιμοποιήθηκαν στοιχεία τα οποία δεν υποστηρίζονται από την έκδοση Android 4.0. Το μοναδικό πρόβλημα που παρουσιάστηκε ήταν η αλλαγή του μεγέθους, αλλά και παραμόρφωσης της εικόνας καθώς επίσης και η διαφορά της θέσης κάποιων στοιχείων μέσα στην οθόνη.

8. ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΤΙΩΣΕΙΣ

Οι μελλοντικές βελτιώσεις θα μπορούσαν να αφορούν τα γραφικά, ώστε να γίνουν πιο ελκυστικά για τα παιδιά. Με την προσθήκη κίνησης ή ακόμη και ηχητικών εφέ, η εφαρμογή θα γινόταν πιο διασκεδαστική και ενδιαφέρουσα για τους μικρούς χρήστες.

Επιπλέον, υπάρχει η σκέψη για εμπλοκίση των υπαρχόντων παιχνιδιών καθώς και αύξηση του αριθμού τους. Όπως π.χ κάποιο παιχνίδι με κρυπτόλεξο ή η δημιουργία ενός γεύματος, έτσι το παιδί να νιώθει ότι μπορεί να συμμετέχει και να διαχειρίζεται τη διατροφή του.

Τέλος, να υπάρξει κάποια συνεργασία με διαβητολόγο επομένως θα μπορέσει αυτή η εφαρμογή να προσεγγίσει πιο ειδικευμένα τον παιδικό διαβήτη, δηλαδή να μπορεί να συμβουλευτείται ένα παιδί την εφαρμογή ώστε ανάλογα με τα επίπεδα της γλυκόζης στον οργανισμό του, να του δίνονται οδηγίες για την χορήγηση ινσουλίνης.

9. Βιβλιογραφία

1. Danny Meetoo, Harry Gopaul, (2005). Empowerment: giving power to people with diabetes, *Journal of Diabetes Nursing* Vol 9 No 1: 28-31.
2. Deterding S., Dixon D., Khaled R., Nacke L., (2011). From Game Design Elements to Gamefulness : Defining “Gamification”, *MindTrek '11 Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*: 9-15.
3. Hebecker R., Schulz S., Luimula M., Czanner S.. A playful interface for children's diabetes self-care.
4. Miller A. S., Cafazzo J. A., & Seto E. (2014). A game plan: Gamification design principles in mHealth applications for chronic disease management, *Health informatics journal*, 1460458214537511.
5. Zichermann G., & Cunningham C. (2011). Gamification by design: Implementing game mechanics in web and mobile apps. " O'Reilly Media, Inc."
6. Deterding S., Sicart M., Nacke L., O'Hara K., & Dixon D. (2011, May). Gamification. using game-design elements in non-gaming contexts, *CHI'11 Extended Abstracts on Human Factors in Computing Systems*: 2425-2428.
7. Funnel MM., Anderson R. (2004). Empowerment and Self-Management of Diabetes, *CLINICAL DIABETES - Volume 22, Number 3*: 123-127.
8. Wahil P., & Karim A. How to Make Healthcare Wellness Programs More Effective?, *Gamification: Rediscover the Power of Engagement*, 40-46.
9. Frøisland, D. H., Årsand, E., & Skårderud, F. (2012). Improving diabetes care for young people with type 1 diabetes through visual learning on mobile phones: mixed-methods study. *Journal of medical Internet research*: 14(4), e111.
10. Thompson, D., Baranowski, T., Buday, R., Baranowski, J., Thompson, V., Jago, R., & Griffith, M. J. (2008). Serious video games for health: how behavioral science guided the design of a game on diabetes and obesity. *Simulation & gaming*.
11. Stone, E. M. (2013). *Dia-BEAT-it: An online game for children and teenagers with type 1 diabetes*.
12. Κατευθυντήριες Οδηγίες για τη Διαχείριση του Διαβητικού Ασθενούς, Ελληνική Διαβητολογική Εταιρία (2013).

10. Ιστογραφία

1. <http://likestinygeia.eu/>
2. <http://mashable.com/category/gamification/>
3. <http://www.pmjournal.gr/gamification-health-pharma/>
4. <http://en.wikipedia.org/wiki/Gamification>
5. http://eikonikhpragmatikothtakaiugeia.blogspot.gr/2013/05/blog-post_3133.html
6. <http://www.suit.gr/what-gamification-is-not/>
7. <http://www.messaggiamo.com/el/pc-games/67052-gamification-what-it-is-and-what-it-means-for-you.html>
8. <http://www.diabetes-in-greece.gr/website/content/living-with-diabetes/living-with-type-2/self-management.aspx>
9. <http://www.mednet.gr/archives/2005-4/pdf/336.pdf>
10. <http://www.forbes.com/sites/ey/2015/02/20/video-the-venture-capital-investor-and-entrepreneur-relationship/>
11. <http://searchhealthit.techtarget.com/opinion/Gamification-in-healthcare-isnt-just-about-playing-games>
12. <http://timesulin.com/theblog/3-diabetes-games-will-educate-kids-diabetes/>
13. <http://www.frankelgroup.com/insights/blog/posts/from-insight-to-impact-the-frankel-blog/2013/10/28/the-healthcare-arena-a-new-venue-for-fun-games>
14. <http://timesulin.com/theblog/diabetes-and-kids-how-to-explain-type-1-diabetes-to-your-child/>
15. <http://timesulin.com/theblog/5-tips-for-managing-type-1-diabetes-in-children/>
16. <http://patients.about.com/od/empowermentbasics/a/wisepatient.htm>